

СПРАВОЧНИК МАГНИТНОГО ДИСКА
(кафедра полупроводниковой электроники
и нанoeлектроники)

В. А. Буслов Т.В. Пашнева

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ

Учебное пособие КТвНиО.doc 4,02Кбайт 22.12.2008 20,0 уч.-изд.л.
(наименование файла) (объем файла) (дата) (объем издания)

В. А. Буслов Т.В. Пашнева

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ

Учебное пособие



Воронеж 2008

ГОУВПО «Воронежский государственный
технический университет»

В. А. Буслов Т.В. Пашнева

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ
В НАУКЕ И ОБРАЗОВАНИИ

Утверждено Редакционно-издательским советом
университета в качестве учебного пособия

Воронеж 2008

УДК 22.183.4

Буслов В.А. Компьютерные технологии в науке и образовании: учеб. пособие/ В.А. Буслов, Пашнева Т.В. Воронеж: ГОУВПО «Воронежский государственный технический университет», 2008. 373 с.

В учебном пособии рассматриваются место компьютерных технологий в современной науке и системе образования.

Учебное пособие соответствует требованиям Государственного образовательного стандарта высшего профессионального образования по направлению 140400 «Техническая физика», специализации 140402 “Физика и техника полупроводников”, дисциплине «Компьютерные технологии в науке и образовании». Учебное пособие подготовлено в электронном виде в текстовом редакторе MS Word for Windows и содержится в файле Учебное пособиеКТвНиО.exe.

Ил. 115. Табл. 2. Библиогр.: 43 назв.

Научный редактор д-р физ.-мат. наук, проф. С.И. Рембеза

Рецензенты: кафедра физики Воронежской государственной технологической академии (зав. кафедрой д-р физ.-мат. наук, профессор Н.Н. Безрядин);
д-р физ.-мат. наук, проф. В.И. Митрохин

© Буслов В. А., Пашнева Т.В., 2008

© Оформление. ГОУВПО «Воронежский государственный технический университет», 2008

ВВЕДЕНИЕ

В последнее время встает вопрос - как использовать компьютерные технологии в обучении и нужны ли они, ведь раньше спокойно обходились и без них. Но прогресс не стоит на месте, а значит, и отказываться от новых внедрений нецелесообразно.

Современные компьютерные технологии предоставляют огромные возможности для развития процесса образования. Еще К. Д. Ушинский заметил: "Детская природа требует наглядности". Сейчас это уже не схемы, таблицы и картинки - они статичны, а более близкая детской природе игра, пусть даже и научно-познавательная.

Информационные технологии позволяют реализовывать принципы дифференцированного и индивидуального подхода к обучению. На занятии преподаватель дает возможность каждому обучаемому самостоятельно работать с учебной информацией, что позволяет ему детально разобрать новый материал по своей схеме. Информационные технологии можно использовать как для очного, так и для дистанционного обучения. Они дают возможность реализовывать мировые тенденции в образовании, возможности выхода в единое мировое информационное пространство. Применение компьютерных технологий позволяет повысить уровень самообразования, мотивации учебной деятельности; дает совершенно новые возможности для творчества, обретения и закрепления различных профессиональных навыков, и, конечно, соответствует социальному заказу, который государство предъявляет к высшей школе.

Используя системы мультимедиа, позволяющие объединить возможности компьютера и знания учителя, стало возможным создание электронных учебников, которые более наглядно, красочно и с мобильным доступом информации

предстанут перед учениками. Содержание учебника включает в себя такие разделы как: теоретическая часть, контрольные задания, лабораторные работы, курсовые работы, вопросы для самопроверки, что присуще и традиционному учебнику, но электронный учебник более компактен (его объем позволяет полностью разместиться на одном диске), в содержании его могут использоваться видеофильмы и фрагменты звука, что придает ему большую привлекательность и оснащенность. В обращении он прост и позволяет ученику легко вернуться к той информации, которую он не понял. Нажать кнопки клавиатуры гораздо быстрее и проще, чем перелистывать страницы учебника назад. Такое новое учебное пособие проявляет у учеников и студентов большой интерес к учебе и желание изучать предмет в более углубленной форме, что, несомненно, сказывается на их уровне знаний. Способ ведения урока, когда учитель связан с каждым учеником класса единой компьютерной сетью, позволяет учителю более детально и персонально подойти к вопросу обучения ученика, проверить его уровень знаний и наверстать "пробелы" в его знаниях.

Мультимедийные технологии открывают возможности преподавателям отказаться от свойственных традиционному обучению рутинных видов деятельности преподавания, предоставив ему возможность использовать интеллектуальные формы труда, освобождают от изложения значительной части учебного материала и рутинных операций, связанных с отработкой умений и навыков.

Информационные технологии позволяют учителям с высокой скоростью обмениваться опытом, благодаря дистанционному общению, а также повышать квалификацию и познавать новые методы обучения.

Благодаря новым мультимедиа технологиям, стало возможным использовать компьютерные программы как иллюстративный материал, проводить тестирование и

контрольные работы, решать творческие задачи, участвовать в дистанционных уроках, сочетать традиционные домашние задания с заданиями, для выполнения которых используются компьютеры, создавать уроки-игры для каждого ученика и др.

Компьютер предлагает поиграть, проверяя знания и в конечном итоге выводя оценку. Такой подход к учебной деятельности снимает внутреннюю напряженность, которая свойственна детям во время проверочной или контрольной работ. Дает точные результаты проверки знаний. И облегчает учителям проведение самостоятельной работы со школьниками и студентами.

Внедряя новые технологии в учебный процесс, мы даем возможность ученику не только лучше выучить предмет, но и научиться свободно владеть компьютером.

Многие задания по предмету в компьютерном варианте, позволяют развить творческие способности ученика, взглянуть на предмет с другой стороны и проявить себя в новой деятельности.

Несомненные плюсы внедрения новейших информационных технологий позволяют сделать новый шаг к будущему, где компьютер будет средством реализации своих возможностей и талантов.

Для этого в настоящее время надо активно работать над созданием новых методик преподавания, которые будут направлены на расширение кругозора ученика, повышение его уровня знаний по предмету, развитие творческих способностей, а также на обучение свободному и грамотному владению компьютером.

К сожалению, пока наше общество еще сталкивается с проблемой нехватки квалифицированных кадров, которых нужно воспитывать со студенческой скамьи. Острой становится проблема невладения компьютером самих преподавателей. Необходима переподготовка учителей в

соответствии с новыми требованиями к обучению. Объединить опыт и знания педагога с возможностями компьютера - главная задача перехода на новый уровень образования в современных условиях.

Внедрение информационных технологий в образование дает возможность выбора оптимального набора технологий для организации учебного процесса. При выборе необходимо учитывать их соответствие индивидуальным качествам обучаемых и специфическим особенностям конкретных предметных областей.

При работе с мультимедийными технологиями учащиеся с самого начала вовлечены в активную познавательную деятельность. В ходе такого обучения они учатся не только приобретать и применять знания, но и находить необходимые для них средства обучения и источники информации, уметь работать с этой информацией.

В условиях, когда компьютер занимает все более важную и неотъемлемую часть в современном обществе, использование компьютерных технологий в учебном процессе позволяет обеспечить будущее страны грамотным поколением, способным разрабатывать и внедрять новые идеи во все сферы наук. /1/

1. КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В СОВРЕМЕННОМ ОБЩЕСТВЕ

1.1 Представление об информационном обществе

В истории развития цивилизации произошло несколько информационных революций - преобразований общественных отношений из-за кардинальных изменений в сфере обработки информации. Следствием подобных преобразований являлось приобретение человеческим обществом нового качества.

Первая революция связана с изобретением письменности, что привело к гигантскому качественному и количественному скачку. Появилась возможность передачи знаний от поколения к поколениям.

Вторая (середина XVI в.) вызвана изобретением книгопечатания, которое радикально изменило индустриальное общество, культуру, организацию деятельности.

Третья (конец XIX в.) обусловлена изобретением электричества, благодаря которому появились телеграф, телефон, радио, позволяющие оперативно передавать и накапливать информацию в любом объеме.

Четвертая (70-е гг. XX в.) связана с изобретением микропроцессорной технологии и появлением персонального компьютера. На микропроцессорах и интегральных схемах создаются компьютеры, компьютерные сети, системы передачи данных (информационные коммуникации). Этот период характеризуют три фундаментальные инновации:

- переход от механических и электрических средств преобразования информации к электронным;
- миниатюризация всех узлов, устройств, приборов, машин;
- создание программно-управляемых устройств и процессов.

Для создания более целостного представления об этом периоде целесообразно познакомиться с приведенной ниже справкой о смене поколений электронно-вычислительных машин (ЭВМ) и сопоставить эти сведения с этапами в области обработки и передачи информации:

- 1-е поколение (начало 50-х гг.). Элементная база – электронные лампы. ЭВМ отличались большими габаритами, большим потреблением энергии, малым быстродействием, низкой надежностью, программированием в кодах.
- 2-е поколение (с конца 50-х гг.). Элементная база –

полупроводниковые элементы. Улучшились по сравнению с ЭВМ предыдущего поколения все технические характеристики. Для программирования используются алгоритмические языки.

- 3-е поколение (начало 60-х гг.). Элементная база – интегральные схемы, многослойный печатный монтаж. Резкое снижение габаритов ЭВМ, повышение их надежности, увеличение производительности. Доступ с удаленных терминалов.

- 4-е поколение (с середины 70-х гг.). Элементная база – микропроцессоры, большие интегральные схемы. Улучшились технические характеристики. Массовый выпуск персональных компьютеров. Направления развития: мощные многопроцессорные вычислительные системы с высокой производительностью, создание дешевых микроЭВМ.

- 5-е поколение (с середины 80-х гг.). Началась разработка интеллектуальных компьютеров, пока не увенчавшаяся успехом. Внедрение во все сферы компьютерных сетей и их объединение, использование распределенной обработки данных, повсеместное применение компьютерных информационных технологий.

Последняя информационная революция выдвигает на первый план новую отрасль - информационную индустрию, связанную с производством технических средств, методов, технологий для производства новых знаний. Важнейшими составляющими информационной индустрии становятся все виды информационных технологий, особенно телекоммуникации. Современная информационная технология опирается на достижения в области компьютерной техники и средств связи. В связи с этим дадим некоторые определения:

Информация – организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационной

потребностей и реализации прав граждан.

Информационные ресурсы – отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах.

Информационная система – это организационно упорядоченная совокупность документов, информации технологий, в том числе и использование средств вычислительной техники и связей реализующих информационные процессы.

Пользователь информации – субъект, обращающейся к информационной системе или к посреднику за необходимой ему информацией.

Информационная технология (ИТ) - процесс, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления.

Телекоммуникации - дистанционная передача данных на базе компьютерных сетей и современных технических средств связи.

Усложнение индустриального производства, социальной, экономической и политической жизни, изменение динамики процессов во всех сферах деятельности человека привели, с одной стороны, к росту потребностей в знаниях, а с другой – к созданию новых средств и способов удовлетворения этих потребностей.

Бурное развитие компьютерной техники и информационных технологий послужило толчком к развитию общества, построенного на использовании различной информации и получившего название информационного общества.

1.2 Как понимают ученые информационное общество

Японские ученые считают, что в информационном обществе процесс компьютеризации даст людям доступ к надежным источникам информации, избавит их от рутинной работы, обеспечит высокий уровень автоматизации обработки информации в производственной и социальной сферах. Движущей силой развития общества должно стать производство информационного, а не материального продукта. Материальный же продукт станет более информационно емким, что означает увеличение доли инноваций, дизайна и маркетинга в его стоимости.

В информационном обществе изменятся не только производство, но и весь уклад жизни, система ценностей, возрастет значимость культурного досуга по отношению к материальным ценностям. По сравнению с индустриальным обществом, где все направлено на производство и потребление товаров, в информационном обществе производятся и потребляются интеллект, знания, что приводит к увеличению доли умственного труда. От человека потребуется способность к творчеству, возрастет спрос на знания.

Материальной и технологической базой информационного общества станут различного рода системы на базе компьютерной техники и компьютерных сетей, информационной технологии, телекоммуникационной связи.

Информационное общество – общество, в котором большинство работающих занято производством, хранением, переработкой и реализацией информации, особенно высшей ее формы – знаний.

В реальной практике развития науки и техники передовых стран в конце XX в. постепенно приобретает зримые очертания созданная теоретиками картина

информационного общества. Прогнозируется превращение всего мирового пространства в единое компьютеризированное и информационное сообщество людей, проживающих в электронных квартирах и коттеджах. Любое жилище оснащено всевозможными электронными приборами и компьютеризированными устройствами. Деятельность людей будет сосредоточена главным образом на обработке информации, а материальное производство и производство энергии будет возложено на машины.

Уже опубликован ряд фактических материалов, свидетельствующих, что это не утопия, а неизбежная реальность недалекого будущего.

При переходе к информационному обществу возникает новая индустрия переработки информации на базе компьютерных и телекоммуникационных информационных технологий.

Ряд ученых выделяют характерные черты информационного общества:

- решена проблема информационного кризиса, т.е. разрешено противоречие между информационной лавиной и информационным голодом;
- обеспечен приоритет информации по сравнению с другими ресурсами;
- главной формой развития станет информационная экономика;
- в основу общества будут заложены автоматизированные генерация, хранение, обработка и использование знаний с помощью новейшей информационной техники и технологии;
- информационная технология приобретет глобальный характер, охватывая все сферы социальной деятельности человека;
- формируется информационное единство всей человеческой цивилизации;

- с помощью средств информатики реализован свободный доступ каждого человека к информационным ресурсам всей цивилизации;

- реализованы гуманистические принципы управления обществом и воздействия на окружающую среду.

Кроме положительных моментов прогнозируются и опасные тенденции:

- все большее влияние на общество средств массовой информации;

- информационные технологии могут разрушить частную жизнь людей и организаций;

- существует проблема отбора качественной и достоверной информации;

- многим людям будет трудно адаптироваться к среде информационного общества. Существует опасность разрыва между "информационной элитой" (людьми, занимающимися разработкой информационных технологий) и потребителями.

Ближе всех на пути к информационному обществу стоят страны с развитой информационной индустрией, к числу которых следует отнести США, Японию, Англию, Германию, страны Западной Европы. В этих странах уже давно одним из направлений государственной политики является направление, связанное с инвестициями и поддержкой инноваций в информационную индустрию, в развитие компьютерных систем и телекоммуникаций.

1.3 Роль информатизации в развитии общества

Деятельность отдельных людей, групп, коллективов и организаций сейчас все в большей степени начинает зависеть от их информированности и способности эффективно использовать имеющуюся информацию. Прежде чем предпринять какие-то действия, необходимо провести большую работу по сбору и переработке информации, ее

осмыслению и анализу. Отыскание рациональных решений в любой сфере требует обработки больших объемов информации, что подчас невозможно без привлечения специальных технических средств.

Возрастание объема информации особенно стало заметно в середине XX в. Лавинообразный поток информации хлынул на человека, не давая ему возможности воспринять эту информацию в полной мере. В ежедневно появляющемся новом потоке информации ориентироваться становилось все труднее. Подчас выгоднее стало создавать новый материальный или интеллектуальный продукт, нежели вести розыск аналога, сделанного ранее. Образование больших потоков информации обуславливается:

- чрезвычайно быстрым ростом числа документов, отчетов, диссертаций, докладов и т.п., в которых излагаются результаты научных исследований и опытно-конструкторских работ;

- постоянно увеличивающимся числом периодических изданий по разным областям человеческой деятельности;

- появлением разнообразных данных (метеорологических, геофизических, медицинских, экономических и др.), записываемых обычно на магнитных лентах и поэтому непопадающих в сферу действия системы коммуникации. Как результат – наступает информационный кризис (взрыв), который имеет следующие проявления:

- появляются противоречия между ограниченными возможностями человека по восприятию и переработке информации и существующими мощными потоками и массивами хранящейся информации. Так, например, общая сумма знаний менялась вначале очень медленно, но уже с 1900 г. она удваивалась каждые 50 лет, к 1950 г. удвоение происходило каждые 10 лет, к 1970 г. – уже каждые 5 лет, с 1990 г. – ежегодно;

- существует большое количество избыточной информации, которая затрудняет восприятие полезной для потребителя информации;

- возникают определенные экономические, политические и другие социальные барьеры, которые препятствуют распространению информации. Например, по причине соблюдения секретности часто необходимой информацией не могут воспользоваться работники разных ведомств.

Эти причины породили весьма парадоксальную ситуацию – в мире накоплен громадный информационный потенциал, но люди не могут им воспользоваться в полном объеме в силу ограниченности своих возможностей. Информационный кризис поставил общество перед необходимостью поиска путей выхода из создавшегося положения. Внедрение ЭВМ, современных средств переработки и передачи информации в различные сферы деятельности послужило началом нового эволюционного процесса, называемого *информатизацией*, в развитии человеческого общества, находящегося на этапе индустриального развития.

Информатизация общества – организованный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления, организаций, общественных объединений на основе формирования и использования информационных ресурсов.

Рассмотрим этот процесс более подробно.

История развития информатизации началась в США с 60-х гг., затем с 70-х гг. – в Японии и с конца 70-х – в Западной Европе.

Современное материальное производство и другие

сферы деятельности все больше нуждаются в информационном обслуживании, переработке огромного количества информации. Универсальным техническим средством обработки любой информации является компьютер, который играет роль усилителя интеллектуальных возможностей человека и общества в целом, а коммуникационные средства, использующие компьютеры, служат для связи и передачи информации. Появление и развитие компьютеров – это необходимая составляющая процесса информатизации общества.

Информатизация общества является одной из закономерностей современного социального прогресса. Этот термин все настойчивее вытесняет широко используемый до недавнего времени термин "компьютеризация общества". При внешней схожести этих понятий они имеют существенное различие.

При компьютеризации общества основное внимание уделяется развитию и внедрению технической базы компьютеров, обеспечивающих оперативное получение результатов переработки информации и ее накопление.

При информатизации общества основное внимание уделяется комплексу мер, направленных на обеспечение полного использования достоверного, исчерпывающего и своевременного знания во всех видах человеческой деятельности.

Таким образом, "информатизация общества" является более широким понятием, чем "компьютеризация общества", и направлена на скорейшее овладение информацией для удовлетворения своих потребностей. В понятии "информатизация общества" акцент надо делать не столько на технических средствах, сколько на сущности и цели социально-технического прогресса. Компьютеры являются базовой технической составляющей процесса информатизации общества.

Информатизация на базе внедрения компьютерных и телекоммуникационных технологий является реакцией общества на потребность в существенном увеличении производительности труда в информационном секторе общественного производства, где сосредоточено более половины трудоспособного населения. Так, например, в информационной сфере США занято более 60% трудоспособного населения, в СНГ – около 40%.

В настоящее время все страны мира в той или иной степени осуществляют процесс информатизации. Неправильно выбранная стратегия информатизации или ее недостаточные динамизм и мобильность могут привести к существенным, а подчас драматическим изменениям во всех сферах жизни страны. Как известно, первая страна, которая начала информатизацию – это США. Другие промышленно развитые страны мира, поняв перспективность и неизбежность этого направления, достаточно быстро сориентировались и стали наращивать темпы внедрения компьютеров и средств телекоммуникаций. В настоящее время вся деловая и политическая пресса США полна бесконечных дискуссий о потере рынков сбыта этой страной в компьютерной, телекоммуникационной и микроэлектронной областях за счет вытеснения США другими развитыми странами (Японией, Германией и др.).

Этот опыт важно учесть при разработке государственной политики информатизации нашей страны, так как, производя информационные технологии, можно иметь все преимущества и условия для развития других высоких технологий и экономики. В большинстве развитых стран понимают, что без чрезвычайных усилий отставание в области информационных и коммуникационных технологий может стать необратимым для их развития в целом. Руководители некоторых стран "третьего мира" с нарастающей тревогой наблюдают за все большим

отставанием их от промышленно развитых стран, осуществляющих информатизацию. Это может привести к тому, что страна будет восприниматься как сырьевой придаток сообщества информационно и промышленно развитых стран. Это в полной мере относится и к России.

Для сопоставления и оценки возможностей страны интерес могут представлять концепция и основные идеи программы информатизации в Японии, которая в настоящее время занимает лидирующее положение в мире по производству современных информационных продуктов, услуг и технологий.

Основные идеи японского проекта информатизации. В настоящее время Япония находится на второй стадии информатизации.

Цель японского проекта – связать те услуги, которые раньше предлагались отдельно. Для этого все виды информации от телефонных посланий и телепрограмм до собственно компьютерной продукции должны передаваться по одному общему кабелю. В перспективе каждый абонент кабельной сети сможет получить несколько услуг одновременно. Большое внимание в проекте также уделяется созданию терминалов для неопытных пользователей с интеллектуальным интерфейсом, где ввод информации осуществляется голосом.

Предполагается, что полное осуществление проекта информатизации займет 20 лет и потребует около 100 млрд. дол. капиталовложений.

Кроме того, несмотря на неудачу с созданием компьютеров 5-го поколения, принята рассчитанная на 10 лет общей стоимостью 480 млн. дол. программа разработки новых типов компьютеров:

- компьютеров с высокой степенью параллелизма обработки информации, в которых одновременно выполняют сложные операции десятки – сотни процессоров;

- компьютеров с нейронными сетями, работа которых аналогична функционированию мозга;
- компьютеров, в которых передача информации осуществляется светом.

В любой стране независимо от уровня ее развития понимают в той или иной мере неизбежность и необходимость претворения в жизнь идей информатизации общества. Многие страны имеют национальные программы информатизации с учетом местных особенностей и условий. Однако при создании и внедрении таких программ следует опираться на опыт передовых стран, учесть их успехи и неудачи, отразить в них существующие и перспективные тенденции информатизации.

Для успешной реализации программы информатизации желательно следовать общим для всего мирового сообщества принципам:

- отказ от стремления в первую очередь обеспечить экономический рост страны;
- необходимость замены экономической структуры, основанной на тяжелой промышленности, структурой, базирующейся на наукоемких отраслях;
- признание приоритетного характера информационного сектора. Основой успешного экономического развития становится создание новой инфраструктуры и сектора услуг, способных поддержать национальную экономику;
- широкое использование достижений мировой науки и техники;
- вложение значительных финансовых средств в информатизацию, как государственную, так и частную;
- объявление роста благосостояния страны и ее граждан за счет облегчения условий коммуникации и обработки информации главной целью информатизации.

Результатом процесса информатизации является

создание информационного общества, где манипулируют не материальными объектами, а символами, идеями, образами, интеллектом, знаниями. Если рассмотреть человечество в целом, то оно в настоящее время переходит от индустриального общества к информационному.

Для каждой страны ее движение от индустриального этапа развития к информационному определяется степенью информатизации общества.

1.4 Об информационной культуре

В период перехода к информационному обществу кроме решения описанных выше проблем необходимо подготовить человека к быстрому восприятию и обработке больших объемов информации, овладению им современными средствами, методами и технологией работы. Кроме того, новые условия работы порождают зависимость информированности одного человека от информации, приобретенной другими людьми. Поэтому уже недостаточно уметь самостоятельно осваивать и накапливать информацию, а надо научиться такой технологии работы с информацией, когда подготавливаются и принимаются решения на основе коллективного знания. Это говорит о том, что человек должен иметь определенный уровень культуры по обращению с информацией. Для отражения этого факта был введен термин информационная культура.

Информационная культура – умение целенаправленно работать с информацией и использовать для ее получения, обработки и передачи компьютерную информационную технологию, современные технические средства и методы.

Приведем определение информационной культуры: "Информационная культура в узком смысле – это уровень достигнутого в развитии информационного общения людей, а также характеристика информационной сферы

жизнедеятельности людей, в которой мы можем отметить степень достигнутого, количество и качество созданного, тенденции развития, степень прогнозирования будущего".

Для свободной ориентации в информационном потоке человек должен обладать информационной культурой как одной из составляющих общей культуры. Информационная культура связана с социальной природой человека. Она является продуктом разнообразных творческих способностей человека и проявляется в следующих аспектах:

- в конкретных навыках по использованию технических устройств (от телефона до персонального компьютера и компьютерных сетей);

- в способности использовать в своей деятельности компьютерную информационную технологию, базовой составляющей которой являются многочисленные программные продукты;

- в умении извлекать информацию из различных источников: как из периодической печати, так и из электронных коммуникаций, представлять ее в понятном виде и уметь ее эффективно использовать;

- во владении основами аналитической переработки информации;

- в умении работать с различной информацией;

- в знании особенностей информационных потоков в своей области деятельности.

Информационная культура вбирает в себя знания из тех наук, которые способствуют ее развитию и приспособлению к конкретному виду деятельности (кибернетика, информатика, теория информации, математика, теория проектирования баз данных и ряд других дисциплин). Неотъемлемой частью информационной культуры являются знание новой информационной технологии и умение ее применять как для автоматизации рутинных операций, так и в неординарных ситуациях, требующих нетрадиционного

творческого подхода.

В информационном обществе необходимо начать овладевать информационной культурой с детства, сначала с помощью электронных игрушек, а затем привлекая персональный компьютер. Для высших учебных заведений социальным заказом информационного общества следует считать обеспечение уровня информационной культуры студента, необходимой для работы в конкретной сфере деятельности. В процессе привития информационной культуры студенту в вузе наряду с изучением теоретических дисциплин информационного направления много времени необходимо уделить компьютерным информационным технологиям, являющимся базовыми составляющими будущей сферы деятельности. Причем качество обучения должно определяться степенью закрепленных устойчивых навыков работы в среде базовых информационных технологий при решении типовых задач сферы деятельности.

В информационном обществе центр тяжести приходится на общественное производство, где существенно повышаются требования к уровню подготовки всех его участников. Поэтому в программе информатизации следует особое внимание уделить информатизации образования как направления, связанного с приобретением и развитием информационной культуры человека. Это, в свою очередь, ставит образование в положение "объекта" информации, где требуется так изменить содержание подготовки, чтобы обеспечить будущему специалисту не только общеобразовательные и профессиональные знания в области информатики, но и необходимый уровень информационной культуры. Повсеместное внедрение персонального компьютера во все сферы народного хозяйства, новые его возможности по организации "дружественной" программной среды, ориентированной на пользователя, использование телекоммуникационной связи, обеспечивающей новые

условия для совместной работы специалистов, применение информационных технологий для самой разнообразной деятельности, постоянно растущая потребность в специалистах, способных ее осуществлять, ставят перед государством проблему по пересмотру всей системы подготовки на современных технологических принципах. В нашей стране решение этой проблемы находится на начальной стадии, поэтому целесообразно учесть опыт наиболее развитых стран, к числу которых относятся США, Япония, Англия, Германия, Франция, где этот процесс уже получил значительное развитие.

Таким образом, информационное общество – общество, в котором большинство работающих заняты производством, хранением, переработкой и реализацией информации, особенно высшей её формы – знаний.

Существует немало препятствий в России для перехода к информационному обществу:

- Слабое развитие качественных систем связи
- Слабое развитие технической базы, недостаточное количество компьютеров
- Низкий общий уровень компьютерной грамотности среди широких слоев населения
- Недостаточное развитие информационных ресурсов, низкая информационная наполняемость сети. /2/

2. КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В НАУКЕ

2.1 Автоматизированные системы научных исследований

Повышение эффективности фундаментальных и прикладных научных исследований становится важным фактором ускорения научно-технического прогресса.

Особое значение для повышения эффективности науки

приобретает автоматизация научных исследований, позволяющая получать более точные и полные модели исследуемых объектов и явлений, ускорять ход научных исследований и снижать их трудоемкость, изучать сложные объекты и процессы, исследование которых традиционными методами затруднительно или невозможно. Применение автоматизированных систем научных исследований (АСНИ) и комплексных испытаний образцов новой техники наиболее эффективно в тех современных областях науки и техники, которые имеют дело с использованием больших объемов информации. К ним, прежде всего, относятся:

- ядерная физика (сбор и обработка экспериментальных данных, получаемых на реакторах, ускорителях и установках термоядерного синтеза);

- физика плазмы и твердого тела;

- радиофизика и электроника;

- астрономия и радиоастрономия;

- космические исследования (обработка информации, получаемой с искусственных спутников для нужд народного хозяйства);

- геология и геофизика (разведка полезных ископаемых);

- исследования Мирового океана, экологические исследования, прогнозирование погоды и стихийных бедствий;

- биология и медицина (исследования в области молекулярной биологии, микробиологического синтеза, диагностики заболеваний);

- химическая технология (моделирование технологических процессов, получение материалов с заданными свойствами);

- исследования сложных технологических процессов в промышленности;

- исследования и разработки в области энергетики

(электростанции, сети электропередачи, энергетические системы);

- исследования и разработки в области транспортных коммуникаций, сетей связи и сетей вычислительных машин;
- натурные и стендовые испытания сложных технических объектов (летательных аппаратов, транспортных устройств, машин, сооружений);
- экономика, социальные исследования, право и языкознание.

Автоматизированные системы научных исследований и комплексных испытаний образцов новой техники обеспечивают получение значительного народнохозяйственного эффекта. Этот эффект образуется от повышения производительности труда в исследовательских и испытательных подразделениях, улучшения технико-экономических характеристик разрабатываемых объектов на основе получения и использования более точных моделей этих объектов, сокращения дорогостоящих натуральных испытаний, исключения некоторых стадий опытно-конструкторских работ, что в конечном счете приводит к снижению затрат на разработку объектов новой техники.

АСНИ отличаются от других типов автоматизированных систем (АСУ, АСУТП, САПР и т.д.) характером информации, получаемой на выходе системы. Прежде всего - это обработанные или обобщенные экспериментальные данные, но главное - полученные на основе этих данных математические модели исследуемых объектов, явлений или процессов. Адекватность и точность таких моделей обеспечивается всем комплексом методических, программных и других средств системы. В АСНИ могут использоваться также и готовые математические модели для изучения поведения тех или иных объектов и процессов, а также для уточнения самих этих моделей. АСНИ поэтому являются системами для

получения, корректировки или исследования моделей, используемых затем в других типах автоматизированных систем для управления, прогнозирования или проектирования.

Как правило, все типы АСНИ должны создаваться на базе серийных средств вычислительной техники широкого применения (процессоров, устройств памяти на магнитных лентах и дисках, печатающих устройств, дисплеев и т.п.). Однако в АСНИ может применяться и специальная аппаратура для сопряжения ЭВМ с исследуемыми объектами. Эта аппаратура должна обеспечивать разнообразные функции предварительной обработки информации, иметь гибкую структуру и максимальную взаимозаменяемость модулей и блоков.

Поэтому создание аппаратуры сопряжения ЭВМ с объектами является одним из важнейших направлений работ, обеспечивающих эффективную разработку и развитие различных типов АСНИ. Блоки и модули аппаратуры сопряжения должны выпускаться серийно в соответствии с международными стандартами.

При разработке общепромышленных руководящих методических материалов по созданию автоматизированных систем научных исследований и комплексных испытаний образцов новой техники были учтены Государственные стандарты и общепромышленные руководящие методические материалы по созданию автоматизированных систем различного назначения.

2.2 Цели создания АСНИ

АСНИ создаются в организациях и на предприятиях в целях:

- обеспечения высоких темпов научно-технического прогресса;

- повышения эффективности и качества научных исследований на основе получения или уточнения с помощью АСНИ математических моделей исследуемых объектов, явлений или процессов, а также применения этих моделей для проектирования, прогнозирования и управления;

- повышения эффективности разрабатываемых с помощью АСНИ объектов, уменьшения затрат на их создание;

- получения качественно новых научных результатов, достижение которых принципиально невозможно без применения АСНИ;

- сокращения сроков, уменьшения трудоемкости научных исследований и комплексных испытаний образцов новой техники.

Достижение целей создания АСНИ обеспечивается путем:

- систематизации и совершенствования процессов научных исследований и испытаний на основе применения математических методов и средств вычислительной техники;

- комплексной автоматизации исследовательских работ в научно-исследовательской организации с необходимой перестройкой ее структуры и кадрового состава;

- повышения качества управления научными исследованиями;

- применения эффективных математических методов организации и планирования экспериментов;

- использования методов обработки и представления результатов научных исследований и испытаний в виде математических моделей, имеющих заданную форму;

- автоматизации трудоемких работ;

- замены натуральных испытаний и макетирования математическим моделированием.

2.3 Определение АСНИ

Автоматизированная система научных исследований и комплексных испытаний образцов новой техники (АСНИ) - это программно-аппаратный комплекс на базе средств вычислительной техники, предназначенный для проведения научных исследований или комплексных испытаний образцов новой техники на основе получения и использования моделей исследуемых объектов, явлений и процессов.

Программно-аппаратный комплекс АСНИ состоит из средств методического, программного, технического, информационного и организационно-правового обеспечения.

Взаимодействие исследуемого объекта, явления или процесса с АСНИ осуществляется через аппаратуру сопряжения, входящую в состав программно-аппаратного комплекса.

Взаимодействие подразделений научно-исследовательской организации или предприятия с АСНИ регламентируется средствами организационно-правового обеспечения системы.

2.4 Функции АСНИ

Основная функция АСНИ состоит в получении результатов научных исследований (комплексных испытаний) путем автоматизированной обработки экспериментальных данных и другой информации, получения и исследования моделей объектов, явлений и процессов на основе применения математических методов, автоматизированных процедур, планирования и управления экспериментом.

Автоматизированные процедуры в АСНИ состоят в том, что исследования (испытания) объектов, явлений и процессов, получение и исследование математических

моделей осуществляется путем взаимодействия пользователя с АСНИ в режиме диалога.

В АСНИ могут осуществляться автоматические процедуры, при которых обработка данных, идентификация или построение математических моделей производятся без участия человека.

В АСНИ могут применяться также процедуры планирования и управления экспериментом, при которых использование моделирования корректирует условия эксперимента, а экспериментальная информация используется для выбора математической модели из некоторого заданного множества таких моделей.

Результатом функционирования АСНИ является подтверждение (отклонение) гипотез или совокупность законченных математических моделей, удовлетворяющая заданным требованиям, а также обработанные результаты исследований, наблюдений и измерений.

Функционирование АСНИ должно обеспечивать получение выходных документов, выполненных в заданной форме и содержащих результаты научных исследований или испытаний, а также рекомендации по использованию этих результатов для прогнозирования, управления или проектирования.

2.5 Структура АСНИ

Основными структурными звеньями АСНИ являются подсистемы.

Подсистемой АСНИ называется выделенная по некоторым признакам часть АСНИ, обеспечивающая выполнение определенных автоматизированных процедур исследований (испытаний) и получение соответствующих выходных документов. Структура АСНИ показана на рис. 1.

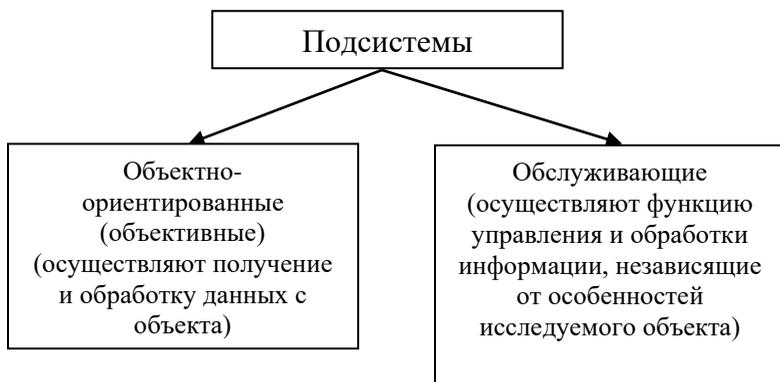


Рис. 1. Структура АСНИ.

Различаются объектно-ориентированные (объектные) и обслуживающие подсистемы АСНИ.

Объектная подсистема осуществляет получение и обработку экспериментальных данных с некоторого объекта.

Объектными могут быть, например, подсистемы:

- обработки экспериментальных данных, получаемых со специализированных установок (ускорителей, спектрометров, испытательных стендов);
- обработки данных на морских судах, системы для сейсморазведки и т.п.;
- коллективного пользования для куста однородных экспериментальных установок или стендов.

Обслуживающая подсистема осуществляет функции управления и обработки информации, не зависящие от особенностей исследуемого явления, объекта или процесса.

Обслуживающими могут быть, например, подсистемы:

- управления АСНИ;
- диалоговых процедур;
- численного анализа;
- планирования и оптимизации эксперимента;
- ввода, обработки и вывода графической информации;

- информационно-поисковых процедур.

Подсистема АСНИ состоит из компонентов, объединенных общей для данной подсистемы процедурой.

Компонентом называется элемент средств обеспечения, выполняющий определенную функцию в подсистеме АСНИ.

Структурное единство подсистемы АСНИ обеспечивается связями между компонентами различных средств обеспечения, образующими подсистему.

Структурное объединение подсистем АСНИ в систему обеспечивается связями между компонентами, входящими в подсистемы.

Средства обеспечения АСНИ состоят из компонентов:

- методического обеспечения;
- программного обеспечения;
- технического обеспечения;
- информационного обеспечения;
- организационно-правового обеспечения.

Компонентами методического обеспечения являются документы, в которых изложены полностью или со ссылкой на первоисточники: теория, методы, способы, математические модели, алгоритмы, алгоритмические специальные языки для описания объектов, терминология, нормативы, стандарты и другие данные, обеспечивающие методологию научных исследований или испытаний в подсистемах АСНИ.

Из состава методического обеспечения могут выделяться компоненты математического и лингвистического обеспечения.

Компонентами программного обеспечения являются документы с текстами программ, программы на машинных носителях и эксплуатационные документы, обеспечивающие функционирование соответствующих подсистем АСНИ.

Программное обеспечение подразделяется на общесистемное и прикладное. Компонентами

общесистемного программного обеспечения являются, например, операционные системы, стандартные управляющие программы на базе операционных систем, трансляторы с алгоритмических языков и языков управления, эмуляторы.

Компонентами прикладного программного обеспечения являются программы и пакеты прикладных программ, предназначенные для осуществления процедур исследований или испытаний.

Компонентами технического обеспечения являются устройства вычислительной и организационной техники, средства и устройства связи с объектом, измерительные и другие устройства или их сочетания, обеспечивающие функционирование соответствующих подсистем АСНИ.

Совокупность компонентов технического обеспечения образует комплекс технических средств АСНИ.

Компонентами информационного обеспечения являются документы, содержащие описания стандартных процедур, типовые математические модели, основные законы, формулы, константы и другие данные, а также файлы и блоки данных на машинных носителях с записью указанных документов, обеспечивающие функционирование соответствующих подсистем АСНИ.

Совокупность компонентов информационного обеспечения образует информационную базу (базу данных) АСНИ.

Компонентами организационно-правового обеспечения АСНИ являются методические и руководящие материалы, положения, инструкции, приказы, штатные расписания, квалификационные требования, инструкции для пользователей и другие документы, обеспечивающие взаимодействие подразделений организации или предприятия при создании, эксплуатации и развитии АСНИ.

2.6 Основные принципы создания АСНИ

При создании и развитии АСНИ рекомендуется применять следующие принципы:

- последовательное расширение сферы автоматизации научных исследований;
- интеграция АСНИ;
- типизация, унификация и стандартизация компонентов АСНИ;
- тиражирование типовых подсистем и компонентов АСНИ;
- применение единой методологии создания и развития АСНИ;
- системный подход к проектированию;
- адаптивность;
- разработка критериев эффективности АСНИ;
- ориентация на методики ведущих в тематике коллективов;
- опережающее развитие базовых решений в головных организациях.

Последовательное расширение сферы автоматизации научных исследований предполагает:

- внедрение средств автоматизации в новые области научных исследований, в первую очередь в те области, где получение новых существенных результатов невозможно без использования средств автоматизации;
- расширение контингента пользователей автоматизированных систем научных исследований - от экспериментаторов до руководителей крупных научных программ;
- автоматизация всех этапов научных исследований от планирования и управления экспериментами до анализа и перспективного планирования основных направлений научных исследований.

Тематическая, функциональная и территориальная интеграция АСНИ должна быть направлена в первую очередь на создание систем коллективного пользования:

- для крупных экспериментальных, исследовательских и опытных установок и комплексных производственных испытаний различных технических объектов в исследовательских и проектных организациях, в ВУЗах, на предприятиях, полигонах и т.п.;

- для отдельных крупных научно-исследовательских организаций, проводящих комплексные исследования сложных объектов;

- для взаимосвязанных единой программой работ или родственных по тематике групп исследовательских и проектных организаций;

- для территориально объединенных групп исследовательских и проектных организаций, некоторых республиканских академий наук, академических и ведомственных научных центров.

Интеграция АСНИ включает в себя:

- создание многомашинных иерархических измерительно-вычислительных комплексов коллективного пользования, обслуживающих несколько экспериментов;

- развитие информационной базы (создание централизованных и распределенных банков научных данных, обмен научными данными по каналам связи между АСНИ в согласованных форматах, унификацию структур данных и типизацию систем управления базами данных);

- развитие общесистемного программного обеспечения (унификацию операционной среды, использование стандартных и создание специализированных телекоммуникационных методов доступа, создание многоабонентских систем реального времени, работающих в режимах мультидоступа).

В качестве основы для создания АСНИ должны

использоваться типовые, проблемно-ориентированные или специализированные измерительно-вычислительные комплексы (ИВК), включающие в себя серийные средства измерительной техники, а также типовое программное обеспечение.

Особое внимание должно быть уделено типовой аппаратуре сопряжения ЭВМ с объектом исследования, созданию типовых программно-управляемых модульных систем для сбора информации и управления сложными объектами. Требования к этой аппаратуре формируются на основе соответствующих государственных и международных стандартов с тем, чтобы обеспечить максимальную совместимость технических и программных средств АСНИ, производимых различными организациями и в различных странах. Необходимо использовать стандарты КАМАК, обеспечивающие аппаратурную и программную совместимость подсистем и компонентов АСНИ.

Важнейшим условием унификации и типизации компонентов и подсистем АСНИ является широкое использование в них агрегатных средств измерительной и вычислительной техники, удовлетворяющих требованиям конструктивной, информационной, эксплуатационной, энергетической и других видов совместимости.

В разработке новых компонентов АСНИ необходимо широко применять аппаратную реализацию наиболее типовых функций обработки данных, операционных систем и других функций управления операционной средой.

Системный подход в проектировании предполагает проведение проектирования на основе системного анализа, включающего решение комплекса технических, экономических, организационных вопросов, решение которых в совокупности обеспечит создание АСНИ оптимальным способом.

Адаптивность предполагает легкую приспособляемость

АСНИ к изменению решаемых с ее помощью задач - scalability.

Разработка критериев эффективности АСНИ должна позволить дать объективную оценку экономического или иного эффекта, получаемого от внедрения АСНИ.

При создании или заимствовании компонентов АСНИ должны обеспечиваться требования к этим компонентам, вытекающие из общесистемных принципов, изложенных выше.

Компоненты методического обеспечения рекомендуется создавать на основе:

- перспективных методов автоматизации научных исследований, поиска новых принципов действия и технических решений;
- эффективных методов математического моделирования исследуемого объекта и его элементов;
- использование методов формализованного описания и имитационного моделирования;
- применения методов планирования и оптимизации эксперимента;
- использования типовых и стандартных процедур обработки данных;
- стандартных вычислительных и расчетных методов.

Компоненты программного обеспечения рекомендуется создавать с использованием следующих требований:

- максимального применения стандартного и серийного программного обеспечения технологии открытых систем;
- адаптируемости к различным конфигурациям ЭВМ и их операционным системам - открытости, переносимости, взаимодействия;
- обеспечения мультипрограммной работы, режима деления времени, работы в режиме диалога;
- модульного построения, расширения и обновления;
- обеспечения контроля и диагностирования;

- применения языков и систем программирования, рекомендованных ГКНТ;
- автоматизации оборота документов;
- в технических заданиях на разработку компонентов программного обеспечения предусматривать требования, обязывающие разработчиков использовать рекомендованные ГКНТ технологии программирования, повышающие производительность труда программистов.

Компоненты технического обеспечения должны создаваться на базе:

- серийных средств вычислительной техники общего назначения;
- серийных агрегатных средств измерительной техники общего назначения;
- специализированных технических средств, если их применение в АСНИ технически и экономически оправдано;
- современных технических средств общего назначения для сопряжения ЭВМ с объектами исследования.

Компоненты информационного обеспечения должны создаваться на основе:

- максимального использования серийных технических и программных средств;
- гибкой организации и открытой структуры, приспособленной к пополнению и объединению открытых систем;
- возможности логической структуризации данных по формальным признакам;
- возможности одновременного использования данных несколькими подсистемами АСНИ;
- обеспечения точности стандартных и нормативных данных;
- разграничения доступа и защиты файлов и блоков данных;
- соответствия международным стандартам открытых

систем.

Компоненты организационно-правового обеспечения АСНИ должны создаваться на основе:

- прогрессивных методов научных исследований и испытаний;
- стандартов и нормативных документов, регламентирующих научные исследования в отрасли;
- современных методов планирования и управления;
- анализа экономической эффективности и применения мер материального стимулирования.

Развитие (совершенствование) компонентов АСНИ осуществляется путем создания новых модификаций (в том числе новых редакций, версий, типов) этих компонентов. /3/

2.7 Модель научных исследований

С учетом вышеизложенного, можно представить следующую модель проведения научных исследований:



Рис. 2. Модель научных исследований.

2.8 Научные ресурсы Интернет

Процессы информатизации, идущие в современном обществе, выдвигают информацию, информационные технологии в качестве основного ресурса развития как отдельных отраслей, так и общества в целом. В широком смысле информационные ресурсы можно определить как всю накопленную человечеством информацию об окружающей действительности, зафиксированную на материальных носителях в любой форме, обеспечивающей ее передачу во времени и пространстве между различными потребителями для решения научных, производственных, управленческих и других задач. /4/

В зависимости от тех или иных критериев вся совокупность информационных ресурсов может быть поделена на отдельные классы или группы. Так, информационные ресурсы, содержащие информацию о прошлом, можно определить как исторические информационные ресурсы. К ним относятся, например, исторические источники, научные труды историков, учебники по истории и т.д. Основная масса подобных ресурсов сосредоточена в библиотеках, архивах, музеях, научно-образовательных учреждениях. Развитие Интернета привело к появлению исторических ресурсов и в глобальной сети, ставшей новым каналом получения и распространения исторической информации, который принципиально отличается от традиционных.

При использовании Интернета в качестве источника информации необходимо иметь общее представление об организационных и технологических особенностях глобальной сети, которые, безусловно, отражаются на размещаемых в ней ресурсах. Можно выделить следующие характерные черты Интернета:

- общедоступность информации, что подразумевает,

естественно при наличии самого выхода в сеть, не только быстрое и, как правило, без особых ограничений получение той или иной информации, но и такое же ее размещение;

- бесконтрольность и отсутствие какой-либо цензуры, что связано с самой "природой" Интернета, который представляет собой саморазвивающуюся систему, совокупность множества независимых сетей;

- сверхоперативная коммуникативность, которая выражается в небывалой скорости размещения актуальной информации в сети;

- динамизм представления информации, что подразумевает с одной стороны постоянное обновление Интернет-ресурсов, а с другой - возможную их миграцию в пределах сети или исчезновение;

- универсальность и сравнительная простота Интернет-технологий, что позволяет создавать и размещать в сети различные по уровню сложности ресурсы, содержащие текстовую, изобразительную, звуковую, видео информацию;

- относительно небольшая в настоящий момент "глубина памяти", то есть максимальный срок существования ресурсов, что определяется "молодостью" самой глобальной сети. /5/

По способу фиксирования информации ресурсы Интернета являются электронными документами. Существуют разные определения этого понятия. В общем виде электронный (или электронно-цифровой) документ - это документ, созданный человеком средствами электронно-вычислительной техники. /6/ При создании электронных документов, которые называют еще машиночитаемыми, информация фиксируется на материальном носителе в виде "машинного кода", что делает ее недоступной для непосредственного восприятия человеком. Представить электронный документ в понятной человеку форме возможно только с помощью специальных аппаратных и программных

средств.

В широком смысле, являясь электронным документом, Интернет-документ имеет и свои отличительные черты, среди которых в качестве основной следует выделить его сетевой характер. В связи с этим Интернет-документ можно определить как документ, созданный человеком (прямо или опосредованно) с помощью средств электронно-вычислительной техники для размещения и функционирования в международной глобальной сети Интернет.

Все Интернет-документы можно разделить на две группы:

1. Документы, созданные самим человеком с помощью какого-то программного средства. К этому типу, например, относятся Web-страницы и состоящие из них Web-сервера, а также сообщения электронной почты.

2. Документы, автоматически создающиеся в процессе функционирования специальных программ, написанных человеком. К этому виду можно отнести лог-файлы (файлы регистрации) Интернет-серверов, лог-файлы поисковых машин, в которых содержатся сведения о работе сети и отдельных компьютеров. /7/

В качестве полноценных информационных ресурсов сейчас используются прежде всего Web-страницы (или сайты), поскольку они создаются целенаправленно для широкого распространения различной информации. Необходимо отметить, что появление WWW-технологии (World Wide Web - Всемирная Паутина) значительно изменило Интернет, превратив перемещение по сети в увлекательное путешествие. Это связано с отличительными свойствами Web, которые заключаются в наличие интерактивных средств гипермедиа и гиперссылок.

Исходя из возможностей использования (или по потенциальному назначению) содержащейся в них

информации научно-образовательные ресурсы можно условно разделить на представительские, исследовательские и образовательные. Предлагаемое разделение не подразумевает автономное существование данных групп ресурсов. Будучи тесно взаимосвязанными, они могут присутствовать в разных комбинациях в рамках одной Web-страницы.

Представительские ресурсы содержат общие сведения о различных организациях исторического профиля: образовательных учреждениях, научно-исследовательских институтах, архивах, музеях, библиотеках и т.д. Кроме того, к представительским ресурсам можно отнести информацию о проводимых конференциях, новых публикациях, различных научно-исследовательских проектах и грантах.

Исследовательские ресурсы подразумевают под собой такие ресурсы, которые можно использовать для научного исследования. К ним относятся, прежде всего, научные источники разных видов, базы данных, а также научные работы ученых. В качестве исследовательских ресурсов можно рассматривать и каталоги библиотек, и путеводители по архивным фондам, которые позволяют находить нужную литературу и предварительно определять границы поиска архивных документов. К исследовательским ресурсам по новейшей истории следует, по всей видимости, отнести материалы современных электронных СМИ, государственных и негосударственных учреждений, общественных движений и партий и т.д.

Образовательные ресурсы включают в себя, прежде всего, электронные версии различных учебных и методических материалов. Это могут быть учебники и учебные пособия, учебные программы различных курсов, тексты источников, карты, энциклопедии, словари и т.д. При этом в качестве образовательных могут использоваться и исследовательские ресурсы, поскольку они также несут

образовательную нагрузку, что делает введенное разграничение в значительной степени условным.

Классификацию научных сетевых ресурсов можно провести и по другим основаниям. Например, исходя из цели их создания, можно выделить специализированные Web-сайты, которые целенаправленно создавались для размещения научных материалов и неспециализированные, для которых научная информация является сопутствующей.

В межгосударственном стандарте на электронные издания, принятом в июле 2002 г., имеется своя система классификации. В соответствии с ней электронные издания по своему целевому назначению делятся на научные, научно-популярные, производственно-практические, учебные, массово-политические, справочные, художественные и другие. /8/

В зависимости от статуса их "авторов", исторические Интернет-ресурсы делятся на корпоративные, то есть сайты различных организаций и учреждений, и персональные. При этом те и другие могут создаваться и поддерживаться одним человеком или целым коллективом. Главное их отличие не в количественном составе создателей, а в том, кого они представляют. В этом смысле первые можно рассматривать как официальные сайты, вторые - как неофициальные. Среди создателей персональных сайтов чаще встречаются не профессиональные ученые, а просто любители, что зачастую сказывается на подборе научного материала и правильном его оформлении.

По подбору и способу организации материалов среди Web-сайтов научно-образовательной направленности можно выделить следующие группы:

- Обычные Web-страницы, представляющие собой целостный комплекс различных электронных документов, объединенных определенной тематикой.

- Коллекции электронных текстов и изображений

(виртуальные библиотеки), в которых собраны преимущественно электронные аналоги письменных произведений, ранее опубликованных в традиционной бумажной форме, или фотографии, плакаты и другие изобразительные материалы.

- Электронные библиотеки, под которыми понимаются сайты традиционных библиотек, предоставляющие доступ к электронным каталогам библиотеки и копиям произведений, составляющих ее книжно-журнальное собрание.

- Электронные архивы, то есть сайты традиционных архивов, которые предоставляют пользователю возможность работать с путеводителями по архивным фондам, с оцифрованными коллекциями архивных документов. /9/

- Словари и энциклопедии - сайты с разветвленной системой поиска, предоставляющие справочную информацию из электронных версий, опубликованных ранее или из оригинальных электронных справочных и энциклопедических изданий.

- Электронные журналы/газеты - это сайты редактируемых периодических изданий, которые могут иметь или не иметь традиционные печатные варианты.

- Каталоги ресурсов, которые содержат только ссылки на различные сетевые ресурсы, сгруппированные в тематические разделы. Подобные каталоги, часто снабжаются системой поиска нужных ссылок по ключевым словам.

Таким образом, существует множество вариантов классификации научных сетевых ресурсов, и эта проблема требует своего дальнейшего изучения.

Использование Интернет в исследовательской и образовательной практике ставит вопрос и об адекватном описании научных сетевых ресурсов.

На данный момент наиболее распространенными типами публикаций источников в Рунете являются научно-популярный и учебный. Для них характерно либо полное

отсутствие научно-справочного аппарата, либо воспроизведение его отдельных элементов. Собственно научных публикаций, которые бы отвечали существующим требованиям, предъявляемым к изданию научных источников, немного. Российские научно-исследовательские центры, являющиеся главными хранителями научной информации, пока не готовы к широкому представлению своих данных в Интернет. Это значительно сужает электронную источниковую базу, которая включает в себя в основном уже публиковавшиеся источники и современные материалы.

Вообще проблема авторства электронных ресурсов и соблюдения авторских прав является одной из самых сложных в сетевом пространстве. С одной стороны Интернет является свободной информационной средой и никем целенаправленно не контролируется, с другой - на размещаемые в сети материалы распространяется закон об авторских и других смежных правах, который ограничивает их использование без разрешения автора или собственника. Легитимная публикация научных источников и других научных материалов требует определенной юридической подготовки. В результате, если строго следовать законодательным нормам, следует признать, что большинство научных ресурсов в сети выставлены там незаконно. В то же время современное законодательство достаточно противоречиво. Например, перевод в электронную форму какого-либо текста на основе дореволюционного издания не преследуется, но сканирование того же самого текста с книги-переиздания 1980-х годов запрещено. Законодательство также не учитывает некоммерческий характер большинства научных сетевых проектов. Можно сказать, что на данный момент нормы закона и складывающаяся практика, к сожалению, не приведены в соответствие друг с другом.

Представленные схемы описания ресурсов Интернет как научных источников требуют доработки и уточнения. В частности, остается нерешенным ряд вопросов, связанных с определением авторства и датировки ресурсов Интернет, например, кого считать автором сайта: разработчика, дизайнера, автора размещенных материалов?

При описании ресурсов Интернет как научных источников решаются, прежде всего, прикладные задачи, облегчающие возможности и формы использования этих ресурсов. Вместе с тем, создание системы описания ресурсов Интернет создает условия для решения задачи их анализа, для вовлечения этих ресурсов в научный оборот и реального превращения сети Интернет в новую информационную среду науки.

3. СОВРЕМЕННЫЕ КОМПЬЮТЕРНЫЕ СИСТЕМЫ

3.1 Архитектура современного персонального компьютера

Всякая новая техника создается, когда возникает большая общественная потребность в ней. К середине нашего века научно-технический прогресс привел к необходимости облегчить труд ученых и инженеров, ускорить выполнение громоздких расчетов и повысить их точность. Автоматизация вычислений, создание мощного, быстродействующего и точного электронного арифмометра - вот о чем думали специалисты - творцы первых ЭВМ. Никто в те далекие теперь годы и не помышлял о будущем информационном перевороте; решалась скромная, на первый взгляд чисто техническая задача. Да и возможности первых образцов вычислительных машин оказались небольшими.

Еще в годы второй мировой войны в США была

собрана из обычных электромагнитных реле, которых тогда были тысячи на любой телефонной станции, вычислительная машина "Марк-1". Считала она в привычной десятичной системе счисления - телефонные реле имели как раз десять позиций своих контактов. Работала машина неторопливо - пока реле отщелкают свое, можно было вручную все посчитать. Поэтому вскоре после войны появились машины на электронных лампах: в США - ENIAC, а в СССР - МЭСМ (малая электронно-счетная машина), которую создал коллектив под руководством С.А. Лебедева. Ламповые ЭВМ уже оказались в состоянии выполнять сотни и тысячи арифметических или логических операций за одну секунду. Они могли обеспечить высокую точность вычислений. Человек уже не мог соперничать с такой машиной.

Жизнь первого поколения ЭВМ оказалась короткой - около десяти лет. Электронные лампы грелись, потребляли много электроэнергии, были громоздки (так, например, машина ENIAC весила 30 тонн, имела 18 тысяч электронных ламп и потребляла 150 киловатт). И что еще хуже - часто выходили из строя. Над компьютерщиками продолжали посмеиваться: чтобы сложить, скажем, два числа, требовалось написать программу из многих машинных команд. Например, такую: "Ввести в машину первое слагаемое; ввести второе; переслать из памяти первое слагаемое в арифметическое устройство (arithmetic unit); переслать в арифметическое устройство второе слагаемое и вычислить сумму; переслать результат в память; вывести это число из памяти машины и напечатать его". Каждую команду и оба слагаемых "набивали" на перфокарты и только потом вводили колоду перфокарт в компьютер и ждали, когда протарахтит печатающее устройство (printing device) и на выползающей из него бумажной ленте будет виден результат - искомая сумма.

В конце пятидесятых - начале шестидесятых годов на

смену электронной лампе пришел компактный и экономичный прибор - транзистор (transistor). Компьютеры сразу стали производительнее и компактнее, уменьшилось потребление электроэнергии. ЭВМ второго поколения "научились" программировать сами для себя. Появились системы автоматизации программирования, состоящие из алгоритмических языков (algorithmic language) и трансляторов (translator) для них. Теперь пользователь изучал язык ЭВМ, приближенный к языку научных, инженерных или экономических расчетов. Например, *Фортран* или *Алгол-60*.

Программа, написанная на известном машине языке, переводилась на язык команд автоматически, с помощью программы-переводчика. Такие программы называли трансляторами, а процесс перевода - трансляцией. Транслятор не только переводил программу с алгоритмического языка на язык команд, но и проверял грамотность составленной пользователем программы, выявлял и классифицировал ошибки, давал советы по их устранению.

Прошло всего 7-8 лет, и это поколение буквально вытолкнули машины следующего, третьего поколения. Перевод вычислительной техники на интегральные микросхемы серьезно удешевил ее, поднял возможности и позволил начать новый этап ее практического применения. Компьютеры вторглись - уже не штучно, а в массовом порядке - практически во все сферы науки, экономики, управления.

Развитие микроэлектроники позволило создать и освоить технологию интегральных схем с особо большой плотностью компоновки. На одном кристалле размером меньше ногтя стали размещать не десятки и сотни, а десятки тысяч транзисторов и других элементов. Большие интегральные схемы (very large scale integration) составили техническую основу, элементную базу ЭВМ четвертого поколения. Их производительность возросла фантастически -

до сотен миллионов операций в секунду.

Подлинный переворот в автоматике и управлении произвели появившиеся в семидесятые годы микропроцессоры и микро-ЭВМ - сверхминиатюрные изделия вычислительной техники. Малый вес и габариты, ничтожное электропотребление - все это позволило встраивать "монолитные" микро-ЭВМ и микропроцессорные наборы непосредственно в средства связи, машины, механизмы, приборы и другие технические устройства, чтобы наилучшим образом управлять их работой и контролировать ее.

ЭВМ третьего-четвертого поколения стали многоязычными и многопрограммными: они получили возможность вести диалог со многими пользователями одновременно и решать задачи, запрограммированные на разных языках.

Основное направление в развитии современных компьютеров (пятого и шестого поколения) - разработка машины, более похожей на человека по способам ввода и хранения информации и методам решения задач. Различные области информатики занимаются изучением этих проблем - задач искусственного интеллекта (artificial intelligence), экспертных систем (expert systems) и представления информации (information presentation).

3.2 Магистрально-модульный принцип построения современного компьютера

Под архитектурой компьютера понимается его логическая организация, структура, ресурсы, т. е. средства вычислительной системы, которые могут быть выделены процессу обработки данных на определенный интервал времени. Архитектура современных ПК основана на магистрально-модульном принципе.

Модульный принцип позволяет потребителю самому подобрать нужную ему конфигурацию компьютера и производить при необходимости его модернизацию. Модульная организация системы опирается на магистральный (шинный) принцип обмена информации. Магистраль или системная шина - это набор электронных линий, связывающих воедино по адресации памяти, передачи данных и служебных сигналов процессор, память и периферийные устройства.

Обмен информацией между отдельными устройствами ЭВМ производится по трем многоуровневым шинам, соединяющим все модули, - шине данных, шине адресов и шине управления.

Подключение отдельных модулей компьютера к магистрали на физическом уровне осуществляется с помощью контроллеров, а на программном обеспечивается драйверами. Контроллер принимает сигнал от процессора и дешифрует его, чтобы соответствующее устройство смогло принять этот сигнал и отреагировать на него. За реакцию устройства процессор не отвечает - это функция контроллера. Поэтому внешние устройства ЭВМ заменяемы, и набор таких модулей произволен.

Разрядность шины данных задается разрядностью процессора, т. е. количеством двоичных разрядов, которые процессор обрабатывает за один такт.

Данные по шине данных могут передаваться как от процессора к какому-либо устройству, так и в обратную сторону, т. е. шина данных является двунаправленной. К основным режимам работы процессора с использованием шины передачи данных можно отнести следующие: запись/чтение данных из оперативной памяти и из внешних запоминающих устройств, чтение данных с устройств ввода, пересылка данных на устройства вывода.

Выбор абонента по обмену данными производит

процессор, который формирует код адреса данного устройства, а для ОЗУ - код адреса ячейки памяти. Код адреса передается по адресной шине, причем сигналы передаются в одном направлении, от процессора к устройствам, т. е. эта шина является однонаправленной.

По шине управления передаются сигналы, определяющие характер обмена информацией, и сигналы, синхронизирующие взаимодействие устройств, участвующих в обмене информацией.

Внешние устройства к шинам подключаются посредством интерфейса. Под интерфейсом понимают совокупность различных характеристик какого-либо периферийного устройства ПК, определяющих организацию обмена информацией между ним и центральным процессором. В случае несовместимости интерфейсов (например, интерфейс системной шины и интерфейс винчестера) используют контроллеры.

Чтобы устройства, входящие в состав компьютера, могли взаимодействовать с центральным процессором, в IBM-совместимых компьютерах предусмотрена система прерываний (Interrupts). Система прерываний позволяет компьютеру приостановить текущее действие и переключиться на другие в ответ на поступивший запрос, например, на нажатие клавиши на клавиатуре. Ведь с одной стороны, желательно, чтобы компьютер был занят возложенной на него работой, а с другой - необходима его мгновенная реакция на любой требующий внимания запрос. Прерывания обеспечивают немедленную реакцию системы.

3.3 Периферийные и внутренние устройства

Прогресс компьютерных технологий идет семимильными шагами. Каждый год появляются новые процессоры, платы, накопители и прочие периферийные

устройства. Рост потенциальных возможностей ПК и появление новых более производительных компонентов неизбежно вызывает желание модернизировать свой компьютер. Однако нельзя в полной мере оценить новые достижения компьютерной технологии без сравнения их с существующими стандартами.

Разработка нового в области ПК всегда базируется на старых стандартах и принципах. Поэтому знание их является основополагающим фактором для (или против) выбора новой системы.

В состав ЭВМ входят следующие компоненты:

- центральный процессор (CPU);
- оперативная память (memory);
- устройства хранения информации (storage devices);
- устройства ввода (input devices);
- устройства вывода (output devices);
- устройства связи (communication devices).

Во всех вычислительных машинах до середины 50-х годов устройства обработки и управления представляли собой отдельные блоки, и только с появлением компьютеров, построенных на транзисторах, удалось объединить их в один блок, названный процессором.

Процессор - это мозг ЭВМ. Он контролирует действия всех остальных устройств (devices) компьютера и координирует выполнение программ. Процессор имеет свою внутреннюю память, называемую регистрами, управляющее и арифметико-логическое устройства.

Процесс общения процессора с внешним миром через устройства ввода-вывода по сравнению с информационными процессами внутри него протекает в сотни и тысячи раз медленнее. Это связано с тем, что устройства ввода и вывода информации часто имеют механический принцип действия (принтеры, клавиатура, мышь) и работают медленно. Чтобы освободить процессор от простоя при ожидании окончания

работы таких устройств, в компьютер вставляются специализированные микропроцессоры-контроллеры (от англ. controller - управляющий). Получив от центрального процессора компьютера команду на вывод информации, контроллер самостоятельно управляет работой внешнего устройства. Окончив вывод информации, контроллер сообщает процессору о завершении выполнения команды и готовности к получению следующей.

Число таких контроллеров соответствует числу подключенных к процессору устройств ввода и вывода. Так, для управления работой клавиатуры и мыши используется свой отдельный контроллер. Известно, что даже хорошая машинистка не способна набирать на клавиатуре больше 300 знаков в минуту, или 5 знаков в секунду. Чтобы определить, какая из ста клавиш нажата, процессор, не поддерживаемый контроллером, должен был бы опрашивать клавиши со скоростью 500 раз в секунду. Конечно, по его меркам это не бог весть какая скорость. Но это значит, что часть своего времени процессор будет тратить не на обработку уже имеющейся информации, а на ожидание нажатий клавиш клавиатуры.

Таким образом, использование специальных контроллеров для управления устройствами ввода-вывода, усложняя устройство компьютера, одновременно разгружает его центральный процессор от непроизводительных трат времени и повышает общую производительность компьютера.

Существует два типа оперативной памяти - память с произвольным доступом (RAM или random access memory) и память, доступная только на чтение (ROM или read only memory). Процессор ЭВМ может обмениваться данными с оперативной памятью с очень высокой скоростью, на несколько порядков превышающей скорость доступа к другим носителям информации, например дискам.

Оперативная память с произвольным доступом (RAM) служит для размещения программ, данных и промежуточных результатов вычислений в процессе работы компьютера. Данные могут выбираться из памяти в произвольном порядке, а не строго последовательно, как это имеет место, например, при работе с магнитной лентой. Память, доступная только на чтение (ROM) используется для постоянного размещения определенных программ (например, программы начальной загрузки ЭВМ). В процессе работы компьютера содержимое этой памяти не может быть изменено.

Оперативная память - временная, т. е. данные в ней хранятся только до выключения ПК. Для долговременного хранения информации служат дискеты, винчестеры, компакт-диски и т. п. Конструктивно элементы памяти выполнены в виде модулей, так что при желании можно сравнительно просто заменить их или установить дополнительные и тем самым изменить объем общей оперативной памяти компьютера. Основными характеристиками элементов (микросхем) памяти являются: тип, емкость, разрядность и быстродействие.

В настоящее время отдельные микросхемы памяти не устанавливаются на материнскую плату. Они объединяются в специальных печатных платах, образуя вместе с некоторыми дополнительными элементами модули памяти (SIMM- и DIMM-модули).

Устройства хранения информации используются для хранения информации в электронной форме. Любая информация - будь это текст, звук или графическое изображение, - представляется в виде последовательности нулей и единиц. Ниже перечислены наиболее распространенные устройства хранения информации.

- Винчестеры (hard discs)

Жесткие диски - наиболее быстрые из внешних устройств хранения информации. Кроме того, информация,

хранящаяся на винчестере, может быть считана с него в произвольном порядке (диск - устройство с произвольным доступом).

Емкость диска современного персонального компьютера составляет десятки гигабайт. В одной ЭВМ может быть установлено несколько винчестеров.

- Оптические диски (CD или DVD-Rom)

Лазерные диски, как их еще называют, имеют емкость около 600 – 700 мегабайт (для CD-диска) и 4,5 гигабайт (для DVD-диска) и обеспечивают только считывание записанной на них однажды информации в режиме произвольного доступа. Существуют CD и DVD-диски предназначенные для многократной записи и перезаписи информации на специальных приводах. Скорость считывания информации определяется устройством, в которое вставляется компакт-диск (CD или DVD-Rom drive).

- Магнито-оптические диски

В отличие от оптических дисков магнито-оптические диски позволяют гораздо быстрее, чем CD или DVD-диски читать и записывать информацию.

- Флоппи диски (floppy discs)

В основе этих устройств хранения лежит гибкий магнитный диск, помещенный в твердую оболочку (конверт). Для того чтобы прочесть информацию, хранящуюся на дискете, ее необходимо вставить в дисковод (floppy disc drive) компьютера. Емкость современных дискет всего 1.44 мегабайта. По способу доступа дискета подобна винчестеру. В настоящее время этот тип носителей информации практически исчез и встречается на устаревших компьютерах.

- Zip и Jaz Iomega discs

Это относительно новые носители информации, которые призваны заменить гибкие магнитные диски. Их можно рассматривать, как быстрые и большие по емкости (100 мегабайт - Zip, 1 гигабайт - Jaz) дискеты. Эти носители и

приводы для них весьма дороги и используются для хранения важных данных. Большого распространения они не получили.

- Магнитные ленты (magnetic tapes)

Современные магнитные ленты, хранящие большие объемы информации (до нескольких гигабайт), внешне напоминают обычные магнитофонные кассеты и характеризуются строго последовательным доступом к содержащейся на них информации. Используются в основном для периодического резервного копирования данных с серверов. Эти узкоспециализированные и дорогостоящие носители широкого распространения не получили.

Устройства ввода передают информацию в ЭВМ от различных внешних источников. Информация может быть представлена в весьма различных формах: текст - для клавиатуры (keyboard), звук - для микрофона (microphone), изображение - для сканера (scanner).

Клавиатура - одно из самых распространенных на сегодня устройств ввода информации в компьютер. Она позволяет нажатием клавиш вводить символьную информацию.

Ключевой принцип работы клавиатуры заключается в том, что она воспринимает нажатия клавиш и преобразует их в двоичный код, индивидуальный для каждой клавиши.

Но указывать место на экране монитора, в котором компьютер что-то должен изменить, с помощью клавиатуры неудобно. Для этого существует специальное устройство ввода - мышь.

Принцип ее действия основан на измерении направления и величины поворота шарика, находящегося в нижней части мыши. Когда мы перемещаем мышь по поверхности стола, шарик поворачивается. Специальные датчики измеряют поворот шарика. После преобразования результатов измерения в двоичный код они передаются в компьютер. По ним процессор выводит на экран условное

изображение указателя (обычно в форме стрелки). Существуют разновидности этого устройства - оптические мыши, принцип действия которых основан на отслеживании перемещения луча света.

Мышь не позволяет вводить числовую и буквенную информацию, но удобна для работы с графическими объектами, изображенными на экране.

Сканер - устройство ввода графической информации. Его особенность - способность считывать изображение непосредственно с листа бумаги.

Принцип действия сканера напоминает работу человеческого глаза. Освещенный специальным источником света, находящимся в самом сканере, лист бумаги с текстом или рисунком "осматривается" микроскопическим "электронным глазом". Диаметр участка изображения, воспринимаемого таким "глазом", составляет 1/20 миллиметра и соответствует диаметру человеческого волоса. Яркость считываемой в данный момент точки изображения кодируется двоичным числом и передается в компьютер. Для того чтобы осмотреть стандартный лист бумаги, "электронному глазу" приходится строку за строкой обходить его, передавая закодированную информацию об освещенности каждой точки изображения в компьютер.

Монитор - устройство вывода на экран текстовой и графической информации. Мониторы бывают цветными и монохромными. Они могут работать в одном из двух режимов: текстовом или графическом.

В текстовом режиме экран монитора условно разбивается на отдельные участки - знакоместа, чаще всего на 25 строк по 80 символов (знакомест). В каждое знакоместо может быть выведен один из 256 заранее определенных символов. В число этих символов входят большие и малые латинские буквы, цифры, определенные символы, а также псевдографические символы, используемые для вывода на

экран таблиц и диаграмм, построения рамок вокруг участков экрана и так далее. В число символов, изображаемых на экране в текстовом режиме, могут входить и символы кириллицы.

На цветных мониторах каждому знакоместу может соответствовать свой цвет символа и фона, что позволяет выводить красивые цветные надписи на экран. На монохромных мониторах для выделения отдельных частей текста и участков экрана используется повышенная яркость символов, подчеркивание и инверсное изображение.

Графический режим предназначен для вывода на экран графической информации (рисунки, диаграммы, фотографии и т. п.). Разумеется в этом режиме можно выводить и текстовую информацию в виде различных надписей, причем эти надписи могут иметь произвольный шрифт, размер и др.

В графическом режиме экран состоит из точек, каждая из которых может быть темной или светлой на монохромных мониторах и одного или нескольких цветов - на цветном. Количество точек на экране называется разрешающей способностью монитора в данном режиме. Следует заметить, что разрешающая способность не зависит напрямую от размеров экрана монитора.

Принтер - устройство для вывода результатов работы компьютера на бумагу. Само название произошло от английского слова printer, означающего "печатник" (печатающий).

Первые принтеры создавали изображение из множества точек, получающихся под действием иголок, ударяющих через красящую ленту по бумаге и оставляющих на ней след. Иголки закреплены в печатающей головке и приводятся в движение электромагнитами. Сама же головка движется горизонтально, печатая строку за строкой. Количество иголок составляет 8 или 24 при одной и той же высоте печатающей головки. Во втором случае их делают тоньше, а получаемое

изображение оказывается более "мелкозернистым".

Такой принтер преобразует электрические сигналы, выдаваемый компьютером, в движение иглонок. Принтеры, использующие для получения изображения механический (ударный) принцип, называют матричными.

Матричные принтеры создают сильный шум и требуют частой замены красящей ленты, поэтому в 80-х годах был предложен другой способ печати на бумаге - струйный.

Принцип, лежащий в основе струйной печати с использованием жидких чернил, состоит в нанесении капелек чернил непосредственно на поверхность бумаги, пленки или ткани. Импульсная печатающая головка струйного принтера, подобно головке матричного принтера, состоит из вертикального ряда камер, способных нанести на бумагу одну или несколько вертикальных полосок. Число камер, входящих в состав головки, может достигать 48. Это позволяет получать очень качественное изображение.

Существуют как черно-белые, так и цветные струйные принтеры. Последние, кроме головки с черными чернилами, имеют еще печатную головку с чернилами трех или более цветов.

Кроме матричных и струйных принтеров, широкое распространение получили и, так называемые, лазерные принтеры. Принцип их работы достаточно сложен и требует глубокого знания физики, поэтому нами рассматриваться не будет. Эти принтеры при своей относительно высокой стоимости очень экономичны в эксплуатации и намного менее требовательны к качеству бумаги, по сравнению со струйными принтерами.

Устройства связи необходимы для организации взаимодействия отдельных компьютеров между собой, доступа к удаленным принтерам и подключения локальных сетей к общемировой сети Интернет. Примерами таких устройств являются сетевые карты (ethernet cards) и модемы

(modems). Скорость передачи данных устройствами связи измеряется в битах в секунду (а также в кбит/с и мбит/с). Модем, используемый для подключения домашнего компьютера к сети Интернет, обычно обеспечивает пропускную способность до 56 кбит/с, а сетевая карта - до 100 мбит/с.

3.4 Типы и назначение компьютеров

Существование различных типов компьютеров определяется различием задач, для решения которых они предназначены. С течением времени появляются новые типы задач, что приводит к появлению новых типов компьютеров. Поэтому приведенное ниже деление очень условно.

Различают:

- суперкомпьютеры;
- специализированные компьютеры-серверы;
- встроенные компьютеры-невидимки (микропроцессоры);
- персональные компьютеры.

Для выполнения изначального назначения компьютеров - вычислений - на рубеже 60-70 годов были созданы специализированные ЭВМ, так называемые суперкомпьютеры.

Суперкомпьютеры - специальный тип компьютеров, создающихся для решения предельно сложных вычислительных задач (составления прогнозов, моделирования сложных явлений, обработки сверхбольших объемов информации). Принцип работы суперкомпьютера заключается в том, что он способен выполнять несколько операций параллельно. Для этого часто используется многопроцессорная архитектура, когда несколько (иногда десятки или сотни) процессоров объединяются общей шиной в одно целое.

Одной из ведущих компаний мира в производстве суперкомпьютеров является компания Cray Research. Ее основатель, человек-легенда Сеймур Крей, уже в середине 70-х годов построил компьютер Cray-1, который поражал мир своим быстродействием: десятки и даже сотни миллионов арифметических операций в секунду.

Как известно, скорость распространения любого сигнала не превышает скорости света в вакууме - 300 тысяч километров в секунду, или 300 миллионов метров в секунду. Если компьютер выполняет 300 миллионов операций в секунду, то за время выполнения одной операции сигнал успевает пройти не более одного метра. Отсюда следует, что расстояние между частями суперкомпьютера, выполняющими одну операцию, не может превосходить нескольких десятков сантиметров. И действительно, суперкомпьютеры компании Cray были очень компактны и выглядели как "бублик" диаметром менее двух метров. Этот "бублик" занимался только вычислениями. Для общения с человеком и доставки данных для вычислений к "бублику" были подключены несколько достаточно производительных обычных компьютеров.

Компьютер, работающий в локальной или глобальной сети, может специализироваться на оказании информационных услуг другим компьютерам, на обслуживании других компьютеров. Такой компьютер называется сервером от английского слова serve (в переводе - обслуживать, управлять). В локальной сети один из компьютеров может выполнять функции файлового сервера, т. е. использоваться для долговременного хранения файлов.

Основная задача, решаемая файловыми серверами, - организация хранения, доступа и обмена данными (информацией) между компьютерами, людьми и другими источниками и поставщиками информации. Требования к серверам иные, чем к суперкомпьютеру. Важно наличие у них

устройств хранения информации (типа магнитных дисков) большой емкости, скорость же обработки информации не столь критична. Такие компьютеры также могут быть многопроцессорными.

В классе серверов выделяется подкласс суперсерверов, необходимых в тех случаях, когда, с одной стороны, желательна централизация данных, а с другой стороны, к этим данным необходимо обеспечить доступ очень большому количеству потребителей.

Несколько компьютеров, объединенные сетью и работающие над решением одной задачи называют вычислительным кластером. Несколько таких кластеров могут быть расположены в разных уголках мира, но трудиться сообща над решением одной задачи.

Кроме привычных компьютеров с клавиатурами, мониторами, дисководами, сегодняшний мир вещей наполнен компьютерами-невидимками. Микропроцессор представляет собой компьютер в миниатюре. Кроме обрабатывающего блока, он содержит блок управления и даже память (внутренние ячейки памяти). Это значит, что микропроцессор способен автономно выполнять все необходимые действия с информацией. Многие компоненты современного персонального компьютера содержат внутри себя миниатюрный компьютер. Массовое распространение микропроцессоры получили и в производстве, там где управление может быть сведено к отдаче ограниченной последовательности команд.

Микропроцессоры незаменимы в современной технике. Например, управление современным двигателем - обеспечение экономии расхода топлива, ограничение максимальной скорости движения, контроль исправности и т. д. - немыслимо без использования микропроцессоров. Еще одной перспективной сферой их использования является бытовая техника - применение микропроцессоров придает ей

новые потребительские качества.

В 1975 году появился первый персональный компьютер. С самого начала их выпуска стало ясно, что невысокая цена и достаточные вычислительные возможности этого нового класса компьютеров будут способствовать их широкому распространению.

Персональные компьютеры совершили компьютерную революцию в профессиональной деятельности миллионов людей и оказали огромное влияние на все стороны жизни человеческого общества. Компьютеры этого типа стали незаменимым инструментом работы инженеров и ученых. Особо велика их роль при проведении научных экспериментов, требующих сложных и длительных вычислений.

В последние годы появилась разновидность персонального компьютера - так называемый домашний компьютер. По сути, он ничем не отличается от персонального, только используется для других целей: развлекательных и образовательных.

Идея сетевого компьютера, работающего только в сети и представляющего собой упрощенный вариант персонального компьютера, все больше занимает умы разработчиков. Эти компьютеры часто называют сетевыми терминалами. Такому компьютеру не нужно хранить программы или обрабатывать большое количество информации, он в любой момент может получить их по сети с сервера. Примером таких компьютеров могут служить банкоматы, электронные справочные на вокзалах, точки доступа в Интернет-кафе. /10/

3.5 Нейрокомпьютеры

Нейрокомпьютеры	являются	перспективным
направлением	развития	современной

высокопроизводительной вычислительной техники, а теория нейронных сетей и нейроматематика представляют собой приоритетные направления российской вычислительной науки. Основой активного развития нейрокомпьютеров является принципиальное отличие нейросетевых алгоритмов решения задач от однопроцессорных, малопроцессорных, а также транспьютерных алгоритмов. Для данного направления развития вычислительной техники не так важен уровень развития отечественной микроэлектроники, поэтому оно позволяет создать основу построения российской элементной базы суперкомпьютеров.

Нейрокомпьютинг – это научное направление, занимающееся разработкой вычислительных систем шестого поколения – нейрокомпьютеров, которые состоят из большого числа параллельно работающих простых вычислительных элементов (нейронов). Элементы связаны между собой, образуя нейронную сеть. Они выполняют единообразные вычислительные действия и не требуют внешнего управления. Большое число параллельно работающих вычислительных элементов обеспечивают высокое быстродействие.

В настоящее время разработка нейрокомпьютеров ведется в большинстве промышленно развитых стран.

Нейрокомпьютеры позволяют с высокой эффективностью решать целый ряд интеллектуальных задач. Это задачи распознавания образов, адаптивного управления, прогнозирования, диагностики и т.д.

Нейрокомпьютеры отличаются от ЭВМ предыдущих поколений не просто большими возможностями. Принципиально меняется способ использования машины. Место программирования занимает обучение, нейрокомпьютер учится решать задачи.

Обучение – корректировка весов связей, в результате которой каждое входное воздействие приводит к

формированию соответствующего выходного сигнала. После обучения сеть может применять полученные навыки к новым входным сигналам. При переходе от программирования к обучению повышается эффективность решения интеллектуальных задач.

Вычисления в нейронных сетях существенно отличаются от традиционных. В силу высокой параллельности их можно рассматривать как коллективное явление. В нейронной сети нет локальных областей, в которых запоминается конкретная информация. Вся информация запоминается во всей сети.

Толчком к развитию нейрокомпьютинга послужили биологические исследования. По данным нейробиологии нервная система человека и животных состоит из отдельных клеток – нейронов. В мозге человека их число достигает 10^{10} – 10^{12} . Каждый нейрон связан с 10^3 – 10^4 другими нейронами и выполняет сравнительно простые действия. Время срабатывания нейрона – 2-5 мс. Совокупная работа всех нейронов обуславливает сложную работу мозга, который в реальном времени решает сложнейшие задачи. Отличия нейрокомпьютеров от вычислительных устройств предыдущих поколений:

- параллельная работа очень большого числа простых вычислительных устройств обеспечивает огромное быстроедействие;
- нейронная сеть способна к обучению, которое осуществляется путем настройки параметров сети;
- высокая помехо- и отказоустойчивость нейронных сетей;
- простое строение отдельных нейронов позволяет использовать новые физические принципы обработки информации для аппаратных реализаций нейронных сетей.

Нейронные сети находят свое применение в системах распознавания образов, обработки сигналов, предсказания и

диагностики, в робототехнических и бортовых системах. Нейронные сети обеспечивают решение сложных задач за времена порядка времен срабатывания цепочек электронных и/или оптических элементов. Решение слабо зависит от неисправности отдельного нейрона. Это делает их привлекательными для использования в бортовых интеллектуальных системах.

Разработки в области нейрокомпьютеров поддерживаются целым рядом международных и национальных программ. В настоящее время эксплуатируется не менее 50 нейросистем в самых различных областях – от финансовых прогнозов до экспертизы.

Разработки в области нейрокомпьютинга ведутся по следующим направлениям:

- разработки нейроалгоритмов;
- создание специализированного программного обеспечения для моделирования нейронных сетей;
- разработка специализированных процессорных плат для имитации нейросетей;
- электронные реализации нейронных сетей;
- оптоэлектронные реализации нейронных сетей.

В настоящее время наиболее массовым направлением нейрокомпьютинга является моделирование нейронных сетей на обычных компьютерах, прежде всего персональных. Моделирование сетей выполняется для их научного исследования, для решения практических задач, а также при определении значений параметров электронных и оптоэлектронных нейрокомпьютеров.

В основу искусственных нейронных сетей положены следующие черты живых нейронных сетей, позволяющие им хорошо справляться с нерегулярными задачами:

- простой обрабатывающий элемент – нейрон;
- очень большое число нейронов участвует в обработке информации;

- один нейрон связан с большим числом других нейронов (глобальные связи);
- изменяющиеся веса связей между нейронами;
- массивная параллельность обработки информации.

Прототипом для создания нейрона послужил биологический нейрон головного мозга. Биологический нейрон имеет тело, совокупность отростков – дендритов, по которым в нейрон поступают входные сигналы, и отросток – аксон, передающий выходной сигнал нейрона другим клеткам. Точка соединения дендрита и аксона называется синапсом. Упрощенно функционирование нейрона можно представить следующим образом:

- нейрон получает от дендритов набор (вектор) входных сигналов;

- в теле нейрона оценивается суммарное значение входных сигналов. Однако входы нейрона неравнозначны. Каждый вход характеризуется некоторым весовым коэффициентом, определяющим важность поступающей по нему информации. Таким образом, нейрон не просто суммирует значения входных сигналов, а вычисляет скалярное произведение вектора входных сигналов и вектора весовых коэффициентов;

- нейрон формирует выходной сигнал, интенсивность которого зависит от значения вычисленного скалярного произведения. Если оно не превышает некоторого заданного порога, то выходной сигнал не формируется вовсе – нейрон “не срабатывает”;

- выходной сигнал поступает на аксон и передается дендритам других нейронов.

Поведение искусственной нейронной сети зависит как от значения весовых параметров, так и от функции возбуждения нейронов. Известны три основных вида функции возбуждения: пороговая, линейная и сигмоидальная.

Для пороговых элементов выход устанавливается на одном из двух уровней в зависимости от того, больше или меньше суммарный сигнал на входе нейрона некоторого порогового значения. Для линейных элементов выходная активность пропорциональна суммарному взвешенному входу нейрона.

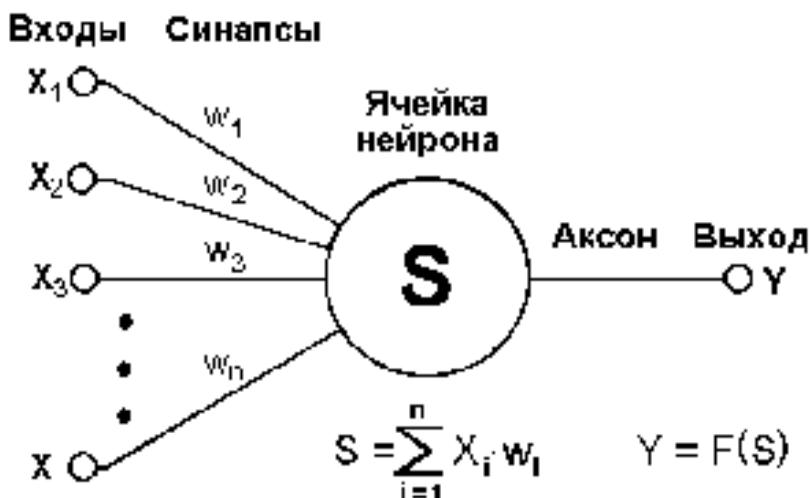


Рис. 3. Искусственный нейрон

Для сигмоидальных элементов в зависимости от входного сигнала, выход варьируется непрерывно, но не линейно, по мере изменения входа. Сигмоидальные элементы имеют больше сходства с реальными нейронами, чем линейные или пороговые, но любой из этих типов можно рассматривать лишь как приближение.

Нейронная сеть представляет собой совокупность большого числа сравнительно простых элементов - нейронов, топология соединений которых зависит от типа сети. Чтобы создать нейронную сеть для решения какой-либо конкретной задачи, мы должны выбрать, каким образом следует

соединять нейроны друг с другом, и соответствующим образом подобрать значения весовых параметров на этих связях. Может ли влиять один элемент на другой, зависит от установленных соединений. Вес соединения определяет силу влияния.

3.6 Модели нейронных сетей

Модель Маккалоха. Теоретические основы нейроматематики были заложены в начале 40-х годов. В 1943 году У. Маккалох и его ученик У. Питтс сформулировали основные положения теории деятельности головного мозга. Ими были получены следующие результаты:

- разработана модель нейрона как простейшего процессорного элемента, выполняющего вычисление переходной функции от скалярного произведения вектора входных сигналов и вектора весовых коэффициентов;
- предложена конструкция сети таких элементов для выполнения логических и арифметических операций;
- сделано основополагающее предположение о том, что такая сеть способна обучаться, распознавать образы, обобщать полученную информацию.

Несмотря на то, что за прошедшие годы нейроматематика ушла далеко вперед, многие утверждения Маккалоха остаются актуальными и поныне. В частности, при большом разнообразии моделей нейронов принцип их действия, заложенный Маккалохом и Питтсом, остается неизменным. Недостатком данной модели является сама модель нейрона "пороговой" вид переходной функции. В формализме У. Маккалоха и У. Питтса нейроны имеют состояния 0, 1 и пороговую логику перехода из состояния в состояние. Каждый нейрон в сети определяет взвешенную сумму состояний всех других нейронов и сравнивает ее с порогом, чтобы определить свое собственное состояние.

Пороговый вид функции не предоставляет нейронной сети достаточную гибкость при обучении и настройке на заданную задачу. Если значение вычисленного скалярного произведения, даже незначительно, не достигает до заданного порога, то выходной сигнал не формируется вовсе и нейрон "не срабатывает". Это значит, что теряется интенсивность выходного сигнала (аксона) данного нейрона и, следовательно, формируется невысокое значение уровня на взвешенных входах в следующем слое нейронов.

Модель Розенблата. Серьезное развитие нейрокибернетика получила в работах американского нейрофизиолога Френсиса Розенблата (Корнелльский университет). В 1958 году он предложил свою модель нейронной сети. Розенблат ввел в модель Маккаллока и Питтса способность связей к модификации, что сделало ее обучаемой. Эта модель была названа перцептроном. Первоначально перцептрон представлял собой однослойную структуру с жесткой пороговой функцией процессорного элемента и бинарными или многозначными входами. Первые перцептроны были способны распознавать некоторые буквы латинского алфавита. Впоследствии модель перцептрона была значительно усовершенствована.

Перцептрон применялся для задачи автоматической классификации, которая в общем случае состоит в разделении пространства признаков между заданным количеством классов. В двухмерном случае требуется провести линию на плоскости, отделяющую одну область от другой. Перцептрон способен делить пространство только прямыми линиями (плоскостями).

Алгоритм обучения перцептрона выглядит следующим образом:

- системе предъявляется эталонный образ;
- если выходы системы срабатывают правильно, весовые коэффициенты связей не изменяются;

- если выходы срабатывают неправильно, весовым коэффициентам дается небольшое приращение в сторону повышения качества распознавания.

Серьезным недостатком перцептрона является то, что не всегда существует такая комбинация весовых коэффициентов, при которой имеющееся множество образов будет распознаваться данным перцептроном. Причина этого недостатка состоит в том, что лишь небольшое количество задач предполагает, что линия, разделяющая эталоны, будет прямой. Обычно это достаточно сложная кривая, замкнутая или разомкнутая. Если учесть, что однослойный перцептрон реализует только линейную разделяющую поверхность, применение его там, где требуется нелинейная, приводит к неверному распознаванию (эта проблема называется линейной неразделимостью пространства признаков). Выходом из этого положения является использование многослойного перцептрона, способного строить ломаную границу между распознаваемыми образами.

Описанная проблема не является единственной трудностью, возникающей при работе с перцептронами - также слабо формализован метод обучения перцептрона.

Перцептрон поставил ряд вопросов, работа над решением которых привела к созданию более "разумных" нейронных сетей и разработке методов, нашедших применение не только в нейрокибернетике (например, метод группового учета аргументов, применяемый для идентификации математических моделей).

Модель Хопфилда. В 70-е годы интерес к нейронным сетям значительно упал, однако работы по их исследованию продолжались. Был предложен ряд интересных разработок, таких, например, как когнитрон, и т.п.), позволяющих распознавать образы независимо от поворота и изменения масштаба изображения.

Автором когнитрона является японский ученый И.

Фукушима.

Новый виток быстрого развития моделей нейронных сетей, который начался лет 15 тому назад, связан с работами Амари, Андерсона, Карпентера, Кохонена и других, и в особенности, Хопфилда, а также под влиянием обещающих успехов оптических технологий и зрелой фазы развития СБИС для реализации новых архитектур.

Начало современному математическому моделированию нейронных вычислений было положено работами Хопфилда в 1982 году, в которых была сформулирована математическая модель ассоциативной памяти на нейронной сети.

Показано, что для однослойной нейронной сети со связями типа "все на всех" характерна сходимости к одной из конечного множества равновесных точек, которые являются локальными минимумами функции энергии, содержащей в себе всю структуру взаимосвязей в сети. Понимание такой динамики в нейронной сети было и у других исследователей. Однако, Хопфилд и Тэнк показали как конструировать функцию энергии для конкретной оптимизационной задачи и как использовать ее для отображения задачи в нейронную сеть. Этот подход получил развитие и для решения других комбинаторных оптимизационных задач. Привлекательность подхода Хопфилда состоит в том, что нейронная сеть для конкретной задачи может быть запрограммирована без обучающих итераций. Веса связей вычисляются на основании вида функции энергии, сконструированной для этой задачи.

Развитием модели Хопфилда для решения комбинаторных оптимизационных задач и задач искусственного интеллекта является машина Больцмана, предложенная и исследованная Джефери Е, Хинтоном и Р. Земелом. В ней, как и в других моделях, нейрон имеет состояния 1, 0 и связь между нейронами обладает весом. Каждое состояние сети характеризуется определенным

значением функции консенсуса (аналог функции энергии). Максимум функции консенсуса соответствует оптимальному решению задачи.

Модель сети с обратным распространением. Способом обратного распространения (back propagation) называется способ обучения многослойных нейронных сетей (НС).

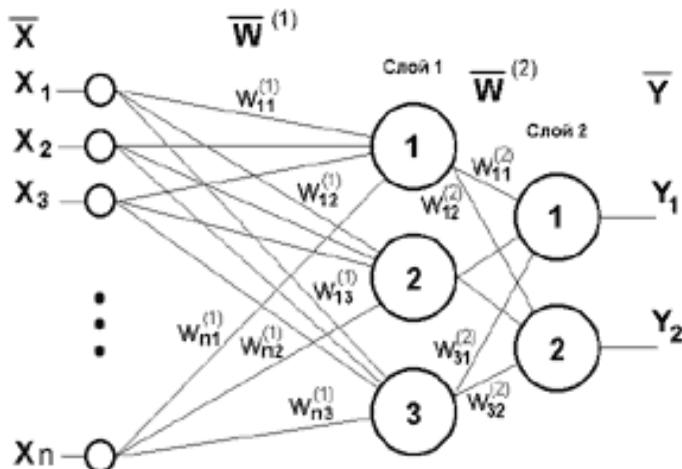


Рис. 4. Многослойная нейронная сеть

В таких НС связи между собой имеют только соседние слои, при этом каждый нейрон предыдущего слоя связан со всеми нейронами последующего слоя. Нейроны обычно имеют сигмоидальную функцию возбуждения. Первый слой нейронов называется входным и содержит число нейронов, соответствующее распознаваемому образу. Последний слой нейронов называется выходным и содержит столько нейронов, сколько классов образов распознается. Между входным и выходным слоями располагается один или более скрытых (теневых) слоев. Определение числа скрытых слоев и числа нейронов в каждом слое для конкретной задачи является неформальной задачей. Принцип обучения такой

нейронной сети базируется на вычислении отклонений значений сигналов на выходных процессорных элементах от эталонных и обратном "прогоне" этих отклонений до породивших их элементов с целью коррекции ошибки.

Еще в 1974 году Поль Дж. Вербос изобрел значительно более эффективную процедуру для вычисления величины, называемой производной ошибки по весу. Процедура, известная теперь как алгоритм обратного распространения, стала одним из наиболее важных инструментов в обучении нейронных сетей. Однако этому алгоритму свойственны и недостатки, главный из которых - отсутствие сколько-нибудь приемлемых оценок времени обучения. Понимание, что сеть в конце концов обучится мало утешает, если на это могут уйти годы. Тем не менее, алгоритм обратного распространения имеет широчайшее применение. /11/

3.7 Алгоритмы обучения персептрона

Простой метод. Персептрон является двухуровневой, не рекуррентной сетью, вид которой показан на рис. 5.

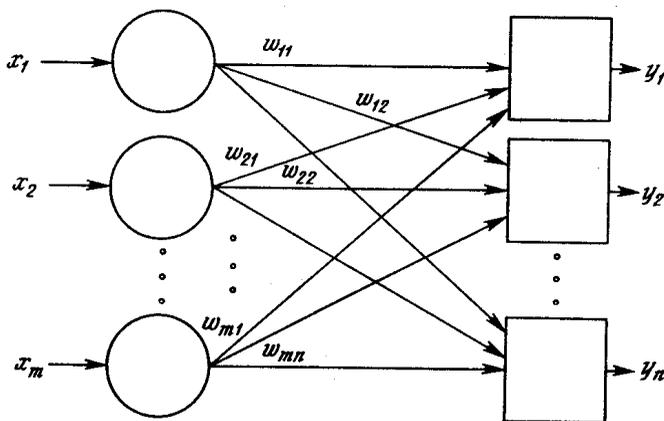


Рис. 5. Однослойная нейронная сеть.

Она использует алгоритм обучения с учителем; другими словами, обучающая выборка состоит из множества входных векторов, для каждого из которых указан свой требуемый вектор цели. Компоненты входного вектора представлены непрерывным диапазоном значений; компоненты вектора цели являются двоичными величинами (0 или 1). После обучения сеть получает на входе набор непрерывных входов и вырабатывает требуемый выход в виде вектора с бинарными компонентами.

Алгоритм обучения осуществляется следующим образом:

1. Рандомизируются все веса сети в малые величины.

2. На вход сети подается входной обучающий вектор X и вычисляется сигнал NET от каждого нейрона, используя стандартное выражение:

$$NET_j = \sum_i x_i w_{ij}$$

3. Вычисляется значение пороговой функции активации для сигнала NET от каждого нейрона следующим образом:

- $OUT_j = 1$, если NET_j больше чем порог θ_j ,

- $OUT_j = 0$ в противном случае.

Здесь θ_j представляет собой порог, соответствующий нейрону j (в простейшем случае, все нейроны имеют один и тот же порог).

4. Вычисляется ошибка для каждого нейрона посредством вычитания полученного выхода из требуемого выхода:

$$error_j = target_j - OUT_j.$$

5. Каждый вес модифицируется следующим образом:

$$W_{ij}(t+1) = w_{ij}(t) + a x_i \text{error}_j.$$

6. Повторяются шаги со второго по пятый до тех пор, пока ошибка не станет достаточно малой.

Метод обучения Уидроу-Хоффа. Как мы видели, персептрон ограничивается бинарными выходами. Уидроу вместе со студентом университета Хоффом расширили алгоритм обучения персептрона на случай непрерывных выходов, используя сигмоидальную функцию. Кроме того, они разработали математическое доказательство того, что сеть при определенных условиях будет сходиться к любой функции, которую она может представить. Их первая модель – Адалин – имеет один выходной нейрон, более поздняя модель – Мадалин – расширяет ее на случай с многими выходными нейронами.

Выражения, описывающие процесс обучения Адалина, очень схожи с персептронными. Существенные отличия имеются в четвертом шаге, где используются непрерывные сигналы NET вместо бинарных OUT. Модифицированный шаг 4 в этом случае реализуется следующим образом:

4. Вычисляется ошибка для каждого нейрона посредством вычитания полученного выхода из требуемого выхода:

$$\text{error}_j = \text{target}_j - \text{NET}_j. /12/$$

3.8 Квантовые компьютеры

Квантовый компьютер — это гипотетическое вычислительное устройство, существенно использующее при работе квантовомеханические эффекты, такие как квантовая суперпозиция и квантовый параллелизм. Предполагается, что это позволит преодолеть некоторые ограничения классических компьютеров.

Кубиты Идея квантовых вычислений, впервые

высказанная Ю. И. Маниным и Р. Фейнманом состоит в том, что квантовая система из L двухуровневых квантовых элементов (кубитов) имеет 2^L линейно независимых состояний, а значит, вследствие принципа квантовой суперпозиции, 2^L -мерное гильбертово пространство состояний. Операция в квантовых вычислениях соответствует повороту в этом пространстве. Таким образом, квантовое вычислительное устройство размером L кубит может выполнять параллельно 2^L операций.

Предположим, что имеется один кубит. В таком случае после измерения, в так называемой классической форме, результат будет 0 или 1. В действительности кубит квантовый объект и поэтому, вследствие принципа неопределенности, может быть и 0, и 1 с определенной вероятностью. Если кубит равен 0 (или 1) со стопроцентной вероятностью, его состояние обозначается с помощью символа $|0\rangle$ (или $|1\rangle$) — в обозначениях Дирака. $|0\rangle$ и $|1\rangle$ — это базовые состояния. В общем случае квантовое состояние кубита находится между базовыми и записывается, в виде $a|0\rangle + b|1\rangle$, где $|a|^2$ и $|b|^2$ — вероятности измерить 0 или 1 соответственно; $a, b \in \mathbb{C}$. Более того, сразу после измерения кубит переходит в базовое квантовое состояние, аналогичное классическому результату. Пример:

Имеется кубит в квантовом состоянии

$$\frac{4}{5}|0\rangle - \frac{3}{5}|1\rangle$$

В этом случае, вероятность получить при измерении:

0 составляет $(4/5)^2 = 16/25 = 64\%$,

1 соответственно $(-3/5)^2 = 9/25 = 36\%$.

Допустим, при измерении мы получили 1.

Тогда кубит перескакивает в новое квантовое состояние $0*|0\rangle+1*|1\rangle=|1\rangle$, то есть, при следующем измерении этого кубита мы получим 1 со стопроцентной вероятностью.

Перейдём к системе из двух кубитов. Измерение каждого из них может дать 0 или 1. Поэтому у системы 4 классических состояния: 00, 01, 10 и 11. Аналогичные им базовые квантовые состояния: $|00\rangle$, $|01\rangle$, $|10\rangle$ и $|11\rangle$. И наконец, общее квантовое состояние системы имеет вид $a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$. Теперь $|a|^2$ — вероятность измерить 00 и т. д. Отметим, что $|a|^2+|b|^2+|c|^2+|d|^2=1$ как полная вероятность.

В общем случае, системы из L кубитов у неё 2^L классических состояний (00000, 00001, ... , 11111), каждое из которых может быть измерено с вероятностями 0—100 %.

Таким образом, одна операция над группой кубитов затрагивает все значения, которые она может принимать, в отличие от классического бита. Это и обеспечивает беспрецедентный параллелизм вычислений.

Вычисление Упрощённая схема вычисления на квантовом компьютере выглядит так: берётся система кубитов, на которой записывается начальное состояние. Затем состояние системы или её подсистем изменяется посредством базовых квантовых операций. В конце измеряется значение, и это результат работы компьютера.

Оказывается, что для построения любого вычисления достаточно двух базовых операций. Квантовая система даёт результат, только с некоторой вероятностью являющийся правильным. Но за счёт небольшого увеличения операций в алгоритме можно сколь угодно приблизить вероятность получения правильного результата к единице.

С помощью базовых квантовых операций можно симулировать работу обычных логических элементов, из которых сделаны обычные компьютеры. Поэтому любую

задачу, которая решена сейчас, квантовый компьютер решит и почти за такое же время. Следовательно, новая схема вычислений будет не слабее нынешней.

Чем же квантовый компьютер лучше классического? Большая часть современных ЭВМ работают по такой же схеме: n бит памяти хранят состояние и каждый такт времени изменяются процессором. В квантовом случае, система из n кубитов находится в состоянии, являющимся суперпозицией всех базовых состояний, поэтому изменение системы касается всех 2^n базовых состояний одновременно. Теоретически новая схема может работать намного (в экспоненциальное число раз) быстрее классической. Практически, (квантовый) алгоритм Гровера поиска в базе данных показывает квадратичный прирост мощности против классических алгоритмов. Пока в природе их не существует.

Алгоритмы:

- Алгоритм Гровера позволяет найти решение уравнения $f(x) = 1$, $0 \leq x < N$ за время $O(\sqrt{N})$.

- Алгоритм Шора позволяет разложить натуральное число n на простые множители за полиномиальное от $\log(n)$ время (для обычного компьютера полиномиальный алгоритм неизвестен).

- Алгоритм Дойча — Джоза позволяет «за одно вычисление» определить, является ли функция двоичной переменной $f(n)$ постоянной ($f_1(n) = 0$, $f_2(n) = 1$ независимо от n) или «сбалансированной» ($f_3(0) = 0$, $f_3(1) = 1$; $f_4(1) = 1$, $f_4(0) = 0$).

Было показано, что не для всякого алгоритма возможно «квантовое ускорение». То есть не все алгоритмы имеет смысл реализовывать в квантовом компьютере. Скорее всего, квантовый компьютер будущего будет представлять некий блок, реализующий вычисления по квантовому принципу в составе более-менее привычно организованного компьютера.

Квантовая телепортация Алгоритм телепортации

реализует точный перенос состояния одного кубита (или системы) на другой. В простейшей схеме используются 4 кубита: источник, приемник и два вспомогательных. Отметим, что в результате работы алгоритма первоначальное состояние источника разрушится — это пример действия общего принципа невозможности клонирования — невозможно создать точную копию квантового состояния, не разрушив оригинал. На самом деле, довольно легко создать одинаковые состояния на кубитах. К примеру, измерив 3 кубита, мы переведём каждый из них в базовые состояния (0 или 1) и хотя бы на двух из них они совпадут. Не получится скопировать произвольное состояние, и телепортация — замена этой операции.

Телепортация позволяет передавать квантовое состояние системы с помощью обычных классических каналов связи. Таким образом, можно, в частности, получить связанное состояние системы, состоящей из подсистем, удалённых на большое расстояние.

Применение квантовых компьютеров. Может показаться, что квантовый компьютер — это разновидность аналоговой вычислительной машины. Но это не так: по своей сути это цифровое устройство, но с аналоговой природой.

Основные проблемы, связанные с созданием и применением квантовых компьютеров:

- необходимо обеспечить высокую точность измерений;
- внешние воздействия могут разрушить квантовую систему или внести в нее искажения.

Приложения криптографии Благодаря огромной скорости разложения на простые множители, квантовый компьютер позволит расшифровывать сообщения, закодированные при помощи многих популярных криптографических алгоритмов, таких как RSA. До сих пор этот алгоритм считается сравнительно надёжным, так как

эффективный способ разложения чисел на простые множители для классического компьютера в настоящее время неизвестен.

Применение идей квантовой механики уже открыли новую эпоху в области криптографии, так как методы квантовой криптографии открывают новые возможности в области передачи сообщений, которые даже теоретически нельзя «расшифровать». Уже существуют коммерческие образцы систем подобного рода.

Реализация На данный момент, наибольший квантовый компьютер составлен из 7 кубитов. Этого достаточно, чтобы реализовать алгоритм Шора и разложить 15 на простые множители 3 и 5.

Канадская компания D-Wave продемонстрировала в феврале 2007 года образец квантового компьютера, состоящего из 16 кубит. Устройство получило название Orion. /13/

3.9 Биокomпьютеры

Биокomпьютер Эдлмана — в 1994 году Леонард Эдлман (en: Leonard Adleman), профессор университета Южной Калифорнии, на примере биомолекулярного кода продемонстрировал, что с помощью пробирки с ДНК можно весьма эффективно решать классическую комбинаторную «задачу о коммивояжере» (кратчайший маршрут обхода вершин графа). Классические компьютерные архитектуры требуют множества вычислений с опробованием каждого варианта.

Метод ДНК позволяет сразу сгенерировать все возможные варианты решений с помощью известных биохимических реакций. Затем возможно быстро отфильтровать именно ту молекулу-нить, в которой закодирован нужный ответ – в данном случае решения задачи

о коммивояжере – самую короткую молекулу.

Проблемы, возникающие при этом:

- Требуется чрезвычайно трудоемкая серия реакций, проводимых под тщательным наблюдением.
- Существует проблема масштабирования задачи.

Биокомпьютер Эдлмана отыскивал оптимальный маршрут обхода для 7 вершин графа. Но чем больше вершин графа, тем больше биокомпьютеру требуется ДНК-материала.

Было подсчитано, что при масштабировании методики Эдлмана для решения задачи обхода не 7 пунктов, а около 200, вес ДНК для представления всех возможных решений превысит вес нашей планеты.

Биомолекулярные вычисления или молекулярные компьютеры или даже ДНК- или РНК-вычисления — все эти термины появились на стыке таких различных наук как молекулярная генетика и вычислительная техника.

Биомолекулярные вычисления — это собирательное название для различных техник, так или иначе связанных с ДНК или РНК. При ДНК-вычислениях данные представляются не в форме нулей и единиц, а в виде молекулярной структуры, построенной на основе спирали ДНК. Роль программного обеспечения для чтения, копирования и управления данными выполняют особые ферменты.

Основой всей системы хранения биологической информации, а стало быть, и ДНК-компьютеров, является способность атомов водорода, входящих в азотистые соединения (аденин, тимин, цитозин и гуанин), при определенных условиях притягиваться друг к другу, образуя невалентно связанные пары. С другой стороны, эти вещества могут валентно связываться с сочетаниями молекулы сахара (дезоксирибозы) и фосфата, образуя так называемые нуклеотиды. Нуклеотиды, в свою очередь, легко образуют полимеры длиной в десятки миллионов оснований. В этих

супермолекулах фосфат и дезоксирибоза играют роль поддерживающей структуры (они чередуются в цепочке), а азотистые соединения кодируют информацию.

Молекула получается направленной: начинается с фосфатной группы и заканчивается дезоксирибозой. Длинные цепочки ДНК называют нитями, короткие — олигонуклеотидами. Каждой молекуле ДНК соответствует еще одна ДНК — так называемое дополнение Ватсона — Крика. Она имеет противоположную направленность, нежели оригинальная молекула. В результате притяжения аденина к тимину и цитозина к гуанину получается знаменитая двойная спираль, обеспечивающая возможность удвоения ДНК при размножении клетки. Задача удвоения решается с помощью специального белка-энзимы — полимеразы. Синтез начинается только если с ДНК прикреплен кусочек ее дополнения, Данное свойство активно используется в молекулярной биологии и молекулярных вычислениях. По сути своей полимеразы — это реализация машины Тьюринга, состоящая из двух лент и программируемого пульта управления. Пульт считывает данные с одной ленты, обрабатывает их по некоторому алгоритму и записывает на другую ленту. Полимераза также последовательно считывает исходные данные с одной ленты (ДНК) и на их основе формирует ленту с результатами вычислений (дополнение Ватсона — Крика). /14/

4. СБОР И ОБРАБОТКА ИНФОРМАЦИИ

4.1 Сбор и обработка экспериментальных результатов. Платы сбора данных

Благодаря широкому распространению и доступности IBM-совместимые ПК стали все чаще использоваться в системах сбора и обработки данных. О тех специальных

средствах, которые необходимы для подключения к компьютеру приборов и датчиков, и пойдет речь в данной главе.

Платы сбора данных (ПСД) применяются в научных исследованиях, на производстве, в медицине и во многих других областях. Они входят в состав измерительных комплексов, автоматизированных производственных систем, систем сбора информации. Мы рассмотрим имеющиеся на российском рынке платы сбора и обработки данных, подключаемые к IBM-совместимым ПК, обсудим их основные параметры.



Рис. 6. Внешний вид платы сбора данных.

Зная возможности современных ПСД и вносимые ими погрешности, а также структурные, схемотехнические и конструкторско-технологические решения, необходимо сделать квалифицированный выбор платы для конкретной задачи или сформировать обоснованное техническое задание на ее разработку. Выбор ПСД определяется рядом требований к скорости сбора данных, числу каналов, уровням входных

напряжений, разрешающей способности и полосе частот АЦП. Немаловажным фактором является и стоимость.

Платы сбора данных не стандартизированы и могут в разных комбинациях включать следующие устройства:

1. Аналого-цифровые преобразователи АЦП - одно из самых сложных устройств, входящих в состав ПСД. Служит для преобразования непрерывных (аналоговых) сигналов в цифровую форму, воспринимаемую компьютером. Основные параметры - время преобразования, число разрядов, погрешность преобразования в конечной точке шкалы, интегральная и дифференциальная нелинейности.

2. Схемы цифрового (дискретного) ввода-вывода. Позволяют вводить и выводить информацию, представленную в цифровом виде. Могут применяться для подключения датчиков, управления оборудованием, генерации тестовых сигналов, а также для связи с периферийными устройствами. Основные параметры – число цифровых линий, скорость обмена данными и нагрузочные характеристики.

3. Цифро-аналоговые преобразователи (ЦАП). Применяются для моделирования аналоговых сигналов и управления некоторыми видами оборудования. Основные параметры - время установления напряжения, скорость нарастания сигнала, разрядность и диапазон изменения выходного напряжения.

4. Счетчики и таймеры. Могут использоваться для запуска АЦП, генерации прерываний, подсчета внешних импульсов (событий), измерения частоты, измерения длительности цифровых сигналов и генерации прямоугольных импульсов. Основные параметры — разрядность и тактовая частота.

5. Аналоговые фильтры. Предназначены для фильтрации нежелательных частотных составляющих входного аналогового сигнала и устранения частотных

наложений в спектре сигнала на выходе АЦП.

6. Контроллеры двигателей. Представляют собой специализированные схемы управления и контроля параметров работы двигателей.

7. Пиковые детекторы. Регистрируют превышение сигналом пиковых (заранее оговоренных) значений входного сигнала.

8. Аналоговые схемы гальванической развязки. Служат для изоляции шины земли компьютера от аналоговых цепей устройств. В медицинских системах позволяют обеспечить безопасность при измерениях на открытых участках тела пациента. Все перечисленные устройства могут вносить искажения в форму и величину измеренного сигнала.

9. Цифровые схемы гальванической развязки. Изолируют цепи компьютера от цепей подключенных к нему цифровых приборов, что улучшает помехоустойчивость измерительных и управляющих комплексов, защищает ПК от разрядов статического электричества, предотвращает возникновение проблем, связанных с некачественным заземлением приборов. Основной параметр-напряжение пробоя.

10. Схемы формирования запросов прерываний. Предназначены для формирования запросов прерываний, необходимых для быстрой реакции процессора на внешние события, например на поступление данных. Используются в измерительных и управляющих системах реального времени.

11. Схемы прямого доступа к памяти. Используются при высокоскоростном обмене данными между ПСД и памятью ЭВМ, который происходит без участия процессора. Этот режим наиболее эффективен при передаче больших массивов данных.

12. Цифровые сигнальные процессоры (Digital Signal Processors, DSP). Применяются при необходимости сложной обработки поступающих сигналов или при наличии огромных

потоков данных, с которыми не справляется основной процессор ПК.

При выборе платы для решения каждой конкретной задачи следует учитывать все перечисленные выше компоненты.

Кроме уже названных устройств, платы сбора данных оснащаются различными устройствами предварительной обработки сигналов. Схемы выборки и хранения используются для запоминания мгновенного значения входного сигнала на время аналого-цифрового преобразования. Для минимизации погрешностей работы схем АЦП во время преобразования величина входного сигнала АЦП должна быть неизменной.

4.2 Аппаратные средства ПСД

Инструментальные усилители с переключаемым коэффициентом усиления применяются в тех случаях, когда приходится иметь дело либо с низкоуровневыми сигналами, либо с сигналами, характеризующимися широким динамическим диапазоном. Усилители могут иметь однополюсные или дифференциальные входы.

Схемы защиты входа препятствуют попаданию сигналов, параметры которых превышают допустимые, на последующие схемы (например, АЦП).

Схемы аналоговых мультиплексоров позволяют поочередно подавать сигналы от множества различных источников на один АЦП, что снижает стоимость ПСД.

Схемы буферной памяти дают возможность накапливать информацию о поступающем сигнале (число запоминаемых выборок сигнала зависит от объема буфера) до наступления некоторого события, например, превышения сигналом порогового уровня. Могут использоваться и в тех ситуациях, когда задержка обмена со стороны компьютера

приводит к потере данных.

Платы сбора данных разных модификаций и разных фирм-производителей существенно отличаются друг от друга как по функциональным возможностям, так и по характеристикам. Во многих случаях не обязательно покупать дорогие ПСД, однако, чтобы не ошибиться в выборе, необходимо тщательно проанализировать наиболее жесткие требования конкретной задачи, и в первую очередь – к аналоговой части ПСД. Важно найти такое устройство, функции и параметры которого наилучшим образом соответствуют вашим целям.

При выборе ПСД также не следует забывать о необходимости написания программного обеспечения для работы с ними. Иногда наличие в комплекте поставки соответствующих программ может оказаться решающим фактором выбора.

4.3 Параметры аналогового тракта ПСД

Наиболее сложной и ответственной подсистемой ПСД является аналоговый тракт, включающий АЦП, мультиплексоры, устройства выборки-запоминания и инструментальные усилители. Сделать его высококачественным очень непросто, так как в компьютере имеется множество источников сильных импульсных помех.

Основные параметры аналогового тракта ПСД:

- время преобразования - интервал времени с момента подачи команды начала преобразования до появления на выходе АЦП цифрового кода, соответствующего входному напряжению;
- погрешность сдвига - смещение характеристики преобразования в точке начала координат графика (см. рис. 7);

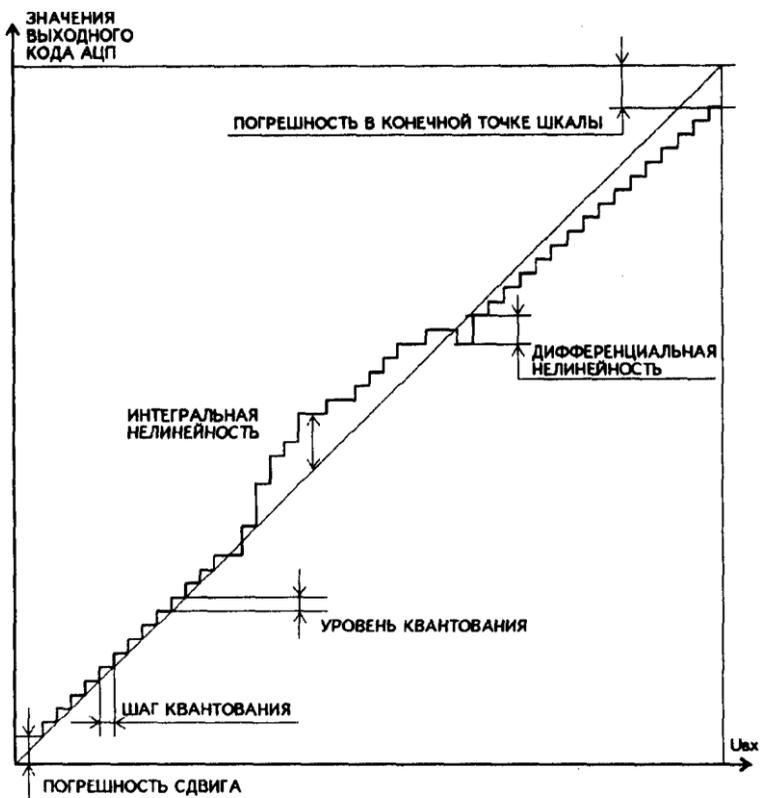


Рис. 7. Характеристики АЦП.

- погрешность усиления (погрешность преобразования в конечной точке шкалы) - отклонение характеристики преобразования в конечной точке от идеальной после устранения сдвига;
- интегральная нелинейность - максимальное отклонение квантованного сигнала от идеальной характеристики преобразования за вычетом половины идеальной величины шага квантования;
- дифференциальная нелинейность - максимальная

разность выходных кодов АЦП при подаче на вход последовательно двух значений сигналов, различающихся на шаг квантования, т.е. на минимальную величину $1/2N$, где N - число разрядов АЦП;

- апертурная задержка - задержка момента фактического начала преобразования относительно момента поступления команды преобразования;

- апертурная неопределенность - переменная составляющая апертурной задержки;

- разрешающая способность - параметр, обычно равный половине шага квантования АЦП (приводится в рекламных материалах некоторых поставщиков);

- входная полоса частот преобразователя – диапазон частот входного сигнала, в котором параметры аналогового тракта выдерживаются с заданной точностью;

- коэффициент нелинейных искажений – отношение суммы мощностей гармоник к мощности основной спектральной составляющей;

- отношение сигнал/шум (ОСШ) - отношение среднеквадратических значений сигнала и шума в заданной полосе частот,

- эффективная разрядность - разрядность, учитывающая все виды погрешностей (все ошибки преобразователя, обусловленные дифференциальной и интегральной нелинейностями, апертурной неопределенностью и пропуском кодов, выступают как составляющие некоторой суммарной среднеквадратической погрешности);

- межканальное проникание - уровень помехи, проникающей в выбранный канал мультиплектора из соседних каналов;

- температурный коэффициент преобразователя - коэффициент температурной зависимости всех приведенных выше параметров.

Число разрядов в регистре АЦП не может быть основным показателем качества платы. Говорить о разрядности АЦП имеет смысл только тогда, когда величина приведенного ко входу шума не превышает единицы младшего разряда. Чтобы оценить аналоговый тракт данной конкретной платы, нужно обратить внимание на показатели ОСШ и эффективной разрядности АЦП.

Для измерения всех перечисленных параметров аналогового тракта необходимы специальная аппаратура и соответствующие методики. Справочные данные на используемые в ПСД микросхемы АЦП определяют наилучшие возможные характеристики платы, однако кроме АЦП в аналоговых цепях устанавливается множество других элементов, сказывается влияние цифровых схем и помех от монитора, источника питания, вентиляторов и других устройств. Более того, даже знание точных значений параметров аналогового тракта ПСД не гарантирует построения системы сбора данных с нужными характеристиками. Важно правильно включить ПСД в измерительную систему. На рис. 8 показаны два варианта соединения источника сигнала с инструментальным усилителем.

Первый вариант рассчитан на малые расстояния, второй применяется при подключении удаленных датчиков. Экранированная витая пара присоединяется к усилителю так, чтобы наведенные помехи были для дифференциального усилителя синфазными, а сигнал датчика - противофазным. Если же необходимо проводить измерения с особо высокой точностью (14 разрядов и выше), приходится применять кабель с двойным экраном для защиты провода от электрических и магнитных полей. Обычно это экран из медной и стальной оплетки. /15/

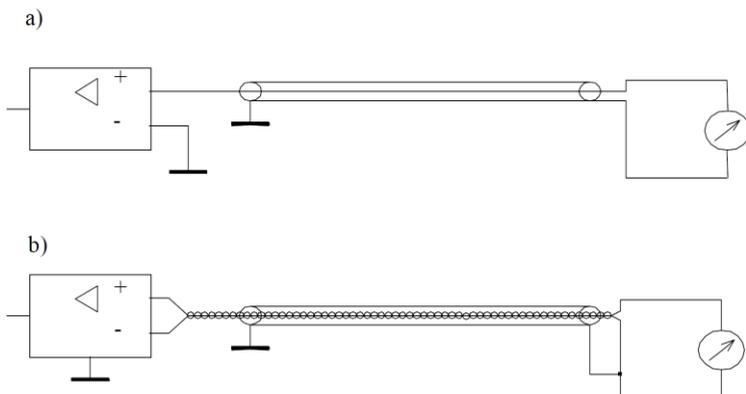


Рис. 8. Два варианта соединения источника сигнала с инструментальным усилителем. Подключение источника сигнала к однополюсному (а) и к дифференциальному (b) входам.

4.4 Обработка экспериментальных результатов

4.4.1 Интерполяция

Интерполяция — в вычислительной математике способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений.

Многим из тех, кто сталкивается с научными и инженерными расчётами часто приходится оперировать наборами значений, полученных экспериментальным путём или методом случайной выборки. Как правило, на основании этих наборов требуется построить функцию, на которую могли бы с высокой точностью попадать другие получаемые значения. Такая задача называется аппроксимацией кривой. Интерполяцией называют такую разновидность аппроксимации, при которой кривая построенной функции проходит точно через имеющиеся точки данных.

Существует также близкая к интерполяции задача, которая заключается в аппроксимации какой-либо сложной функции другой, более простой функцией. Если некоторая функция слишком сложна для производительных вычислений, можно попытаться вычислить её значение в нескольких точках, а по ним построить, то есть интерполировать, более простую функцию. Разумеется, использование упрощенной функции не позволяет получить такие же точные результаты, какие давала бы первоначальная функция. Но в некоторых классах задач достигнутый выигрыш в простоте и скорости вычислений может перевесить получаемую погрешность в результатах.

Рассмотрим систему несовпадающих точек x_i ($i \in 0, 1, \dots, N$) из некоторой области D . Пусть значения функции f известны только в этих точках: $y_i = f(x_i)$, $i = 1, \dots, N$.

Задача интерполяции состоит в поиске такой функции F из заданного класса функций, что $F(x_i) = y_i$, $i = 1, \dots, N$.

- Точки x_i называют узлами интерполяции, а их совокупность — интерполяционной сеткой.
- Пары (x_i, y_i) называют точками данных или базовыми точками.
- Разность между «соседними» значениями $\Delta x_i = x_i - x_{i-1}$ — шагом интерполяционной сетки. Он может быть как переменным так и постоянным.
- Функцию $F(x)$ — интерполирующей функцией или интерполянтном.

Интерполяция помогает нам узнать какое значение может иметь такая функция в точке, отличной от указанных.

К настоящему времени существует множество различных способов интерполяции. Выбор наиболее подходящего алгоритма зависит от ответов на вопросы: как точен выбираемый метод, каковы затраты на его использование, насколько гладкой является

интерполяционная функция, какого количества точек данных она требует и т. п.

4.4.2 Способы интерполяции

На практике чаще всего применяют интерполяцию полиномами. Это связано прежде всего с тем, что полиномы легко вычислять, легко аналитически находить их производные и множество полиномов плотно в пространстве непрерывных функций (теорема Вейерштрасса).

Линейная (геометрическая) интерполяция — интерполяция алгебраическим двучленом $P_1(x) = ax + b$ функции f , заданной в двух точках x_0 и x_1 отрезка $[a, b]$.

Геометрическая интерпретация. Геометрически это означает замену графика функции f прямой, проходящей через точки $(x_0, f(x_0))$ и $(x_1, f(x_1))$.

Уравнение такой прямой имеет вид:

$$\frac{y - f(x_0)}{f(x_1) - f(x_0)} = \frac{x - x_0}{x_1 - x_0}$$

отсюда для $x \in [x_0, x_1]$

$$f(x) \approx y = P_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

Это и есть формула линейной интерполяции, при этом

$$f(x) = P_1(x) + R_1(x)$$

где $R_1(x)$ — погрешность формулы:

$$R_1(x) = \frac{f''(\psi)}{2}(x - x_0)(x - x_1), \quad \psi \in [x_0, x_1]$$

Справедлива оценка:

$$|R_1(x)| \leq \frac{M_2}{2} \max |(x-x_0)(x-x_1)| = \frac{M_2 h^2}{8},$$

$$M_2 = \max_{[a,b]} |f''(x)|, \quad h = x_1 - x_0.$$

Линейная интерполяция применяется для уплотнения таблиц, то есть когда необходимо заполнить свободные ячейки данными.

Если использовать для интерполяции более сложную зависимость, то необходимо применять метод конечных разностей.

Формула линейной интерполяции является частным случаем интерполяционной формулы Лагранжа и интерполяционной формулой Ньютона.

Интерполяционный многочлен Лагранжа — многочлен минимальной степени, принимающий данные значения в данном наборе точек. Для $n+1$ пар чисел $(x_0, y_0), (x_1, y_1) \dots (x_n, y_n)$, где все x_i различны, существует единственный многочлен $L(x)$ степени не более n , для которого $L(x_i) = y_i$.

В простейшем случае $n = 1$ это линейный многочлен, график которого — прямая, проходящая через две заданные точки.

Лагранж предложил способ вычисления таких многочленов:

$$L(x) = \sum_{j=0}^n y_j l_j(x)$$

где базисные полиномы определяются по формуле:

$$l_j(x) = \prod_{i=0, j \neq i}^n \frac{x - x_i}{x_j - x_i} = \frac{x - x_0}{x_j - x_0} \dots \frac{x - x_{j-1}}{x_j - x_{j-1}} \frac{x - x_{j+1}}{x_j - x_{j+1}} \dots \frac{x - x_n}{x_j - x_n}$$

легко видеть что $l_j(x)$ обладают такими свойствами:

- Это полиномы степени n
- $l_j(x_j) = 1$
- $l_j(x_i) = 0$ при $i \neq j$

Отсюда следует, что $L(x)$, как линейная комбинация $l_j(x)$, может иметь степень не больше n , и $L(x_j) = y_j$, для случая равномерного распределения по отрезку узлов интерполяции.

В указанном случае можно выразить x_i через расстояние между узлами интерполяции h и начальную точку x_0 :

$$x_j \equiv x_0 + jh$$

и, следовательно,

$$x_i - x_j \equiv (i - j)h$$

Подставив эти выражения в формулу базисного полинома и вынеся h за знаки перемножения в числителе и знаменателе, получим

$$l_i(x) = \prod_{j=0, i \neq j}^n \frac{(x - x_j)}{(x_i - x_j)} = \frac{\prod_{j=0, i \neq j}^n (x - x_0 - jh)}{h^{n-1} \prod_{j=0, i \neq j}^n (i - j)} = \frac{h^{n-1} \prod_{j=0, i \neq j}^n \left(\frac{x-x_0}{h} - j\right)}{h^{n-1} \prod_{j=0, i \neq j}^n (i - j)}$$

Теперь можно ввести замену переменной

$$y = \frac{x - x_0}{h}$$

и получить полином от u , который строится с использованием только целочисленной арифметики. Недостатком данного подхода является факториальная сложность числителя и знаменателя, что требует использования алгоритмов с многобайтным представлением чисел.

4.5 Сглаживание данных эксперимента

Сглаживание данных эксперимента является специальной операцией усреднения с помощью интерполяционных многочленов, обеспечивающей получение «уточненного» значения \bar{y}_i по заданному значению y_i и ряду близлежащих значений $(\dots, y_{i-1}, y_i, y_{i+1}, \dots)$, известных со случайной погрешностью. Примерами алгоритмов сглаживания являются:

1. Линейное сглаживание по трем точкам.

Линейное сглаживание по трем точкам выполняется с помощью следующих формул:

$$\bar{y}_0 = \frac{5 \cdot y_0 + 2 \cdot y_1 - y_2}{6}$$

$$\bar{y}_i = \frac{y_{i-1} + y_i + y_{i+1}}{3}; \quad 1 \leq i \leq N - 1$$

$$\bar{y}_N = \frac{5 \cdot y_N + 2 \cdot y_{N-1} - y_{N-2}}{6}$$

где N — номер последней точки (ординаты y_i).

2. Линейное сглаживание по пяти точкам.

Линейное сглаживание по пяти точкам выполняется с помощью следующих формул:

$$\bar{y}_0 = \frac{3 \cdot y_0 + 2 \cdot y_1 + y_2 - y_4}{5}$$

$$\bar{y}_1 = \frac{4 \cdot y_0 + 3 \cdot y_1 + 2 \cdot y_2 + y_3}{10}$$

$$\bar{y}_i = \frac{y_{i-2} + y_{i-1} + y_i + y_{i+1} + y_{i+2}}{5}; \quad 2 \leq i \leq N-2$$

$$\bar{y}_{N-1} = \frac{4 \cdot y_N + 3 \cdot y_{N-1} + 2 \cdot y_{N-2} + y_{N-3}}{10}$$

$$\bar{y}_N = \frac{3 \cdot y_N + 2 \cdot y_{N-1} + y_{N-2} - y_{N-4}}{5}$$

где N - номер последней точки (ординаты y_i).

3. Нелинейное сглаживание по семи точкам.

Нелинейное сглаживание по семи точкам — операция усреднения с помощью интерполяционного многочлена третьей степени. Выполняется с помощью формул:

$$\bar{y}_0 = \frac{39 \cdot y_0 + 8 \cdot y_1 - 4 \cdot (y_2 + y_3 - y_4) + y_5 - 2 \cdot y_6}{42}$$

$$\bar{y}_1 = \frac{8 \cdot y_0 + 19 \cdot y_1 + 16 \cdot y_2 + 6 \cdot y_3 - 4 \cdot y_4 - 7 \cdot y_5 + 4 \cdot y_7}{42}$$

$$\bar{y}_2 = \frac{-4 \cdot y_0 + 16 \cdot y_1 + 19 \cdot y_2 + 12 \cdot y_3 + 2 \cdot y_4 - 4 \cdot y_5 + y_6}{42}$$

$$\bar{y}_i = \frac{7 \cdot y_i + 6 \cdot (y_{i+1} + y_{i-1}) + 3 \cdot (y_{i+2} + y_{i-2}) - 2 \cdot (y_{i+3} + y_{i-3})}{21};$$

$$3 \leq i \leq N-3$$

$$\bar{y}_{N-2} = \frac{-4 \cdot y_N + 16 \cdot y_{N-1} + 19 \cdot y_{N-2} + 12 \cdot y_{N-3} + 2 \cdot y_{N-4} - 4 \cdot y_{N-5} + y_{N-6}}{42}$$

$$\bar{y}_{N-1} = \frac{8 \cdot y_N + 19 \cdot y_{N-1} + 16 \cdot y_{N-2} + 6 \cdot y_{N-3} - 4 \cdot y_{N-4} - 7 \cdot y_{N-5} + 4 \cdot y_{N-6}}{42}$$

$$\bar{y}_N = \frac{39 \cdot y_N + 8 \cdot y_{N-1} - 4 \cdot y_{N-2} - 4 \cdot y_{N-3} + y_{N-4} + 4 \cdot y_{N-5} - 2 \cdot y_{N-6}}{42}$$

где N - номер последней точки (ординаты y_i).

Следует заметить, что несмотря на название, метод линеен относительно входных данных (т.е. при увеличении всех величин входных данных в два раза результат тоже вырастет в два раза). /16/

4.6 Аппроксимация

Аппроксимация, или приближение — математический метод, состоящий в замене одних математических объектов другими, в том или ином смысле близкими к исходным, но более простыми. Аппроксимация позволяет исследовать числовые характеристики и качественные свойства объекта, сводя задачу к изучению более простых или более удобных объектов (например, таких, характеристики которых легко вычисляются или свойства которых уже известны). В теории чисел изучаются диофантовы приближения, в частности приближения иррациональных чисел рациональными. В геометрии рассматриваются аппроксимации кривых ломанными. Некоторые разделы математики в сущности целиком посвящены аппроксимации, например, теория приближения функций, численные методы анализа.

Из курса математики известны 3 способа задания функциональных зависимостей:

- 1) аналитический
- 2) графический
- 3) табличный

Табличный способ обычно возникает в результате эксперимента.

Недостаток табличного задания функции заключается в том, что найдутся значения переменных которые неопределены таблицей. Для отыскания таких значений определяют приближающуюся к заданной функцию, называемой аппроксимирующей, а действие замены аппроксимацией.

Аппроксимация заключается в том, что используя имеющуюся информацию по $f(x)$ можно рассмотреть другую функцию $\varphi(x)$ близкую в некотором смысле к $f(x)$, позволяющую выполнить над ней соответствующие операции и получить оценку погрешность такой замены (см. рис.9):

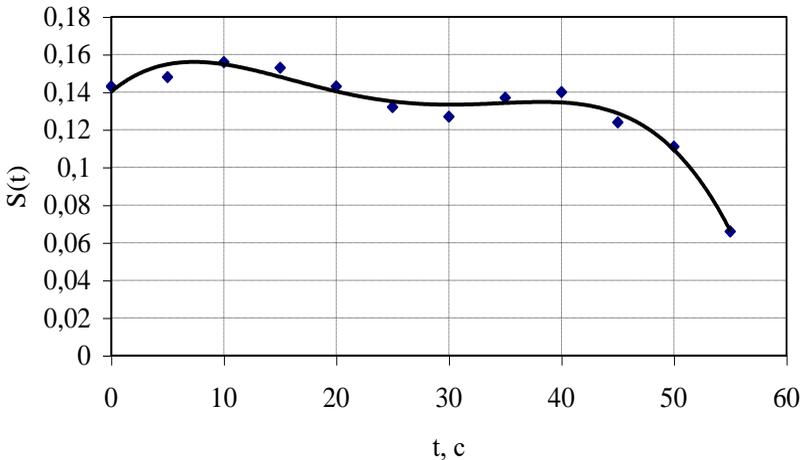


Рис. 9. Пример аппроксимации. Точки – данные эксперимента, $\varphi(x)$ - аппроксимирующая функция /17/.

4.7 Сплайн

Под сплайном (от англ. spline - планка, рейка) обычно понимают агрегатную функцию, совпадающую с функциями более простой природы на каждом элементе разбиения своей области определения.

Классический сплайн одной переменной строится так: область определения разбивается на конечное число отрезков, на каждом из которых сплайн совпадает с некоторым алгебраическим полиномом. Максимальная степень из использованных полиномов называется степенью сплайна. Разность между степенью сплайна и получившейся гладкостью называется дефектом сплайна. Например, непрерывная ломаная есть сплайн степени 1 и дефекта 1.

Сплайны имеют многочисленные применения как в математической теории, так и в разнообразных вычислительных приложениях. В частности, сплайны двух переменных интенсивно используются для задания поверхностей в различных системах компьютерного моделирования.

Функции, подобные тем, что сейчас называют сплайнами, были известны математикам давно, начиная как минимум с Эйлера, но их интенсивное изучение началось, фактически, только в середине XX века. В 1946 году Исаак Шёнберг (Isaac Jacob Schoenberg) впервые употребил этот термин в качестве обозначения класса полиномиальных сплайнов. До 1960 годов сплайны были в основном инструментом теоретических исследований, они часто появлялись в качестве решений различных экстремальных и вариационных задач, особенно в теории приближений.

После 1960 года с развитием вычислительной техники началось использование сплайнов в компьютерной графике и моделировании, что продолжается по сей день (например, Кривые Безье или NURBS).

4.8 Интерполяция сплайнами

Интерполяция сплайнами третьего порядка - это быстрый, эффективный и устойчивый способ интерполяции функций. Наравне с рациональной интерполяцией, сплайн-интерполяция является одной из альтернатив полиномиальной интерполяции.

В основе сплайн-интерполяции лежит следующий принцип. Интервал интерполяции разбивается на небольшие отрезки, на каждом из которых функция задается полиномом третьей степени. Коэффициенты полинома подбираются таким образом, чтобы выполнялись определенные условия (какие именно, зависит от способа интерполяции). Общие для всех типов сплайнов третьего порядка требования - непрерывность функции и, разумеется, прохождение через предписанные ей точки. Дополнительными требованиями могут быть линейность функции между узлами, непрерывность высших производных и т.д.

Основными достоинствами сплайн-интерполяции являются её устойчивость и малая трудоемкость. Системы линейных уравнений, которые требуется решать для построения сплайнов, очень хорошо обусловлены, что позволяет получать коэффициенты полиномов с высокой точностью. В результате даже про очень больших N вычислительная схема не теряет устойчивость. Построение таблицы коэффициентов сплайна требует $O(N)$ операций, а вычисление значения сплайна в заданной точке - всего лишь $O(\log(N))$.

4.9 Линейный сплайн

Линейный сплайн - это сплайн, составленный из полиномов первой степени, т.е. из отрезков прямых линий.

Точность интерполяции линейными сплайнами невысока, также следует отметить, что они не обеспечивают непрерывности даже первых производных. Однако в некоторых случаях кусочно-линейная аппроксимация функции может оказаться предпочтительнее, чем аппроксимация более высокого порядка. Например, линейный сплайн сохраняет монотонность переданного в него набора точек.

На графике рис. 10 приведен пример линейного сплайна, интерполирующего функцию $f = \cos(0.5 \cdot \pi \cdot x)$ на отрезке $[-1, 1]$.

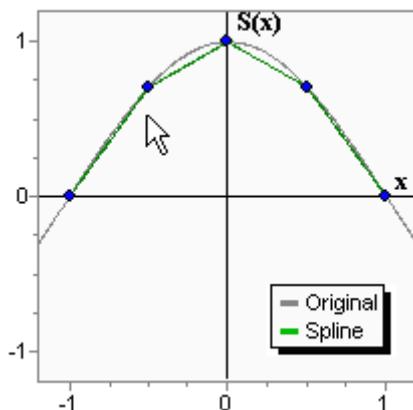


Рис. 10. Пример интерполяции линейным сплайном.

4.10 Сплайн Эрмита

Сплайн Эрмита - это сплайн третьего порядка, производная которого принимает в узлах сплайна заданные значения. В каждом узле сплайна Эрмита задано не только значение функции, но и значение её первой производной. Сплайн Эрмита имеет непрерывную первую производную, но вторая производная у него разрывна.

На графике рис. 11 приведен пример сплайна Эрмита, интерполирующего функцию $f = \cos(0.5 \cdot \pi \cdot x)$ на отрезке $[-1, 1]$. Можно видеть, что точность интерполяции значительно лучше, чем у линейного сплайна.

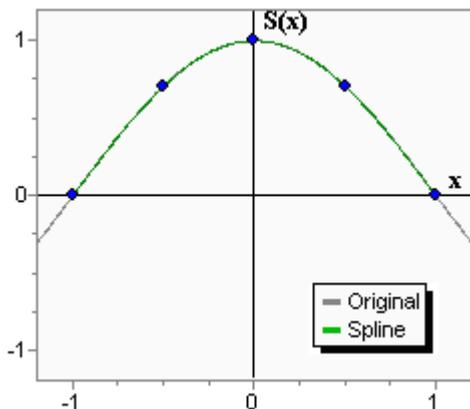


Рис. 11. Пример интерполяции кубическим сплайном (сплайном Эрмита).

4.11 Кубический сплайн

Все сплайны, рассмотренные ниже, являются кубическими сплайнами - в том смысле, что они являются кусочно-кубическими функциями. Однако, когда говорят "кубический сплайн", то обычно имеют в виду конкретный вид кубического сплайна, который получается, если потребовать непрерывности первой и второй производных. Кубический сплайн задается значениями функции в узлах и значениями производных на границе отрезка интерполяции (либо первых, либо вторых производных).

Если известно точное значение первой производной на обеих границах, то такой сплайн называют

фундаментальным. Погрешность интерполяции таким сплайном равна $O(h^4)$.

Если значение первой (или второй) производной на границе неизвестно, то можно задать т.н. естественные граничные условия $S''(A) = 0$, $S''(B) = 0$, и получить естественный сплайн. Погрешность интерполяции естественным сплайном составляет $O(h^2)$. Максимум погрешности наблюдается в окрестностях граничных узлов, во внутренних узлах точность интерполяции значительно выше.

Ещё одним видом граничного условия, которое можно использовать, если неизвестны граничные производные функции, является условие типа "сплайн, завершающийся параболой". В этом случае граничный отрезок сплайна представляется полиномом второй степени вместо третьей (для внутренних отрезков по-прежнему используются полиномы третьей степени). В ряде случаев это обеспечивает большую точность, чем естественные граничные условия.

Наконец, можно сочетать различные типы граничных условий на разных границах. Обычно так имеет смысл делать, если у нас есть только часть информации о поведении функции на границе (например, производная на левой границе - и никакой информации о производной на правой границе).

На графике рис. 12 приведен пример кубического сплайна, интерполирующего функцию $f = \cos(0.5 \cdot \pi \cdot x)$ на отрезке $[-1, 1]$. Можно видеть, что точность интерполяции близка к точности Эрмита сплайна.

4.12 Сплайн Акимы

Сплайн Акимы - это особый вид сплайна, устойчивый к выбросам. Недостатком кубических сплайнов является то, что они склонны осциллировать в окрестностях точки,

существенно отличающейся от своих соседей.

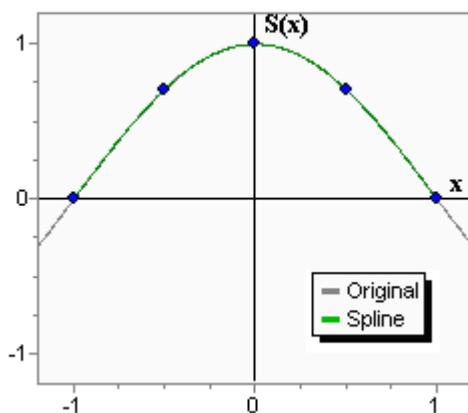


Рис. 12. Пример интерполяции кубическим сплайном.

На графике рис. 13 приведен набор точек, содержащий один выброс.

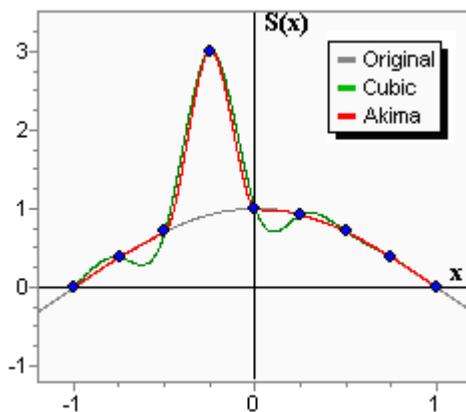


Рис. 13. Пример интерполяции сплайном Акимы.

Зеленым цветом обозначен кубический сплайн с естественными граничными условиями. На отрезках интерполяции, граничащих с выбросом, сплайн заметно отклоняется от интерполируемой функции - сказывается влияние выброса. Красным цветом обозначен сплайн Акимы. Можно видеть, что, в отличие от кубического сплайна, сплайн Акимы в меньшей мере подвержен влиянию выбросов - на отрезках, граничащих с выбросом, практически отсутствуют признаки осцилляции.

Важным свойством сплайна Акимы является его локальность - значения функции на отрезке $[x_i, x_{i+1}]$ зависят только от $f_{i-2}, f_{i-1}, f_i, f_{i+1}, f_{i+2}, f_{i+3}$. Вторым свойством, которое следует принимать во внимание, является нелинейность интерполяции сплайнами Акимы - результат интерполяции суммы двух функций не равен сумме интерполяционных схем, построенных на основе отдельных функций. Для построения сплайна Акимы требуется не менее 5 точек. Во внутренней области (т.е. между x_2 и x_{N-3} при нумерации точек от 0 до $N-1$) погрешность интерполяции имеет порядок $O(h^2)$.

4.13 В-сплайн

Более сложный тип интерполяции — так называемая интерполяция В-сплайнами. В отличие от обычной сплайн-интерполяции, сшивка элементарных В-сплайнов производится не в известных точках x_i а в других точках u_i , координаты которых предлагается ввести пользователю. Поэтому В-сплайн редко проходит через управляющие вершины (рис. 14).

Сплайны могут быть полиномами 1, 2 или 3 степени (линейные, квадратичные или кубические). Применяется интерполяция В-сплайнами точно так же, как и обычная сплайн-интерполяция, различие состоит только в

определении вспомогательной функции коэффициентов сплайна. /18/

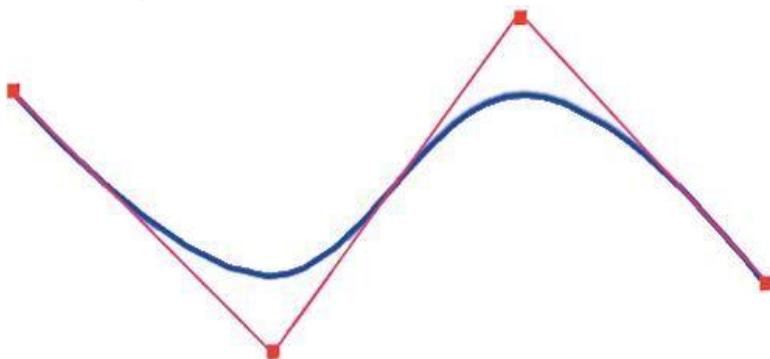


Рис. 14. Пример интерполяции В-сплайном.

4.14 Оцифровка графических данных. Программное обеспечение

Как уже не раз отмечалось, очень часто в справочной и научно-технической литературе функциональные зависимости даются не формулами, а *графиками*. Причем сами функциональные зависимости, по которым строились графики, не приводятся. Нет и таблиц, по которым построены графики. На рис. 15 в качестве примера представлен один из таких графиков, отражающий влияние скорости воды и ее температуры на удельное гидравлическое сопротивление в некоем фильтрующем материале, через который воду прокачивают.

Рисунок взят из справочной документации одной известной фирмы, поставляющей водоочистное оборудование.

Подобные графики приводятся не только для качественного описания тех или иных явлений (гидравлическое сопротивление растет при увеличении

скорости и падает при повышении температуры, если говорить о рис. 15), но и для их количественной оценки – для расчетов.

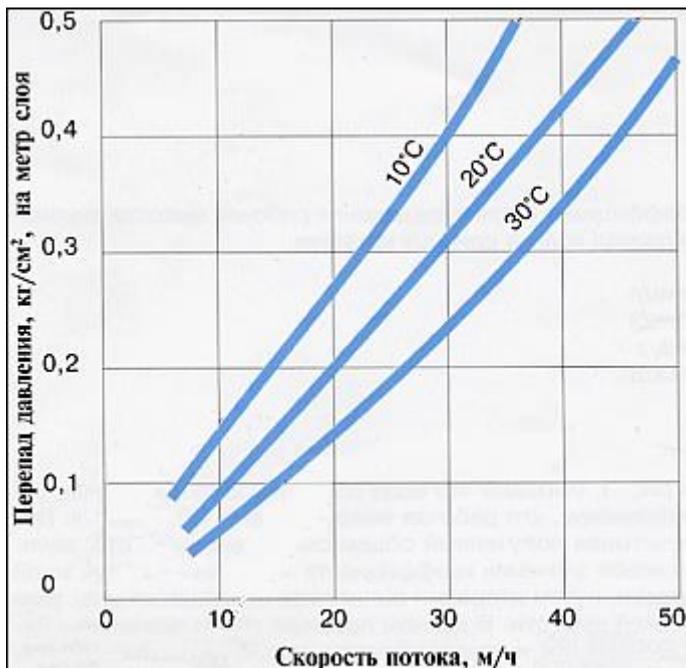


Рис. 15. Пример графика, отражающего влияние скорости воды и ее температуры на удельное гидравлическое сопротивление в некоем фильтрующем материале, через который прокачивают воду.

В упомянутой фирменной документации описан расчет этого гидравлического сопротивления по методике «вождения пальцем по графику»: отложите по оси абсцисс значение скорости (первый аргумент), поднимитесь до нужной температуры или мысленно проведите недостающую кривую (изотерму – второй аргумент), поверните налево и

считайте ответ (значение функции двух аргументов) на оси ординат.

В справочниках встречаются также и более сложные графические зависимости – разного рода *номограммы* с инструкциями такого рода: отложите значение первого аргумента на первой шкале, а второго – на второй; соедините точки прямой линией и на средней шкале считайте ответ. На рис. 16 представлен образец такой номограммы, моделирующей функцию уже трех аргументов.

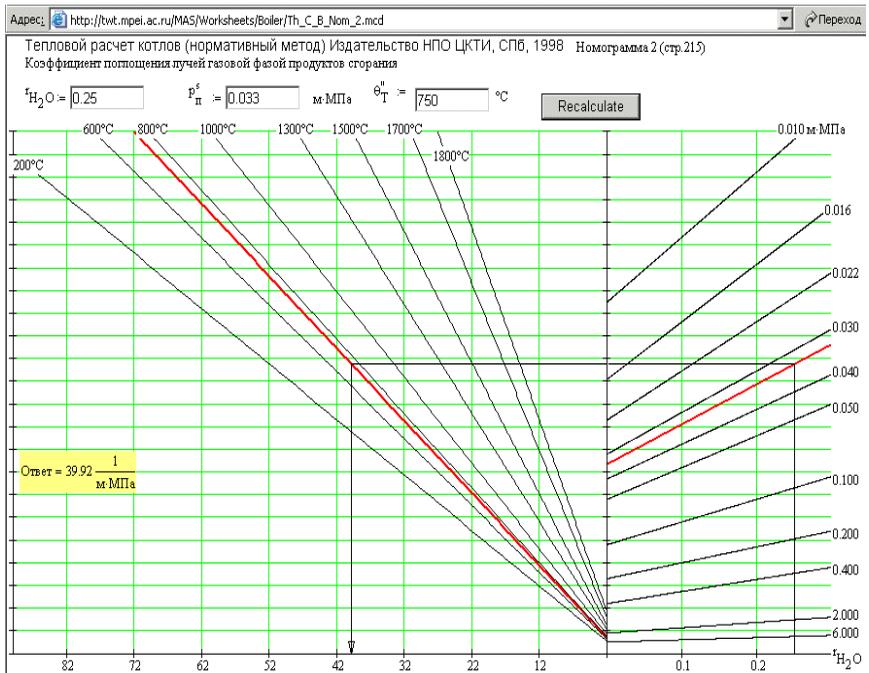


Рис. 16. Образец такой номограммы, моделирующей функцию трех аргументов.

В номограммах (а на них выросло целое поколение инженеров) «тонет» физика задачи – ее качественная оценка,

зато повышается точность таких «графических» расчетов. Выпускались даже нехитрые механические устройства типа логарифмической линейки с вшитыми в них алгоритмами. Такие устройства были особо популярны у штурманов, прокладывающих маршруты морских и воздушных судов до наступления эры бортовых компьютеров и глобальных систем позиционирования. Сейчас что-то подобное можно купить в газетных киосках – совместил на двух дисках свой вес и рост и узнал, пора ли переходить на диету...

Но для современных расчетов с использованием компьютеров или просто калькуляторов более подходят не графики и номограммы, а формулы, которые часто не приводятся в технической литературе по ряду причин.

Во-первых, формулы не даются из благих намерений освободить читателя от расчетов. Тем более это зачастую и не расчет в привычном понимании этого слова, а некая оценка, прикидка того или иного параметра. В той же документации, откуда взят рис. 15, рекомендовано при выборе насоса для фильтра (а его напор – это произведение удельного гидравлического сопротивления на высоту фильтрующего материала) увеличить расчетное гидравлическое сопротивление на 10–20% (так называемый инженерный запас, нивелирующий помимо прочего и ошибки считывания «пальцем» чисел с графика). «Расчетные» графики – это атавизм тех времен, когда даже простейший расчет был проблемой для инженера. Теперь же рабочий стол инженера оборудован ну если не компьютером, то как минимум программируемым калькулятором и проблем с расчетами не должно быть. Но «расчетные» графики без соответствующей формулы продолжают появляться даже во вновь издаваемых справочниках. Иногда в справочнике можно видеть некий гибрид графика и формулы. На рис. 17 показан пример такого расчета, когда через график определяется зависимость функции (удельное гидравлическое сопротивление некоего

фильтрующего материала) от первого аргумента (от скорости потока через фильтр), а по эмпирической формуле – от второго аргумента – от температуры.

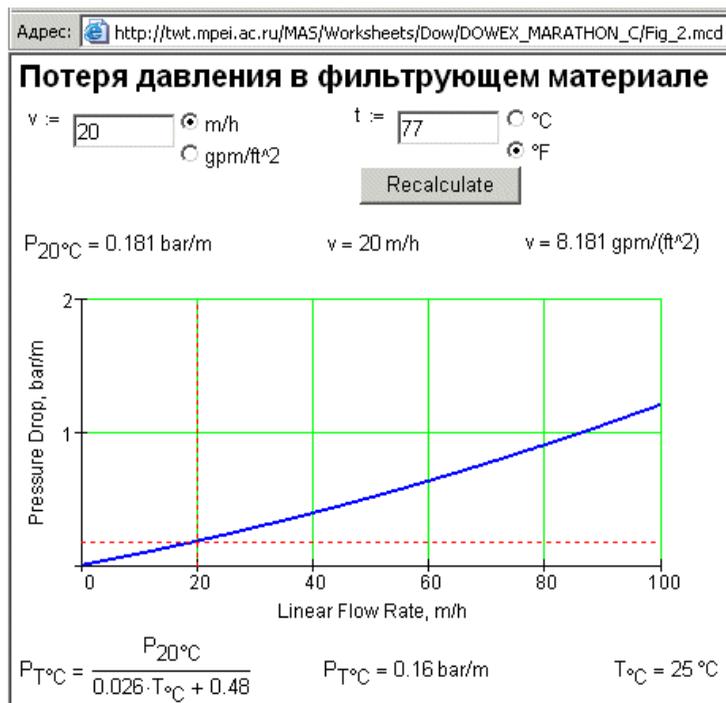


Рис. 17 Пример «расчетного графика»: через график определяется зависимость функции (удельное гидравлическое сопротивление некоего фильтрующего материала) от первого аргумента (от скорости потока через фильтр), а затем по эмпирической формуле – от второго аргумента – от температуры.

Во-вторых, нередко никакой формулы не было, и нет, так как на графиках даны результаты некой графической обработки опытных точек. Кривые, показанные на рис. 15,

получены после испытания фильтрующего материала на специальном стенде, где есть возможность менять скорость потока, а также температуру воды и замерять перепады давления. Полученные экспериментальные точки были проставлены на графике, вблизи них неким лекалом или статистической обработкой были проведены линии, а затем сами точки стерты. В научной же (не технической) литературе считается хорошим тоном оставлять на графике экспериментальные точки и показывать различного рода доверительные интервалы. В последнее время получает распространение практика ссылок из научных статей на сайт, где хранятся первичные протоколы опытов, по которым читатель (оппонент) может не только проверить выводы автора, но и дать свою трактовку результатов. Можно идти дальше и делать ссылки на программу с расчетом по этому графику. На бумаге (в справочной документации – см. рис. 15) видна качественная картина явления, а на сайте, поддерживающего эту документацию, прописан соответствующий расчет. Для этого можно:

- попытаться связаться с автором и попросить его дать формулу, если она, конечно, была и по ней был построен сам график;
- вывести самому нужную формулу, опираясь на «физику» задачи; в нашем случае, например, сопротивление фильтрующего материала может зависеть от скорости фильтруемой жидкости (воды) в степени близкой к квадратной, а сами кривые выходят из начала координат: $y=k \cdot x^2$ – здесь достаточно установить значение коэффициента k ;
- подойти к задаче несколько формально (феноменологически) – провести интерполяцию или аппроксимацию по точкам с опорой на стандартные функции – полином некой степени, например.

В данной главе в первую очередь будет рассказано о

том, как график из справочника можно «оцифровать» – получить массив данных (векторы и матрицы).

График, конечно, можно «оцифровать» докомпьютерными (безкомпьютерными) средствами – взять в руки линейку и калькулятор и вручную составить таблицу значений точек на графике. Но современные аппаратные и программные средства компьютеров позволяют автоматизировать эту работу.

4.15 Оцифровка графиков средствами MathCAD

Первый шаг на пути оцифровки «бумажного» графика – это его сканирование, получение его электронного растрового изображения – файла. Такой файл (изображение) требует минимальной последующей доработки – поворота при необходимости изображения с помощью какого-нибудь графического редактора так, чтобы ось ординат была строго горизонтальна, а ось абсцисс, соответственно, вертикальна. Но часто и этого не требуется, если страница справочника с графиком была установлена в сканере аккуратно без перекосов. Для этого иногда приходится, увы, разрывать книги на отдельные страницы и ровно обрезать их края. С другой стороны, если не книгу (страницу книги) несут к сканеру, а сам сканер (ручной сканер) несут к книге и проводят им по раскрытой странице, то может потребоваться дальнейшая обработка – разворот графика и т.д. На рис. 18 показан график, отсканированный из справочника «Физические величины» (М.: Энергоатомиздат, 1991) и показывающий влияние температуры и солености морской воды на скорость звука в ней.

Далее будет рассказано, как такой и другие подобные графики можно «оцифровать», будут раскрыты некоторые приемы такой работы в среде MathCAD.

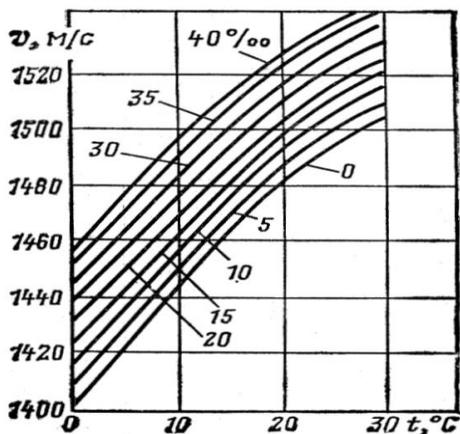


Рис. 18. Пример графика, отсканированного из книги.

На рис. 19 показано, как в MathCAD-документ командой `Insert Picture` из панели инструментов `Matrix` вставлен рисунок одной кривой графика, показанного на рис. 18.

Для этого график предварительно был отредактирован компьютерным ластиком – из него было убрано все лишнее, а сама линия была сохранена на диске файлом с именем `CC_Plot.jpg`. Оператор `READBMP` (см. рис. 18) формирует матрицу M , хранящую цвета точек (оттенки серого) растрового изображения в кодировке от 0 (черный цвет) до 255 (белый). Здесь становится понятен смысл слова «растровое изображение» – некое мозаичное изображение, формируемое отдельными точками (квадратиками). На рис. 18 показан фрагмент этой матрицы («мозаики») в диапазоне столбцов от нулевого до шестого и строк от 302-й до 312-й (левый нижний угол матрицы M , левый край кривой). Далее в программе перебором столбцов (цикл с параметром i) и строк (параметр j) ведется заполнение векторов X и Y координатами точек с черным цветом (код со значением 0, $M_{i,j} = 0$).

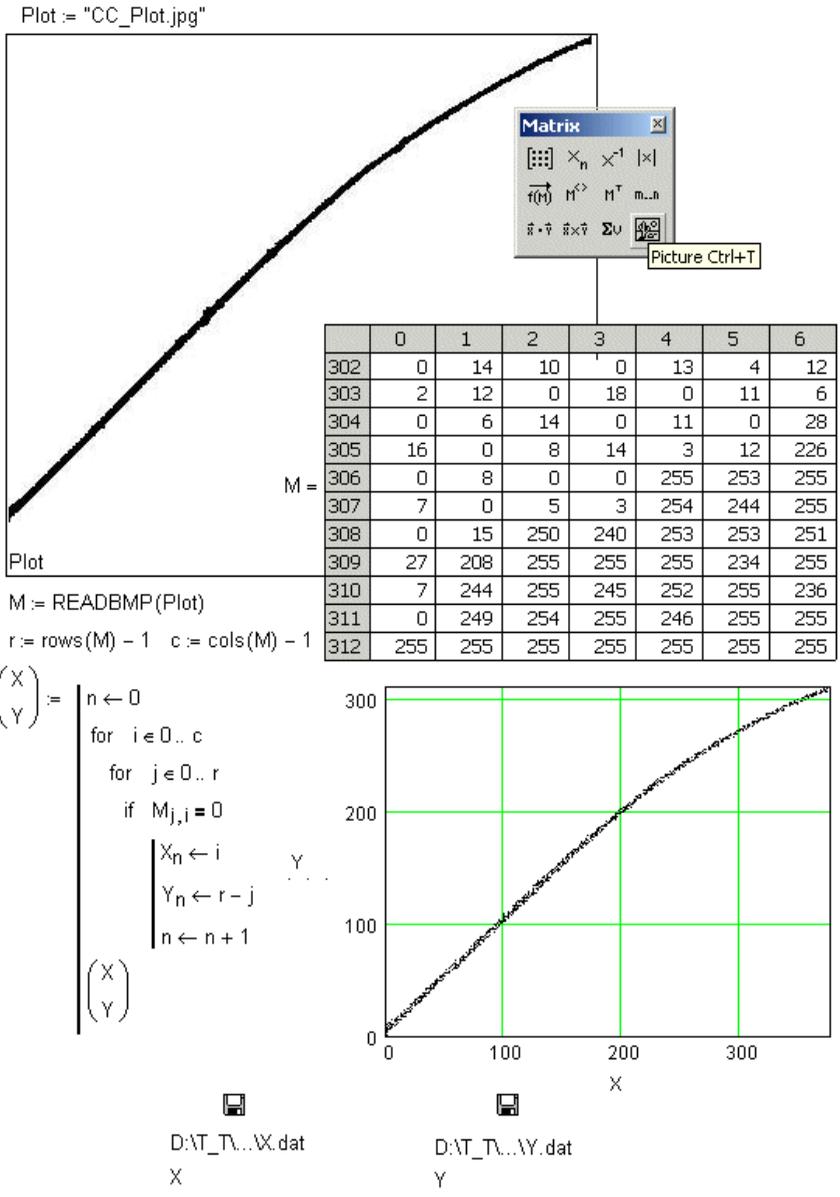


Рис. 19. Пример оцифровки графика средствами MathCAD.

Заодно точки «переворачиваются» оператором $y_n \leftarrow r-j$: у графика начало координат – это нижний левый угол, а у матрицы – верхний левый. После этого точки отображаются на X-Y-графике, а из координаты записываются на диск файлами с именами X.dat и Y.dat.

На рис. 20 показана статистическая обработка векторов X и Y, значения которых считываются с диска, а затем подвергаются статистической обработке.

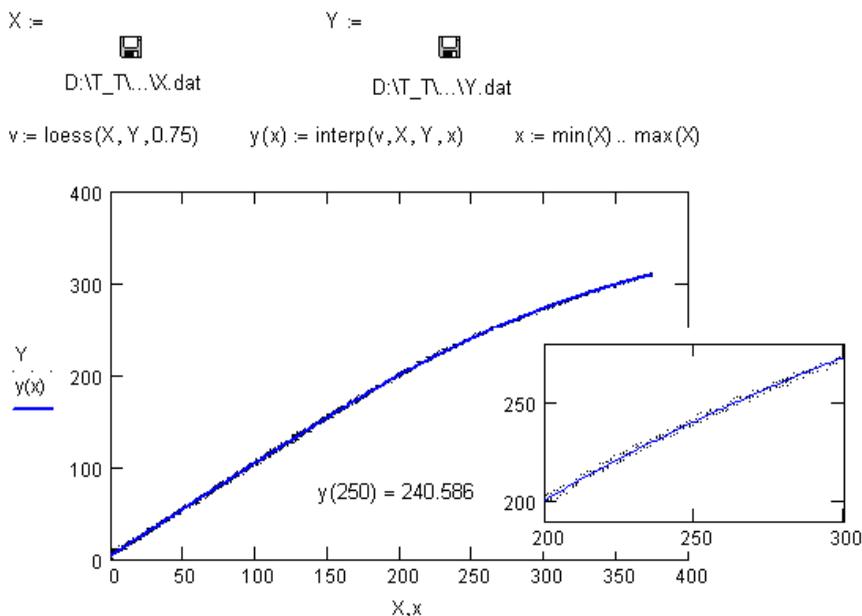


Рис. 20. Дальнейшая обработка оцифрованного графика.

Далее будут задействованы некоторые специальные (статистические) встроенные функции Mathcad. Их описание можно найти в Интернет в виде гл. 16 книги Гурского Д.А. и Турбиной Е.С. Гурского и Турбиной «Вычисления в Mathcad 12» (Издательство Питер, 2005 г.). Скачать данную главу можно с сайта издательства «Питер» /19/ или с Интернета

/20/.

Проведенная статистическая обработка позволила получить функцию $y(x)$, по которой можно считать ($y(250) = 240.586$, например) и строить график – см. на рис. 20 увеличенный фрагмент кривой, «пронизывающей» точки.

Другой подход к оцифровке графика показан на рис. 21.

Графики в настоящее время все чаще и чаще можно найти не только на «бумаге» справочников, но и в электронном виде в файлах, хранящихся и в Интернет. Так, на рис. 20 исходный график некой зависимости взят из статьи, открытой с помощью Adobe Reader (программы, свободно распространяемой в Интернет и служащей для открытия так называемых pdf-документов – документов, созданных в среде текстового процессора Adobe Acrobat – см. www.adobe.com). Этот график можно перенести в среду, например, графического редактора Paint, входящего в состав операционной системы Windows, там убрать лишнее, отметить цветными крапинками узловые точки на кривой (на рис. 21 – это одиннадцать точек пересечения кривой с вертикальной сеткой) и открыть этот графический файл в среде MathCAD. На рис. 21 в матрице M ищутся уже не черные точки с кодом 0 как на рис. 18, а точки с кодом 76, фиксирующим цвет, которым были отмечены узловые точки графика.

4.16 Оцифровка графиков другими средствами

С графическими файлами, хранящими отсканированные (рис. 18) или скопированные из электронных документов (рис. 21 вверху) графики, можно, а в ряде случаев и целесообразно поступить иначе, чем показано на рис. 19 и рис. 21. Такой файл можно открыть в среде какого-либо пакета, специально созданного для оцифровки графиков.

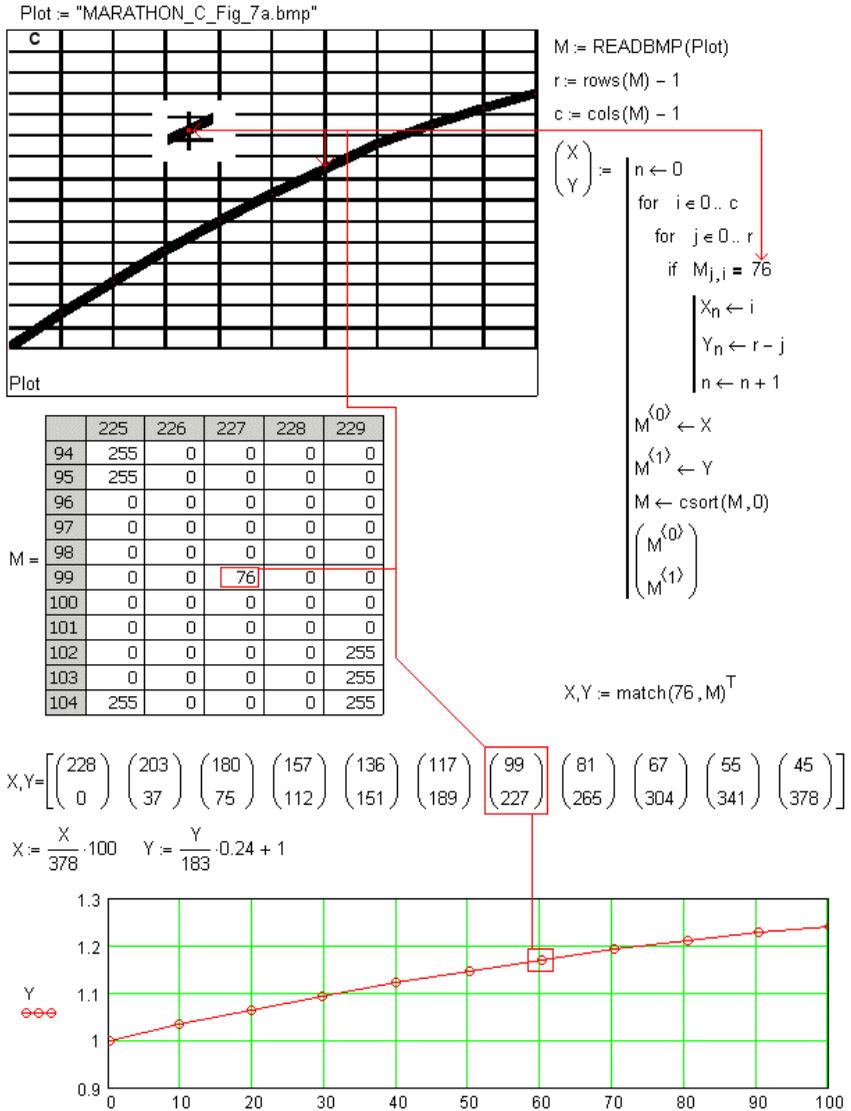


Рис. 21. Другой подход к оцифровке графиков в MathCAD.

На рис. 22 показана работа одного из таких пакетов, написанного В. Писковым и который можно скачать из Сети по адресу <http://twf.mpei.ac.ru/ТТНВ/5/3/AIGraph.exe>.

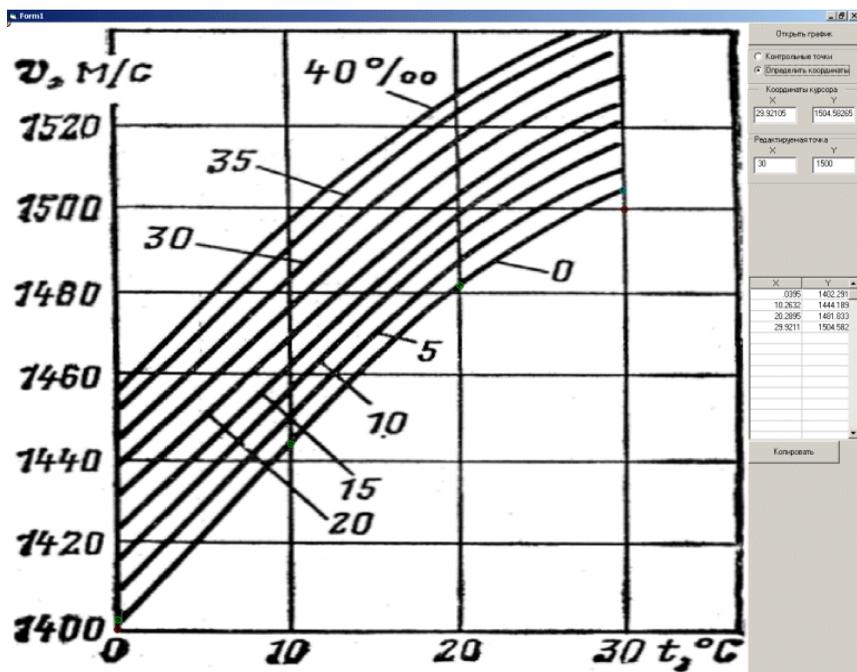


Рис. 22. Оцифровка графика при помощи программы AIGraph.

На рис. 22 зафиксирована работа с программой с данной программой. Если подвести курсор мыши к нужной точке графика и щелкнуть по левой кнопки мыши, то в полях X и Y (см. правый край рис. 21) появятся координаты данной точки (режим «Определить координаты»). Перед такой работой через режим «Контрольные точки» (см. правый верхний угол на рис.21) необходимо оттарировать график – отметить две

любые точки на графике (к качеству таких точек обычно выбирают находящиеся на противоположных концах диагонали графика) и указать их координаты. В нашем случае это будут $X=0$ $Y=1400$ и $X=30$ $Y=1540$ – две отдаленные друг от друга точки пересечения линии сетки графика. После этого при работе в режиме «Определить координаты» в полях X и Y будут фиксироваться истинные координаты точек на графике, которые затем можно перенести в вектора и матрицы Mathcad-документа. На рис. 19 (см. выше) векторы X и Y хранили не координаты точек на графике, а значение индексов матрицы M . На другом рисунке выше, на рис. 21, эти значения были подправлены через операторы $X = 100 X/378$ и $Y = 0.24 \cdot Y/183 + 1$. В программу, отображенную на рис. 22, заложена также технология, когда пользователь проводит курсором мыши по выбранной кривой графика, нажав при этом на левую кнопку мыши. В этом случае в полях X и Y автоматически генерируются соответствующие вектора значений координат точек графика с заданным заранее шагом. Более подробная инструкция по работе с программой, отображенной на рис. 22, находится в пакете.

Координаты точек рисунка без расчета истинных значений координат точек графика можно получить, работая с графическим редактором Paint (см. рис. 23), входящим в состав Windows.

Этот способ не самый удобный, но и он может пригодиться, если под рукой нет ничего более подходящего.

/21/

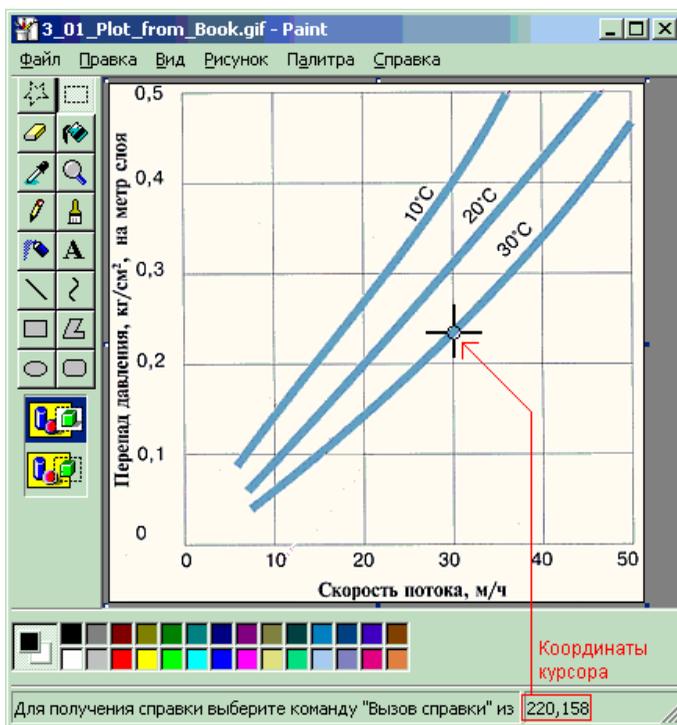


Рис. 23. Координаты точек рисунка без расчета истинных значений координат точек графика полученные при помощи графического редактора Paint, входящего в состав Windows.

4.17 Процесс оформления научных работ и используемые программные средства. Редактор TeX

TeX является мировым стандартом подготовки научной документации. В нем и только в нем принимаются статьи в научные журналы. Только в TeX должна быть выполнена диссертация на соискание той или иной степени по физике, математике, химии. А кое-где, например, в Финляндии, требуют, чтобы любая научная работа, даже по

гуманитарным наукам, была представлена в аналог нашей ВАК в формате TeX. Иначе ее просто не утвердят.

TeX ориентирован на создание печатных публикаций, причем объемных и со сложной структурой. Поэтому сам язык поддерживает все то, что требуется от мощного текстового процессора или издательской системы: стилевую разметку, перекрестные ссылки, сноски, многоколоночную верстку, колонтитулы, вставку графики и таблиц. Плюс средства описания формул. Так что количество управляющих тегов весьма и весьма значительно. Полный их перечень занимает около 50 страниц.

Что такое TEX и LATEX. TEX (произносится «тех», пишется также «TeX») — это созданная замечательным американским математиком и программистом Дональдом Кнудом (Donald E. Knuth) система для верстки текстов с формулами. Сам по себе TEX представляет собой специализированный язык программирования (Кнут не только придумал язык, но и написал для него транслятор, причем таким образом, что он работает совершенно одинаково на самых разных компьютерах), на котором пишутся издательские системы, используемые на практике. Точнее говоря, каждая издательская система на базе TEXа представляет собой пакет макроопределений (макропакет) этого языка. LATEX (произносится «латех» или «лэйтех», пишется также «LaTeX») — это созданная Лесли Лампортом (Leslie Lamport) издательская система на базе TEXа.

Прежде, чем углубиться в изучение собственно LATEXа, скажем несколько слов о других издательских системах на базе TEXа. Наряду с LATEXом, распространены также макропакеты Plain TEX и AMS-TEX. Макропакет Plain TEX был разработан самим Дональдом Кнудом, рассматривавшим его в качестве платформы для построения более сложных систем; на практике он используется и как средство для обмена текстами (текст, подготовленный для

Plain TEXа, сравнительно несложно переделать в исходный текст для того же LATEXа). Что касается AMS-TEXа, то эта издательская система сориентирована на важный, но узкий круг приложений: верстку статей для математических журналов, издаваемых Американским Математическим Обществом. Соответственно, в AMS-TEXе предусмотрено большое количество весьма изощренных возможностей для создания сложных математических формул, но при этом нет многих вещей, которые естественно было бы ожидать в издательских системах общего назначения (например, автоматической нумерации частей документа). LATEX в этом отношении более гармоничен. Наконец, недавно появился AMS-LATEX, призванный сочетать мощь LATEXа как издательской системы и изощренные возможности набора формул, предоставляемые AMS-TEXом./21/

4.17.1 Как проходит работа с системой TEX

В дальнейшем мы будем отмечать, какие свойства системы специфичны для TEXа, а какие относятся вообще к LATEXу и ко всем издательским системам на его базе, но при первом чтении вы можете об этих тонкостях не задумываться и воспринимать слова TEX и LATEX как синонимы.

Все, что сказано в этом разделе, применимо не только к TEXу, но и к любому другому макропакету для TEXа, хотя мы для краткости будем говорить только про LATEX.

Для начала автор должен подготовить файл с текстом, оснащенный командами для LATEXа, который по традиции имеет расширение `.tex`. Дальнейшая работа протекает в два этапа. Сначала надо обработать файл с помощью программы-транслятора; в результате получается файл с расширением `.dvi` (`device-independent` — не зависящий от устройства).

Теперь полученный файл (его называют еще `dvi-`

файлом) можно, с помощью программ, называемых dvi-драйверами, распечатать на лазерном или точечно-матричном принтере, посмотреть на экране (текст будет в таком же виде, как он появится на печати) и т. д. (для разных устройств есть разные драйверы). Неудовлетворенный результатом, автор вносит изменения в исходный файл — и цикл повторяется.

На самом деле повторений цикла будет больше, так как при синтаксических ошибках в исходном тексте транслятор будет выдавать сообщения об ошибках, которые приходится исправлять.

Кроме того, для создания исходного текста нужно, естественно, уметь обращаться с каким-нибудь текстовым редактором. /23/

4.17.2 Основные понятия работы с LATEX

Исходный файл. Исходный файл для системы LATEX представляет собой собственно текст документа вместе со спецсимволами и командами, с помощью которых системе передаются указания касательно размещения текста. Этот файл можно создать любым текстовым редактором, но при этом необходимо, чтобы в итоге получился так называемый «чистый» текстовый файл (ASCII-файл). Это означает, что текст не должен содержать шрифтовых выделений, разбивки на страницы и т. п.

Исходный текст документа не должен содержать переносов (TEX сделает их сам). Слова отделяются друг от друга пробелами, при этом TEX не различает, сколько именно пробелов вы оставили между словами (чтобы вручную управлять пробелами между словами, есть специальные команды, о которых пойдет речь позже). Конец строки также воспринимается как пробел. Отдельные абзацы должны быть отделены друг от друга пустыми строками

(опять-таки все равно, сколько именно пустых строк стоит между абзацами, важно, чтоб была хоть одна).

В правой колонке приведен фрагмент исходного текста, а в левой — то, как он будет выглядеть на печати после обработки системой LATEX.

Слова разделяются пробелами,
а абзацы — пустыми строками.
Абзацный отступ в исходном
тексте оставлять не надо:
он получается
автоматически.

Слова разделяются пробелами,
а абзацы ---
пустыми строками.
Абзацный отступ в
исходном тексте
оставлять не
надо: он получается
автоматически.

Как можно заметить из этого примера, тратить время на форматирование исходного текста тоже незачем.

Спецсимволы. Большинство символов в исходном тексте прямо обозначает то, что будет напечатано (если в исходном тексте стоит запятая, то и на печати выйдет запятая). Следующие 10 символов:

{ } \$ & # % _ ^ ~ \

имеют особый статус. Если вы употребите их в тексте «просто так», то скорее всего получите сообщение об ошибке (и на печати не увидите того, что хотелось). Печатное изображение знаков, соответствующих первым семи из них, можно получить, если в исходном тексте поставить перед соответствующим символом без пробела знак \ (по-английски он называется «backslash»):

Курс тугрика повысился на 7%,
и теперь за него дают \$200.

Курс тугрика повысился на
7\%, и теперь за него
дают \\$200.

Если символ % употреблен в тексте не в составе

комбинации `\%`, то он является «символом комментария»: все символы, расположенные на строке после него, `TEX` игнорирует (и его самого тоже). С помощью символа `%` в исходный текст можно вносить пометки «для себя»:

Это пример.
Жил-был у бабушки серенький
козлик.

Это `%` глупый
`%` Лучше: поучительный ←
пример.
Жил-был у бабушки
сере`%` нький козлик.

Обратите внимание на последний пример: после знака процента игнорируется вся строка, включая ее конец, который в нормальных условиях играет роль пробела, с другой стороны, начальные пробелы в строке игнорируются всегда. Поэтому `TEX` не видит пробела между кусками слов `сере` и `нький`, и они благополучно складываются в слово «серенький».

Скажем вкратце о смысле остальных спецсимволов. Фигурные скобки ограничивают группы в исходном файле (их рассмотрим далее). Знак доллара ограничивает математические формулы. При наборе математических же формул используются знаки `_` и `^` («знак подчеркивания» и «крышка»). Знак `~` обозначает «неразрывный пробел» между словами. Со знака `\` начинаются все `TEX`овские команды. Знаки `#` и `&` используются в более сложных конструкциях `TEX`а, наконец, символы

`<` `>` `|`

в тексте употреблять можно в том смысле, что сообщения об ошибке это не вызовет, но на печатается при этом нечто, совсем на эти символы не похожее. Подлинное место для этих символов, так же как и для символов `=` и `+` — математические формулы.

Команды и их задание в тексте. Задание печатного знака процента с помощью последовательности символов `\%`

— пример важнейшего понятия ТЕХа, называемого командой. С точки зрения их записи в исходном тексте, команды делятся на два типа. Первый тип — команды, состоящие из знака \ и одного символа после него, не являющегося буквой. Именно к этому типу относятся команды \{,\}, . . .\%, о которых шла речь выше.

Команды второго типа состоят из \ и последовательности букв, называемой именем команды (имя может состоять и из одной-единственной буквы). Например, команды \TeX и \LaTeX генерируют эмблемы систем ТЕХ и LАТЕХ соответственно. В имени команды, а также между \ и именем, не должно быть пробелов, имя команды нельзя разрывать при переносе на другую строку.

В именах команд прописные и строчные буквы различаются. Например, \large, \Large и \LARGE — это три разные команды (они задают различные размеры шрифта).

После команды первого типа (из \ и не-буквы) пробел в исходном тексте ставится или не ставится в зависимости от того, что вы хотите получить на печати:

В чем разница между \$1 и \$ 1?

В чем разница между
\\$1 и \\$ 1?

После команды из \ и букв в исходном тексте обязательно должен стоять либо пробел, либо символ, не являющийся буквой (это необходимо, чтобы ТЕХ смог определить, где кончается имя команды и начинается дальнейший текст).

Вот примеры с командой \sl (она переключает шрифт на наклонный):

38 попугаев.
Подарок мартышке.

\sl38 попугаев.
\sl Подарок мартышке.

Если бы мы написали `\slПодарок` мартышке, то при трансляции TEX зафиксировал бы ошибку (типичную для начинающих) и выдал сообщение о том, что команда `\slПодарок` не определена.

С другой стороны, если после команды из `\` и букв в исходном тексте следуют пробелы, то при трансляции они игнорируются. Если необходимо, чтобы TEX все-таки «увидел» пробел после команды в исходном тексте (например, чтобы сгенерированное с помощью команды слово не сливалось с последующим текстом), надо этот пробел специально организовать. Один из возможных способов — поставить после команды пару из открывающей и закрывающей фигурной скобок `{}` (так что TEX будет знать, что имя команды кончилось), и уже после них сделать пробел, если нужно. Иногда можно также поставить команду `\` (`backslash` с пробелом после него), генерирующую пробел. Вот пример:

Освоить LATEX проще, чем TEX.
Человека, который знает систему
TEX и любит ее, можно
назвать TEXником.

Освоить `\LaTeX\` проще,
чем `\TeX`. Человека,
который знает систему
`\TeX{}` и любит ее, можно
назвать `\TeXником`.

В последней строчке этого примера сознательно не создали пробела после команды `\TeX`, чтобы эмблема TEXа слилась с последующим текстом.

Структура исходного текста. LATEX-файл должен начинаться с команды `\documentstyle` задающей стиль оформления документа. Пример:

```
\documentstyle{book}
```

Слово `book` в фигурных скобках указывает, что документ будет оформлен, как книга: все главы будут

начинаться с нечетной страницы, текст будет снабжен колоннитулами некоторого определенного вида и т. п. Кроме стиля book, в стандартный комплект LATEXа входят стили article (для оформления статей), report (нечто среднее между article и book) и letter (для оформления деловых писем так, как это принято в США). Чтобы задать оформление документа с помощью одного из этих стилей, надо в фигурных скобках после команды \documentstyle указать вместо book название требуемого стиля. Стандартные стили можно (а иногда и нужно) менять, можно создавать и новые стили, но пока что будем исходить из стандартных стилей.

После команды \documentstyle могут следовать команды, относящиеся ко всему документу и устанавливающие различные параметры оформления текста, например, величину абзацного отступа (вообще-то все эти параметры определяются используемым стилем, но может случиться, что Вам понадобится сделать в них изменения). Далее должна идти команда

```
\begin{document}
```

Только после этой команды может идти собственно текст. Если Вы поместите текст или какую-нибудь команду, генерирующую текст (например, \LaTeX) до \begin{document}, то LATEX выдаст сообщение об ошибке. Часть файла, расположенная между \documentstyle и \begin{document}, называется преамбулой. Заканчиваться файл должен командой

```
\end{document}
```

Если даже после \end{document} в файле и написано еще что-то, LATEX это проигнорирует.

Следующий пример показывает минимальный LATEX-

файл, составленный по всем правилам. Ничего интересного в результате его обработки не напечатается, но уж зато и сообщения об ошибках Вы не получите.

```
\documentstyle{article}
\begin{document}
Проба пера.
\end{document}
```

Группы. Вторым важнейшим понятием ТЭХа является понятие группы. Чтобы понять, что это такое, рассмотрим пример.

При обработке ТЭХом исходного файла набор текста в каждый момент идет каким-то вполне определенным шрифтом (он называется текущим шрифтом). Изначально текущим шрифтом является «обычный» прямой шрифт (полученному он называется «roman»).

Команда `\sl`, с которой мы уже столкнулись в разделе выше, переключает текущий шрифт на наклонный, а команда `\bf` — на полужирный:

Полужирный шрифт начнется со следующего слова; дальше так и пойдет. Теперь *наклонный* Теперь *шрифт*, и снова полужирный до нового переключения.

Полужирный шрифт начнется со `\bf` следующего слова; дальше так и пойдет. `\sl` наклонный шрифт, `\bf` и полужирный, до нового переключения.

Но как же быть, если нужно печатать полужирным шрифтом не весь текст, а только его часть? Можно было бы включить в текст команду `\rm`, переключающую шрифт снова на «обычный». Но есть более простой способ: часть текста, которую вы хотите оформить полужирным шрифтом, можно заключить в фигурные скобки, и дать команду `\bf`

внутри этих скобок. Тогда сразу же после закрывающей фигурной скобки TEX «забудет» про то, что шрифт переключался на полужирный, и будет продолжать набор тем шрифтом, который был до скобок:

Полужирным шрифтом набрано
только **это** слово; после скобок
все идет, как прежде.

Полужирным шрифтом набрано
только **{bf это}** слово; после
скобок все идет, как прежде.

Сами по себе фигурные скобки не генерируют никакого текста и не влияют на шрифт; единственное, что они делают — это ограничивают группу внутри файла. Как правило, задаваемые командами TEXa изменения различных параметров (в нашем случае — текущего шрифта) действуют в пределах той группы, внутри которой была дана соответствующая команда, по окончании группы (после закрывающей фигурной скобки, соответствующей той фигурной скобке, что открывала группу) все эти изменения забываются и восстанавливается тот режим, который был до начала группы. Проиллюстрируем все сказанное следующим примером, в котором используется еще команда `\it` (она переключает шрифт на курсивный):

Сначала переключим шрифт на **полужирный**, затем на *курсив*; временно перейдем снова на **полужирный**; посмотрите, как восстановится шрифт после конца группы.

Сначала {переключим шрифт `\bf` на полужирный, затем на `\it` курсив; временно перейдем снова на `{bf полужирный;}` посмотрите, как восстановился} шрифт после кон{ца г}руппы.

Как видите, группы могут быть вложены друг в дружку. Обратите внимание, что внутри внешней группы полужирный шрифт начался не с того места, где была открывающая скобка, а только после команды `\bf` (именно команда, а не скобка, переключает шрифт). Шутки ради

создали еще одну группу из двух последних буквы слова конца, первой буквы слова группы и пробела между ними; как и должно быть, на печати это никак не отразилось: ведь внутри скобок мы ничего не делали!

Трюк с постановкой пары скобок `{ }` после имени команды, о котором шла речь выше - тоже пример использования групп. В этом случае скобки ограничивают «пустую» группу; ставятся они в качестве не-букв, ограничивающих имя команды и при этом никак не влияющих на печатный текст.

Фигурные скобки в исходном тексте должны быть сбалансированы: каждой открывающей скобке должна соответствовать закрывающая. Если Вы почему-либо не соблюли это условие, при трансляции вы получите сообщение об ошибке.

Некоторые команды, называемые глобальными, сохраняют свое действие и за пределами той группы, где они были употреблены. Всякий раз, когда идет речь о глобальной команде, это будет специально оговариваться.

Параметры. Наряду с текущим шрифтом, о котором уже шла речь, TEX в каждый момент обработки исходного текста учитывает значения различных параметров, таких, как величина абзацного отступа, ширина и высота страницы, расстояние по вертикали между соседними абзацами, а также великое множество других важных вещей. Расскажем, как можно менять эти параметры, если это понадобится.

Параметры TEX обозначаются аналогично командам: с помощью символа `\` («backslash»), за которым следует либо последовательность букв, либо одна не - буква. Например, `\parindent` обозначает в TEXе величину абзацного отступа; если нам понадобилось, чтобы абзацный отступ равнялся двум сантиметрам, можно написать так:

```
\parindent=2cm
```

Аналогично поступают и в других случаях: чтобы изменить параметр, надо написать его обозначение, а затем, после знака равенства, значение, которое мы «присваиваем» этому параметру; в зависимости от того, что это за параметр, это может быть просто целое число, или длина (как в разобранным примере), или еще что-нибудь.

Команды с аргументами. Команды наподобие `\LaTeX` или, скажем, `\bf` действуют «сами по себе»; многим командам, однако, необходимо задать аргументы. Первый пример тому дает команда `\documentstyle:` слово, указываемое в фигурных скобках — ее аргумент; если его не указать, то произойдет ошибка. В LATEXе аргументы команд бывают обязательные и необязательные. Обязательные аргументы задаются в фигурных скобках; если для команды предусмотрено наличие обязательных аргументов, она без них правильно работать не будет. У многих команд предусмотрены также и необязательные аргументы, которые влияют на работу команды, коль скоро они указаны, но при этом нормальная работа команды не нарушится и при их отсутствии. Необязательные аргументы задаются в квадратных скобках.

В частности, у команды `\documentstyle` предусмотрен один обязательный аргумент, о котором уже шла речь, и один необязательный: в квадратных скобках перед обязательным аргументом можно указать список (через запятую) так называемых стилевых опций, то есть дополнительных особенностей оформления. Например, если мы хотим, чтобы книга набиралась шрифтом кегля 12 вместо кегля 10, принятого по умолчанию, и притом в две колонки, мы должны начать файл командой

```
\documentstyle[12pt,twocolumn]{book}
```

Необязательных аргументов может быть предусмотрено

несколько; иногда они должны располагаться до обязательных, иногда после. В любом случае порядок, в котором должны идти аргументы команды, надо строго соблюдать. Между скобками, в которые заключены аргументы, могут быть пробелы, но не должно быть пустых строк.

Окружения. Еще одна важная конструкция LATEXа — это окружение (environment). Окружение — это фрагмент файла, который начинается с текста

```
\begin{Имя_окружения}
```

где {Имя_окружения} представляет собой первый обязательный (и, возможно, не единственный) аргумент команды `\begin`. Заканчивается окружение командой

```
\end{Имя_окружения}
```

(команда `\end` имеет только один аргумент — имя завершаемого ей окружения).

Например:

Все строки этого абзаца будут центрированы; переносов слов не будет, если только слово, как дезоксирибонуклеиновая кислота, не длиннее строки.

```
\begin{center}
```

Все строки этого абзаца центрированы; переносов слов не будет, если только слово, как дезоксирибонуклеиновая кислота, не длиннее строки.

```
\end{center}
```

Каждой команде `\begin`, открывающей окружение, должна соответствовать закрывающая его команда `\end` (разумеется, с тем же именем окружения в качестве аргумента).

Важнейшим свойством окружений является то, что они действуют и как фигурные скобки: часть файла, находящаяся внутри окружения, образует группу. Например, внутри окружения `center` в вышеприведенном примере можно было бы сменить шрифт, скажем, командой `\it`, и при этом после команды `\end{center}` восстановился бы тот шрифт, что был перед окружением.

Звездочка после имени команды. В LATEXе некоторые команды и окружения имеют варианты, в которых непосредственно после имени команды или окружения ставится звездочка `*`. Например, команда `\section` означает «начать новый раздел документа», а команда `\section*` означает «начать новый раздел документа, не нумеруя его». После имени команды со звездочкой пробелы не игнорируются; если команда со звездочкой имеет аргументы, пробела между звездочкой и аргументами быть не должно.

Единицы длины. Многие параметры, используемые TEXом, являются размерами в нижеследующей таблице собраны единицы длины (кроме нескольких экзотических), которые можно использовать в TEXе при задании размеров.

pt	пункт ~0.35 миллиметра
pc	пика=12pt
mm	миллиметр
cm	сантиметр=10mm
in	дюйм=25,4mm

Можно задавать размеры с помощью любой из этих единиц; при записи дробного числа можно использовать как десятичную запятую, так и десятичную точку (в таблице мы использовали оба способа); прописные и строчные буквы в обозначениях единиц длины не различаются.

Даже если длина, которую вы указываете TEXу, равна

нулю, все равно необходимо указать при этом нуле какую-нибудь из используемых TEXом единиц длины. Например, если написать

```
\parindent=0
```

то вы получите сообщение об ошибке; вместо 0 надо было бы писать, например, 0pt или 0in.

Кроме перечисленных, в TEXе используются еще две «относительные» единицы длины, размер которых зависит от текущего шрифта. Это em, приблизительно равная ширине буквы M текущего шрифта, и ex, приблизительно равная высоте буквы x текущего шрифта. Эти единицы удобно использовать в командах, которые должны работать единообразно для шрифтов разных размеров. В частности, расстояние в 1em на глаз обычно воспринимается как «один пробел».

Автоматическая генерация ссылок. LATEX предоставляет возможность организовать ссылки на отдельные страницы или разделы документа таким образом, чтобы программа сама определяла номера страниц или разделов в этих ссылках. Объясним это на примере.

Представим себе, что Вам нужно сослаться на какое-то место в вашем тексте. Проще всего указать страницу, на которой это место находится, написав «. . . как мы уже отмечали на стр. 99» или что-то в этом роде. Проблема, однако, в том, что заранее нельзя угадать, на какую страницу печатного текста попадет это место. Вместо того, чтобы гадать, можно сделать следующее:

1. Пометить то место, на которое вы хотите сослаться в дальнейшем (или предшествующем) тексте;

2. В том месте текста, где вы хотите поместить ссылку, поставить команду-ссылку на вашу метку.

Конкретно это реализуется так. Помечается любое место

текста с помощью команды `\label`. Эта команда имеет один обязательный аргумент (помещаемый, стало быть, в фигурных скобках) — «метку». В качестве метки можно использовать любую последовательность букв, цифр и знаков препинания (не содержащую пробелов, фигурных скобок и символов `\`).

На пример, эта команда может иметь вид:

```
\label{wash}
```

Ссылка на страницу, на которой расположена метка, производится с помощью команды `\pageref`. У нее также один обязательный аргумент — та самая метка, на которую вы хотите сослаться. Пример:

Обязательно мойте руки перед едой, чтобы не заболеть. Как известно (см. [стр 99](#)), руки надо мыть.

Обязательно мойте [руки\label{wash}](#) перед едой, чтобы не заболеть. Как известно (см. [\стр~\pageref{wash}](#)), руки надо мыть.

Обратите внимание, что мы поставили команду `\label` рядом с ключевым словом «руки» без пробела, чтобы гарантировать, что будет помечена именно та страница, на которую попало это слово.

В этом примере мы использовали еще значок `~`, чтоб при печати сокращение «стр.» попало на ту же строку, что и номер страницы, и команду `\` (backslash с пробелом), чтобы на печати пробел после сокращения «см.» не получился больше, чем надо.

После того, как вы впервые вставите в свой файл команды `\label` и `\pageref`, при трансляции вы получите сообщение о том, что ваша ссылка не определена, а не печати

или при просмотре увидите на месте своих ссылок вопросительные знаки. Дело в том, что в этот момент LATEX еще не знает значения ваших меток: он только записывает информацию о них в специальный файл (с тем же именем, что у обрабатываемого файла, и расширением .aux); при следующем запуске он прочтет эту информацию и подставит ссылки. В промежутке между двумя запусками в файл могли быть внесены изменения, что может привести к сдвигу нумерации страниц. Если такие изменения действительно произошли, LATEX сообщит вам об этом и попросит запустить программу еще раз, чтобы получить корректные ссылки.

Если вы после двух запусков подряд получите сообщение о неопределенной ссылке, значит, в исходном тексте присутствует ошибка (вероятнее всего, опечатка в аргументе команды `\pageref`; возможно, вы забыли включить в текст команду `\label`).

На место, помеченное с помощью команды `\label`, можно сослаться с помощью команды `\ref`, а не `\pageref` — тогда на печати получится не номер страницы, а номер раздела документа, в котором находится метка, или номер рисунка, или номер элемента в «нумерованном перечне».... — пометить с возможностью ссылки можно почти любой элемент документа. /24/

5. СОВРЕМЕННЫЕ АЛГОРИТМИЧЕСКИЕ ТЕХНОЛОГИИ

5.1 Технологии построения корпоративных информационных систем

С некоторым опозданием по сравнению с Западом, в России медленно, но верно начинают понимать всю важность

комплексного подхода в автоматизации предприятий и организаций. На собственном горьком опыте и благодаря множеству публикаций в компьютерной прессе многие осознали, что эффективность автоматизации в первую очередь зависит от того, насколько широко она охватывает все сферы деятельности юридического лица. Отчасти именно поэтому в последнее время стала столь популярной идея построения корпоративных информационных систем (КИС). И хотя понятие корпоративности подразумевает наличие довольно крупной, территориально-распределенной информационной системы, все же вполне правомерно присовокупить сюда системы любых предприятий, вне зависимости от их масштаба и формы собственности. В конце концов, любая фирма, организация или государственное учреждение, имея сегодня в своем активе сеть с одним сервером и десятком компьютеров, по всем правилам развития, может или даже должна существенно расширяться завтра. Кроме того, наверное, все без исключения информационные системы начинают создаваться с какого-либо одного подразделения, реализующего некоторый самостоятельный, но не обязательно самый главный, цикл деятельности, и это ничуть не мешает проверять действенность комплексного подхода автоматизации. Поэтому далее в контексте данной статьи упоминания КИС имеют непосредственное отношение к любым информационным системам, и правильнее даже будет, сохранив общую аббревиатуру КИС, заменить первое слово новомодного понятия, переименовав его в комплексные информационные системы. Вызвано это прежде всего тем, что речь пойдет о том, как построить комплексную инфосистему вне зависимости от формы собственности и профиля деятельности организации.

5.2 Функционал КИС как определяющий фактор выбора ее структуры

Эффективное управление современным предприятием представляет собой довольно нетривиальную задачу, учитывая многообразие используемых ресурсов и высокую скорость изменения операционного окружения. Основными функциями управления являются, как известно, планирование, организация, активизация, координация, контроль и анализ, которые осуществляются в многомерном пространстве различных областей деятельности предприятия. Формируемые в ходе выполнения вышеперечисленных функций управленческие решения служат отправным моментом для конкретных исполнителей. В связи с тем, что автоматизация исполнения должностных обязанностей и отдельных поручений фактически стала в последнее время стандартом де-факто, особую остроту приобретает проблема автоматизации непосредственно управленческих функций.

Таким образом, наиболее существенной чертой комплексной информационной системы должно стать расширение контура автоматизации для получения замкнутой, саморегулирующейся системы, способной гибко и оперативно перестраивать принципы своего функционирования.

Очевидно, что в состав КИС должны войти средства для документационного обеспечения управления, информационной поддержки предметных областей, коммуникационное программное обеспечение, средства организации коллективной работы сотрудников и другие вспомогательные (технологические) продукты. Из этого, в частности, следует, что обязательным требованием к КИС является интеграция большого числа программных продуктов.

Подобная широкопрофильная система должна в равной,

максимально допустимой, степени удовлетворить все подразделения организации, по возможности сохранить существующие бизнес-процессы, а также методы и структуру управления. Без привлечения автоматизации практически нельзя контролировать постоянно меняющиеся бизнес-процессы.

Наполнение предметной части КИС может существенно изменяться в зависимости от профиля деятельности предприятия, включая, например, программное обеспечение для финансового анализа, складскую программу либо PDM-систему. Некоторая часть КИС определяется такими характеристиками, как масштаб организации и объемы информационных работ. С их увеличением становится актуальным внедрение специализированных модулей делопроизводства и архивного хранения, которые способны поддерживать крупные электронные архивы смешанной документации с обеспечением необходимого уровня надежности и безопасности хранения информации.

В дополнение к функционалу, структуру КИС определяют и реализующие данный функционал технологии. С этой точки зрения, современные информационные системы должны отвечать целому набору обязательных требований. Среди них, в первую очередь, стоит отметить использование архитектуры клиент-сервер с возможностью применения большинства промышленных БД, обеспечение безопасности с помощью различных методов контроля и разграничения доступа к информационным ресурсам, поддержку распределенной обработки информации, модульный принцип построения из оперативно-независимых функциональных блоков с расширением за счет открытых стандартов (API, COM и другие), а также поддержку технологий Internet/intranet.

Кроме того, немаловажную роль играют и другие — эксплуатационные — характеристики: легкость

администрирования, эргономичность, наличие локализованного (русифицированного) интерфейса.

Наиболее органичным и эффективным способом построения КИС, при котором были бы выполнены вышеперечисленные функции и требования к технологичности, является использование в качестве ядра всего информационного комплекса системы автоматизации деловых процессов.

Для того чтобы понять, почему это именно так, необходимо вспомнить, что, фактически, деятельность любой организации представляет собой нечто иное, как совокупность выработанных в повседневной практике деловых процессов, в которые вовлечены финансовые, материальные, кадровые, информационные и прочие виды ресурсов. Именно деловые процессы определяют порядок взаимодействия отдельных сотрудников и целых отделов, а также принципы построения информационных систем. Поэтому автоматизация предприятия, исходя из делового процесса, наиболее логична, и самое главное, — вполне реальна благодаря современным workflow-системам, выступающим в роли связующего звена, вокруг которого и будут интегрироваться другие программные продукты. Сформированная таким образом КИС характеризуется одновременно универсальностью и эффективностью. Она способна автоматизировать деятельность предприятия практически любой отрасли и при этом позволяет сохранить специфические, критически важные нюансы управленческого и организационного ноу-хау. Кроме того, автоматизацию не нужно начинать с нуля. Интегрированный комплекс может создаваться на базе разрозненных автоматизированных рабочих мест, т. е. с использованием уже имеющегося системного и прикладного программного обеспечения. Не стоит забывать также, что при применении в предложенной структуре современных workflow-систем, которые

характеризуются открытостью и широкими возможностями по настройке, легко достигается динамичная и гибкая модификация целей и функций КИС.

5.3 Создание инфосистем на основе системы автоматизации деловых процессов

Сегодня существует целый ряд систем автоматизации деловых процессов (САДП), заслуживающих самого пристального внимания потребителя, который собрался проводить комплексную автоматизацию. Из зарубежных систем это, в первую очередь, Action Workflow фирмы Action Technologies и продукт фирмы Staffware Inc., который так и называется Staffware; из отечественных — ничуть не уступающая зарубежным конкурентам система WorkRoute компании ВЕСТЬ АО, получившая признание на западном рынке.

Прежде чем перейти непосредственно к рассмотрению вопроса о построении КИС на подобных системах, имеет смысл вкратце познакомиться с их терминами и принципами функционирования.

Работа workflow-систем, как правило, основывается на том, что большая часть деловых процессов представляет собой периодически повторяемую, отрегулированную последовательность действий (выполнение заданий), которая может быть легко формализована. Для этой цели в них с помощью специальных редакторов создаются так называемые карты деловых процессов, которые описывают, кто, когда, на каком рабочем месте (возможно, в удаленном филиале), с помощью каких программ и как должен обработать те или иные данные. Заложенное в карту описание делового процесса позволяет автоматизировать формирование, активизацию, выполнение и контроль различных заданий.

Карта делового процесса создается, а обычно просто

рисуеться мышью с использованием нескольких графических примитивов и затем может быть легко изменена (рис. 24).

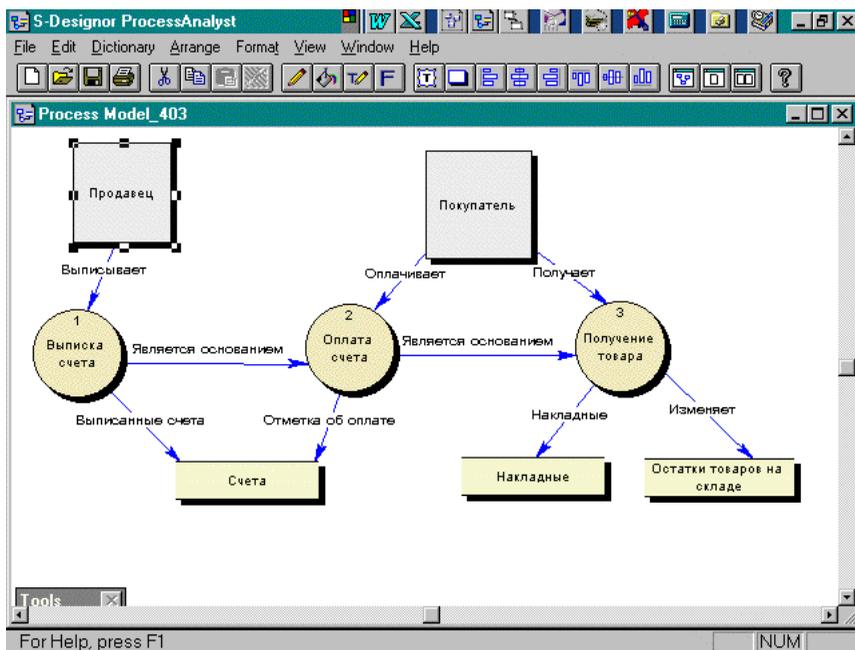


Рис. 24. Карта делового процесса.

Таким образом, без всякого программирования можно за считанные минуты получить реально работающее workflow-приложение. В некоторых workflow-системах создание информационных моделей деловых процессов возможно только с помощью программирования, что представляет собой довольно кропотливый процесс, требующий к тому же специальных знаний. Так, в Action Workflow программирование используется для разработки электронных форм, которые являются неотъемлемой частью бизнес-модели, обеспечивая взаимодействие системы с пользователем на этапах делового процесса.

Важно отметить, что, несмотря на общий подход, workflow-системы сильно различаются по возможностям карт деловых процессов, в связи с чем при выборе такой системы необходимо прежде всего обратить внимание, насколько сложными могут быть структуры деловых процессов и какие в них поддерживаются типы этапов. Стандартный набор должен обязательно включать простой узел (выполнение элементарного действия, например редактирование первого варианта технического проекта), условие (ветвление дальнейшего хода делового процесса в зависимости от условий), ветвление (безусловное разделение процесса на несколько параллельных ветвей), объединение ветвей, скрипт (встроенный язык программирования для автоматического выполнения таких операций, как, скажем, обращение в базу данных внешней прикладной программы с извлечением из нее предварительной информации по техническому заданию), множественные точки входа и выхода из делового процесса.

Также должна существовать возможность определять в контексте карты переменные различных типов, несущие любую смысловую нагрузку и влияющие на ход выполнения работы (допустим, название контрагента по сделке, сумма сделки, дата завершения этапа). Разумеется, должен быть встроенный редактор для создания экранных форм, которые на каждом этапе делового процесса отображают переменные и формируют пользовательский интерфейс workflow-приложения.

Следует помнить, что значения переменных, в идеале, должны считываться не только из базы данных workflow-системы, но и из баз данных прикладных программ, поддерживающих наиболее распространенные промышленные стандарты систем управления базами данных (БД). Это позволяет интегрировать систему автоматизации деловых процессов с внешними приложениями в разрезе совместного использования данных. Что же касается

встроенного языка программирования, о котором выше уже шла речь, то к нему, вполне очевидно, предъявляются такие требования, как простота (например, он должен быть семантически совместим с каким-либо распространенным языком — на сегодняшний день предпочтительнее всего VBA), эффективность, наличие широких возможностей по управлению деловыми процессами и связанными с ними данными. Крайне желательно, чтобы скрипт мог работать с OLE-серверами, запускать внешние программы, взаимодействовать с МАРІ-совместимыми почтовыми системами. Кроме того, учитывая, что workflow-система рассматривается нами как основа КИС, для получения полной интеграции с другими программами и облегчения этого процесса, скорее всего, потребуется наличие открытого программного интерфейса API, который бы позволил управлять системой из внешних программ.

Международной организацией, курирующей разработку стандартов и спецификаций на системы класса workflow, является Workflow Management Coalition (WfMC). Теперь, после небольшого отступления, вернемся к проблеме построения КИС на базе системы автоматизации деловых процессов.

5.4 Функциональные подсистемы КИС

Если workflow-система отвечает большинству вышеперечисленных требований, то это позволит легко объединить вокруг нее любые современные приложения, поддерживающие те или иные стандарты межпрограммного взаимодействия. Ясно, что функциональная направленность интегрированного комплекса в принципе ничем не ограничена, однако некоторые сферы деятельности носят более распространенный характер, нежели другие, и поэтому заслуживают интеграции в первую очередь.

Системы управления документами. Прежде всего, это относится к делопроизводству, иначе говоря, к комплексу операций по созданию, управлению и исполнению документов, ведению электронного архива, организации офисного документооборота. Для реализации таких функций необходимо объединить workflow-систему с системой управления документами (СУД). К системам данного класса относятся, например, DOCS Open американской фирмы PC DOCS, DocuLive (Siemens Nixdorf), Documentum (Documentum, Inc.). Как правило, СУД имеют богатые возможности по интеграции с внешними приложениями (офисными и прикладными программами), которые и “снабжают” СУД документами. Кроме того, рынок СУД изначально ориентирован на КИС масштаба предприятия, в связи с чем все промышленные системы выполнены в архитектуре клиент-сервер и способны работать практически на всех программно-аппаратных платформах, т. е. характеризуются отличной масштабируемостью, переносимостью, безопасностью и надежностью хранения данных, а также обеспечивают распределенный режим работы.

Если составные части КИС поддерживают довольно широкий список оборудования и серверного программного обеспечения, это дает возможность уменьшить затраты, так как увеличивается вероятность того, что необходимые базовые продукты в организации уже есть. На сегодняшний день основными платформами, на которых должны функционировать формирующие КИС СУД, САДП и прикладное программное обеспечение, следует считать Windows NT Server, Novell NetWare, основные разновидности Unix и промышленные БД Oracle, Microsoft SQL Server, Oracle или Sybase.

Важно отметить, что КИС на основе САДП и СУД являются довольно универсальными. Подобные комплексы,

благодаря имеющимся инструментам интеграции, позволяют объединить офисный, (организационно-распорядительный) документооборот с инженерным, в который входит техническая, технологическая и чертежно-конструкторская документация (она, как правило, разрабатывается в САПР и ГИС, например в AutoCAD, MicroStation, КОМПАС), а также любые другие виды информации, вплоть до мультимедиа. Кроме того, в состав КИС может органично влиться программы бухгалтерского, складского и кадрового учета.

Здесь, правда, надо сделать маленькую оговорку, что интеграция осуществляется только между программами, отвечающими некоему набору технических требований, которые ясны из описанных выше приемов построения автоматизированного комплекса. Минимальный уровень интеграции обеспечивает наличие открытых кодов командной строки: лучше, если программа поддерживает стандарт OLE Automation, а совсем хорошо, если она в дополнение ко всему этому имеет сетевую версию, использующую для хранения своих структурированных данных SQL-сервер. Тогда возможно создание мощного и гибкого инструмента, отвечающего современным требованиям по безопасности и надежности. Справедливости ради нужно заметить, что большинство отечественных фирм-разработчиков программного обеспечения уже выпустили или в ближайшее время выпускают версии программ, соответствующие промышленным стандартам межпрограммного взаимодействия, поэтому данное ограничение на интеграцию, скорее всего, не окажется существенным препятствием на пути построения КИС. Вряд ли стоит напоминать, что все зарубежные, да и отечественные офисные пакеты уже поддерживают OLE и поэтому прекрасно интегрируются.

И если уж разговор зашел о документообороте, то стоит обратить внимание на еще один нюанс. Существующие

системы автоматизации деловых процессов, как правило, поддерживают одну из двух метафор маршрутизации: жесткую или свободную. Первая из них как раз и определяется картой делового процесса, которая конструируется заранее. Таким образом, применение жесткой маршрутизации допустимо там, где технология обработки документов или просто выполнения заданий хорошо формализуется. Большинство зарубежных систем управления относится именно к этому классу программ, что в принципе вполне объяснимо — за границей любят порядок во всем, в том числе и в деловых процессах. В то же время одной из особенностей российского ведения хозяйства является наличие случайного или субъективного фактора, вносящего некоторую неопределенность в структуру деловых процессов. В связи с этим в большинстве отечественных организаций обычно используется свободная маршрутизация, когда последовательность этапов деловых процессов определяется (и доопределяется) на стадии выполнения заданий.

Таким образом, мы приходим к заключению, что САДП должна учитывать и национальные особенности управления, и потому, в нашем случае, поддерживать обе парадигмы — и жесткую, и свободную маршрутизацию. Подобный симбиоз двух разных подходов позволяет избежать крайностей и получить то сочетание жесткой и свободной маршрутизации, которое наиболее оптимально соответствует специфике предприятия.

Средства обработки бумажных документов. Полный переход к электронной форме информации пока еще невозможен по ряду объективных причин, обусловленных не ограничениями современных информационных технологий (они как раз уже вполне созрели для этого и имеют в своем арсенале все необходимые инструменты), а внешними по отношению к ним факторами (законодательство, инерция мышления, недостаточный уровень компьютеризации в

среднем по стране). Поэтому, говоря о концепции построения КИС, нельзя не упомянуть такую актуальную на сегодняшний день проблему, как совмещение в документообороте предприятия электронных и бумажных документов.

Практически все современные СУД, хранящие документы, предоставляют те или иные функции по обработке бумажных документов. Обычно они реализуются с помощью специального модуля, который позволяет автоматизировать сканирование и сохранение образов многостраничных документов, а также их распознавание и аннотирование. Последняя функция, в частности, дает возможность наносить на дополнительные слои произвольные пометки, сохраняя при этом первооснову.

При использовании специальных аппаратных средств (например, высокопроизводительных сканеров с поддержкой коррекции изображений и удаления искажений, дополнительных плат постобработки, фильтрации изображений) такой модуль позволяет создать линию массового (промышленного) ввода документов. Применение подобных комплексов обеспечивает не только высокую скорость преобразования данных, но и экономит место на носителях информации, благодаря более компактному представлению данных (за счет увеличения качества изображений и более мощных алгоритмов сжатия).

Важно, что часть участвующих в документообороте бумаг (по сути дела все, кроме тех, что предназначены для внешнего легитимного использования или предоставляются для отчетности в государственные органы) может быть уже сегодня переведена в электронную форму и именно в таком виде визироваться, согласовываться и утверждаться, что, кстати говоря, уже сделано в некоторых наиболее динамично развивающихся отечественных организациях. В результате появляется возможность реализовать смешанный электронно-

бумажный документооборот с необходимым заказчику соотношением бумажных и безбумажных технологий.

Учитывая сравнительно большой объем документов, представленных в графическом виде, имеет смысл вспомнить о включении в КИС поддержки носителей информации с низкой удельной стоимостью хранения (например, CD-ROM, CD-RW, магнитооптические диски и библиотеки, стримеры), которые позволяют создавать сравнительно дешевые электронные архивы объемами до нескольких терабайт. Очевидно, что наиболее оптимальным модулем для выполнения функций интеграции с подобными хранилищами следует рассматривать СУД. Именно при разработке КИС желательно выбирать СУД, которая поддерживает иерархическое многоуровневое (в смысле использования различных классов носителей) хранение документов. Например, оперативный архив документов может храниться на быстрых дисковых массивах сервера, обширные справочные материалы в магнитооптической библиотеке, а архив документации за прошлые годы — на CD-ROM.

Системы поддержки принятия решений. Следующим немаловажным моментом в функционировании КИС является необходимость обеспечить помимо средств генерации данных также и средства их анализа. Имеющиеся во всех современных СУД и БД средства построения запросов и различные механизмы поиска хотя и облегчают извлечение нужной информации, но все же не способны дать достаточно интеллектуальную ее оценку, т. е. сделать обобщение, группирование, удаление избыточных данных и повысить достоверность за счет исключения ошибок и обработки нескольких независимых источников информации (как правило, не только корпоративных баз данных, но и внешних, расположенных, например, в Internet). Проблема эта становится чрезвычайно важной в связи с лавинообразным возрастанием объема информации и увеличением требований

к инфосистемам по производительности — сегодня успех в управлении предприятием во многом определяется оперативностью принятия решений, данные для которых и предоставляет КИС. В этом случае на помощь старым методам приходит оперативная обработка данных (On-Line Analytical Processing, OLAP). Сила OLAP заключается в том, что в отличие от классических методов поиска запросы здесь формируются не на основе жестко заданных (или требующих для модификации вмешательства программиста и, следовательно, времени, т. е. об оперативности речь идти не может) форм, а с помощью гибких нерегламентированных подходов. OLAP обеспечивает выявление ассоциаций, закономерностей, трендов, проведение классификации, обобщения или детализации, составление прогнозов, т. е. предоставляет инструмент для управления предприятием в реальном времени.

Не останавливаясь на тонкостях организации различных моделей OLAP (например, таких, как радиальная схема, “звезда”, “снежинка” или многомерные таблицы), суть работы OLAP можно описать как формирование и последующее использование для анализа массивов предварительно обработанных данных, которые еще называют предвычисленными индексами. Их построение становится возможным исходя из одного основополагающего предположения, — будучи средством принятия решений, OLAP работает не с оперативными базами данных, а со стратегическими архивами, отличающимися низкой частотой обновления, интегрированностью, хронологичностью и предметной ориентированностью. Именно неизменность данных и позволяет вычислять их промежуточное представление, ускоряющее анализ гигантских объемов информации.

Сегодня доступен целый ряд различных систем OLAP, ROLAP (реляционный OLAP), MOLAP (многомерный OLAP)

— Oracle Express, Essbase (Arbor Software), MetaCube (Informix) и другие. Все они представляют собой дополнительные серверные модули для различных БД, способные обрабатывать практически любые данные. Интеграция КИС с системой оперативного анализа информации позволит во много раз увеличить эффективность первой, поскольку данные в ней будут не просто храниться, а работать.

Системы, основанные на применении Internet-технологий. Неотъемлемой чертой современных КИС стало применение технологий Internet. Учитывая этот факт, при выборе составляющих КИС необходимо отдавать предпочтение программам (СУД, САДП), которые поддерживают полноценную работу из обычного браузера, фактически, имеют так называемый тонкий клиент и специальное серверное программное обеспечение, обеспечивающее функционирование данного клиента. Как правило, такое техническое решение позволяет использовать стандартные хранилища данных (библиотеки документов, базы данных) из локальных, корпоративных и глобальных сетей, не требуя существенных затрат на дополнительное администрирование и поддержание целостности, надежности и безопасности хранения данных.

Рассматривая вопрос применения Internet-технологий, нельзя не затронуть такую важную проблему, как обеспечение информационной безопасности. Для предотвращения несанкционированного доступа к документам и для исключения возможных диверсий злоумышленников встроенных средств СУД и САДП недостаточно. Поэтому в состав КИС обязательно должны войти специальные программно-аппаратные средства защиты.

Они, в частности, позволяют шифровать данные, поддерживают электронную цифровую подпись и могут

проводить на ее основе аутентификацию пользователей. Все это обеспечивает достоверность и целостность информации внутри КИС. В качестве подобной системы криптографической защиты информации можно, например, использовать одну из модификаций (в зависимости от операционной системы и требуемой сложности защиты) СКЗИ “Верба” (разработка Московского отделения Пензенского научно-исследовательского электротехнического института). Обычно СКЗИ представляют собой открытые системы, допускающие интеграцию с внешними программами, но необходимо обратить особое внимание на то, сертифицирована ли СКЗИ и по какому классу. В России сертификацией подобных систем занимается ФАПСИ.

Эффективность программных средств защиты может быть существенно повышена за счет применения аппаратных и биометрических средств: аппаратных ключей, смарт-карт, устройств распознавания отпечатков пальцев, сетчатки глаза, голоса, лица, оцифрованной подписи.

В дополнение к ним на стыке сегментов локальных сетей и Internet желательна установка брандмауэров — средств контроля за внешними (входящими и исходящими) соединениями. (Наиболее типичным примером системы данного класса является CheckPoint FireWall-1 фирмы CheckPoint Software.) Они позволяют отслеживать передачу информации практически всех известных на сегодняшний день протоколов Internet.

Средства стратегического планирования. Возможности САДП по оперативному контролю и управлению предприятием в некоторых случаях недостаточны. Прежде всего, это имеет место в крупных организациях, где ведутся долгосрочные проекты с привлечением большого числа различных ресурсов. Как известно, для решения подобных задач применяются системы

управления проектами (например, Microsoft Project, Symantec Time Line). Введение их в состав КИС позволит осуществлять стратегическое планирование и управление предприятием.

Практически все системы управления проектами (СУП) берут за основу некий план проекта, составляемый на предварительном этапе. Затем в соответствии с этим планом автоматически организуются выполнение, анализ и управление рабочими этапами плана. Методологии САДП и СУП, несмотря на различие в масштабах автоматизируемых действий, в известной степени перекликаются, что позволяет установить двустороннюю синхронизацию между схемами workflow и стратегическим планом. Здесь важно отметить, что в рамках СУП календарное планирование и ход выполнения этапов происходят в полуавтоматическом режиме. Фактически выдача рабочих заданий, контроль их исполнения и анализ (оценка) ситуации осуществляются пользователем вручную. Интеграция с workflow-системой кардинально решает эту проблему.

В терминах САДП проект представляет собой деловой процесс, который состоит из последовательных и параллельных этапов с монотонным характером развития, т. е. подразумевает, что с течением времени не происходит возврата на предыдущие этапы. Кроме того, в стандартной практике планирования проектов можно отметить отсутствие условных этапов, достижимых только при выполнении некоторого условия на одном из предшествующих.

Таким образом, САДП реализует все метафоры управления СУП, что дает возможность по разработанному проекту создать описание делового процесса, формирующего детализированные задания для каждого из его участников, и контролировать процесс его исполнения, предоставляя обратную связь руководителю проекта и всем другим сотрудникам предприятия, которые заинтересованы в получении информации по проектам.

Чисто технически интеграция современных САДП и СУП не вызывает вопросов, поскольку обе системы, как правило, соответствуют стандартам межпрограммного взаимодействия и могут разделять данные. Например, система управления проектами Time Line 6.5 поддерживает стандарты ODBC, OLE 2.0 и DDE. SQL-архитектура базы данных Time Line 6.5 позволяет получать доступ к данным о проекте, используя внешние приложения, в том числе и workflow-системы, и строить распределенные корпоративные системы управления.

В результате интеграции данных систем информация по задачам проекта из СУП автоматически отправляется исполнителям, и система автоматизации деловых процессов обеспечивает механизм согласования исполнения работ. По мере выполнения работ участники проекта могут быстро отчитываться о состоянии порученных им заданий, сообщая о проценте выполнения задачи, допустим, по электронной почте (ее поддержка встроена во многие workflow-системы), и эта информация будет автоматически передаваться в систему управления проектами, где используется для актуализации плана проекта.

Подведем итог вышеизложенному. Комплексная информационная система — это совокупность технических и программных средств предприятия, реализующих идеи и методы автоматизации. Комплексная автоматизация подразумевает перевод в плоскость компьютерных технологий всех основных деловых процессов организации. И использование специальных программных средств, обеспечивающих информационную поддержку бизнес-процессов, в качестве основы КИС представляется наиболее оправданным и эффективным. Современные системы управления деловыми процессами позволяют интегрировать вокруг себя различное программное обеспечение, формируя единую информационную систему. Тем самым решаются

проблемы координации деятельности сотрудников и подразделений, обеспечения их необходимой информацией и контроля исполнительской дисциплины, а руководство получает своевременный доступ к достоверным данным о ходе производственного процесса и имеет средства для оперативного принятия и воплощения в жизнь своих решений. И, что самое главное, полученный автоматизированный комплекс представляет собой гибкую открытую структуру, которую можно перестраивать на лету и дополнять новыми модулями или внешним программным обеспечением.

Информационная система может строиться с применением послойного принципа. Так, в отдельные слои можно выделить специализированное программное обеспечение (офисное, прикладное), непосредственно workflow, систему управления документами, программы поточного ввода документов, а также вспомогательное программное обеспечение для связи с внешним миром и обеспечения доступа к функционалу системы через коммуникационные средства (e-mail, Internet/intranet). Среди преимуществ такого подхода следует отметить возможность внесения изменений в отдельные программные компоненты, расположенные в одном слое, без необходимости коренных переделок на других слоях, обеспечить формальную спецификацию интерфейсов между слоями, поддерживающих независимое развитие информационных технологий и реализующих их программных средств. Причем применение открытых стандартов позволит безболезненно осуществлять переход с программных модулей одного производителя на программы другого (например, замена почтового сервера или СУД). Кроме того, послойный подход позволит повысить надежность и устойчивость к сбоям системы в целом. /25/

6. ПАКЕТ WEB-ДИЗАЙНА FLASH-MX

6.1 Основы работы с программой FLASH-MX. Основные понятия. Объект, символ, экземпляр

При построении любого Flash-фильма используется объектно-ориентированный подход. Это означает, что все элементы фильма интерпретируются как объекты того или другого типа, для каждого из которых заданы некоторые свойства и определен набор допустимых операций. Например, для объекта «Текст» должен быть установлен размер символов, способ начертания, цвет и т. д. Текст можно определенным образом редактировать, вырезать; копировать, создавать на его основе текстовые гиперссылки и т. п. То же самое можно сказать о графических изображениях и о звуке. Тем не менее, при работе с Flash вместо понятия «объект» чаще используется термин *символ* (Symbol). Основное различие между ними состоит в следующем.

Символ представляет собой своеобразный шаблон объекта с определенным набором свойств. Символ хранится в специальной *библиотеке символов* (Library) и может быть многократно использован как в одном и том же фильме, так и в нескольких фильмах. Каждая новая копия символа, помещенная в фильм, называется *экземпляром символа* (Instance). Экземпляр наследует все свойства самого символа, и между ними устанавливается связь: при изменении свойств символа соответствующие изменения автоматически применяются ко всем его экземплярам. Очевидно, что такой подход существенно экономит силы и время создателя фильма. Кроме того, механизм символов позволяет сократить и размер фильма: если в нем используется несколько экземпляров символа, то информация о его свойствах не дублируется. Вместе с тем, вы можете изменять некоторые

свойства конкретного экземпляра, что не влияет на свойства символа-оригинала. Например, можно изменить размер и цвет экземпляра, а если речь идет о звуковом символе — добавить тот или иной эффект.

Как правило, динамика в Flash-фильмах обеспечивается за счет того, что в течение некоторого интервала времени изменяются те или иные свойства экземпляра (например, координаты, цвет, размер, прозрачность и т. д.), то есть изменяется состояние экземпляра. С каждым состоянием экземпляра связан отдельный *кадр* фильма (Frame). Кадр, соответствующий изменению состояния экземпляра, называется *ключевым кадром* (Keyframe). Ключевой кадр сам рассматривается как объект соответствующего типа, свойства которого пользователь может изменять. Для ключевых кадров предусмотрены специальные функции и команды, предназначение которых мы поясним немного позже.

Динамика смены кадров фильма описывается с помощью *временной диаграммы* (Timeline). В качестве параметров временной диаграммы можно указывать частоту смены кадров, моменты начала и завершения движения объектов и т. д.

В фильме может использоваться несколько различных объектов. Состояние каждого из них может изменяться независимо от других, либо вообще оставаться неизменным (если, например, некоторый объект используется в качестве фона). Чтобы упростить описание поведения различных элементов фильма, каждый из них помещается, как правило, на отдельный *слой* (Layer). Разработчики Flash для пояснения роли слоев в фильме сравнивают их с листами прозрачной кальки. Собрав воедино «стопку» таких листов, вы можете получить некую сцену, на которой действуют различные «персонажи». Кстати о сценах. *Сцена* (Scene) - это еще один термин, используемый при работе с Flash. Каждая сцена представляет собой определенное сочетание слоев. Для

простых фильмов бывает достаточно создать и описать одну-единственную сцену, содержащую один слой. Для более сложных может потребоваться создать несколько различных сцен. Переход от одной сцены к другой определяется уже не временной диаграммой, а несколько иным механизмом. В простейшем случае сцены фильма выполняются последовательно, в соответствии с их порядковыми номерами. Для более сложного построения фильма используются средства языка ActionScript.

При создании сложных фильмов достаточно важную роль играет еще одно понятие — *клип* (Clip, или Movie clip). Клип — это специальный тип символа. Он представляет собой как бы мини-фильм, для которого создается собственная временная диаграмма и устанавливаются собственные параметры (например, частота смены кадров). Клип, как и любой другой элемент фильма, может быть включен в библиотеку символов для многократного использования в фильме. Каждому экземпляру клипа может быть назначено собственное имя.

Любой элемент фильма может быть использован и внутри клипа. Разрешается также создавать «вложенные» клипы. Если требуется описать некоторые дополнительные условия активизации клипа внутри фильма, то для этого могут быть использованы средства языка ActionScript. В состав клипа могут включаться также интерактивные элементы (например, кнопки).

6.2 Последовательность действий при создании Flash-фильма

При запуске большинства Windows-приложений автоматически создается «заготовка» (пустой документ, бланк) того типа, для работы с которым предназначено данное приложение. Например, при открытии окна

текстового процессора Word в нем появляется пустой документ, в который вы можете ввести требуемый текст, добавить графику и т. д. Для нового документа используются параметры, установленные по умолчанию (размер и стиль шрифта, ширина полей и т. п.). Чтобы документ выглядел так, как хотели бы именно вы, целесообразно установить требуемые значения соответствующих параметров.

Подобным образом организована и работа с редактором Flash. При его запуске автоматически создается «заготовка» будущего фильма (точнее, его первой сцены). Если это необходимо, автор может установить собственные значения общих свойств фильма, таких как частота смены кадров, размеры окна, используемого для отображения фильма, цвет фона, и некоторые другие.

После установки общих параметров фильма можно переходить к формированию первой сцены фильма. Решив, какие именно объекты должны присутствовать на этой сцене, вы последовательно помещаете каждый из объектов на новый слой. Установка параметров слоя — это один из наиболее ответственных этапов построения фильмов. Поскольку в фильме слои (или помещенные на них объекты) тем или иным образом взаимодействуют друг с другом, после описания очередного слоя может оказаться, что требуется отредактировать созданные ранее. В этом нет ничего страшного, это вполне естественно, и Flash предоставляет автору возможность скорректировать предыдущие действия.

Очередной объект, подлежащий включению в фильм, может быть либо взят из библиотеки символов, либо создан заново.

Если объект пока существует только в вашем воображении или на бумаге, то Flash превращается в графический редактор, предоставляющий все основные инструменты для рисования, закрашивания, ретуширования и комбинирования различных изображений. При этом как

отдельные компоненты создаваемого объекта, так и законченные изображения могут быть импортированы из других графических форматов.

Те объекты, которые предполагается использовать в фильме неоднократно (хотя бы дважды), целесообразно сразу создавать как новые символы.

Хотя Flash и обладает достаточно «дружественным» интерфейсом, создание Flash-фильмов является весьма непростым делом. Поэтому после завершения очередного шага полезно проверить полученный результат, *протестировать* его. Flash позволяет тестировать как отдельные сцены фильма, так и фильм в целом. Если тестирование прошло успешно, не забывайте сохранить результат своей работы на диске (*в отличие от того же Word, Flash не поддерживает функцию автоматического сохранения*).

6.3 Создание и редактирование символов

Символ – это объект (элемент фильма), который включен в библиотеку фильма и может быть неоднократно использован в этом же или в другом фильме.

Экземпляр символа – это его копия, помещенная на стол или включенная в состав другого символа. Экземпляры могут достаточно сильно отличаться от символа-оригинала цветом, размером и даже типом. При этом вносимые в экземпляр изменения не влияют на оригинал. Вместе с тем, любые изменения оригинала приводят к соответствующим изменениям всех его экземпляров, независимо от их размещения.

Преимущество символов по сравнению с обычными объектами состоит в том, что их применение существенно ускоряет процесс разработки фильма и уменьшает его размер. Кроме того, только на основе символов могут создаваться

интерактивные элементы фильма.

Во Flash существует три основных типа символов: графические, кнопки и клипы.

Графический символ (graphic symbol) используется в фильме в качестве статического или анимированного изображения. Его поведение описывается с помощью временной диаграммы основного фильма. Графическому символу не могут быть назначены интерактивные средства управления и звуковое сопровождение.

Символ-кнопка (button symbol) предназначен для включения в фильм интерактивных кнопок, реагирующих на действия пользователя и управляющих воспроизведением фильма. С этой целью символу-кнопке могут быть назначены события (events), на которые она должна реагировать, и действия (Actions), которые инициализируются этими событиями. Кроме того, с кнопкой может связано звуковое сопровождение.

Символ-клип (movie clip symbol) служит для включения в библиотеку фильма повторно используемых анимаций. Каждый клип имеет свою собственную временную диаграмму, которая воспроизводится независимо от временной диаграммы основного фильма. Клип может содержать собственные интерактивные элементы управления, а также клипы более низкого уровня вложенности. С другой стороны, и кнопка может содержать клип, позволяющий ее «оживить».

Библиотека символов позволяет автору использовать в новом фильме символы, созданные ранее либо им самим, либо разработчиками Flash. Для упрощения выполнения подобных операций, а также для выполнения некоторых других действий с символами в Flash MX предусмотрено четыре типа библиотек:

- библиотека фильма (Library);
- общая библиотека (Common Library);

- постоянная библиотека (Permanent library);
- разделяемая библиотека (Shared Library).

При вставке в фильм символа из любой «чужой» библиотеки он автоматически помещается в библиотеку фильма. Кроме того, с точки зрения пользовательского интерфейса работа с различными библиотеками реализована во Flash одинаково.

Для работы с библиотекой предназначено специальное диалоговое окно, формат которого показан на рис. 25.

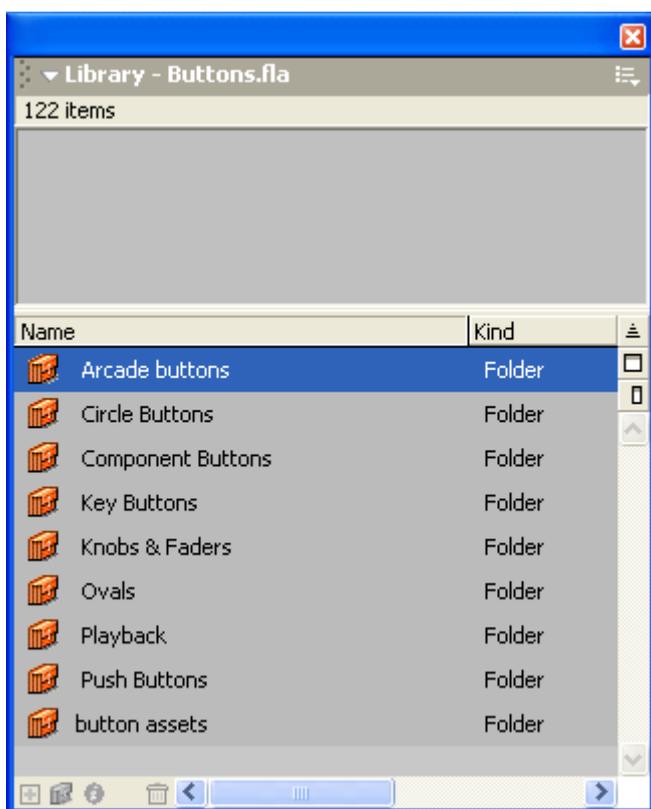


Рис. 25. Формат окна Открыть символа

Окно содержит следующие элементы интерфейса:

- **счетчик объектов библиотеки** (*Counter items*) — это текстовая строка, в которой отображается число объектов, входящих в состав библиотеки; если некоторые символы библиотеки объединены в папку, то они также включаются в общее число объектов, при этом сама папка не учитывается; если же в библиотеке имеется пустая папка, то она учитывается как самостоятельный объект;

- **область просмотра**, предназначенная для визуального представления символа, выбранного в расположенном ниже списке; если символ соответствует звуковому элементу фильма, то в качестве его визуального представления отображается амплитудно-частотная характеристика сигнала (рис. 26); если символ является анимированным изображением, клипом или звуком, то в верхнем правом углу области просмотра появляются две кнопки, которые позволяют запустить/остановить воспроизведение;

- **раскрывающееся меню библиотеки** (*Options*); чтобы его открыть, следует щелкнуть на кнопке меню; содержащиеся в нем команды будут рассмотрены ниже;

- **список объектов библиотеки**, который реализован в виде таблицы; она содержит следующие данные: имя объекта — *Name*, его тип - *Kind*, число включений в фильм - *Use Count*, признак связи с разделяемой библиотекой — *Linking*, дата последнего изменения — *Date Modified* (последние три столбца на рисунке не показаны);

- **кнопка упорядочивания списка** (*Sort Order*) позволяет поменять текущее расположение объектов в списке на противоположное.

Кнопки, расположенные в нижней части окна библиотеки, обеспечивают:

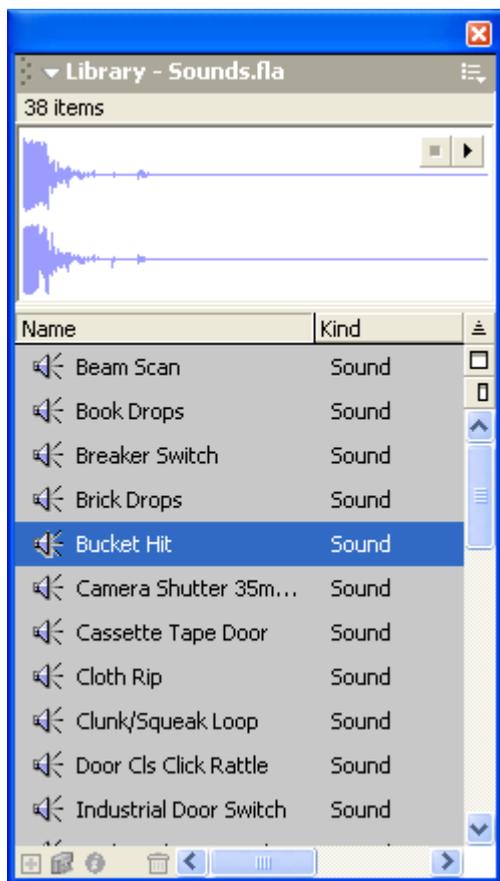


Рис. 26. Визуальное представление звукового символа

- добавление символа в библиотеку (кнопка *New Symbol*); чтобы добавить символ в папку, требуется ее предварительно выделить;
- добавление в библиотеку новой папки (кнопка *New Folder*);
- открытие дополнительного диалогового окна, помощью которого можно просмотреть или изменить свойства выбранного символа (*Properties*);

- удаление выбранного объекта (*Delete*); щелчок на этой кнопке приводит к открытию окна, в котором требуется подтвердить намерение, поскольку восстановить удаленный объект невозможно.

Раскрывающееся меню библиотеки содержит команды, позволяющие изменять параметры как отдельных символов, так и библиотеки в целом. Всего таких команд более двух десятков. Поскольку доступность той или иной команды зависит от текущей ситуации работы с библиотекой, то зачастую вместо этого меню удобнее использовать контекстные меню символов библиотеки. Чтобы открыть такое меню, необходимо предварительно выбрать символ в списке, щелкнув на нем левой кнопкой мыши.

Из команд меню библиотеки в данном подразделе рассмотрены только те, которые не зависят от типа библиотеки и вида выбранного символа. К таким командам относятся (некоторые из них показаны на рис. 27):

- *New Symbol* (Создать символ) – вызов диалогового окна для создания нового символа;

- *New Folder* (Создать папку) — в список добавляется новая строка, в которой вы можете изменить имя созданной папки (рис. 10.4); чтобы поместить символ в папку, достаточно перетащить его мышью на строку папки;

- *New Font* (Создать шрифт) — вызов диалогового окна для создания шрифта-символа;

- *New Video* (Создать видео) — в список добавляется новый «пустой» символ видео-клипа; чтобы сопоставить ему реальный объект, необходимо дважды щелкнуть мышью на значке нового символа и с помощью открывшегося диалогового окна (рис. 10.5) импортировать файл с видео-клипом;

- *Rename* (Переименовать) — имя выбранного в списке символа становится доступным для редактирования;

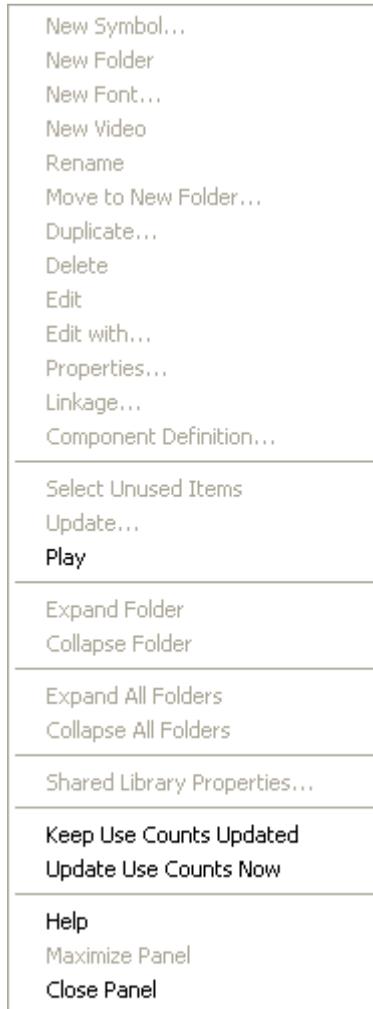


Рис. 27. Фрагмент раскрывающегося меню библиотеки

- *Move to New Folder* (Переместить в новую папку) — вызов диалогового окна для ввода имени папки, куда следует переместить выбранный символ;
- *Duplicate* (Дублировать) — вызов диалогового окна

для создания копии выбранного символа; по сути команда аналогична команде создания нового символа, поскольку позволяет редактировать все свойства символа;

- *Delete* (Удалить) — удаление символа из библиотеки; в дополнительном диалоговом окне можно указать, следует ли удалять также экземпляры символа;

- *Edit* (Правка) — включение режима редактирования символа;

- *Edit with* (Редактировать с помощью...) — вызов внешнего приложения, ассоциированного с данным символом (например, внешнего графического редактора);

- *Properties* (Свойства) — вызов диалогового окна для изменения параметров выбранного символа;

- *Linkage* (Связывание) — вызов диалогового окна для установки параметров экспорта/импорта выбранного символа; подробнее процедура связывания описана в подразделе «Разделяемая библиотека»;

- *Component Definition* (Определение компонента) — вызов диалогового окна для создания и установки параметров так называемого компонента; компонент — это специальный вид символа-клипа, который используется при создании интерактивных элементов фильма; подробнее о них рассказано в одиннадцатой главе;

- *Select Unused Items* (Выбрать не используемые) — символы библиотеки, не использованные ни в одном фильме, визуально выделяются в списке;

- *Keep Use Counts Updated* (Сохранять измененные показания счетчика) — данная команда работает как флажок: если в меню возле ее имени стоит «птичка», то ведется сквозной подсчет числа созданных экземпляров выбранного символа за все время его существования; если птичку убрать, то значение счетчика будет обнуляться при каждом открытии окна библиотеки;

- *Update Use Counts Now* (Восстановить показания

счетчика) — если вы сначала отказались от сквозного подсчета, а потом передумали, то эта команда позволит восстановить показания счетчика.

6.4 Преобразование в символ существующего объекта

Символы могут создаваться двумя альтернативными способами: либо сначала создается некоторый объект, который затем преобразуется в символ, либо создается «пустая» заготовка символа, которая наполняется соответствующим содержимым.

Чтобы преобразовать существующий объект (или несколько объектов) в символ, необходимо:

1. Выбрать на столе объект (объекты), подлежащие преобразованию в символ.

2. В меню *Insert* (Вставить) выбрать команду *Convert to Symbol...* (Преобразовать в символ).

3. В открывшемся диалоговом окне *Convert to Symbol* указать параметры символа (рис. 28).

4. Щелкнуть кнопку *OK*; созданный символ будет автоматически добавлен в библиотеку фильма.

С помощью диалогового окна *Convert to Symbol* могут быть установлены следующие параметры символа:

- *Name* (Имя) — имя символа, под которым он будет включен в библиотеку фильма;

- *Behavior* (Поведение) — тип символа (клип, кнопка или графика);

- *Registration* (Регистрация) — положение регистрационной точки символа; данное поле является интерактивным: чтобы указать положение точки, следует щелкнуть мышью на соответствующем маркере (регистрационная точка отображается черным цветом, пустые маркеры — белым).



Рис. 28. Диалоговое окно установки параметров символа

Кнопка *Advanced* (Дополнительно) обеспечивает переход в расширенный формат окна *Convert to Symbol*. В этом случае оно содержит две дополнительные группы элементов, которые применяются при связывании создаваемого символа с разделяемой библиотекой. Расширенный формат окна был приведен на рис. 10.13. Чтобы вернуться в сокращенный (основной) формат, следует щелкнуть на кнопке *Basic* (Основной), которая заменяет кнопку *Advanced*.

С помощью окна *Convert to Symbol* можно не только преобразовать в символ «обычный» объект, но также изменить тип символа. Для этого достаточно изменить положение переключателя *Behavior*. Правда, при переходе от менее сложного типа к более сложному такое преобразование будет выполнено формально, и дальнейшее редактирование потребует выполнения ручной. Например, если вы захотите

«превратить» в кнопку графический символ, то одного щелчка в позиции *Button* будет недостаточно: потребуется еще описать поведение вновь созданной кнопки. Вместе с тем, вы легко можете сделать из кнопки графический символ. При этом, разумеется, кнопка потеряет свою интерактивность.

Чтобы создать новый пустой символ:

1. Убедитесь, что на столе отсутствуют выбранные объекты, и затем выполните одно из следующих действий:

- в меню *Insert* выберите команду *New Symbol* (Создать символ);
- в нижнем левом углу окна библиотеки (если оно открыто) щелкните кнопку *New Symbol*;
- выберите команду с тем же именем из раскрывающегося меню окна библиотеки.

2. В открывшемся диалоговом окне *Symbol Properties* (см. рис. 28) укажите имя и тип символа.

При создании нового пустого символа Flash автоматически переходит в режим редактирования символа. Об этом свидетельствует появление имени символа в панели инструментов рабочей области, а также значка регистрационной точки символа в виде тонкого перекрестия.

Пустой символ, несмотря на отсутствие содержимого, автоматически добавляется в библиотеку фильма.

6.5 Редактирование символов и экземпляров

Повторим еще раз, что редактирование символа приводит к модификации всех экземпляров этого символа, причем во всех фильмах, в которых имеется ссылка на него. Поэтому прежде, чем приступить к редактированию символа, подумайте, не лучше ли изменить несколько его экземпляров.

Во Flash предусмотрено три варианта редактирования символа:

- в режиме редактирования символа (именно такой способ используется при создании нового пустого символа);
- в отдельном окне;
- в контексте сцены, то есть непосредственно на столе.

При использовании первого или второго варианта формат окна рабочей области изменяется таким образом, что на столе виден только редактируемый символ.

При выборе третьего варианта остальные объекты остаются видны, но отображаются более блеклыми по сравнению с обычным состоянием (рис. 29).

Чтобы перейти к редактированию символа, достаточно щелкнуть на нем правой кнопкой мыши и выбрать в контекстном меню одну из трех команд:

- *Edit* (Правка) — включение режима редактирования символа;
- *Edit in Place* (Правка по месту) — редактирование символа в контексте сцены;
- *Edit in New Window* (Правка в новом окне) — редактирование символа в новом окне.

В любом случае над панелью временной диаграммы появляется имя редактируемого символа, а на его изображении — отметка точки привязки в виде крестика; кроме того, если символ является кнопкой или клипом, временная диаграмма фильма заменяется временной диаграммой символа.

Для редактирования графического символа и кнопки могут применяться инструменты рисования и закрашивания (предварительно символ требуется выделить с помощью инструментов *Arrow*, *Subselect*, *Free Transform* или *Lasso*), а также средства трансформации изображения (поворот, наклон, отражение и т. д.).

Возможности по редактированию клипа зависят от того, каким образом создавалась анимация; соответствующие вопросы рассматриваются в главе «Анимация».

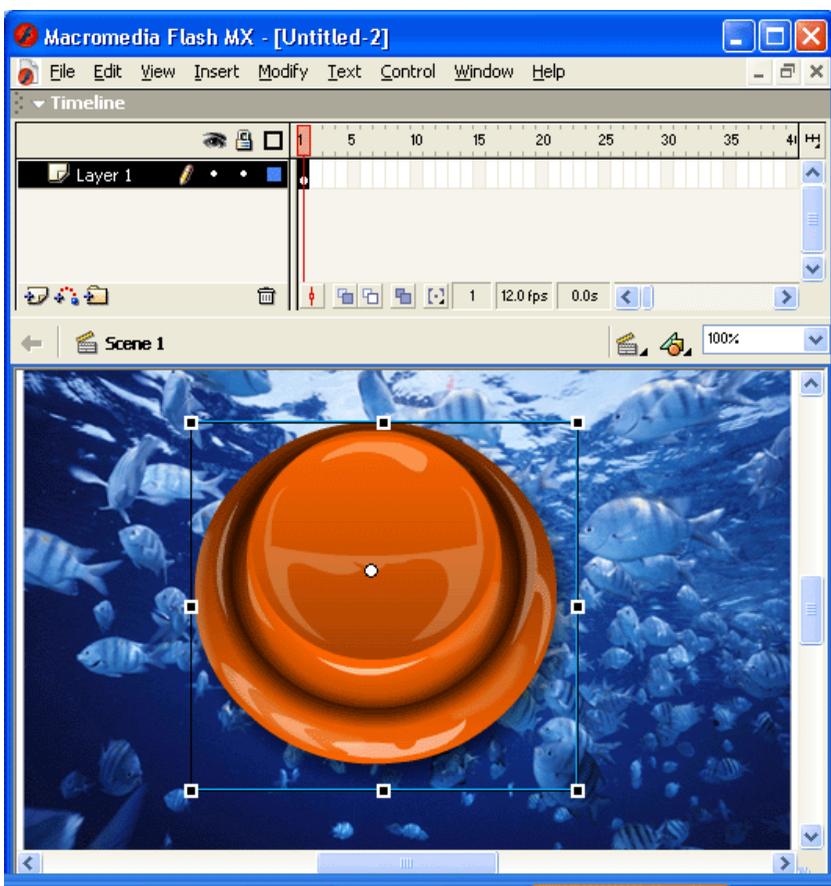


Рис. 29. Формат окна Flash при редактировании символа в контексте сцены

Если при редактировании символа открыть библиотеку, содержащую этот символ, то можно видеть, как вносимые изменения отображаются в окне библиотеки.

После завершения работы с символом следует выйти из режима его редактирования, щелкнув на имени сцены или (при редактировании символа в отдельном окне) выбрав

сцену в раскрывающемся меню *Edit Scene*.

Редактирование экземпляра символа. Каждый экземпляр имеет собственные атрибуты, которые могут редактироваться без изменения соответствующих атрибутов символа. В частности, вы можете изменять цвет и прозрачность экземпляра, переопределять его тип (например, преобразовать графический символ в кнопку); вы можете также наклонять, вращать или масштабировать экземпляр без того, чтобы воздействовать на символ.

Атрибуты экземпляра сохраняются вместе с ним. Если вы редактируете символ или связываете экземпляр с другим символом, все внесенные изменения атрибутов экземпляра остаются действительными.

Чтобы редактировать атрибуты экземпляра, необходимо использовать элементы управления, размещенные на панели инспектора свойств экземпляра. Формат панели зависит от типа экземпляра (графика, кнопка или клип), а также от того, какое именно изменение вы собираетесь внести в параметры экземпляра. Общими для всех форматов являются (рис. 30):

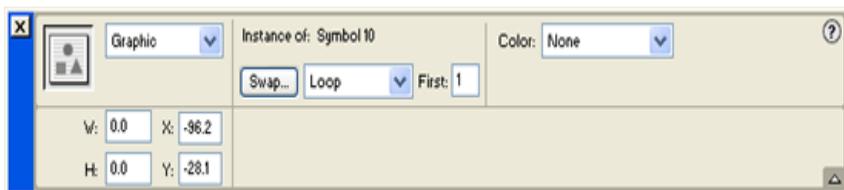


Рис. 30. Формат инспектора свойств экземпляра графического символа

- раскрывающийся список *Symbol Behavior*, который позволяет установить для данного экземпляра тот или иной тип символа; список содержит соответствующие варианты (*Movie Clip*, *Button* и *Graphic*);
- текстовая область *Instance of*, в ней отображается имя

символа, с которым связан данный экземпляр;

- кнопка *Swap Symbol* (Сменить символ); щелчок на кнопке открывает диалоговое окно, которое позволяет связать редактируемый экземпляр с другим символом фильма;

- раскрывающийся список *Color Styles* (Параметры цвета), который позволяет выбрать параметр цвета, подлежащий индивидуальной настройке для данного экземпляра; от выбранного в списке пункта зависит формат связанной со списком группы элементов управления; возможные варианты рассмотрены ниже.

Итак, список *Color Styles* и связанные с ним элементы управления обеспечивают изменение следующих визуальных атрибутов экземпляра:

- яркости (*Brightness*);
- оттенка цвета (*Tint*);
- прозрачности (*Alpha*).

Каждому из них соответствует одноименный пункт списка. Кроме того, для создания более сложных цветовых эффектов может быть использован еще один пункт — *Advanced* (Продвинутый). В этом случае вы можете создавать новые цветовые оттенки, комбинируя их с прозрачностью экземпляра.

При выборе пункта *Brightness* на панели инспектора свойств экземпляра появляется единственный дополнительный элемент *Brightness Amount* (Регулятор яркости), выполненный в традиционном для Macromedia стиле (рис. 31).

Яркость устанавливается как относительная величина (в процентах) и может изменяться в диапазоне от -100% (соответствует черному цвету) до 100% (соответствует белому цвету).

При выборе пункта *Tint* на панели инспектора свойств экземпляра отображается целая группа дополнительных элементов управления (рис. 32).

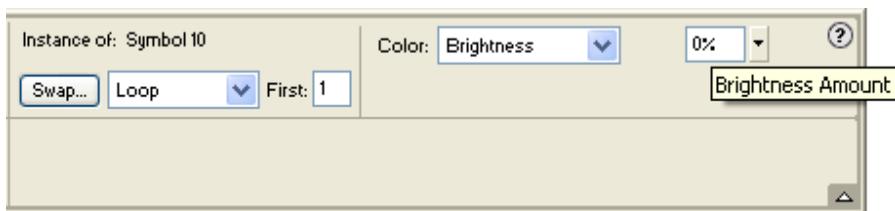


Рис. 31. Изменение яркости экземпляра

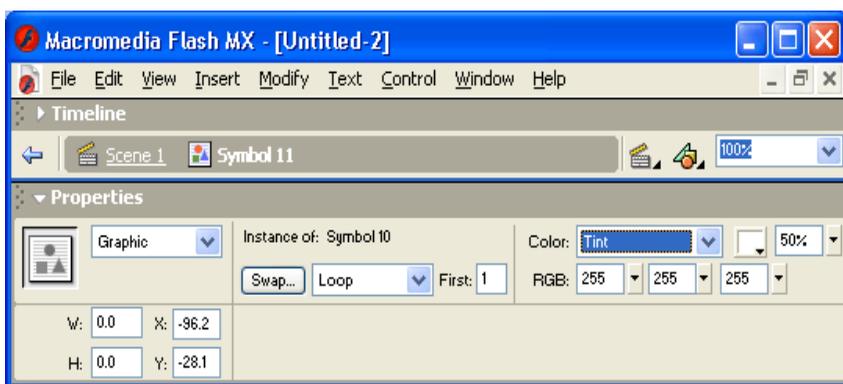


Рис. 32. Изменение оттенка цвета экземпляра

Это обусловлено тем, что оттенок цвета определяется соотношением красного, синего и зеленого цветов (в системе RGB — Red, Green, Blue) при фиксированной насыщенности цвета. Значение кода для каждого из этих цветов устанавливается с помощью соответствующего регулятора. Значение может лежать в диапазоне от 0 до 256. Полученный оттенок тут же отображается на кнопке палитры. Насыщенность указывается в процентах (в диапазоне от 0 до 100), также с помощью регулятора.

Формат панели инспектора свойств экземпляра при установке прозрачности экземпляра (*Alpha*) аналогичен тому, который используется при установке яркости (рис. 33).

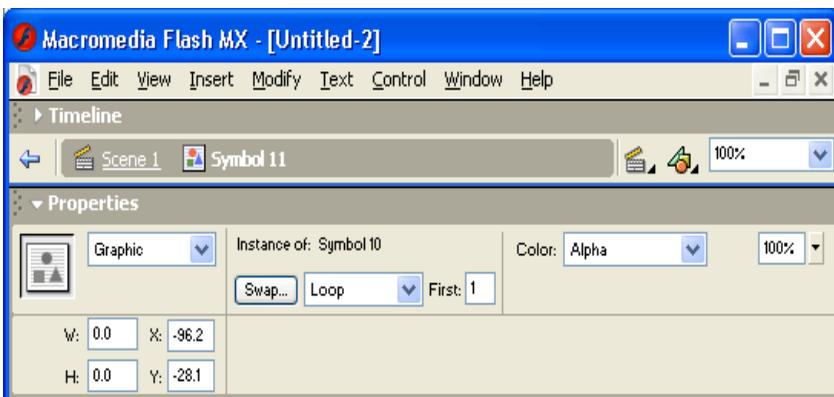


Рис. 33. Изменение прозрачности экземпляра

Прозрачность определяет степень «видимости» экземпляра на столе. Измеряется также в процентах и может лежать в диапазоне от 0% (абсолютная прозрачность) до 100%. Посредством динамического изменения этого атрибута (с помощью несложного сценария на ActionScript) можно создать эффект исчезновения либо, наоборот, неожиданного появления того или иного персонажа.

При выборе пункта *Advanced* на панели инспектора свойств появляется кнопка *Settings* (Установки), щелчок на которой открывает дополнительное диалоговое окно (рис. 34). С его помощью вы можете «за один заход» установить все перечисленные выше параметры цвета экземпляра.

Для экземпляра графического символа с помощью инспектора свойств могут быть изменены атрибуты, связанные с режимом воспроизведения встроенной в символ анимации (см. рис. 30):

- цикличность воспроизведения; она устанавливается с помощью раскрывающегося списка *Options for graphics* (Настройки для графического символа), который содержит три варианта:

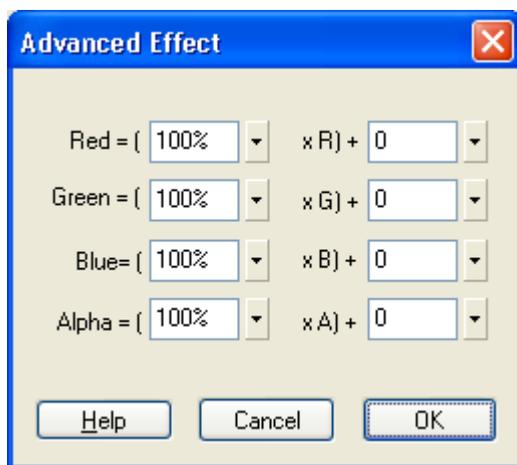


Рис. 34. Диалоговое окно Advanced Effect

- *Loop* (Зациклить);
- *Play Once* (Однократное воспроизведение);
- *Single Frame* (Одиночный кадр);
- номер начального кадра, который указывается в текстовом поле *First* (Начальный кадр).

6.6 Работа с текстом

В любой Flash-фильм может быть добавлен текст. Подобно работе в обычных текстовых редакторах, для текста может быть установлен размер, шрифт, стиль, интервал, цвет и способ выравнивания. Вы можете трансформировать шрифт подобно другим объектам — поворачивать, масштабировать, наклонять. При этом сохраняется возможность редактирования его символов.

На основе фрагмента текста может быть создана гиперссылка.

Flash-фильм может содержать динамические текстовые

поля, а также поля, редактируемые пользователем. Редактируемые поля в Flash-фильме имеют то же предназначение, что и текстовые поля, помещаемые в форму на Web-странице: с их помощью от пользователя может быть получена та или иная информация, подлежащая обработке на сервере или клиентским сценарием. И, наконец, в Flash MX вы можете создавать прокручиваемые многострочные текстовые области.

При необходимости текст может быть конвертирован в графический объект, и тогда с его символами разрешается работать как с отдельными графическими фигурами.

Чтобы добавить текстовое поле в Flash-фильм, следует выполнить следующие действия:

1. Включить инструмент *Text Tool* на панели *Tools*.
2. Щелкнуть мышью в той позиции на столе, куда требуется поместить текстовое поле; в результате в этой позиции появится текстовый курсор, окруженный выделяющей рамкой.
3. Набрать текст.

Обратите внимание, что в правом углу выделяющей рамки (верхнем или нижнем) имеется маркер, который определяет тип текстового поля:

- прямоугольный маркер в правом верхнем углу соответствует текстовому полю фиксированной ширины (*Fixed text block*); в таком поле выполняется автоматический перенос на следующую строку, если очередной символ не умещается на текущей строке; ширина поля устанавливается пользователем посредством перетаскивания маркера; после публикации фильма на Web-сервере содержимое такого поля не может быть изменено с помощью сценария или читателем Web-страницы;
- круглый маркер в правом верхнем углу соответствует расширяемому текстовому полю (*Expanded text block*); это однострочное текстовое поле, ширина которого

автоматически увеличивается при вводе текста; данный вариант используется по умолчанию; содержимое такого поля также не может быть изменено с помощью сценария или читателем Web-страницы;

- белый прямоугольный маркер в правом нижнем углу соответствует текстовому полю, предназначенному для ввода динамически изменяемого содержимого — *Dynamic text block* или *Input text block*; содержимое поля первого типа может изменяться с помощью сценария, а второго - читателем Web-страницы (например, при вводе данных в форму);

- черный прямоугольный маркер в правом нижнем углу указывает на текстовое поле, содержимого которого может перемещаться (прокручиваться) внутри текстовой области с помощью вертикальной и/или горизонтальной полосы прокрутки.

Чтобы изменить атрибуты текстового блока, слова (фрагмента текста) или отдельного символа, необходимо:

1. Включить инструмент *Text*.
2. Выбрать (выделить) мышью соответствующий фрагмент текста.
3. В меню *Text* выбрать соответствующую команду, либо воспользоваться элементами управления, имеющимися на панели инспектора свойств текста.

Все команды меню *Text* разделены на три группы (рис. 35):

- в первую входят «традиционные» команды работы с текстом (*Font, Size, Style*), которые, собственно, представляют собой каскадные меню, содержащие возможные значения соответствующих атрибутов текста;

- вторую группу образуют две также достаточно распространенные команды — *Align* (Выравнивание) и *Tracking* (Трекинг);

- в третью группу включена единственная команда - - *Scrollable* (Перемещаемый), обеспечивающая установку

соответствующего свойства текстового поля.

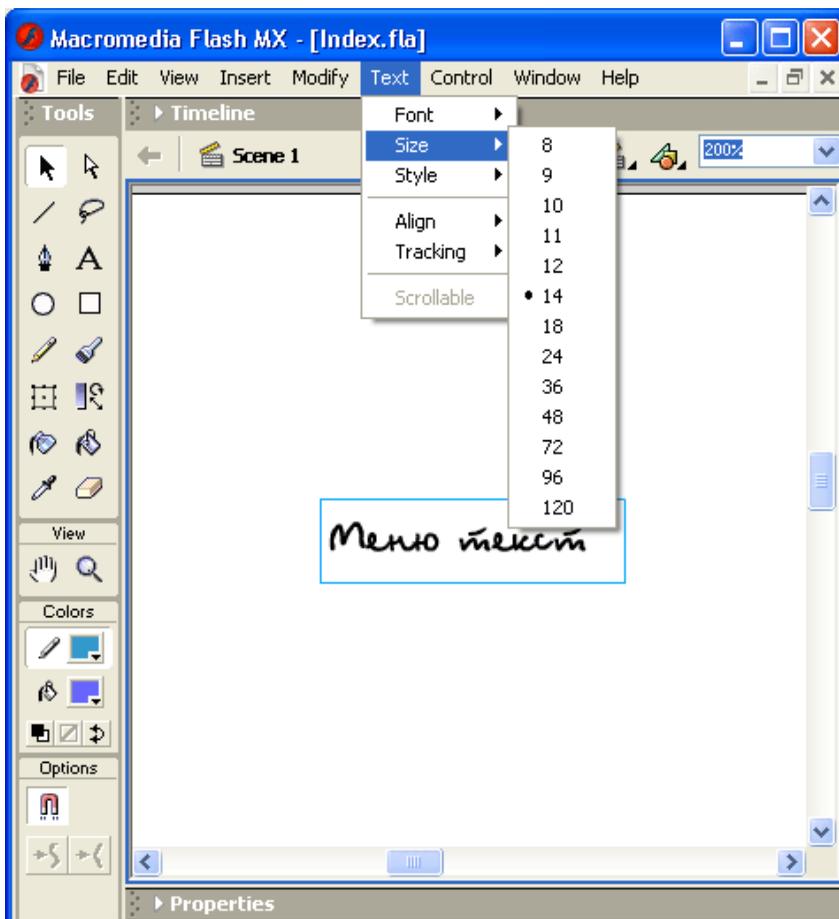


Рис. 35. Состав команд меню Text

Более удобным средством для изменения параметров текста является инспектор свойств. Его формат зависит от типа текстового поля.

Статический текст (Static Text) — это текстовое содержимое фильма, которое не может быть изменено после

публикации фильма.

Формат инспектора свойств статического текста показан на рис. 36.

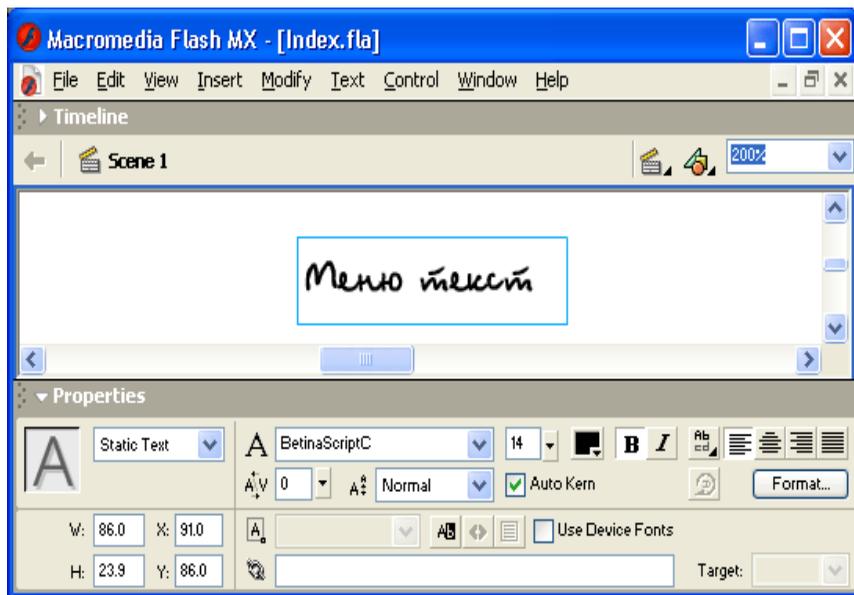


Рис. 36. Инспектор свойств статического текста

Из представленных на панели элементов управления дополнительного пояснения требуют, вероятно, следующие:

- текстовое поле *URL*; выделив на столе текстовый блок и указав в этом поле полный Web-адрес (URL - Universal Resources Locator) связанного ресурса, вы получите самую настоящую гиперссылку; правда, в отличие от других визуальных редакторов, она будет представлена не подчеркнутым шрифтом, а шрифтом с штриховой линией. Используемый во Flash механизм создания гиперссылок имеет два недостатка: во-первых, URL приходится вводить вручную, и, во-вторых, Flash не проверяет корректность URL

даже для локальных файлов;

- раскрывающийся список *Target* (Целевой); он позволяет указать, в какое окно браузера следует загрузить ресурс, вызванный по указанному URL; список становится доступен при вводе в поле *URL* хотя бы одного символа; список содержит следующие варианты (они аналогичны стандартным значениям атрибута *TARGET* языка HTML):

- *_blank* - вызываемый документ всегда загружается в новое, неименованное окно;

- *_self* - вызываемый документ всегда загружается в то же окно, в котором была инициирована ссылка;

- *_parent* - вызываемый документ загружается в родительское окно (то есть в окно более высокого уровня иерархии); если данное окно не имеет «родителя», или само является таковым, то результат аналогичен применению значения *_self*;

- *_top* - вызываемый документ загружается в окно самого верхнего уровня; (если данное окно само является таковым, то результат аналогичен применению значения *_self*);

- кнопка *Selectable* (Выделяемый); она используется как флажок: щелчок на кнопке позволяет установить или отменить для текста соответствующее свойство; при этом следует иметь в виду, что возможность выделения фрагмента текста не означает возможность его редактирования; выбранный фрагмент может быть скопирован, например, в буфер обмена;

- кнопка *Change Direction of Text* (Изменить направление текста) позволяет открыть меню, с помощью которого вы можете выбрать ориентацию надписи: горизонтальную или вертикальную; меню содержит три команды (рис. 37):

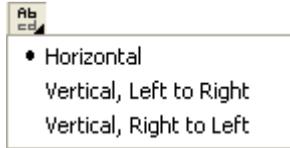


Рис. 37. Кнопка *Rotation* (Поворот) для изменения расположения символов в надписи

- *Horizontal* (Горизонтальный);
- *Vertical, Left to Right* (Вертикальный, слева направо);
- *Vertical, Right to Left* (Вертикальный, справа налево);

При установке вертикальной ориентации текста изменяется состав кнопок форматирования и, кроме того, становится доступна кнопка *Rotation*:

- кнопка *Rotation* (Поворот) играет роль флажка и позволяет изменять расположение символов в надписи (рис. 37);

- флажок *Use Device Font* (Использовать шрифт устройства) – если флажок установлен, то при воспроизведении фильма Flash-плеер использует физические шрифты, установленные для браузера читателя; это может уменьшить размер файла фильма и повысить удобочитаемость для мелкого шрифта.

Возле правой границы панели инспектора свойств находится кнопка *Format* (Формат), щелчок на которой открывает, дополнительное диалоговое окно, позволяющее установить параметры форматирования текстового поля (рис. 38).

Это окно содержит четыре текстовых поля, совмещенных с ползунковыми регуляторами «в стиле Macromedia»:

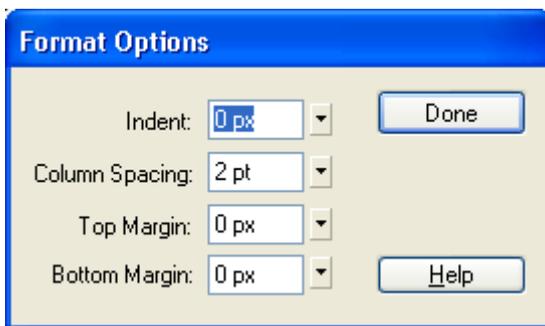


Рис. 38. Диалоговое окно Format Option

- *Indent* (Отступ) — величина абзацного отступа для новой строки (в пикселах);
- *Line Spacing* (Межстрочный интервал) — расстояние между соседними строками (в пунктах); для вертикального текста этот параметр задает расстояние между столбцами текста;
- *Left Margin* (Левое поле) — расстояние между левой границей текстового поля и первым символом строки;
- *Right Margin* (Правое поле) — расстояние между правой границей текстового поля и последним символом строки.

Диалоговое окно *Format Option* идентично для всех типов текстовых полей.

Динамический (или изменяемый текст) Dynamic Text — это такое текстовое поле, содержимое которого может изменяться с помощью сценария на языке ActionScript (то есть в процессе воспроизведения фильма).

Возможность работы с динамическим текстовым полем из сценария на ActionScript обусловлена тем, что каждое такое поле представляет собой экземпляр специального объекта языка ActionScript, который называется *TextField*.

Динамическому текстовому полю может быть присвоено уникальное имя (идентификатор), по которому

можно обращаться к полю, чтобы установить или изменить содержимое и формат этого поля. Например, вы можете динамически изменять для такого поля символьную кодировку.

Для установки атрибутов динамического текста панель инспектора свойств содержит следующие элементы (рис. 39):

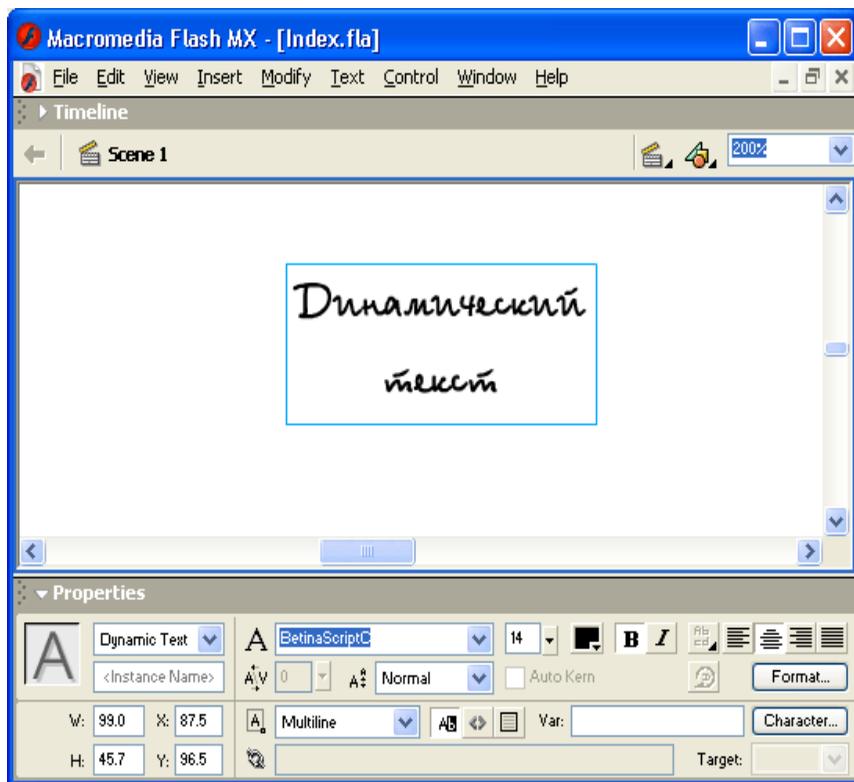


Рис. 39. Формат инспектора свойств для динамического текста

- текстовое поле *Instance Name* (Имя образца), в котором указывается имя текстового поля; несмотря на то,

что имя поля выводится (непосредственно в нем) символами серого цвета, которые обычно обозначают в Windows-приложениях заблокированный элемент интерфейса, в данном случае ввод разрешен;

- раскрывающийся список *Line type* (Тип строки) форматов текстового поля:

- *Single Line* (Однострочное);

- *Multiline* (Многострочное);

- *Multiline no wrap* (Многострочное без переносов);

- кнопка *Render text as HTML*; если она нажата, то заданные параметры форматирования текста (такие как размер, стиль, использование в качестве гиперссылки) при публикации фильма будут сохранены в виде соответствующих HTML-тэгов;

- кнопка *Show Border* (Показать рамку); если она нажата, то текстовое поле будет окружено рамкой;

- флажок *Selectable* (Выбираемый); его назначение аналогично рассмотренному для статического текста;

- текстовое поле *Var* (от Variable — Переменная) предназначено для ввода имени переменной, связанной с создаваемым текстовым полем; об использовании переменных для управления элементами фильма будет рассказано в подразделе «Создание интерактивных элементов фильма»;

- кнопка *Characters* (Символы), щелчок на которой открывает дополнительное диалоговое окно *Character Options*, позволяющее установить параметры использования встроенного шрифта, используемого для текстового поля; окно содержит три переключателя (рис. 40):

- *No Characters* (Никакие символы) — информация об используемом шрифте не экспортируется в Flash-фильм при его публикации;

- *All Characters* (Все символы) – в Flash-фильм включается информация о всех символах шрифта;

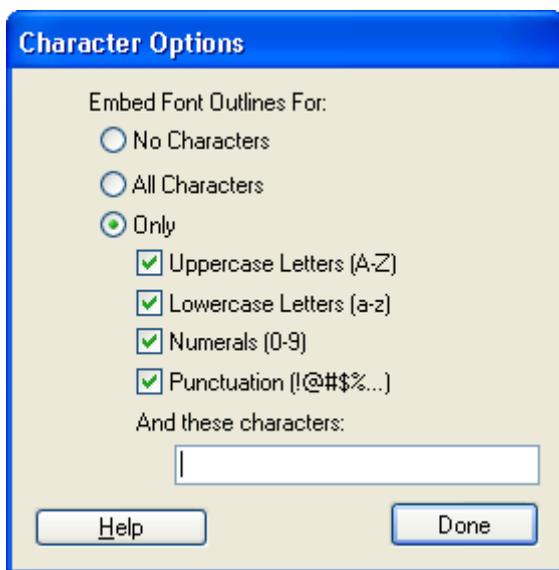


Рис. 40. Диалоговое окно Character Options

○ *Only* (Избранные) — в Flash-фильм включается информация о только о тех символах шрифта, которые указаны с помощью расположенных ниже флажков.

На основе динамического текстового поля может быть создана перемещаемая (прокручиваемая) текстовая область. Пример такой текстовой области показан на рис. 41.

Для создания прокручиваемой текстовой области необходимо выполнить следующие действия:

1. Включите инструмент *Text* и с помощью мыши очертите на столе границы текстовой области.

2. На панели инспектора свойств с помощью раскрывающегося списка *Line type* выберите формат текстового поля; при этом полезно придерживаться следующих правил:

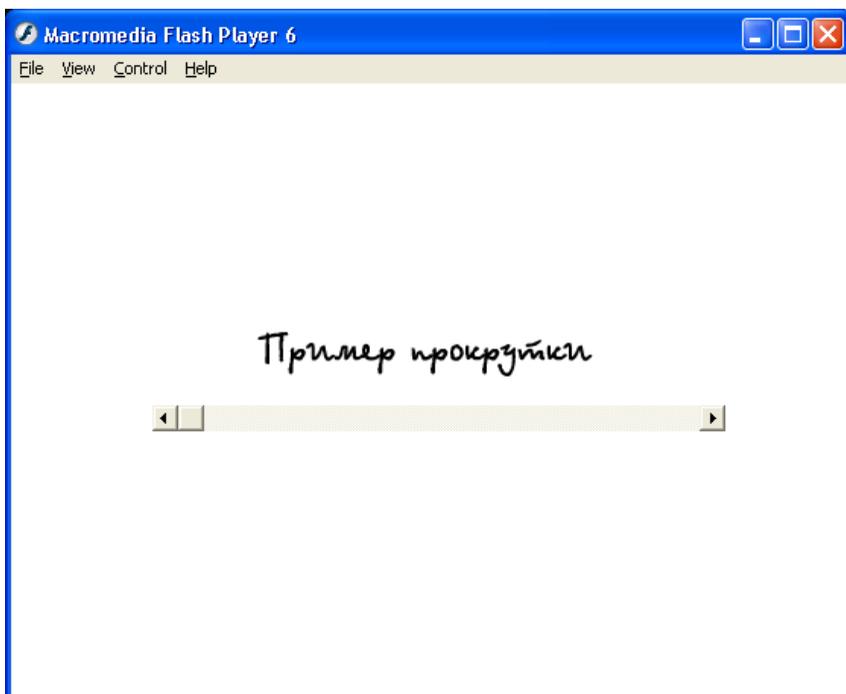


Рис. 41. Пример прокручиваемой текстовой области

- если вы хотите использовать вертикальную полосу прокрутки, то следует выбрать тип *Multiline* или *Multiline no wrap*;
 - если вам требуется поле только с горизонтальной полосой прокрутки, то следует выбрать тип *Single Line*;
 - для создания поля с вертикальной и горизонтальной полосами прокрутки выберите тип *Multiline no wrap*.
3. При необходимости установите требуемые значения других параметров текстового поля (стиль и размер шрифта, цвет и т. д.).
4. Не выключая инструмент *Text*, переместите указатель

мышью за пределы текстовой области и щелкните левой кнопкой; при этом текстовый курсор внутри поля исчезнет, и оно окажется выделенным черной пунктирной рамкой.

5. Вновь переместите указатель мыши внутрь текстовой области и щелкните правой (!) кнопкой, чтобы открыть контекстное меню.

6. В контекстном меню выберите команду *Scrollable* (Прокручиваемый).

7. Убедитесь, что в рабочей области Flash открыта панель *Components* (Компоненты); если панель закрыта, в меню *Window* выберите пункт *Components*; данная панель (рис. 42) предназначена для включения в фильм элементов управления (кнопок, флажков, списков и т. д.); подробнее о работе с панелью *Components* рассказано в главе «Создание интерактивных фильмов».

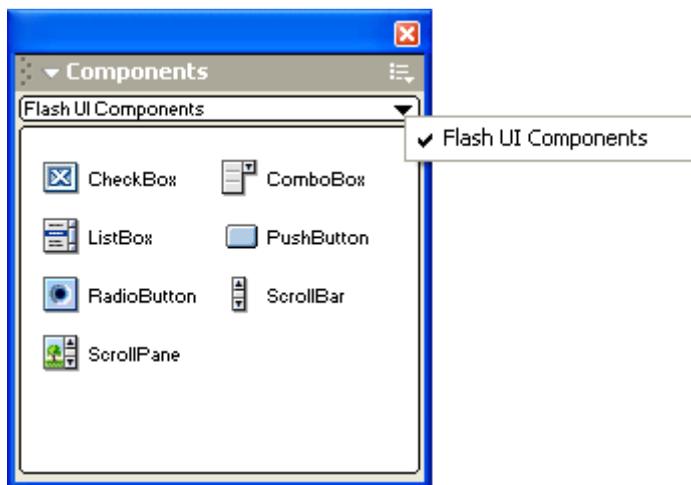


Рис. 42. Панель Components

8. На панели *Components* найдите элемент *ScrollBar* (на рис. 42) и перетащите его мышью в текстовую область; при

этом учитывайте следующее:

- Flash автоматически устанавливает размер полосы прокрутки в соответствии с высотой (для вертикальной) и длиной (для горизонтальной) текстовой области;

- чтобы получить вертикальную полосу прокрутки, следует установить элемент *ScrollBar* вдоль вертикальной границы области;

- чтобы получить горизонтальную полосу прокрутки, следует установить элемент *ScrollBar* поперек горизонтальной границы текстовой области таким образом, чтобы указатель мыши оказался в пределах текстовой области, как показано на рис. 43;

- Flash создает ползунок на полосе прокрутки только в том случае, если он действительно необходим; то есть, например, если текстовая область содержит только одну строку, то ползунок на вертикальной полосе прокрутки будет отсутствовать.

Интерактивность полосы прокрутки, как и других элементов управления, помещенных в Flash-фильм, «проявляется» лишь после его экспорта в формат SWF, а также в режиме тестирования фильма. Чтобы перейти в такой режим, необходимо в меню Control основного окна Flash выбрать пункт Test Movie (Тестирование фильма) или Test Scene (Тестирование сцены).

Если вам в результате перетаскивания элемента *ScrollBar* по какой-либо причине не удалось получить полосу прокрутки желаемого типа, вы можете установить его вручную. Для этого необходимо:

1. Включить инструмент *Arrow* и щелкнуть мышью на той полосе прокрутки, которая вас «не послушалась», чтобы выбрать ее.

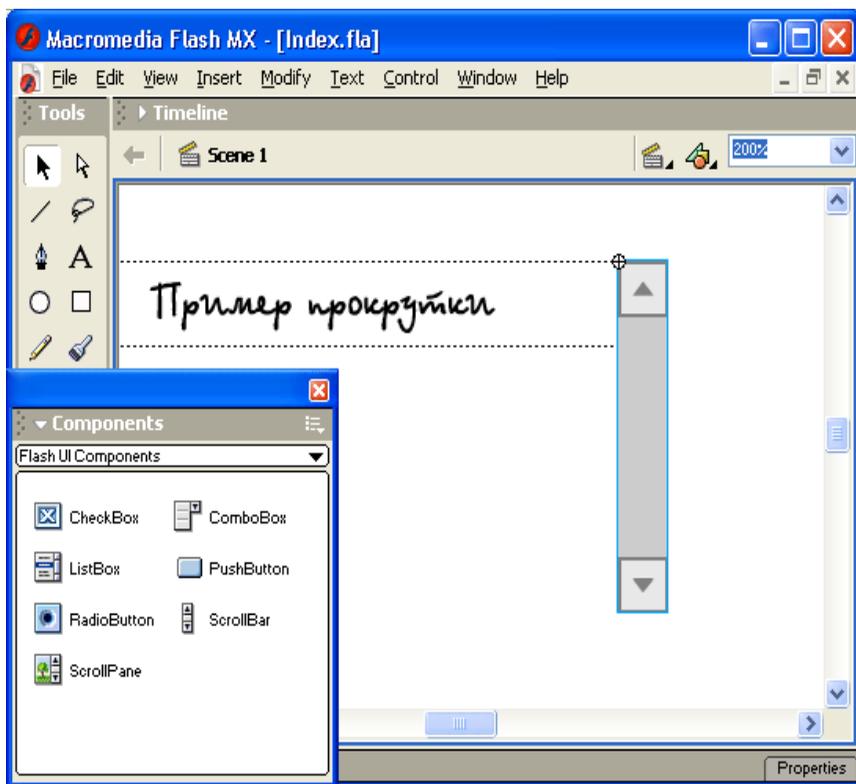


Рис. 43. Создание горизонтальной полосы прокрутки из вертикальной

2. В открывшейся панели инспектора свойств этого элемента (рис. 43) щелкнуть в поле свойств элемента, на строке *Horizontal*; в результате в строке появится окно раскрывающегося списка.

3. Щелкнуть на кнопке списка и в появившемся окне выбрать значение *true* (истина), если вам нужна горизонтальная полоса прокрутки или *false* (ложь), если вы хотите получить вертикальную полосу прокрутки.

После того, как текстовая область будет дополнена

полосами прокрутки, вы можете включить инструмент *Text*, щелкнуть внутри области и ввести необходимый текст.

Наряду со стандартными средствами форматирования текста Flash поддерживает также возможность создания специфических эффектов. К ним, в частности, относятся масштабирование текста, наклон, поворот, зеркальное отражение. После выполнения любой из операций трансформации текст остается текстом, то есть его по-прежнему можно редактировать не как графический объект, а как набор текстовых символов.

6.7 Работа с отдельными объектами

Во Flash термин «объект» используется для обозначения любого элемента фильма, находящегося на столе. Используя инструменты Flash, вы можете перемещать, копировать, удалять, трансформировать объекты, размещать их один за или перед другим «в глубину» экрана, выравнивать их относительно друг друга и группировать. Вы можете также связать объект с некоторым URL (то есть использовать его в качестве гиперссылки). Однако при выполнении манипуляций над объектами следует иметь в виду, что изменения одного объекта могут повлечь изменения другого. Это относится в первую очередь к графике.

Чтобы выполнить над объектом те или манипуляции, его необходимо предварительно **выбрать**. В связи с этим уместно сделать одно достаточно важно пояснение. В англоязычной литературе для обозначения операции выбора объекта используется термин Selection. В русскоязычных изданиях это слово зачастую переводят как **выделение**, что не совсем правильно. Дело в том, что между выбором и выделением (highlight - подсветка) существует принципиальная разница: выбор производит пользователь (например, щелкнув мышью на объекте), а выделение

является ответной реакцией программы на действие пользователя. Для визуального выделения выбранного объекта относительно других обычно (но не всегда) используются специальные средства: инверсный цвет, штриховка и т. д.

Почему мы обратили ваше внимание на различие между выбором и выделением? Объясняется это тем, что во Flash, в отличие от многих других программ, используется несколько различных способов выделения выбранного объекта. Способ выделения зависит от того, каким образом выбран объект, и что вы собираетесь с ним в дальнейшем делать.

Во Flash можно выбрать объект целиком либо только его часть, группу объектов, экземпляр символа, текстовый блок (одно слово или несколько). Для выбора объектов, расположенных в рабочей области, удобнее всего использовать инструменты, размещенные на панели *Tools*

Прежде, чем перейти к описанию особенностей применения различных инструментов, еще раз подчеркнем, что при работе с графическими объектами следует рассматривать их как состоящие из двух компонентов: контура (*Stroke*) и заливки (*Fill*). Свойства этих компонентов (цвет, форма, взаимное расположение) могут изменяться пользователем независимо друг от друга. Можно также удалить один из этих компонентов.

Инструментов выбора пять (рис. 44):

- *Arrow Tool* (Стрелка), с помощью которого можно выбрать на столе произвольную область прямоугольной формы; область может содержать группу объектов, один объект или его часть; инструмент воздействует и на контур объекта, и на заливку; то есть если в область выбора попадает и контур, и заливка, то выбраны будут оба компонента; выбранная область может быть перемещена или отредактирована независимо от остальной части изображения;



Рис. 44. Инструменты для выбора

- *Subselection Tool* (Выбор подобласти), который позволяет выбрать (выделить) объект целиком; если

выбранная область захватывает несколько объектов, то выбранными оказываются все эти объекты; данный инструмент позволяет изменять форму объекта, но выделенный объект не может быть перемещен;

- *Lasso Tool* (Петля), обеспечивающий выбор области произвольной формы; в эту область могут быть включены несколько объектов (или их фрагменты), отдельный объект, или его часть; инструмент воздействует и на контур объекта, и на заливку; выделенная область может быть перемещена или отредактирована независимо от остальной части изображения;

- *Free Transform Tool* (Произвольная трансформация), с помощью которого можно выбрать на столе произвольную область прямоугольной формы; область может содержать группу объектов, один объект или его часть; инструмент воздействует и на контур объекта и на заливку; отличие от инструмента *Arrow* состоит в том, что данный инструмент обладает большими функциональными возможностями по редактированию контура и заливки объекта;

- *Pen Tool* (Перо), который на самом деле относится к инструментам рисования, но может применяться для" выбора контура объекта; подробнее работа с этим инструментом рассмотрена в следующей главе.

Чтобы включить требуемый инструмент, достаточно щелкнуть на соответствующей кнопке. При этом автоматически изменится и содержимое поля *Options* (если для данного инструмента предусмотрена возможность его дополнительной настройки). Дополнительные параметры могут быть установлены, в частности, для инструментов *Arrow*, *Lasso* и *Free Transform*.

Выбор одного объекта или его части. Поскольку анимация во Flash может быть построена на изменении таких свойств объекта, как цвет, форма, прозрачность, то при работе с графическим объектом пользователь может выбрать

либо объект целиком, либо только его контур, либо, наоборот, только внутреннюю часть объекта (заливку).

Инструмент *Arrow*. Чтобы выбрать весь объект, следует включить на панели *Tools* инструмент *Arrow* и затем выполнить одно из двух действий.:

- щелкнуть дважды мышью внутри объекта;
- установить указатель мыши за пределами объекта, и, нажав кнопку мыши, переместить указатель таким образом, чтобы весь объект оказался внутри области, ограниченной прямоугольником (рис. 45).

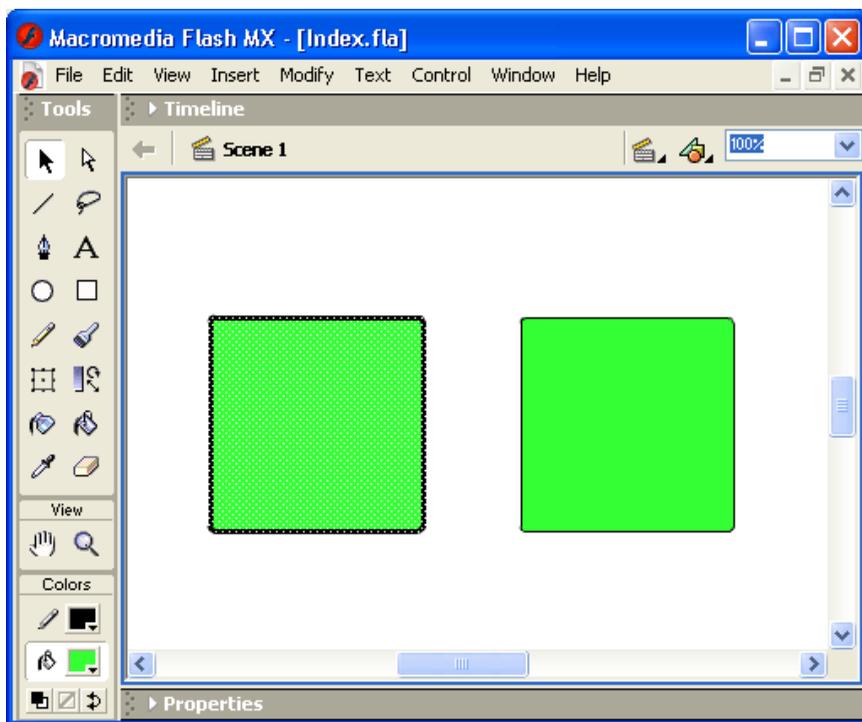


Рис. 45. Выбор объекта с помощью инструмента *Arrow*

Выбранный объект отображается покрытым мелкой

сеткой, а контур – более толстым по сравнению с обычным состоянием (вот оно — выделение).

Для выбора контура объекта следует подвести указатель к границе объекта и щелкнуть (один раз) левой кнопкой мыши. В результате контур будет выделен таким же образом, как и при выборе всего объекта, а внутренняя часть объекта останется не выделенной (рис. 46).

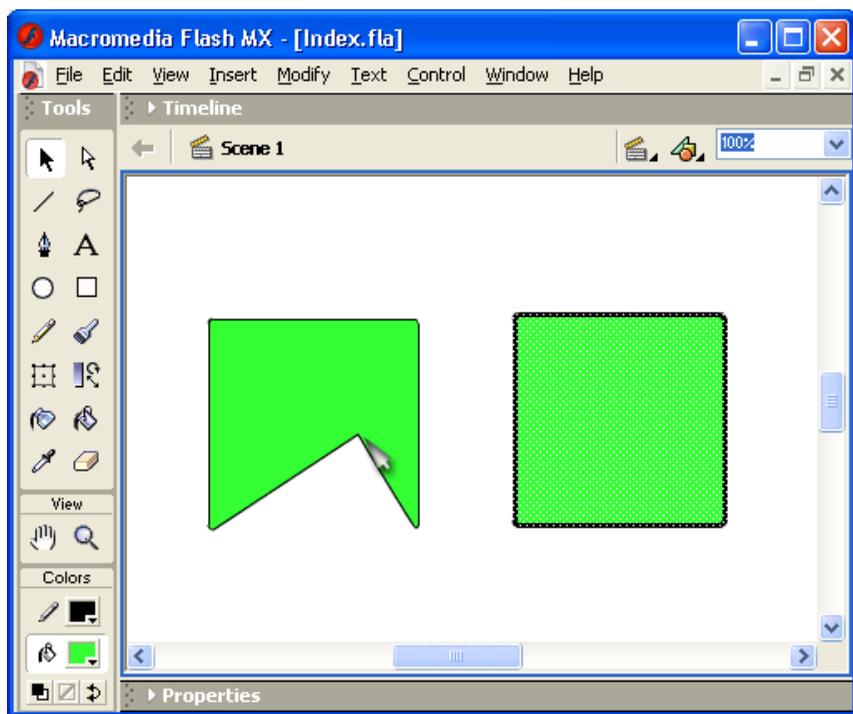


Рис. 46. Выбор контура объекта с помощью инструмента Arrow

О том, что указатель находится в так называемой горячей зоне объекта (то есть в зоне «видимости» объекта), свидетельствует небольшая дуга рядом с указателем. Это,

кстати, свидетельствует и о том, что контур объекта может быть изменен. Как это сделать, будет рассказано ниже.

Если контур объекта представляет собой многоугольник (то есть имеет углы), то щелчок на любом участке (отрезке) контура приводит к выбору только этого участка. Чтобы выбрать весь такой контур, следует щелкнуть последовательно на всех отрезках контура, удерживая нажатой клавишу <Shift>.

Выбор заливки объекта выполняется аналогично выбору контура: включив инструмент *Arrow*, следует щелкнуть (один раз!) внутри объекта. Заливка будет выделена, а контур — нет.

Чтобы отменить выбор (снять выделение) объекта или его части, достаточно щелкнуть мышью в любой свободной позиции рабочей области.

При включении инструмента *Arrow* в поле *Options* панели инструментов отображаются кнопки-модификаторы, позволяющие установить дополнительные параметры работы этого инструмента (рис. 47):

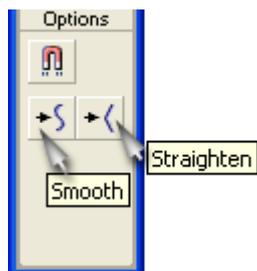


Рис. 47. Кнопки-модификаторы для инструмента *Arrow*

- *Snap to Objects* (Привязка к объекту) — если модификатор включен, то перемещаемый объект «связывается» с другим объектом (становится его частью); операция связывания выполняется в том случае, если кнопка мыши будет отпущена, когда перемещаемый объект

находится в «горячей зоне» другого объекта; индикатором такой ситуации служит размер черной окружности, отображаемой возле указателя мыши (рис. 48а, б);

- *Smooth* (Сглаживание) — щелчок на кнопке позволяет преобразовать ломаную линию в кривую (рис. 49); для этого предварительно требуется выбрать все отрезки ломаной, углы между которыми требуется сгладить;

- *Straighten* (Выпрямление) — щелчок на кнопке обеспечивает уменьшение изогнутости линии; в предельном случае с помощью нескольких щелчков любую кривую можно превратить в прямую.

Чтобы выбрать произвольную часть объекта с помощью инструмента *Arrow*, следует установить указатель мыши за пределами объекта, и, нажав кнопку мыши, переместить его таким образом, чтобы внутри прямоугольной области оказалась выбираемая часть объекта (рис. 50).

Инструмент *Arrow* можно также применять для редактирования контуров объектов и других линий. Для этого требуется подвести указатель к контуру (он не должен быть выбран!). Когда возле указателя появится значок «горячей области», нажать левую кнопку мыши и, не отпуская ее, переместить указатель в нужном направлении (рис. 51).

Замечание Инструмент Free Transform при выполнении операции выбора действует аналогично инструменту *Arrow*. Однако основное его предназначение — редактирование объекта. Поэтому особенности применения этого инструмента рассмотрены далее.

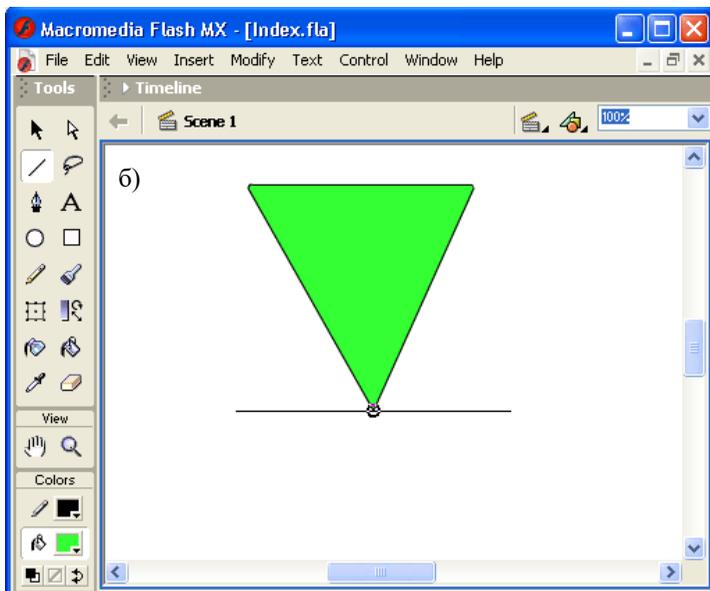
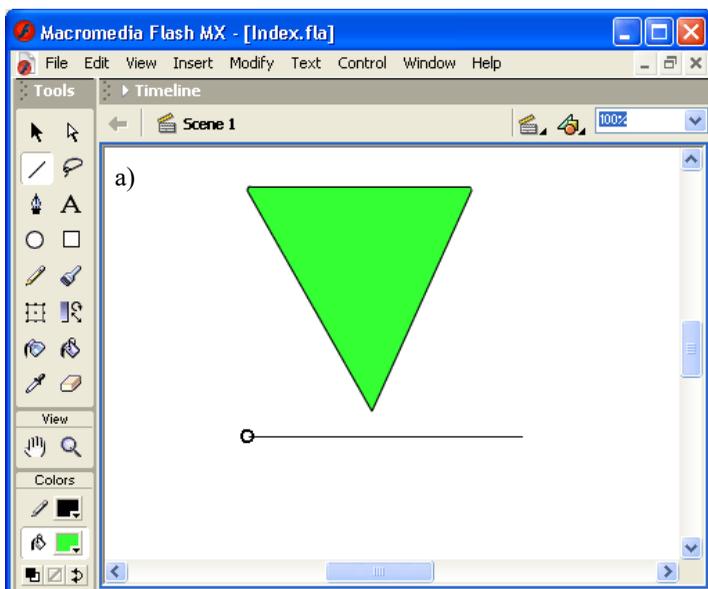


Рис. 48. Иллюстрация к связыванию объектов

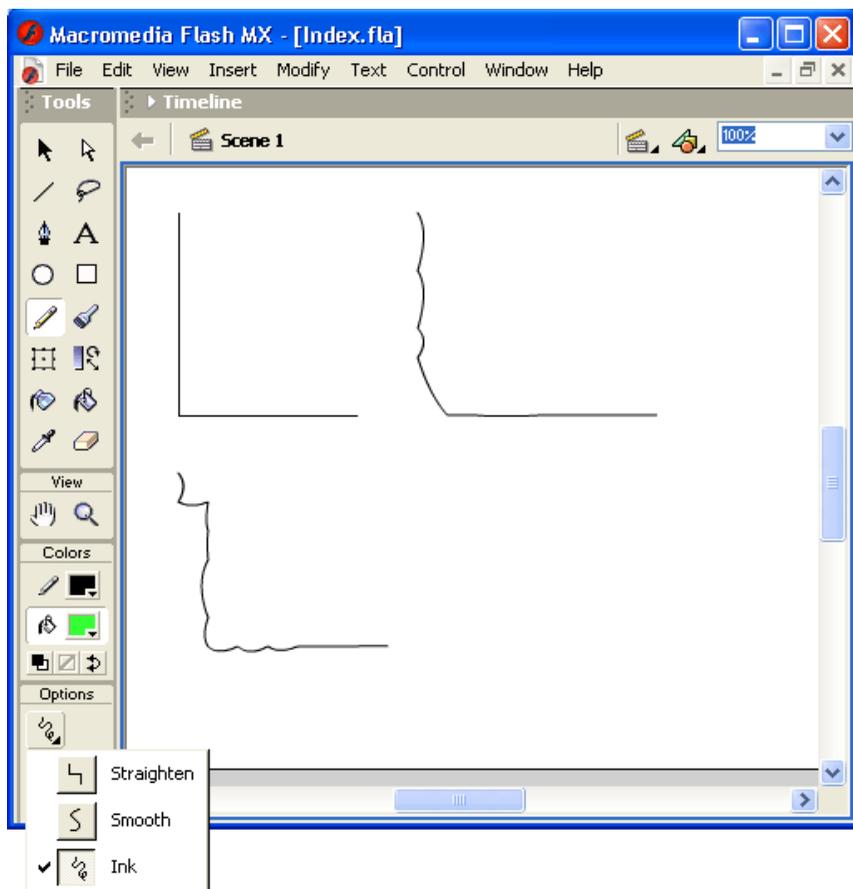


Рис. 49. Пример сглаживания ломаной с помощью инструмента Arrow

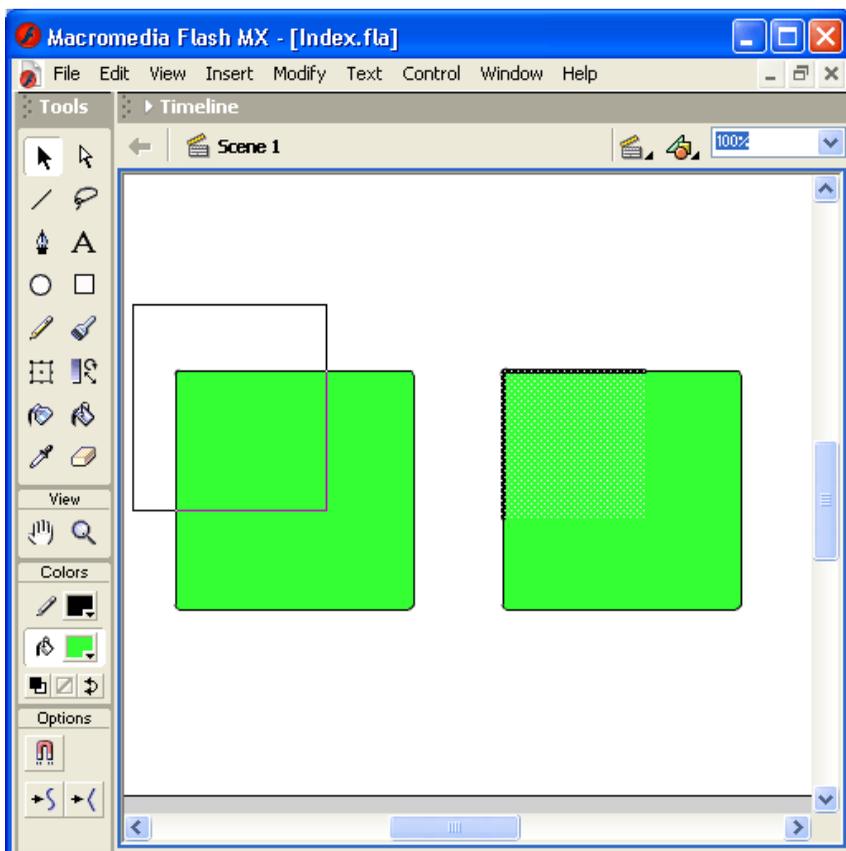


Рис. 50. Выбор части объекта с помощью инструмента Arrow

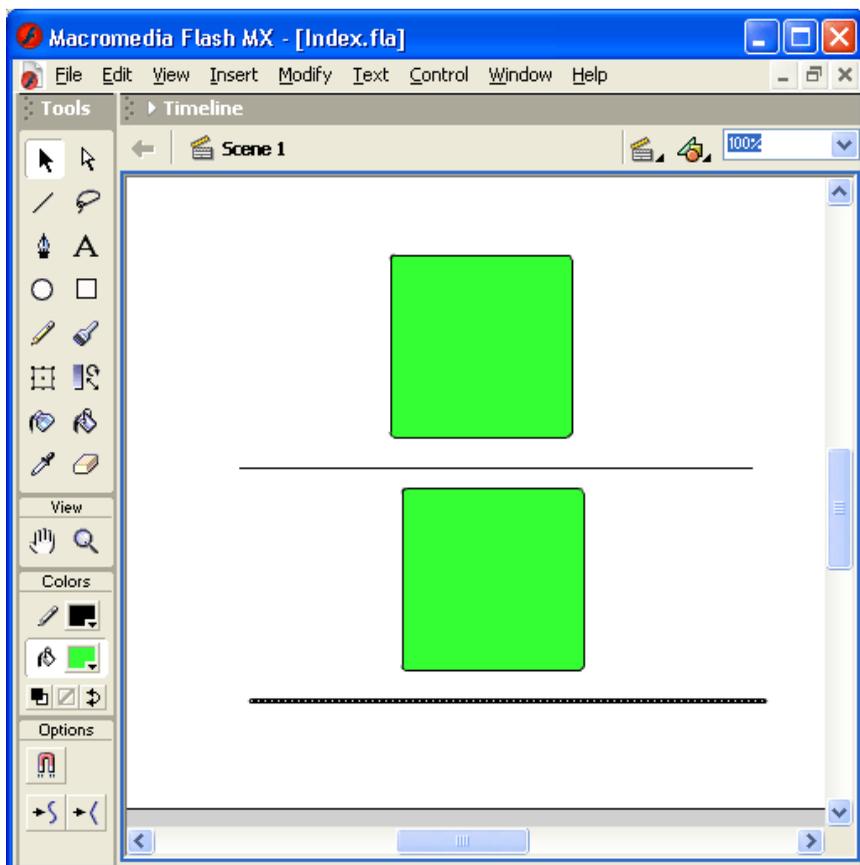


Рис. 51. Редактирование линии с помощью инструмента Arrow

Инструмент Lasso. Объект может быть выбран также с помощью инструмента *Lasso*. Для этого требуется очертить произвольную, границу вокруг объекта (рис. 52).

Чтобы выбрать произвольную часть объекта с помощью инструмента *Lasso*, следует установить указатель мыши в требуемой точке (либо за пределами, либо внутри объекта), и, нажав кнопку мыши, очертить выбираемую часть (рис. 53).

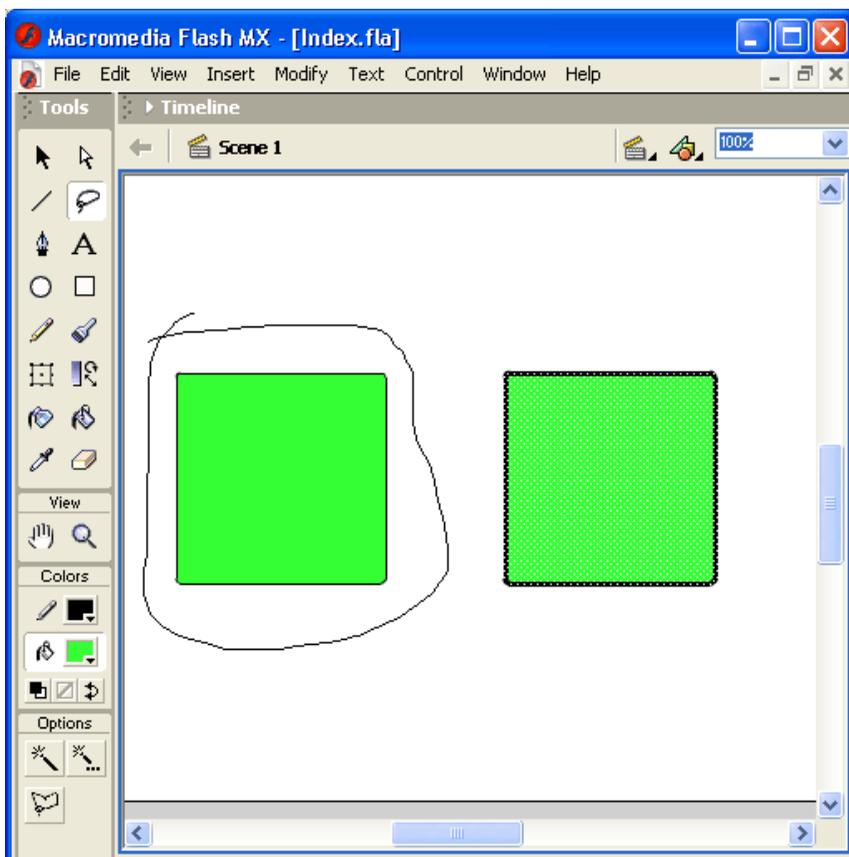


Рис. 52. Выбор объекта с помощью инструмента Lasso

Для инструмента *Lasso* предусмотрено два модификатора (рис. 54):

- *Magic Wand* (Волшебная палочка) — включение данного режима позволяет сглаживать границы выбранной области; параметры сглаживания устанавливаются в дополнительном диалоговом окне *Magic Wand Properties*, которое вызывается на экран с помощью одноименной кнопки (см. рис. 54);

- *Polygon Mode* (Режим многоугольника) - включение данного режима позволяет формировать область выбора, ограниченную прямыми линиями; для создания очередного угла требуется щелкнуть (один раз) кнопкой мыши; чтобы завершить выбор, требуется щелкнуть дважды.

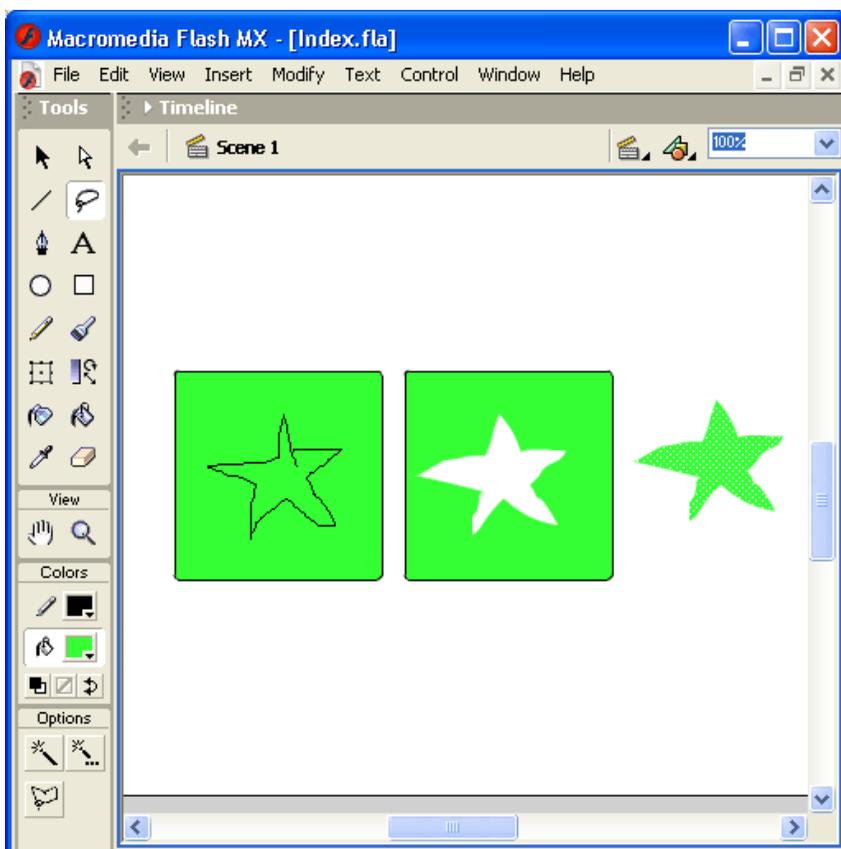


Рис. 53. Выбор части объекта с помощью инструмента Lasso



Рис. 54. Кнопки-модификаторы для инструмента Lasso

Для режима *Magic Wand* могут регулироваться следующие параметры (рис. 55):

- *Threshold* (Порог) — пороговое значение углов (в градусах) контура области выбора, которые подлежат сглаживанию;

- *Smoothing* (Сглаживание) — степень сглаживания; значения параметра выбираются из соответствующего раскрывающегося списка, который содержит четыре варианта (они перечислены ниже в порядке возрастания степени сглаживания):

- *Pixel* (По пикселям);
- *Rough* (Приблизительное);
- *Normal* (Обычное);
- *Smooth* (Гладкое).

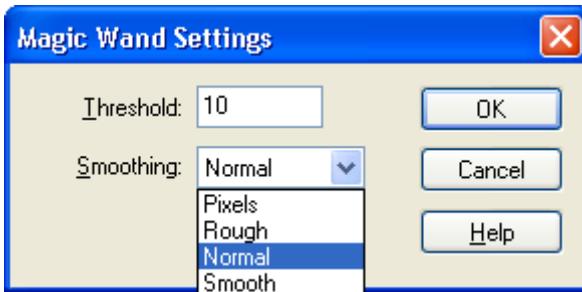


Рис. 55. Диалоговое окно Magic Wand Properties

Чтобы выбрать часть объекта, содержащую фрагмент

контура и заливки, можно также воспользоваться инструментом *Line*.

Для выбора произвольной части объекта с помощью инструмента *Line* необходимо:

1. Включить инструмент *Line*.
2. Установить указатель мыши в требуемой точке (либо за пределами, либо на контуре объекта), и, нажав кнопку мыши, провести прямую линию, отделяющую выбираемую часть от остальной части объекта.
3. Включить инструмент *Arrow* и щелкнуть дважды для выбора заливки и контура объекта (рис. 56) или один раз для выбора только заливки или контура.

Для отмены выбора, сделанного с помощью инструмента *Line*, достаточно включить инструмент *Arrow* или *Lasso* и щелкнуть левой кнопкой мыши в пустой позиции стола или рабочей области (если она является интерактивной).

Инструмент Subselection Инструмент *Subselection* отличается от рассмотренных выше двумя свойствами:

- во-первых, при выборе объекта с его помощью объект рассматривается как единое целое, без деления на контур и заливку;
- во-вторых, его работа основана на применении так называемых кривых Безье.

Под кривыми Безье (Bezier) понимается система формального (то есть на основе формул) описания графических элементов. Эта система является, по сути, основой векторной компьютерной графики.

С помощью инструмента *Subselection* объект можно выбрать одним из двух способов:

- заключить объект в прямоугольную область (как при использовании инструмента *Arrow*);

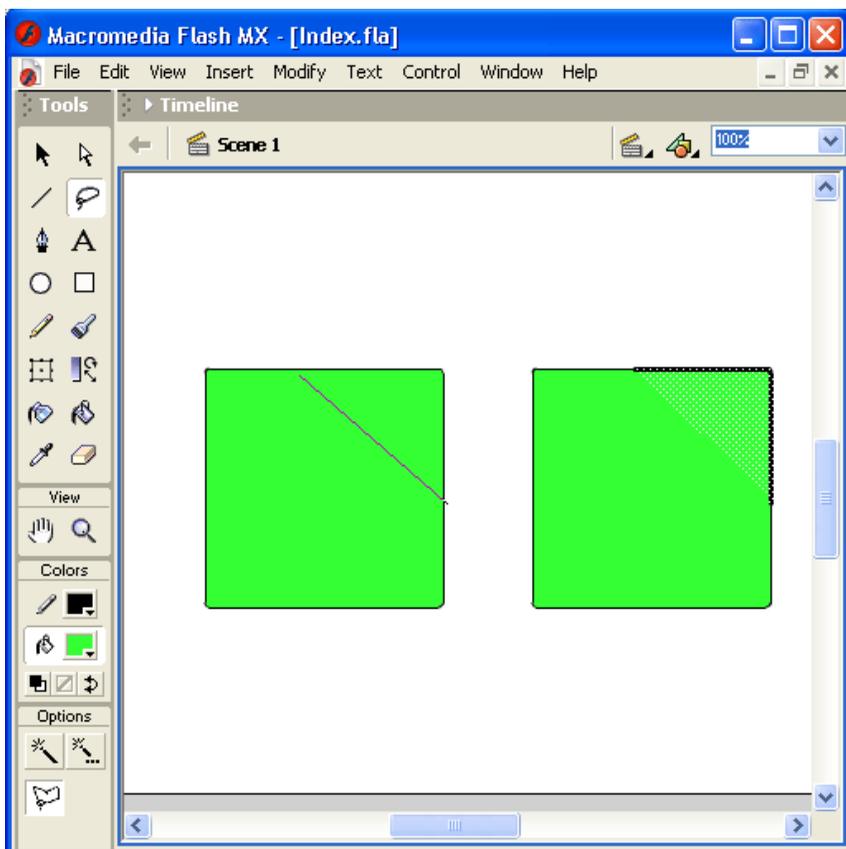


Рис. 56. Выбор части объекта с помощью инструмента Line

- подвести указатель мыши к контуру объекта и, когда возле него появится индикатор «горячей зоны», щелкнуть левую кнопку мыши.

В обоих случаях по периметру объекта появятся «чувствительные» точки, с помощью которых можно изменять форму объекта. На самом деле эти точки представляют собой средство для изменения параметров кривых Безье, из которых сформирован данный объект.

Различают два вида таких точек: точки касания и маркеры касательных. Перемещая точки касания и маркеры, можно изменять параметры кривых и, соответственно, форму объекта.

Так вот, при первом способе выбора объекта по его периметру отображаются маркеры касательных, а при втором — точки касания и (при определенных условиях) маркеры.

На рис. 57 показан объект, выбранный первым способом (слева — «в натуральную величину», справа — с двукратным увеличением).

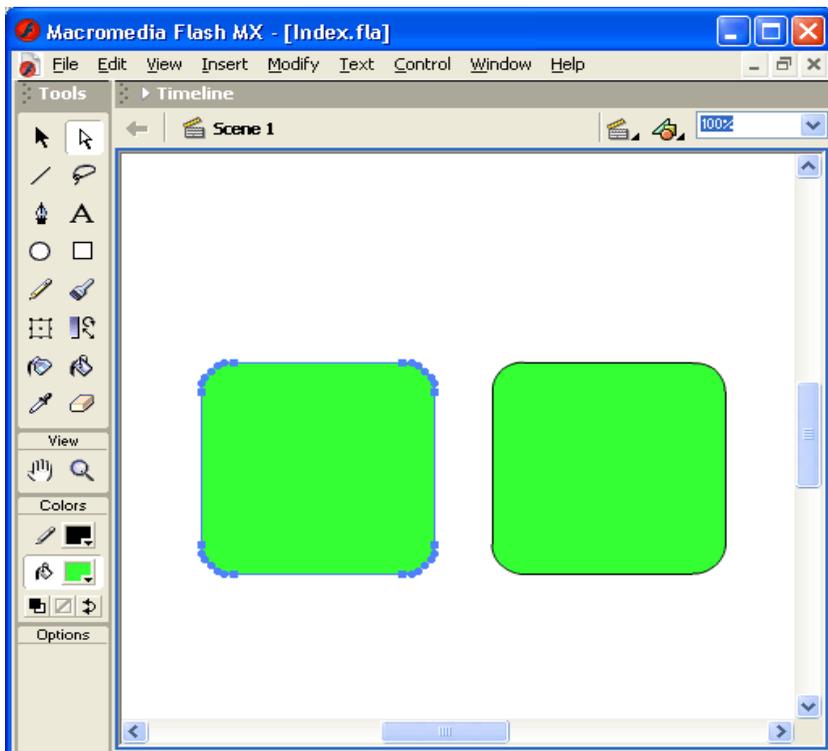


Рис. 57. Если объект заключить в выделяющую рамку, то по его периметру появятся касательные с маркерами

На рис. 58 показано изменение формы объекта посредством перемещения касательной.

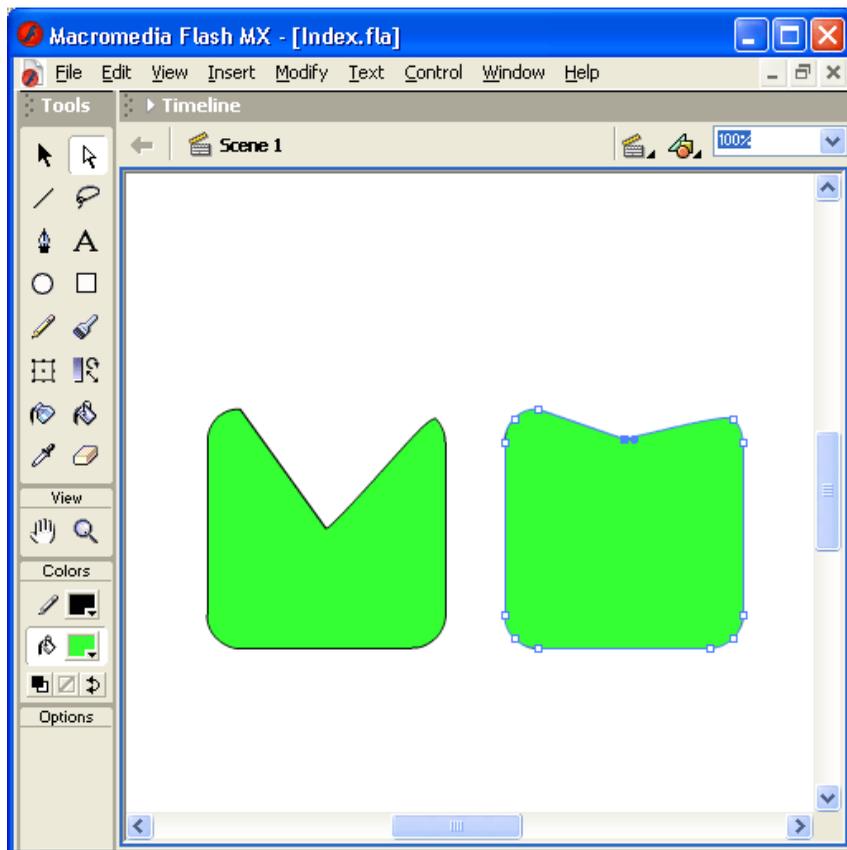


Рис. 58. Изменение формы объекта посредством перемещения касательной

При втором способе выбора объекта (то есть щелчком мыши) возможны два варианта:

- если указатель окажется между точками касания, то

возле него отображается черный квадратик (рис. 59, слева); в этом случае после щелчка мышью на контуре объекта отображаются только точки касания;

- если указатель попадет на точку касания, то возле него отображается белый квадратик (рис. 59, справа); в этом случае после щелчка мышью на контуре объекта отображаются и точки касания и касательные с маркерами (они темнее точек касания);

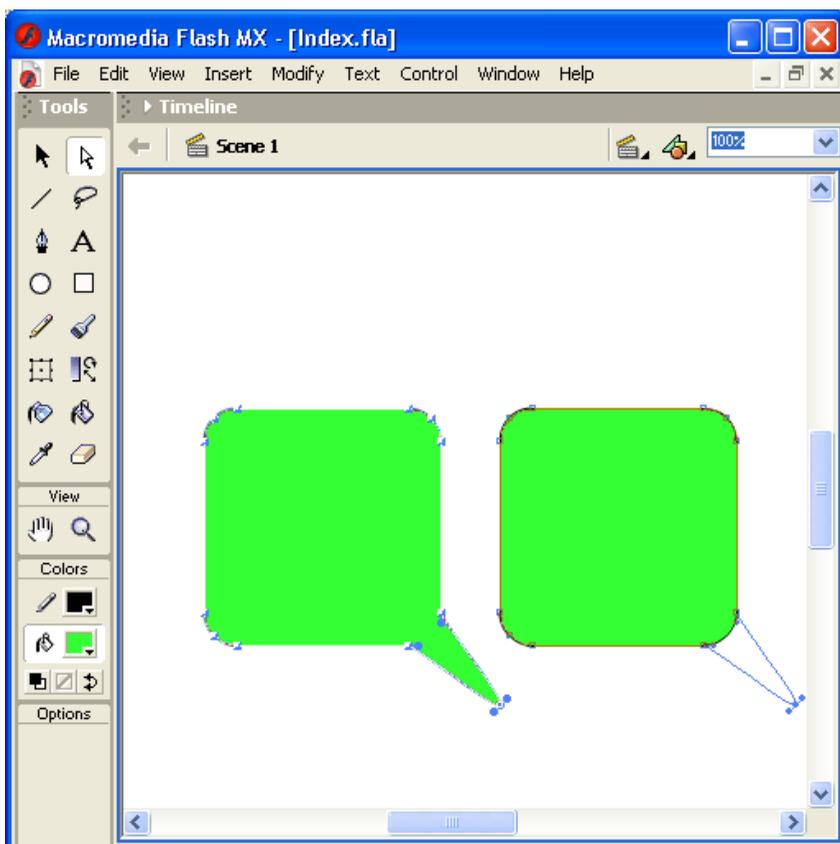


Рис. 59. Результат выбора зависит от того, попал ли указатель мыши на точку касания

На рис. 60 показано, как можно изменить форму объекта, переместив точку касания.

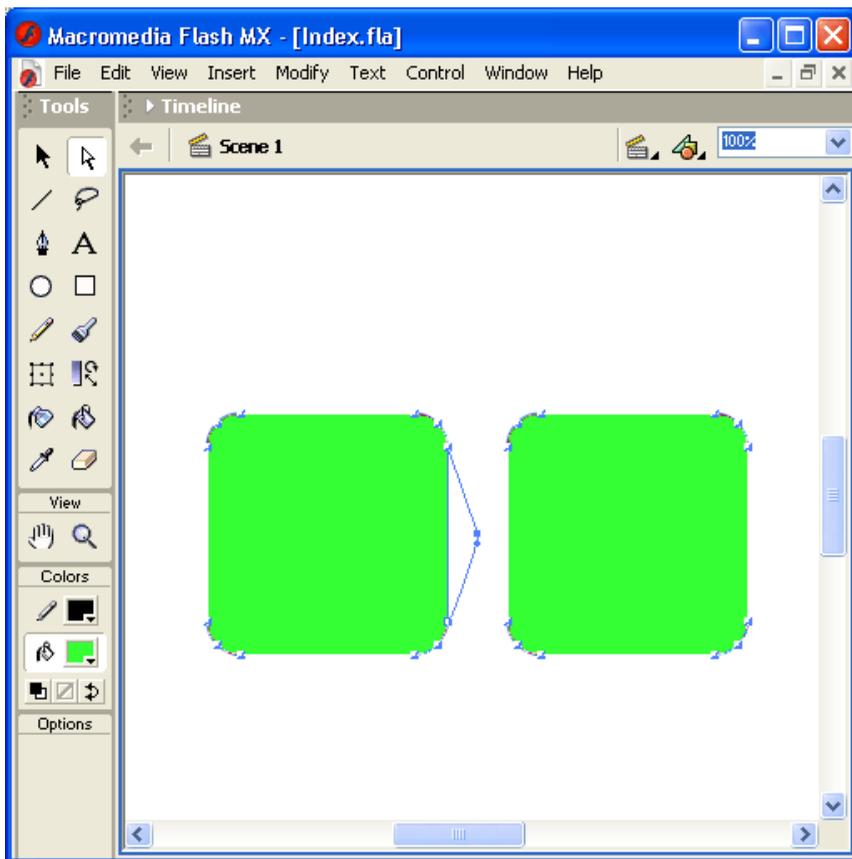


Рис. 60. Изменение формы объекта посредством перемещения точки касания

Подводя указатель к контуру объекта (между точками касания) и нажав кнопку мыши, объект можно перемещать по столу.

Над выбранным объектом (или несколькими объектами)

могут быть выполнены следующие операции:

- изменение положения точки трансформации;
- масштабирование;
- поворот и вращение;
- трансформация;
- выравнивание.

Инструмент *Free Transform* и панель *Transform*

Практически все операции по трансформированию объекта могут быть выполнены с помощью инструмента *Free Transform* и панели *Transform*.

При включении инструмента *Free Transform* становятся доступны четыре кнопки-модификатора (рис. 61):



Рис. 61. Кнопки-модификаторы инструмента *Free Transform*

- *Rotate and Skew* (Поворот и наклон); Q *Scale* (Масштабирование);
- *Distort* (Искажение); Q *Envelope* (Изгиб).

Каждая из них работает как переключатель, то есть одновременно нельзя включить два или более режимов. Тем не менее, когда ни одна из кнопок не нажата, обеспечивается наиболее универсальный режим работы инструмента *Free Transform*, при котором доступны почти все основные его функции.

Трансформирование объекта выполняется с помощью маркеров, расположенных на выделяющей рамке. Каждый маркер связан с определенной функцией. Например, угловые

позволяют поворачивать объект. Чтобы пользователям было удобнее различать предназначение маркеров, с каждым из них связан свой вариант указателя мыши. На рис. 62 показаны все возможные варианты указателей.

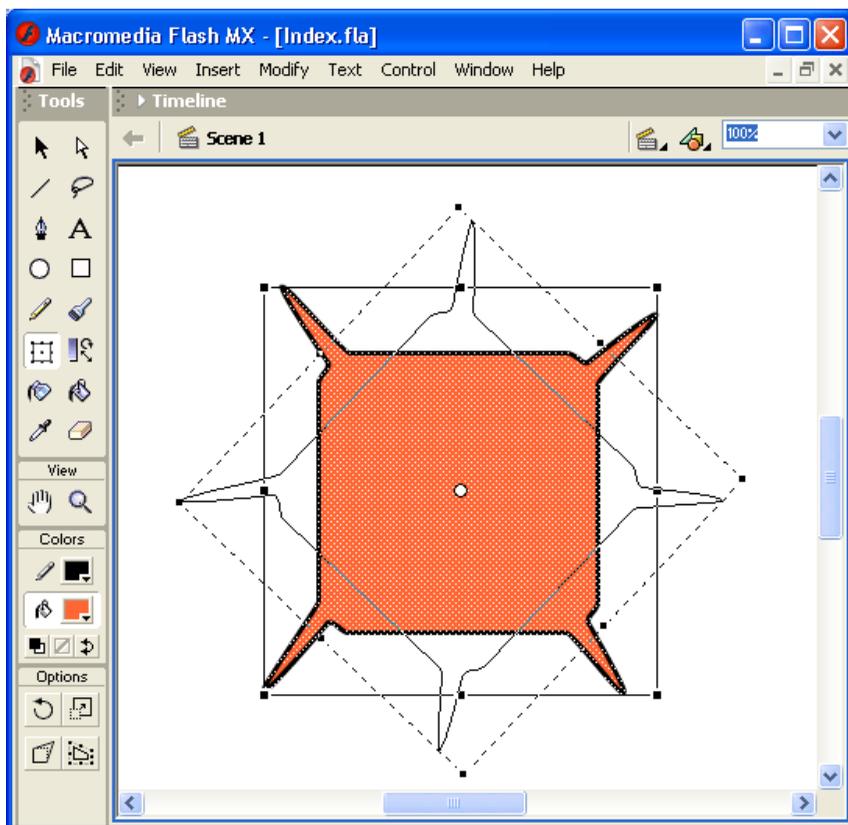


Рис. 62. Возможные варианты указателей для универсального режима инструмента Free Transform

Разумеется, в каждый момент времени вы можете видеть только один вариант указателя, и «фотомонтаж», приведенный на рис. 62, призван лишь показать схему

распределения различных функций между маркерами.

При включенном модификаторе *Rotate and Skew* возможности инструмента *Free Transform* ограничиваются лишь функциями поворота и наклона. В этом режиме виды указателей, соответствующие другим функциям, не отображаются.

При включенном модификаторе *Scale* возможности инструмента *Free Transform* ограничиваются функциями изменения размера. Соответственно, другие виды указателей не отображаются.

Модификаторы *Distort* и *Envelope* (они появились лишь в последней версии — Flash MX) существенно отличаются от двух других:

- во-первых, при включении любого из этих режимов не отображается так называемая точка трансформации объекта (об этом понятии — в следующем подразделе);
- во-вторых, в каждом из них используется только один вид указателя, не применяемый ни в одном из других режимов работы инструмента *Free Transform* (в форме широкой белой стрелки).

Подробнее особенности применения модификаторов *Distort* и *Envelope* рассмотрены в подразделе «Искажение объекта и создание перспективы».

Панель *Transform* имеет в целом то же предназначение, что и инструмент *Free Transform*, однако перечень реализуемых с ее помощью функций несколько отличается.

Если при выбранной вами конфигурации рабочей области панель *Transform* не отображается на экране, выберите в меню *Window* команду *Transform* или нажмите сочетание клавиш <Ctrl> + T.

Формат панели *Transform* показан на рис. 63.

Из элементов интерфейса панели в дополнительном пояснении нуждаются только две кнопки, расположенные в ее нижнем правом углу:

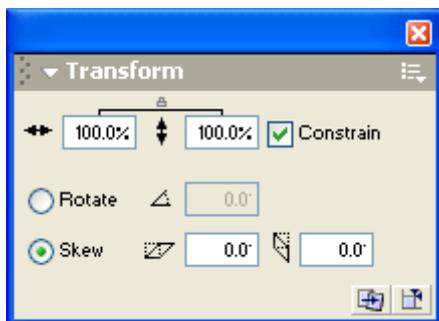


Рис. 63. Формат панели Transform

- *Reset* (Сброс), которая предназначена для восстановления исходных параметров объекта;
- *Copy And Apply Transform* (Копировать и применить трансформацию), которая обеспечивает предварительное создание копии исходного изображения и последующую трансформацию оригинала; в результате вы получаете как бы два совмещенных изображения, различающихся выполненной трансформацией; пример использования такого режима для операции поворота показан на рис. 64.

Режим *Copy And Apply Transform* весьма полезен при создании покадровой анимации.

Во Flash MX (в отличие от предыдущих версий) любой элемент фильма – графический объект, символ, группа, текстовый блок – имеют так называемую точку трансформации (*transformation point*).

Точка трансформации - это некоторая точка объекта, относительно которой Flash выполняет позиционирование и преобразование объекта (поворот, наклон и т. д.). По умолчанию точка трансформации совпадает с геометрическим центром объекта.

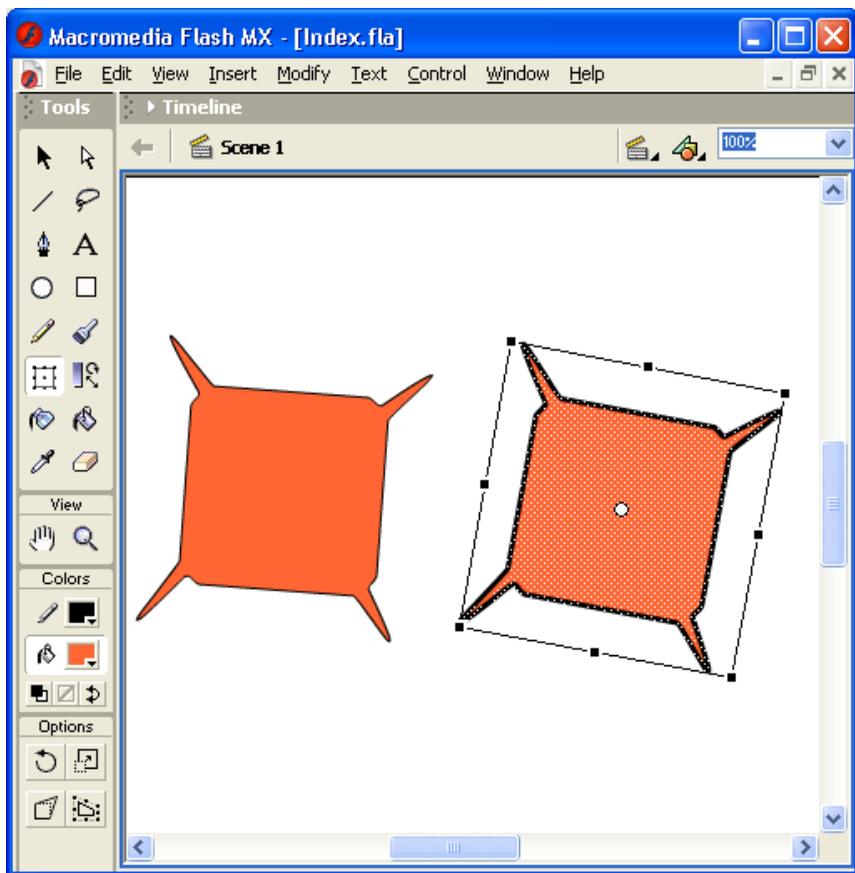


Рис. 64. Пример использования режима Copy And Apply Transform (слева — исходный объект, справа — результат его трансформации)

Визуально эта точка отображается только в режиме трансформации объекта, то есть если объект выбран с помощью инструмента *Free Transform*. Она представляет собой небольшой кружок (рис. 65).

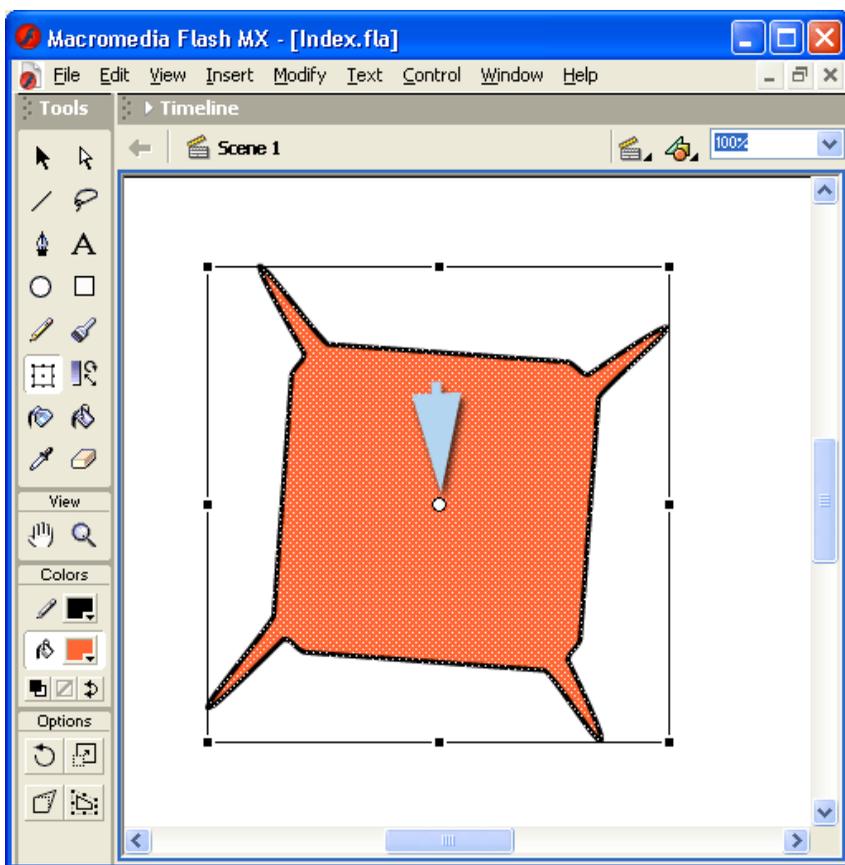


Рис. 65. Примеры расположения точек трансформации объектов: графического (слева), текста (в центре) и символа (справа)

Символы и экземпляры символов имеют помимо точки трансформации еще и точку регистрации (registration point), отображаемую в виде небольшого крестика.

Перемещая точку трансформации, вы можете изменять результат выполнения операций позиционирования и

преобразования объекта. Чтобы изменить положение точки трансформации, необходимо:

1. Включить инструмент *Free Transform* и выбрать с его помощью требуемый объект фильма.

2. Подвести к точке трансформации указатель мыши, и когда возле него появится маленький кружок, нажать левую кнопку мыши.

3. Не отпуская кнопку мыши, переместить точку трансформации на новую позицию.

Рис. 66 иллюстрирует различие в выполнении операции поворота в зависимости от положения точки трансформации (слева — исходное, в геометрическом центре фигуры, справа — после перемещения точки).

Масштабирование объекта Под масштабированием (Scaling) объекта понимается увеличение или уменьшение размеров объекта по горизонтали либо по вертикали, или по обоим измерениям одновременно.

Экземпляры символов, группы и текстовые блоки масштабируются относительно их точек привязки.

Масштабирование объекта можно выполнить одним из трех способов:

- прямым манипулированием (с помощью мыши);
- с помощью команды *Scale* (Масштаб);
- указав размер объекта в панели *Transform*.

Чтобы масштабировать объект с помощью мыши, следует выполнить следующие действия:

1. Включить инструмент *Free Transform* и выбрать с его помощью требуемый объект фильма.

2. Установить указатель на один из маркеров масштабирования (см. рис. 62) и, нажав кнопку мыши, переместить его в нужном направлении; об изменении размера объекта можно судить по изменению его контура, отображаемого при перемещении мыши (рис. 67).

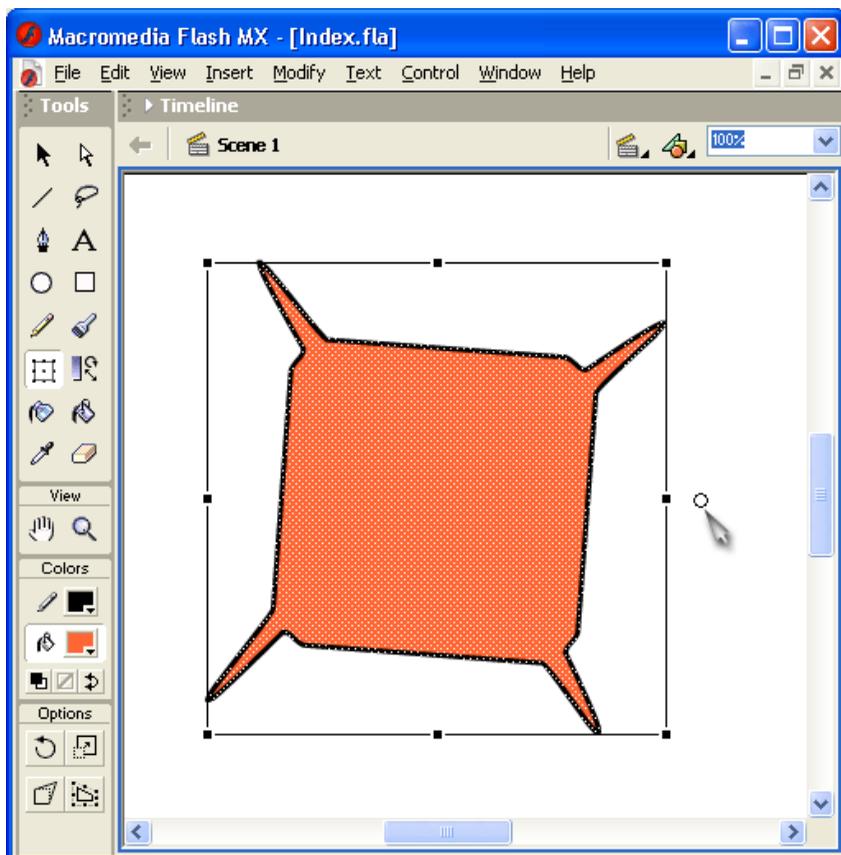


Рис. 66. Пример перемещения точки трансформации

Обратите внимание, что хотя в приведенном примере выполняется пропорциональное изменение объекта по обоим измерениям, началом координат служит не центр выделенной области, а точка трансформации объекта.

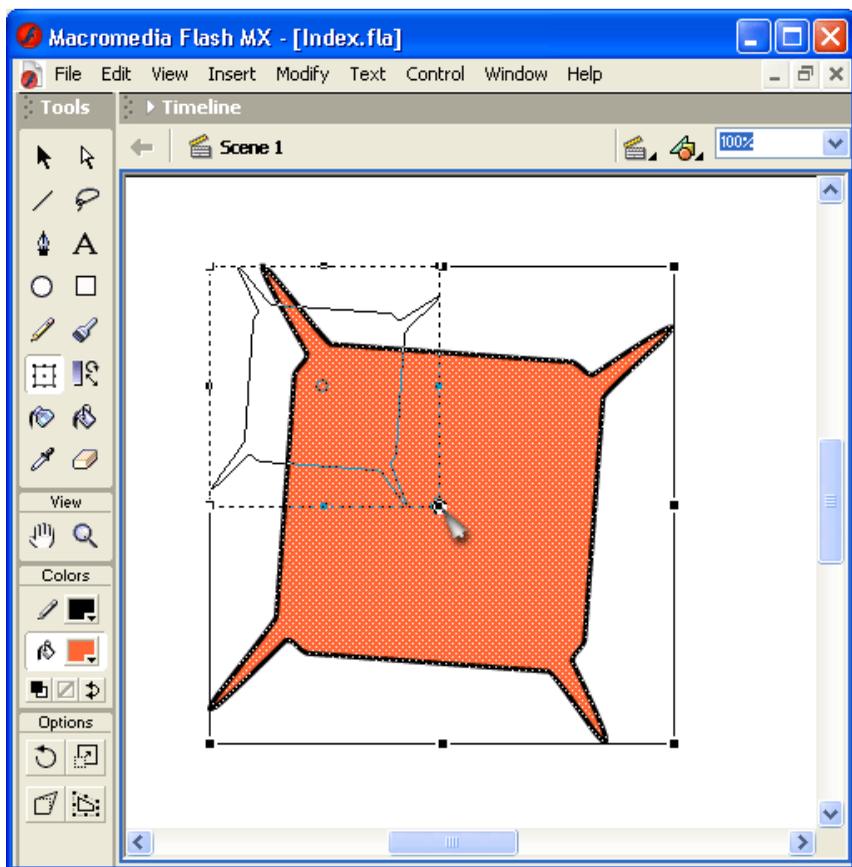


Рис. 67. Масштабирование объекта с помощью мыши

Чтобы масштабировать объект с помощью команды *Scale*, следует выполнить следующие действия:

1. Выбрать требуемый объект фильма с помощью инструмента *Arrow*.
2. В меню *Modify* выбрать каскадное меню *Transform*, а в нем — команду *Scale* (Масштаб); в результате ее выполнения объект окажется заключен в выделяющую рамку с маркерами.

3. Установить указатель на один из маркеров и, нажав кнопку мыши, переместить его в нужном направлении.

Чтобы изменить размер объекта с помощью панели *Transform*, необходимо:

1. Выбрать требуемый объект фильма с помощью инструмента *Arrow*.

2. В меню *Window* выбрать команду *Transform*.

3. В соответствующем текстовом поле указать новый размер объекта (в процентах, исходному размеру соответствует значение 100%) и нажать клавишу <Enter>.

Если требуется обеспечить пропорциональное изменение размера по обоим измерениям, то предварительно следует установить флажок *Constrain* (Согласовать); в этом случае изменение одного из размеров приводит к автоматическому изменению другого.

Поворот объекта Поворот (*Rotating*) - это изменение положения объекта относительно его точки трансформации.

Поворот объекта можно выполнить одним из трех способов:

- прямым манипулированием (с помощью мыши);
- с помощью команды *Rotate* (Повернуть);
- указав угол поворота в панели *Transform*.

Чтобы выполнить поворот объекта с помощью мыши, следует выполнить следующие действия:

1. Включить инструмент *Free Transform* и выбрать с его помощью требуемый объект фильма.

2. Установить указатель на один из угловых маркеров и, нажав кнопку мыши, переместить его в нужном направлении; о величине угла поворота объекта можно судить по положению его контура, отображаемого при перемещении мыши; пример поворота объекта был показан на рис. 66.

Чтобы изменить положение объекта с помощью панели *Transform*, необходимо:

1. Выбрать требуемый объект фильма с помощью

инструмента *Arrow*.

2. Установить переключатель *Rotate* (Повернуть) и в соответствующем текстовом поле указать угол поворота объекта (в градусах, исходному размеру соответствует значение 0.0); для поворота по часовой стрелке следует вводить положительные значения угла поворота, для поворота против часовой стрелки - отрицательные.

3. Нажать клавишу <Enter>.

Отражение объекта Отражение объекта (Flipping) - это создание зеркальной копии объекта относительно некоторой оси симметрии (горизонтальной или вертикальной).

Чтобы отразить объект, необходимо открыть меню *Modify*, и в каскадном меню *Transform* выбрать одну из двух команд:

- *Flip Vertical* (Отразить по вертикали);
- *Flip Horizontal* (Отразить по горизонтали).

Интересный вариант операции отражения может быть реализован с помощью инструмента *Free Transform*. Для этого необходимо либо в «универсальном» режиме, либо при включенном модификаторе *Scale* переместить с помощью мыши любой из центральных маркеров в том направлении, куда следует «отразить» объект (рис. 68).

Наклон объекта Наклон объекта (Skewing) — это искажение («перекос») объекта по одной из осей координат при одной неподвижной стороне выделяющей рамки. На рис. 69 показан наклон по горизонтали, а на рис. 70 - наклон объекта по вертикали.

Наклон объекта может быть выполнен либо путем прямого манипулирования, либо с помощью панели *Transform*.

Чтобы наклонить объект с помощью мыши, следует выполнить следующие действия:

1. Включить инструмент *Free Transform* и выбрать с его помощью требуемый объект фильма.

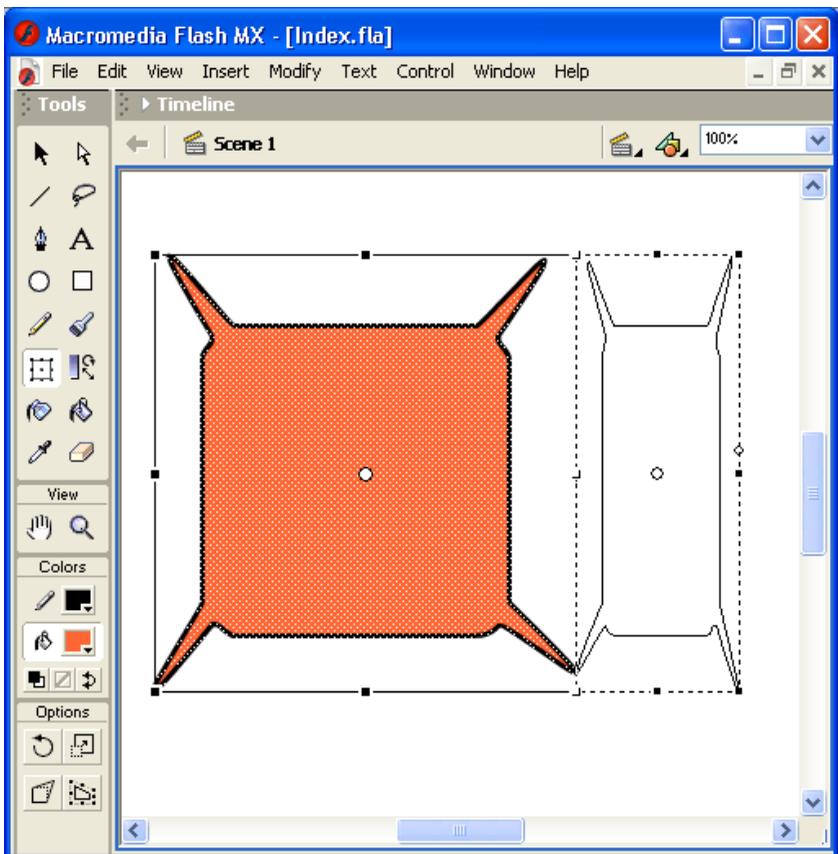


Рис. 68. Отражение объекта с помощью инструмента Free Transform

2. Поместить указатель недалеко от одного из центральных маркеров и добиться того, чтобы указатель принял соответствующий вид.

3. Нажав кнопку мыши, переместить указатель в нужном направлении.

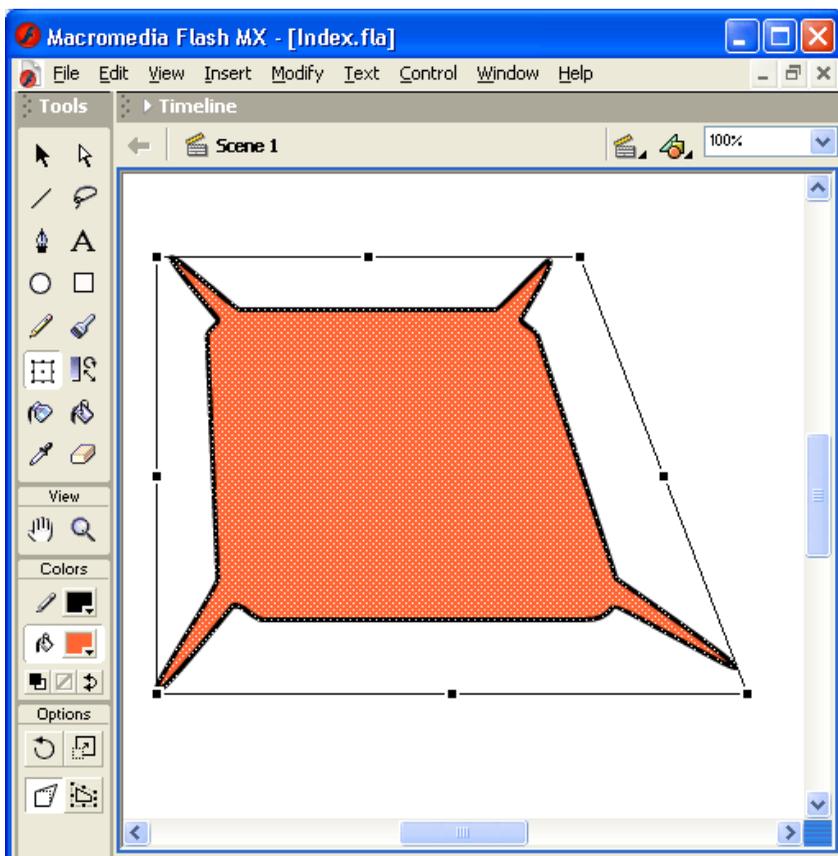


Рис. 69. Наклон объекта по горизонтали

Чтобы выполнить наклон объекта с помощью панели *Transform*, необходимо:

1. Выбрать требуемый объект фильма с помощью инструмента *Arrow*.
2. Установить переключатель *Skew* (Наклон).
3. В соответствующем текстовом поле указать угол наклона объекта (в градусах, исходному положению соответствует значение 0, 0) и нажать клавишу <Enter>; для

наклона по часовой стрелке следует вводить положительные значения угла, для наклона против часовой стрелки — отрицательные.

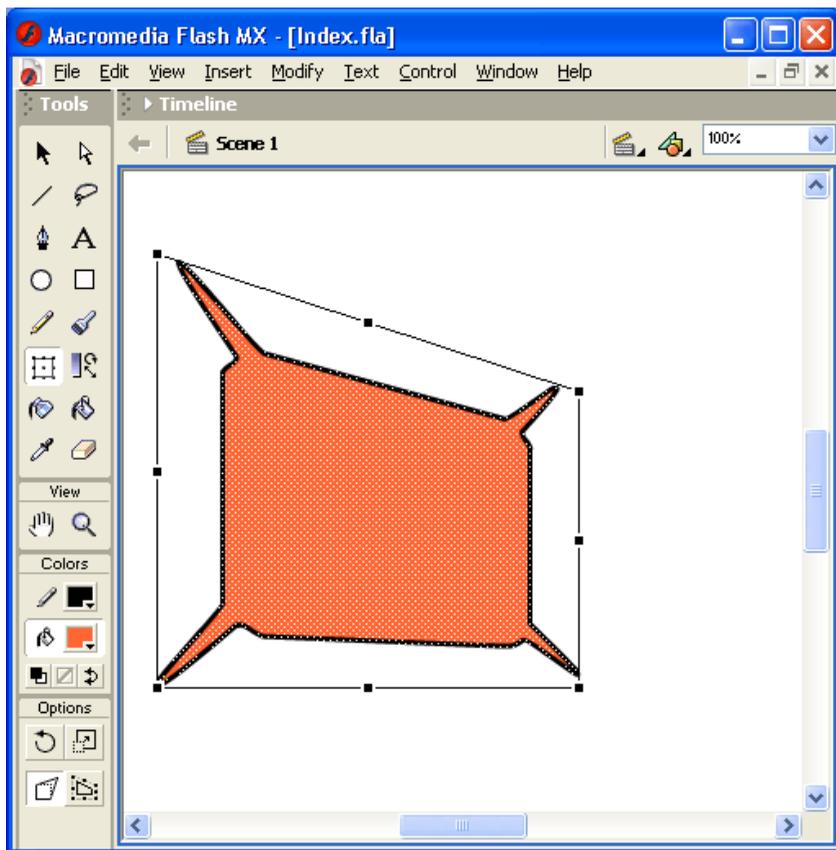


Рис. 70. Наклон объекта по вертикали

Если необходимо наклонить объект одновременно и по горизонтали и по вертикали, следует ввести требуемые значения в обоих полях, как показано на рис. 71 (клавишу <Enter> следует нажимать только после указания обоих значений).

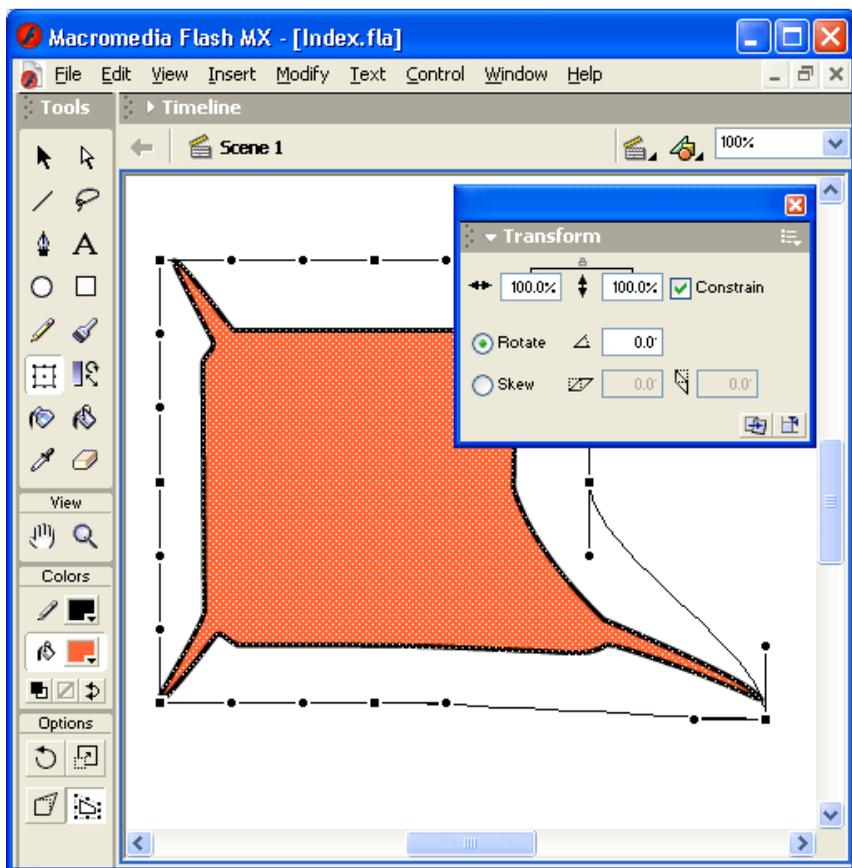


Рис. 71. Наклон объекта с помощью панели Transform

Достоинство применения панели *Transform* состоит в том, что она позволяет выполнить наклон объекта по двум измерениям одновременно.

Создание эффекта перспективы и искажение Эти две новые функции стали доступны в Flash MX благодаря включению в его состав инструмента *Free Transform*, который уже упоминался неоднократно выше. Указанные функции включаются с помощью кнопок-модификаторов

Distort и *Envelope* соответственно.

На рис. 72 приведен пример создания перспективы для фрагмента текста (правда, для этого требуются некоторые предварительные преобразования, о которых рассказывается в разделе «Работа с текстом»).

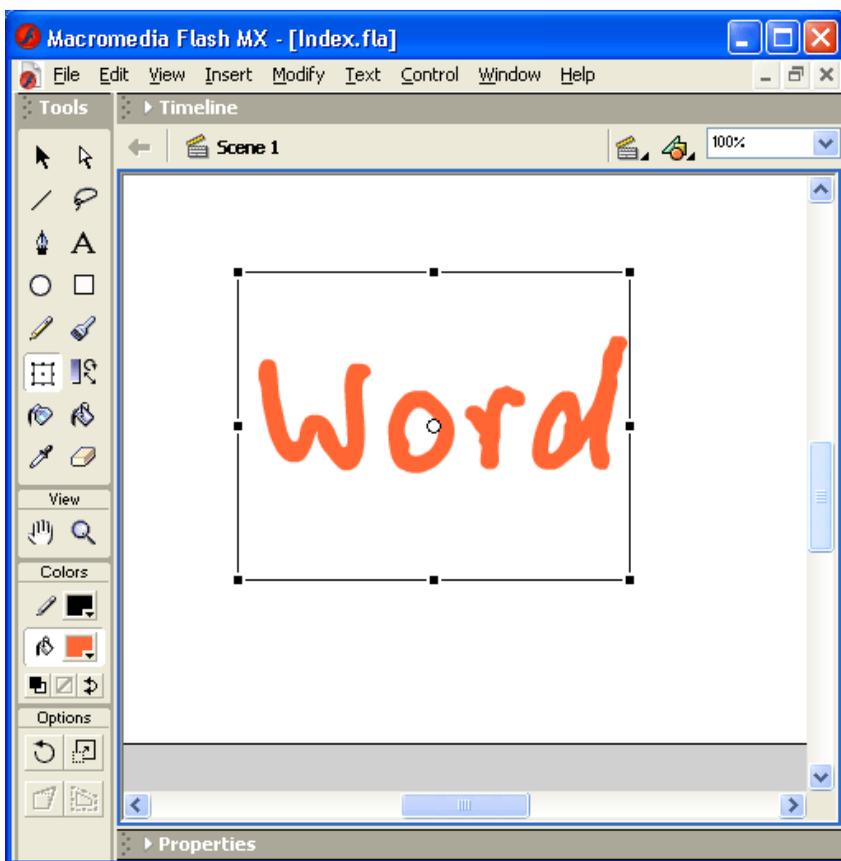


Рис. 72. Пример создания эффекта перспективы

Для создания эффекта перспективы необходимо:

1. Выбрать объект с помощью инструмента *Free*

Transform.

2. Включить кнопку-модификатор *Distort*.

3. Подвести указатель мыши к одному из угловых маркеров и, когда указатель примет форму белой стрелки, нажать кнопку и переместить маркер в нужном направлении; при этом остальные угловые маркеры остаются неподвижны, что и обеспечивает достижение требуемого эффекта (рис. 73).

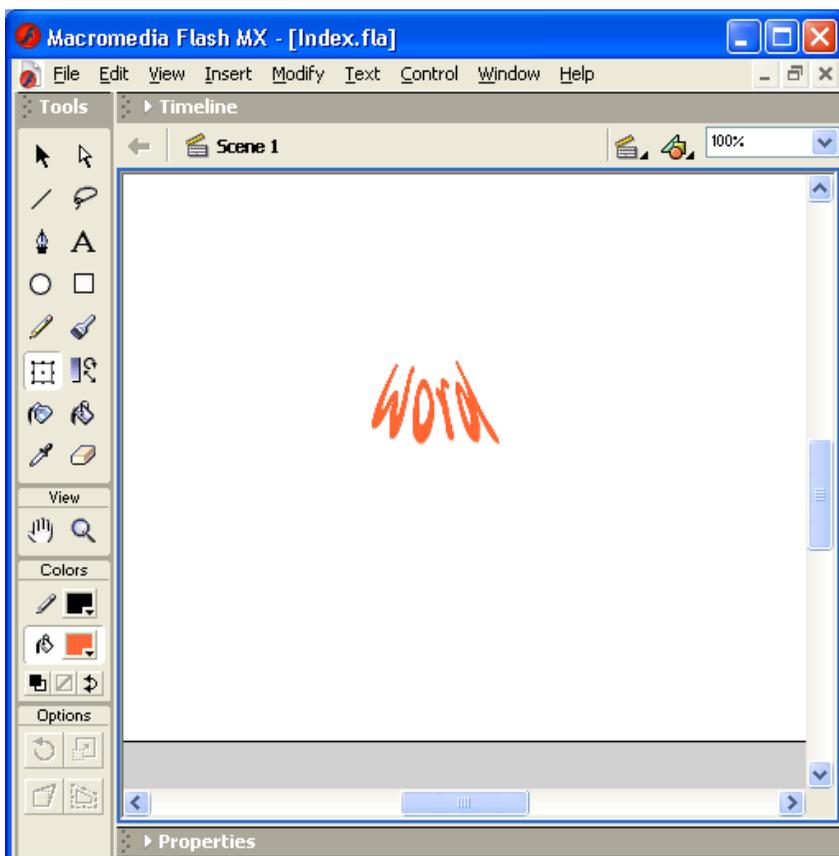


Рис. 73. Перемещение маркера в режиме *Distort*

При включении кнопки-модификатора *Envelope* на выделяющей рамке объекта появляется большое количество дополнительных маркеров, каждый из которых позволяет перемещать произвольным образом связанный с ним участок объекта (рис. 74).

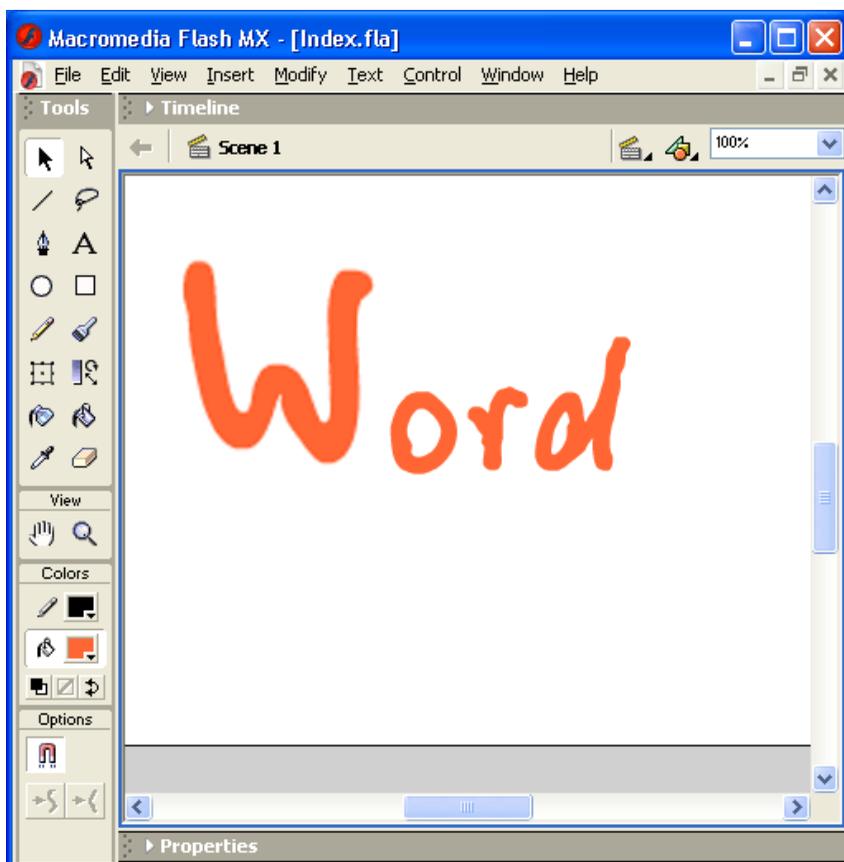


Рис. 74. Трансформация объекта в режиме Envelope

Из рисунка видно, что при перемещении маркера он играет роль точки касания к кривой Безье. Другими словами,

режим *Envelope* аналогичен применению инструмента *Subselection*.

6.8 Создание анимации в пакете FLASH-MX

В зависимости от располагаемого времени, сюжета фильма и собственных художественных способностей вы можете выбрать один из двух способов «оживления» персонажей вашего фильма:

- покадровую анимацию (Frame-by-frame, «кадр за кадром»), когда каждый следующий кадр вы создаете собственными руками (либо импортируете из внешнего источника);
- автоматическую анимацию (tweened-анимацию), когда вы выстраиваете лишь ключевые кадры, а все промежуточные кадры Flash формирует самостоятельно.

Необходимо отметить, что оба механизма могут использоваться совместно, причем не только в рамках одного фильма, но и применительно к одному объекту. Например, более сложные в сюжетном плане фрагменты могут быть основаны на покадровой анимации, а фрагменты с «предсказуемым» развитием сюжета получены с помощью автоматической.

Такое объединение возможно благодаря использованию для каждого объекта единой оси времени, представленной в окне редактора временной диаграммой - Time Line. Объединив на одной временной диаграмме «линии жизни» нескольких объектов, можно получить сцену, в которой участвуют несколько «персонажей».

Особенность применения автоматической анимации состоит в том, что с ее помощью можно анимировать только один объект на данном слое. Для создания сцены, в которой предполагается наличие нескольких анимированных объектов, требуется разместить каждый из них на отдельном

слое.

Независимо от того, какой механизм используется для создания отдельных кадров, суть анимирования заключается в том, чтобы отразить изменение объекта во времени.

Основной принцип анимации - каждому моменту времени — свой кадр. Соответственно, процедура создания простейшего анимированного фильма во Flash состоит в том, чтобы подготовить изображения объекта, отражающие его изменение, и расставить их по временной оси

6.9 Основные элементы TimeLine. Простой и ключевой кадры

Временная диаграмма. Над рабочей областью расположена панель временной диаграммы (Timeline). Временная диаграмма, как и стол, принадлежит конкретной сцене фильма. Она позволяет описать взаимное расположение слоев на сцене, последовательность изменения состояния объектов, представленных на сцене, и выполнить некоторые другие операции. Временная диаграмма является основным инструментом при создании анимации и при описании поведения интерактивных элементов фильма.

Панель временной диаграммы имеет весьма сложную структуру и содержит большое число элементов управления. Кроме того, в зависимости от установленных параметров, внешний вид временной диаграммы изменяется в достаточно широком диапазоне. Тем не менее, от умения работать с временной диаграммой в значительной степени зависит эффективность работы с Flash в целом.

В данном подразделе мы рассмотрим только основные компоненты панели временной диаграммы и наиболее общий ее формат. В последующих подразделах вопросы, связанные с ее использованием, будут уточняться и конкретизироваться;

Итак, в общем случае на панели временной диаграммы

могут быть представлены следующие элементы (рис. 75):

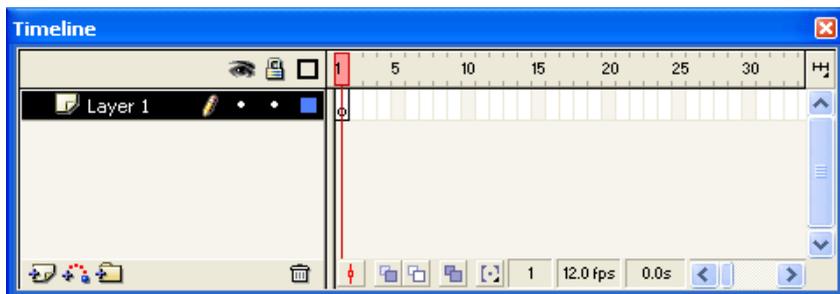


Рис. 75. Панель временной диаграммы

- описание слоев текущей сцены фильма; описание представлено в виде своеобразной таблицы, содержащей несколько столбцов; в них указываются названия слоев и их атрибуты; подробнее эта часть панели временной диаграммы будет рассмотрена в главе «Создание анимации»;

- собственно временная диаграмма, содержащая шкалу кадров, изображение «считывающей головки» и временные диаграммы для каждого слоя сцены;

- выпадающее меню, которое позволяет выбирать формат представления кадров на временной диаграмме;

- строка состояния окна временной диаграммы, на которой выводится информация о некоторых параметрах фильма, а также имеются кнопки для управления отображением кадров анимации на столе.

Рассмотрим подробнее элементы временной диаграммы.

Начнем со шкалы кадров. Шкала является общей для всех слоев сцены. На ней отображена нумерация кадров в возрастающем порядке. Шаг нумерации равен 5 (он остается неизменным при любом формате кадров).

Считывающая головка (Play head) является своеобразным индикатором, указывающим текущий (активный) кадр анимации. При создании очередного кадра и

при воспроизведении фильма считывающая головка перемещается вдоль временной диаграммы автоматически. Вручную (с помощью мыши) ее можно перемещать только после того, как анимированный фильм будет создан. Причем перемещать ее можно в обоих направлениях; при этом будет изменяться и состояние ани-мированных объектов.

Временная диаграмма отдельного слоя представляет собой графическое изображение последовательности кадров. Ниже видно, что само по себе изображение кадров на временной диаграмме весьма информативно. В частности, по изображению кадров можно определить, какой способ анимирования использован для данного слоя. Ниже приведена краткая характеристика различных форматов временной диаграммы слоя (точнее, расположенного в нем объекта).

Ключевые кадры для tweened-анимации перемещения обозначаются как черные точки, соединенные линией со стрелкой, на светло-синем фоне (линия со стрелкой заменяет все промежуточные кадры)

Ключевые кадры для tweened-анимации трансформации обозначаются как черные точки, соединенные линией со стрелкой, на светло-зеленом фоне (линия со стрелкой заменяет все промежуточные кадры)

Пунктирная линия указывает, что конечный (заключительный) ключевой кадр отсутствует

Одиночный ключевой кадр в виде черной точки, после которого идет последовательность светло-серых кадров, означает, что содержимое ключевого кадра не изменяется; заключительный ключевой кадр в этом случае обозначается пустым прямоугольником

Маленькая буква *a* указывает, что данному кадру с помощью панели Actions (Действия) было назначено некоторое действие.

Красный флажок указывает, что кадр содержит метку

или комментарий.

Непрерывная последовательность ключевых кадров, обозначенных черными точками, означает покадровую анимацию.

Желтый якорек говорит о том, что данный кадр содержит именованную метку, используемую для навигации между кадрами фильма; если позволяет место на диаграмме, рядом с якорем выводится имя метки.

Внешний вид временной диаграммы также существенно зависит от выбранного формата кадров. Выбор, как уже было сказано, выполняется с помощью выпадающего меню *Frame View*. Действие команд меню распространяется на все строки (слои) временной диаграммы. Пункты меню разделены на четыре подгруппы (рис. 76).

Команды из первой позволяют установить размер ячейки кадра по горизонтали; возможны следующие варианты:

- *Tiny* (Крошечный);
- *Small* (Мелкий);
- *Normal* (Обычный);
- *Medium* (Средний);
- *Large* (Крупный).

Формат *Large* целесообразно использовать для просмотра параметров звукового сопровождения.

Во вторую группу входит единственная команда — *Short* (Короткий), которая обеспечивает уменьшение размера ячейки кадра по вертикали. Такое изменение влияет и на формат левой части временной диаграммы (где расположено описание слоев), как показано на рис. 77.

В третью группу также включена только одна команда — *Tinted Frames* (Окрашенные кадры). При выборе данного пункта используется цветовая индикация кадров для различных видов анимации (см. выше). По умолчанию цветовая индикация кадров включена.

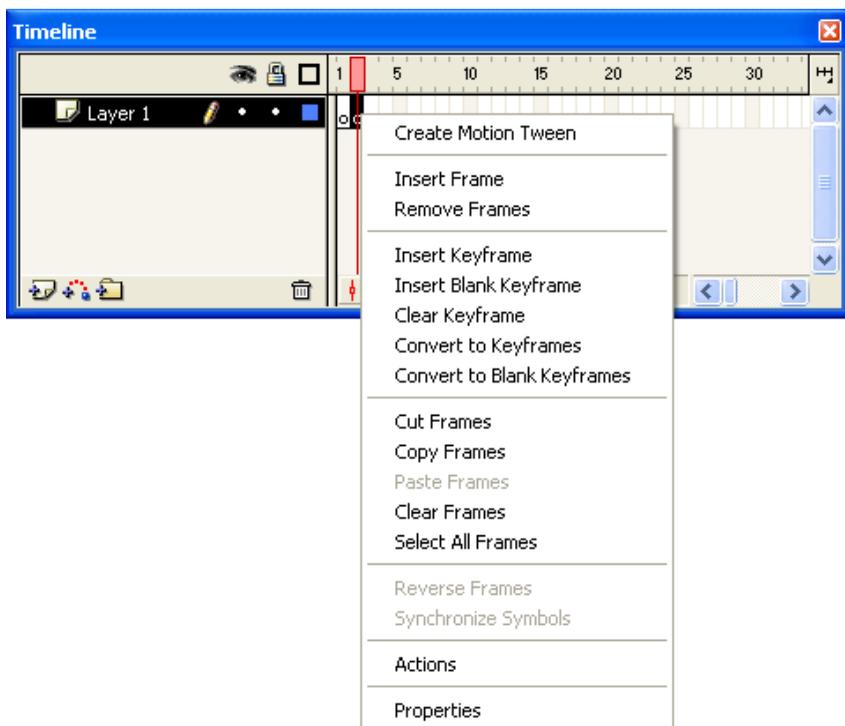


Рис. 76. Выпадающее меню формата кадров

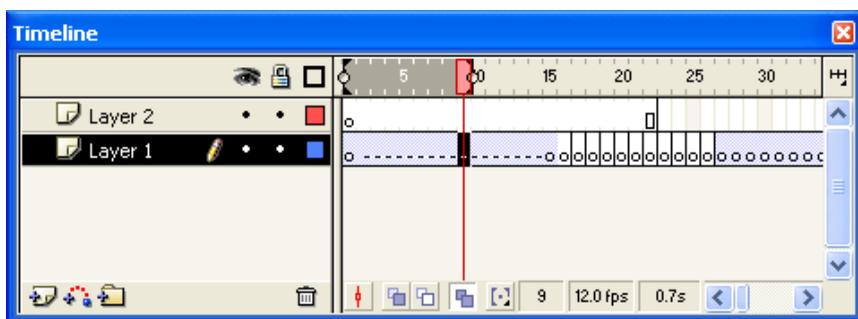


Рис. 77. Изменение формата временной диаграммы по команде Short (сравн. с рис. 75)

Последняя, четвертая, группа состоит из двух команд:

- *Preview* (Предварительный просмотр) — команда позволяет поместить непосредственно во временную диаграмму в уменьшенном виде содержимое ключевых кадров анимации;

- *Preview In Context* (Предварительный просмотр в контексте) -данную команду целесообразно использовать в том случае, если в каждом ключевом кадре изменяются два или более объектов, поскольку она позволяет вставить в ячейку кадра на временной диаграмме все пространство сцены; однако в этом случае изображение кадра уменьшается еще в большей степени, чем при использовании команды *Preview*.

Создание последовательности ключевых кадров. Основным инструментом при создании покадровой анимации является панель временной диаграммы. С ее помощью вы можете создавать, удалять и перемещать кадры анимации, изменять режимы просмотра.

Можно сказать, что мультик, созданный с применением покадровой анимации, представляет собой последовательность ключевых кадров, с каждым из которых связано некоторое изображение (картинка) на столе.

При воспроизведении мультика эффект анимации достигается за счет того, что картинки на столе сменяют друг друга. Каждый покадровый мультик характеризуется двумя основными параметрами:

- количеством ключевых кадров (Keyframe);
- частотой смены кадров (во Flash она измеряется как число кадров, Показанных за одну секунду - frame per second, fps).

В общем случае оба эти параметра влияют на создаваемый визуальный эффект (на плавность или, наоборот, дискретность движений, «превращений» и т. д.). Но

все же ведущая роль здесь принадлежит первому параметру, а также тому, насколько сильно последующий ключевой кадр отличается от предыдущего.

На временной диаграмме ключевые кадры изображаются серыми прямоугольничками с черной точкой внутри. При воспроизведении фильма считывающая головка перемещается от одного кадра к другому, отмечая текущий кадр. Чтобы увидеть изображение, связанное с конкретным кадром, необходимо щелкнуть мышью на значке этого кадра на временной диаграмме. На рис. 78 показаны в качестве примера два ключевых кадра (первый и последний) небольшого мультика «про часы», в котором один кадр отличается от другого положением стрелок. Всего мультик содержит 6 кадров, частота смены кадров равна 2.

Для описания процедуры создания покадровой анимации воспользуемся другим примером. Предположим, что «героем» фильма будет бомба с подожженным фитилем. Будем считать, что для раскрытия столь сложного сюжета достаточно четырех кадров:

1. Бомба с зажженным фитилем.
2. Огонь перемещается по фитилю.
3. Огонь дошел до бомбы.
4. Взрыв.

С учетом описанного сюжета, последовательность работы должна быть следующей:

1. Создайте на столе четыре изображения, соответствующие указанным выше кадрам, как показано на рис. 79 (для первого и второго кадров используется одна и та же бомба, но с разной длиной фитиля).

2. Создайте новый файл фильма с «чистым» столом, щелкнув на кнопке *New* (Создать) основной панели инструментов Flash.

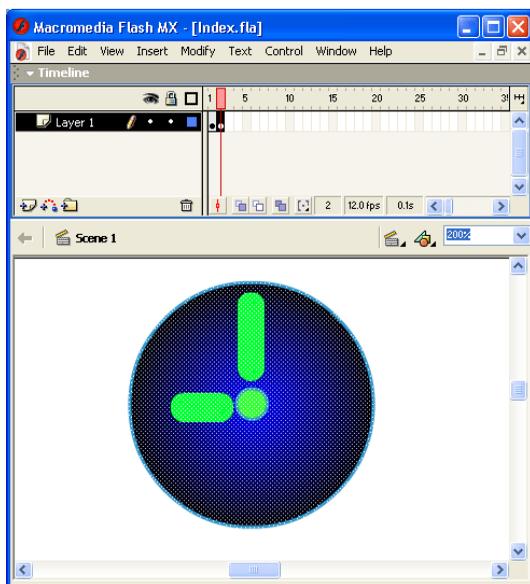
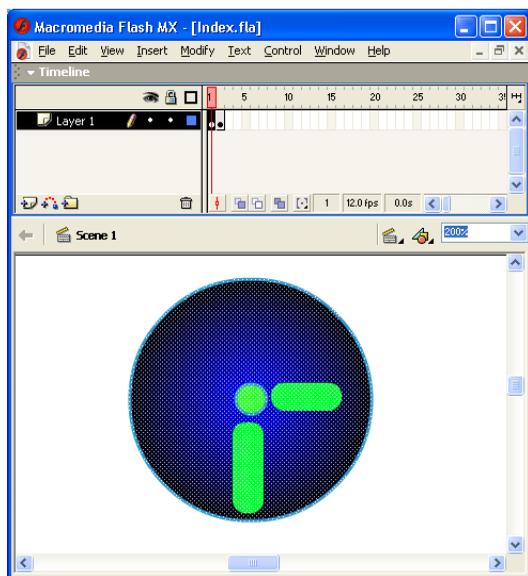


Рис. 78. Два кадра из мультлика «про часы»

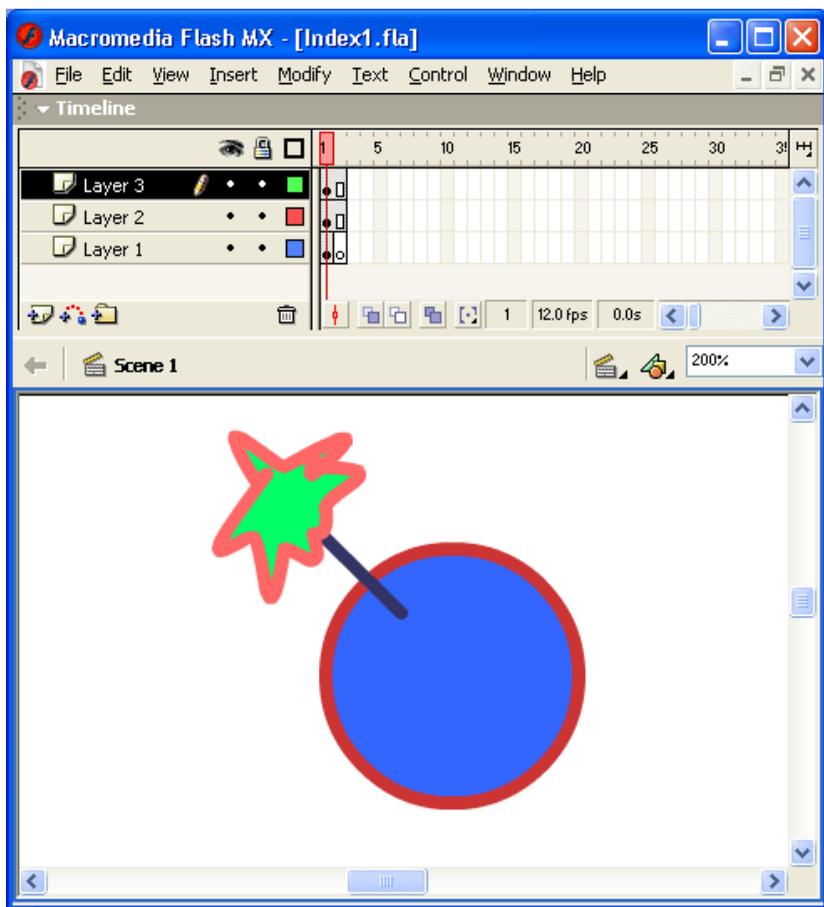


Рис. 79. Изображения, соответствующие кадрам будущего фильма

3. На временной диаграмме в ячейке первого кадра щелкните правой кнопкой мыши и в контекстном меню выберите команду *Insert Keyframe* (Вставить ключевой кадр).

4. Из окна, в котором создавались изображение бомбы, скопируйте первое изображение и поместите его где-нибудь в верхней части стола; при этом первый кадр на временной

диаграмме будет помечен как ключевой (черной точкой), рис. 80.

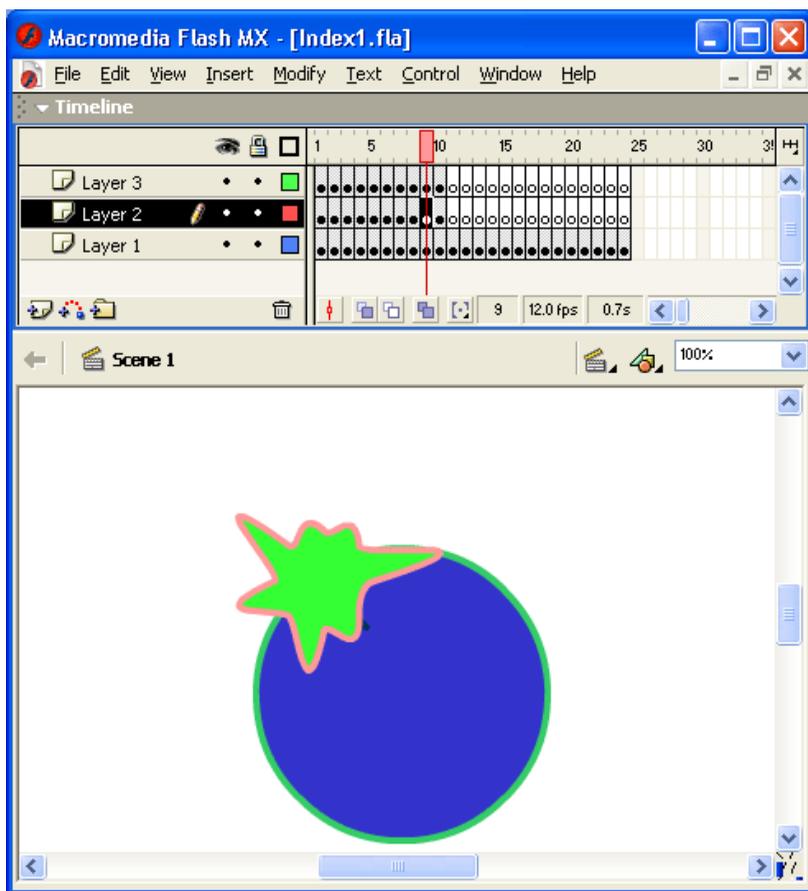


Рис. 80. Вид окна Flash после создания первого кадра

5. Щелкните правой кнопкой мыши в ячейке второго кадра и в контекстном меню выберите команду *Insert Keyframe*; при этом кадр будет сразу помечен как ключевой, поскольку для него наследуется изображение из предыдущего

кадра; воспользуйтесь им: переместите изображение огня ниже по фитилю.

6. Щелкните правой кнопкой мыши в ячейке третьего кадра и в контекстном меню вновь выберите команду *Insert Keyframe*; новый кадр также будет помечен как ключевой, и для него сохранится предыдущее изображение; вы можете либо отредактировать его, либо просто заменить нужным.

7. Создайте четвертый и пятый кадры, выбирая в контекстном меню команду *Insert Keyframe* и помещая на стол соответствующее изображение.

После выполнения завершающего шага окно Flash должно выглядеть примерно так, как показано на рис. 81.

Собственно, на этом создание первого «мультика» закончено. Не забудьте только сохранить его на диске.

После завершения работы полезно проверить, все ли получилось так, как вы хотели.

Чтобы воспроизвести фильм, достаточно просто нажать клавишу <Enter> (альтернативный вариант — выбрать в меню *Control* команду *Play* — «воспроизвести»). При этом считывающая головка панели временной диаграммы автоматически перемещается от текущего к последнему ключевому кадру. Вы можете принудительно выбрать любой кадр для просмотра, щелкнув в соответствующей ячейке временной диаграммы левой кнопкой мыши.

В рассмотренном нами примере изображения, составляющие содержание кадров, были созданы заранее, на столе другого фильма. Однако для большинства несложных фильмов удобнее создавать изображение кадра непосредственно на столе текущего фильма. Для этого необходимо:

1. Создать на временной диаграмме новый ключевой кадр.

2. Изменить свойства объектов, расположенных на столе (форму, положение и т. д.).

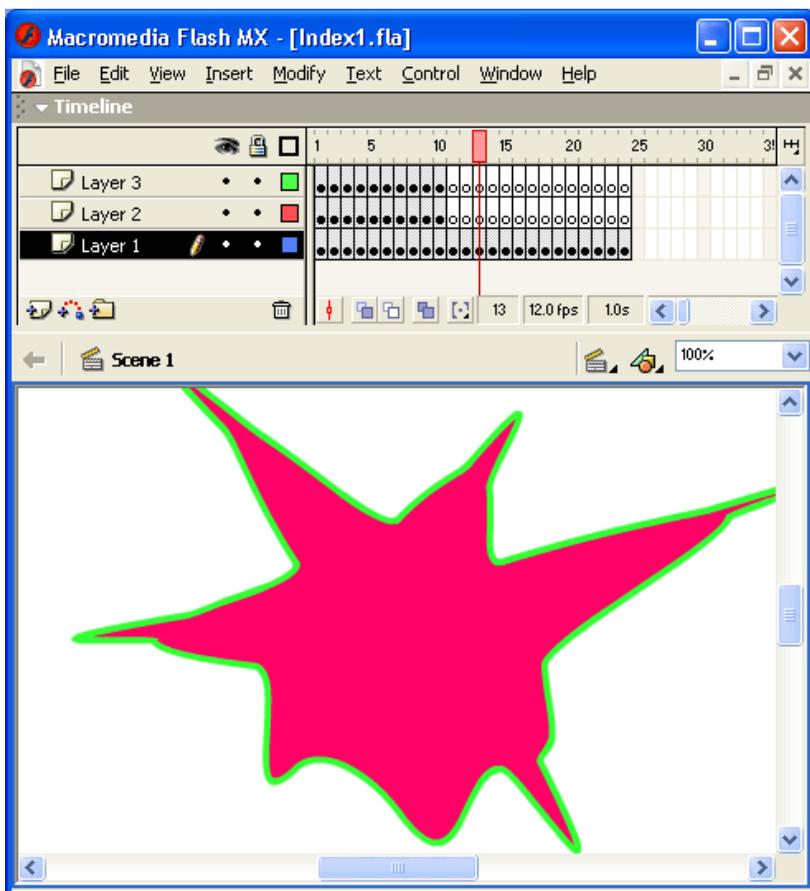


Рис. 81. Вид окна Flash после создания завершающего кадра анимации

6.10 Анимация трансформации и анимация движения

Flash может создавать два типа tweened-анимации:

- анимацию движения (motion tweening);
- анимацию трансформирования объекта (shape

tweening).

Анимация движения может быть автоматически построена для экземпляра символа, группы или текстового поля.

При создании анимации движения требуется установить для некоторого кадра такие атрибуты объекта, как позиция на столе, размер, угол поворота или наклона, и затем изменить значения этих атрибутов в другом кадре. Flash интерполирует значения изменяемых атрибутов для промежуточных кадров, создавая эффект последовательного перемещения или преобразования.

Вы можете создавать анимацию движения, используя один из двух методов:

- с помощью панели инспектора свойств кадра;
- с помощью команды *Create Motion Tween* (Создать анимацию движения).

Для описания обоих вариантов воспользуемся простейшим примером: предположим, что требуется «перекатить» шарик через стол слева направо.

Итак, для анимации движения шарика с помощью инспектора свойств кадра требуется выполнить следующие действия.

1. Убедитесь, что панель инспектора свойств присутствует на экране (ее формат на этом шаге значения не имеет). Если она закрыта, выберите в меню *Window* основного окна пункт *Properties*.

2. Включите инструмент *Oval* и нарисуйте шарик в левой части стола; обратите внимание, что первый кадр на временной диаграмме при этом стал помечен как ключевой.

3. Включите инструмент *Arrow*, выберите шарик и сгруппируйте (объедините в одно целое) контур и заливку шара с помощью команды *Group*, входящей в меню *Modify* основного окна Flash; в результате шар окажется заключен в голубую выделяющую рамку.

4. Щелкните правой кнопкой мыши в ячейке того кадра на временной диаграмме, который вы хотите сделать последним кадром анимации (например, 10-й), и в контекстном меню выберите команду *Insert Keyframe* (Вставить ключевой кадр); в результате промежуток между первым и последним ключевыми кадрами будет заполнен одноцветными (светло-серыми) ячейками простых кадров, как показано на рис. 82.

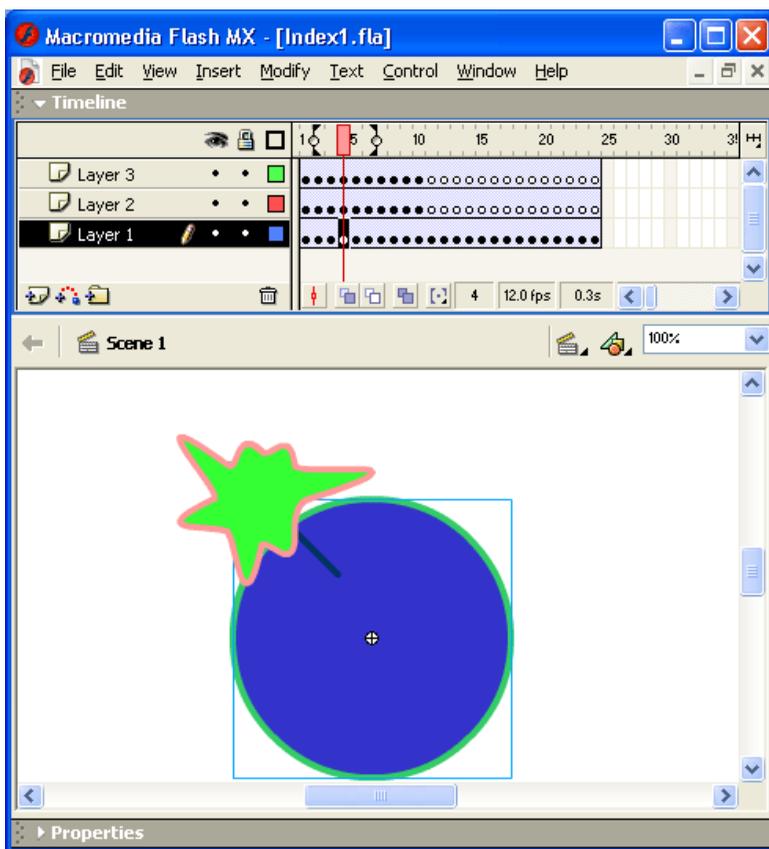


Рис. 82. Вид панели временной диаграммы после выполнения 4-го шага

5. Перетащите шар на новую позицию в правой части стола.

6. Щелкните левой кнопкой мыши в ячейке первого ключевого кадра; это приведет к одновременному выполнению двух действий: изображение шарика переместится на исходную позицию и изменится формат панели инспектора свойств: на ней будут представлены параметры выбранного (первого) кадра.

7. В панели инспектора свойств выберите в раскрывающемся списке *Tween* пункт *Motion*; при этом формат панели изменится, и на ней появятся элементы интерфейса, позволяющие установить параметры анимации; пока они нас не интересуют, значительно важнее изменения, произошедшие на временной диаграмме: первый и последний ключевые кадры теперь соединены стрелкой на сиреновом фоне (рис. 83). Это говорит о том, что создание анимации успешно завершено.

Если на панели временной диаграммы вместо стрелки появилась пунктирная линия, значит вы в чем-то ошиблись. Взгляните на панель инспектора свойств кадра: если там появилась кнопка с предупреждающим знаком (рис. 84), то Flash пытается сообщить вам об этом. Щелкните на этой кнопке, чтобы открыть окно с пояснениями по возникшей ситуации.

Возможны две основные причины неудачи: либо вы пытаетесь анимировать не сгруппированные объекты (в рассматриваемом примере — контур и заливку), либо данный слой содержит более одного сгруппированного объекта или символа.

Во многих случаях оказывается достаточным удалить лишний объект, чтобы Flash выполнил анимацию. Если этого не произошло, после исправления ошибки повторите процедуру создания tweened-анимации еще раз.

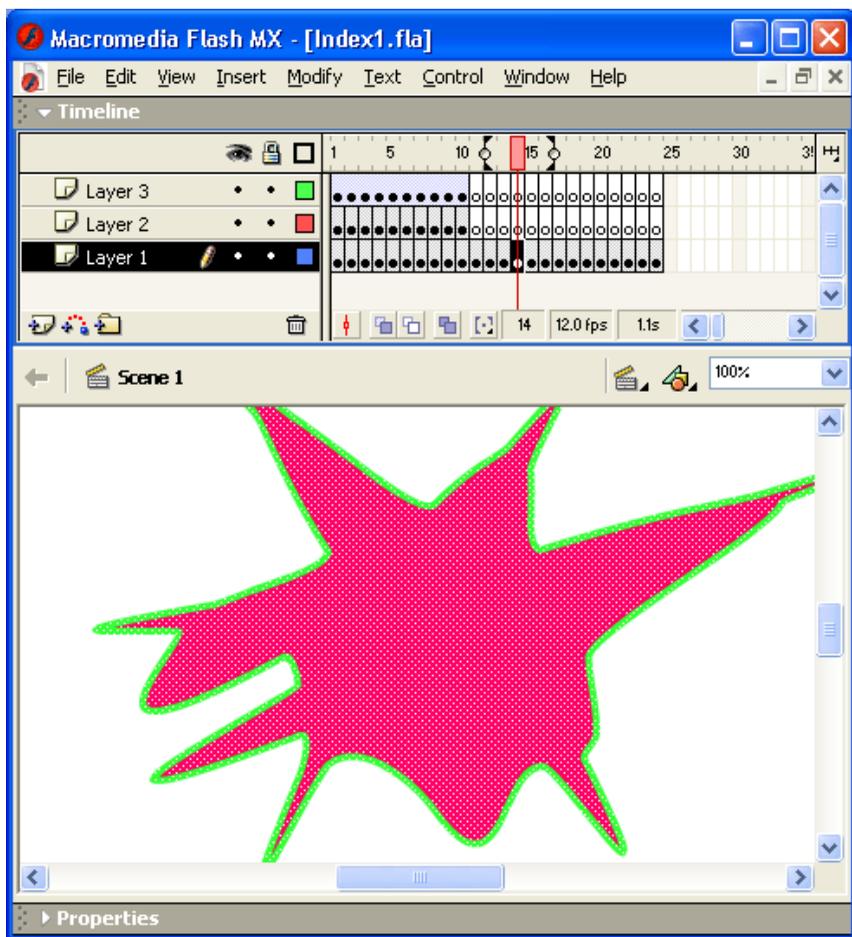


Рис. 83. Вид панели временной диаграммы после завершения создания анимации



Рис. 84. Вид панели инспектора свойств кадра при наличии ошибки

Для воспроизведения «мультика» используются те же средства, что и для покадровой анимации - можно просто нажать клавишу <Enter> (предварительно лучше снять выделение с шарика, хотя это не обязательно).

Для анимации движения шарика с помощью команды *Create Motion Tween* требуется выполнить следующие действия (считаем, что шарик в первом кадре уже присутствует).

1. Щелкните правой кнопкой в ячейке первого кадра и в контекстном меню выберите команду *Create Motion Tween* (Создать анимацию движения); при этом изображение шара будет автоматически преобразовано в графический символ с именем *tween1* (визуальным признаком преобразования служит появление точки привязки в центре шара и выделяющей рамки).

2. Щелкните правой кнопкой в ячейке кадра, который вы хотите сделать последним в анимационной последовательности (например, 10-й) и в контекстном меню выберите команду *Insert Frame* (Вставить кадр); в результате между первым и последним кадрами появится пунктирная линия, как показано на рис. 85.

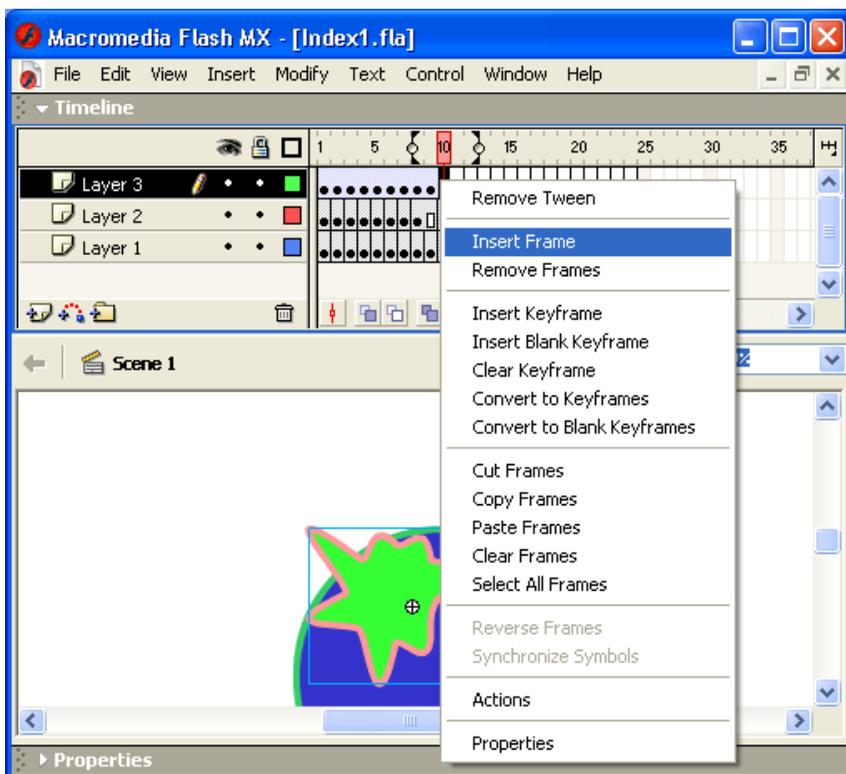


Рис. 85. Вид панели временной диаграммы после выполнения команд Insert Frame

3. Переместите шар на новую позицию (в правую часть стола); при этом последний кадр анимации будет автоматически преобразован в ключевой (в ячейке появится черная точка), а пунктирная линия будет заменена линией со стрелкой (рис. 86).

Создание анимации на этом завершено. Если вы теперь взгляните на панель инспектора свойств кадра, то увидите, что в списке *Tween* выбран вариант *Motion*.

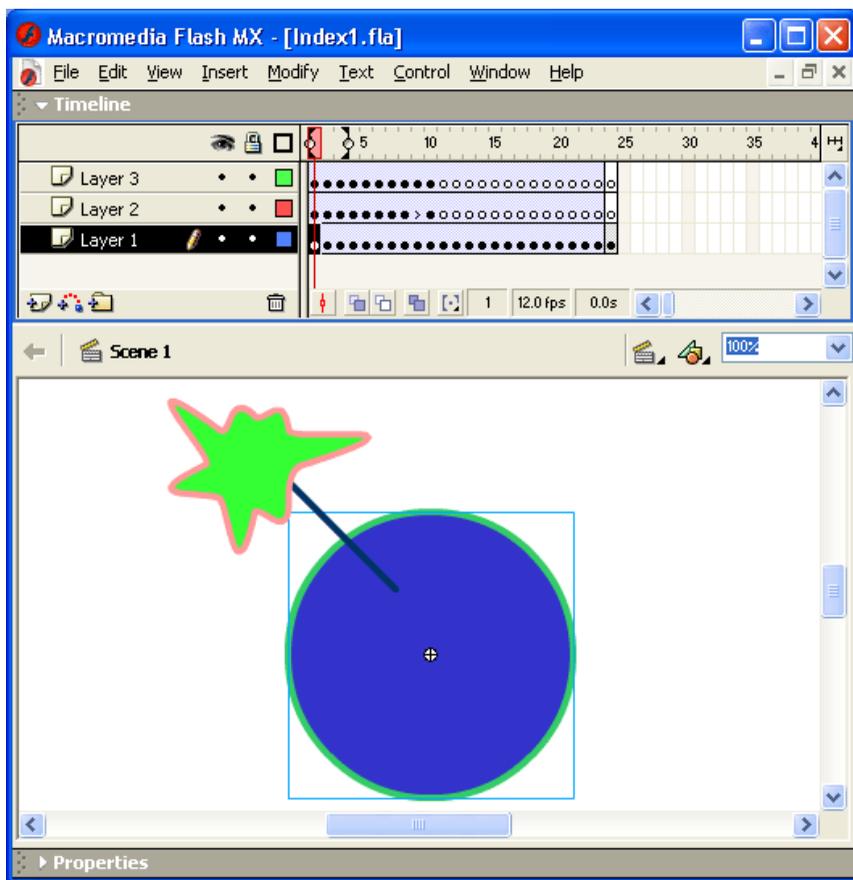


Рис. 86. Вид окна Flash после завершения создания анимации

6.11 Автоматическая анимация трансформации объекта

Используя анимацию трансформации, вы можете создавать эффект плавного «перетекания» объекта из одной формы в другую. Причем результирующая форма может не иметь абсолютно ничего общего с исходной. Например, в

электронной справочной системе Flash приведен пример «превращения»(рис. 87).

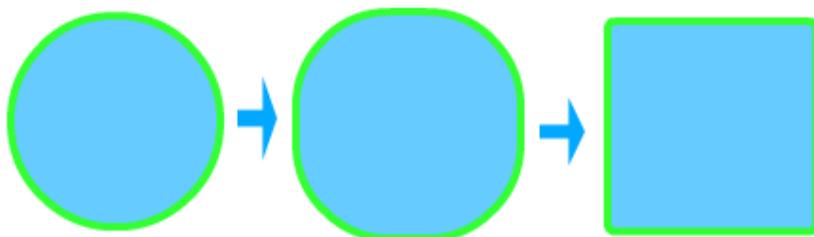


Рис. 87. Пример трансформации объекта

Еще раз повторим, что анимирование стандартных операций видоизменения объекта, которые выполняются с помощью инструментов панели *Tools* (наклон, изменение размера и т. п.) не рассматриваются во Flash как операции трансформации и могут быть добавлены в качестве «сопутствующих» изменений при создании анимации движения объекта.

Flash не может автоматически анимировать трансформацию символов, сгруппированных объектов, текстовых полей и растровых изображений. Чтобы сделать объекты этих типов доступными для автоматической трансформации, к ним требуется применить процедуру разбиения (*Break Apart*).

Если вы решили включить в свой фильм tweened-анимацию трансформации, имейте в виду следующее обстоятельство. По умолчанию Flash пытается произвести переход от одной формы к другой «кратчайшим путем». Поэтому промежуточные кадры могут оказаться весьма неожиданными для вас. В качестве иллюстрации к сказанному ниже приведен рисунок (рис. 88), на котором показан промежуточный кадр трансформации «часов». Здесь с помощью tweened-анимации минутная стрелка пытается

«переползти» с 12 на 6 «короткой дорогой».

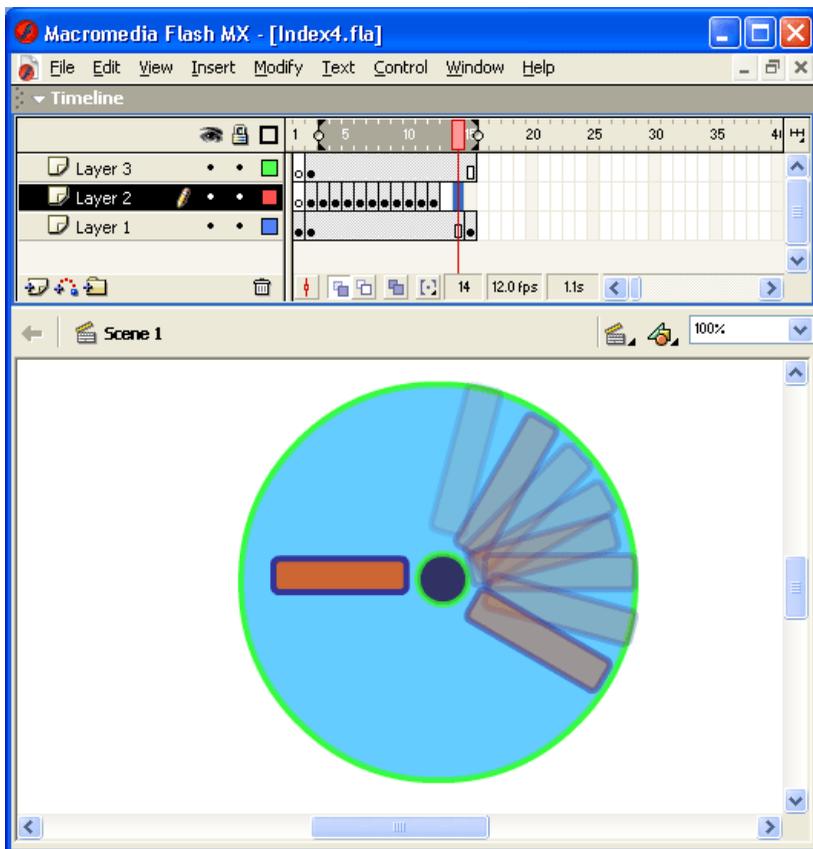


Рис. 88. Пример промежуточного кадра при tweened анимации трансформации

В отличие от анимации движения, tweened-анимация трансформации может быть создана только одним способом — с помощью панели инспектора свойств кадра.

Для пояснения технологии создания такой анимации вернемся к примеру с «часами»: предположим, что требуется переместить минутную стрелку с «12» на «6».

При решении указанной задачи требуется выполнить следующие действия.

1. Убедитесь, что панель инспектора свойств присутствует на экране. Если она закрыта, выберите в меню *Window* основного окна пункт *Properties*.

2. Включите инструмент *Oval* и нарисуйте с его помощью все элементы «часов», показанные на рис. 88, слева.

3. Включите инструмент *Arrow* и выберите изображение, которое вы хотите трансформировать.

4. Щелкните правой кнопкой в ячейке того кадра, который вы хотите сделать последним в анимации (например, 10-й), и в контекстном меню выберите команду *Insert Keyframe*; в результате между первым и последним кадрами на временной диаграмме появится последовательность простых кадров серого цвета, как показано на рис. 89, вверху.

5. Замените на столе исходное изображение тем, в которое оно должно быть трансформировано; в рассматриваемом примере для этого достаточно с помощью инструмента *Free Transform* повернуть минутную стрелку и переместить на новую позицию.

6. Щелкните правой кнопкой в ячейке первого кадра анимации; это приведет к двум последствиям: на столе появится исходное изображение и изменится формат панели инспектора свойств.

7. В раскрывающемся списке *Tween* выберите вариант *Shape* (Форма); в результате первый и последний кадры анимации на временной диаграмме будут соединены стрелкой на светло-зеленом фоне (рис. 89, внизу); это говорит о том, что создание анимации успешно завершено.

Выбор пункта *Shape* в списке *Tween* приводит к изменению формата панели инспектора свойств. С ее помощью могут быть установлены следующие дополнительные параметры анимации трансформирования объекта (рис. 90):

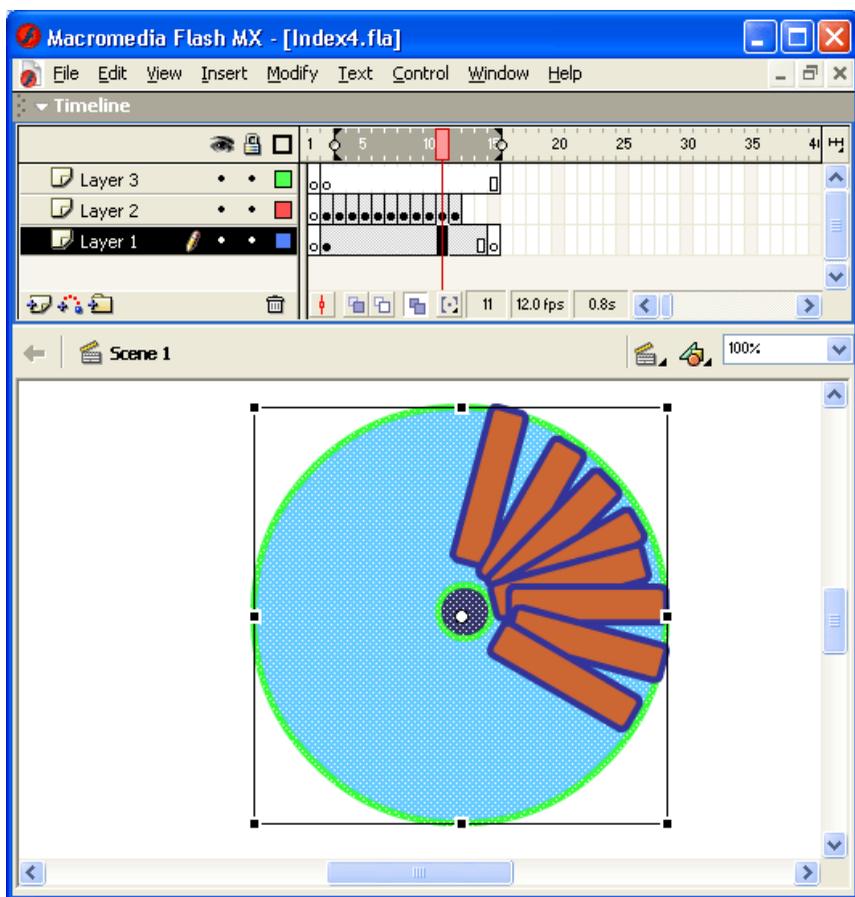


Рис. 89. Вид временной диаграммы при анимировании трансформации объекта

- скорость изменений; по умолчанию изменения протекают с постоянной скоростью, однако вы можете управлять ею, изменяя значения параметра *Ease* (Замедление): отрицательные значения этого параметра (от -1 до -100) означают, что изменения будут постепенно

ускоряться, а положительные значения (от 1 до 100), наоборот, означают постепенное замедление изменений;

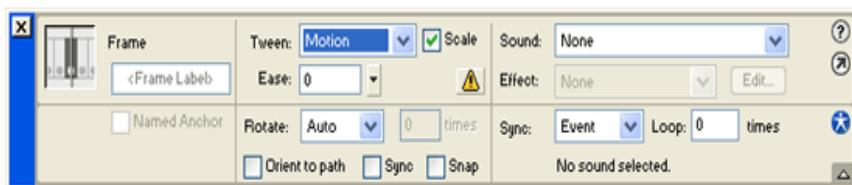


Рис. 90. Параметры анимации трансформирования объекта

- способ трансформации; он определяется значением, выбранным в списке *Blend* (Переход):

- *Distributive* (Размазанный) — Flash создает анимацию, в которой промежуточные формы являются более сглаженными;

- *Angular* (Угловатый) -- Flash создает анимацию, при которой в промежуточных формах сохраняются очевидные углы и прямые линии.

Обратите внимание, что для создания анимации трансформации объекта не требуется (более того - **нельзя**) выполнять группирование контура и заливки, равно как и других элементов изображения, подлежащих трансформированию. Достаточно все эти элементы просто выбрать с помощью инструмента *Arrow*.

Для управления более сложными изменениями формы используются так называемые узловые точки формы (*shape hints*), которые определяют, как фрагменты первоначальной формы будут перенесены в новую форму. Другими словами, узловые точки используются для идентификации тех точек исходной формы, взаимное расположение которых требуется сохранить. Наиболее характерный пример использования узловых точек — анимация мимики лица, при которой

некоторые его части (глаза, в частности) не должны участвовать в трансформации. На рис. 91 показаны два варианта трансформации — без расстановки и с расстановкой узловых точек. Во втором варианте глаза трансформации не подвергаются.

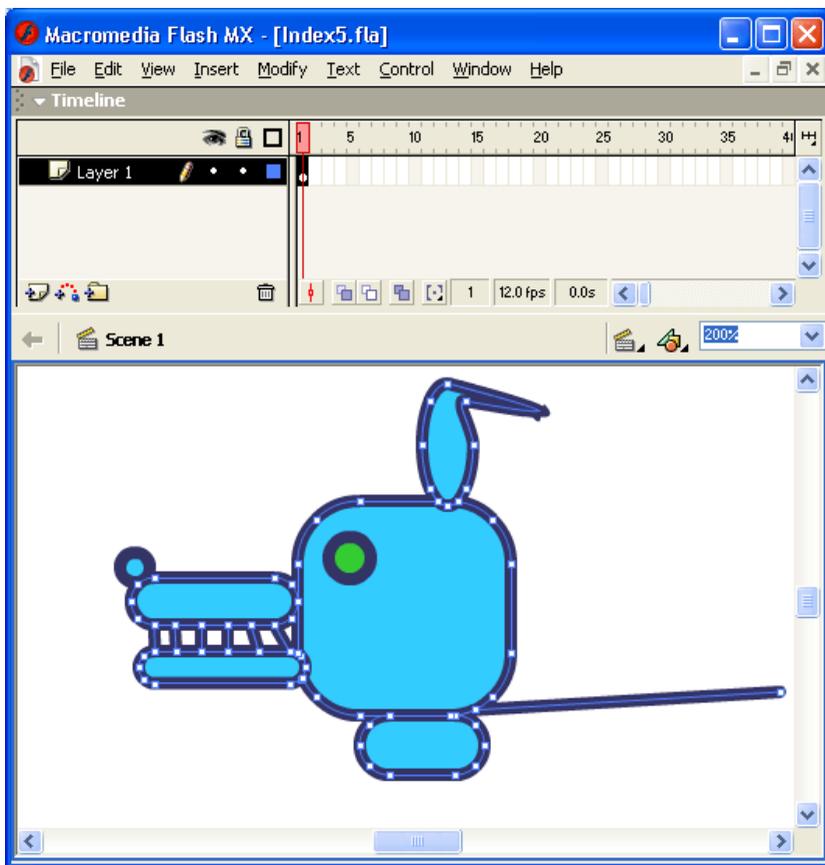


Рис. 91. Пример трансформации без расстановки и с расстановкой узловых точек

Узловые точки обозначаются на изображении

небольшими кружками с буквами. Буквы (от *a* до *z*) используются в качестве имен (идентификаторов) узловых точек. Каждой узловой точке на исходном изображении должна соответствовать точка с тем же именем на результирующем изображении. Всего для одной фигуры может быть использовано не более 26 узловых точек (по числу букв алфавита). Узловые точки на исходном изображении окрашены в желтый цвет, на результирующем — зеленым.

Для расстановки узловых точек требуется выполнить следующие действия:

1. Щелкните левой кнопкой мыши в ячейке кадра, соответствующего исходному изображению.

2. В меню *Modify* выберите каскадное меню *Shape*, а в нем — команду *Add Shape Hint* (Добавить узловую точку); в результате на изображении появится «заготовка» первой узловой точки - кружок красного цвета с буквой *a*.

3. Переместите его мышью на ту точку изображения, которую вы хотите пометить как узловую.

4. Щелкните левой кнопкой мыши в ячейке кадра, соответствующего результирующему изображению; на столе будет присутствовать красный кружок с той же буквой, что и в исходном кадре.

5. Переместите кружок в точку изображения, которая должна соответствовать отмеченной в исходном кадре; после перемещения кружок изменит цвет на зеленый.

6. Вернитесь на первый кадр анимированной последовательности и убедитесь, что цвет узловой точки изменился на желтый.

7. Если требуется продолжить расстановку узловых точек, повторите описанную процедуру для каждой из них.

Расстановка узловых точек возможна только после создания *tweened*-анимации трансформации.

Расстановку узловых точек следует выполнять с

помощью инструмента Arrow при включенном модификаторе Snap to Objects.

Если вы используете при трансформации несколько узловых точек, то выполняйте их расстановку по часовой стрелке, начиная с левого верхнего угла изображения. Это обусловлено тем, что Flash обрабатывает узловые точки в алфавитном порядке.

После расстановки узловых точек целесообразно провести тестовый запуск фильма и при необходимости скорректировать расстановку точек. Чтобы изменить положение узловой точки, достаточно просто перетащить ее на новое место (это можно сделать как на первом, так и на последнем кадре последовательности). Кроме того, всегда можно удалить лишние точки или добавить новые. После того, как результат вас удовлетворит, вы можете спрятать (скрыть) узловые точки. Перечисленные операции удобнее всего выполнять с помощью контекстного меню. Чтобы его открыть, следует щелкнуть на одной из узловых точек правой кнопкой мыши. Меню содержит четыре команды (рис. 92), из которых первые три доступны только для узловых точек исходного изображения:

- *Add Hint* (Добавить узловую точку) — при выполнении команды на столе появляется заготовка новой узловой точки;

- *Remove Hint* (Удалить узловую точку) — узловая точка, на которой вы щелкнули правой кнопкой мыши, открывая меню, будет удалена;

- *Remove All Hints* (Удалить все узловые точки) — удаление всех узловых точек;

- *Show Hints* (Показать узловые точки) — данный режим используется по умолчанию (рядом с именем команды стоит маркер); повторный ее выбор приводит к тому, что узловые точки станут невидимы; этот вариант следует использовать только после достижения требуемого

результата, поскольку для возврата в режим показа узловых точек вам придется выбрать в меню *Modify* команду *Add Shape Hint*. необходимого количества.

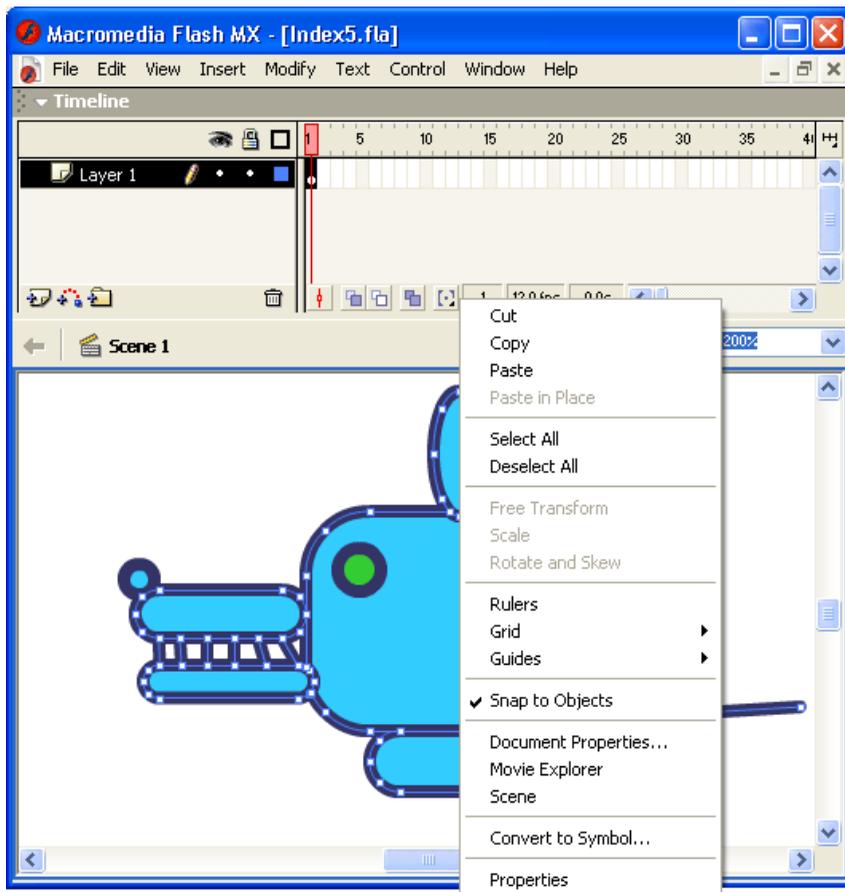


Рис. 92. Контекстное меню узловой точки

6.12 Публикация и экспорт фильма. Параметры публикации HTML-документа

Основным форматом Flash-фильма, который обеспечивает его просмотр с помощью Flash-плеера (либо автономно, либо через окно Web-браузера), является формат SWF. Это единственный формат, который поддерживает все интерактивные возможности фильма. Тем не менее, во многих случаях при размещении Flash-фильма на сервере SWF-файл должен быть дополнен и другими файлами. Прежде всего это файл HTML-документа, посредством которого производится загрузка SWF-файла в браузер: сначала в браузер загружается HTML-файл, содержащий вызов Flash-плеера, а тот, в свою очередь, уже открывает SWF-файл. Кроме того, необходимо предусмотреть ситуацию, когда у посетителей вашего файла отсутствует Flash-плеер. В этом случае полноценный фильм можно заменить последовательностью изображений в графических форматах, поддерживаемых браузером посетителя (например, JPEG, PNG или анимированным GIF).

Чтобы создать все файлы, необходимые для размещения фильма на Web-сервере, достаточно единственной команды *Publish* (Опубликовать), входящей в меню *File*. Она обеспечивает конвертирование исходного FLA-файла не только в формат SWF, но и в альтернативные графические форматы. Кроме того, с помощью этой команды генерируется и HTML-документ, предназначенный для запуска фильма с заданными параметрами.

Если вы достаточно хорошо знакомы с HTML, то предварительно можете скорректировать шаблон, на основе которого выполняется генерация HTML-документа. Для редактирования шаблона требуется использовать внешний (по отношению к Flash) HTML-редактор.

Если необходимо экспортировать FLA-файл в

единственный файл определенного типа, целесообразно использовать команду *Export* (Экспорт), которая также входит в меню *File*.

Установка параметров публикации При тестировании фильма, а также при публикации его с помощью команды *Publish* используются параметры публикации, установленные по умолчанию. Чтобы установить собственные значения этих параметров, следует воспользоваться командой *Publish Settings* (Параметры публикации) из меню *File*. Выбор данной команды приводит к открытию диалогового окна, с помощью которого и выполняются требуемые изменения.

В исходном состоянии окно содержит три вкладки (рис. 93):

- *Formats* (Форматы) — данная вкладка предназначена для выбора форматов файлов, которые должны быть созданы при публикации фильма; дополнительно вы можете ввести собственное имя для каждого генерируемого файла, сняв флажок *Use default names* (Использовать имена, установленные по умолчанию); если необходимо, можно указать полный маршрут доступа к файлу, причем для каждого файла свой; при выборе одного из дополнительных форматов в окне *Publish Settings* создается соответствующая вкладка; если выбран какой-либо графический формат (GIF, JPEG, PNG), то в HTML автоматически будет добавлен соответствующий текст, обеспечивающий при отсутствии Flash-плеера замену фильма графикой;
- *Flash* — данная вкладка обеспечивает установку параметров экспорта файла FLA в формат SWF;
- *HTML* - элементы этой вкладки обеспечивают выбор некоторых дополнительных параметров размещения фильма на HTML-странице.

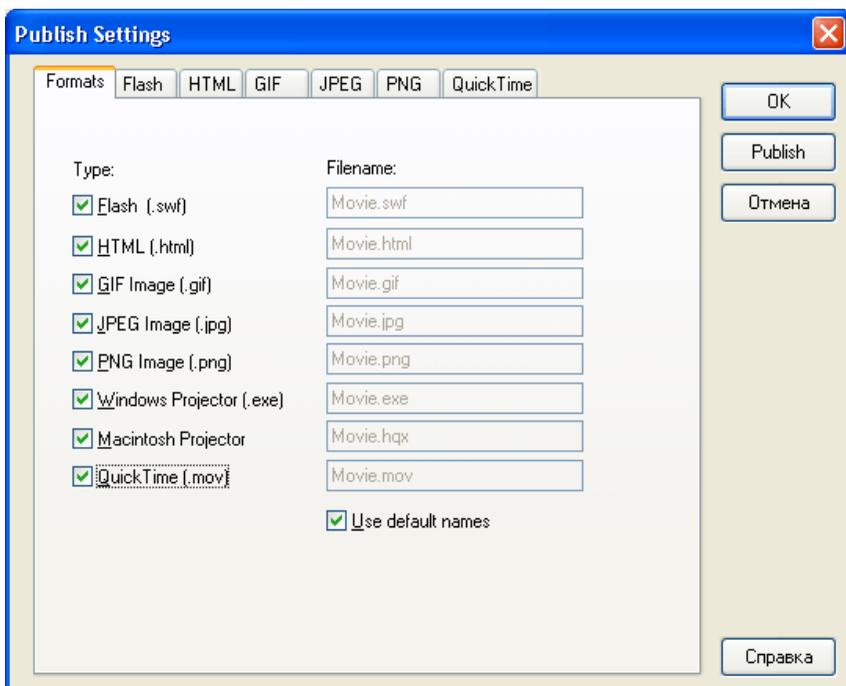


Рис. 93. Окно установки параметров публикации

Кнопка *OK* обеспечивает сохранение установленных параметров без инициализации команды *Publish*; введенные значения будут использоваться по умолчанию для всех последующих публикаций.

Кнопка *Publish*, помимо сохранения установленных параметров, инициализирует выполнение команды *Publish*.

Вкладка *Flash* содержит следующие элементы (рис. 94):

- Раскрывающийся список *Version* позволяет выбирать версию Flash-плеера, для которой должен быть сгенерирован SWF-файл;
- Раскрывающийся список *Load Order* (Очередность загрузки), который позволяет выбрать очередность загрузки слоев первого кадра фильма: либо снизу вверх (*Bottom up*),

либо сверху вниз (*Top down*);

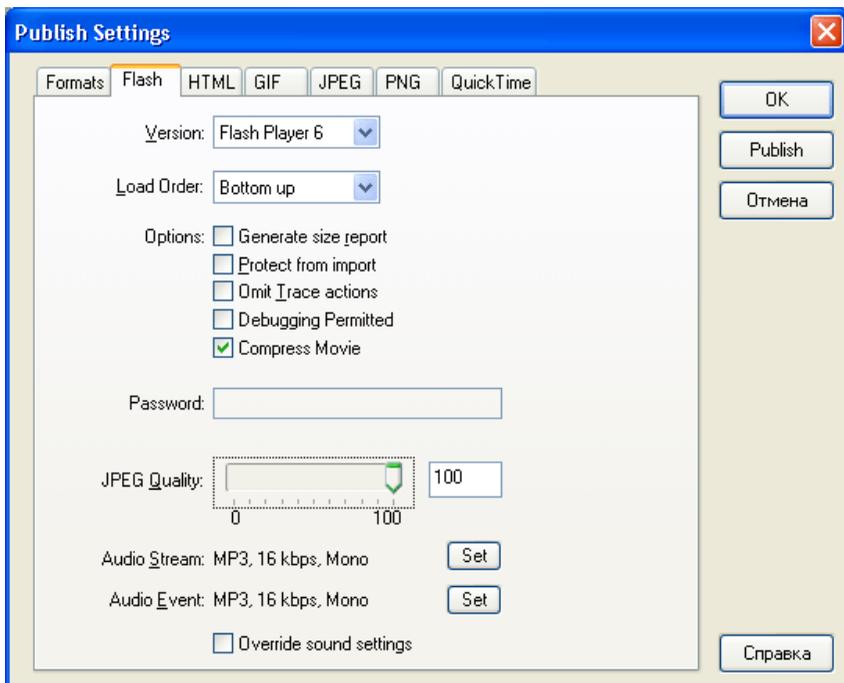


Рис. 94. Вкладка Flash

- Флажок *Generate Size Report* (Создать отчет о размере); если он установлен, то при публикации фильма создается отчет о размерах кадров фильма; отчет сохраняется в отдельном текстовом файле;
- Флажок *Omit Trace Actions* (Отменить действие *Trace*); если он установлен, то Flash игнорирует действие *Trace*, которое используется при отладке сценариев на ActionScript;
- Флажок *Protect from Import* (Защитить от импорта); установка флажка препятствует импорту SWF-файла из окна плеера (или браузера) в формат FLA; при установке флажка

целесообразно также ввести в текстовом поле *Password* пароль, позволяющий (как бы) защитить ваш фильм от несанкционированного доступа;

- Флажок *Debugging Permitted* (Разрешить отладку); установка флажка активизирует отладчик и разрешает удаленную отладку фильма (то есть после его размещения на Web-сервере); в этом случае вы можете ввести в расположенном ниже текстовом поле *Password* пароль, чтобы защитить фильм от несанкционированного доступа;

- Флажок *Compress Movie* (Сжатие фильма), если установлен, указывает на необходимость включения алгоритма дополнительного сжатия фильма при его экспорте в формат SWF;

- Ползунковый регулятор *JPEG Quality* (Качество JPEG-изображения) и связанное с ним текстовое поле обеспечивают выбор приемлемого (на ваш взгляд) качества JPEG-изображений, используемых в фильме;

- Кнопки *Set* (Установить) позволяют изменить параметры звукового сопровождения фильма;

- Флажок *Override Sound Settings* (Изменять параметры звука), если установлен, разрешает индивидуальную установку параметров для отдельных элементов звукового сопровождения; это позволяет создавать две версии звукового сопровождения: лучшего качества (но большего размера) — для локального использования и более низкого качества — для публикации в Интернете.

Вкладка *HTML* содержит следующие элементы (рис. 95):

- раскрывающийся список *Template* (Шаблон) позволяет выбрать тип шаблона, в соответствии с которым будет сгенерирован HTML-документ; список содержит следующие основные варианты (всего их более десятка):

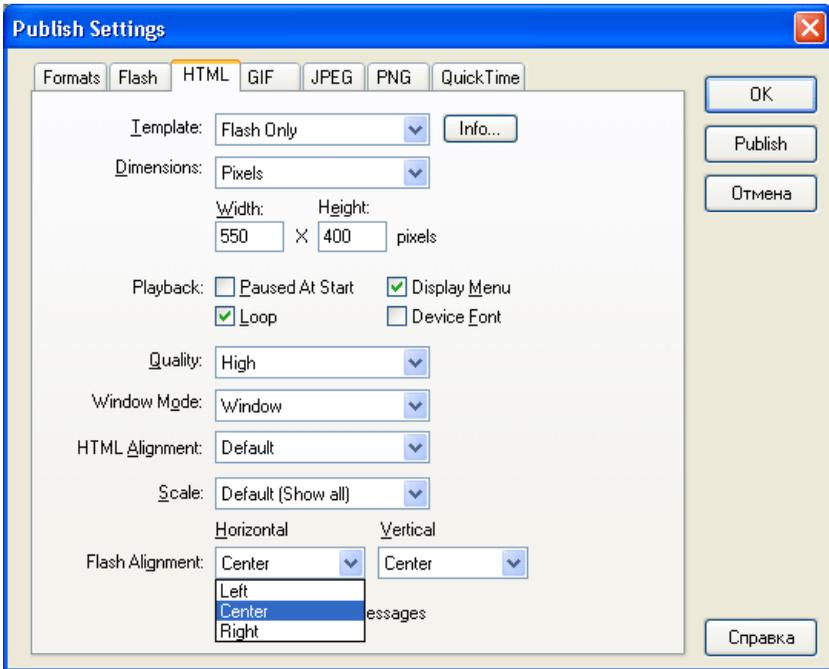


Рис. 95. Вкладка HTML

- *Flash Only* (Только Flash) - шаблон обеспечивает включение в HTML-документ только тех тэгов, которые необходимы для загрузки фильма (то есть тэгов <OBJECT> и <EMBED>); данный шаблон используется по умолчанию;

- *Flash with Named Anchor* (Flash с именованными якорями) - шаблон обеспечивает включение в HTML-документ дополнительных гиперссылок, на основе которых может выполняться навигация между кадрами фильма;

- *Image Map* (Сенсорная карта) — шаблон обеспечивает включение в HTML-документ тэга с параметрами, позволяющими создавать клиентскую

сенсорную карту; при выборе данного шаблона необходимо на вкладке *Formats* разрешить генерацию файла в одном из графических форматов;

- *Quit Time* — шаблон позволяет включить в HTML-документ Flash-фильм в виде проектора в формате *Quit Time*;

- раскрывающийся список *Dimensions* (Размеры) совместно с полями *Width* и *Height* позволяет выбрать способ измерения и размеры окна фильма (в данном случае более уместным было бы слово «экран»);

- флажок *Paused at Start* (Пауза перед запуском); если он установлен, то фильм после загрузки не воспроизводится, пока пользователь не инициализирует воспроизведение сам; по умолчанию воспроизведение фильма начинается сразу после его загрузки;

- флажок *Loop* (Цикл); если установлен, то воспроизведение фильма циклически повторяется, пока страница не будет закрыта;

- флажок *Display Menu* (Показывать меню); установка флажка разрешает использование контекстного меню клипа, вызываемого щелчком правой кнопкой мыши;

- флажок *Device Font* (Физический шрифт); его установка разрешает применение функции сглаживания (*anti-aliasing*) к шрифтам, не установленным на компьютере пользователя;

- раскрывающийся список *Window Mode* (Режим окна) предназначен для выбора варианта представления содержимого фильма на Web-странице:

- *Window* — Flash-плеер использует собственное прямоугольное окно, отображаемое внутри окна браузера; этот вариант обеспечивает наибольшую скорость воспроизведения;

- *Opaque Windowless* (Непрозрачный, без окна) — фильм воспроизводится непосредственно на

странице, «без экрана», при этом он может заслонять другие объекты страницы;

- *Transparent Windowless* (Прозрачный, без окна) — вариант отличается от предыдущего тем, что через фильм «просвечивают» закрытые им другие элементы страницы.

Чтобы оценить установленные вами параметры публикации, целесообразно воспользоваться командой *Publish Preview* (Предварительный просмотр публикации), входящей в меню *File*. Данная команда по сути представляет собой заголовок каскадного меню, состав которого изменяется в зависимости от того, какие форматы публикации были установлены на вкладке *Formats* диалогового окна *Publish Settings*. Например, если помимо основных форматов (SWF и HTML) были выбраны графические форматы GIF и JPEG, меню будет выглядеть, как показано на рис. 96. Если вы выберете вариант GIF, то это приведет к загрузке в браузер фильма в формате GIF.

Публикация фильма в дополнительных форматах В предыдущем разделе было отмечено, что Flash обеспечивает возможность публикации фильма в одном из графических форматов, поддерживаемых HTML — GIF, JPEG и PNG. Такая возможность полезна в тех случаях, если вы не уверены, что потенциальные посетители вашего сайта располагают Flash-плеером, либо достаточно быстрым соединением с Интернетом.

Для любого из указанных форматов вы можете указать требуемые параметры, которые на ваш взгляд в наибольшей степени соответствуют характеру изображения и предполагаемой аудитории. Для этого необходимо:

1. Открыть диалоговое окно *Publish Settings* и перейти на вкладку *Formats*.
2. Поставить один или несколько флажков, соответствующих тем форматам, для которых вы хотите

изменить параметры публикации; при этом окно *Publish Settings* будет дополнено новыми вкладками (рис. 97).



Рис. 96. Состав меню Publish Prewiev при использовании дополнительных форматов публикации фильма

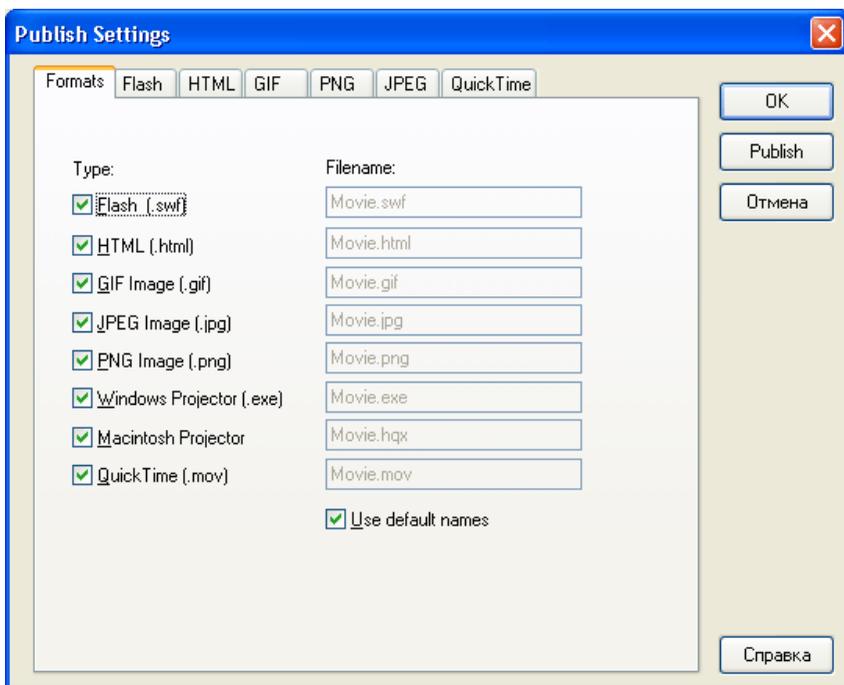


Рис. 97. Дополнительные вкладки для публикации фильма в графических форматах

3. Перейти на требуемую вкладку и установить параметры публикации.

В частности, для формата GIF на соответствующей вкладке могут быть заданы (рис. 98):

- геометрические размеры изображения; они устанавливаются с помощью полей *Width Height*; если поставить флажок *Match Movie*, то будут использованы размеры стола фильма;
- тип изображения (статическое или анимированное); выбор выполняется с помощью пары переключателей *Playback*; для ани-мированного GIF можно дополнительно определить цикличность воспроизведения:

- переключатель *Loop Continuously* означает непрерывное воспроизведение;
- переключатель *Repeat* связан с полем *times*, в котором вводится количество повторений;

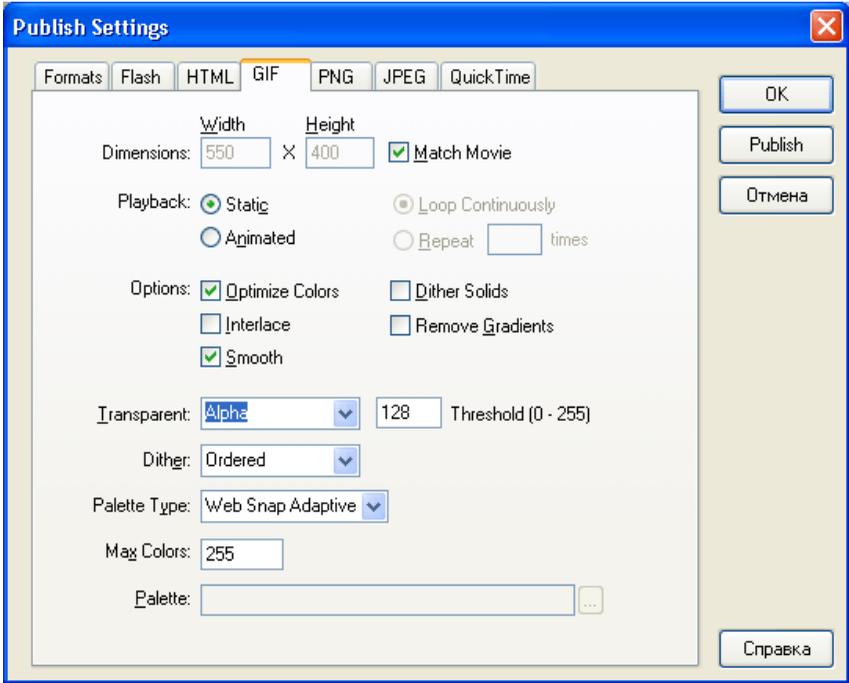


Рис. 98. Вкладка GIF окна Publish Settings

- уровень прозрачности изображения; выбирается из раскрывающегося списка *Transparent*;
- цветовая палитра; выбор палитры производится с помощью раскрывающегося списка *Palette Type*; при выборе варианта *Custom* вы можете указать файл, содержащий используемую палитру.

На основе любого из графических форматов вы можете создать так называемую карту-изображение (или сенсорную

карту) — Image Map. Сенсорные карты на Web-страницах достаточно хорошо зарекомендовали себя как удобное средство навигации по сайту. Для создания сенсорной карты требуется соответствующим образом настроить шаблон публикации.

Использование шаблонов Шаблон представляет собой текстовый файл, который содержит «заготовку» для создания HTML-документа, обеспечивающего загрузку Flash-фильма в окно браузера. От обычного HTML-документа шаблон отличается тем, что в нем используется специальный заголовок и так называемые переменные шаблона. Переменная шаблона - это предопределенная последовательность символов, начинающаяся со знака доллара \$, например, \$LO.

При генерации HTML-документа Flash заменяет переменные шаблона значениями параметров, указанных в окне *Publish Settings*. Если вы достаточно хорошо знаете HTML, то можете открыть файл шаблона в текстовом редакторе (например, в *Блокноте*) или в HTML-редакторе и скорректировать его таким образом, чтобы получился требуемый HTML-документ. В частности, вы можете добавить в него сценарии на JavaScript или VBScript. Если в шаблоне не останется ни одной переменной, Flash вообще ничего не будет изменять в этом файле.

Все файлы шаблонов находятся в папке *Flash MX\First Run\HTML* и имеют расширение *html*.

На рис. 99 показан шаблон *Default.html*, открытый в редакторе *Блокнот*.

В таблице 1 приведены переменные шаблона для некоторых параметров публикации фильма.

```

1.htm - Блокнот
Файл Правка Формат Вид Справка
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>1.htm</TITLE>
<META HTTP-EQUIV="CONTENT-TYPE"
CONTENT="TEXT/HTML; CHARSET=WINDOWS-1251">
<META NAME="GENERATOR" CONTENT="ABBY FineReader
6.0">
</HEAD>
<body background=../FON.JPG>
<ul>
<li><font face="Arial, Helvetica,
sans-serif"><b><font size=3><a href="#1">Глава
12. Публикация фильма </a></font></b>
</font></li>
<ul>
<li><font face="Arial, Helvetica, sans-serif"
size="3"><a href="#2">Оптимизация
и тестирование фильма</a></font> </li>
<li><font face="Arial, Helvetica, sans-serif"
size="3"><a href="#3">Установка
параметров публикации</a></font> </li>
<li><font face="Arial, Helvetica, sans-serif"
size="3"><a href="#4">Публикация

```

Рис. 99. Пример шаблона

Таблица 1. Переменные шаблона для некоторых параметров публикации фильма

Параметр публикации	Переменная шаблона
Заголовок шаблона	\$TT
Начало описания шаблона	\$DS
Конец описания шаблона	\$DF
Заголовок HTML-документа	\$TI

Цвет фона HTML-документа	\$BG
Ширина окна фильма	\$WI
Высота окна фильма	\$HE
Имя файла фильма (.swf)	\$MO
Список URL фильма	\$MU
Текст	\$MT
Параметры для тэга <OBJECT>	\$PO
Параметры для тэга <EMBED>	\$PE
Ширина GIF-изображения	\$OW
Высота GIF-изображения	\$GH
Ширина JPEG-изображения	\$JW
Высота JPEG-изображения	\$JH
Режим окна	\$WM

Из приведенных в таблице переменных шаблона дополнительного пояснения требуют две — $\$MU$ и $\$MT$.

Переменная $\$MU$ указывает Flash, что необходимо сформировать список URL, используемых в фильме, и поместить его в текст документа в виде комментария. Такой список может оказаться полезным при проверке корректности используемых URL.

Переменная $\$MT$ указывает Flash, что требуется вставить весь текст из фильма в тело HTML-документа в виде комментария. Это позволит поисковым машинам проиндексировать содержимое фильма и в последующем использовать ключевые слова при поиске.

6.13 Основы создания интерактивных фильмов в пакете FLASH-MX. Создание элементов интерфейса

В простом анимированном фильме (наподобие

рассмотренных в предыдущих главах) Flash воспроизводит кадры временной диаграммы в определенной последовательности, которая остается неизменной, сколько бы раз вы не просматривали фильм. В интерактивном фильме читатель (или зритель, как вам будет угодно) имеет возможность использовать клавиатуру, мышь или то и другое, чтобы перейти к некоторому фрагменту или сцене фильма, переместить объекты, ввести информацию, а также выполнить многие другие интерактивные операции.

Интерактивность Flash-фильма обеспечивается за счет включения в него так называемых сценариев, которые представляют собой набор инструкций на языке ActionScript. Каждая инструкция инициируется при наступлении определенного связанного с ней события. События могут быть самые разнообразные: достижение считывающей головкой определенного кадра, нажатие пользователем клавиши на клавиатуре, щелчок кнопкой мыши и т.д.

Подобно другим современным языкам сценариев (типа JavaScript или VBScript), ActionScript - это объектно-ориентированный язык программирования. Объекты в ActionScript могут содержать данные или их графическое представление в виде символа определенного типа. Для создания сложных сценариев или новых типов объектов нужно иметь некоторые навыки в программировании. Вместе с тем, для создания несложных интерактивных фильмов совсем необязательно изучать язык ActionScript, поскольку события и процедуры могут быть назначены элементам фильма с помощью диалоговых окон и панелей свойств.

В отличие от предшествующих версий, в Flash MX появился вполне «дееспособный» редактор сценариев, и даже имеется встроенный отладчик, который позволяет обнаружить ошибки, допущенные при использовании синтаксических конструкций языка.

Создание сценария для кнопки Кнопка является

наиболее удобным и простым средством взаимодействия пользователя с Flash-фильмом. Кнопке может быть назначено любое из действий, входящих в раздел *Actions*. Кроме того, назначив кнопке несколько управляющих событий и сопоставив каждому из них определенное действие (или сценарий), вы можете превратить одну кнопку в целый пульт управления фильмом. Хотя в общем случае, разумеется, каждому действию можно сопоставить отдельную кнопку. Как, например, в фильме (рис. 100).

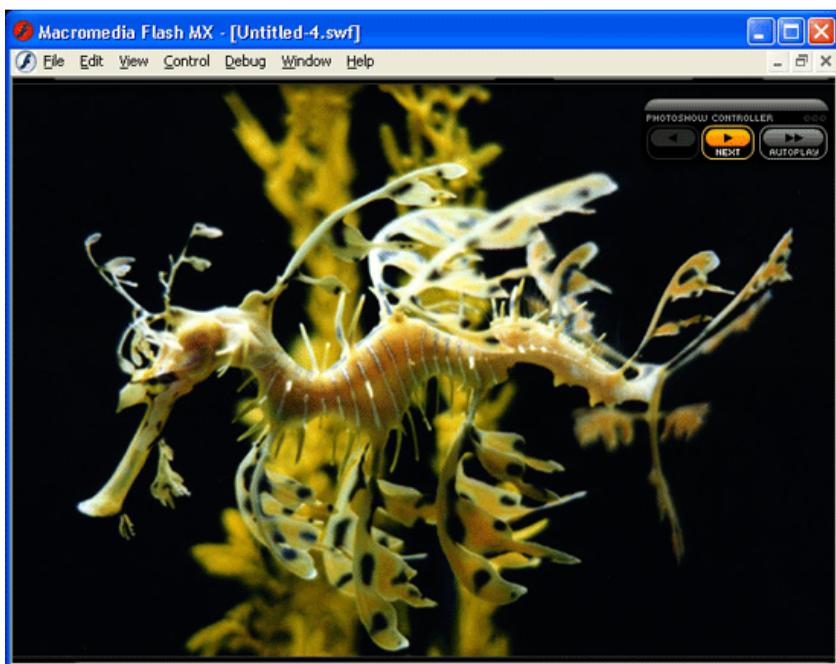


Рис. 100. Пример применения кнопок в интерактивном фильме

Простейшими командами управления фильмом являются запуск и прерывание его воспроизведения. Эти команды реализуются с помощью predetermined

действий *Play* и *Stop*. Процедуру назначения этих действий двум кнопкам рассмотрим на примере фильма, в котором шарик прокатывается столу, падает и разбивается. В качестве кнопок используем экземпляры символов из общей библиотеки *Buttons*.

Итак, для включения в фильм интерактивных кнопок необходимо выполнить следующие действия:

1. Добавьте в фильм новый слой, который будет использоваться в качестве дополнительного фона; именно в фоновый слой следует добавить кнопки, чтобы они присутствовали на сцене на всем протяжении фильма.

2. Щелкните мышью в ячейке первого ключевого кадра в «кнопочном» слое.

3. Перетащите поочередно из окна библиотеки на стол две кнопки, которые вы решили использовать в фильме; в результате окно Flash будет выглядеть примерно так, как показано на рис. 101.

4. Выберите первую из кнопок, с помощью которой фильм должен приостанавливаться.

5. С помощью инспектора свойств кнопки откройте панель *Actions* и щелкните в списке *Actions Toolbox* на значке раздела *Actions*.

6. Откройте подраздел *Movie Control*, отыщите в нем действие *Stop* и дважды щелкните на нем мышью; в результате в окне сценария появится текст сценария, содержащий обработчик *onfrelease*); он обеспечивает инициализацию действия *stop()* при наступлении события *release* (то есть при щелчке на кнопке); пустые скобки после имени процедуры означают, что данная процедура используется без параметров; соответствующее сообщение выводится и в окне параметров, как показано на рис. 102.

7. Не закрывая панель *Actions*, выберите на столе вторую кнопку и назначьте ей действие *play*, повторив описанную выше процедуру.

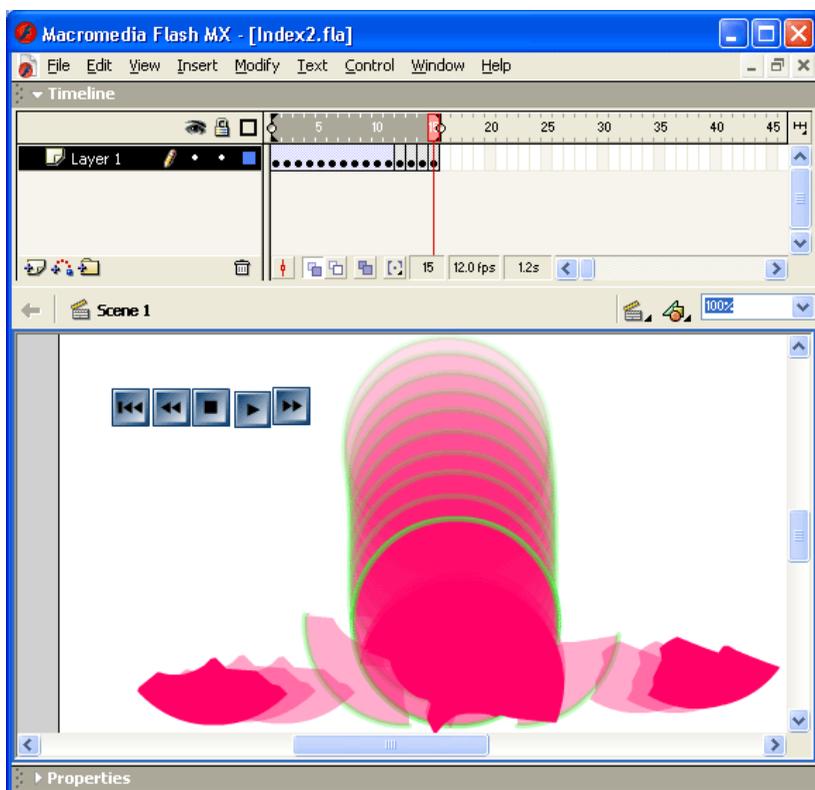


Рис. 101. Окно фильма после добавления кнопок

В отличие от обычной анимации, протестировать многие сценарии в режиме редактирования невозможно. Чтобы проверить, как работают созданные кнопки, следует выбрать в меню *Control* команду *Test Movie* или *Test Scene*. В результате фильм будет Экспортирован в формат SWF и затем воспроизведен с помощью Flash-плеера. Чтобы вернуться в режим редактирования, достаточно просто закрыть окно плеера.

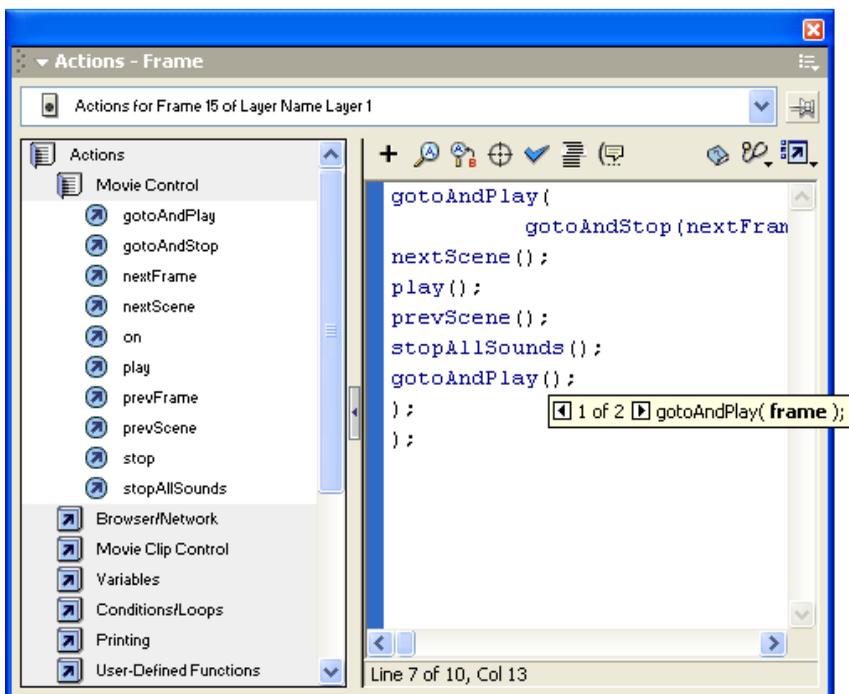


Рис. 102. Панель Actions после назначения кнопке действия stop

Компоненты интерфейса. В предыдущих версиях Flash использовалось понятие Smart Clip, то есть «сложный клип». Это специальный тип символов-клипов, на основе которых в Flash-фильмах создавались элементы пользовательского интерфейса (кнопки, флажки, переключатели и т. д.). Так вот «компоненты» (Components) — это «прямые наследники» Smart Clip.

В состав Flash MX включены семь типов компонентов:

- *CheckBox* — флажок;
- *ComboBox* — комбинированный список (то есть раскрывающийся список с возможностью редактирования пунктов);

- *ListBox* - список;
- *PushButton* - кнопка;
- *RadioButton* — переключатель;
- *ScrollBar* — полоса прокрутки;
- *ScrollPane* - окно.

С точки зрения программирования, на ActionScript компоненты представляют собой как бы подкласс объектов *MovieClip*, для которого предусмотрены специальные параметры и некоторые дополнительные методы. Параметры компонентов могут устанавливаться как во время разработки фильма, так и во время его воспроизведения, с помощью сценария. Методы, применимые к компонентам, собраны в специальном разделе списка *Action Toolbox*, который называется *Flash UI Components* (рис. 103).

Наиболее распространенный вариант применения компонентов — это создание на их основе аналога формы, помещаемой на Web-странице. Вместе с тем, на основе компонентов вы можете создать для своего Flash-фильма «традиционный» для Windows-приложений пользовательский интерфейс. В этом случае работа пользователя с фильмом будет мало чем отличаться от работы с обычным приложением.

При использовании компонентов в качестве элементов формы Flash обеспечивает:

- пересылку введенных в форму данных на сервер по заданному адресу;
- обработку введенных данных на клиентской машине средствами языка ActionScript.

Другими словами, возможности Flash по работе с формами ни в чем не уступают возможностям HTML. Более того, в отличие от HTML страниц, Flash-фильм может оставаться загруженным в браузере при обмене информацией с сервером, без необходимости перезагрузки.

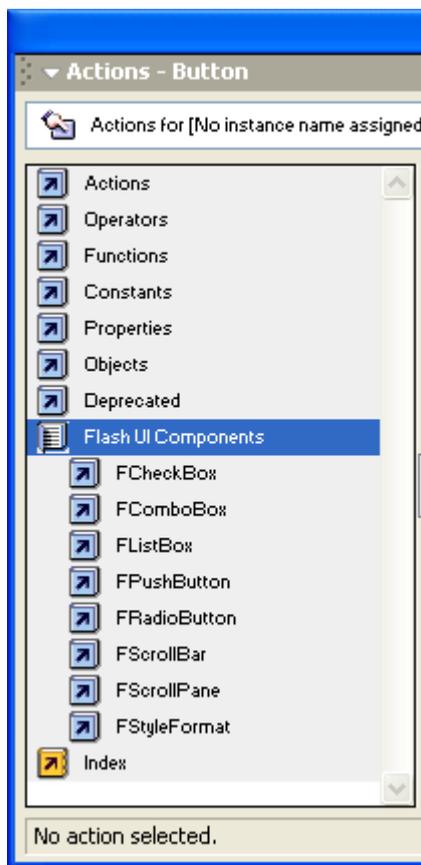


Рис. 103. Раздел Flash UI Components списка Action Toolbox

Сохранение информации в базе данных на сервере и ее выдача по запросу из фильма позволяет создавать динамически обновляемое и персонализированное содержание фильма. Например, на основе форм Flash-фильма могут быть созданы «доска объявлений», или тележка в электронном магазине, которая «помнит», какой товар пользуется спросом у данного пользователя. При этом

серверное приложение может быть написано на любом из применяемых для этого языков программирования (например, на Perl, ASP или PHP).

Особенность Flash-форм по сравнению с формами, созданными с помощью HTML, состоит в следующем.

В HTML все элементы формы объединяются явным образом в едином контейнере с помощью тэга `<FORM>`. Во Flash таким контейнером служит сам фильм. Это означает, что все интерактивные элементы интерфейса (кнопки, текстовые поля ввода, списки) воспринимаются Flash как элементы одной формы.

Благодаря этому для пересылки данных на сервер может быть создана единственная кнопка — аналог стандартной HTML-кнопки *Submit*. Этой кнопке следует назначить действие *getURL* с указанием используемого метода пересылки.

С другой стороны, если в фильме используется какой-либо один элемент интерфейса, он также интерпретируется Flash как законченная самостоятельная форма. В этом случае действие *getURL* следует назначить непосредственно данному элементу.

Вставку компонентов в фильм удобнее всего выполнять с помощью специальной панели, которая так и называется — *Components*. Ее формат приведен на рис. 104.

Для установки исходных параметров экземпляра компонента, включенного в фильм, целесообразно воспользоваться инспектором свойств. Его формат для компонентов несколько отличается от используемого для других элементов фильма. Основное отличие состоит в том, что панель инспектора свойств компонента имеет две вкладки (рис. 105):

- *Properties* (Свойства), с помощью элементов которой устанавливаются свойства компонента как экземпляра символа-клипа; эта вкладка одинакова для всех компонентов;

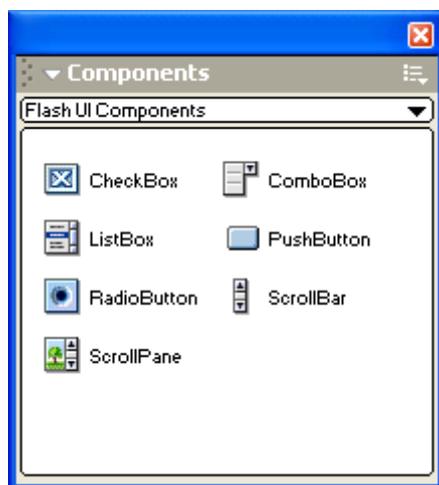


Рис.104. Формат панели Components



Рис. 105. Формат инспектора свойств компонента

- *Parameters* (Параметры), предназначенная для установки параметров компонента как элемента пользовательского интерфейса; для каждого типа компонента набор параметров различен.

Рассмотрим формирование простого элемента интерфейса – кнопки. В обычных Windows-приложениях кнопки (PushButton или просто Button) используются для инициализации тех или иных действий. Кнопки, входящие в форму, размещенную на HTML-странице, обычно служат для

инициализации пересылки введенных данных (кнопка *Submit*), либо для сброса введенных значений (кнопка *Reset*). Кнопки, реализованные в виде компонентов и помещаемые в Flash-фильм, могут применяться для инициализации любых действий. Фактически, они ничем не отличаются от символов-кнопок, рассмотренных выше, за исключением того, что для кнопки-компонента определены лишь два визуальных образа: для исходного состояния и когда она «нажата». Соответственно, кнопка-компонент не «реагирует» на перемещение указателя.

Для кнопки-компонента предусмотрен метод *setClickHandler*, который позволяет назначить кнопке функцию-обработчика единственного события — щелчка на кнопке (*Click*). Например, если в фильме имеется кнопка *button_1*, то в сценарии вы можете написать такую конструкцию: *button_1.setClickHandler(«onClick»)*; здесь *onClick* — это имя функции-обработчика, которая должна быть определена для той же временной диаграммы, к которой относится кнопка *button_1*.

- помощью инспектора свойств для кнопки могут быть заданы два параметра (рис. 106):
- *Label* — текстовая метка, отображаемая на кнопке в качестве ее имени;
- *Change Handler* — имя функции-обработчика (то есть значение параметра метода *setClickHandler*).

Назначение и редактирование процедур кнопки. Для задания более сложных операций, которые выполняются при взаимодействии пользователя с элементами интерфейса, используется специальный язык программирования — *ActionScript*.

Существует специальная панель редактора *Actions* для создания сценариев на *ActionScript*. Причем работать он может в двух режимах: обычном (то есть предназначенном для «обычных» пользователей) — *Normal*, и в экспертном

(для «продвинутых» пользователей) — *Expert*.

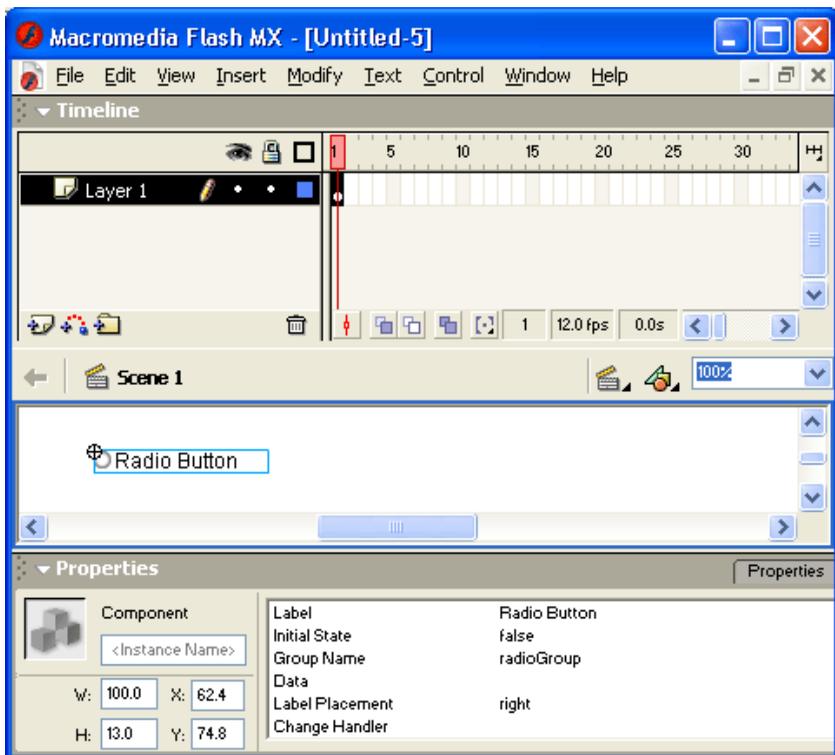


Рис. 106. Формат инспектора свойств кнопки

В первом случае пользователю предлагается как бы визуальный режим работы и доля ручного кодирования сведена к минимуму, либо вообще равна нулю. Во втором режиме пользователь получает большую свободу действий, но при этом лишается некоторых вспомогательных средств. В качестве предварительной иллюстрации к сказанному на рис. 107 показаны оба формата панели применительно к одному и тому же «мини-сценарию», состоящему из единственной инструкции.

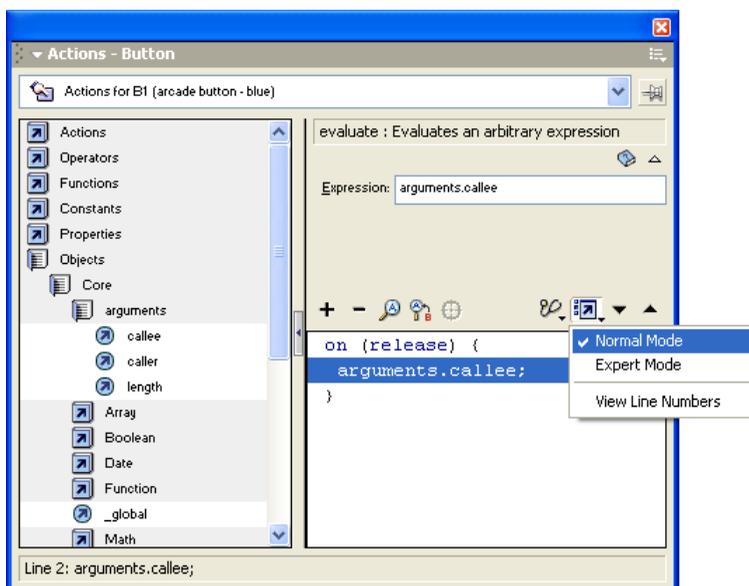
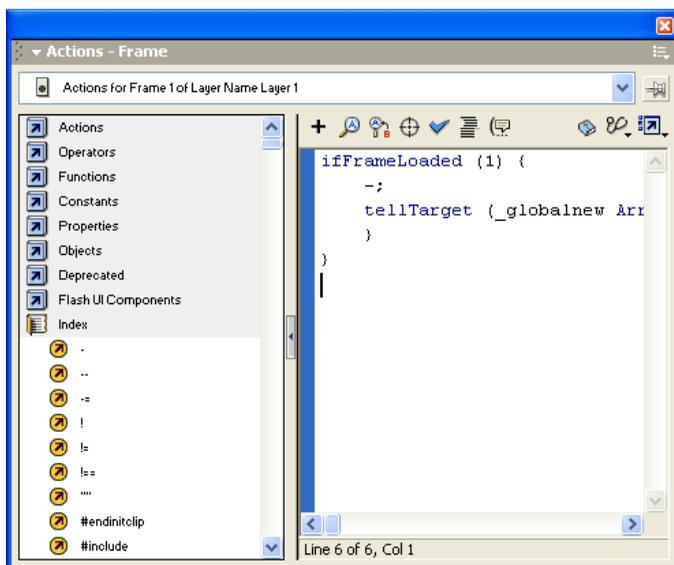


Рис. 107. Экспертный (вверху) и обычный (внизу) форматы панели Actions

Ниже особенности каждого из форматов будут рассмотрены подробно.

Существует несколько способов вызова на экран панели *Actions*. Однако наиболее надежный и корректный способ состоит в том, чтобы вызывать панель «от имени» того элемента фильма, для которого вы собираетесь создать сценарий.

Чтобы открыть панель *Actions* для создания сценария кадра, необходимо щелкнуть правой кнопкой мыши на временной диаграмме в ячейке соответствующего ключевого кадра и в контекстном меню выбрать команду *Actions*. После того, как панель появится на экране, убедитесь, что создаваемый сценарий будет связан именно с тем кадром, с которым требуется. Для этого достаточно взглянуть на заголовок панели и расположенный под ним список (рис. 108).

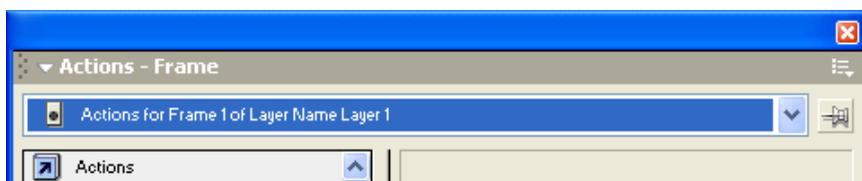


Рис. 108. Заголовок панели *Actions* при создании сценария для кадра

Чтобы открыть панель *Actions* для создания сценария, связанного с кнопкой или с клипом, необходимо:

1. Выбрать соответствующий объект на столе.
2. Открыть панель инспектора свойств.
3. Щелкнуть кнопку *Edit the action script* (Редактировать сценарий действий), расположенную возле правой границы панели инспектора свойств (рис. 109).

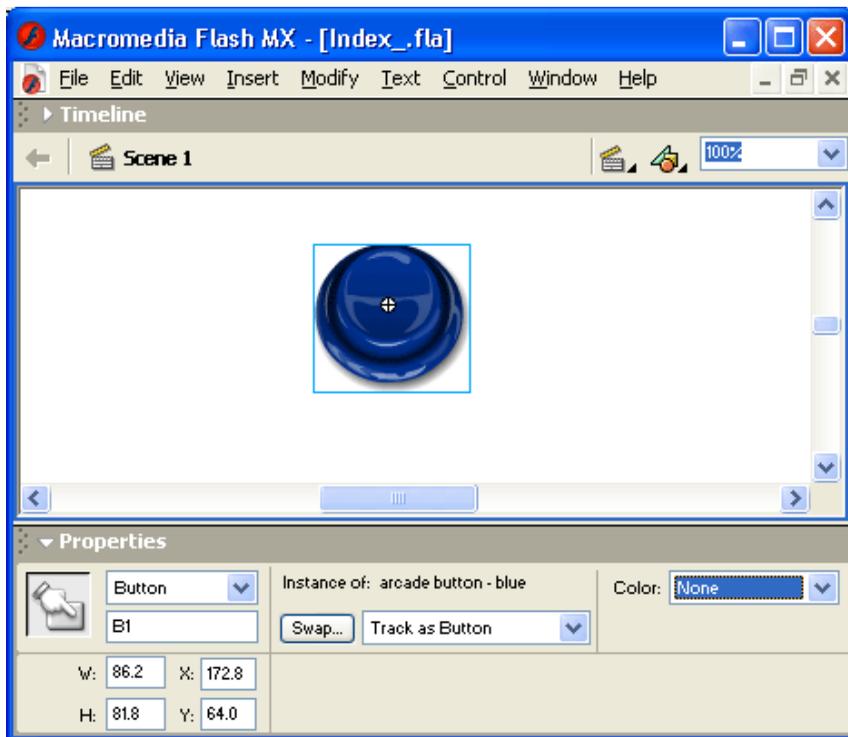


Рис. 109. Кнопка вызова панели Actions для кнопки или клипа

Эта маленькая кнопочка обладает интересной особенностью: если сценарий для данного объекта отсутствует, она белого цвета, после создания сценария она «синее».

При вызове панели *Actions* для кнопки или клипа в ее заголовке отображается соответствующая информация (рис. 110).

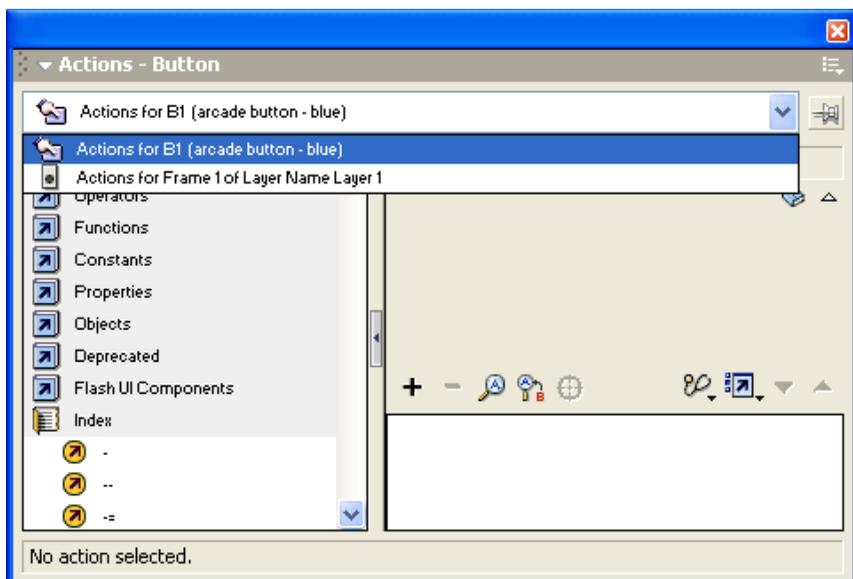


Рис. 110. Заголовок панели Actions при создании сценария для кнопки

Сценарий может быть сопоставлен только конкретному экземпляру кнопки или клипа, помещенному на стол, но не собственно символу, хранящемуся в библиотеке. Назначение сценария экземпляру кнопки или клипа не влияет на свойства других экземпляров.

После того, как панель будет открыта, вы можете выбрать режим работы.

Выбор режима выполняется с помощью соответствующих команд меню *View Options* (Настройка формата), которое открывается щелчком на одноименной кнопке панели (рис. 111).

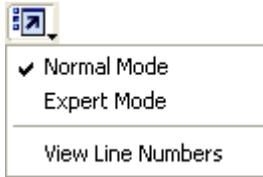


Рис. 111. Меню для выбора режима работы панели Actions

Независимо от выбранного режима, панель содержит следующие основные элементы интерфейса (рис. 112):

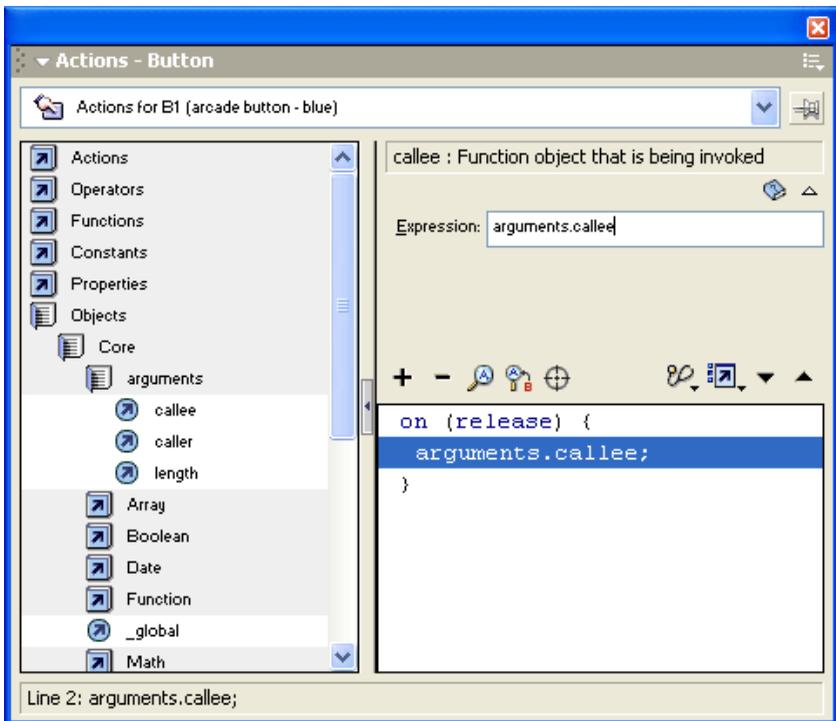


Рис. 112. Основные элементы интерфейса панели Actions

- раскрывающийся список сценариев фильма; он обеспечивает быстрый переход к другому сценарию без необходимости выбора на столе объекта, с которым связан этот сценарий;

- кнопка *Script Pin* (Прикрепить сценарий) позволяет зафиксировать текущий сценарий; речь идет о том, что содержимое панели зависит от выбранного на столе объекта фильма; после фиксации сценария вы можете выбрать на столе другой объект, и содержимое панели при этом не изменится;

- кнопка в виде книжки обеспечивает вызов встроенной справочной системы по языку ActionScript; необходимо отметить, что эта справка значительно подробнее, чем справка в формате HTML;

- панель *Action Toolbox* (Комплект инструментов для действий); каждый элемент, представленный в этом подокне, соответствует определенной конструкции языка ActionScript; двойной щелчок на значке элемента обеспечивает его вставку в текст сценария;

- окно сценария, в котором отображается текст сценария; при работе в режиме Expert Mode в этом окне можно работать как в обычном текстовом редакторе;

- панель инструментов окна сценария; размещенные на ней кнопки повышают удобство работы с текстом сценария; состав кнопок в некоторой степени зависит от режима работы; их назначение будет рассмотрено в двух следующих подразделах;

- строка состояния; она предназначена для вывода дополнительной информации относительно компонентов, включенных в сценарий; в частности, в строке состояния выводится краткое сообщение об ошибке.

- Список *Action Toolbox* содержит перечень следующих компонентов языка ActionScript, которые могут

быть использованы при создании сценария (рис. 113):



Рис. 113. Категории компонентов языка ActionScript

- *Actions* (Действия) - процедуры языка ActionScript, имеющие «универсальный» характер, то есть не зависящие от особенностей класса объектов (в отличие от методов объектов, принадлежащих конкретному классу); в эту же категорию входят традиционные для большинства языков программирования высокого уровня операторы логического управления (проверки логических условий, организации цикла и т. п.);

- *Operators* (Операторы) — операторы логических и арифметических операций (сложение, вычитание, сравнение и т. п.);

- *Functions* (Функции) — predetermined (библиотечные) функции (например, в их число входит функция проверки номера версии Flash-плеера);

- *Properties* (Свойства) - свойства объектов, не зависящие от класса (например, уровень прозрачности

изображения);

- *Objects* (Объекты) — предопределенные объекты, которые могут быть использованы в сценарии (например, существует объект для работы с датой и временем); в этот же раздел включаются пользовательские классы объектов;

- *Deprecated* (Осуждаемые) — конструкции, которые не рекомендуется использовать в сценариях (как устаревшие);

- *Flash UI Components* (Компоненты пользовательского интерфейса).

Кроме перечисленных разделов, в списке *Action Toolbox* имеется также раздел *Index*, который содержит алфавитный перечень всех компонентов языка. /26/

7. ОСНОВНЫЕ ПОНЯТИЯ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Мы приступаем к изучению реляционных баз данных (БД) и систем управления реляционными базами данных. Этот подход является наиболее распространенным в настоящее время, хотя наряду с общепризнанными достоинствами обладает и рядом недостатков. К числу достоинств реляционного подхода можно отнести:

- наличие небольшого набора абстракций, которые позволяют сравнительно просто моделировать большую часть распространенных предметных областей и допускают точные формальные определения, оставаясь интуитивно понятными;

- наличие простого и в то же время мощного математического аппарата, опирающегося главным образом на теорию множеств и математическую логику и обеспечивающего теоретический базис реляционного подхода к организации баз данных;

- возможность ненавигационного манипулирования данными без необходимости знания конкретной физической организации баз данных во внешней памяти.

Реляционные системы далеко не сразу получили широкое распространение. В то время, как основные теоретические результаты в этой области были получены еще в 70-х, и тогда же появились первые прототипы реляционных БД, долгое время считалось невозможным добиться эффективной реализации таких систем. Однако отмеченные выше преимущества и постепенное накопление методов и алгоритмов организации реляционных баз данных и управления ими привели к тому, что уже в середине 80-х годов реляционные системы практически вытеснили с мирового рынка ранние БД.

В настоящее время основным предметом критики реляционных БД является не их недостаточная эффективность, а присущая этим системам некоторая ограниченность (прямое следствие простоты) при использовании в так называемых нетрадиционных областях (наиболее распространенными примерами являются системы автоматизации проектирования), в которых требуются предельно сложные структуры данных. Еще одним часто отмечаемым недостатком реляционных баз данных является невозможность адекватного отражения семантики предметной области. Другими словами, возможности представления знаний о семантической специфике предметной области в реляционных системах очень ограничены. Современные исследования в области постреляционных систем главным образом посвящены именно устранению этих недостатков.

7.1 Общие понятия реляционного подхода к организации баз данных. Основные концепции и термины

Введем на сравнительно неформальном уровне основные понятия реляционных баз данных, а также определим существо реляционной модели данных. Основной

целью является демонстрация простоты и возможности интуитивной интерпретации этих понятий. В дальнейших лекциях будут приводиться более формальные определения, на которых основывается математическая теория реляционных баз данных.

Основными понятиями реляционных баз данных являются тип данных, домен, атрибут, кортеж, первичный ключ и отношение. Для начала покажем смысл этих понятий на примере отношения СОТРУДНИКИ, содержащего информацию о сотрудниках некоторой организации (рис. 114):

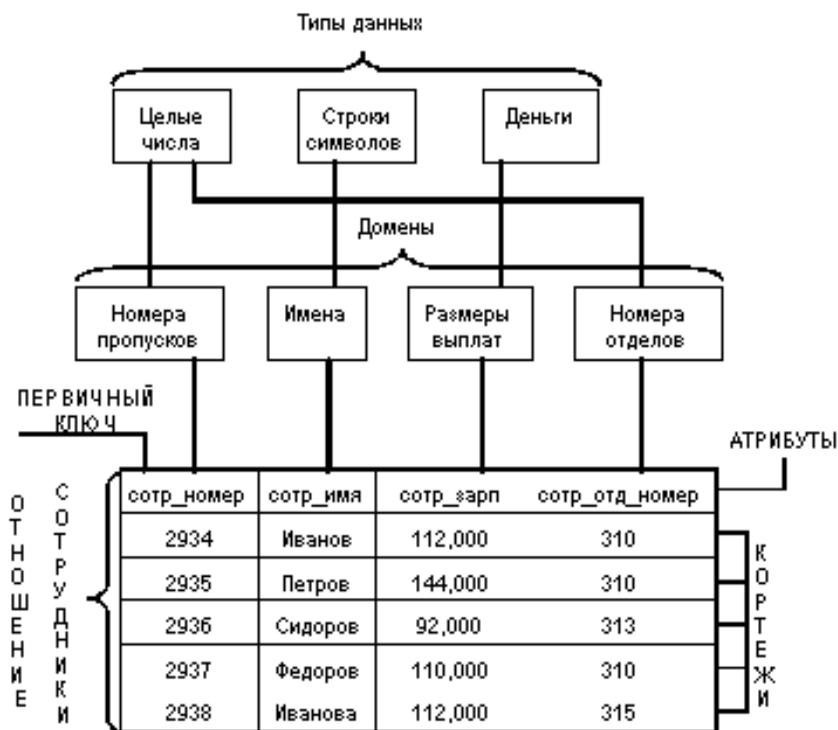


Рис. 114. Схема основных понятий реляционной базы данных.

Тип данных Понятие тип данных в реляционной модели данных полностью адекватно понятию типа данных в языках программирования. Обычно в современных реляционных БД допускается хранение символьных, числовых данных, битовых строк, специализированных числовых данных (таких как "деньги"), а также специальных "темпоральных" данных (дата, время, временной интервал). Достаточно активно развивается подход к расширению возможностей реляционных систем абстрактными типами данных (соответствующими возможностями обладают, например, системы семейства Ingres/Postgres). В нашем примере мы имеем дело с данными трех типов: строки символов, целые числа и "деньги".

Домен Понятие домена более специфично для баз данных, хотя и имеет некоторые аналогии с подтипами в некоторых языках программирования. В самом общем виде домен определяется заданием некоторого базового типа данных, к которому относятся элементы домена, и произвольного логического выражения, применяемого к элементу типа данных. Если вычисление этого логического выражения дает результат "истина", то элемент данных является элементом домена.

Наиболее правильной интуитивной трактовкой понятия домена является понимание домена как допустимого потенциального множества значений данного типа. Например, домен "Имена" в нашем примере определен на базовом типе строк символов, но в число его значений могут входить только те строки, которые могут изображать имя (в частности, такие строки не могут начинаться с мягкого знака).

Следует отметить также семантическую нагрузку понятия домена: данные считаются сравнимыми только в том случае, когда они относятся к одному домену. В нашем

примере значения доменов "Номера пропусков" и "Номера групп" относятся к типу целых чисел, но не являются сравнимыми. Заметим, что в большинстве реляционных БД понятие домена не используется, хотя в Oracle V.7 оно уже поддерживается.

Схема отношения, схема базы данных Схема отношения - это именованное множество пар {имя атрибута, имя домена (или типа, если понятие домена не поддерживается)}. Степень или "арность" схемы отношения - мощность этого множества. Степень отношения СОТРУДНИКИ равна четырем, то есть оно является 4-арным. Если все атрибуты одного отношения определены на разных доменах, осмысленно использовать для именованния атрибутов имена соответствующих доменов (не забывая, конечно, о том, что это является всего лишь удобным способом именованния и не устраняет различия между понятиями домена и атрибута).

Схема БД (в структурном смысле) - это набор именованных схем отношений.

Кортеж, отношение Кортеж, соответствующий данной схеме отношения, - это множество пар {имя атрибута, значение}, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения. "Значение" является допустимым значением домена данного атрибута (или типа данных, если понятие домена не поддерживается). Тем самым, степень или "арность" кортежа, т.е. число элементов в нем, совпадает с "арностью" соответствующей схемы отношения. Попросту говоря, кортеж - это набор именованных значений заданного типа.

Отношение - это множество кортежей, соответствующих одной схеме отношения. Иногда, чтобы не путаться, говорят "отношение-схема" и "отношение-экземпляр", иногда схему отношения называют заголовком отношения, а отношение как набор кортежей - телом

отношения. На самом деле, понятие схемы отношения ближе всего к понятию структурного типа данных в языках программирования. Было бы вполне логично разрешать отдельно определять схему отношения, а затем одно или несколько отношений с данной схемой.

Однако в реляционных базах данных это не принято. Имя схемы отношения в таких базах данных всегда совпадает с именем соответствующего отношения-экземпляра. В классических реляционных базах данных после определения схемы базы данных изменяются только отношения-экземпляры. В них могут появляться новые и удаляться или модифицироваться существующие кортежи. Однако во многих реализациях допускается и изменение схемы базы данных: определение новых и изменение существующих схем отношения. Это принято называть эволюцией схемы базы данных.

Обычным житейским представлением отношения является таблица, заголовком которой является схема отношения, а строками - кортежи отношения-экземпляра; в этом случае имена атрибутов именуют столбцы этой таблицы. Поэтому иногда говорят "столбец таблицы", имея в виду "атрибут отношения". Когда мы перейдем к рассмотрению практических вопросов организации реляционных баз данных и средств управления, мы будем использовать эту житейскую терминологию. Этой терминологии придерживаются в большинстве коммерческих реляционных БД.

Реляционная база данных - это набор отношений, имена которых совпадают с именами схем отношений в схеме БД.

Как видно, основные структурные понятия реляционной модели данных (если не считать понятия домена) имеют очень простую интуитивную интерпретацию, хотя в теории реляционных БД все они определяются абсолютно формально и точно.

7.2 Фундаментальные свойства отношений

Остановимся теперь на некоторых важных свойствах отношений, которые следуют из приведенных ранее определений:

Отсутствие кортежей-дубликатов То свойство, что отношения не содержат кортежей-дубликатов, следует из определения отношения как множества кортежей. В классической теории множеств по определению каждое множество состоит из различных элементов.

Из этого свойства вытекает наличие у каждого отношения так называемого *первичного ключа* - набора атрибутов, значения которых однозначно определяют кортеж отношения. Для каждого отношения по крайней мере полный набор его атрибутов обладает этим свойством. Однако при формальном определении первичного ключа требуется обеспечение его "минимальности", т.е. в набор атрибутов первичного ключа не должны входить такие атрибуты, которые можно отбросить без ущерба для основного свойства - однозначно определять кортеж. Понятие первичного ключа является исключительно важным в связи с понятием целостности баз данных.

Отсутствие упорядоченности кортежей Свойство отсутствия упорядоченности кортежей отношения также является следствием определения отношения-экземпляра как множества кортежей. Отсутствие требования к поддержанию порядка на множестве кортежей отношения дает дополнительную гибкость БД при хранении баз данных во внешней памяти и при выполнении запросов к базе данных. Это не противоречит тому, что при формулировании запроса к БД, например, на языке SQL можно потребовать сортировки результирующей таблицы в соответствии со значениями некоторых столбцов. Такой результат, вообще говоря, не отношение, а некоторый упорядоченный список кортежей.

Отсутствие упорядоченности атрибутов Атрибуты отношений не упорядочены, поскольку по определению схема отношения есть множество пар {имя атрибута, имя домена}. Для ссылки на значение атрибута в кортеже отношения всегда используется имя атрибута. Это свойство теоретически позволяет, например, модифицировать схемы существующих отношений не только путем добавления новых атрибутов, но и путем удаления существующих атрибутов. Однако в большинстве существующих систем такая возможность не допускается, и хотя упорядоченность набора атрибутов отношения явно не требуется, часто в качестве неявного порядка атрибутов используется их порядок в линейной форме определения схемы отношения.

Атомарность значений атрибутов Значения всех атрибутов являются атомарными. Это следует из определения домена как потенциального множества значений простого типа данных, т.е. среди значений домена не могут содержаться множества значений (отношения). Принято говорить, что в реляционных базах данных допускаются только нормализованные отношения или отношения, представленные в первой нормальной форме. Потенциальным примером ненормализованного отношения является следующее (рис. 115):

НОМЕР_ОТДЕЛА	ОТДЕЛ		
	СОТР_НОМЕР	СОТР_ИМЯ	СОТР_ЗАРП
310	2934	Иванов	112,000
	2935	Петров	112,500
313	2937	Федоров	110,000
315	2938	Иванова	112,000

Рис. 115. Пример ненормализованного отношения.

Можно сказать, что здесь мы имеем бинарное отношение, значениями атрибута ОТДЕЛЫ которого являются отношения. Заметим, что исходное отношение СОТРУДНИКИ является нормализованным вариантом отношения ОТДЕЛЫ:

СОТР_НОМЕР	СОТР_ИМЯ	СОТР_ЗАРП	СОТР_ОТД_НОМЕР
2934	Иванов	112,000	310
2935	Петров	144,000	310
2936	Сидоров	92,000	313
2937	Федоров	110,000	310
2938	Иванова	112,000	315

Нормализованные отношения составляют основу классического реляционного подхода к организации баз данных. Они обладают некоторыми ограничениями (не любую информацию удобно представлять в виде плоских таблиц), но существенно упрощают манипулирование данными. Рассмотрим, например, два идентичных оператора занесения кортежа:

«Зачислить сотрудника Кузнецова (пропуск номер 3000, зарплата 115,000) в отдел номер 320» и

«Зачислить сотрудника Кузнецова (пропуск номер 3000, зарплата 115,000) в отдел номер 310».

Если информация о сотрудниках представлена в виде отношения СОТРУДНИКИ, оба оператора будут выполняться одинаково (вставить кортеж в отношение СОТРУДНИКИ). Если же работать с ненормализованным отношением ОТДЕЛЫ, то первый оператор выразится в занесение кортежа, а второй - в добавление информации о Кузнецове в множественное значение атрибута ОТДЕЛ кортежа с

первичным ключом 310.

7.3 Реляционная модель данных

Когда в предыдущих разделах мы говорили об основных понятиях реляционных баз данных, мы не опирались на какую-либо конкретную реализацию. Эти рассуждения в равной степени относились к любой системе, при построении которой использовался реляционный подход.

Другими словами, мы использовали понятия так называемой реляционной модели данных. Модель данных описывает некоторый набор родовых понятий и признаков, которыми должны обладать все конкретные БД и управляемые ими базы данных, если они основываются на этой модели. Наличие модели данных позволяет сравнивать конкретные реализации, используя один общий язык.

Хотя понятие модели данных является общим, и можно говорить о иерархической, сетевой, некоторой семантической и т.д. моделях данных, нужно отметить, что это понятие было введено в обиход применительно к реляционным системам и наиболее эффективно используется именно в этом контексте. Попытки прямолинейного применения аналогичных моделей к дореляционным организациям показывают, что реляционная модель слишком "велика" для них, а для постреляционных организаций она оказывается "мала".

Общая характеристика Наиболее распространенная трактовка реляционной модели данных, по-видимому, принадлежит Дейту, который воспроизводит ее (с различными уточнениями) практически во всех своих книгах. Согласно Дейту реляционная модель состоит из трех частей, описывающих разные аспекты реляционного подхода: структурной части, манипуляционной части и целостной части.

В структурной части модели фиксируется, что

единственной структурой данных, используемой в реляционных БД, является нормализованное n -арное отношение. По сути дела, в предыдущих двух разделах этой лекции мы рассматривали именно понятия и свойства структурной составляющей реляционной модели.

В манипуляционной части модели утверждаются два фундаментальных механизма манипулирования реляционными БД - реляционная алгебра и реляционное исчисление. Первый механизм базируется в основном на классической теории множеств (с некоторыми уточнениями), а второй - на классическом логическом аппарате исчисления предикатов первого порядка. Мы рассмотрим эти механизмы более подробно на следующей лекции, а пока лишь заметим, что основной функцией манипуляционной части реляционной модели является обеспечение меры реляционности любого конкретного языка реляционных БД: язык называется реляционным, если он обладает не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционное исчисление.

Целостность сущности и ссылок Наконец, в целостной части реляционной модели данных фиксируются два базовых требования целостности, которые должны поддерживаться в любой реляционной БД. Первое требование называется *требованием целостности сущностей*. Объекту или сущности реального мира в реляционных БД соответствуют кортежи отношений. Конкретно требование состоит в том, что любой кортеж любого отношения отличим от любого другого кортежа этого отношения, т.е. другими словами, любое отношение должно обладать первичным ключом. Как мы видели в предыдущем разделе, это требование автоматически удовлетворяется, если в системе не нарушаются базовые свойства отношений.

Второе требование называется *требованием целостности по ссылкам* и является несколько более

сложным. Очевидно, что при соблюдении нормализованности отношений сложные сущности реального мира представляются в реляционной БД в виде нескольких кортежей нескольких отношений. Например, представим, что нам требуется представить в реляционной базе данных сущность ОТДЕЛ с атрибутами ОТД_НОМЕР (номер отдела), ОТД_КОЛ (количество сотрудников) и ОТД_СОТР (набор сотрудников отдела). Для каждого сотрудника нужно хранить СОТР_НОМЕР (номер сотрудника), СОТР_ИМЯ (имя сотрудника) и СОТР_ЗАРП (заработная плата сотрудника). Как мы вскоре увидим, при правильном проектировании соответствующей БД в ней появятся два отношения: ОТДЕЛЫ (ОТД_НОМЕР, ОТД_КОЛ) (первичный ключ - ОТД_НОМЕР) и СОТРУДНИКИ (СОТР_НОМЕР, СОТР_ИМЯ, СОТР_ЗАРП, СОТР_ОТД_НОМ) (первичный ключ - СОТР_НОМЕР).

Как видно, атрибут СОТР_ОТД_НОМ появляется в отношении СОТРУДНИКИ не потому, что номер отдела является собственным свойством сотрудника, а лишь для того, чтобы иметь возможность восстановить при необходимости полную сущность ОТДЕЛ. Значение атрибута СОТР_ОТД_НОМ в любом кортеже отношения СОТРУДНИКИ должно соответствовать значению атрибута ОТД_НОМ в некотором кортеже отношения ОТДЕЛЫ. Атрибут такого рода называется внешним ключом, поскольку его значения однозначно характеризуют сущности, представленные кортежами некоторого другого отношения (т.е. задают значения их первичного ключа). Говорят, что отношение, в котором определен внешний ключ, ссылается на соответствующее отношение, в котором такой же атрибут является первичным ключом.

Требование целостности по ссылкам, или требование внешнего ключа состоит в том, что для каждого значения внешнего ключа, появляющегося в ссылающемся отношении,

в отношении, на которое ведет ссылка, должен найтись кортеж с таким же значением первичного ключа, либо значение внешнего ключа должно быть неопределенным (т.е. ни на что не указывать). Для нашего примера это означает, что если для сотрудника указан номер отдела, то этот отдел должен существовать.

Ограничения целостности сущности и по ссылкам должны поддерживаться БД. Для соблюдения целостности сущности достаточно гарантировать отсутствие в любом отношении кортежей с одним и тем же значением первичного ключа. С целостностью по ссылкам дела обстоят несколько более сложно.

Понятно, что при обновлении ссылающегося отношения (вставке новых кортежей или модификации значения внешнего ключа в существующих кортежах) достаточно следить за тем, чтобы не появлялись некорректные значения внешнего ключа. Но как быть при удалении кортежа из отношения, на которое ведет ссылка?

Здесь существуют три подхода, каждый из которых поддерживает целостность по ссылкам. Первый подход заключается в том, что запрещается производить удаление кортежа, на который существуют ссылки (т.е. сначала нужно либо удалить ссылающиеся кортежи, либо соответствующим образом изменить значения их внешнего ключа). При втором подходе при удалении кортежа, на который имеются ссылки, во всех ссылающихся кортежах значение внешнего ключа автоматически становится неопределенным. Наконец, третий подход (каскадное удаление) состоит в том, что при удалении кортежа из отношения, на которое ведет ссылка, из ссылающегося отношения автоматически удаляются все ссылающиеся кортежи.

В развитых реляционных БД обычно можно выбрать способ поддержания целостности по ссылкам для каждой отдельной ситуации определения внешнего ключа. Конечно,

для принятия такого решения необходимо анализировать требования конкретной прикладной области.

7.4 Базисные средства манипулирования реляционными данными

В предыдущей лекции мы говорили про три составляющих реляционной модели данных. Две из них - структурную и целостную составляющие - мы рассмотрели более или менее подробно, а манипуляционной части реляционной модели данных посвящается эта лекция.

Как мы отмечали в предыдущей лекции, в манипуляционной составляющей определяются два базовых механизма манипулирования реляционными данными - основанная на теории множеств реляционная алгебра и базирующееся на математической логике (точнее, на исчислении предикатов первого порядка) реляционное исчисление. В свою очередь, обычно рассматриваются два вида реляционного исчисления - исчисление доменов и исчисление предикатов.

Все эти механизмы обладают одним важным свойством: они замкнуты относительно понятия отношения. Это означает, что выражения реляционной алгебры и формулы реляционного исчисления определяются над отношениями реляционных БД и результатом вычисления также являются отношения. В результате любое выражение или формула могут интерпретироваться как отношения, что позволяет использовать их в других выражениях или формулах.

Как мы увидим, алгебра и исчисление обладают большой выразительной мощностью: очень сложные запросы к базе данных могут быть выражены с помощью одного выражения реляционной алгебры или одной формулы реляционного исчисления. Именно по этой причине именно эти механизмы включены в реляционную модель данных.

Конкретный язык манипулирования реляционными БД называется реляционно полным, если любой запрос, выражаемый с помощью одного выражения реляционной алгебры или одной формулы реляционного исчисления, может быть выражен с помощью одного оператора этого языка.

Известно (и мы не будем это доказывать), что механизмы реляционной алгебры и реляционного исчисления эквивалентны, т.е. для любого допустимого выражения реляционной алгебры можно построить эквивалентную (т.е. производящую такой же результат) формулу реляционного исчисления и наоборот. Почему же в реляционной модели данных присутствуют оба эти механизма?

Дело в том, что они различаются уровнем *процедурности*. Выражения реляционной алгебры строятся на основе алгебраических операций (высокого уровня), и подобно тому, как интерпретируются арифметические и логические выражения, выражение реляционной алгебры также имеет процедурную интерпретацию. Другими словами, запрос, представленный на языке реляционной алгебры, может быть вычислен на основе вычисления элементарных алгебраических операций с учетом их старшинства и возможного наличия скобок. Для формулы реляционного исчисления однозначная интерпретация, вообще говоря, отсутствует. Формула только устанавливает условия, которым должны удовлетворять кортежи результирующего отношения. Поэтому языки реляционного исчисления являются более непроцедурными или декларативными.

Поскольку механизмы реляционной алгебры и реляционного исчисления эквивалентны, то в конкретной ситуации для проверки степени реляционности некоторого языка БД можно пользоваться любым из этих механизмов.

Заметим, что крайне редко алгебра или исчисление принимаются в качестве полной основы какого-либо языка

БД. Обычно (как, например, в случае языка SQL) язык основывается на некоторой смеси алгебраических и логических конструкций. Тем не менее, знание алгебраических и логических основ языков баз данных часто бывает полезно на практике.

В нашем изложении мы в основном следуем подходу Дейта, примененному (хотя и не изобретенному) им в последнем издании книги "Введение в системы баз данных". Для экономии времени и места мы не будем вводить каких-либо строгих синтаксических конструкций, а в основном ограничимся рассмотрением материала на содержательном уровне.

Реляционная алгебра Основная идея реляционной алгебры состоит в том, что коль скоро отношения являются множествами, то средства манипулирования отношениями могут базироваться на традиционных теоретико-множественных операциях, дополненных некоторыми специальными операциями, специфичными для баз данных.

Существует много подходов к определению реляционной алгебры, которые различаются набором операций и способами их интерпретации, но в принципе, более или менее равносильны. Мы опишем немного расширенный начальный вариант алгебры, который был предложен Коддом. В этом варианте набор основных алгебраических операций состоит из восьми операций, которые делятся на два класса - теоретико-множественные операции и специальные реляционные операции. В состав теоретико-множественных операций входят операции:

- объединения отношений;
- пересечения отношений;
- взятия разности отношений;
- прямого произведения отношений.

Специальные реляционные операции включают:

- ограничение отношения;

- проекцию отношения;
- соединение отношений;
- деление отношений.

Кроме того, в состав алгебры включается операция присваивания, позволяющая сохранить в базе данных результаты вычисления алгебраических выражений, и операция переименования атрибутов, дающая возможность корректно сформировать заголовок (схему) результирующего отношения.

7.5 Проектирование реляционных БД

При проектировании базы данных решаются две основных проблемы:

- Каким образом отобразить объекты предметной области в абстрактные объекты модели данных, чтобы это отображение не противоречило семантике предметной области и было по возможности лучшим (эффективным, удобным и т.д.)? Часто эту проблему называют проблемой логического проектирования баз данных.

- Как обеспечить эффективность выполнения запросов к базе данных, т.е. каким образом, имея в виду особенности конкретной БД, расположить данные во внешней памяти, создание каких дополнительных структур (например, индексов) потребовать и т.д.? Эту проблему называют проблемой физического проектирования баз данных.

В случае реляционных баз данных трудно представить какие-либо общие рецепты по части физического проектирования. Здесь слишком много зависит от используемой БД. Например, при работе с БД Ingres можно выбирать один из предлагаемых способов физической организации отношений, при работе с System R следовало бы прежде всего подумать о кластеризации отношений и требуемом наборе индексов и т.д. Поэтому мы ограничимся

вопросами логического проектирования реляционных баз данных, которые существенны при использовании любой реляционной БД.

Более того, мы не будем касаться очень важного аспекта проектирования - определения ограничений целостности (за исключением ограничения первичного ключа). Дело в том, что при использовании БД с развитыми механизмами ограничений целостности (например, SQL-ориентированных систем) трудно предложить какой-либо общий подход к определению ограничений целостности. Эти ограничения могут иметь очень общий вид, и их формулировка пока относится скорее к области искусства, чем инженерного мастерства. Самое большее, что предлагается по этому поводу в литературе, это автоматическая проверка непротиворечивости набора ограничений целостности.

Так что будем считать, что проблема проектирования реляционной базы данных состоит в обоснованном принятии решений о том,

- из каких отношений должна состоять БД и
- какие атрибуты должны быть у этих отношений.

7.6 Проектирование реляционных баз данных с использованием нормализации

Сначала будет рассмотрен классический подход, при котором весь процесс проектирования производится в терминах реляционной модели данных методом последовательных приближений к удовлетворительному набору схем отношений. Исходной точкой является представление предметной области в виде одного или нескольких отношений, и на каждом шаге проектирования производится некоторый набор схем отношений, обладающих лучшими свойствами. Процесс проектирования представляет собой процесс нормализации схем отношений, причем

каждая следующая нормальная форма обладает свойствами лучшими, чем предыдущая.

Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений. Примером набора ограничений является ограничение первой нормальной формы - значения всех атрибутов отношения атомарны. Поскольку требование первой нормальной формы является базовым требованием классической реляционной модели данных, мы будем считать, что исходный набор отношений уже соответствует этому требованию.

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- первая нормальная форма (1NF);
- вторая нормальная форма (2NF);
- третья нормальная форма (3NF);
- нормальная форма Бойса-Кодда (BCNF);
- четвертая нормальная форма (4NF);
- пятая нормальная форма, или нормальная форма проекции-соединения (5NF или PJ/NF).

Основные свойства нормальных форм:

- каждая следующая нормальная форма в некотором смысле лучше предыдущей;
- при переходе к следующей нормальной форме свойства предыдущих нормальных свойств сохраняются.

В основе процесса проектирования лежит метод нормализации, декомпозиция отношения, находящегося в предыдущей нормальной форме, в два или более отношения, удовлетворяющих требованиям следующей нормальной формы /27/

7.7 Нормализация базы данных

Нормализация таблиц базы данных - первый шаг на пути проектирования структуры реляционной базы данных. Строго говоря, конечно, не самый первый - сначала надо решить, что же мы вообще будем хранить в базе, то есть определиться со структурой полей, их типами и размерностью, смыслом хранимой в них информации. Но это, как говорится, подразумевается по умолчанию.

Теория нормализации реляционных баз данных была разработана в конце 70-х годов 20 века. Согласно ей, выделяются шесть нормальных форм, пять из которых так и называются: первая, вторая, третья, четвертая, пятая нормальная форма, а также нормальная форма Бойса-Кодда, лежащая между третьей и четвертой.

База данных считается нормализованной, если ее таблицы (по крайней мере, большинство таблиц) представлены как минимум в третьей нормальной форме. Часто многие таблицы нормализуются до четвертой нормальной формы, иногда, наоборот, производится денормализация. Использование таблиц в пятой нормальной форме (вернее сказать, сознательного приведения их к пятой нормальной форме) в реальных базах данных встречается крайне редко.

Главная цель нормализации базы данных - устранение избыточности и дублирования информации. В идеале при нормализации надо добиться, чтобы любое значение хранилось в базе в одном экземпляре, причем значение это не должно быть получено расчетным путем из других данных, хранящихся в базе.

Наверно, нет смысла подробно рассматривать примеры нормализации таблиц. Такой информации и в Интернете, и в книгах более чем достаточно. Напомню только, каким основным требованиям должна удовлетворять каждая из

нормальных форм.

Первая нормальная форма. Первая нормальная форма:

- запрещает повторяющиеся столбцы (содержащие одинаковую по смыслу информацию)
- запрещает множественные столбцы (содержащие значения типа списка и т.п.)
- требует определить первичный ключ для таблицы, то есть тот столбец или комбинацию столбцов, которые однозначно определяют каждую строку

Вторая нормальная форма Вторая нормальная форма требует, чтобы неключевые столбцы таблиц зависели от первичного ключа в целом, но не от его части. Маленькая ремарочка: если таблица находится в первой нормальной форме и первичный ключ у нее состоит из одного столбца, то она автоматически находится и во второй нормальной форме.

Третья нормальная форма Чтобы таблица находилась в третьей нормальной форме, необходимо, чтобы неключевые столбцы в ней не зависели от других неключевых столбцов, а зависели только от первичного ключа. Самая распространенная ситуация в данном контексте - это расчетные столбцы, значения которых можно получить путем каких-либо манипуляций с другими столбцами таблицы. Для приведения таблицы в третью нормальную форму такие столбцы из таблиц надо удалить.

Нормальная форма Бойса-Кодда Нормальная форма Бойса-Кодда требует, чтобы в таблице был только один потенциальный первичный ключ. Чаще всего у таблиц, находящихся в третьей нормальной форме, так и бывает, но не всегда. Если обнаружился второй столбец (комбинация столбцов), позволяющий однозначно идентифицировать строку, то для приведения к нормальной форме Бойса-Кодда такие данные надо вынести в отдельную таблицу.

Четвертая нормальная форма Для приведения таблицы, находящейся в нормальной форме Бойса-Кодда, к четвертой

нормальной форме необходимо устранить имеющиеся в ней многозначные зависимости. То есть обеспечить, чтобы вставка / удаление любой строки таблицы не требовала бы вставки / удаления / модификации других строк этой же таблицы.

Пятая нормальная форма Таблицу, находящуюся в четвертой нормальной форме и, казалось бы, уже нормализованную до предела, в некоторых случаях еще можно бывает разбить на три или более (но не на две!) таблиц, соединив которые, мы получим исходную таблицу. Получившиеся в результате такой, как правило, весьма искусственной, декомпозиции таблицы и называют находящимися в пятой нормальной форме. Формальное определение пятой нормальной формы таково: это форма, в которой устранены зависимости соединения. В большинстве случаев практической пользы от нормализации таблиц до пятой нормальной формы не наблюдается.

Разработаны специальные формальные математические методы нормализации таблиц реляционных баз данных. На практике же толковый проектировщик баз данных, детально познакомившись с предметной областью, как правило, достаточно быстро набросает структуру, в которой большинство таблиц находятся в четвертой нормальной форме.

Зачем нужна нормализация? Главное, чего мы добьемся, проведя нормализацию базы данных - это устранение (или, по крайней мере, серьезное сокращение) избыточности, дублирования данных. Как следствие, значительно сокращается вероятность появления противоречивых данных, облегчается администрирование базы и обновление информации в ней, сокращается объем дискового пространства.

Но не все так просто и хорошо. Зачастую, чтобы извлечь информацию из нормализованной базы данных,

приходится конструировать очень сложные запросы, которые к тому же, бывает, работают довольно медленно - из-за, главным образом, большого количества соединений таблиц. Поэтому, чтобы увеличить скорость выборки данных и упростить программирование запросов, нередко приходится идти на выборочную денормализацию базы. /28/

8. ДИСТАНЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

В последние десятилетия появилась новая проблема развития образования в России. Знания стареют каждые 3-5 лет, а технологические знания – каждые 2-3 года. Пройдет еще немного времени и это будет 1,5-2 года. Объем знаний выпускников ВУЗов удваивается каждые 3-4 года. Если не менять образовательных технологий, то качество подготовки специалистов будет объективно отставать от требуемого на рынке труда. Усвоение знаний студентами с помощью информационных и коммуникационных технологий по самым нижним оценкам на 40-60% быстрее, или больше, в единицу времени, чем с обычными технологиями (за один и тот же период дается больше знаний, или модно сокращать сроки обучения в ВУЗах).

Учебно-методическая база, образовательно-информационные технологии любого ВУЗа таковы, что они вообще не зависят от того, дневная ли это форма обучения, или заочная, или удаленная. Если знания, вест учебный материал, вся его дидактическая составляющая оформлены и находятся в формализованном виде, в компьютерах, то все равно, в принципе, куда подать эти знания: то ли в аудиторию (соответственно оснащенную), то ли находящемуся за пределами учебного заведения пользователю.

Одним из видов инноваций в организации профессионального образования является введение

дистанционного обучения, которое мне и хотелось бы проанализировать в своей работе.

В последние десятилетия дистанционные образовательные технологии в России получили интенсивное развитие. Министерством образования РФ разработано специальное направление, научно-методическая программа, выделены средства на развитие и становление дистанционного образования.

О термине "дистанционное обучение" Термин "дистанционное обучение" (distance education) еще до конца не устоялся как в русскоязычной, так и в англоязычной педагогической литературе. Встречаются такие варианты как "дистантное образование" (distant education), "дистантное обучение" (distant learning). Некоторые зарубежные исследователи, отводя особую роль телекоммуникациям в организации дистанционного обучения, определяют его как телеобучение (teletraining). Но все же наиболее часто употребляется термин "дистанционное обучение".

8.1 Технологические основы дистанционного обучения

Дистанционное обучение в виде заочного обучения зародилось еще в начале XX века. Сегодня заочно можно получить не только высшее образование, но и изучить иностранный язык, подготовиться к поступлению в ВУЗ и т.д. Однако в связи с плохо налаженным взаимодействием между преподавателями и студентами и отсутствием контроля над учебной деятельностью студентов-заочников в периоды между экзаменационными сессиями качество подобного обучения оказывается хуже того, что можно получить при очном обучении.

Современные компьютерные телекоммуникации способны обеспечить передачу знаний и доступ к

разнообразной учебной информации наравне, а иногда и гораздо эффективнее, чем традиционные средства обучения. Эксперименты подтвердили, что качество и структура учебных курсов, равно как и качество преподавания при дистанционном обучении, зачастую намного лучше, чем при традиционных формах обучения. Новые электронные технологии, такие как интерактивные диски CD-ROM, электронные доски объявлений, мультимедийный гипертекст, доступные через глобальную сеть Интернет с помощью интерфейсов Mosaic и WWW могут не только обеспечить активное вовлечение учащихся в учебный процесс, но и позволяют управлять этим процессом в отличие от большинства традиционных учебных сред. Интеграция звука, движения, образа и текста создает новую необыкновенно богатую по своим возможностям учебную среду, с развитием которой увеличится и степень вовлечения учащихся в процесс обучения. Интерактивные возможности, используемых в системе дистанционного обучения (СДО) программ и систем доставки информации, позволяют наладить и даже стимулировать обратную связь, обеспечить диалог и постоянную поддержку, которые невозможны в большинстве традиционных систем обучения.

8.2 Дистанционное обучение в мире

По данным зарубежных экспертов к 2010 году минимальным уровнем образования, необходимым для выживания человечества, стало высшее образование. Обучение такой массы студентов по очной (дневной) форме вряд ли выдержат бюджеты даже самых благополучных стран. Поэтому не случайно за последние десятилетия численность обучающихся по нетрадиционным технологиям растет быстрее числа студентов дневных отделений. Мировая тенденция перехода к нетрадиционным формам образования

прослеживается и в росте числа ВУЗов, ведущих подготовку по этим технологиям. За период 1900-1960 гг. их было создано 79, за 1960-1970 гг. - 70, а только за 1970-1980 гг. - 87.

Долговременная цель развития СДО в мире – дать возможность каждому обучающемуся, живущему в любом месте, пройти курс обучения любого колледжа или университета. Это предполагает переход от концепции физического перемещения студентов из страны в страну к концепции мобильных идей, знаний и обучения с целью распределения знаний посредством обмена образовательными ресурсами /29/.

Интерактивное взаимодействие учителя и учащихся
Термин "интерактивное взаимодействие" широко используется как в отечественной, так и в зарубежной педагогической литературе. В узком смысле слова (применительно к работе пользователя с программным обеспечением вообще) интерактивное взаимодействие - это диалог пользователя с программой, т.е. обмен текстовыми командами (запросами) и ответами (приглашениями). При более развитых средствах ведения диалога (например, при наличии возможности задавать вопросы в произвольной форме, с использованием "ключевого" слова, в форме с ограниченным набором символов) обеспечивается возможность выбора вариантов содержания учебного материала и режима работы. Чем больше существует возможностей управлять программой, чем активнее пользователь участвует в диалоге, тем выше интерактивность. В широком смысле интерактивное взаимодействие предполагает диалог любых субъектов друг с другом с использованием доступных им средств и методов. При этом предполагается активное участие в диалоге обеих сторон - обмен вопросами и ответами, управление ходом диалога, контроль за выполнением принятых решений и т.п.

Телекоммуникационная среда, предназначенная для общения миллионов людей друг с другом, является априори интерактивной средой. При дистанционном обучении субъектами в интерактивном взаимодействии будут выступать преподаватели и студенты, а средствами осуществления подобного взаимодействия – электронная почта, телеконференции, диалоги в режиме реального времени и т.д.

8.3 Организационно-методические модели дистанционного обучения (ДО)

1. Обучение по типу экстерната. Обучение, ориентированное на ВУЗовские (экзаменационные) требования, предназначалось для студентов, которые по каким-то причинам не могли посещать стационарные учебные заведения. Так, в 1836 году был организован Лондонский университет, основной задачей которого в те годы была помощь и проведение экзаменов на получение тех или иных аттестатов, степеней и пр. для студентов, не посещавших обычные учебные заведения. Эта задача сохранилась и поныне наряду со стационарным обучением студентов.

2. Обучение на базе одного университета. Это уже целая система обучения для студентов, которые обучаются не стационарно (on-campus), а на расстоянии, заочно или дистанционно, т.е. на основе новых информационных технологий, включая компьютерные телекоммуникации (off-campus). Такие программы для получения разнообразных аттестатов образования разработаны во многих ведущих университетах мира. Так, Новый университет Южного Уэльса в Австралии проводит заочное и дистанционное обучение для 5000 студентов, тогда, как стационарно в нем обучается только 3000 студентов.

3. Сотрудничество нескольких учебных заведений.

Такое сотрудничество в подготовке программ заочного дистанционного обучения позволяет сделать их более профессионально качественными и менее дорогостоящими. Подобная практика реализована, например, в межуниверситетской телеобразовательной программе Кеприкон, в разработке которой приняли участие университеты Аргентины, Боливии, Бразилии, Чили и Парагвая. Другим примером подобного сотрудничества может служить программа "Содружество в образовании". Главы Британских стран содружества встретились в 1987 году с тем, чтобы договориться об организации сети дистанционного обучения для всех стран содружества. Перспективная цель программы - дать возможность любому гражданину стран содружества, не покидая своей страны и своего дома, получить любое образование на базе функционирующих в странах содружества колледжей и университетов.

4. Автономные образовательные учреждения, специально созданные для целей ДО. Самым крупным подобным учреждением является Открытый университет (The Open University) в Лондоне, на базе которого в последние годы проходят обучение дистанционно большое число студентов не только из Великобритании, но из многих стран Содружества. В США примером такого университета могут служить Национальный технологический университет (штат Колорадо), который готовит студентов по различным инженерным специальностям совместно с 40 инженерными колледжами. В 1991 году университет объединил эти 40 колледжей сетью ДО при теснейшем сотрудничестве с правительством штата и сферой бизнеса.

5. Автономные обучающие системы. Обучение в рамках подобных систем ведется целиком посредством ТВ или радиопрограмм, а также дополнительных печатных пособий. Примерами такого подхода к обучению на

расстоянии могут служить американо-самоанский телевизионный проект.

6. Неформальное, интегрированное дистанционное обучение на основе мультимедийных программ. Такие программы ориентированы на обучение взрослой аудитории, тех людей, которые по каким-то причинам не смогли закончить школьное образование. Такие проекты могут быть частью официальной образовательной программы, интегрированными в эту программу (примеры таких программ существуют в Колумбии), или специально ориентированные на определенную образовательную цель (например, Британская программа грамотности), или специально нацеленные на профилактические программы здоровья, как, например, программы для развивающихся стран.

8.4 Организационно-технологические модели ДО

Едиичная медиа - использование какого-либо одного средства обучения и канала передачи информации. Например, обучение через переписку, учебные радио- или телепередачи. В этой модели доминирующим средством обучения является, как правило, печатный материал. Практически отсутствует двусторонняя коммуникация, что приближает эту модель дистанционного обучения к традиционному заочному обучению.

Мультимедиа - использование различных средств обучения: учебные пособия на печатной основе, компьютерные программы учебного назначения на различных носителях, аудио- и видеозаписи и т.п. Однако, доминирует при этом передача информации в "одну сторону". При необходимости используются элементы очного обучения - личные встречи обучающихся и преподавателей, проведение итоговых учебных семинаров или консультаций, очный

прием экзаменов и т.п. Эту технологическую модель мы рассмотрим более подробно ниже. За главный объект мы возьмем электронный учебник (ЭУ).

Гипермедиа - модель дистанционного обучения третьего поколения, которая предусматривает использование новых информационных технологий при доминирующей роли компьютерных телекоммуникаций. Простейшей формой при этом является использование электронной почты и телеконференций, а также аудиообучение (сочетание телефона и телефакса). При дальнейшем развитии эта модель дистанционного обучения включает использование комплекса таких средств как видео, телефакс и телефон (для проведения видеоконференций) и аудиографику при одновременном широком использовании видеодисков, различных гиперсредств, систем знаний и искусственного интеллекта.

8.5 Виртуальные университеты

Созданные университетами учебные сервера - это, в некотором роде, расширение стен самого университета. В его виртуальных аудиториях так же, как и в основных, можно будет со временем и лекцию послушать, и лабораторную на виртуальном стенде выполнить, и найти средства для проектирования, выполнения расчетов, моделирования спроектированного устройства и т. д. Но возможно, что все вышеперечисленное станет прерогативой специализированных виртуальных университетов - электронных открытых университетов без стен. Тем более что ВУЗам, подключаемым к Internet на средства из фонда Сороса, не разрешено коммерческое использование доступа к Сети, поэтому средства для оплаты онлайн-услуг (и не только на подписку на энциклопедии) придется изыскивать дополнительно. Есть и много других проблем, препятствующих созданию виртуальных университетов в

традиционных университетах со стенами. Сведения о проектах и первых попытках создания виртуальных университетов можно найти в сети Internet.

Дешевле ли обучать по Internet? Эксперты считают, что телекоммуникационное интерактивное преподавание обходится на 20-25% дешевле традиционного. Microsoft считает, что стоимость сетевого обучения может снизиться как минимум вдвое против традиционного, поскольку преподаватель в состоянии давать уроки, находясь в любой точке земного шара; да и особого компьютерного оборудования при этом не требуется. Экономия может быть достигнута и за счет других факторов. Взяв на вооружение СДО, учебный отдел может быть уверен, что все учащиеся пользуются одними и теми же и, кроме того, самыми свежими учебно-методическими материалами. Ведь обновлять учебные пособия с помощью Internet гораздо легче. Интересное наблюдение, но оказалось, что при обучении в СДО проще производить отсев малоспособных учеников. Пассивно вести себя на обычных семинарах - легко, а на электронных - невозможно. В результате на СДО-курсах отсеивается большее число учеников, чем на традиционных. Разберем более подробно.

8.6 Дистанционное обучение в ВУЗе: модели и технологии

Термин "дистанционное обучение" означает такую организацию учебного процесса, при которой преподаватель разрабатывает учебную программу, главным образом базирующуюся на самостоятельном обучении студента. Такая среда обучения характеризуется тем, что учащийся в основном, а зачастую и совсем отделен от преподавателя в пространстве или во времени, в то же время, студенты и преподаватели имеют возможность осуществлять диалог

между собой с помощью средств телекоммуникации. Дистанционное обучение позволяет учиться жителям регионов, где нет иных возможностей для профессиональной подготовки или получения качественного высшего образования, нет университета нужного профиля или преподавателей требуемого уровня квалификации.

С середины 70-х годов во многих странах стали появляться учебные заведения нового типа, называемые "открытый", "дистанционный" университет; "электронный", "виртуальный" колледж. Они имеют оригинальную организационную структуру, используют своеобразный набор педагогических приемов, экономических механизмов функционирования.

Цель этой работы - проанализировать возникающий на наших глазах спектр новых учебных заведений и выделить их базовые модели. Важно не просто рассмотреть существующие модели образовательных учреждений, функционирующих на основе использования дистанционных методов обучения и информационных технологий, но и понять, какие новые формы могут появиться в будущем, что определяет их разнообразие.

На данном этапе исследования представляется целесообразным рассмотреть три интегрированных фактора:

- технологический;
- педагогический;
- организационный.

Характер первого из них определяется информационными технологиями, используемыми для разработки, доставки, поддержки учебных курсов и учебного процесса в целом.

Значение второго фактора определяется набором методов и приемов, применяемых в ходе учебного процесса.

Третий фактор, организационный, характеризует специфику организационной структуры образовательного

учреждения дистанционного обучения.

"Идеальная модель" дистанционного обучения включает в себя интегрированную учебную среду, с вариантным определением роли различных компонент - технологических, педагогических, организационно-методических /30, 31/.

8.7 Основные типы технологий, применяемых в учебных заведениях нового типа

В качестве первого фактора (интегрированной характеристики) университета дистанционного обучения рассматривается тип используемых в учебном процессе информационных технологий. При этом необходимо подчеркнуть два важных аспекта.

Во-первых, такая очередность рассмотрения факторов вовсе не означает присвоение наивысшего приоритета технологии в организации учебного процесса. Как бы мощны и совершенны ни были технологические применения, они должны служить образовательным (педагогическим) целям, а не наоборот. Но с другой стороны, нельзя и недооценивать роль новых информационных технологий, которые зачастую предлагают качественно новые возможности реализации образовательного процесса.

Во-вторых, приведенный ниже перечень основных технологий, применяемых в университетах дистанционного обучения, конечно же не означает, что какая-то конкретная модель должна характеризоваться применением лишь одной из них. Мультимедиа-подход, основанный на использовании нескольких взаимодополняющих информационных технологий, представляется наиболее эффективным в области образования.

Используемые сегодня технологии дистанционного образования можно разделить на три большие категории:

- неинтерактивные (печатные материалы, аудио-, видео-носители);
- средства компьютерного обучения (электронные учебники, компьютерное тестирование и контроль знаний, новейшие средства мультимедиа);
- видеоконференции - развитые средства телекоммуникации по аудиоканалам, видеоканалам и компьютерным сетям.

Средства оперативного доступа к информации по компьютерным сетям придали качественно новые возможности дистанционному обучению. В российских ВУЗах они активно развиваются в виде применения электронных учебников и технологии обмена текстовой информацией с помощью асинхронной электронной почты.

Развитые средства телекоммуникации, использование спутниковых каналов связи, передача упакованного видеоизображения по компьютерным сетям только совсем недавно стали применяться в практике дистанционного образования. Это связано с отсутствием развитой инфраструктуры связи, высокой стоимостью каналов связи и используемого оборудования.

Видеокассеты - это уникальное средство для дистанционного обучения практически по любой дисциплине. Не требуя больших расходов на тиражирование учебных видеоматериалов, видеомагнитофон получил широкое распространение во всех странах. Видеокассеты используются обычно как компоненты наборов учебных материалов, частично заменяя традиционные лекции.

Электронная почта экономически и технологически является наиболее эффективной технологией, которая может быть использована в процессе обучения для доставки содержательной части учебных курсов и обеспечения обратной связи студента с преподавателем. В то же самое время она имеет ограниченный педагогический эффект из-за

невозможности реализации "диалога" между преподавателем и студентами, принятого в традиционной форме обучения. Однако, если студенты имеют постоянный доступ к персональному компьютеру с модемом и телефонному каналу, электронная почта позволяет реализовать гибкий и интенсивный процесс консультаций.

Оперативный доступ к разделяемым информационным ресурсам позволяет получить интерактивный доступ к удаленным базам данных, информационно-справочным системам, библиотекам при изучении конкретной дисциплины. Данный *режим доступа ON-LINE* позволяет в течение секунд осуществить передачу необходимого учебного материала, компьютерных программ при помощи таких компьютерных систем как GOPHER, WWW, VERONICA из крупных научно-педагогических центров, и из локальных узлов сети Internet, общее количество которых в мире превышает 1.25 миллиона.

Видеоконференции с использованием компьютерных сетей предоставляют возможность организации самой дешевой среднего качества видеосвязи. Данный тип видеоконференций может быть использован для проведения семинаров в небольших (5-10 человек) группах, индивидуальных консультаций, обсуждения отдельных сложных вопросов изучаемого курса. Помимо передачи звука и видеоизображения компьютерные видеоконференции обеспечивают возможность совместного управления экраном компьютера: создание чертежей и рисунков на расстоянии, передачу фотографического и рукописного материала.

Видеоконференции по цифровому спутниковому каналу с использованием видеокомпрессии совмещают высокое качество передаваемого видеоизображения и низкую стоимость проведения видеоконференции (более чем на два порядка меньше, чем при использовании обычного аналогового телевизионного сигнала). Эта технология может

оказаться эффективными при относительно небольшом объеме лекций (100-300 часов в год) и большом числе обучаемых (1000-5000 студентов) для проведения обзорных лекций, коллективных обсуждений итогов курсов и образовательных программ.

В конце 80-х гг. в России технологии необходимые для дистанционного обучения были либо не развиты, либо отсутствовали вообще. Преподавание в ВУЗах основывалось на печатных изданиях, для развития тематики использовались заранее записанные программы, режиссура, телевизионные передачи в прямом эфире. Сегодня некоторые из новых информационных технологий становятся доступны для налаживания дистанционного обучения. Практически все они основаны на цифровых методах обработки информации и охватывают спектр от программ гипермедиа, которые позволяют студенту самому управлять используемой информацией, до разнообразных систем баз данных доступных через глобальные компьютерные сети.

Основным фактором при выборе информационных технологий как средств обучения должен быть их образовательный потенциал. Однако проведенные исследования показывают, что это не так даже в наиболее технологически развитых странах (США, Канада, Великобритания, Германия и Япония). В России экономическая и технологическая ситуация такова, что выбор средств зависит не от их педагогического потенциала и даже не от их стоимости, а от их распространенности.

Телекоммуникации добавляют новую размерность к дистанционному обучению и очень быстро развиваются в России в виде применения асинхронной электронной почты. Проведенный анализ позволил выделить базовые параметры, которые являются существенными при выборе информационных технологий для применения в программах дистанционного обучения в российских ВУЗах (Таблица 2).

Таблица 2. Сравнительные характеристики информационных технологий

Технология	Характеристики
Аудио-визуальные носители (печатные материалы, аудио-, видеокассеты).	<ul style="list-style-type: none"> - Низкая коммуникационная интерактивность; - Стоимость производства линейно зависит от числа обучаемых; - Хорошо известны методики разработки учебных материалов; - Высокая долговечность.
Компьютерное обучение, асинхронная электронная почта.	<ul style="list-style-type: none"> - Средняя степень интерактивности; - Наиболее развитая инфраструктура в России; - Низкая стоимость.
Видеоконференции по компьютерной сети Internet в режиме реального времени.	<ul style="list-style-type: none"> - Высокая степень интерактивности; - Наиболее развитая в мире инфраструктура сети; - Использование широко распространенных платформ компьютеров; - Низкая стоимость.
Видеоконференции по цифровому выделенному спутниковому каналу с использованием видеокompрессии	<ul style="list-style-type: none"> - Высокая степень интерактивности - Хорошее качество передачи изображения; - Снижение более чем на два порядка, требований к пропускной способности канала по сравнению с аналоговым телевизионным сигналом; - Высокая стоимость.
Видеоконференции по аналоговому спутниковому каналу.	<ul style="list-style-type: none"> - Высокая степень интерактивности; - Максимально возможное качество передачи изображения с минимальной технологической задержкой передачи изображения и звука; - Высокая стоимость.

8.8 Методы дистанционного университетского образования

Важным интегрированным фактором типологии дистанционных университетов является совокупность используемых в учебном процессе педагогических методов и приемов. Выбрав в качестве критерия способ коммуникации преподавателей и обучаемых, эти методы (приемы) можно классифицировать следующим образом:

1) Методы обучения посредством взаимодействия обучаемого с образовательными ресурсами при минимальном участии преподавателя и других обучаемых (самообучение). Для развития этих методов характерен мультимедиа подход, когда при помощи разнообразных средств создаются образовательные ресурсы: печатные, аудио-, видео-материалы, и что особенно важно для электронных университетов - учебные материалы, доставляемые по компьютерным сетям. Это, прежде всего:

- интерактивные базы данных;
- электронные журналы;
- компьютерные обучающие программы (электронные учебники).

В интерактивных базах данных систематизируются массивы данных, которые могут быть доступны посредством телекоммуникаций. Используя эти ресурсы, разработчики курсов, например, могут поддерживать локальные базы данных как для студентов, так и для преподавателей. Другим решением является предоставление доступа к внешним базам данных. Число баз данных, доступных через компьютерные сети быстро растет.

Так, в Murdoch University каталог библиотечных услуг, которые доступны через Internet, насчитывает более 70 страниц. Студенты и преподаватели Государственного университета Огайо имеют доступ по крайней мере к 9

главным библиотекам и целому ряду баз данных через Internet. Пользователи CompuServe имеют доступ к таким базам данных, как Academic American Encyclopedia, Dissertation Abstracts, ERIC, Magazine Database Plus, Peterson's College Database.

Электронные журналы представляют собой периодические издания, которые распространяются среди подписчиков через компьютерные сети. Они становятся все более важным источником получения информации и обучения. Как утверждалось в U.S. News & World Report (1994), более 2700 газет в 1994 году предпринимали ту или иную попытку издания электронных версий, в то время как в 1989 году таких газет было лишь 42. Strangelove составил в 1992 году справочник, который включал 35 электронных журналов и 90 информационных бюллетеней, доступных через Internet. Студенты подписываются на такие журналы с целью использования их как неотъемлемой части курса или как дополнения к работе.

Компьютерные обучающие программы представляют собой программное обеспечение, которое может использоваться на удаленном компьютере через компьютерную сеть. Сеанс связи с удаленным компьютером может осуществляться при помощи, например, модемной связи или Telnet услуг в Internet.

2) Методы индивидуализированного преподавания и обучения, для которых характерны взаимоотношения одного студента с одним преподавателем или одного студента с другим студентом (обучение "один к одному"). Эти методы реализуются в дистанционном образовании в основном посредством таких технологий, как телефон, голосовая почта, электронная почта. Развитие теленаставничества (система "тьюторов"), опосредованного компьютерными сетями, является важным компонентом учебного процесса в электронных университетах.

3) Методы, в основе которых лежит представление студентам учебного материала преподавателем или экспертом, при котором обучающиеся не играют активную роль в коммуникации (обучение "один ко многим").

Эти методы, свойственные традиционной образовательной системе, получают новое развитие на базе современных информационных технологий. Так, лекции, записанные на аудио- или видеокассеты, читаемые по радио или телевидению, дополняются в современном дистанционном образовательном процессе так называемыми "э-лекциями" (электронными лекциями), т.е. лекционным материалом, распространяемым по компьютерным сетям с помощью систем досок объявлений (BBS). Э-лекция может представлять собой подборку статей или выдержек из них, а также учебных материалов, которые готовят обучающихся к будущим дискуссиям. На базе технологии электронной доски объявлений развивается также метод проведения учебных электронных симпозиумов, представляющих собой серию выступлений нескольких авторитетов ("первых спикеров").

4) Методы, для которых характерно активное взаимодействие между всеми участниками учебного процесса (обучение "многие ко многим"). Значение этих методов и интенсивность их использования существенно возрастает с развитием обучающих телекоммуникационных технологий. Иными словами, интерактивные взаимодействия между самими обучающимися, а не только между преподавателем и обучающимися, становятся важным источником получения знаний. Развитие этих методов связано с проведением учебных коллективных дискуссий и конференций. Технологии аудио-, аудиографических и видео-конференций позволяют активно развивать такие методы в дистанционном образовании. Особую роль в учебном процессе дистанционных университетов играют компьютерные конференции, которые позволяют всем

участникам дискуссии обмениваться письменными сообщениями как в синхронном, так и в асинхронном режиме, что имеет большую дидактическую ценность.

Компьютерно-опосредованные коммуникации позволяют активнее использовать такие методы обучения, как дебаты, моделирование, ролевые игры, дискуссионные группы, мозговые атаки, методы Дельфи, методы номинальной группы, форумы, проектные группы.

Так, метод "*мозговой атаки*" представляет собой стратегию взаимодействия, позволяющую группам студентов эффективно генерировать идеи. Этот метод поощряет членов группы мыслить творчески и развивать идеи других членов группы. Основной целью метода мозговой атаки является создать фонд идей по определенной теме. При мозговой атаке исключается критицизм, поощряются свободные ассоциативные суждения.

Процедура *Дельфи* представляет собой метод для выработки надежного консенсуса номинальной группы студентов посредством серии анкетных опросов. Термин номинальная группа происходит от того, что студенты только номинально представляют собой группу на первоначальной стадии генерации идей. Первоначально каждого участника такой группы просят сформулировать и проранжировать идеи. Затем составляется общий список идей обычно путем выявления идей, которые получили самый высокий приоритет у отдельных участников, затем вторые по значимости и т.д. до тех пор, пока список у каждого участника не будет исчерпан. После этого все приглашаются к обсуждению идей. После дискуссии проводится голосование, в ходе которого членов группы просят проранжировать идеи, которые были генерированы в ходе дискуссии. В University of Auckland была разработана программная система для поддержки синхронных групповых занятий (groupware system), которая применялась в курсе по

менеджменту.

С целью классификации дистанционных университетов по педагогическим принципам, лежащим в основе их учебной практики, целесообразно выделить следующие принципы телематических систем образования:

- интерактивность учебного процесса;
- обучение как диалог;
- адаптивность обучения;
- гибкость учебного материала;
- «передаваемость» материала в дистанционном образовании;
- активность обучаемого.

Дистанционные образовательные учреждения обычно основываются не на каком-то одном из этих принципов, а на их совокупности. Тем не менее, обычно выделяются доминирующие.

8.9 Основные типы организационных структур дистанционного образования

Основные типы организационных структур университетского дистанционного образования включают в себя:

- Подразделения заочного (дистанционного) образования в традиционных университетах;
- Консорциум университетов;
- Открытые университеты;
- Виртуальные университеты.

Характерной тенденцией дистанционного образования является объединение организационных структур университетов. Так, в последние годы стал развиваться новый тип организационной структуры дистанционного университетского образования - *консорциум университетов*. Дистанционные образовательные услуги оказывает

специальная организация, объединяющая и координирующая деятельность нескольких университетов. Консорциум университетов предлагает набор курсов, разработанных в различных университетах - от курсов для абитуриентов до курсов на получение ученых степеней. В 70-х и 80-х годах во многих странах были учреждены *национальные открытые университеты*. Они использовали многие организационные принципы заочного обучения. Но в целом открытое образование привнесло много нового в образовательную систему. Принцип открытости образования означает свободу зачисления в число обучаемых и составления индивидуального учебного плана, а также свободу места, времени и темпов обучения. В основе открытого образования - богатая и детально разработанная образовательная среда, в которой обучаемый ориентируется вполне самостоятельно, стремясь к достижению стоящих перед ним образовательных целей.

В основе новой системы образования лежит принцип *открытости*, который применительно к высшему образованию означает:

- *открытое поступление* в высшее учебное заведение, т.е. отказ от любых условий и требований для зачисления, кроме достижения необходимого возраста (18 лет);
- *открытое планирование* обучения, т.е. свобода составления индивидуальной программы обучения путем выбора из системы курсов;
- *свобода в выборе времени и темпов обучения*, т.е. прием студентов в ВУЗ в течение всего года и отсутствие фиксированных сроков обучения;
- *свобода в выборе места обучения*: студенты физически отсутствуют в учебных аудиториях основную часть учебного времени и могут самостоятельно выбирать, где обучаться.

Проведение принципа открытости привело к

значительным организационным новшествами, которые стали практически осуществимы именно благодаря внедрению новых технологий хранения, переработки и передачи информации. Так, например, в 90-х годах появилась новая модель дистанционного образования на базе технологий проведения телеконференций. Эта модель называется *телеобучением* или *телеобразованием*. В этом случае проведение телеконференций, которые могут быть и в реальном времени, является главной формой взаимодействия между учителем и обучающимся, расширяя это взаимодействие, ранее осуществляемое, главным образом, по почте. При этом телеконференции могут проводиться как между преподавателем и учениками, так и между самими обучающимися. Это могут быть аудио-, аудиографические, видео- и компьютерные телеконференции.

Модель телеобразования появилась недавно, но она ведет к радикальным изменениям в организации современного образования. Это ярко проявляется в том, что на базе этой модели стала развиваться новая организационная форма современного образования - *виртуальные университеты*. Эта форма обучения рассматривается нами как новая, только что наметившаяся модель образования. В этой модели полностью реализуются те потенциальные возможности перестройки системы образования, которые имеют технологии телеконференций, используемые в учебных целях. Эти технологии позволяют группам учащихся и отдельным обучаемым встречаться с преподавателями и между собой, находясь на любом расстоянии друг от друга. Такие современные средства коммуникации дополняются компьютерными обучающими программами, которые замещают печатные тексты, аудио- и видеопленки. Появление такой модели дистантного образования ведет к тому, что образование осуществляется не только на расстоянии, но и независимо от какого-либо учреждения.

Такая модель еще не реализована полностью. Она сталкивается с существенными трудностями, в частности, проблемой получения общественного признания и права выдавать дипломы и сертификаты, присваивать соответствующие степени (проблема аккредитации виртуального университета). Преодоление этих трудностей и полное развитие модели виртуального университета будет означать глубокие изменения в организационной структуре современного образования.

8.10 Дистанционное образование в России

Современный этап развития Российской высшей школы характеризуется очень интенсивным взаимопроникновением методик образования западной школы в Российскую и наоборот. В России активно развиваются крупные университетские центры по образу ведущих центров США и Европы. Для современного этапа характерно создание ведущими ВУЗами своих филиалов. Это резко расширяет рынок образовательных услуг и экономит средства вкладываемые в образование, но ведет к ухудшению качества образования, если не внести коррективы в методы образования.

Возможное ухудшение качества образования обусловлено следующими факторами:

- возможным отсутствием в филиалах достаточного количества квалифицированных педагогических кадров;
- невозможностью быстрого создания необходимой материальной учебно-лабораторной базы в филиале;
- экономической нецелесообразностью развертывания в филиале полнокомплектных лабораторных комплексов и лекционных мультимедийных систем из-за малого числа студентов;
- отсутствие в филиалах традиций и опыта постановки

и проведения научно-исследовательских и учебных работ и экспериментов.

Разрешение сложившейся проблемы возможно на основании внедрения в сферу образования дистанционного обучения на базе новых информационных технологий и современного подхода к созданию и функционированию учебного процесса. Основные направления такого подхода:

- информатизация имеющегося учебного и научного лабораторного оборудования на базе современных средств и технологий;

- разработка нового поколения учебной техники с использованием компьютерных моделей, анимаций и физического моделирования исследуемых объектов, процессов и явлений, ориентированных на решение следующих задач: акцентирование внимания на физической стороне исследуемого процесса; сокращение рутинной части образовательного процесса за счет автоматизации систем управления, измерения и обработки результатов; лабораторный стенд должен охватывать большой раздел лабораторных работ прикладного тематического направления; лабораторные стенды должны обладать системой телекоммуникации, обеспечивающий режимы удаленного и коллективного использования оборудования, интегрируя лабораторные стенды в систему дистанционного образования;

Методология образования должна поддерживать компьютерные формы обучения, контроля знаний, получения индивидуального задания, моделирования изучаемых процессов, проведения эксперимента, анализа и обработки результатов эксперимента, в том числе и в режиме удаленного доступа.

Создание системы удаленного доступа филиалов ВУЗов и небольших ВУЗов к ресурсам своих базовых ВУЗов и через них к ведущим учебным и научным лабораторно-

исследовательским центрам страны.

Эти три направления (*компьютеризация оборудования, методология образования на базе информационных средств, компьютерных форм и удаленный доступ*) являются сутью концепции создания и внедрения комплекса дистанционного образования.

Приведем один из примеров внедрения системы ДО в Российские ВУЗы. Южно-Уральский Государственный Университет (ЮУрГУ) крупнейший ВУЗ России в своем составе содержит 30 факультетов, более 100 кафедр, около десятка филиалов в городах Южного Урала и Западной Сибири. В соответствии с принятой концепцией развития ВУЗа как университетского центра создается необходимая система телекоммуникации. Университет имеет выделенные каналы связи почти со всеми филиалами и оптоволоконные линии внутри университетского городка.

Удаленный доступ и мультимедиа в методическом подходе университета трактуется более широко. Мультимедиа не только в понятии компьютер индивидуального пользователя, а это системы подачи информации на прозрачных носителях через видеопроекторы и через мультимедийные крупноформатные видеостенки в поточных аудиториях. Такие видеостенки через корпоративную сеть университета имеет выход на любой сервер университета или в INTERNET и позволяет транслировать телеконференции между корпусами и лабораториями университета и его филиалами без специального резервирования каналов.

Дистанционное обучение (ДО) более приближено к обычному обучению, т.к. кроме классического ДО (кейсовая система и использование компьютерных сетей для выдачи заданий, тестов и программ методик, не требующих специальной техники, выходящей за пределы корпуса компьютера) активно используются физические

компьютеризованные лабораторные комплексы с удаленным доступом и имитаторы лабораторных работ. Эти работы нашли свое развитие в нетрадиционных для ДО курсах таких как: “Сопrotивление материалов”, “Детали машин и основы конструирования”, “Станки и инструменты”, “Электрические сети и станции”, “Радиолокация” и другие. Элементы видео и аудио сопровождения позволяют создать эффект присутствия и взглянуть в (посмотреть на эксперимент) с нетрадиционной точки зрения. Найден нетрадиционный подход к совершенствованию видеокурсов, как основного компонента "кейсовой" системы классического ДО. С этой целью создан специализированный видео-комплекс со студией нелинейного мoнажа.

8.11 Электронный учебник как средство дистанционного обучения

Как говорилось выше, мы разберем одну из наиболее часто применяемых технологий в ДО и в очном образовании также. А именно, это электронный учебник, наиболее часто встречающаяся форма представления нового материала. Кроме этого ЭУ может включать одновременно тренажеры, лабораторные работы, а также тесты; т.е. одновременно – это и ПО по предоставлению знаний и по их контролю.

Основные принципы дистанционного обучения (ДО): установление интерактивного общения между обучающимся и обучающим без обеспечения их непосредственной встречи и самостоятельное освоение определенного массива знаний и навыков по выбранному курсу и его программе при заданной информационной технологии.

Главной проблемой развития дистанционного обучения является создание новых методов и технологий обучения, отвечающих телекоммуникационной среде общения. В этой среде ярко проявляется то обстоятельство, что учащиеся не

просто пассивные потребители информации, а в процессе обучения они создают собственное понимание предметного содержания обучения.

На смену прежней модели обучения должна прийти новая модель, основанная на следующих положениях: в центре технологии обучения — учащийся; суть технологии — развитие способности к самообучению; учащиеся играют активную роль в обучении; в основе учебной деятельности — сотрудничество.

В связи с этим требуют пересмотра методики обучения, модели деятельности и взаимодействия преподавателей и обучаемых. Я считаю ошибочным мнение многих российских педагогов-практиков, развивающих технологии дистанционного образования, что дистанционный учебный курс можно получить, просто переведя в компьютерную форму учебные материалы традиционного очного обучения.

Успешное создание и использование дистанционных учебных курсов должно начинаться с глубокого анализа целей обучения, дидактических возможностей новых технологий передачи учебной информации, требований к технологиям дистанционного обучения с точки зрения обучения конкретным дисциплинам, корректировки критериев обученности.

Дидактические особенности курса ДО обуславливают новое понимание и коррекцию целей его внедрения, которые можно обозначить следующим образом:

- стимулирование интеллектуальной активности учащихся с помощью определения целей изучения и применения материала, а также вовлечения учащихся в отбор, проработку и организацию материала;
- усиление учебной мотивации, что достигается путем четкого определения ценностей и внутренних причин, побуждающих учиться;
- развитие способностей и навыков обучения и

самообучения, что достигается расширением и углублением учебных технологий и приемов.

К числу дидактических принципов, затрагиваемых компьютерными технологиями передачи информации и общения, в первую очередь следует отнести:

- принцип активности;
- принцип самостоятельности;
- принцип сочетания коллективных и индивидуальных форм учебной работы;
- принцип мотивации;
- принцип связи теории с практикой;
- принцип эффективности.

В связи с этими принципами средства учебного назначения, которые используются в образовательном процессе ДО, должны обеспечивать возможность:

- индивидуализировать подход к ученику и дифференцировать процесс обучения;
- контролировать обучаемого с диагностикой ошибок и обратной связью;
- обеспечить самоконтроль и самокоррекцию учебно-познавательной деятельности учащегося;
- демонстрировать визуальную учебную информацию;
- моделировать и имитировать процессы и явления;
- проводить лабораторные работы, эксперименты и опыты в условиях виртуальной реальности;
- прививать умение в принятии оптимальных решений;
- повысить интерес к процессу обучения;
- передать культуру познания и др. Хотелось бы подчеркнуть особую важность определения целей курса.

Для построения четкого плана курса необходимо:

- определить основные цели, устанавливающие, что учащиеся должны изучить;
- конкретизировать поставленные цели, определив, что учащиеся должны уметь делать;

- спроектировать деятельность учащегося, которая позволит достичь целей.

Очень важно добиваться того, чтобы поставленные цели помогли определить, что ожидается от учащихся после изучения этого курса. Конкретизация целей позволяет дать представление о том, что учащийся в состоянии будет сделать в конце каждого урока. Фактически необходима постановка целей для каждого урока курса.

Цели помогают сконцентрироваться на развитии познавательной деятельности учащихся и определить, на какой стадии он находится.

Правильно сформулированные цели позволят учащимся:

- настроить мышление на тему обучения;
- сфокусировать внимание на наиболее важных проблемах;
- тщательно подготовиться к тестам, заданиям и другим средствам оценивания.

Деятельность должна быть спроектирована в соответствии со сформулированными целями.

При планировании и разработке дистанционных учебных курсов необходимо принимать во внимание, что основные три компоненты деятельности педагога, а именно *изложение учебного материала, практика, обратная связь*, сохраняют свое значение и в курсах ДО.

Разработанный и реализованный подход к дистанционному обучению заключается в следующем:

- перед началом дистанционного обучения производится психологическое тестирование учащегося с целью разработки индивидуального подхода к обучению;
- учебный материал представлен в структурированном виде, что позволяет учащемуся получить систематизированные знания по каждой теме;

Содержание предлагаемого к освоению курса

дистанционного обучение педагогически отработано и систематизировано и состоит из комплекса психологических тестов, программы обучения и электронного учебника, который удовлетворяет вышеизложенным принципам.

Первоначально обучающемуся высылаются комплекс психологических тестов и пробный урок. Полученные результаты психологического тестирования обрабатываются и на основе этого строится психологический портрет учащегося, с помощью которого выбираются методы и индивидуальная стратегия обучения.

Программа обучения — один из наиболее важных видов раздаточных материалов для учащихся, обучающихся дистанционно. Учащиеся обращаются к ней для получения точной и ясной информации. Такое руководство включает в себя:

- информацию о системе дистанционного обучения, методах ДО;
- биографическую информацию о преподавателе;
- технологию построения учебного курса;
- цели курса;
- критерии окончания обучения;
- часы телефонных консультаций;
- описание экзаменов, проектов письменных работ;
- другие инструкции.

Электронный учебник, содержит собственно учебные материалы для дистанционного обучения, разделен на не зависимые темы-модули, каждая из которых дает целостное представление об определенной тематической области, и способствует индивидуализации процесса обучения, т.е. обучающийся может выбрать из вариантов обучения: изучение полного курса по предмету или изучение только конкретных тем.

Огромное количество фактов, примеров приведенных выше показывают необходимость создания и расширения ДО

в России и ее регионах, как неотъемлемый фактор развития квалифицированного, интеллектуального, высоко профессионального и просто здорового общества. В данной работе я рассмотрела ДО с педагогической и, кроме того, с технической точек зрения и убедилась в важности решения данной проблемы.

Дистанционное образование открывает студентам доступ к нетрадиционным источникам информации, повышает эффективность самостоятельной работы, дает совершенно новые возможности для творчества, обретения и закрепления различных профессиональных навыков, а преподавателям позволяет реализовывать принципиально новые формы и методы обучения с применением концептуального и математического моделирования явлений и процессов /30, 31, 32/.

9. КОМПЬЮТЕРНОЕ ТЕСТИРОВАНИЕ. МЕТОДЫ И АЛГОРИТМЫ

9.1 Компьютерное тестирование

Идея компьютерного тестирования напрямую проистекает от идеи программированного контроля знаний. Программированный контроль знаний, в свою очередь, явился неизбежной реакцией на некоторые проблемы прежде всего высшего образования в России. Собственно, примерно те же проблемы распространяются и на школьное образование, но последнее, в силу традиционной косности, очень слабо восприимчиво к новым технологиям.

Основной проблемой любого образования (и не только российского, кстати), является отсутствие четкого контроля за качеством усвоения материала. Причем если в школьной практике учитель еще более-менее имеет возможность с определенной периодичностью проверять уровень текущих

знаний ученика, то в ВУЗе преподаватель целый семестр выдает материал и лишь в конце семестра убеждается в уровне его усвоения. Само собой, в системе высшего образования подразумевается, что студенты должны в достаточной степени заниматься и самостоятельным образованием, однако, это предполагаемое самостоятельное получение знаний остается целиком и полностью на совести студента, и преподаватель абсолютно не может знать, кто именно из студентов хоть что-то делает самостоятельно. С получением большим числом обучаемых доступа в Internet положение усугубилось еще и тем, что теперь даже сдача рефератов не подразумевает абсолютно никакой работы с информацией; частенько студенты даже не считают нужным целиком прочесть то, что распечатывают из Сети /33/.

Необходимость систематического контроля за усвоением материала сомнений не вызывает. Прежде всего это давало бы экономию времени преподавателя, который при отсутствии обратной связи вынужден или повторять положения, которые студентами давно усвоены, или излагать положения, основанные на фактах, плохо усвоенных студентами. Во вторую очередь, систематический контроль за уровнем знаний учащихся стимулирует повышение качества обучения за счет усиления акцента на трудных для усвоения положениях и повышении ответственности обучаемых за результаты самостоятельной работы (в случае, естественно, когда преподаватель в этом заинтересован).

Важным моментом систематического программированного контроля знаний является его объективность, что обусловлено переносом акцента с карательной функции на информативную. Только в таком случае учащийся не будет бояться контроля и изобретать способы получения повышенной оценки, и только в таком случае преподаватель будет получать реальную картину знаний учащегося.

Технически программированный контроль знаний прост - учащимся выдается некий бумажный носитель (расцвет программированного контроля вызвал к жизни релейно-ламповых "электронных" монстров, которые по сей день можно видеть на экзаменах по сдаче на водительские права), на котором записаны вопросы и варианты ответов, один (или несколько) из которых являются правильными. Учащемуся остается лишь расставить крестики против правильных ответов.

Подобная технология позволила совершить качественный скачок в осуществлении обратной связи между преподавателем и студентом. Программированный контроль, состоящий из 8-10 вопросов, проводится за очень короткий срок - от 5 до 10 минут, и при этом преподаватель может получить полноценную информацию об усвоении пройденного материала всей учебной группой одновременно. Кроме того, техническая реализация программированного контроля позволила полностью избежать списывания, давая возможность предложить каждому учащемуся свой вариант программированной карты.

Недостатком программированного контроля в его до-компьютерном виде являлась высокая трудоемкость создания программированных карт, которые (в идеале) требовались на каждое занятие, и сложность их последующей обработки. С появлением компьютерных технологий у преподавателей появилась возможность резко снизить трудоемкость и подготовки контроля, и обработки результатов.

Выделяют пять общих требований к тестам:

- валидность;
- определенность (общепонятность);
- простота;
- однозначность;
- надежность.

Валидность теста – это адекватность. Различают

содержательную и функциональную валидность: первая – это соответствие теста содержанию контролируемого учебного материала, вторая – соответствие теста оцениваемому уровню деятельности.

Выполнение требования определенности (общедоступности) теста необходимо не только для понимания каждым учеником того, что он должен выполнить, но и для исключения правильных ответов, отличающихся от эталона.

Требование простоты теста означает, что тест должен иметь одно задание одного уровня, т.е. не должен быть комплексным и состоять из нескольких заданий разного уровня. Необходимо отличать понятие “комплексный тест” от понятия “трудный тест”. Трудность теста принято характеризовать числом операций P , которое надо выполнить в тесте: $P < 3$ – первая группа трудности; $P = 3-10$ – вторая группа трудности. Не следует также смешивать понятия простоты-комплексности и легкости-трудности с понятием сложности.

Однозначность определяют как одинаковость оценки качества выполнения теста разными экспертами. Для выполнения этого требования тест должен иметь эталон. Для измерения степени правильности используют коэффициент $K_{\alpha} = P_1 / P_2$, где P_1 – количество правильно выполненных существенных операций в тесте или батарее тестов; P_2 – общее количество существенных операций в тесте или батарее тестов. Существенными считают те операции в тесте, которые выполняются на проверяемом уровне усвоения. Операции, принадлежащие к более низкому уровню в число существенных не входят. При $K_{\alpha} \geq 0.7$ считают, что деятельность на данном уровне усвоена.

Понятие надежности тестирования определяют как вероятность правильного измерения величины K_{α} . Количественный показатель надежности $r \in [0, 1]$.

Требование надежности заключается в обеспечении устойчивости результатов многократного тестирования одного и того же испытуемого. Надежность теста или батареи тестов растет с увеличением количества существенных операций $P/34/$.

Итак, при реализации систем компьютерного тестирования необходимо, на мой взгляд, придерживаться именно этих пяти требований к создаваемым тестам. Но проблема компьютерного тестирования стоит намного острее. Реализация в системах тестирования описанных выше пяти требования к тестам не означает того, что созданный комплекс будет отвечать всем требованиям преподавателя и учащегося.

Большинство программных продуктов не дают возможности преподавателю и студенту, учителю и ученику отойти в реальном учебном процессе от традиционных методик: лекционного курса, конспекта, очного контроля знаний, контрольных работ, зачетов, экзаменов. Недостаток этот можно определить следующим: компьютерный курс является авторским по определению, и поэтому обеспечивает высокое качество образования только при соответствующем сопровождении автором (который, в большинстве случаев, не обладает достаточными знаниями в области информационных технологий). Хотя отдельные компоненты компьютерного обучающего, контролирующего или обучающе-контролирующего курса могут использоваться как независимые учебные модули другими преподавателями (а также и при самостоятельном освоении темы), максимальный эффект, скорее всего, может быть достигнут только во взаимодействии с автором- разработчиком курса.

Если же в образовательный процесс, основанный на авторском мультимедиа курсе, включается другой преподаватель, возникает опасность конфликта личностей, так как на едином образовательном поле сталкиваются не

только различные способы методической организации учебного процесса, но и разные личностные подходы.

Что касается проверки качества знаний, неформальный характер процесса оценивания знаний требует применения трудно поддающихся обработке преподавателем компьютерных тестов, необходима активная обратная связь, помогающая оценить правильность усвоения материала, должна быть четко выражена определенность и результативность /35/.

Именно неформальность знаний как таковых, и процесса проверки знаний в частности, породило множество проблем в области компьютерного тестирования, таких как необъективность оценивания, трудность понимания учащимися подготовленных вопросов, медленная работа компьютерных систем, и т.п.

9.2 Методы и модели тестирования

Интеллектуальное тестирование предполагает наличие модели знаний, модели самого процесса тестирования и оценивания. Так можно охарактеризовать в общем все разработки в этой области. Рассмотрим некоторые из них более подробно.

9.2.1 Модели распознавания образа уровня знаний

Традиционная Российская система оценивания знаний обучаемых основана на лингвистических оценках, по которым устанавливается стипендия, производится учет успеваемости, проставляются записи в зачетных книжках за период обучения и др.

Вместе с тем, такая новая образовательная процедура как образовательное тестирование по альтернативному признаку предполагает оценивание уровня знаний в

диапазоне от нуля до ста, что порождает проблему распознавания лингвистического образа знаний по результатам такого образовательного тестирования.

Под образом уровня знаний понимаются обучаемые, принадлежащие к множеству (группе), знания которых по “эталону уровня знаний” отнесены к лингвистическим оценкам неудовлетворительно (**D**), удовлетворительно (**C**), хорошо (**B**), отлично (**A**).

Под распознаванием образа уровня знаний понимается процедура принятия решения о принадлежности конкретного обучаемого к одному из указанных образов на основании сравнения его образовательных достижений при тестировании с характеристиками образа.

При тестировании по альтернативному признаку используется закрытая форма теста, характеристиками которой являются: функция плотности распределения неправильных ответов $f(d)$, приемлемый уровень неправильных ответов q_0 , неприемлемый уровень неправильных ответов q_1 , риск заниженной оценки знаний a , риск завышенной оценки знаний b , функция оценивания знаний $f(Q)$, объем образовательной информации N , объем выборки заданий теста n и критерий принятия решений в виде предельного числа неправильных ответов K .

Перечисленные характеристики являются взаимозависимыми, но не обладающими достаточным свойством четкости. В условиях их нечеткости для распознавания образа уровня знаний обучаемых вполне допустимо для нормально реализованной образовательной услуги принять модель распределения неправильных ответов по закону редких случайных событий Пуассона и функцию оценивания уровня знаний сформировать по этому же закону /36/.

Поскольку образовательная информация в банке заданий теста N в их выборке n представляется как

статистическая совокупность, а задания теста обучаемому в компьютерном варианте всегда для выполнения выдаются последовательно, то для распознавания образа уровня знаний, возможно воспользоваться последовательным критерием Вальда. При этом прием дополнительное принципиальное условие, что задания теста однородны по количеству образовательной информации по конкретной учебной дисциплине, поскольку аналитических методов классификации заданий по мере их сложности или трудности пока не разработано.

Будем обозначать гипотезу о приемлемом уровне знаний H_0 , а гипотезу о неприемлемом уровне знаний H_1 . Пусть в результате последовательного поступления заданий теста в объеме n получены неправильные ответы $d_1, d_2, d_3 \dots d_n$. При известной функции оценивания знаний по закону Пуассона последовательный критерий Вальда позволяет по выборке объемом n классифицировать обучаемых по уровню знаний на три подобраза по количеству областей принятия решений. Для того, чтобы иметь четыре образа необходимо произвести для каждой из трех областей повторное последовательное тестирование /37/.

В предлагаемой процедуре рекомендуется использовать два способа распознавания образа уровня знаний: нормальный и усиленный. При этом задаются только четыре исходные характеристики теста q_1, q_0, a и b .

По нормальному способу по первой выборке заданий теста n_1 производится классификация обучаемых на три предварительные области (уровни): низкая, нормальная и высокая. По второй выборке заданий теста $n_2 = n_1$ или $n_2 < n_1$ для уровня низкий ужесточаются исходные характеристики q_0 и q_1 и обучаемые аттестуются по трем образам D, C и B. Для нормального уровня ужесточаются характеристики a и b и обучаемые аттестуются по трем образам C, B и A. Для высокого уровня тестирование осуществляется без изменения

исходным q_0 , q_1 , **a** и **b** и обучаемые аттестуются на два образа В и А. К достоинствам нормального способа относится то, что обучаемые по второй выборке могут существенно улучшить свои образовательные достижения, что отвечает требованиям закона «О защите прав потребителей продукции и услуг».

По усиленному способу по первой выборке обучаемые классифицируются только на два уровня: низкий и высокий. По второй выборке для низкого уровня ужесточаются характеристики q_0 и q_1 и обучаемые аттестуются только на два образа D и C. По второй выборке для высокого уровня ужесточаются характеристики **a** и **b** и обучаемые аттестуются только на два образа В и А. Достоинством усиленного способа является более уверенное распознавание образа уровня знаний и поэтому его рекомендуется использовать в тех случаях, когда снижено доверие к реализуемой образовательной услуге.

Существенным отличием предлагаемых методов распознавания образа уровня знаний при тестировании от известных является заранее заданная погрешность распознавания, заложенная в рисках принятия решений **a** и **b**, использование наиболее мощного критерия Вальда и достаточно простые и апробированные в статистическом приемочном контроле способы ужесточения планов тестирования, что позволяет использовать для решения практических задач распознавания образа уровня знаний международный стандарт ИСО 8423-91 «Статистические методы. Последовательные планы выборочного контроля по альтернативному признаку». Это способствует повышению достоверности компьютерного тестирования, что позволяет использовать предложенные теоретические разработки для обоснования методов оценивания знаний в предлагаемом едином экзамене довузовского образования, в процессе реформирования Российской системы образования /37/.

9.2.2 Предметно - критериальная методика составления тестов

В каждом курсе есть ключевые моменты, особенно важные темы, без знания которых невозможно усвоение более сложного материала в процессе учебы или которые будут необходимы в работе по специальности. На устном экзамене при личном контакте со студентом преподаватель обязательно оценивает понимание студентом этих тем. При автоматизированном тестировании можно учесть важность каких-либо разделов курса, увеличив долю вопросов по этим разделам в общем количестве вопросов. Но это не всегда удобно для составителя теста, потому что не всегда наиболее важные разделы содержат больше всего материала.

Предлагаемая методика предусматривает учет таких параметров, как степень важности и объем изучаемого материала в разделах курса.

При составлении теста преподаватель делит курс на темы T_1, T_2, \dots, T_k и оценивает степень важности S_i и объем изучаемого материала V_i по каждой теме T_i . Количество вопросов n_i по каждой теме T_i должно соответствовать (быть пропорционально) объему изучаемого материала V_i .

Минимальное количество вопросов n_i по каждой теме T_i определяется в соответствии с методикой с учетом параметра V_i .

Знания по каждому разделу курса оцениваются по пятибалльной (а фактически по четырехбалльной) системе. Оценке «отлично» (5) соответствует вероятность правильного ответа от p_3 до 1; оценке «хорошо» (4) соответствует вероятность правильного ответа от p_2 до p_3 ; оценке «удовлетворительно» (3) соответствует вероятность правильного ответа от p_1 до p_2 ; оценке «неудовлетворительно» (2) соответствует вероятность

правильного ответа менее p_1 . Следует отметить, что вероятности p_1 , p_2 и p_3 ($0 < p_1 \leq p_2 \leq p_3 < 1$) задаются преподавателем с учетом структуры теста и могут быть изменены. Абсолютное количество (или доля) правильных ответов, достаточное для получения соответствующей оценки, определяется по специальной методике.

Итак, преподаватель:

- разбивает курс на темы (разделы) T_1, T_2, \dots, T_k ;
- определяет их объемы V_1, V_2, \dots, V_k и степень важности S_1, S_2, \dots, S_k ;
- определяет структуру теста – количество m вариантов ответов на каждый вопрос;
- задает p_1, p_2, p_3 – уровни знаний студента (или вероятности выбора правильного ответа), соответствующие оценкам: “2” – $0 \leq p < p_1$, “3” – $p_1 \leq p < p_2$, “4” – $p_2 \leq p < p_3$, “5” – $p_3 \leq p \leq 1$.

p_1 должно быть заметно больше $1/m$ – вероятности выбора правильного ответа наугад.

После этого вычисляется минимальное количество вопросов n , необходимое для того, чтобы при заданных параметрах m, p_1, p_2, p_3 и заданном уровне значимости ϵ на основании испытания статистических гипотез можно было поставить оценку «5», «4», «3» или «2» за определенный раздел курса.

Минимальное количество вопросов n будет содержать тест по теме с минимальным значением $V_j = \min\{V_1, V_2, \dots, V_k\}$; $n_j = n$.

Минимальное количество вопросов по темам T_1, T_2, \dots, T_k определяется пропорционально их объемам, V_1, V_2, \dots, V_k .

По ответам студента вычисляется оценка O_i по каждой теме T_i ($1 \leq i \leq k$) как результат испытания статистических гипотез

При вычислении итоговой отметки за тест (курс) O учитывается степень важности S_i каждого раздела T_i .

Получившаяся итоговая оценка O округляется до целых.

Следует отметить, что описанная выше методика позволяет давать студентам тест поэтапно, по мере изучения и усвоения материала отдельных разделов курса, и выводить итоговую оценку с учетом результатов промежуточного тестирования /38/.

9.2.3 Метод определения количества образовательной информации

Теория образовательного тестирования должна формироваться на частных законах и закономерностях таких научных направлений как информатиология, общая статистика, статистический приемочный контроль, квалиметрия, педагогика, психология, исследование операций, теория принятия решений и др. Прямое применение теоретических разработок из указанных научных направлений не дает заметных практических результатов по оцениванию знаний по причине нематериальности знаний, как объекта исследований. Задачу формирования теории образовательного тестирования можно сформулировать как задачу поиска оптимальной структуры специфических законов и закономерностей тестологии, позволяющую оценить знания с заданной погрешностью.

Для решения задач подобного класса наиболее успешно используются генетические методы, основанные на реализации генетических алгоритмов, позволяющих осуществить направленный перебор частных законов и закономерностей по наиболее приемлемым направлениям для формирования отечественной теории образовательного тестирования.

В отличие от традиционного случайного поиска приемлемых решений, алгоритмы генетического поиска используют аналоги или близость имеющихся решений во

многих областях знаний к поиску оптимального набора специфических законов, обеспечивающих объективность, достоверность и точность оценивания уровня знаний, воспроизведенных обучаемыми в процедурах тестирования. Такой направленный перебор частных законов является эволюционным и имеет очень много сходств с операторами, применяемыми в генетических алгоритмах и процедурах, происходящих с живыми организмами в природе.

9.2.4 Модель Раша

Система тестирования на основе модели Раша обладает важными достоинствами, среди которых, прежде всего, необходимо отметить следующие.

Модель Раша превращает измерения, сделанные в дихотомических и порядковых шкалах в линейные измерения, в результате качественные данные анализируются с помощью количественных методов. Это позволяет использовать широкий спектр статистических процедур.

Оценка трудности тестовых заданий не зависит от выборки испытуемых, на которых была получена и, аналогично, оценка уровня знаний испытуемых не зависит от используемого набора тестовых заданий.

Пропуск данных для некоторых комбинаций (испытуемый — тестовое задание) не является критическим.

Сама система тестирования достаточно проста, по сравнению с другими аналогичными системами она характеризуется наименьшим числом параметров — только один параметр уровня знаний для каждого испытуемого и только один параметр трудности для каждого задания.

Модель Раша опирается на четкие и конструктивные понятия "трудность задания" и "уровень знаний". Так, одно задание считается более трудным, чем другое, если вероятность правильного ответа на первое задание меньше,

чем на второе, независимо от того, кто их выполняет. Аналогично, более подготовленный студент имеет большую вероятность правильно ответить на все задания, чем менее подготовленный.

Благодаря простой структуре модели существуют удобные вычислительные процедуры для многоаспектной проверки адекватности модели: для всего набора тестовых результатов, для каждого испытуемого, для каждого задания и для каждого конкретного ответа.

"Остатки", получаемые при аппроксимации результатов тестирования моделью можно использовать для выделения различных типов испытуемых.

Однако, несмотря на 40-летний опыт применения этой системы тестирования за рубежом во многих областях знания, прежде всего в образовании, медицине и психологии, до сих пор продолжаются дискуссии об истинной ценности и эффективности системы тестирования на основе модели Раша. До сих пор существуют две крайние точки зрения на эту модель тестирования.

Наиболее убежденные сторонники модели Раша утверждают следующее: "Можно ли собрать или построить или сформулировать данные так, чтобы они соответствовали определению измерения (модели Раша)? Если нет, — то такие данные бесполезны".

Их наиболее последовательные оппоненты утверждают следующее: "Данные — это данные, а модель — это конструкция исследователя, которая подвержена ошибкам". Например, при построении регрессии, выбрасывая те или иные данные, можно получить любую зависимость, но мы тем самым ограничиваем реальный мир данных. Таким образом, мы создаем искусственную переменную, о которой мало что знаем.

Для практики одним из наиболее важных критериев является точность оценивания. Поэтому выбор темы в

значительной степени обусловлен противоречивой информацией относительно точности системы тестирования на основе модели Раша. Кроме того, не удалось найти работы, в которых проведен всесторонний анализ точности модели Раша. В известных работах только даются те или иные общие рекомендации по использованию этих моделей.

Чем больше точность, тем лучше работает модель. В случае отсутствия ошибок измерения любая модель в смысле точности измерения работает идеально. Но на практике ошибки всегда есть и поэтому важно знать, насколько точные оценки позволяет получать та или иная модель.

На основе имитационного моделирования исследуются точность оценивания уровней знаний и трудностей заданий, а также число итераций, требуемых для вычисления этих оценок (методом наибольшего правдоподобия) в многофакторной ситуации в зависимости от:

- диапазона уровней знаний испытуемых;
- диапазона трудностей заданий;
- степени соответствия диапазонов уровней знаний испытуемых и трудностей заданий;
- числа испытуемых;
- числа заданий;
- степени соответствия данных модели;
- доли пропущенных данных.

Для статистической обработки результатов моделирования используется многофакторный дисперсионный анализ /39/.

9.3 Абсолютная временная шкала измерения знаний

Знания являются абсолютной субстанцией: они либо есть, либо их нет. По крайней мере, так считается в любой форме традиционного оценивания знаний — как на выпускных экзаменах в школах, так и на вступительных

экзаменах в вузы. Поэтому интересно проанализировать возможности абсолютных шкал оценки и при переходе к *измерению* знаний на основе тестов.

В данных исследованиях изучаются возможности так называемой «абсолютной временной шкалы оценивания знаний». Формулируются ее принципы. Формулируются этапы последовательного перехода от традиционной формы экзаменов к тестовой форме этого подхода, на их основе – требования к созданию тестовых материалов этого подхода.

Анализируется опыт использования данного подхода на вступительных экзаменах в Тверском государственном университете на протяжении 4-х лет.

Изучается диагностический потенциал данного подхода. Формулируется принцип «трехуровневого абстрагирования» для диагностических тестирований. Ниже показана «диаграмма знаний» по математике, полученная в результате обработки данных тестирования выпускников одной из школ г.Твери.

Было обнаружено, что тестирование с использованием абсолютной шкалы оценки имеет ценность диагностическую даже более, чем для итоговых экзаменов.

Изучается уровень достоверности результатов компьютерного тестирования в данном подходе и соотношение «случайного» и «достоверного» в итоговой оценке. На рисунках приведена зависимость (в данном подходе) итоговой оценки по математике от времени тестирования:

Было обнаружено, за все время тестирования (40 минут) в первые 15 минут (первые 4 задания по математике) оценка менялась наиболее заметно. За последние же 10 минут итоговая оценка изменялась не более чем на 10 баллов — доля «случайного» в итоговой оценке.

Таким образом, при использовании абсолютной шкалы данного подхода существует возможность ответить на

вопросы: 1) существует ли предел, к которому сходится итоговая оценка с увеличением времени тестирования (или количества заданий теста); 2) какова погрешность «измерения знаний» если прервать тестирование в некоторый определенный момент, например через 40 минут.

Еще одно очевидное преимущество абсолютной шкалы оценивания – итоговая оценка появляется на экране компьютера сразу же после выполнения теста испытуемым /40/.

9.4 Методика статистического анализа качества обучения

Предлагаемая методика основывается на том, что учебный процесс является частным случаем технологического процесса и ему должны быть свойственны такие же методы анализа, какие приняты для производственных процессов. Однако слепо перенести подобные методики нельзя, особенно это касается содержательного анализа процесса.

Для того чтобы проанализировать учебный процесс нужно иметь, во-первых, критерий качества обучения, а, во-вторых, проследить его изменение во времени. В качестве наиболее информативного критерия качества обучения следует использовать степень обученности учащихся — СОУ. Этот критерий основан на статистике полученных учащимися оценок за выполнение отдельных заданий или контрольных работ. Оценки входят в СОУ с «весом» равным интегралу вероятности получения данной оценки для некоторого «типového» распределения оценок.

В качестве такого «типového» распределения используется стандартное распределение Гаусса с параметрами: среднее значение оценки — 4 и стандартное отклонение — 1,39 /41/. Такое распределение обладает одним

особым свойством: для этого распределения значения СОУ и качественной успеваемости совпадают и составляют 0,64. Это свойство выделяет «типовое» распределение среди других распределений со средней оценкой 4.

Расчеты для «типового» распределения показывают, что если СОУ больше 0,76, то обученность «отличная», если СОУ от 0,5 до 0,76, то обученность «хорошая», если СОУ от 0,24 до 0,5, то обученность «удовлетворительная», если менее 0,24, то «неудовлетворительная».

Для оценки изменения СОУ во времени используется известная в математической статистике методика, связанная с критерием « $3s$ ». Согласно этой методике, если какой либо процесс идет нормально, то отдельные значения должны укладываться в интервал « $3s$ » относительно среднего значения (s — стандартное отклонение) с определенной точностью. Те значения, которые не укладываются в заданный интервал, являются отклонениями от стандартного распределения. Чем меньше таких отклонений, тем больше соответствие анализируемого распределения стандартному. Что касается применения этой методики для технологических процессов, то ее надо скорректировать — следует учитывать только те значения, которые выходят за нижнюю границу интервала.

Если взять отношение числа значений попадающих в интервал « $3s$ » к общему количеству значений, то такую величину можно назвать коэффициентом стандартности распределения, а в случае рассмотрения учебного процесса — коэффициентом отлаженности учебного процесса (КОУП). Расчеты показывают, что если значение КОУП больше 0,94, то процесс можно считать «отлично отлаженным», если КОУП от 0,84 до 0,94 — «хорошо отлаженным», если КОУП от 0,69 до 0,84 — «почти отлаженным», если менее 0,69 — «не отлаженным».

Для общей оценки учебного процесса можно

перемножить среднее значение СОУ по предмету за год на КОУП. Полученную величину можно трактовать как фактор качества учебного процесса (ФКУП). Этот фактор имеет большее число градаций, чем СОУ и КОУП. «Отличному» качеству соответствует ФКУП больше 0,71, «очень хорошему» от 0,64 до 0,71, «хорошему» от 0,41 до 0,64, «удовлетворительному» от 0,17 до 0,41 и «неудовлетворительному» менее 0,17 /42/.

9.5 Модель адаптивного тестового контроля

Процедура тестирования предполагает анализ ответов на последовательность тестовых заданий определенной сложности. Проведем аналогию с поведением поискового алгоритма оптимизации для некоторой гипотетической функции Y , максимум которой необходимо найти. В задачах оценивания по тестированию — это максимум функции уровня знаний.

Реализация поискового алгоритма сводится к последовательному анализу локальной окрестности функционала Y , оценки градиента и выбора очередной области исследования. Если при оценке градиента имеют место помехи, то нельзя говорить о сходимости алгоритма. В обычном смысле он сходится вообще не будет, а будет “блуждать” вокруг области экстремума.

Аналогично можно поступить в случае тестового контроля. Если ответ правильный, то предполагается, что уровень подготовки студента выше сложности предъявленной задачи и он способен решать задачи заданной сложности, в противном случае — неспособен. Это подобно оценке градиента некоторой гипотетической функции регрессии, в которой градиент сам является случайной величиной.

Предлагается использовать следующий подход.

Считаем, что если тестируемый решил задание, то у него появляется желание решить более сложное задание. Если нет — то им будет сделана еще одна попытка решения задания той же сложности. Если оно также не решено, то предъявляется задача пониженной сложности. Если сразу не решено менее сложное задание, то к решению предлагается задача меньшей сложности. Аналогично происходит процесс повышения сложности заданий. В результате, если исключить этап обучения при решении задач, студент выберет для себя определенный уровень сложности, вокруг которого и будет размываться сложность заданий.

Таким образом, функция «уровня знаний» является преобразованием функции «сложности» задачи через «способность решения задач» определенной «сложности». В этом высказывании термины «уровень знаний», «способность решения задач» и «сложности» носят нечеткий характер. Поэтому для формализации этих понятий целесообразно использование аппарата нечетких множеств. Кроме того, в указанной постановке заметна разница между «сложностью» и «способностью решения задач».

Понятия «сложность» и «уровень знаний» — это некоторые нечеткие переменные (только переменные, хотя они и задаются функцией), в то время как «способность решения задач» является нечетким отношением нечетких переменных «сложности» и «уровня знаний». Количество баллов также является переменной, однако эта переменная может не анализироваться, поскольку является преобразованием «уровня знаний».

При моделировании ответов в настоящее время наиболее развит анализ IRT теории, которая использует для моделирования вероятностей правильных ответов логистическую кривую /43/.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Использование компьютерных технологий в образовании. Черных В. В. <http://oimoim-1.hosting.parking.ru/reader.asp?nomer=323>
2. Сайт Сибирского государственного университета путей сообщения.
http://www.ssti.ru/kpi/informatika/Content/biblio/b1/inform_man/index.html
3. Автоматизированные Системы Научных Исследований. Для чего нужны АСНИ?
http://pmi.ulstu.ru/new_project/new/1.html
4. Попов И.И. Информационные ресурсы и системы: реализация, моделирование, управление. М.: ТПК "Альянс". 1996.
5. Боброва Е.В. Интернет-документ как объект архивного хранения // Информационный бюллетень Ассоциации "История и компьютер". N 26/27. М., 2000. С. 109-110.
6. Ларьков Н.С. Документоведение: Учебное пособие. Часть 1. - Томск: ТГУ, 2000. С. 26-34.
7. Боброва Е.В. Интернет-документ как объект архивного хранения- С. 108-109.
8. ГОСТ 7.83-2001. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Электронные издания. Дата введения 01.07.2002.
9. Юмашева Ю.Ю. Дорога в ад вымощена благими намерениями. Нормативно-правовое рондо // Круг идей: Историческая информатика в информационном обществе. Труды Конференции Ассоциации "История и компьютер. М., 2001. С. 388.
10. Учебник по курсу "Информатика и информационные технологии" <http://stud.h16.ru/education/Book1/>
11. <http://www.citforum.ru/hardware/neirocomp/index.shtml>

12. Образовательный математический сайт
<http://www.exponenta.ru>
13. Википедия — свободная энциклопедия. Квантовый компьютер. <http://ru.wikipedia.org>
14. Википедия — свободная энциклопедия. Биокомпьютер.
<http://ru.wikipedia.org>
15. Платы сбора данных. С.Н. Шиляев, П.И. Руднев.
Аппаратные средства – «Мир ПК», №3, 1993 г.
16. Дьяконов В. П. Справочник по алгоритмам и программам на языке бейсик для персональных ЭВМ. — М.: Наука, 1987. С. 240.
17. Википедия — свободная энциклопедия. Аппроксимация.
<http://ru.wikipedia.org>
18. Википедия — свободная энциклопедия. Слайны.
<http://ru.wikipedia.org>
19. Сайт издательства «Питер» www.piter.com.
20. http://twf.mpei.ac.ru/ochkov/Mathcad_12/Гурский_Турбин_а_Вычисления_в_Маткад_Глава_16.pdf
21. В.Очков. График на расчетном сайте. Глава 3.
<http://twf.mpei.ac.ru/ТТНВ/5/3/index.html>
22. <http://www.neosoft.ru/text/tex1.htm>
23. <http://tex.msu.ru/texts/texbook.html>
24. <http://xtalk.msk.su/~ott/ru/tex/index.html>
25. С.А.Короткий Концепция построения комплексных информационных систем.
<http://www.cfin.ru/vernikov/kias/vest.shtml>
26. Иллюстрированный самоучитель по Flash MX
27. С.Д. Кузнецов. Основы современных баз данных. Информационно-аналитические материалы.
<http://www.citforum.ru/database/osbd/contents.shtml>
28. Интернет и базы данных. Часть 02. Нормализация базы данных. <http://www.wwwmaster.ru/article.php?nart=21>
29. Гозман Л.Я., Шестопап Е.Б. Дистанционное обучение на пороге XXI века. Ростов – на – Дону: «Мысль», 1999. –

- 368 с.
30. Кларин М.В. Инновации в обучении. Метафоры и модели. М.: «Наука», 1997. – 398 с.
 31. Педагогика: Учебное пособие для студентов педагогических учебных заведений / В.А.Сластенин, И.Ф.Исаев, А.И.Мищенко, Е.Н.Шиянов. – 4-е изд. – М.: Школьная Пресса, 2002. – 512 с.
 32. Шахмаев Н.М. Технические средства дистанционного обучения. М. – «Знание», 2000. – 276 с.
 33. Ваньков Е.А. «Технологии компьютерного тестирования» // реферат;
 34. Кузнецов А. А. «Универсальная автоматизированная обучающая система. Подсистема контроля знаний.» // дипломный проект;
 35. Соловей Е.В. «Автоматизированная система контроля знаний «Цепь знаний». Сетевая версия» // дипломный проект;
 36. Моисеев В.Б., Пятирублевый Л.Г., Таранцева К.Р. «Информационный подход к выбору решений в системах адаптивного тестирования».
 37. <http://ito.edu.ru/2001/ito/VI/VI-0-28.html>
 38. <http://ito.edu.ru/2001/ito/VI/VI-0-19.html>;
 39. <http://ito.edu.ru/2001/ito/VI/VI-0-1.html>;
 40. <http://ito.edu.ru/2001/ito/VI/VI-0-32.html>
 41. Касьянова Н. В. «Создание системы компьютерного контроля как результат новых информационных технологий в обучении», Восточноукраинский Национальный Университет (ВНУ), Украина, г.Луганск // материалы конференции ИТО-2001.
 42. <http://ito.edu.ru/2001/ito/VI/VI-0-2.html>;
 43. <http://ito.edu.ru/2001/ito/VI/VI-0-17.html>

ОГЛАВЛЕНИЕ

Введение	3
1. Компьютерные технологии в современном обществе	6
1.1 Представление об информационном обществе	6
1.2 Как понимают ученые информационное общество	10
1.3 Роль информатизации в развитии общества	12
1.4 Об информационной культуре	19
2. Компьютерные технологии в науке	22
2.1 Автоматизированные системы научных исследований	22
2.2 Цели создания АСНИ	25
2.3 Определение АСНИ	27
2.4 Функции АСНИ	27
2.5 Структура АСНИ	28
2.6 Основные принципы создания АСНИ	32
2.7 Модель научных исследований	37
2.8 Научные ресурсы Интернет	38
3. Современные компьютерные системы	45
3.1 Архитектура современного персонального компьютера	45
3.2 Магистрально-модульный принцип построения современного компьютера	48
3.3 Периферийные и внутренние устройства	50
3.4 Типы и назначение компьютеров	59
3.5 Нейрокомпьютеры	62
3.6 Модели нейронных сетей	68
3.7 Алгоритмы обучения перцептрона	73
3.8 Квантовые компьютеры	75
3.9 Биокомпьютеры	80
4. Сбор и обработка информации	82
4.1 Сбор и обработка экспериментальных результатов. Платы сбора данных	82
4.2 Аппаратные средства ПСД	86
4.3 Параметры аналогового тракта ПСД	87
4.4 Обработка экспериментальных результатов	91
4.4.1 Интерполяция	91
4.4.2 Способы интерполяции	93
4.5 Сглаживание данных эксперимента	96
4.6 Аппроксимация	98
4.7 Сплайн	100

4.8 Интерполяция сплайнами	101
4.9 Линейный сплайн	101
4.10 Сплайн Эрмита	102
4.11 Кубический сплайн	103
4.12 Сплайн Акимы	104
4.13 B-сплайн	106
4.14 Оцифровка графических данных. Программное обеспечение	107
4.15 Оцифровка графиков средствами MathCAD	113
4.16 Оцифровка графиков другими средствами	117
4.17 Процесс оформления научных работ и используемые программные средства. Редактор TEX	121
4.17.1 Как проходит работа с системой TEX	123
4.17.2 Основные понятия работы с LATEX	124
5. Современные алгоритмические технологии	138
5.1 Технологии построения корпоративных информационных систем	138
5.2 Функционал КИС как определяющий фактор выбора ее структуры	140
5.3 Создание инфосистем на основе системы автоматизации деловых процессов	143
5.4 Функциональные подсистемы КИС	146
6. Пакет WEB-дизайна FLASH-MX	158
6.1 Основы работы с программой FLASH-MX. Основные понятия. Объект, символ, экземпляр	158
6.2 Последовательность действий при создании Flash-фильма	160
6.3 Создание и редактирование символов	162
6.4 Преобразование в символ существующего объекта	170
6.5 Редактирование символов и экземпляров	172
6.6 Работа с текстом	179
6.7 Работа с отдельными объектами	194
6.8 Создание анимации в пакете FLASH-MX	233
6.9 Основные элементы TimeLine. Простой и ключевой кадры	234
6.10 Анимация трансформации и анимация движения	245
6.11 Автоматическая анимация трансформации объекта	252
6.12 Публикация и экспорт фильма. Параметры публикации HTML-документа	262
6.13 Основы создания интерактивных фильмов в пакете FLASH-MX. Создание элементов интерфейса	275
7. Основные понятия реляционных баз данных	294
7.1 Общие понятия реляционного подхода к организации баз	

данных. Основные концепции и термины	295
7.2 Фундаментальные свойства отношений	300
7.3 Реляционная модель данных	303
7.4 Базисные средства манипулирования реляционными данными	307
7.5 Проектирование реляционных БД	310
7.6 Проектирование реляционных баз данных с использованием нормализации	311
7.7 Нормализация базы данных	313
8. Дистанционные технологии в образовании	316
8.1 Технологические основы дистанционного обучения	317
8.2 Дистанционное обучение в мире	318
8.3 Организационно-методические модели дистанционного обучения (ДО)	320
8.4 Организационно-технологические модели ДО	322
8.5 Виртуальные университеты	323
8.6 Дистанционное обучение в ВУЗе: модели и технологии	324
8.7 Основные типы технологий, применяемых в учебных заведениях нового типа	326
8.8 Методы дистанционного университетского образования	331
8.9 Основные типы организационных структур дистанционного образования	335
8.10 Дистанционное образование в России	338
8.11 Электронный учебник как средство дистанционного обучения	341
9. Компьютерное тестирование. Методы и алгоритмы	346
9.1 Компьютерное тестирование	346
9.2 Методы и модели тестирования	351
9.2.1 Модели распознавания образа уровня знаний	351
9.2.2 Предметно - критериальная методика составления тестов	355
9.2.3 Метод определения количества образовательной информации	357
9.2.4 Модель Раша	358
9.3 Абсолютная временная шкала измерения знаний	360
9.4 Методика статистического анализа качества обучения	362
9.5 Модель адаптивного тестового контроля	364
Библиографический список	366

Учебное издание

Буслов Вадим Александрович
Пашнева Татьяна Владимировна

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ

В авторской редакции

Компьютерная верстка В.А. Буслов

Подписано к изданию 22.12.2008
Уч.-изд. л. 20,0

ГОУВПО «Воронежский государственный технический
университет» 394026 Воронеж, Московский просп., 14