

Задание

1. Выполнить практическое задание по образцу
2. Базу, созданную по образцу, выслать для проверки

С уважением, Хвастова Светлана Ивановна

!!! Если возникнут вопросы обращаться по телефону 0721389311.

Электронная почта: xvsviv@rambler.ru

Учебная практика на тему «Разработка пользовательского приложения при помощи средств системы управления базами данных Microsoft Access»

Пример разработки приложения для работы с базами данных СУБД Access

Одним из наиболее распространенных сценариев в разработке приложений является отображение данных из БД на форме **Приложения Windows Forms**. Поэтому с учетом процессов, которые используются при работе с наборами данных, предлагаемых Microsoft (см. рисунки 1-2), и с учетом того, что представленные задачи учебные, в примерах будем рассматривать следующие процессы:

1. Создание БД.
2. Разработка приложения Windows:
 - создание проекта;
 - создание интерфейса пользователей;
 - создание подключения к данным БД и отображение их на форме;
 - редактирование, проверка и сохранение данных в приложении;
 - разработка программного кода приложения.
3. Выполнение приложения.

В базу данных с именем **DataBase2** поступают сведения о студентах и оценках, полученных этими студентами по трем предметам на двух контрольных. Предположим, что данные хранятся в двух таблицах, одна из

которых называется **СТУДЕНТЫ**, а другая **ОЦЕНКИ**. Проект должен реализовать расчет сведений о тех студентах, которые получили пятерок на второй контрольной больше, чем на первой.

Документ должен иметь заголовок, информацию о времени его создания, дате проведения контрольных, а также информацию обо всех студентах, повысивших успеваемость, с указанием групп и специальностей.

1. Создание базы данных DataBase2.

Для построения таблиц **СТУДЕНТЫ** и **ОЦЕНКИ** необходимо создать их структуры, вид которых приведен на рисунках 1 и 2.

Заполнение таблиц данными можно выполнить в режиме таблицы. После заполнения таблиц **СТУДЕНТЫ** и **ОЦЕНКИ** они могут принимать вид, приведенный на рисунках 3 и 4.

Таблицы базы данных будут связаны по схеме «один-ко-многим», как на рисунке 5.

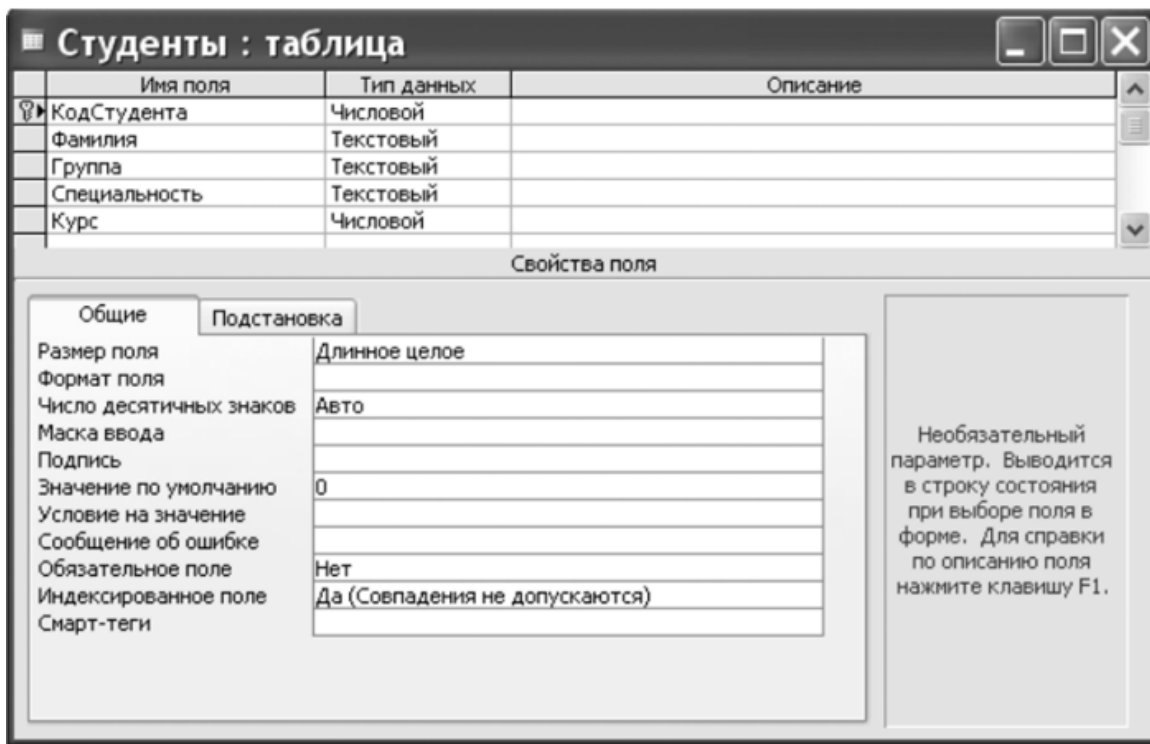


Рисунок 1 – Структура таблицы студенты

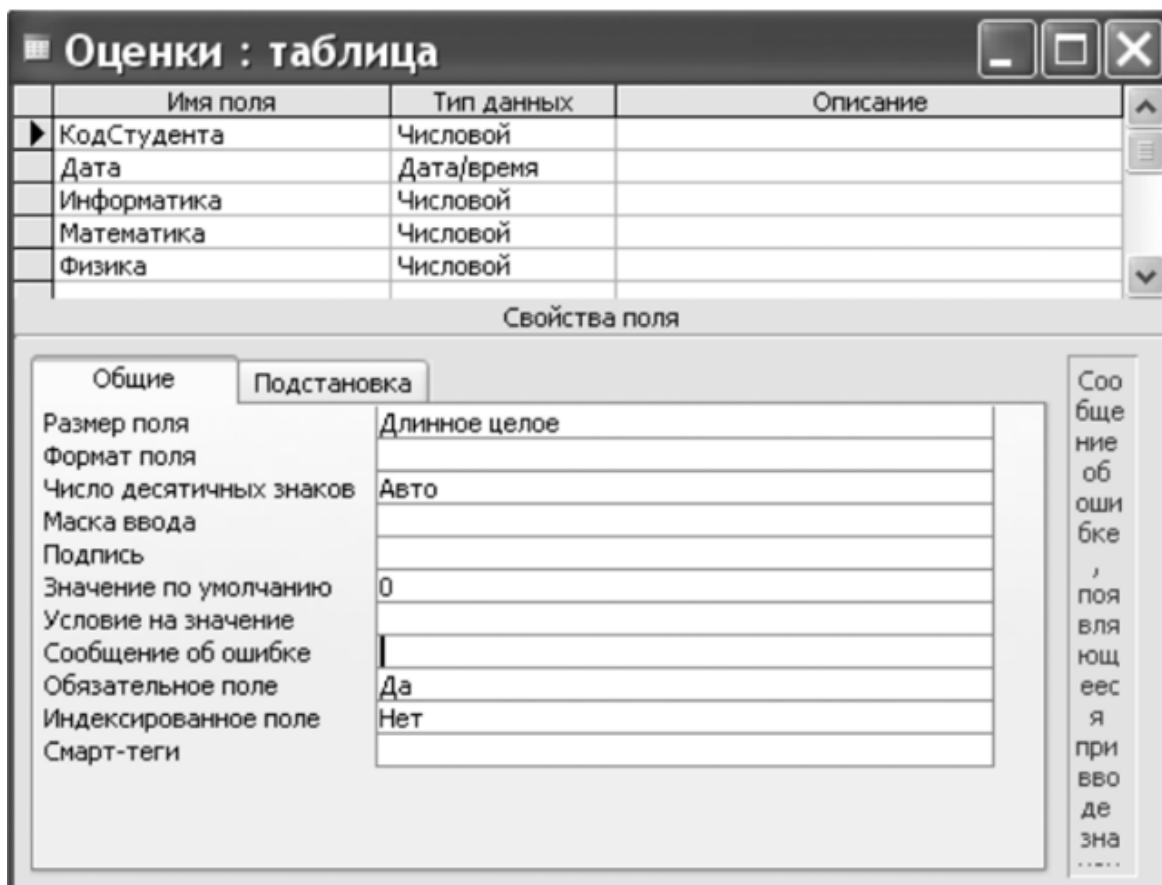


Рисунок 2 – Структура таблицы ОЦЕНКИ

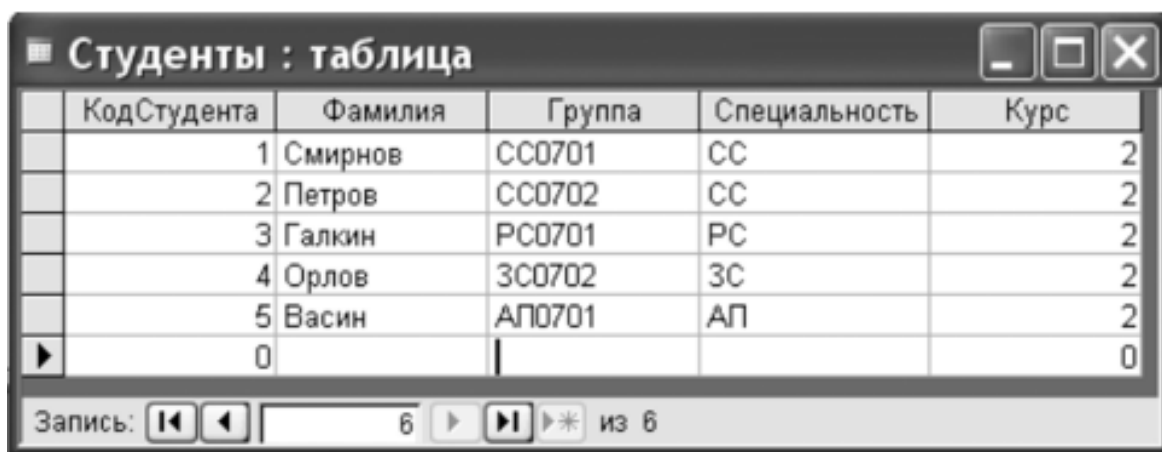


Рисунок 3 – Данные таблицы студенты

КодСтудент	Дата	Информати	Математик	Физика
	23.02.2009	4	5	5
1	05.03.2009	5	4	3
2	23.02.2009	3	2	2
2	05.03.2009	3	4	4
3	23.02.2009	4	5	3
3	05.03.2009	5	2	4
4	23.02.2009	5	4	4
4	05.02.2009	4	5	4
5	23.03.2009	3	2	3
5	05.03.2009	5	5	5

Рисунок 4 – Данные таблицы ОЦЕНКИ



Рисунок 5 – Схема данных базы DataBase2

2. Разработка Приложения Windows для работы с БД DataBase2

2.1. Создание нового проекта.

Создать проект Оценки студентов.

2.2. Создание интерфейса пользователей.

Создать интерфейс пользователей из двух форм Form1 и Form2, которые изображены на рисунках 6 и 7.

2.3. Подключение набора данных приложения к базе данных DataBase2 СУБД Access и отображение их на форме Form1.

Процедура подключения приложения VS к данным БД DataBase2 может быть упрощена за счет использования Мастера Настройки источников данных.

Form1

Таблица "Студенты"

	КодСтудента	Фамилия	Группа	Специальность	Курс
*					

Переход к форме2

Таблица "Оценки"

	КодСтудента	Дата	Информатика	Математика	Физика
*					

Выход

Рисунок 6 – Форма Form1 проекта Оценки студентов

Form2

Справка об успеваемости

выдана 12 марта 2009 г.

Контрольные проводились

Список студентов повысивших успеваемость

Возврат в первую форму

Рисунок 7 – Форма Form2 проекта Оценки студентов

После завершения работы **Мастера** данные в окне **Источники данных** становятся доступными для перетаскивания на форму.

Для создания набора данных при помощи **Мастера настройки источника данных** необходимо выполнить следующее:

– В элементе Главного меню **Проект** выполнить команду **Добавить новый источник данных**. В появившемся окне **Мастер настройки источника данных** выбрать **Источник данных** для приложения – **База данных** и нажать кнопку **Далее**.

– В окне **Мастер настройки источника данных** выбрать тип источника – **Набор данных** и нажать **Далее**. Мастер отобразит форму – **Выбор подключения базы данных**, которая помогает установить соединение с базой данных, создав связывающую строку. Нажать **Создать подключение**.

– В появившемся окне **Добавить подключение** ввести путь к файлу базы данных или щелкнуть кнопку **Обзор**, чтобы найти этот файл.

– Нажать кнопку **Проверить подключение**. **Visual Studio** попытается открыть указанный файл базы данных, используя связывающую строку, созданную мастером. Если база данных имеет правильный формат, то будет выдано сообщение **Проверка подключения выполнена**.

– Закрывать окно сообщения, а затем закрыть окно **Добавить подключение**. Мастер отобразит предупреждение, указывающее, что была выбрана новая локальная база данных, и спросит, следует ли скопировать базу данных в папку проекта. Это сообщение появляется только при установке первого соединения с файлом локальной базы данных. При повторном выполнении этого упражнения данного сообщения не будет. Чтобы избежать создания дополнительной копии базы данных, щелкните на **Нет**.

– В **Мастере настройки источника данных** включить опцию **Да**, сохранить подключение и нажать кнопку **Далее**.

Будет выдан запрос на выбор подмножества объектов базы данных, которые будут использоваться в данном проекте.

Выберите соответствующие таблицы.

– Чтобы добавить функциональность в приложение, т. е. отобразить таблицы базы данных на форме, необходимо в окне **Источники данных** выделить таблицу **СТУДЕНТЫ** и перетащить ее на форму. Становятся видны названия полей. Это происходит благодаря тому, что подключается элемент управления **DataGridView**, позволяющий иметь на форме элемент в виде таблицы с заголовками, делая тем самым удобным зрительное восприятие таблицы БД, а также навигационную панель (вверху формы) **BindingNavigator**, имеющую кнопки для перемещения по записям, добавления, удаления и сохранения записей в наборе **DataSet**.

Для отображения на форме таблицы **ОЦЕНКИ** необходимо проделать аналогичные шаги.

Первая форма будет иметь вид, как на рисунке 6.

Перетаскивание элементов из окна **Источники данных** в форму автоматически добавляет нужный код для заполнения набора данных в обработчик событий **Form1_Load()** (см. рисунок 8).

```
Private Sub Form1_Load(ByVal sender As System.Object, _  
                      ByVal e As System.EventArgs) _  
                      Handles MyBase.Load  
    'TODO: Данная строка кода позволяет загрузить данные  
    'в таблицу "Database2DataSet.Оценки".  
    'При необходимости она может быть перемещена или удалена.  
    Me.ОценкиTableAdapter.Fill(Me.Database2DataSet.Оценки)  
End Sub
```

Рисунок 8 – Код обработчика событий **Form1_Load ()**

3. Редактирование, проверка и сохранение данных в приложении

После заполнения набора данных обычно производится добавление, редактирование или удаление данных перед их отправкой обратно источнику данных, другому процессу или приложению. Поскольку каждая запись в наборе данных представляется объектом **DataRow**, изменения набора данных выполняются посредством работы с отдельными строками.

В Приложении Windows Forms архитектура привязки к данным занимается отправкой изменений из элементов управления с привязкой к данным в набор данных, поэтому не требуется в явном виде обновлять набор данных при помощи собственного кода.

Наборы данных поддерживают множество версий строк данных для нахождения изначальных записей в источнике данных. Перед выполнением обновления источника данных, возможно, потребуется проверить отдельные строки.

Изменение данных в **DataSet** представляет собой процесс управления фактическими данными в отдельных объектах, которые составляют набор данных. Изменение данных в таблицах данных подобно изменению данных в таблице любой базы данных – процесс может включать вставку, обновление и удаление записей таблицы.

Кроме изменения фактических данных также можно запрашивать **DataTable** для возвращения определенных строк данных, например, отдельных строк, определенных версий строк (исходные и предложенные), только измененных строк и строк, которые содержат ошибки.

4. Разработка программного кода приложения.

Реализация модуля **Module1** представлена на рисунке 9.

```
Module Module1
  Public Sub vivod(ByVal t As String, ByVal B As TextBox)
    B.Text = t
  End Sub
  Public Sub way(ByRef t As String, ByRef d As DataRowCollection, _
    ByVal mark As DataRowCollection)
    Dim i, j, k, p, n As Integer
    n = d.Count
    For i = 0 To 2 * (n - 1) Step 2
      k = 0 : p = 0
      For j = 2 To 4
        If mark(i)(j) = 5 Then k = k + 1
        If mark(i + 1)(j) = 5 Then p = p + 1
      Next
      If p > k Then
        t = t & d(i \ 2)(1) & " группа " & d(i \ 2)(2) & _
          " специальность: " & d(i \ 2)(3) & vbCrLf
      End If
    Next
  End Sub
End Module
```

Рисунок 9 – Код модуля **Module1**

С первой формой будет связан код, приведенный на рисунке 10.

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        Me.Hide()
        Form2.Show() 'Переход ко второй форме
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button2.Click
        End
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: Данная строка кода позволяет загрузить данные
        'в таблицу "Database2DataSet.Оценки".
        'При необходимости она может быть перемещена или удалена.
        Me.ОценкиTableAdapter.Fill(Me.Database2DataSet.Оценки)
    End Sub
End Class
```

Рисунок 10 – Код первой формы Form1

Со второй формой будет связан код, приведенный на рисунке 11.

```
Public Class Form2
    Public t As String
    Private Sub Form2_Load(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles MyBase.Load
        Dim d As DataRowCollection
        Dim mark As DataRowCollection
        d = Form1.Database2DataSet.Студенты.Rows
        mark = Form1.Database2DataSet.Оценки.Rows
        TextBox2.Text = "Контрольные проводились: " & vbCrLf & _
            "Первая : " & mark(0)(1) & vbCrLf
        TextBox2.Text = TextBox2.Text & "Вторая: " & mark(1)(1)
        t = ""
        way(t, d, mark)
        vivod(t, TextBox3)
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles Button1.Click
        Me.Hide() : Form1.Show()
    End Sub
End Class
```

Рисунок 11 – Код второй формы Form2

Текущая дата в документе этой формы определяется с помощью элемента DateTimePicker, который выбирается из панели элементов.

Процедура way() реализует алгоритм, позволяющий выводить на экран документы о студентах, которые повысили свою успеваемость на второй контрольной по сравнению с первой.

5. Выполнение приложения.

Результат выполнения программы представлены на рисунках 12 и 13.

КодСтудента	Фамилия	Группа	Специальность	Курс
1	Смиронов	СС0701	СС	2
2	Петров	СС0702	СС	2
3	Галкин	РС0701	РС	2
4	Орлов	ЭС0702	ЭС	2
5	Васин	АП0701	АП	2

КодСтудента	Дата	Информатика	М атематика	Физика
1	23.02.2009	4	5	5
1	05.03.2009	5	4	3
2	23.02.2009	3	2	2
2	05.03.2009	3	4	4
3	23.02.2009	4	5	3
3	05.03.2009	5	2	4
4	23.02.2009	5	4	4
4	05.02.2009	4	5	4
5	23.03.2009	3	2	3

Рисунок 12 – Результат на форме Form1

Справка об успеваемости

выдана 11 апреля 2009 г.

Контрольные проводились

Контрольные проводились:
Первая: 23.02.2009
Вторая: 05.03.2009

Список студентов повысивших успеваемость

Васин группа АП0701 специальность: АП

Рисунок 13 – Результат на форме Form2