

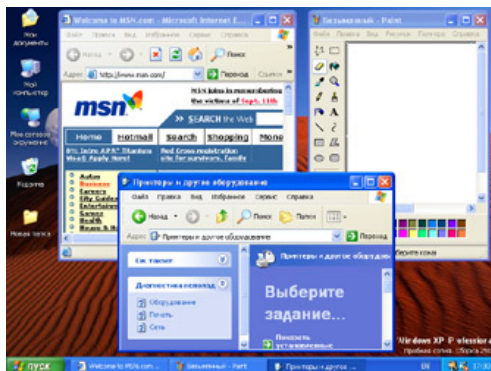
ГОУВПО
«Воронежский государственный технический университет»

Кафедра «Автоматизированные и вычислительные системы»

61 - 2008

ТЕХНОЛОГИЯ СОЗДАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО
ИНТЕРФЕЙСА

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению курсовой работы
по дисциплине «Технология программирования»
для студентов специальности 230101
«Вычислительные машины, комплексы, системы и сети»
заочной и сокращенной заочной форм обучения



Воронеж 2008

Составители: канд. техн. наук Н.И. Гребенникова,
канд. техн. наук Т.И. Сергеева

УДК 681.3

Технология создания пользовательского интерфейса: методические указания к выполнению курсовой работы по дисциплине «Технология программирования» для студентов специальности 230101 «Вычислительные машины, комплексы, системы и сети» заочной и сокращенной заочной форм обучения / ГОУВПО «Воронежский государственный технический университет»; сост. Н.И. Гребенникова, Т.И. Сергеева. Воронеж, 2008. 46 с.

Методические указания содержат перечень тем курсовых работ и практических заданий к ним, а также правила оформления элементов курсовой работы.

Предназначены для студентов второго курса.

Табл. 2. Ил. 8. Библиогр.: 6 назв.

Рецензент д-р техн. наук, проф. В.Л. Бурковский

Ответственный за выпуск зав. кафедрой д-р техн. наук, проф. С.Л. Подвальный

Печатается по решению редакционно-издательского совета Воронежского государственного технического университета

© ГОУВПО «Воронежский государственный
технический университет», 2008

Введение

Данные методические указания содержат теоретические сведения по разработке пользовательского интерфейса в среде программирования Delphi, задания для выполнения курсовых работ по дисциплине «Технология программирования» и рекомендации по оформлению курсовой работы в соответствии с требованиями СТП ВГТУ 62-2007.

1 Методические рекомендации по выполнению курсовой работы

1.1 Требования к содержанию пояснительной записки

Каждая курсовая работа имеет одинаковую структуру. Расчетно-пояснительная записка должна содержать: титульный лист (приложение А), задание на курсовую работу (приложение Б), лист для замечаний руководителя (приложение В), содержание (примерный вариант содержания представлен в приложении Г), введение, теоретическая часть, практическая часть, заключение, список литературы (пример оформления литературы представлен в приложении Д).

Титульный лист оформляется в соответствии с требованиями стандартов ГОСТ и требованиями ВГТУ по образцу приложения А.

Задание на курсовую работу представлено на втором листе пояснительной записки и оформлено по образцу приложения Б. Задание помимо единых для всех заголовочных строк содержит:

- перечень вопросов, подлежащих разработке;
- объем работ;
- сроки выполнения этапов;
- срок защиты курсовой работы;
- подпись руководителя и дату выдачи задания;
- подпись студента и дату получения задания.

В перечне вопросов, подлежащих разработке, требуется указать поставленные в задании цели курсовой работы, например, освоить вопросы, касающиеся технологии проектирования и разработки программной системы, освоить технологии тестирования программ, освоить технологию разработки пользовательского интерфейса в среде Delphi, реализовать практическое задание с использованием выбранной технологии проектирования и разработки программной системы, оформить расчетно-пояснительную записку.

Объем работ описывает необходимые действия по реализации поставленной цели. Сюда можно включить поиск и обзор литературных источников, изучение необходимых технологий проектирования, разработки, отладки и тестирования программных систем, реализацию практических заданий, оформление пояснительной записки.

В соответствии с указанным выше объемом работ все перечисленные работы распределяются по этапам и для каждого из них устанавливаются сроки их выполнения.

Лист замечаний руководителя остается пустым. При проверке пояснительной записки руководитель вписывает свои замечания по выполненной работе.

Содержание пояснительной записки имеет обычный смысл и включает наименование ее разделов и подразделов и соответствующие номера страниц.

Введение должно содержать цель курсовой работы, перечень решаемых задач для достижения поставленной цели. Во введении должно быть кратко отражено содержание основных разделов курсовой работы. Объем введения 1-4 страницы.

Теоретическая часть содержит обзор литературы и предлагаемых технологий решения по вопросам, указанным в теоретическом задании. Объем теоретической части 10-15 страниц.

В теоретической части могут быть изложены следующие вопросы:

общие вопросы организации пользовательского интерфейса; типовые элементы интерфейса;

средства разработки пользовательского интерфейса;
создание таблиц базы данных и организация ввода информации с помощью форм;

обработка данных (реализация сортировки записей в таблице, фильтрация записей таблицы, реализация вычислений).

Практическая часть описывает выполнение практического задания. Содержит описание технологии решения применительно к предложенному практическому заданию. Необходимо использовать видеogramмы (скриншоты) для пояснения произведенных вычислений. Объем практической части занимает 10-15 страниц.

В практической части могут быть изложены следующие вопросы:

структура таблиц из выбранного варианта;
создание и вид форм с введенными данными;
созданная форма для сортировки и процедуры реализации сортировки из задания;

форма для фильтрации, процедуры реализации фильтрации;

реализация вычислений из выбранного варианта.

В заключении кратко описываются решаемая задача, освоенные технологии вычислений или обработки данных. Оцениваются изученные технологии с точки зрения удобства и эффективности использования для решаемых задач. Объем заключения – 1-2 страницы.

Список использованной литературы включает литературные источники, ссылки на которые должны быть сделаны по ходу выполнения работы. При использовании в пояснительной записке по курсовой работе цитат из литературы, периодических изданий или официальных нормативных документов обязательно следует делать ссылки на источники, список которых составляется в порядке их упоминания в тексте. Ссылки записываются в квадратных скобках в тексте записки внутри предложения.

Список используемых литературных источников должен содержать не менее пяти наименований, существующих на

бумажных носителях, и не менее двух адресов порталов Internet. В список литературы включаются только источники, использованные в работе и содержащие существенную информацию по разрабатываемой теме. Объем списка литературы – 1 страница.

1.2 Требования к оформлению пояснительной записки

Общий объем расчетно-пояснительной записки по курсовой работе составляет 20-30 листов формата А4, выполненных в редакторе Word for Windows одной из последних версий на момент формирования расчетно-пояснительной записки.

К оформлению записки предъявляются следующие требования:

шрифт Times New Roman Cyr 14(или 12 пт.);

полуторный (или одинарный) интервал;

страницы пронумерованы, номера располагаются внизу страницы по центру;

абзацы выровнены по ширине, первая строка абзаца имеет отступ 1.25 см;

названия разделов, введение, заключение, список литературы выводятся прописными буквами и выровнены по центру; названия подразделов выводятся с первой прописной буквы, остальные строчные и оформляются как абзацы;

подписи к рисункам выводятся по центру под рисунком в виде «Рисунок 1.1 – Схема базы данных»;

название таблицы выводится над таблицей без абзацного отступа в виде «Таблица 1.2 – Структура таблицы ПРИБОРЫ».

2 Разработка элементов приложения в Delphi

2.1 Структура приложения в Delphi

Созданное в Delphi приложение – это проект, состоящий из элементов:

код проекта (.dpr),

описания форм (.dfm),
модули форм (.pas),
модули (.pas),
параметры проекта (.dof),
описание ресурсов (.res).

Целесообразно для каждого нового файла проекта создавать свою папку, чтобы не путаться в многочисленных файлах, относящихся к конкретному проекту.

Для каждой формы создается 2 файла:
описания формы .dfm,
сам модуль .pas.

Сразу после создания нового приложения Delphi создает пустую форму (Form 1). Эта форма содержит: заголовок (Form 1); кнопки минимизации-максимизации, закрытия окна; элементы изменения окна; кнопку вызова системного меню этого окна, т.е. те элементы, без которых невозможна работа с любым окном в Windows. Для этой формы Delphi автоматически создает 2 файла с описанием и модулем.

2.2 Свойства формы

Caption – заголовок.

Color – цвет.

Cursor определяет графический вид курсора. Вид курсора можно выбрать из списка.

FormStyle (тип TFormStyle) – стиль формы

fsNormal – стандартный стиль;

fsMDIForm – родительская форма в многодокументном приложении;

fsMDIFChild – дочерняя

fsStayOnTop – форма изображается поверх других окон. Обычно используется для заставки.

BorderStyle – (при TFormBorderStyle) – вид и поведение ограничивающей рамки.

bsDialog – диалоговая форма,

bsSingle – форма с неизменяемыми размерами,

bsNone – форма без рамки и заголовка, не изменяет размеры.

(по умолчанию) bsSizeable – обычная рамка, форма с неизменяемыми размерами, со строкой заголовка.

bsToolWindow – форма панели инструментов,

bsSizeToolWindow – форма панели инструментов с неизменяемыми размерами.

Свойство **BorderIcons** (тип TBorderIcons) определяет возможность изображения 4 – х кнопок.

biSystemMenu – кнопки системного меню,

biMinimize – кнопки свертывания,

biMaximize – кнопки восстановления,

biHelp – кнопки справки для вызова контекстно-ориентированной справки.

Если присутствует кнопка системного меню, то ни одна кнопка не отображается.

Если есть кнопки Min/Max, то кнопка Help не отображается. Кнопки Min/Max возможны только для обычной формы панели инструментов с изменяющимися размерами.

Свойства **Hint** и **ShowHint** позволяют отобразить подсказку. В свойстве Hint набирают подсказку, при этом, чтобы подсказка выводилась при остановке мыши, необходимо свойству ShowHint присвоить значение True.

Свойства **Width, Height** – размеры формы в пикселах.

ClientWidth, ClientHeight (тип Integer) задают ширину и высоту клиентской части формы.

(В клиентской части отображаются компоненты формы, в неклиентской – рамка, заголовок, строка главного меню).

При необходимости могут появляться полосы прокрутки.

Свойство **Icon** (тип TIcon) определяют пиктограмму в левой области заголовка формы.

Если пиктограмма не задана, то используется пиктограмма из свойства Icon объекта Application.

Position (типа TPosition) задаёт размещение и размер формы:

poDesigned – определены при конструировании формы значениями Left, Top, Width, Height,
poScreenCenter – расположение в центре экрана,
poDefault – Windows автоматически сам определяет начальную позицию и размеры формы,
poDefaultPosOnly – Windows автоматически определяет только позицию, а не размеры,
poDefaultSizeOnly – Windows автоматически определяет только размеры.

Свойство **Active** (тип Boolean) задаёт активность формы (обычно визуально отмечается синим заголовком), т.е. фокус ввода.

Свойство **WindowState** (тип TWindowState) определяет состояние формы:

wsNormal – обычное (по умолчанию),

wsMinimized – минимизация,

wsMaximized – максимизация.

Свойство **ActivControl** (типа TWinControl) определяет, какой элемент формы находится в фокусе ввода.

Например: Form1.ActiveControl:=Edit2;

Эту же операцию проводит метод SetFocus.

Edit2.SetFocus;

Свойство **AutoScroll** (типа Boolean) определяет автоматическое появление полосы прокрутки.

Процедура ScrollInView (AControl:TControl) автоматически изменяет полосы прокрутки так, чтобы заданный параметром AControl элемент был виден в отображаемой области.

Свойство **KeyPreview** (тип Boolean) определяет, будет ли форма обрабатывать события клавиатуры, прежде чем их обработают элементы управления формы.

2.3 Элементарные компоненты интерфейса

Компонент **LABEL** используют для отображения надписи.

Можно настроить следующие свойства компонента **Label**:

- с помощью значения **false** свойства **AutoSize** можно запретить автоматическую коррекцию размера надписи по размеру текста надписи;

- с помощью значения **taCenter** свойства **Alignment** можно выровнять надпись по центру формы;

- с помощью значения **true** свойства **WordWrap** можно установить автоматический перенос слов надписи на другую строку, если весь текст не помещается в отведенный размер;

- с помощью значения **true (false)** свойства **Transparent** можно сделать надпись прозрачной (непрозрачной);

- с помощью свойства **Caption** можно задать текст надписи (например, Фамилия);

- с помощью свойства **Font** изменить цвет и размер шрифта.

Компонент **EDIT** используют для создания строки ввода и редактирования данных. Компонент позволяет водить строку, использовать клавиши управления курсором, **BackSpace**, **Delete**, выделять фрагмент текста. Введенные символы сохраняются в свойстве **Text**.

Компонент обычно помещают на форме справа от созданной надписи.

Инициализировать его значения пробелами можно, убрав содержимое свойства **Text**.

Компонент **BUTTON** позволяет создать кнопку, с которой можно связать различные действия. Для создания кнопки на вкладке стандартных компонентов выбирают компонент **Button**. Поместив его на форму, меняют название кнопки с помощью свойства **Caption**.

2.4 Создание базы данных и таблиц

2.4.1 Создание псевдонима базы данных

При работе с таблицами локальных БД (в число которых входят таблицы СУБД **Paradox** и **dBase**) сама база данных раз-

мещается в каталоге на диске и хранится в виде набора файлов. Для хранения одной таблицы создается отдельный файл. Такие же отдельные файлы создаются для хранения индексов таблицы и мемо-полей.

Предположим, что на диске (С: или Х:) в папке с фамилией разработчика (на английском языке) будет создана папка Tables, где будут размещаться таблицы базы данных.

Обращение к БД из утилит и программы осуществляется по *псевдониму* базы данных. Псевдоним должен быть зарегистрирован в файле конфигурации конкретного компьютера при помощи утилиты BDE Administrator. Необходимо присвоить какой-то псевдоним (например, BIBL) создаваемой БД. Для этого поступают следующим образом.

- Запустить утилиту BDE Administrator (Пуск, Программы, Borland Delphi 7, BDE Administrator).
- Выбрать в главном меню окна утилиты пункты Object, New (или выбрать вкладку Databases, вызвать на дереве Database контекстное меню, выбрать пункт New, щелкнуть по кнопке ОК).
- В появившемся окне оставить тип создаваемой БД без изменений (STANDARD) и щелкнуть по кнопке ОК.
- В левом поле окна администратора заменить имя Standard1 на имя псевдонима (например, BIBL), в строке PATH щелкнуть по кнопке ... и выбрать созданную ранее папку (например, TABLES).
- Для запоминания псевдонима в левом окне администратора БД на имени псевдонима щелчком правой кнопки мыши вызвать контекстное меню и выбрать пункт Apply (применить), в появившемся диалоговом окне щелкнуть по кнопке ОК (рисунок 2.1).
- Закрыть окно утилиты BDE Administrator: создание псевдонима завершено и к нему можно обращаться из других утилит и приложений. Однако каталог, на который ссылается псевдоним БД, еще пуст. Необходимо создать в нем таблицы БД.

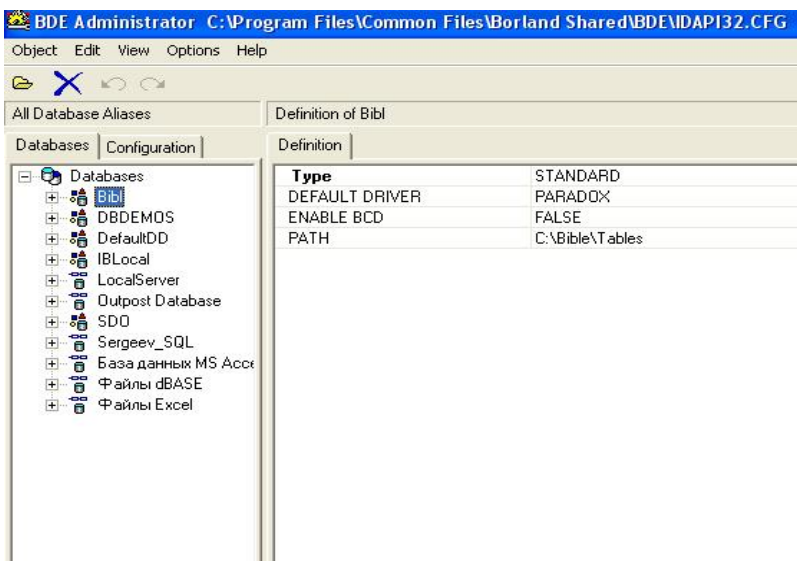


Рисунок 2.1 - Окно BDE Администратора

2.4.2 Установка рабочей директории и создание таблиц

Для установки рабочей директории необходимо запустить утилиту Database Desktop (DBD) (Пуск, Программы, Borland Delphi 7, Database Desktop). Затем выбирают пункты меню File, Working Directory, с помощью кнопки BROWSE выбирают каталог для размещения файлов базы данных (например, созданную папку TABLES) и в выпадающем списке Aliases выбирают имя псевдонима. После этого щелкают по кнопке ОК.

Создание таблиц базы данных.

Создание таблицы. Таблицы Paradox 7 создаются с помощью утилиты Database Desktop. Для создания таблицы запускают утилиту Database Desktop, выбирают пункты меню **File, New, Table**. В появившемся окне **Create Table** оставляют без изменений тип создаваемой таблицы (Paradox 7) и нажи-

мают кнопку **ОК**. После этого появится окно определения структуры таблицы БД (рисунок 2.2).

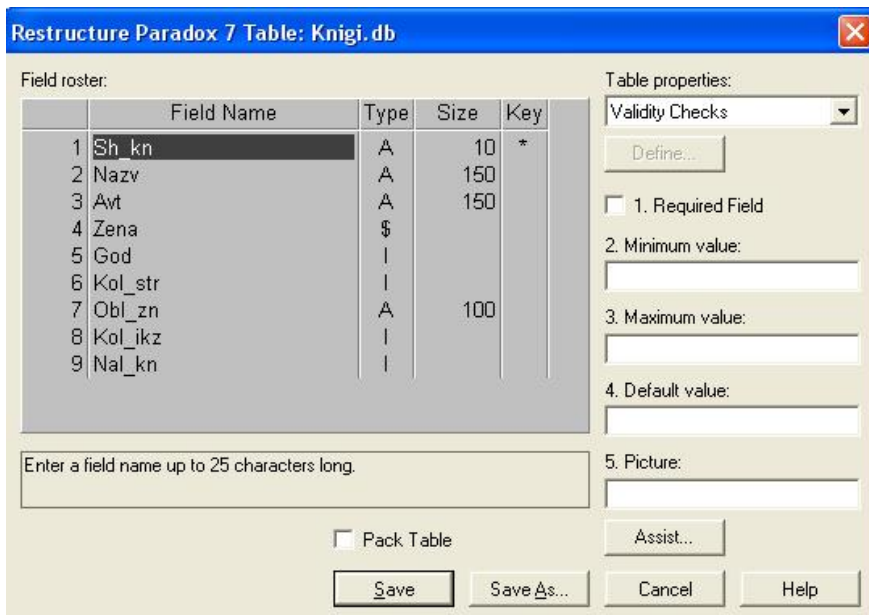


Рисунок 2.2 - Структура таблицы Knigi

Каждая строка таблицы описывает одно поле. Назначения столбцов:

- Field Name – имя поля (английские буквы);
- Type – тип поля (например, A – символьное длиной до 255 символов, N – число с плавающей точкой до 15 значащих цифр, S – денежный формат, I – целочисленные значения, D – дата, + - автоинкрементное поле и т.д.);
- Size – размер поля (для строковых полей);
- Key – содержит звездочку «*», если поле входит в состав первичного ключа (выставляется клавишей пробел).

Типы полей. В таблицах Paradox могут использоваться типы данных, представленные в таблице 1.1.

Таблица 1.1 - Типы данных

Type	Size	Тип	Описание
A	1-255	Alpha	Текстовое поле указанной длины
N		Number	Числа с плавающей запятой в диапазоне от -10^{307} до $+10^{307}$ с 15 значащими десятичными разрядами
\$		Money	Денежное поле. Содержит вещественные числа с фиксированной запятой, 6 знаками целой части и 2 знаками дробной части числа
S		Short	Целые числа в диапазоне от -32768 до $+32767$
I		Long Integer	Целые числа в диапазоне от $-2\,147\,483\,648$ до $+2\,147\,483\,647$
#	0-32	BCD	Двоично-десятичные вещественные числа. Size – количество разрядов после запятой
D	Date		Дата в диапазоне от 1.01.0000 до 31.12.9999
O	0-240	OLE	Объект OLE
L		Logical	Логическое поле. Содержит значение True или False

Продолжение таблицы 1.1

+		Autoincrement	Автоинкрементное поле
Y	1-255	Bytes	Набор из Size байтов

2.4.3 Выбор языкового драйвера таблицы

Для отображения русского текста без искажения при вводе информации в символьные поля необходимо установить драйвер для русского языка.

- В диалоговом окне описания таблицы в списке **Table properties** необходимо выбрать **Table Language**.
- Щелкнуть по кнопке **Modify**.
- В появившемся окне в списке **Language** выбрать **Pdox ANSI Cyrillic**.

Процедуру установки нужного языкового драйвера следует проводить до создания индексов по символьным полям и установки связей между таблицами.

2.4.4 Сохранение таблицы

Для запоминания таблицы в окне описания таблицы щелкают по кнопке **Save As**. В появившемся окне задают имя таблицы и выбирают созданную вами папку для размещения таблиц.

2.5 Создание меню приложения

Создание меню приложения осуществляется следующим образом.

1 Запустить Delphi (Пуск, Программы, Borland Delphi, Delphi). Автоматически откроется форма 1.

2 На панели компонентов перейти во вкладку **Standard**, выбрать компоненту **Mainmenu** и поместить ее на форму.

3 На компоненте вызывают контекстное меню и выбирают пункт **Menu Designer....** Открывается диалоговое окно, в котором формируют элементы меню. Для текущего элемента меню необходимо задать подпись.

4 На панели Object (Инспектора объектов) в поле Caption набирают название текущего пункта меню (например, Редактирование) и нажимают клавишу Enter.

5 Выделяют новый пункт меню (пустой), появившийся справа, и повторяют действия п. 4.

6 Для создания меню второго уровня выделяют щелчком пункт меню первого уровня и нажимают на клавиатуре стрелку вниз. В поле **Caption** вносят название пункта и нажимают клавишу Enter. Эти действия повторяют для каждого пункта меню второго уровня (рисунок 2.3).



Рисунок 2.3 - Форма, содержащая меню

7 После создания меню закрывают окно Menu Designer и сохраняют форму. Для сохранения выбирают пункты меню File, Save All, выбирают папку, где должны сохраняться модули.

Привязка пункта меню к форме. Пункты меню должны быть привязаны к вызову определенных форм.

- 1 Привязка пункта «Выход» к соответствующему действию:
 - в режиме редактирования осуществить двойной щелчок по пункту «Выход»;
 - в открывшейся процедуре набрать `Form1.Close`;
 - сохранить произведенные действия.
- 2 Привязка пункта меню к вызову соответствующей формы:
 - создать новую форму (кнопка `NewForm`, 4-я слева на панели инструментов);
 - вызвать список модулей кнопкой `ViewUnit` (1-я слева) и выбрать в списке `unit1`, в тексте процедуры найти слово `implementation`, для доступа к модулю `unit2` ниже этого слова набрать `uses unit2`;
 - переключиться на главную форму;
 - дважды щелкнуть по привязываемому пункту меню и в открывшейся процедуре ввести: имя формы. `ShowModal`; (например: `Form2.ShowModal`);
 - Проверить вызов формы при выборе пункта меню (кнопка `Запуск` на панели инструментов или `F9`).
 - Выход из режима запуска – закрыть окно первой формы.

Редактирование внешнего вида формы. Можно осуществить следующие действия по редактированию формы:

- в строке **Caption** задают подпись на русском языке;
- в строке **Bordericons** для **biMaximize** выбирают `False` – это означает запрет на распаивание окна;
- в строке **Position** задают способ выравнивания формы на экране (например, по центру – `poDesktopCenter`);
- в строке **Color** выбирают цвет заливки формы;
- вставка картинки на форму – вкладка **Additional**, кнопка **Image**, свойство **Picture**, кнопка **Load**, осуществляют выбор файла с картинкой. Для вставленной картинки в панели свойств можно установить в строке **stretch** значение **True** (установка размера картинки по размеру формы); в строке

transparent значение **True** (установка прозрачного фона для картинки).

2.6 Создание формы для работы с одной таблицей

Создать форму для работы с одной таблицей можно следующим образом.

- Выбрать или создать новую форму (выбрать - кнопка **View Form** (2-я кнопка слева), создать - кнопка **New Form** на панели инструментов, 4-я слева) для размещения на ней формы.
- Выбрать вкладку **BDE**, щелкнуть по кнопке **Table** (таблица) и в форме. После этого изображение кнопки останется в форме.
- В окне Инспектора Объектов устанавливаются для свойства **DatabaseName** псевдоним БД (например, **VIBLDATE**), выбрав его из списка или введя вручную. Для свойства **TableName** (имя таблицы БД) выбирают имя таблицы, для которой создают форму. Для свойства **Active** устанавливается значение **True**. В этот момент произойдет реальное связывание компонента **Table** (он по умолчанию имеет имя **Table1**) с выбранной таблицей.
- Выбрать вкладку **Data Access**. Щелкнуть по кнопке **DataSource** (источники данных) и в форме. Данный компонент служит в качестве связующего звена между не визуальными компонентами (в данном случае **Table1**) и визуальными компонентами, которые будут добавлены позже. Для данного компонента в свойстве **DataSet** (имя набора данных) установить значение **Table1**, выбрав его из списка.
- Выбрать вкладку **Data Controls**. Щелкнуть по кнопке **DBGrid** и в форме. Данный компонент служит для отображения записей набора данных в табличной форме. Можно изменить размер формы, растащив границы. Для данного компонента в свойстве **DataSource** уста-

навливают значение **DataSource1** (это имя присвоено по умолчанию), выбрав его из списка.

- Сохранить форму и проект на диске в созданной вами папке APP (**File, Save All**).
- Запустить программу (кнопка **RUN** на панели инструментов, или **F9**, или пункты меню **RUN, RUN**). Работающая программа открывает непосредственный доступ к данным в выбранной таблице.

Для добавления записи нужно нажать на клавиатуре клавишу **Insert** или, находясь на последней записи набора данных, нажать стрелку вниз. Таблица автоматически перейдет в режим добавления новой записи.

После ввода значений в поля записи запомнить запись в наборе данных можно, перейдя на другую запись при помощи клавиш управления курсором.

Отказаться от запоминания записи можно, нажав кнопку **Esc**.

Для изменения записи следует переместить указатель текущей записи в нужное место и изменить значения там, где это нужно. Набор данных автоматически перейдет в режим редактирования.

Для удаления записи следует установить на нее указатель текущей записи и нажать **Ctrl+Del**.

После ввода и редактирования данных вернуться в режим редактирования формы, закрыв окно просмотра формы.

2.7 Редактирование формы

Удаление из формы автоинкрементного столбца. Этот столбец необходим для обеспечения уникальности записей и не несет никакой смысловой нагрузки. Его можно убрать из формы. Для этого поступают следующим образом.

- Открыть окно редактора полей. Для этого дважды щелкнуть по компоненту **Table2** в окне формы (или щелкнуть по нему правой кнопкой мыши, выбрать

Fields Editor). На экране появится пустое окно редактора полей.

- Щелкнуть по окну редактора полей правой кнопкой мыши и выбрать пункт меню **Add Fields**. Будет показан список всех полей таблицы. Выделить (при помощи мыши и клавиши Shift) все поля, кроме автоинкрементного и нажать кнопку **ОК**. Теперь список редактора полей будет включать все отмеченные поля.

Русификация столбцов в компоненте DBGrid. Выделить компонент **DBGrid**. Вызвать на нем контекстное меню и выбрать пункт **Columns Editor**. На экране появится окно редактора столбцов. Для отображения в данном окне списка полей щелкают по кнопке **Add All Fields**. Выделить в окне нужное поле. В окне Инспектора Объектов можно изменить следующие свойства.

Раскрыть список **Title** (щелчок по плюсу).

- В поле **Caption** задать имя на русском языке.
- В поле **Color** выбрать из списка цвет фона.
- В поле **Alignment** выбрать способ выравнивания (например, taCenter – по центру).

Раскрыть список **Font** (щелчок по плюсу).

- В поле **Size** выбрать размер шрифта.
- В поле **Color** выбрать цвет символов.

Создание надписи. Перейти во вкладку **Standard**. Щелкнуть по кнопке **Label** и в форме, где будет надпись. В окне Инспектора Объектов задают:

в строке **Caption** русское название надписи;

в строке **Aligment** способ выравнивания (например, taCenter – по центру);

в строке **Color** цвет фона;

в раскрывающемся списке **Font** и в поле **Color** выбирают цвет символов, в поле **Size** вводят размер;

в списке **Style** для нужного начертания устанавливают значение **True**;

в строке **Transparent** устанавливают **True** для прозрачного фона надписи.

Создание панели кнопок навигации. Перейти во вкладку **DataControls**, щелкнуть по кнопке **DBNavigator** и в форме там, где будет панель с кнопками (например, под формой для каждой таблицы). Выделить компонент и в окне Инспектора Объектов в поле **DataSource** выбрать значение **DataSource1**.

Создание кнопки выхода из формы. Перейти во вкладку **Standard**. Щелкнуть по кнопке **Button** и в форме, где будет кнопка выхода. В окне Инспектора объектов в поле **Caption** набрать слово **Выход**. Перейти во вкладку **Events**. Дважды щелкнуть по строке **onClick**, при этом происходит переход в процедуру обработки событий, где необходимо набрать оператор:

```
Form1.Close;
```

Проверить работу формы (F9). Осуществить возврат в редактор (Ctrl+F2).

Пример формы для работы с одной таблицей представлен на рисунке 2.4.

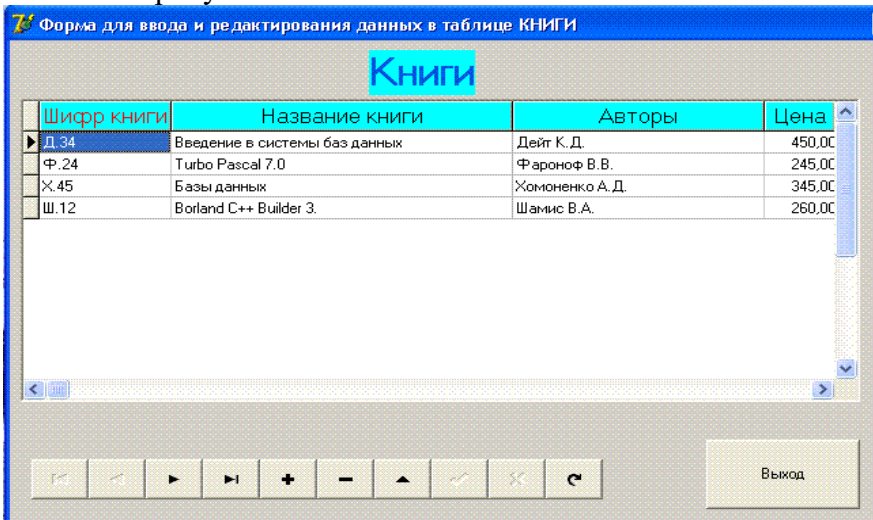


Рисунок 2.4 - Форма в режиме редактирования

3 Основные методы работы с данными

3.1 Сортировка

Для осуществления сортировки **необходимо создать вторичные индексы** (первичный индекс создается автоматически по ключевым полям).

Создание вторичных индексов осуществляется следующим образом:

запустить утилиту Database Desktop (Пуск, Программы, Delphi 7, Database Desktop);

открыть таблицу, для которой создаются индексы (File, Open, Table, выбрать имя таблицы, перейти в режим редактирования структуры – Table, Restructure);

в окне свойств выбрать Secondary Indexes, кнопка Define, выбрать поле, по которому создается индекс; кнопка Add; кнопка ОК, ввести имя индекса (например, Ind_Avt).

Для **отображения результатов сортировки** на форму необходимо поставить следующие компоненты (рисунок 3.1):

компонент TTable (Таблица); настроить свойства – имя базы данных (DatabaseName), имя таблицы (TableName), активизировать таблицу (Active – true);

компонент DataSource (источник данных); настроить свойство DataSet – Table1;

компонент DbGrid (таблица для отображения информации); настроить свойство DataSource – DataSource1.

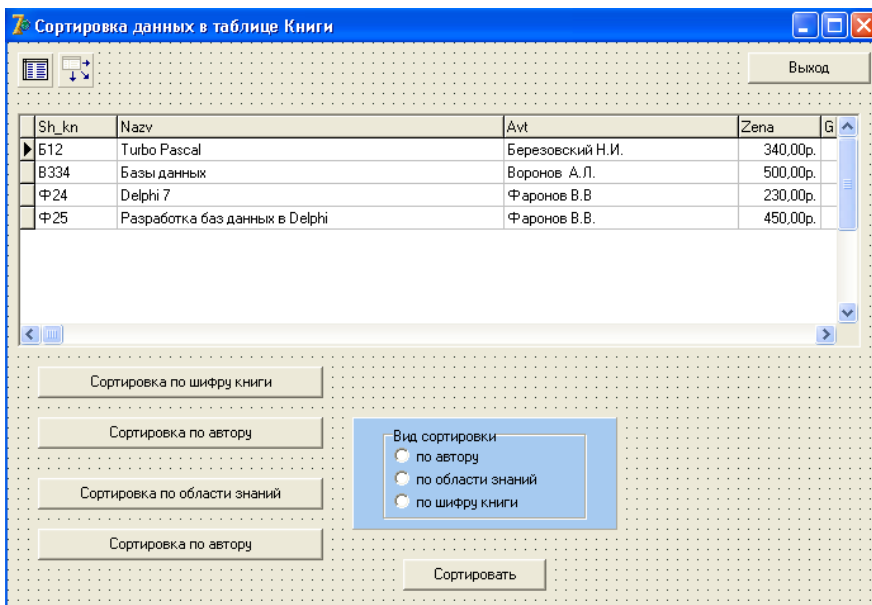


Рисунок 3.1 - Форма для реализации сортировки

Сортировка с помощью отдельной кнопки осуществляется следующим образом.

1 вариант – с использованием имени индекса:

Поставить на форму компонент Кнопка (**Button1**), для свойства **OnClick** данной кнопки написать процедуру

Table1.IndexName:='Ind_shgr';

Данная процедура открывает индекс по имени **Ind_shgr** (данный индекс создан для поля **Sh_gr** – шифр книги, имя индекса – **Ind_shgr**). Открытие индекса автоматически сортирует записи по индексному полю.

2 вариант - с использованием имени поля, по которому создан индекс:

Поставить на форму компонент Кнопка (**Button2**), для свойства **OnClick** данной кнопки написать процедуру

Table1.IndexFieldNames:='Avt';

Данная процедура открывает индекс, созданный по полю Avt (авторы).

Сортировка с помощью группы переключателей осуществляется следующим образом.

1 На форму можно поставить компонент **Panel1**, удалить надпись в свойстве **Caption**.

2 На панель поставить компонент **Radiogroup1**, для него в свойстве **Caption** задать надпись **Вид сортировки**; в свойстве **Items** задать подписи для переключателей (например, по автору, по области знаний, по шифру книги).

3 Добавить на форму кнопку **Button1** (задать надпись Сортировать), для свойства **OnClick** данной кнопки написать процедуру (**ItemIndex** – номер выбранного переключателя)

```
case RadioGroup1.ItemIndex of
0: Table1.IndexName:='Ind_Avt';
1: Table1.IndexName:='Ind_Oblzn';
2: Table1.IndexName:='Ind_shgr';
end;
```

4 При выборе первого переключателя будет сортировка по полю Авторы, при выборе второго переключателя будет сортировка по полю Область знаний, при выборе третьего переключателя – по полю Шифр книги.

3.2 Вычисления

Для осуществления вычислений по данным из таблицы необходимо поставить компонент Table на форму и связать его с таблицей базы данных.

Для контроля производимых вычислений можно поставить на форму компоненты DataSource и DBGrid и настроить их свойства.

Вычисления можно оформить как процедуры, которые запускаются при щелчке по кнопке (компонент Button).

Вывод результата вычислений можно осуществить через компонент Label.

Возможный вид формы представлен ниже (рисунок 3.2).

Шифр книги	Название	Авторы	Цена	Год издания	Кольво стр.	блясть знанч	Кольво экз.	Наличие книг
Б12	Turbo Pascal	Березовский Н.И.	340,00р.	2004	567	Программир	100	100
С67	Базы данных	Сергеева Т.И.	120,00р.	2003	180	Базы данные	50	50
Ф24	Delphi 7	Фаронов В.В.	230,00р.	2001	765	Программир	50	50
Ф25	Разработка баз дан	Фаронов В.В.	450,00р.	2003	876	Базы данные	20	50
Ф44	Разработка приложе	Фалашин Е.К.	340,00р.	2007	400	Базы данные	50	50

Вычислить суммарную стоимость книг
Суммарная стоимость=77500

Максимальная цена
max=450

Минимальное количество книг
min=20

Рисунок 3.2 - Форма для реализации вычислений

Вычисление суммарной стоимости книг осуществляется при щелчке по кнопке Вычислить суммарную стоимость книг (**Button1**). Результат выводится в компонент **Label1**. Процедура вычисления следующая:

```
procedure TForm7.Button1Click(Sender: TObject);
var
  sumk: real;
begin
  sumk:=0;
  Table1.First;
  While not Table1.eof do
  begin
```

```

sumk:=sumk+Table1['Zena']*Table1['Kol_ikz'];
Table1.Next;
end;
Label1.Caption:='Суммарная
стоимость='+FloatToStr(sumk);
end;

```

Вычисление максимальной стоимости книг осуществляется при щелчке по кнопке Максимальная цена (**Button2**). Результат выводится в компонент **Label2**. Процедура вычисления следующая:

```

procedure TForm7.Button2Click(Sender: TObject);
var
max: real;
begin
Table1.First;
max:=Table1['Zena'];
While not Table1.eof do
begin
if max < Table1['Zena'] then max:=Table1['Zena'];
Table1.Next;
end;
Label2.Caption:='max='+FloatToStr(max);
end;

```

Вычисление минимального количества книг осуществляется при щелчке по кнопке Минимальное количество книг (**Button3**). Результат выводится в компонент **Label3**. Процедура вычисления следующая:

```

procedure TForm7.Button3Click(Sender: TObject);
var
min: integer;
begin
Table1.First;
min:=Table1['Kol_ikz'];
While not Table1.eof do

```

```

begin
if min > Table1['Kol_ikz'] then min:=Table1['Kol_ikz'];
Table1.Next;
end;
Label3.Caption:='min='+IntToStr(min);
end;

```

3.3 Фильтрация

Фильтрация по заданному выражению, числовому полю, текстовому полю осуществляется следующим образом.

1 Необходимо создать форму следующего вида (рисунок 3.3).

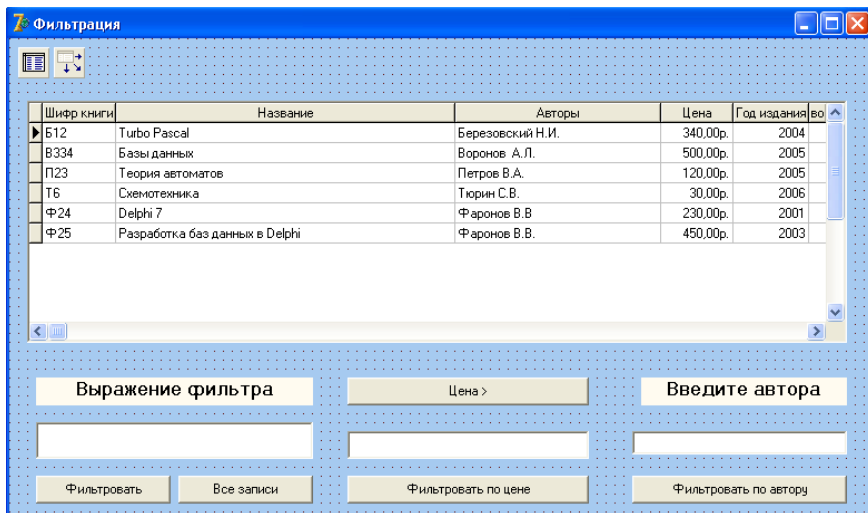


Рисунок 3.3 - Вид формы для реализации фильтрации

На форму помещены компоненты:

Table1, DataSource1, DBGrid1;

надпись Label1 (в свойстве Caption написать **Выражение фильтра**);

компонент Edit1 (свойство Text стереть) используют для ввода выражения фильтрации;
компонент кнопка Button1 (Caption – Фильтровать);
компонент кнопка Button2 (Caption - Все записи);
компонент кнопка Button3 (Caption – Цена >);
компонент Edit2 (свойство Text стереть) используют для ввода цены;
компонент Button4 (Caption – Фильтровать по цене);
компонент Label2 (в свойстве Caption написать **Введите автора**);
компонент Edit3 (свойство Text стереть);
компонент Button5 (Caption – Фильтровать по автору).

2 С данной формой необходимо связать следующие обработчики событий.

Процедура для кнопки «Все записи».

```
procedure TForm8.Button2Click(Sender: TObject);
begin
Table1.Filtered:=False; (*отключение фильтрации*)
Edit1.Text:=''; (*очистка поля от введенного фильтра*)
Edit3.Text:="";
end;
```

Процедура для кнопки «Фильтровать».

```
procedure TForm8.Button1Click(Sender: TObject);
begin
Table1.Filtered:=True; (*включение фильтрации*)
Table1.Filter:=Edit1.Text; (*определение условия
фильтрации*)
end;
```

Примеры выражений фильтра:

Obl_zn='Базы данных'

Zena > 300

(Zena > 300) and (God > 2003)

Avt = 'Ф*'

Zena = 500 or Zena=450

Процедура для формы в целом.

```
procedure TForm8.FormCreate(Sender: TObject);
```

```
begin
```

```
Table1.FilterOptions:=[foCaseInsensitive]; (*регистр букв не  
учитывается*)
```

```
Table1.Filtered:=False; end;
```

Фильтрация по числовому значению осуществляется с помощью процедуры следующего вида:

```
procedure TForm8.Button4Click(Sender: TObject);
```

```
begin
```

```
Table1.Filtered:=True;
```

```
Table1.Filter:='Zena>'+Edit2.Text;
```

```
end;
```

Фильтрация по тексту выполняется следующим образом.

```
procedure TForm8.Button5Click(Sender: TObject);
```

```
var a: string[20];
```

```
begin
```

```
Table1.Filtered:=True;
```

```
a:='Avt=';
```

```
Table1.Filter:=a+'"+Edit3.Text+'"; (*чтобы поставить ‘  
надо поставить две ‘*)
```

```
end;
```

Фильтрация по дате. Создадим форму следующего вида (рисунок 3.4).

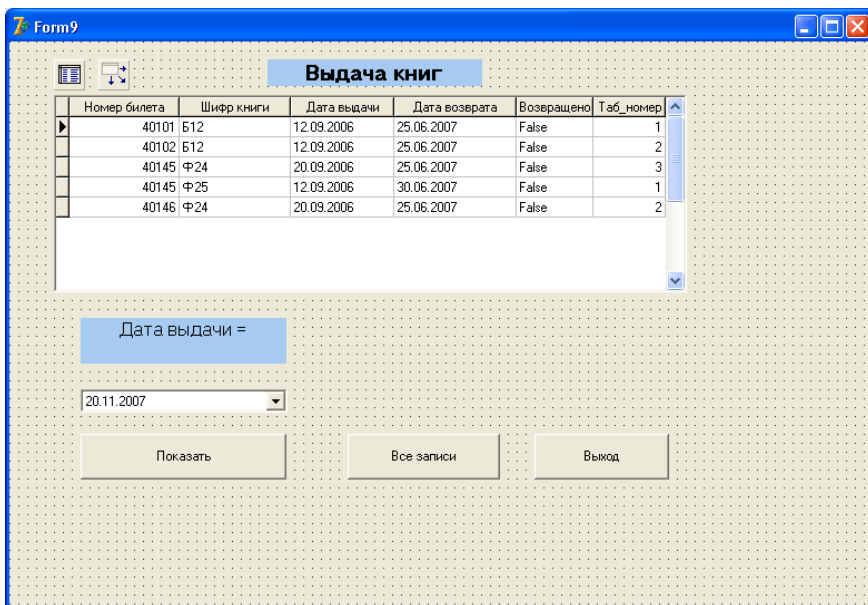


Рисунок 3.4 - Фильтрация по дате

Дата выдачи – это компонент Label1.

Дата создана как компонент DateTimePicker1 (вкладка Win32). При щелчке по кнопке раскрывающегося списка выводится календарь для выбора даты.

Кнопки **Показать** (Button1), **Все записи** (Button2), **Выход** (Button3).

Для кнопки **Показать** создан следующий обработчик событий.

```

procedure TForm9.Button1Click(Sender: TObject);
var
a:string[20];
begin
Table1.Filtered:=True;
a:=DateToStr(DateTimePicker1.Date);
Table1.Filter:='Dat_vid='+a+'';
end;

```

Для кнопки **Все записи** создан следующий обработчик событий.

```
procedure TForm9.Button2Click(Sender: TObject);  
begin  
Table1.Filtered:=False;  
end;
```

Для кнопки **Выход** создан следующий обработчик событий.

```
procedure TForm9.Button3Click(Sender: TObject);  
begin  
Form9.Close;  
end;
```

Для **формы в целом** создан следующий обработчик событий.

```
procedure TForm9.FormCreate(Sender: TObject);  
begin  
Table1.Filtered:=False;  
end;
```

4 Варианты курсовых работ

Вариант 1. Разработка приложения для работы с таблицей «Бензин»

Создать таблицу Бензин, таблица имеет следующую структуру: Номер чека, Марка бензина, Дата продажи, Количество, Цена за л. Ключевое поле – номер чека.

Создать форму для ввода данных в таблицу Бензин.

Реализовать следующие виды сортировки:

сортировка таблицы по марке бензина;

сортировка таблицы по количеству;

сортировка таблицы по дате продажи.

Реализовать следующие виды фильтрации:

фильтрация таблицы по марке бензина;

фильтрация по цене за л;

фильтрация таблицы по дате продажи.

Реализовать следующие вычисления:

вычислить суммарную продажу бензина в стоимостном выражении для всех марок бензина и для конкретной марки бензина, которая вводится;

определить покупку бензина с максимальной стоимостью покупки.

Вариант 2. Разработка пользовательского интерфейса для обработки данных в таблице «Стадион»

Создать таблицу Стадион, таблица имеет следующую структуру: Номер группы, Вид занятий, Фамилия и инициалы тренера, Количество занимающихся, Оплата, Продолжительность, Дата начала занятий. Ключевое поле – номер группы.

Создать форму для ввода данных в таблицу Стадион.

Реализовать следующие виды сортировки:

сортировка таблицы по виду занятий;

сортировка по количеству занимающихся;

сортировка таблицы по дате начала занятий.

Реализовать следующие виды фильтрации:

фильтрация таблицы по фамилии тренера;

фильтрация таблицы оплате;

фильтрация таблицы по дате начала занятий.

Реализовать следующие вычисления:

вычислить общую оплату занятий по всем группам и по конкретной группе, номер которой ввести;

определить группу, где минимальная оплата за занятия.

Вариант 3. Разработка приложения для работы с таблицей «Аэропорт»

Создать таблицу Аэропорт, таблица имеет следующую структуру: Номер рейса, Маршрут, Тип самолета, Количество занимающихся, Цена билета, Время вылета, Количество проданных билетов, Дата вылета. Ключевое поле – номер рейса.

Создать форму для ввода данных в таблицу Аэропорт.

Реализовать следующие виды сортировки:

сортировка таблицы по маршруту;

сортировка таблицы по дате вылета;

сортировка таблицы по типу самолета.

Реализовать следующие виды фильтрации:

фильтрация таблицы по маршруту;
фильтрация таблицы по цене билета;
фильтрация таблицы по дате вылета.

Реализовать следующие вычисления:

вычислить общую стоимость проданных билетов по всем рейсам и для конкретного рейса, номер которого вводится;
определить рейс, где максимальная стоимость купленных билетов.

Вариант № 4. Разработка пользовательского интерфейса для обработки данных в таблице «Книги»

Создать таблицу Книги, таблица имеет следующую структуру: Номер книги, Шифр книги, Авторы, Название, Цена, Область знаний, Проданное количество, Номер месяца, Продавец, Год издания. Ключевое поле – номер книги.

Создать форму для ввода данных в таблицу Книги.

Реализовать следующие виды сортировки:

сортировка таблицы по году издания;
сортировка таблицы по номеру месяца;
сортировка таблицы по области знаний.

Реализовать следующие виды фильтрации:

фильтрация таблицы по автору;
фильтрация таблицы по продавцу;
фильтрация таблицы по номеру месяца.

Реализовать следующие вычисления:

вычислить общую стоимость проданных книг и стоимость проданных книг за конкретный месяц, номер которого вводится;

определить книгу, где стоимость купленных книг максимальна.

Вариант № 5. Разработка приложения для работы с таблицей «Автомобильные перевозки»

Создать таблицу Автомобильные перевозки, таблица имеет следующую структуру: Номер заказа, Номер автомобиля, Модель автомобиля, Водитель, Дата перевозки, Цена за перевозку на 1 км, Расстояние. Ключевое поле – номер заказа.

Создать форму для ввода данных в таблицу Автомобильные перевозки.

Реализовать следующие виды сортировки:

сортировка таблицы по дате перевозки;
сортировка таблицы по номеру автомобиля;
сортировка таблицы по модели автомобиля.

Реализовать следующие виды фильтрации:

фильтрация таблицы по номеру автомобиля.
фильтрация таблицы по водителю;
фильтрация таблицы по дате перевозки.

Реализовать следующие вычисления:

вычислить общую стоимость перевозки грузов и стоимость перевозки груза для конкретного заказа, номер которого вводится;

определить заказ, где стоимость перевозки груза минимальна.

Вариант № 6. Разработка пользовательского интерфейса для обработки данных в таблице «Транзисторы»

Создать таблицу Транзисторы, таблица имеет следующую структуру: Номер чека, Тип прибора, Материал, Ток коллектора, Цена, Количество, Дата продажи, Продавец. Ключевое поле – номер чека.

Создать форму для ввода данных в таблицу Транзисторы.

Реализовать следующие виды сортировки:

сортировка таблицы по типу прибора;
сортировка таблицы по дате продажи;
сортировка таблицы по количеству.

Реализовать следующие виды фильтрации:

фильтрация таблицы по материалу;
фильтрация таблицы по цене;
фильтрация по дате продажи.

Реализовать следующие вычисления:

вычислить общую стоимость проданных транзисторов и стоимость проданных транзисторов конкретного типа прибора, тип прибора вводится;

определить тип прибора, где стоимость купленных транзисторов минимальна.

Вариант № 7. Разработка пользовательского интерфейса для обработки данных в таблице «Диоды»

Создать таблицу Диоды, таблица имеет следующую структуру: Номер партии, Тип прибора, Материал, Обратное напряжение, Средний ток, Цена, Количество на складе, Дата закупки. Ключевое поле – номер партии.

Создать форму для ввода данных в таблицу Диоды.

Реализовать следующие виды сортировки:

сортировка таблицы по типу прибора;
сортировка таблицы по количеству на складе;
сортировка таблицы по дате закупки.

Реализовать следующие виды фильтрации:

фильтрация таблицы по материалу;
фильтрация таблицы по цене;
фильтрация по дате закупки.

Реализовать следующие вычисления:

вычислить общую стоимость диодов на складе и стоимость диодов конкретного типа прибора, тип прибора вводится;

определить тип прибора, где стоимость закупленных диодов максимальна.

Вариант № 8. Разработка приложения для работы с таблицей «Мебель»

Создать таблицу Мебель, таблица имеет следующую структуру: Номер гарнитура, Название, Тип мебели, Количество предметов, Цена, Количество на складе, Дата закупки. Ключевое поле – номер гарнитура.

Создать форму для ввода данных в таблицу Мебель.

Реализовать следующие виды сортировки:

сортировка таблицы по количеству предметов;
сортировка таблицы по названию;
сортировка таблицы по дате закупки.

Реализовать следующие виды фильтрации:

фильтрация таблицы по типу мебели;

фильтрация таблицы по цене;

фильтрация по дате закупки.

Реализовать следующие вычисления:

вычислить общую стоимость гарнитуров на складе и стоимость гарнитуров конкретного типа мебели, тип мебели вводится;

определить гарнитур, который имеет максимальную цену.

Вариант № 9. Разработка пользовательского интерфейса для обработки данных в таблице «Студенты»

Создать таблицу Студенты, таблица имеет следующую структуру: Номер зачетки, Шифр группы, Фамилия, Имя, отчество, Стипендия, Дата зачисления. Ключевое поле – номер зачетки.

Создать форму для ввода данных в таблицу Студенты.

Реализовать следующие виды сортировки:

сортировка таблицы по шифру группы;

сортировка таблицы по дате зачисления;

сортировка таблицы по стипендии.

Реализовать следующие виды фильтрации:

фильтрация таблицы по шифру группы;

фильтрация таблицы по стипендии;

фильтрация по дате зачисления.

Реализовать следующие вычисления:

вычислить суммарную стипендию всех студентов и суммарную стипендию студентов конкретной группы, шифр группы вводится;

определить группу, имеющую минимальное количество студентов.

Вариант № 10. Разработка пользовательского интерфейса для обработки данных в таблице «Преподаватели»

Создать таблицу Преподаватели, таблица имеет следующую структуру: Номер преподавателя, Фамилия, Имя, отчество, Кафедра, Должность, Оклад, Дата поступления на работу. Ключевое поле – номер преподавателя.

Создать форму для ввода данных в таблицу Преподаватели.

Реализовать следующие виды сортировки:

сортировка таблицы по фамилии;

сортировка таблицы по дате поступления на работу;

сортировка таблицы по окладу.

Реализовать следующие виды фильтрации:

фильтрация таблицы по кафедре;

фильтрация таблицы по окладу;

фильтрация по дате поступления на работу.

Реализовать следующие вычисления:

вычислить суммарный оклад всех преподавателей и суммарный оклад преподавателей конкретной кафедры, кафедра вводится;

определить кафедру, имеющую максимальный суммарный оклад.

Вариант № 11. Разработка пользовательского интерфейса для обработки данных в таблице «Охрана»

Создать таблицу Охрана, таблица имеет следующую структуру: Номер заказа, Фамилия заказчика, Адрес, Дата начала охраны, Дата окончания охраны, Количество часов, Цена за час, Старший. Ключевое поле – номер заказа.

Создать форму для ввода данных в таблицу Охрана.

Реализовать следующие виды сортировки:

сортировка таблицы по адресу;

сортировка таблицы по дате начала охраны;

сортировка таблицы по цене за час.

Реализовать следующие виды фильтрации:

фильтрация таблицы по старшему;

фильтрация таблицы по цене за час;

фильтрация по дате начала охраны.

Реализовать следующие вычисления:

вычислить стоимость охранных работ по всем заказам и стоимость охранных работ по конкретному заказу, номер заказа вводится;

определить заказ, имеющую максимальную стоимость охранных работ.

Вариант № 12. Разработка пользовательского интерфейса для обработки данных в таблице «Нотариальные услуги»

Создать таблицу Нотариальные услуги, таблица имеет следующую структуру: Номер дела, Фамилия заказчика, Вид услуги, Дата оказания услуги, Стоимость услуги, Скидка, Юрист. Ключевое поле – номер дела.

Создать форму для ввода данных в таблицу Нотариальные услуги.

Реализовать следующие виды сортировки:

сортировка таблицы по виду услуг;

сортировка таблицы по дате оказания услуги;

сортировка таблицы по стоимости услуги.

Реализовать следующие виды фильтрации:

фильтрация таблицы по юристу;

фильтрация таблицы по стоимости услуги;

фильтрация по дате оказания услуги.

Реализовать следующие вычисления:

вычислить стоимость нотариальных услуг по всем заказам и стоимость нотариальных услуг по конкретному виду услуги, вид услуги вводится; при вычислении стоимости учесть скидку;

определить услугу, имеющую максимальную стоимость услуги.

Вариант № 13. Разработка пользовательского интерфейса для обработки данных в таблице «Компьютеры»

Создать таблицу Компьютеры, таблица имеет следующую структуру: Номер чека, Код устройства, Тип устройства, Описание, Дата продажи, Стоимость устройства, Скидка, Продавец. Ключевое поле – номер чека.

Создать форму для ввода данных в таблицу Компьютеры.

Реализовать следующие виды сортировки:

сортировка таблицы по коду устройства;

сортировка таблицы по дате продажи;

сортировка таблицы по продавцу.

Реализовать следующие виды фильтрации:

фильтрация таблицы по типу устройства;

фильтрация таблицы по скидке;

фильтрация по дате продажи.

Реализовать следующие вычисления:

вычислить стоимость проданных устройств и стоимость проданных устройств определенного типа, тип устройства вводится; при вычислении суммарной стоимости учесть скидку;

определить тип устройства, имеющий минимальную суммарную продажу в количественном выражении.

Вариант № 14. Разработка пользовательского интерфейса для обработки данных в таблице «Стекло»

Создать таблицу Стекло, таблица имеет следующую структуру: Номер партии, Код стекла, Название стекла, Размер, Дата закупки, Цена, Количество на складе. Ключевое поле – номер партии.

Создать форму для ввода данных в таблицу Стекло.

Реализовать следующие виды сортировки:

сортировка таблицы по коду стекла;

сортировка таблицы по дате закупки;

сортировка таблицы по названию стекла.

Реализовать следующие виды фильтрации:

фильтрация таблицы по дате закупки;

фильтрация таблицы по количеству на складе;

фильтрация по названию стекла.

Реализовать следующие вычисления:

вычислить стоимость стекла на складе и стоимость стекла определенного размера, размер вводится;

определить номер партии, имеющий максимальную стоимость стекла на складе.

Вариант № 15. Разработка пользовательского интерфейса для обработки данных в таблице «Гостиница»

Создать таблицу Гостиница, таблица имеет следующую структуру: Номер постояльца, Фамилия, Имя, Отчество, Номер комнаты, Категория комнаты, Дата поступления, Количество дней, Цена за сутки. Ключевое поле – номер постояльца.

Создать форму для ввода данных в таблицу Гостиница.

Реализовать следующие виды сортировки:

сортировка таблицы по номеру комнаты;

сортировка таблицы по дате поступления;

сортировка таблицы по фамилии.

Реализовать следующие виды фильтрации:

фильтрация таблицы по дате поступления;

фильтрация таблицы по категории комнаты;

фильтрация по номеру комнаты.

Реализовать следующие вычисления:

вычислить стоимость проживания всех постояльцев и стоимость проживания всех постояльцев, например, в люксе (категория комнаты);

определить номер постояльца, имеющего минимальную стоимость проживания.

Вариант № 16. Разработка пользовательского интерфейса для обработки данных в таблице «Вкладчики»

Создать таблицу Вкладчики, таблица имеет следующую структуру: Номер счета, Фамилия, Имя, Отчество, Вид вклада, Остаток денег, Процент начисления, Дата последней операции по вкладу. Ключевое поле – номер счета.

Создать форму для ввода данных в таблицу Вкладчики.

Реализовать следующие виды сортировки:

сортировка таблицы по фамилии;

сортировка таблицы по дате последней операции;

сортировка таблицы по остатку денег.

Реализовать следующие виды фильтрации:

фильтрация таблицы по дате последней операции;

фильтрация таблицы по виду вклада;

фильтрация по номеру проценту начисления.

Реализовать следующие вычисления:

вычислить общую стоимость всех денег по счетам и общую стоимость всех денег по конкретному виду вклада; вид вклада вводится;

определить номер счета, имеющего максимальный остаток денег на счете.

Вариант № 17. Разработка пользовательского интерфейса для обработки данных в таблице «Автовокзал»

Создать таблицу Автовокзал, таблица имеет следующую структуру: Номер автобуса, Название маршрута, Водитель, Количество мест, Цена за билет, Марка автомобиля, Дата начала эксплуатации. Ключевое поле – номер маршрута.

Создать форму для ввода данных в таблицу Автовокзал.

Реализовать следующие виды сортировки:

сортировка таблицы по названию маршрута;

сортировка таблицы по дате начала эксплуатации;

сортировка таблицы по количеству мест.

Реализовать следующие виды фильтрации:

фильтрация таблицы по дате начала эксплуатации;

фильтрация таблицы по марке автомобиля;

фильтрация по номеру цене за билет.

Реализовать следующие вычисления:

вычислить общую стоимость всех мест по всем автобусам и общую стоимость всех мест в конкретном автобусе; номер автобуса вводится;

определить номер автобуса, имеющего минимальное количество мест.

Вариант № 18. Разработка пользовательского интерфейса для обработки данных в таблице «Стоматология»

Создать таблицу Стоматология, таблица имеет следующую структуру: Номер записи, Номер больного, Фамилия и инициалы больного, Диагноз, Дата посещения, Стоимость лечения, Врач. Ключевое поле – номер записи.

Создать форму для ввода данных в таблицу Стоматология.

Реализовать следующие виды сортировки:

сортировка таблицы по фамилии больного;

сортировка таблицы по дате посещения;
сортировка таблицы по стоимости лечения.

Реализовать следующие виды фильтрации:

фильтрация таблицы по дате посещения;
фильтрация таблицы по врачу;
фильтрация по номеру больного.

Реализовать следующие вычисления:

вычислить общую стоимость лечения всех больных и общую стоимость лечения всех больных конкретного врача; врач вводится;

определить номер записи, где стоимость лечения максимальная.

Вариант № 19. Разработка пользовательского интерфейса для обработки данных в таблице «Лекарства»

Создать таблицу Лекарства, таблица имеет следующую структуру: Номер партии, Номер лекарства, Название лекарства, Группа лекарств, Дата закупки, Количество в упаковке, Цена за упаковку, Количество на складе. Ключевое поле – номер партии.

Создать форму для ввода данных в таблицу Лекарства.

Реализовать следующие виды сортировки:

сортировка таблицы по названию лекарства;
сортировка таблицы по дате закупки;
сортировка таблицы по количеству на складе.

Реализовать следующие виды фильтрации:

фильтрация таблицы по дате закупки;
фильтрация таблицы по группе лекарств;
фильтрация по количеству в упаковке.

Реализовать следующие вычисления:

вычислить общую стоимость лекарств на складе и общую стоимость на складе конкретной группы; группа лекарств вводится;

определить номер партии, где стоимость закупленных лекарств максимальная.

Вариант № 20. Разработка пользовательского интерфейса для обработки данных в таблице «Группа здоровья»

Создать таблицу Группа здоровья, таблица имеет следующую структуру: Номер группы, Вид занятий, Тренер, Дата начала занятий, Продолжительность, Количество людей в группе, Стоимость посещения, День недели, Время, Номер бассейна. Ключевое поле – номер группы.

Создать форму для ввода данных в таблицу Группа здоровья.

Реализовать следующие виды сортировки:

сортировка таблицы по тренеру;

сортировка таблицы по дате начала занятий;

сортировка таблицы по стоимости посещения.

Реализовать следующие виды фильтрации:

фильтрация таблицы по дате начала занятий;

фильтрация таблицы по виду занятий;

фильтрация по количеству людей в группе.

Реализовать следующие вычисления:

вычислить общую стоимость посещения по всем группам и общую стоимость посещения по конкретному виду занятий; вид занятий вводится;

определить номер группы, где минимальное количество людей в группе.

Выбор варианта курсовой работы осуществляется с помощью таблицы 4.1.

Таблица 4.1 - Таблица выбора вариантов курсовой работы

Послед. цифра номера студ. билета	Предпоследняя цифра номера студенческого билета									
	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	11	12	13	14	15	16	17	18	19	20
2	1	2	3	4	5	6	7	8	9	10
3	11	12	13	14	15	16	17	18	19	20
4	1	2	3	4	5	6	7	8	9	10

Продолжение таблицы 4.1

По- след. цифра номера студ. билета	Предпоследняя цифра номера студенческого билета									
	0	1	2	3	4	5	6	7	8	9
5	11	12	13	14	15	16	17	18	19	20
6	1	2	3	4	5	6	7	8	9	10
7	11	12	13	14	15	16	17	18	19	20
8	1	2	3	4	5	6	7	8	9	10
9	11	12	13	14	15	16	17	18	19	20

Список литературы

1 Фаронов В.В. Турбо Паскаль 7.0. / В.В. Фаронов. М.: Нолидж, 1999. 616 с.

2 Фаронов В.В. Программирование баз данных в Delphi 7 / В.В. Фаронов. СПб.: Питер, 2004.

3 Шумаков П.В. Delphi 5. Руководство разработчика баз данных / П.В. Шумаков, В.В. Фаронов. М.: Нолидж, 2001.

4 Бобровский С.И. Delphi 7. Учебный курс / С.И. Бобровский. СПб.: Питер, 2001.

5 Избачков Ю.С. Информационные системы: учебник для вузов / Ю.С. Избачков, В.Н. Петров. 2-е изд. СПб.: Питер, 2005.

6 Бабушкина И.А. Практикум по объектно-ориентированному программированию / И.А. Бабушкина. М.: Бином, 2004. 366 с.

ПРИЛОЖЕНИЕ А

Федеральное агентство по образованию
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»
(ГОУВПО «ВГТУ»)

Факультет автоматики и электромеханики

Кафедра «Автоматизированные и вычислительные системы»
Специальность «Вычислительные машины, комплексы,
системы и сети»

КУРСОВАЯ РАБОТА

по дисциплине «Технология программирования»
Тема: «Разработка приложения для работы
с таблицей «Бензин»»

Расчетно-пояснительная записка

Разработал(а) студент(ка)

Подпись, дата Инициалы, фамилия

Руководитель

Подпись, дата Инициалы, фамилия

Нормоконтролер

Подпись, дата Инициалы, фамилия

Защищена _____

Дата

Оценка _____

2008

ПРИЛОЖЕНИЕ Б

Федеральное агентство по образованию
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ГОУВПО «ВГТУ»)
Кафедра «Автоматизированные и вычислительные системы»

ЗАДАНИЕ на курсовую работу

по дисциплине «Технология программирования»
Тема: «Разработка приложения для работы с таблицей «Бензин»»

Студент группы ВМ-061 Сорокин Александр Николаевич
Фамилия, имя, отчество

Номер варианта _____

Технические условия: осуществить разработку приложения в среде Delphi.

Содержание и объем работы (графические работы, расчеты и прочее): изучение технологии разработки пользовательского интерфейса в среде Delphi; освоение технологии создания таблицы, меню, форм для ввода и редактирования данных, реализации сортировки, фильтрации, вычислений в таблице; составление расчетно-пояснительной записки.

Сроки выполнения этапов: изучение технологии разработки пользовательского интерфейса в Delphi – до 15.03.2008 г., реализация практического задания – до 15.04.2008 г., оформление расчетно-пояснительной записки – до 15.05.2008г.

Срок защиты курсовой работы _____

Руководитель _____

Подпись, дата Инициалы, фамилия

Задание принял студент _____

Подпись, дата Инициалы, фамилия

ПРИЛОЖЕНИЕ В ЗАМЕЧАНИЯ РУКОВОДИТЕЛЯ

ПРИЛОЖЕНИЕ Г

СОДЕРЖАНИЕ

Введение	3
1 Разработка приложений в среде Delphi	5
1.1 Проектирование пользовательского интерфейса.....	5
1.2 Разработка типовых элементов пользовательского интерфейса.....	8
2 Практическая часть.....	18
2.1 Создание таблицы и формы для ввода данных.....	18
2.2 Обработка данных в таблице «Бензин»	22
Заключение	25
Список литературы	27

ПРИЛОЖЕНИЕ Д

Пример оформления списка литературы

1 Проектирование пользовательского интерфейса на персональных компьютерах. Стандарт фирмы ИВМ / Под ред. М. Дадашова. Киев: DBS Ltd, 1991.

2 Калянов Г.Н. CASE – структурный системный анализ. / Г.Н. Калянов. М: Лори, 1996.

3 Шлеер С. Объектно-ориентированный анализ: моделирование мира в состояниях / С. Шлеер, С. Меллор. Киев: Диалектика, 1993.

4 Буч Г. Объектно-ориентированное проектирование с примерами применения / Г. Буч. Киев: Диалектика, 1992.

5 Павлов В.Д. Технология разработки программных средств / В.Д. Павлов. Курск: КГТУ, 1996.

6 Мак-Грегор Дж. Тестирование объектно-ориентированного обеспечения / Дж. Мак-Грегор, Д. Сайкс. Киев: DiaSoft, 2002.

7 Орлов С.А. Технологии разработки программного обеспечения / С.А. Орлов. СПб: Питер, 2003.

8 Майерс Г. Искусство тестирования программ / Г. Майерс. М.: Финансы и статистика, 1982.

ТЕХНОЛОГИЯ СОЗДАНИЯ ПОЛЬЗОВАТЕЛЬСКОГО
ИНТЕРФЕЙСА

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению курсовой работы
по дисциплине «Технология программирования»
для студентов специальности 230101
«Вычислительные машины, комплексы, системы и сети»
заочной и сокращенной заочной форм обучения

Составители: Гребенникова Наталия Ивановна
Сергеева Татьяна Ивановна

В авторской редакции

Подписано в печать 26.02.2008.
Формат 60x84/16. Бумага для множительных аппаратов.
Усл. печ. л. 3,0. Уч.-изд. л. 2,8. Тираж 100 экз. «С» 48.
Заказ № 56.

ГОУВПО «Воронежский государственный
технический университет»
394026 Воронеж, Московский просп., 14