

МИНОБРНАУКИ РОССИИ

Орский гуманитарно-технологический институт (филиал)  
федерального государственного бюджетного образовательного учреждения  
высшего образования «Оренбургский государственный университет»  
(Орский гуманитарно-технологический институт (филиал) ОГУ)

Кафедра программного обеспечения

Методические указания  
для обучающихся по освоению дисциплины

«Б.1.В.ОД.15 Программная инженерия»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

09.03.03 Прикладная информатика

(код и наименование направления подготовки)

Прикладная информатика в экономике

(наименование направленности (профиля) образовательной программы)

Тип образовательной программы

Программа академического бакалавриата

Квалификация

Бакалавр

Форма обучения

Очная

Год начала реализации программы (набора)

2014, 2015, 2016

г. Орск 2017

Методические указания для обучающихся по освоению дисциплины «Б.1.В.ОД.15 Программная инженерия» предназначены для обучающихся очной и заочной форм обучения направления подготовки 09.03.03 Прикладная информатика, профиля «Прикладная информатика в экономике»

Составитель



О.В. Подсобляева

Методические указания рассмотрены и одобрены на заседании кафедры программного обеспечения, протокол № 9 от «07» июня 2017 г.

Заведующий кафедрой программного обеспечения



Е.Е.Сурина

© Подсобляева О.В., 2017  
© Орский гуманитарно-технологический институт (филиал) ОГУ, 2017

## **1 Методические указания по проведению лекционных занятий**

Лекционные занятия в высшем учебном заведении являются основной формой организации учебного процесса и должны быть нацелены на выполнение ряда задач:

- ознакомить студентов со структурой дисциплины;
- изложить основной материал программы курса дисциплины;
- ознакомить с новейшими подходами и проблематикой в данной области;
- сформировать у студентов потребность к самостоятельной работе с учебной, нормативной и научной литературой.

Лекционное занятие представляет собой систематическое, последовательное, монологическое изложение преподавателем-лектором учебного материала, как правило, теоретического характера.

Цель лекции – организация целенаправленной познавательной деятельности студентов по овладению программным материалом учебной дисциплины.

Чтение курса лекций позволяет дать связанное, последовательное изложение материала в соответствии с новейшими данными науки, сообщить слушателям основное содержание предмета в целостном, систематизированном виде.

В ряде случаев лекция выполняет функцию основного источника информации, когда новые научные данные по той или иной теме не нашли отражения в учебниках.

Организационно-методической базой проведения лекционных занятий является рабочий учебный план направления подготовки. При подготовке лекционного материала преподаватель обязан руководствоваться учебными программами по дисциплинам кафедры, тематика и содержание лекционных занятий которых представлена в рабочих программах, учебно-методических комплексах.

При чтении лекций преподаватель имеет право самостоятельно выбирать формы и методы изложения материала, использовать различные технические средства обучения.

Рекомендации по работе студентов с конспектом лекций.

Изучение дисциплины студенту следует начинать с проработки рабочей программы, особое внимание, уделяя целям и задачам, структуре и содержанию курса.

При конспектировании лекций студентам необходимо излагать услышанный материал кратко, своими словами, обращая внимание, на логику изложения материала, аргументацию и приводимые примеры. Необходимо выделять важные места в своих записях. Если непонятны какие-либо моменты, необходимо записывать свои вопросы, постараться найти ответ на них самостоятельно. Если самостоятельно не удалось разобраться в материале, впоследствии необходимо либо на следующей лекции, либо на лабораторном занятии или консультации обратиться к ведущему преподавателю за разъяснениями.

Успешное освоение курса предполагает активное, творческое участие студента путем планомерной, повседневной работы. Лекционный материал следует просматривать в тот же день. Рекомендуемую дополнительную литературу следует прорабатывать после изучения данной темы по учебнику и материалам лекции.

Каждая тема имеет свои специфические термины и определения. Усвоение материала необходимо начинать с усвоения этих понятий. Если какое-либо понятие вызывает затруднения, необходимо посмотреть его суть и содержание в словаре (Интернете), выписать его значение в тетрадь для подготовки к занятиям.

При подготовке материала необходимо обращать внимание на точность определений, последовательность изучения материала, аргументацию, собственные примеры, анализ конкретных ситуаций. Каждую неделю рекомендуется отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам и тестам.

## 2 Методические указания по лабораторным работам

Изучение дисциплины «Программная инженерия» предполагает посещение обучающимися не только лекций, но и лабораторных работ. Лабораторные работы со студентами предназначены для проверки усвоения ими теоретического материала дисциплины.

Основные цели лабораторных работ:

- закрепить основные положения дисциплины;
- проверить уровень усвоения и понимания студентами вопросов, рассмотренных на лекциях и самостоятельно изученных по учебной литературе;
- научить пользоваться нормативной и справочной литературой для получения необходимой информации о конкретных технологиях;
- оказать помощь в приобретении навыков расчета точностных характеристик;
- восполнить пробелы в пройденной теоретической части курса и оказать помощь в его усвоении.

Для контроля знаний, полученных в процессе освоения дисциплины на лабораторных занятиях обучающиеся выполняют задания реконструктивного уровня и комплексное практическое задание.

Целью выполнения задания реконструктивного уровня и комплексного задания студентами является систематизация, закрепление и расширение теоретических знаний, полученных в ходе изучения дисциплины.

Ниже приводятся общие методические указания, которые относятся к занятиям по всем темам:

- в начале каждого лабораторного занятия необходимо сформулировать цель, поставить задачи;
- далее необходимо проверить знания студентами лекционного материала по теме занятий;
- в процессе занятия необходимо добиваться индивидуальной самостоятельной работы студентов;
- знания студентов периодически контролируются путем проведения текущей аттестации (рубежного контроля), сведения о результатах которой доводятся до студентов и подаются в деканат;
- время, выделенное на отдельные этапы занятий, указанное в рабочей программе, является ориентировочным; преподаватель может перераспределить его, но должна быть обеспечена проработка в полном объеме приведенного в рабочей программе материала;
- на первом занятии преподаватель должен ознакомить студентов с правилами поведения в лаборатории и провести инструктаж по охране труда и по пожарной безопасности на рабочем месте;
- преподаватель должен ознакомить студентов со всем объемом лабораторных работ и требованиями, изложенными выше;
- преподаватель уделяет внимание оценке активности работы студентов на занятиях, определению уровня их знаний на каждом занятии.

На лабораторных работах решаются задачи из всех разделов изучаемой дисциплины.

## Лабораторная работа №1 Создание таблиц баз данных

Создайте в BDE Administrator новый псевдоним со своей фамилией. Укажите драйвер БД по умолчанию Paradox, а в параметре Path укажите путь к созданной вами папке



Рисунок 1

Для работы с таблицами БД при проектировании приложения удобно использовать Database Desktop, которая позволяет:

- создавать таблицы
- изменять структуры
- редактировать записи.



Рисунок 2

Кроме того, с помощью Database Desktop можно выполнять и другие действия над БД (создание, редактирование и выполнение визуальных и SQL – запросов, операций с псевдонимами).

Процесс создания новой таблицы начинается с вызова команды File/New/Table (Файл/Новая/Таблица) и происходит в интерактивном режиме. При этом разработчик должен:

- выбрать формат (тип) таблицы
- задать структуру таблицы.

В начале создания новой таблицы в окне Create Table (создание таблицы) выбирается ее формат. По умолчанию предлагается формат таблицы Paradox, который и будем использовать. Для таблиц других форматов, например dBase IV, действия по созданию таблицы практически не отличаются.

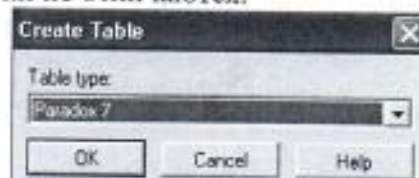


Рисунок 3

После выбора формата таблицы появляется окно определения структуры таблицы, в котором выполняются следующие действия:

- описание полей
- задание ключа
- задание индексов

- определение ограничений на значения полей
- определение условий ссылочной целостности
- задание паролей
- задание языкового драйвера
- задание таблицы для выбора значений

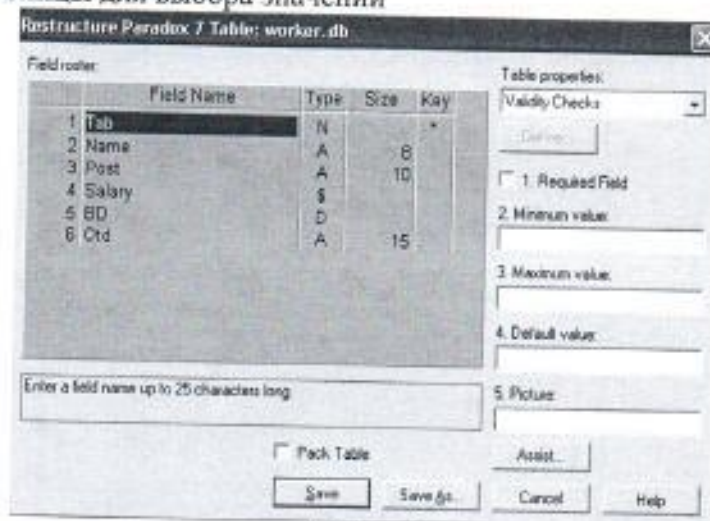


Рисунок 4

В этом списке обязательным является первое действие, т.е. каждая таблица должна иметь хотя бы одно поле. Остальные действия выполняются при необходимости. Часть действий, такие как задание ключа и паролей, производится только для таблиц определенных форматов, например, для таблиц Paradox.

При создании новой таблицы, сразу после выбора ее формата можно не создавать структуру таблицы, а скопировать ее из другой таблицы: при нажатии кнопки Borrow (Взаймы) открывается окно Select Borrow Table (выбор таблицы для заимствования).

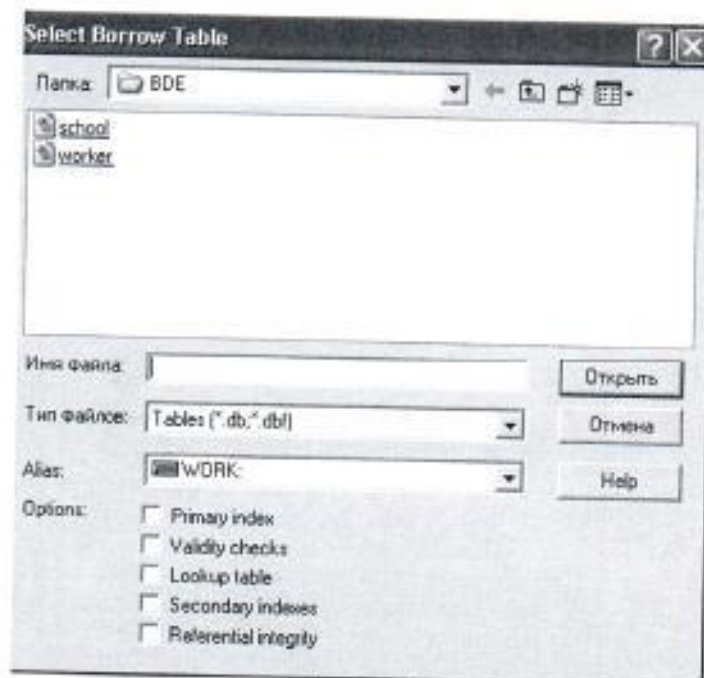


Рисунок 5

В этом окне можно выбрать таблицу и указать копируемые элементы структуры, установив соответствующий флажок, например, Primary index (первичный индекс) для ключа. После нажатия кнопки Open из выбранной таблицы в новую копируются описания полей, а также те элементы, для которых установлен флажок. Если какой-либо элемент в структуре копируемой таблицы отсутствует, то состояние флажка не имеет значения. Например, если в выбранной таблице не определены ограничения ссылочной целостности,

то в новой таблице они не появятся, даже если установлен флажок Referential integrity (ссылочная целостность).

Впоследствии скопированную структуру можно настраивать, изменяя, добавляя или удаляя ее отдельные элементы.

После определения структуры таблицы ее необходимо сохранить, нажав кнопку Save as и указав расположение таблицы на диске и ее имя. В результате на диск записывается новая таблица, первоначально пустая, при этом все необходимые файлы создаются автоматически.

### Описание полей

Центральной частью окна определения структуры таблицы является список Field roster (список полей), в котором указываются поля таблицы. Для каждого поля задаются:

- имя поля – в столбце Field Name
- тип поля – в столбце Type
- размер поля - в столбце Size

Имя поля вводится по правилам, установленным для выбранного формата таблиц.

Type
Alpha
Number
\$(Money)
Short
Long Integer
# (BCD)
Date
Time
@ (Timestamp)
Memo
Formatted Memo
Graphic
OLE
Logical
+ (Autoincrement)
Binary
Bytes

Рисунок 6

Тип поля можно задать непосредственно указав соответствующий символ, выбрав его в списке. Список содержит все типы полей, допустимые для заданного формата таблицы.

Размер поля задается не всегда, необходимость его указания зависит от типа поля. Для полей определенного типа, например, автоинкрементного (+) или целочисленного, размер поля не задается. Для поля строкового типа размер определяет максимальное число символов, которые могут храниться в поле. Добавление к списку полей новой строки выполняется переводом курсора вниз на несуществующую строку, в результате чего эта строка появляется в конце списка.

Для указания ключевых полей в столбце ключа (Key) нужно установить символ \*, переведя в эту позицию курсор и нажав любую клавишу.

Для выполнения остальных действий по определению структуры таблицы используется комбинированный список Table Properties (свойства таблицы).

### Создание приложения BDE

Для примера рассмотрим создание приложения, использующего механизм доступа BDE и позволяющего перемещаться по записям таблицы БД, просматривать и редактировать поля, удалять записи из таблицы, а также вставлять новые. Файл проекта приложения обычно не требует от разработчика выполнения каких – либо действий. Поэтому при создании приложения главной задачей является конструирование форм, в простейшем случае – одной формы.

Вид формы приложения на этапе проектирования показан на рисунке, где в форме размещены компоненты Table1, DataSource1, DBGrid, DBNavigator1.

Компонент Table1 обеспечивает взаимодействие с таблицей БД. Для связи с требуемой таблицей нужно установить в соответствующее значения свойство DataBaseName, указывающее путь к БД, и свойство TableName, указывающее имя таблицы. После задания таблицы для открытия набора данных свойство Active должно быть установлено в значение True.

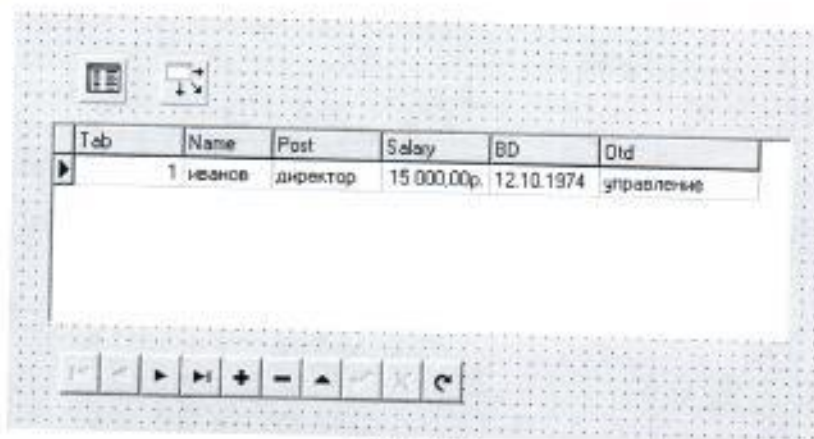


Рисунок 7

Компонент DBGrid отображает содержимое таблицы БД в виде сетки, в которой столбцы соответствуют полям, а строки – записям таблицы. По умолчанию пользователь может просматривать и редактировать данные. Компонент DBNavigator позволяет пользователю перемещаться по таблице, редактировать, вставлять и удалять записи. Компоненты DBGrid и DBNavigator связываются со своим источником данных – компонентов DataSource1 – через свойства DataSource.

## Лабораторная работа №2 Базы данных в Delphi

1) С помощью утилиты DataBase Desktop создайте таблицу для хранения информации об учениках школы. Пусть файл School.db содержит поля:

N – личный номер ученика (числовое поле с автоувеличением)

Name – имя (строковый тип)

Fam – фамилия (строковый тип)

Class – класс (строковый тип)

Adr – адрес (строковый тип)

2) В программе используется псевдоним Школа. Вам следует использовать свой псевдоним при помощи BDE Administrator.

3) Создайте форму приложения в Delphi. На форму поместите компоненты Table1 (вкладка BDE) и DataSource1 (вкладка Data Access), которые обеспечивают доступ к файлу данных, и установите следующие значения их свойств:

Свойство	Значение
Table1.DatabaseName	Школа
Table1.TableName	School.db
Table1.Active	True



DataSource1.Dataset	Table1
---------------------	--------

4) Для просмотра и редактирования данных в режиме таблицы в форму надо добавить компонент DBGrid1 (вкладка Data Controls).

5) Чтобы задать, какая информация будет отображаться в таблице DBGrid1, необходимо задать источник данных для таблицы (значение свойства DataSource = DataSource1).

Свойство Options.dgTitles разрешает вывод строки заголовка столбцов, Options.dgIndicator разрешает вывод колонки индикатора во время работы с базой данных (треугольник – текущая запись, звездочка – новая запись).

6) Значения уточняющих параметров свойства Columns задаются в окне Editing DBGrid1.Columns. Добавьте в компонент DBGrid1 4 колонки с помощью команды Добавить новый(Ins). Свойство Columns представляет собой массив компонентов типа TColumn. Каждой колонке соответствует элемент массива.

Свойство	Определяет
FieldName	Поле записи, содержимое которого выводится в колонке
Width	Ширину колонки в пикселах
Font	Шрифт, используемый для вывода текста в ячейках колонки
Color	Цвет фона колонки
Alignment	Способ выравнивания текста в ячейках колонки. Текст может быть выровнен по левому краю (taLeftJustify), по центру (taCenter) или по правому краю (taRightJustify)
Title.Caption	Заголовок колонки. Значением по умолчанию является имя поля записи
Title.Alignment	Способ выравнивания заголовка колонки. Заголовок может быть выровнен по левому краю (taLeftJustify), по центру (taCenter) ИЛИ ПО правому краю (taRightJustify)
Title.Color	Цвет фона заголовка колонки
Title.Font	Шрифт заголовка колонки

В простейшем случае для каждой колонки достаточно установить значение свойства FieldName, которое определяет имя поля записи, содержимое которого будет отображаться в колонке, а также значение свойства Title.caption, определяющего заголовок колонки.

7) Добавить к форме компонент DBNavigator, настроив его на работу с таблицей-источником данных (свойству DataSource нужно присвоить значение DataSource1).

8) Откомпилируйте и запустите программу. Следует обратить внимание, что для того чтобы после запуска программы в окне появилась информация или, если база данных пустая, можно было вводить новую информацию, свойство Active таблицы-источника данных должно иметь значение True.

Используя клавиши перемещения курсора вверх и вниз, а также клавиши листания текста страницами (<Page Up> и <Page Down>), можно, перемещаясь от строки к строке, просматривать записи базы данных. Нажав клавишу <Ins>, можно добавить запись, а нажав клавишу <Del> — удалить запись. Для того чтобы внести изменения в поле записи, нужно, используя клавиши перемещения курсора влево и вправо, выбрать необходимое поле и нажать клавишу <F2>.

### Выбор информации из базы данных

При работе с базой данных пользователя, как правило, интересует не все ее содержимое, а некоторая конкретная информация. Найти нужные сведения можно

последовательным просмотром записей. Однако такой способ поиска неудобен и малоэффективен.

Большинство систем управления базами данных позволяют произвести выборку нужной информации путем выполнения *запросов*. Пользователь в соответствии с определенными правилами формулирует запрос, указывая, каким критериям должна удовлетворять интересующая его информация, а система выводит записи, удовлетворяющие запросу.

Для выборки из базы данных записей, удовлетворяющих некоторому критерию, предназначен компонент Query (вкладка BDE).

Компонент Query похож на компонент Table, но, в отличие от последнего, он представляет не всю базу данных (все записи), а только ее часть — записи, удовлетворяющие критерию запроса. В таблице перечислены некоторые свойства компонента Query.

Свойство	Определяет
Name	Имя компонента. Используется компонентом Datasource для связи результата выполнения запроса (набора записей) с компонентом, обеспечивающим просмотр записей, например DBGrid
SQL	Записанный на языке SQL запрос к базе данных (к таблице)
Active	При присвоении свойству значения True активизирует выполнение запроса

Для того чтобы во время разработки программы задать, какая информация будет выделена из базы данных в результате выполнения запроса, свойство SQL должно содержать представленный на языке SQL запрос на выборку данных.

Например, запрос

```
SELECT Fam, Name  
FROM 'Школа:school.db'  
WHERE (Class = '11a')  
ORDER BY Name, Fam
```

обеспечивает выборку из базы данных "Школа" (из таблицы School.db) списка учеников 11-а класса.

Следующий запрос:

```
SELECT Fam, Name FROM "Школа:school.db"  
WHERE (Fam > 'K') and (Fam < 'Л')  
ORDER BY Name, Fam
```

обеспечивает выбор информации об учениках, фамилии которых начинаются на букву К.

Запрос может быть сформирован и записан в свойство SQL во время разработки формы или во время работы программы.

Для записи запроса в свойство SQL во время разработки формы используется редактор списка строк, окно которого открывается в результате щелчка на кнопке с тремя точками в строке свойства SQL окна

#### Object Inspector.

Свойство sql представляет собой список строк. Поэтому чтобы сформировать запрос во время работы программы, нужно, используя метод Add, добавить строки (Sql-инструкции) в список sql.

Ниже приведен фрагмент кода, который формирует запрос на поиск информации о конкретном человеке (критерий выбора — содержимое поля Fam должно совпадать со значением переменной fam).

```
with form1.Query1 do
```

```
begin
```

```
Close; // закрыть файл — результат выполнения предыдущего запроса
```

```
SQL.Clear; // удалить текст предыдущего запроса
```

```

// записываем новый запрос в свойство SQL
SQL.Add('SELECT Fam, Name, Class');
SQL.Add('FROM ":Школа:school.db" WHERE');
SQL.Add('Fam = "' + fam + '"');
SQL.Add('ORDER BY Fam');
Open; // активизируем выполнение запроса
end;

```

Следующая программа (диалоговое окно приведено на рисунке), демонстрирует возможность изменения запроса, точнее, критерия запроса, во время работы программы. Программа обеспечивает вывод как всего списка учеников, так и его части. Например, посредством выполнения запроса выводится информация только о конкретном ученике.

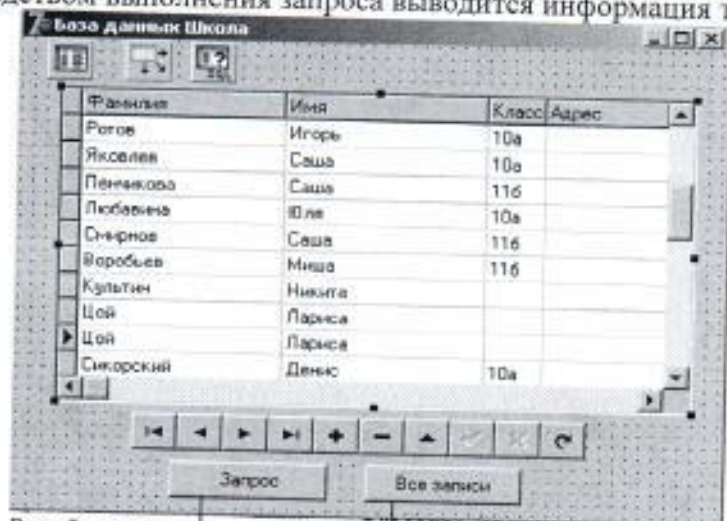


Рис. Форма приложения База данных Школа

```

Button1 | Button2
procedure TForm1.Button1Click(Sender: TObject);
var
  fam: string[30];
begin
  fam:=InputBox('Выборка информации', 'Укажите фамилию и нажмите ОК.', '');
  if fam <> '' then
    begin
      with form1.Query1 do begin
        Close;
        SQL.Clear;
        SQL.Add('SELECT Fam, Name, Class');
        SQL.Add('FROM ":Школа:school.db" WHERE Fam = "'+ fam + '"');
        Open;
      end;
      if Query1.RecordCount <> 0 then
        DataSource1.DataSet:=Query1
      else begin
        ShowMessage('В базе нет данных, удовлетворяющих критерию запроса');
        DataSource1.DataSet:=Table1;
      end;
    end;
end; // конец proc

procedure TForm1.Button2Click(Sender: TObject);
begin
  DataSource1.DataSet:=Table1; // задается источник данных - таблица
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
  DataSource1.DataSet := Table1;
  Table1.Active := True;
end;

```

## Индивидуальные лабораторные работы

### 1 Вариант

1. Создать БД по учету кварт платы из двух таблиц:  
Жильцы(Код жильца, ФИО, Улица, дом, квартира, площадь, кол-во проживающих)  
Платеж (Код платежа, Код жильца, дата, сумма платежа)
2. Определить ключи, связав таблицы
3. Составить запросы:
  - все сведения о квартиросъемщиках с фамилией, начинающейся с «А»
  - данные по текущему месяцу, с подведением итогов
  - расчет квартплаты по каждому жильцу при следующих условиях – 2р. за кв. метр+ 0,5р. за каждого проживающего

### 2 Вариант

1. Создать БД по учету успеваемости студентов  
Предметы(Код\_предмета, название предмета, группа, ФИО преподавателя)  
Студенты (Код\_студента, ФИО, факультет, группа)  
Экзамены( Код, Код предмета, Код\_студента, оценка, дата сдачи)
2. Определить ключи, связав таблицы по заданным полям
3. Составить запросы:
  - все сведения о студентах первого курса технологического факультета
  - средний балл каждого студента заданной с клавиатуры группы
  - сведения о студентах имеющих 2 по предметам
  - средний балл по каждому курсу каждого факультета

### 3 Вариант

1. Создать БД по анализу отгрузки оборудования по заводу
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
  - загрузка оборудования каждого цеха по кварталам, ч/з группировку
  - данные по текущему месяцу, с подведением итогов
  - сведения о заданном оборудовании
  - о начислении штрафа цеху в размере 15%, если не выполнен план.

### 4 Вариант

1. Создать БД по учету поставки продукции по заводу  
Продукция (Код, Наименование, Кол-во, цена)  
Заказчики (Код, ФИО, Адрес), Заказы (Код,
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
  - план поставки продукции по заказчикам в разрезе кварталов
  - информация о заданном заказчике
  - итоговая ведомость оплаты заказчикам
  - о расчете долга по уплате поставки

### 5 Вариант

1. Создать БД с информацией о кол-ве рабочих по цехам, профессиям, разрядам, для анализа средней зарплаты
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
  - средняя з/плата по заводу в разрезе профессий и разрядов

- данные по определенной профессии с выводом информации о з/п
- сведения о заданном цехе
- о начислении премий работникам, перевыполнившим план, ч/з функцию.

#### 6 Вариант

1. Создать БД по учету товара на складе и его отправке клиентам
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
  - о клиентах сделавших заказ на МАХ сумму
  - данные по текущему месяцу, с подведением итогов
  - сведения о недостающих на складе товарах
  - о начислении стоимости за товары каждого вида и заказа в целом

#### 7 Вариант

1. Создать БД по анализу выполнения плана рабочими на п/п
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
  - численность рабочих в определенном цехе
  - данные по выполнению плана рабочими всего п/п
  - сведения о нормативном плане на 1 рабочего (расчитать)
  - о снятии штрафа в размере 1 мин. з/платы если не выполнен план, ч/з функцию

#### 8 Вариант

1. Создать БД по учету выдачи в прокат видео кассет
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
  - о клиенте взявшем более 3 кассет
  - данные по клиентам, которым начисляется штраф, в связи с невозвращением кассет, ч/з функцию
  - сведения о всех клиентах с учетом стоимости
  - о кассетах с фильмами-боевиками

#### 9 Вариант

1. Создать БД по учету выдачи в прокат книг в библиотеке
2. Определить ключи, связав таблицы по заданными полями
3. Составить запросы:
  - о клиенте взявшем более 3 книг
  - данные по клиентам, которым начисляется штраф, в связи с невозвращением книг, ч/з функцию
  - сведения о всех клиентах с учетом стоимости
  - о книгах-детективах

### 3 Методические указания по самостоятельной работе

Для успешного освоения курса «Программная инженерия» необходима самостоятельная работа. В настоящее время актуальными становятся требования к личным качествам современного студента – умению самостоятельно пополнять и обновлять знания, вести самостоятельный поиск необходимого материала, быть творческой личностью.

Самостоятельную работу по освоению дисциплины обучающимся следует

начинать с изучения содержания рабочей учебной программы дисциплины, цели и задач, структуры и содержания курса, основной и дополнительной литературы, рекомендованной для самостоятельной работы.

Самостоятельная учебная деятельность является необходимым условием успешного обучения. Многие профессиональные навыки, способность мыслить и обобщать, делать выводы и строить суждения, выступать и слушать других, – все это развивается в процессе самостоятельной работы студентов.

Самостоятельная работа по освоению дисциплины включает:

- самостоятельное изучение разделов;
- самоподготовку (проработку и повторение лекционного материала и материала учебников и учебных пособий);
- подготовку к лабораторным работам;
- подготовку к рубежному и итоговому контролю.

Самостоятельная учебная работа – условие успешного окончания высшего учебного заведения. Она является равноправной формой учебных занятий, наряду с лекциями, семинарами, экзаменами и зачетами, но реализуемая во внеаудиторное время.

Эффективность аудиторных занятий во многом зависит от того, насколько умело студенты организуют в ходе них свою самостоятельную учебную познавательную деятельность. Такая работа также способствует самообразованию и самовоспитанию, осуществляемому в интересах повышения профессиональных компетенций, общей эрудиции и формированию личностных качеств.

Самостоятельная работа реализуется:

1. непосредственно в процессе аудиторных занятий – на лекциях, лабораторных занятиях, при проведении рубежного контроля;
2. в контакте с преподавателем вне рамок расписания – на консультациях по учебным вопросам, при ликвидации задолженностей, при выполнении индивидуальных заданий;
3. в библиотеке, дома, в общежитии, на кафедре при выполнении студентом учебных задач.

В процессе проведения самостоятельной работы необходимо производить подбор литературных источников, научной периодической печати и т.д.

#### **4 Методические указания по итоговому контролю**

Итоговый контроль знаний по дисциплине «Программная инженерия» проводится в форме экзамена. Для подготовки к итоговому контролю знаний по дисциплине «Программная инженерия» обучающиеся используют перечень вопросов, приведенный в фонде оценочных средств. Экзамен проводится в устной форме. В экзаменационный билет включен один теоретический вопрос. На подготовку студенту отводится 20-25 минут. На дифференцированном зачете ответы обучающегося оцениваются с учетом их полноты, правильности и аргументированности с учетом шкалы оценивания.

Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе профессиональные термины, правильно обосновывает принятое решение.

Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов, владеет необходимыми навыками и приемами их выполнения.

Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.

Оценка «неудовлетворительно» выставляется студенту за отсутствие знаний по дисциплине, представления по вопросам, непонимание материала по дисциплине, наличие коммуникативных «барьеров» в общении, отсутствие ответа на предложенный вопрос.

## 5 Учебно-методическое обеспечение дисциплины

### 5.1 Основная литература

1. Золотов, С.Ю. Проектирование информационных систем : учебное пособие / С.Ю. Золотов ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Эль Контент, 2013. - 88 с. : табл., схем. - ISBN 978-5-4332-0083-8 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=208706>, коэффициент книгообеспеченности 1

2. Стасьшин, В.М. Проектирование информационных систем и баз данных : учебное пособие / В.М. Стасьшин. - Новосибирск : НГТУ, 2012. - 100 с. - ISBN 978-5-7782-2121-5 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=228774>, коэффициент книгообеспеченности 1

### 5.2 Дополнительная литература

1. Проектирование информационных систем [Текст] : учебное пособие / Г. Н. Исаев. - 2-е изд., стер. - Москва : Омега - Л, 2015. - 424 с. - (Высшее техническое образование) - ISBN 978-5-370-03507-4. (ОГТИ СПО 16), коэффициент книгообеспеченности 1

2. Абрамов, Г.В. Проектирование информационных систем : учебное пособие / Г.В. Абрамов, И.Е. Медведкова, Л.А. Коробова. - Воронеж : Воронежский государственный университет инженерных технологий, 2012. - 172 с. : ил., табл., схем. - ISBN 978-5-89448-953-7 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=141626>, коэффициент книгообеспеченности 1

3. Заика, А.А. Разработка прикладных решений для платформы 1С:Предприятие 8.2 в режиме "Управляемое приложение" / А.А. Заика. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 239 с. : ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429019>, коэффициент книгообеспеченности 1

### 5.3 Периодические издания

1. Журнал «Вестник компьютерных и информационных технологий»
2. Журнал «Информационные технологии и вычислительные системы»
3. Журнал «Стандарты и качество»
4. Журнал «Прикладная информатика»

### 5.4 Интернет-ресурсы

#### 5.4.1 Современные профессиональные базы данных и информационные справочные системы:

1. Информационная система «Единое окно доступа к образовательным ресурсам» - <http://window.edu.ru/>
2. КиберЛенинка - <https://cyberleninka.ru/>

3. Университетская информационная система Россия – [uisrussia.msu.ru](http://uisrussia.msu.ru)
4. Бесплатная база данных ГОСТ – <https://docplan.ru/>

#### 5.4.2 Тематические профессиональные базы данных и информационные справочные системы:

1. Портал искусственного интеллекта – [AIPortal](#)
2. Web-технологии – [Web-технологии](#)
3. Электронная библиотека Института прикладной математики им. М.В. Келдыша – [Электронная библиотека публикаций Института прикладной математики им. М.В. Келдыша РАН](#)

#### 5.4.3 Электронные библиотечные системы

1. ЭБС «Университетская библиотека онлайн» – <http://www.biblioclub.ru/>
2. ЭБС Znanium.com – <https://znanium.com/>

#### 5.4.4 Дополнительные Интернет-ресурсы

1. <https://www.ixbt.com> - Интернет-издание о компьютерной технике, информационных технологиях и программных продуктах. На сайте публикуются новости IT, статьи с обзорами и тестами компьютерных комплектующих и программного обеспечения.
2. <http://www.intuit.ru> – ИНТУИТ – Национальный открытый университет.
3. <http://cppstudio.com/> - Основы программирования на языках Си и C++.
4. <https://www.anti-malware.ru/> - Информационно-аналитический центр, посвященный информационной безопасности.
5. <https://developer.mozilla.org/ru/docs/Tools> — Открытые уроки по веб-технологиям и инструментам разработчика.
6. <https://frontender.info> – Электронный журнал по фронтенд-разработке
7. <https://docs.oracle.com/en/java/> - Документация по языку Java.
8. [http://citforum.ru/SE/project/arkhipenkov\\_lectures](http://citforum.ru/SE/project/arkhipenkov_lectures) – Лекции по управлению программными проектами автор А. Архипенков
9. <http://1c.ru/> - сайт фирмы разработчика серии программ "1С:Предприятие", предназначенных для автоматизации управления и учета на предприятиях различных отраслей, видов деятельности и типов финансирования.

#### 5.5 Программное обеспечение, профессиональные базы данных и информационные справочные системы современных информационных технологий

Тип программного обеспечения	Наименование	Схема лицензирования, режим доступа
Операционная система	Microsoft Windows	Подписка Enrollment for Education Solutions (EES) по государственному контракту: № 2К/17 от 02.06.2017 г.;
Текстовый редактор	Notepad++	Свободное ПО, <a href="https://notepad-plus-plus.org/">https://notepad-plus-plus.org/</a>
Интернет-браузер	Google Chrome	Бесплатное ПО, <a href="http://www.google.com/intl/ru/policies/terms/">http://www.google.com/intl/ru/policies/terms/</a>
Векторный графический редактор, редактор диаграмм и блок-схем	Microsoft Visio Standard 2007	Сертификат Microsoft Open License № 46284547 от 18.12.2009 г., академическая лицензия на рабочее место



Тип программного обеспечения	Наименование	Схема лицензирования, режим доступа
Интегрированная среда разработки программного обеспечения	Microsoft Visual Studio Professional 2008	Сертификат Microsoft Open License № 46284547 от 18.12.2009 г., академическая лицензия на рабочее место
	Embarcadero RAD Studio 2010 Professional	Образовательная лицензия по государственному контракту № 32/09 от 17.12.2009 г., сетевой конкурентный доступ
	Turbo Pascal 7.0 for DOS	Образовательная лицензия по государственному контракту № 34/10 от 10.12.2010 г., лицензия на рабочее место
	Borland C++ 3.1 for DOS	Образовательная лицензия по государственному контракту № 34/10 от 10.12.2010 г., лицензия на рабочее место
	Dev-C++	Свободное ПО, <a href="http://www.gnu.org/licenses/gpl.html">http://www.gnu.org/licenses/gpl.html</a>

## 6 Материально-техническое обеспечение дисциплины

Учебные аудитории для проведения занятий лекционного типа, семинарского типа, для проведения групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Для проведения лабораторных работ используются компьютерный класс (ауд. № 4-113, 4-116, 4-117), оборудованный средствами оргтехники, программным обеспечением, персональными компьютерами, объединенными в сеть с выходом в Интернет.

Аудитории оснащены комплектами ученической мебели, техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой, подключенной к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду Орского гуманитарно-технологического института (филиала) ОГУ (ауд. № 4-307).

Наименование помещения	Материально-техническое обеспечение
Учебные аудитории: - для проведения занятий лекционного типа, семинарского типа, - для групповых и индивидуальных консультаций; - для текущего контроля и промежуточной аттестации	Учебная мебель, классная доска, мультимедийное оборудование (проектор, экран, ноутбук с выходом в сеть «Интернет»)
Компьютерные классы № 4-113, 4-116, 4-117	Учебная мебель, компьютеры (29) с выходом в сеть «Интернет», проектор, экран, лицензионное программное обеспечение
Помещение для самостоятельной работы обучающихся, для курсового проектирования (выполнения курсовых работ)	Учебная мебель, компьютеры (3) с выходом в сеть «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Орского гуманитарно-технологического института (филиала) ОГУ, программное обеспечение

Для проведения занятий лекционного типа используются следующие наборы демонстрационного оборудования и учебно-наглядные пособия:

- презентации к курсу лекций.