

Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Воронежский государственный технический университет»

Кафедра конструирования и производства радиоаппаратуры

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению лабораторных работ № 5-6 по дисциплине
«Информационные технологии» для студентов направления
11.03.03 «Конструирование и технология электронных средств»
(профиль «Проектирование и технология радиоэлектронных
средств»)
всех форм обучения

Воронеж 2021

УДК 004.432.2
ББК 32.97

Составители:

д-р. техн. наук М.А. Ромащенко,
канд. техн. наук А.А. Пирогов,
И.В. Свиридова

Методические указания к выполнению лабораторных работ № 5-6 по дисциплине «Информационные технологии» для студентов направления 11.03.03 «Конструирование и технология электронных средств» (профиль «Проектирование и технология радиоэлектронных средств») всех форм обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост. М.А. Ромащенко, А.А. Пирогов, И.В. Свиридова. Воронеж: Изд-во ВГТУ, 2021, 24 с.

Методические указания предназначены для выполнения лабораторных работ № 5-6 по дисциплине «Информационные технологии» студентами очной и заочной форм обучения.

Предназначены для студентов первого курса обучения.

Методические указания подготовлены в электронном виде и содержатся в файле ЛР5-6.pdf.

Ил. 4. Библиогр.: 4 назв.

УДК 004.432.2
ББК 32.97

Рецензент - О. Ю. Макаров, д-р техн. наук, проф.
кафедры конструирования и производства
радиоаппаратуры ВГТУ

*Издается по решению редакционно-издательского совета
Воронежского государственного технического университета*

1. ЛАБОРАТОРНАЯ РАБОТА №5

ВВЕДЕНИЕ В СРЕДУ ПРОГРАММИРОВАНИЯ DELPHI. ИЗУЧЕНИЕ ОСНОВНЫХ КОМПОНЕНТОВ СРЕДЫ DELPHI.

Цель работы: ознакомиться с главными окнами системы Delphi, их назначением и способом получения готовой программы. Изучить основные свойства формы. Закрепить знания о сохранении и последующим открытии проектов и модулей в системе Delphi. Получить представление об основных вкладках палитры компонентов системы Delphi.

Время работы: 8 часов.

1.1. Домашние задания и методические указания по их выполнению

Задание 1 – ознакомиться с главными окнами системы Delphi, их назначением и способом получения готовой программы.

Запустив систему Delphi перед пользователем появляются четыре рабочих окна, которые представлены на рис. 1. Вверху, во всю ширину экрана, окно управления проектом и средой разработки – главное окно Delphi. При его сворачивании сворачиваются и все остальные окна. Слева – Инспектор Объектов. В нём задаются свойства используемых в программе компонентов. И, наконец, в центре одно над другим два окна Delphi, окно формы будущей программы и окно программного кода.

Прямо перед пользователем окно, которое в Delphi называется форма. Именно форма является визуальным прообразом нашей будущей программы. Для получения готовой программы в системе Delphi необходимо сделать следующее:

- привести мышку на зелёный треугольник-стрелку на главном окне, при этом всплывает подсказка **Run (F9)**

- нажать эту пиктограмму (или F9 на клавиатуре), после чего получается готовая Windows-программа.

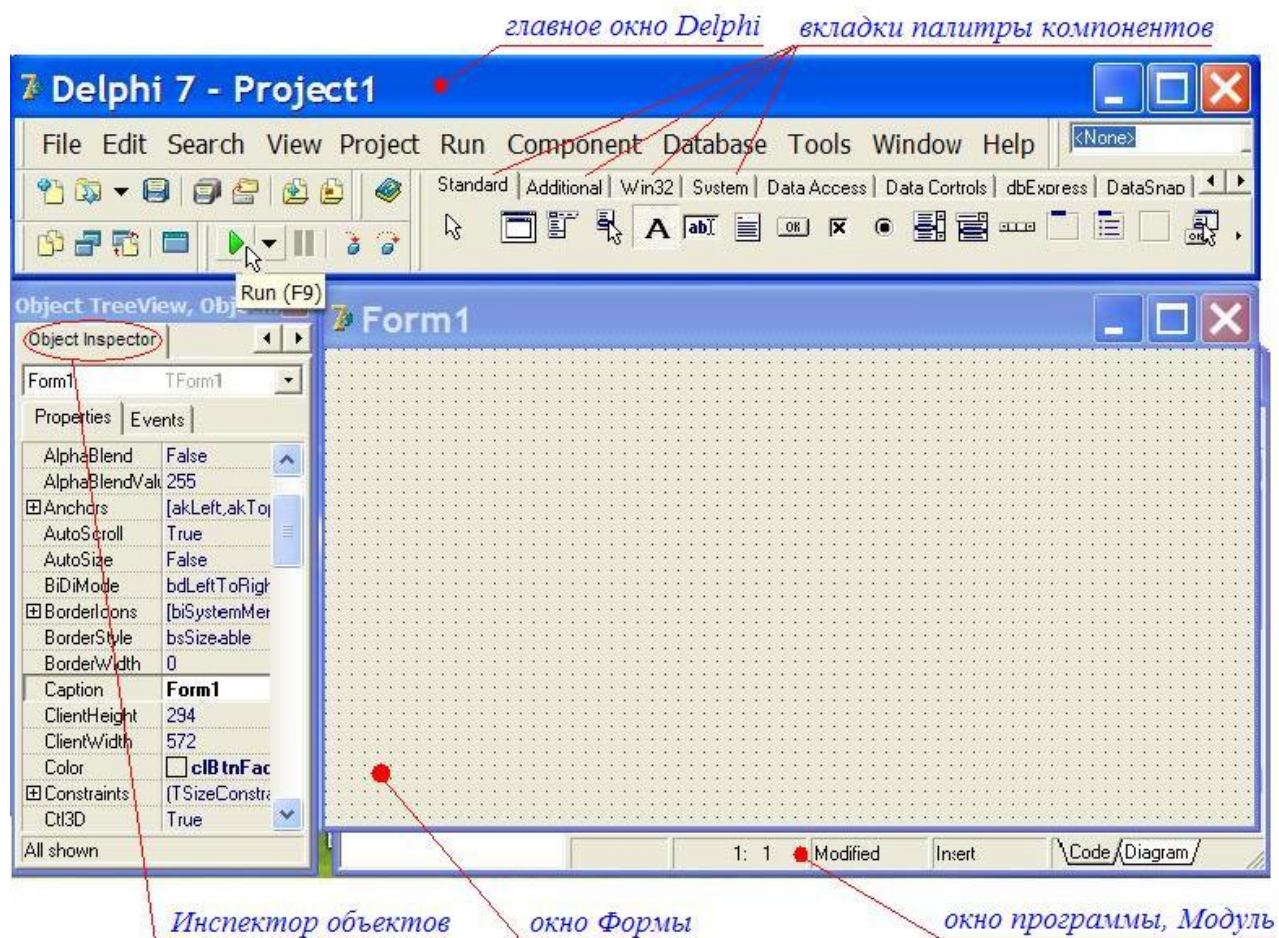


Рис. 1. Окна Delphi

По умолчанию программа создаётся в папке *C:\Program Files\Borland\Delphi7\Projects*. Полученная программа, состоит из одного пустого окна. Тем не менее, она имеет все атрибуты, присущие всем программам в Windows: заголовок, системное меню, кнопки свернуть, развернуть, закрыть, может менять свои размеры.

Задание 2 – ознакомиться с основными свойствами формы.

Рассмотрим основные свойства формы. Надпись на заголовке – свойство *Caption*. По умолчанию это *Form1*, и совпадает со свойством *Name*. В Инспекторе Объектов можно поменять их по своему усмотрению, например, написав в свойстве *Caption* "Простейшая программа".

Цвет – свойство *Color*.

Размеры – ширина и высота в пикселях, свойства *Width* и *Height*. Изменять их можно мышкой, двигая края формы, а затем установить более точные значения в Инспекторе Объектов.

Положение на экране – свойства *Top* и *Left*. Это расстояния в пикселях от левой и верхней границы экрана соответственно. Если необходимое положение формы закрывает элементы управления Delphi, ее можно просто свернуть.

При помощи свойства *Position* можно настроить автоматическое появление формы в нужном месте. Для этого необходимо развернуть выпадающий список:

- при значении *poDesigned* форма появится в том месте, где она находилась при проектировании. Это значение по умолчанию.

- при значениях *poDesktopCenter* и *poScreenCenter* Форма появится в центре рабочего стола или центре экрана соответственно.

Форма может быть вызвана из другого окна-формы. Тогда при значениях *poMainFormCenter* она появится в центре главного окна приложения, *poOwnerFormCenter* – в центре вызывающего окна, так называемого "владельца".

Имеется возможность запрограммировать появление формы в нормальном, минимизированном или максимизированном состоянии. За это отвечает свойство *WindowState* и его значения *wsNormal*, *wsMinimized* и *wsMaximized* соответственно.

При необходимости можно удалить из заголовка окна кнопки "свернуть", "развернуть", "закреть". Их наличием или отсутствием управляет свойство *BorderIcons*. В Инспекторе Объектов слева от него расположен маленький плюсики, открывающий признак наличия или отсутствия нужной кнопки. Соответственно *True* – кнопка будет, *False* – нет.

Также можно управлять границами формы. Для этого служит свойство *BorderStyle*. При значении *bsSizeable* – будет обычная форма, со значениями по умолчанию; *bsNone* – форма вообще не будет иметь границ, включая строку заголовка; *bsSingle* – форма не может менять размеры, но может свернуться или развернуться.

Таким образом, комбинируя свойства *BorderIcons* и

BorderStyle можно, например, получить форму с неизменными размерами и без возможности свернуться и развернуться:

Задание 3 – закрепить знания о сохранении и последующим открытии проектов и модулей в системе Delphi.

По умолчанию Delphi сохраняет новые проекты в папке *C:\Program Files\Borland\Delphi7\Projects*, то если не будет выделено специальное место, то все файлы текущего и всех будущих проектов будут сохранены в одном месте, что в будущем вызовет большие сложности. Рекомендуется сразу в начале работы над новым проектом командой меню *File-»Save All* (или соответствующей кнопкой на главном окне) сохранить проект в специально созданную для него папку. Создать её можно прямо в открывшемся окошке. Проект в целом сохраняется в файле с именем *Project1*. Форма функционирует вместе с программой, описывающей её работу, так называемым модулем. Если модулей в проекте несколько, для каждого из них Delphi создаст отдельный файл, и предложит его сохранить под именем *Unit1*, *Unit2* и т.д. Для файла проекта, и файлов модулей можно дать и своё, более осмысленное название.

Для открытия ранее сохраненного проекта используется команда *Open...* Модуль откроется вместе с формой, и с ним можно будет нормально работать, но *кнопка запуска программы окажется неактивной*. Поэтому вначале нужно открывать файл проекта. Для этого есть специальная команда меню *File-»Open Project...* , или соответствующая кнопка, прямо над кнопкой запуска. Теперь в окне выбора будет только один файл, файл проекта. При его открытии будет открыт и первый модуль и первая форма проекта. Для приведенного выше примера этого достаточно. Если в проекте несколько форм, их модули и формы дополнительно открываются командой *Open...* При необходимости можно открывать модули других проектов, например, для копирования кода.

Задание 4 – закрепить знания об основных вкладках палитры компонентов системы Delphi.

Для работы с формой используются компоненты Delphi. Они располагаются на соответствующих вкладках палитры компонентов на главном окне. Все основные компоненты Delphi находятся на первых четырех вкладках *Standard*, *Additional*, *Win32*, *System*, которые представлены на рис. 2.

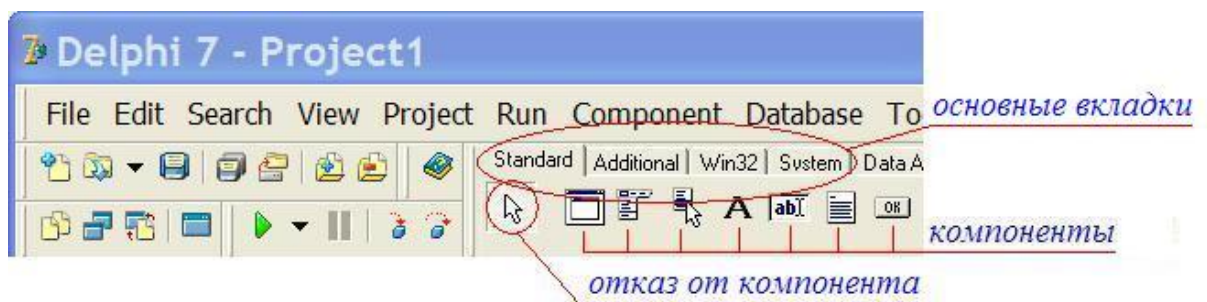


Рис. 2. Основные вкладки с компонентами Delphi

Названия компонентов всплывают в виде подсказок при наведении мышки на пиктограммы. Чтобы перенести компонент на форму, нужно щёлкнуть его мышкой (при этом выбранный компонент выделяется), и затем щёлкнуть в том месте формы, где его предполагается разместить. В дальнейшем компоненты можно свободно перетаскивать по форме мышкой, проектируя нужный интерфейс. Если, щёлкнув по компоненту, необходимо отказаться от его переноса на форму, то щёлчком по стрелке, расположенной слева на вкладке, выделение компонента снимется.

Задание 5 – ознакомиться с наиболее часто используемыми компонентами среды Delphi, такими как *Label*, *Edit*, *Memo* и *Button*.

Рассмотрим компоненты отображения текста, такие как *Label*, *Edit*, *Memo* и *Button*. Выбрав нужный компонент одним щелчком мышки, в Инспекторе Объектов мы можем получить доступ к его свойствам. Прежде всего, у визуальных компонентов есть такие свойства, как:

- ширина компонента, *Width*
- высота компонента, *Height*
- расстояние до левого края формы, *Left*
- расстояние до верхнего края формы, *Top*

Как только компоненты попадают на форму, их свойства получают значения. Примерно разместив на форме компоненты, в Инспекторе Объектов устанавливаем их точные размеры и координаты. Доступ к этим свойствам и их значениям можно получить и программно.

За визуальное представление для каждого компонента отвечают следующие свойства:

- для *Label* и *Button* свойство *Caption*
- для *Edit* свойство *Text*
- для *Memo* свойство *Lines*.

При проектировании формы в Инспекторе Объектов им можно задать любые значения, или удалить их. **Важный момент:** случайно можно задать такое значение, что компонент "пропадёт", сделается невидимым. Чтобы найти его необходимо в Инспекторе Объектов нажать кнопку выпадающего списка и найти там "потерянный" компонент, в Инспекторе Объектов появятся его свойства, и он тут же станет выделенным на форме.

2.2. Лабораторные задания

Задание 1 – Разработать простое приложение, которое по нажатию кнопки будет отображать на форме текст, который пользователь ввел в строке ввода.

Для выполнения лабораторного задания выполните следующую последовательность действий:

- создайте новый проект *File-»New-»Application*
- расположите на форме компонент *Edit* (строка ввода) и очистите его свойство *Text*
- расположите на форме компонент *Button* (кнопка). В его

свойстве **Caption** напишите, например, "Показать"

- расположите на форме компонент **Label** (метка). В его свойстве **Font** установите размер 24, выберите произвольный шрифт и цвет. Очистите свойство **Caption** этого компонента

- дважды щёлкните по компоненту **Button**, при этом откроется окно модуля, где система Delphi уже создала заготовку обработчика нажатия на кнопку

- впишите в тело программного кода обработчика **Label1.Caption**. Смысл этой записи в следующем – точкой в Delphi разделяют объект и его свойства. Как указать объект? По имени. Абсолютно для всех компонентов (и других объектов) это свойство **Name**. Содержится в Инспекторе и в подсказке при наведении мышки на компонент. В нашем случае объект это **Label1**, а свойство, к которому мы обращаемся, это **Caption**

- присвойте свойству **Caption** компонента **Label1** значение, используя для этого оператор присваивания **:=**. Т.о. получится строка **Label1.Caption:=**

- в качестве присваиваемого значения, по условиям поставленной задачи, необходимо вывести текст, который пользователь ввел в компонент **Edit1**. Прочсть введенный текст можно из свойства **Text** этого компонента. Т.о. окончательный вид вписываемой строки примет вид **Label1.Caption:=Edit1.Text;**

- запустите полученную программу на выполнение.

2.3. Контрольные вопросы для отчета работы

1. Опишите главные окна системы Delphi, их назначение и способ получения готовой программы.

2. Перечислите основные свойства формы.

3. Расскажите о сохранении и последующим открытии проектов и модулей в системе Delphi.

4. Опишите названия, расположение и назначение основных вкладок палитры компонентов системы Delphi

5. Расскажите о наиболее часто используемых компонентах среды Delphi, таких как **Label**, **Edit**, **Memo** и **Button**.

6. Каким образом происходит программное изменение свойства компонента?
7. Как программно получить значение свойства компонента?
8. Расскажите ход выполнения лабораторного задания.

2. ЛАБОРАТОРНАЯ РАБОТА №6

ИЗУЧЕНИЕ ПАЛИТРЫ КОМПОНЕНТОВ DELPHI.

Цель работы: закрепить знания об используемых в Delphi палитрах компонентов – *Standart*, *Additional*, *Win32*, *System*. Получить представление о компонентах этих палитр, их назначении и использовании.

Время работы: 8 часов.


2.1. Домашние задания и методические указания по их выполнению


Задание 1 – изучить страницу палитры компонентов *Standart*.



Изучение Delphi естественным образом начинается со страницы палитры компонентов *Standart*. На этой странице расположены стандартные для Windows интерфейсные элементы, применяющиеся наиболее часто. В табл. 1 представлен список компонентов страницы *Standart*.

Таблица 1

Список компонентов страницы *Standart*

Пикто- грамма	Имя компонента	Назначение
	<i>MainMenu</i>	<i>Главное меню программы.</i> Компонент способен создавать и обслуживать сложные

		иерархические меню.
	PopupMenu	Всплывающее меню. Обычно это меню появляется после нажатия правой кнопкой мыши.
	Label	Метка. Используется для размещения не очень длинных сообщений в виде статического текста
	Edit	Строка ввода. Предназначена для ввода пользователем текстовой информации в виде одной строки. Имеет возможности по управлению вводимой информацией, например, переопределению символов до их появления в поле ввода, что используется в формах для ввода пароля.
	Мето	Многострочный текстовый редактор. Используется для ввода пользователем и отображения многострочного текста без функций форматирования.
	Button	Командная кнопка. Используется для реализации в программе команд с помощью обработчика события OnClick этого компонента.
	CheckBox	Независимый переключатель. Используется его свойство Checked (отмечено), имеющее значения true или false, меняющееся при щелчке мышью.
	RadioButton	Зависимый переключатель. Используется для выбора только одного из нескольких вариантов. Для этого компонент объединяется как минимум с одним или несколькими такими же компонентами в группу. Щелчок по компоненту приводит к его выделению и снятию выделения ранее выбранного компонента. Также имеет

		свойство Checked.
	ListBox	Список выбора. Содержит список предлагаемых вариантов (опций) и даёт возможность проконтролировать текущий выбор.
	ComboBox	Выпадающий список выбора. Представляет собой комбинацию компонентов Edit и ListBox.
	ScrollBar	Полоса прокрутки. Представляет собой вертикальную или горизонтальную полосу, управляющую визуальным представлением компонентов, не помещающихся целиком в окне программы.
	GroupBox	Контейнер группы компонентов. Используется для группировки нескольких связанных по смыслу компонентов.
	RadioGroup	Группа зависимых переключателей. Содержит специальные свойства для обслуживания нескольких связанных между собой зависимых переключателей.
	Panel	Панель. Этот компонент, как и GroupBox, служит для объединения нескольких компонентов. Содержит внутреннюю и внешнюю кромки, что позволяет создавать эффекты "вдавленности" и "выпуклости".
	ActionList	Список действий. Служит для централизованной реакции программы на действия пользователя, связанные с выбором одного из группы однотипных управляющих элементов, таких как опции меню, кнопки и т.д.

Задание 2 – изучить страницу палитры компонентов
Additional.

На страницу *Additional* помещены дополнительные компоненты, без некоторых из которых сегодня трудно представить программу для Windows: кнопки с дополнительными свойствами, таблицы, компоненты для размещения изображений и многие другие, представленные в табл. 2.

Таблица 2

Список компонентов страницы *Additional*








Пикто- грамма	Имя компонента	Назначение
	<i>BitBtn</i>	<i>Командная кнопка с пиктограммой.</i> Отличается от стандартной кнопки Button возможностью отображения пиктограммы.
	<i>SpedButton</i>	<i>Пиктографическая кнопка.</i> Обычно используется для быстрого доступа к опциям Главного Меню.
	<i>StringGrid</i>	<i>Таблица строк.</i> Этот компонент обладает мощными возможностями для представления текстовой информации в табличном виде.
	<i>DrawGrid</i>	<i>Таблица изображений.</i> Этот компонент используется для представления изображений в табличном виде.
	<i>Image</i>	<i>Рисунок.</i> Компонент для отображения изображений, в том числе пиктограмм и метафайлов.
	<i>Shape</i>	<i>Фигура.</i> С помощью этого компонента можно вставить на Форму правильную фигуру - прямоугольник, эллипс, окружность.
	<i>Bevel</i>	<i>Кромка.</i> Служит для выделения отдельных частей Формы трёхмерными рамками и полосами.

	ScrollBox	Панель с полосами прокрутки. В отличие от компонента Panel автоматически вставляет полосы прокрутки, если размещённые на нём компоненты отсекаются его границами.
	CheckListBox	Список множественного выбора. Отличается от стандартного компонента ListBox наличием рядом с каждой опцией независимого переключателя типа CheckBox, облегчающего выбор сразу нескольких опций.
	Splitter	Разделитель. Этот компонент создаёт границу между двумя видимыми компонентами и даёт возможность пользователю перемещать её.
	StaticText	Статический текст. Отличается от стандартного компонента <i>Label</i> наличием собственного Windows-окна, что позволяет обводить текст рамкой или выделять его в виде "вдавленной" части Формы.
	Chart	Диаграмма. Этот компонент облегчает создание специальных панелей для графического представления данных.






Задание 3 – изучить страницу палитры компонентов *Win32*.

Эта страница содержит компоненты, представляющие собой интерфейсные элементы для 32-разрядных операционных систем Windows 95/98/NT (В версии системы Delphi 2 эта страница называлась Win95). Использующие эти компоненты программы выглядят в стилистике последних версий операционных систем Windows.

Список компонентов страницы *Win32*

Пикто- грамма	Имя компонента	Назначение
	<i>TabControl</i>	<i>Набор закладок.</i> Каждая закладка представляет собой поле с надписью и/или текстом. Выбор той или иной закладки распознаётся программой и используется для управления содержимым окна компонента.
	<i>PageControl</i>	<i>Набор панелей с закладками.</i> Каждая панель может содержать свой набор интерфейсных элементов и выбирается щелчком по связанной с ней закладке.
	<i>ImageList</i>	<i>Набор рисунков.</i> Представляет собой хранилище для нескольких рисунков одинакового размера, например, пиктограмм для кнопок.
	<i>RichEdit</i>	<i>Многострочный редактор форматированного текста.</i> В отличие от компонента Мемо может изменять такие характеристики текста как шрифт, цвет, выравнивание и т.д. (формат RTF).
	<i>TrackBar</i>	<i>Регулятор.</i> Используется для управления значениями некоторых величин в программах. Например, с его помощью удобно изменять громкость звучания мультимедийных устройств.
	<i>ProgressBar</i>	<i>Индикатор процесса.</i> С помощью этого компонента можно отображать ход исполнения достаточно длительного процесса, например, копирование данных.
	<i>UpDown</i>	<i>Цифровой регулятор.</i> Две кнопки этого

		компонента служат для увеличения (верхняя) или уменьшения (нижняя) связанной с компонентом числовой величины. Для отображения этой величины умеет ассоциироваться с компонентом <i>Edit</i> .
	HotKey	Управляющая клавиша. Компонент служит для ввода управляющих кодов, таких как F1, CTRL+Shift и т.д.
	Animate	Мультипликатор. Предназначен для отображения движущихся изображений (видеоклипов). Имеет ограничение - не может сопровождать видеоклип звуком. Как и компонент <i>ProgressBar</i> , применяется в основном для сопровождения длительных процессов.
	DateTimePicker	Селектор времени/даты. Этот компонент предназначен для ввода или отображения времени или даты.
	MonthCalendar	Календарь. Служит для отображения календаря и выбора даты или диапазона дат.
	TreeView	Дерево выбора. Представляет собой совокупность связанных в древовидную структуру пиктограмм. Обычно используется для просмотра структуры каталогов и других подобных элементов, связанных иерархическим образом.
	ListView	Панель пиктограмм. Организует просмотр нескольких пиктограмм и выбор нужной. Компонент способен располагать пиктограммы в вертикальных или горизонтальных рядах и показывать их в крупном или мелком масштабе.

	HeaderControl <i>l</i>	Управляющий заголовок. Представляет собой горизонтальную или вертикальную полосу, разделённую на ряд смежных секций с надписями. Размеры секций можно менять на этапе работы программы. Обычно используется для изменения размеров столбцов или строк в разного рода таблицах.
	StatusBar	Панель статуса. Предназначена для размещения разного рода служебной информации в окнах редактирования. Пример - нижняя часть рамки окна текстового редактора Word.
	ToolBar	Инструментальная панель. Этот компонент служит контейнером для командных кнопок BitBtn и способен автоматически изменять свои размеры и положение при добавлении или удалении кнопок.
	CoolBar	Инструментальная панель. В отличие от ToolBar , используется как контейнер для размещения стандартных интерфейсных компонентов Windows, таких как Edit , ListBox , ComboBox и т.д.
	PageScroller	Прокручиваемая панель. Служит для размещения узких инструментальных панелей. При необходимости автоматически создаёт по краям панели стрелки прокрутки.





Задание 4 – изучить страницу палитры компонентов **System**.

На этой странице представлены компоненты, которые имеют различное функциональное назначение (например, **Timer** - очень

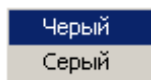
важный в любой программе компонент), в том числе компоненты, поддерживающие стандартные для Windows технологии межпрограммного обмена данными OLE и DDE.

Таблица 4

Список компонентов страницы *System*

Пикто- грамма	Имя компонента	Назначение
	<i>Timer</i>	<i>Таймер.</i> Этот компонент служит для отсчёта интервалов реального времени. Имеет ограничение - не может обрабатывать интервалы менее 55 миллисекунд.
	<i>PaintBox</i>	<i>Окно для рисования.</i> Создаёт прямоугольную область, предназначенную для прорисовки графических изображений.
	<i>MediaPlayer</i>	<i>Мультимедийный проигрыватель.</i> С помощью этого компонента можно управлять различными мультимедийными устройствами.
	<i>OleContainer</i>	<i>OLE-контейнер.</i> Служит приёмником связываемых или внедряемых объектов.
	<i>DDEClientCon</i> <i>nv</i>	<i>DDE-связь.</i> Совместно с DDEClientItem используется для создания клиентской программы в DDE-связи.
	<i>DDEClientItem</i> <i>m</i>	<i>DDE-тема.</i> Определяет тему DDE-связи в клиентском приложении.
	<i>DDEServerCon</i> <i>nv</i>	<i>DDE-связь.</i> Совместно с DDEServerItem используется для создания серверной программы в DDE-связи.
	<i>DDEServerItem</i> <i>m</i>	<i>DDE-тема.</i> Определяет тему DDE-связи в серверном приложении.

2.2. Лабораторные задания

Задание 1 – Разработать приложение «Калькулятор», демонстрирующее основные свойства изученных компонентов. Разрабатываемое приложение должно иметь форму представленную на рис. 3. При нажатии на кнопку «Выход», приложение должно завершать работу. При нажатии на кнопку « \Rightarrow » или выборе пункта меню «Расчитать», в объекте *Label1* должно выводиться сумма чисел, записанных в объектах *Edit1* и *Edit2*. При нажатии на приложение правой кнопкой мыши должно появляться всплывающее окно, в котором указаны возможные цвета, которые может принимать приложение, в следующем виде . При выборе пользователем желаемого цвета, внешний вид приложения должен соответствующим образом измениться.

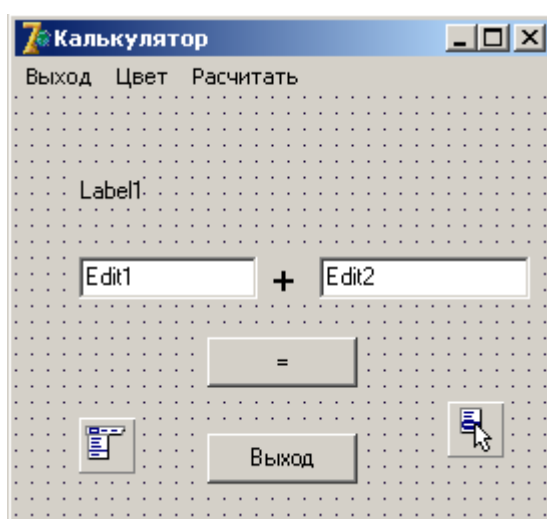


Рис. 3. Внешний вид разрабатываемой формы

При выполнении задания можно придерживаться следующего алгоритма действий:

1. Разместить, все необходимые компоненты на форме, как показано на рис. 3.
2. Изменить название формы с *Form1* на *Калькулятор*. Для этого, необходимо выделить форму и в Инспекторе Объектов на

закладке *Propertis (Свойства)* найти свойство *Caption* и изменить его с *Form1* на *Калькулятор*.

3. Перейти в Инспекторе Объектов на закладку *Events (События)*. Найти событие *OnCreat* и щелкнуть по нему 2 раза. В результате откроется редактор кода и сформируется процедура-обработчик *procedure TForm1.FormCreate(Sender: TObject);*.

4. Внести в данную процедуру изменения:

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
    Edit1.Text:='0';
```

```
    Edit2.Text:='0';
```

```
end;
```

Это необходимо для, того, чтобы при запуске программы в компонентах *Edit1* и *Edit2* были выведены 0.

5. На компоненте *BitBtn3*, который будет использоваться для выхода из приложения, изменим надпись на *Выход* (используя свойство *Caption*).

6. Кликнуть 2 раза по кнопке *Выход*, и в автоматически сгенерированной процедуре-обработчике вписать следующий код:

```
procedure TForm1.Button3Click(Sender: TObject);
```

```
begin
```

```
    Close;
```

```
end;
```

Это приведет, к тому, что по нажатию на эту кнопку будет происходить закрытие формы (выход из приложения). Точно такое же действие необходимо выполнить для процедуры-обработчика пункта *Выход* компонента *MainMenu1*

```
procedure TForm1.N3Click(Sender: TObject);
```

```
begin
```

```
    Close;
```

```
end;
```

7. На компоненте *BitBtn2*, который будет использоваться для выполнения суммирования, изменить надпись на = (используя свойство *Caption*).

8. Кликнуть 2 раза по кнопке =, и в автоматически сгенерированной процедуре-обработчике вписать следующий код

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
Var
```

```

    Znach1, Znach2: Real; //Переменные складываемых
значений
    Code: Integer; //Переменная для кода ошибки
    Sum: Real; //Результат суммирования
    s: String; //Строка
begin
    //Чтение данных с компонентов Edit1 и Edit2 и присвоение
их переменным Znach1 и Znach2
    Val(Edit1.Text,Znach1,Code);
    Val(Edit2.Text,Znach2,Code);
    Sum:=Znach1 + Znach2;
    Str(Sum,s);
    Label1.Caption:=s; //Вывод результат на форме
калькулятора с помощью компонента Label1
end;

```

Точно такое же действие необходимо выполнить для процедуры-обработчика пункта **Рассчитать** компонента **MainMenu1**.

8. Кликнуть 2 раза по компоненту **PopupMenu1**. В результате появиться окно, в котором необходимо указать названия пунктов всплывающего меню, появляющегося при нажатии на правую клавишу мышки, как представлено на рис. 4.

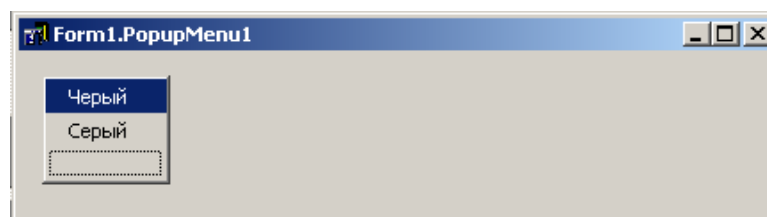


Рис. 4. Работа с всплывающим меню

9. Кликнуть 2 раза по пункту **Черный** и в автоматически сгенерированной процедуре-обработчике вписать следующий код:

```

procedure TForm1.N1Click(Sender: TObject);
begin
    Form1.Color := clBlack;
end;

```

10. Кликнуть 2 раза по пункту **Серый** и в автоматически сгенерированной процедуре-обработчике вписать следующий код:

```

procedure TForm1.N2Click(Sender: TObject);

```

begin

Form1.Color := clBtnFace;

end;

11. Для того, чтобы всплывающее меню срабатывало для формы разрабатываемого калькулятора, необходимо выбрать форму приложения, в Инспекторе Объектов на закладке *Events (События)* найти событие *PopUpMenu* и установить для него значение *PopUpMenu1*. В результате, при нажатии правой кнопки мыши по форме, будет активироваться компонент *PopUpMenu1*. всплывать окно с помощью, которого можно менять цвет формы. Такие же действия необходимо произвести и для компонента *MainMenu1*.

Задание 2 – На основании выполненного задания 1 разработать полноценный калькулятор, позволяющий производить основные арифметические операции над двумя числами.

2.3. Контрольные вопросы для отчета работы

1. Расскажите об известных Вам компонентах со страницы палитры компонентов *Standart*.

2. Расскажите об известных Вам компонентах со страницы палитры компонентов *Additional*.

3. Расскажите об известных Вам компонентах со страницы палитры компонентов *Win32*.

4. Расскажите об известных Вам компонентах со страницы палитры компонентов *System*.

5. Расскажите о компонентах используемых Вами при выполнении лабораторных заданий.

6. Как происходит программирование всплывающего (pop-up) меню?

7. Как происходит программирование главного (main menu) меню?

8. Расскажите ход выполнения лабораторных заданий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Фаронов Валерий Васильевич. Турбо Паскаль 7.0: Начальный курс: учеб. пособие / В.В. Фаронов. – 7-е изд., перераб. – М.: Нолидж, 2002. – 576 с.: ил.
2. Фаронов В. В. Турбо Паскаль: учеб. пособие / В.В. Фаронов. – СПб.: Питер, 2007. – 367 с.: ил.
3. Информатика: Базовый курс: учеб. пособие для втузов / под ред. С.В. Симоновича. СПб.: Питер, 2003. – 640 с.: ил.
4. Архангельский А.Я. Программирование в Delphi: учебник по классическим версиям Delphi / А.Я. Архангельский. – М.: Бином, 2006. – 1152 с.: ил.

СОДЕРЖАНИЕ

1. Лабораторная работа №9	1
2. Лабораторная работа №10	8
Библиографический список	21

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению лабораторных работ № 5-6 по дисциплине
«Информационные технологии» для студентов направления
11.03.03 «Конструирование и технология электронных средств»
(профиль «Проектирование и технология радиоэлектронных
средств»)
всех форм обучения

Составители:
д-р. техн. наук М.А. Ромащенко,
канд. техн. наук А.А. Пирогов,
И.В. Свиридова

Компьютерный набор М.А. Ромащенко

Подписано к изданию _____
Уч.-изд. л. _____

ФГБОУ ВО «Воронежский государственный технический
университет»
394026 Воронеж, Московский проспект, 14