



Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

Кафедра экономико-математических методов и информационных технологий

БАЗЫ ДАННЫХ

**Методические указания
к лабораторной работе № 2 для студентов направления подготовки
080500 «Бизнес-информатика»**

Ульяновск
2017

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

Лабораторная работа №2 (8 часов) Лабораторная работа №2. Разработка информационной системы для работы с базой данных (MS Access, Visual Basic)

Цель работы: Приобретение навыков программирования элементов баз данных с использованием языка программирования Visual Basic.

Создание таблицы базы данных

Краткая теория

Access хранит все таблицы базы данных в одном файле. Прежде, чем приступить к созданию таблиц базы данных, необходимо создать файл БД. Тип файла БД имеет расширение .mdb. Это расширение является зарегистрированным в Windows и связывается с программой Access.

Создание таблицы БД состоит из двух этапов: определения структуры (состава и имен полей, типа данных каждого поля, размера поля, ключей и т.д.) и заполнения таблицы данными.

В данной работе рассматривается три способа создания БД:

- в режиме таблицы;
- с помощью мастера таблиц;
- в режиме конструктора таблиц.


Порядок выполнения работы:

1. Создайте новую таблицу в режиме таблицы. Такой режим позволяет пользователю создать таблицу, не определяя предварительно ее структуру. После выбора этого режима сразу открывается пустая таблица, в которую можно ввести данные. При сохранении Access автоматически присвоит соответствующий тип данных каждому полю, т. е. создаст структуру таблицы.

Для выполнения задания выполните следующие действия:

- Загрузите Microsoft Access и в окне диалога **Создание** дважды щелкните на значке шаблона **Новая база данных**.
- В окне диалога **Файл новой БД** укажите имя файла БД и щелкните на кнопке **Создать**. Выберите вариант создания **Таблицы**. В результате на экране появится таблица, содержащая 20 столбцов и 30 строк.
- Переименуйте стандартные имена полей в соответствии с заданием. Для этого дважды щелкните на заголовке соответствующего поля (клетка должна выделиться цветом) и введите новое название.
- Заполните таблицу пятью произвольными записями и покажите результат преподавателю.
- Выйдите из режима создания таблицы без сохранения результатов.

2. Создайте новую таблицу БД, используя мастер таблиц. В этом режиме автоматически создается новая таблица по одному из шаблонов. Предлагается более 40 образцов таблиц. Каждая таблица шаблона содержит набор полей, из которых выбираются те, которые

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

соответствуют по типу данных полям результирующей таблицы. Включаемые в таблицу поля при необходимости могут быть переименованы.

Для создания таблицы БД с помощью Мастера

- В окне БД в режиме **Таблица** щелкните на ярлычке **Создать** и выберите вариант создания **Мастер таблиц**.

- Ознакомьтесь с набором таблиц делового и личного применения.

- В списке **"Образцы таблиц"** выделите таблицу, которая больше подходит по назначению к заданию, и переместите нужные вам поля в список **"Поля новой таблицы"**. Чтобы переименовать любое из добавленных полей, выделите его и воспользуйтесь кнопкой **Переименовать поле**. Тип данных при этом сохраняется. После завершения формирования списка полей, щелкните по кнопке **Далее**.

- В текстовом поле следующего окна задайте имя вашей новой таблицы и выберите автоматический способ задания ключевого поля. Щелкните по кнопке **Далее**.

- Выберите режим **"Непосредственный ввод данных в таблицу"** или **"Ввод данных в таблицу с помощью формы, создаваемой мастером"**. Щелкните по кнопке **Готово**.

- Введите в БД несколько записей и покажите результаты работы преподавателю.

- Выйдите из таблицы без сохранения данных.

3. Создайте новую таблицу в режиме конструктора. В этом режиме создание таблицы начинается с определения ее структуры. Режим конструктора позволяет разработчику произвольно задавать параметры всех элементов структуры таблицы.

Для работы в режиме конструктора

- В окне БД щелкните на вкладке **Таблицы**, а затем на кнопке **Создать**. В окне диалога **Новая таблица** выберите вариант создания **Конструктор**.


- В столбце **Имя поля** введите название нового поля, нажмите клавишу <Таб> и выберите для него тип данных из списка столбца **Тип данных**. Для поля **"Пол"** в списке столбца **Тип данных** выберите **Мастер подстановок**. Мастер строит для данного поля список значений на основе введенных пользователем фиксированных значений или на основе полей другой таблицы с данными для подстановки. Для поля **"Военнообязан"** выберите логический тип данных.

- На вкладке **Общие** укажите размер поля (число символов), формат поля и т. д. Для поля **"Телефон"** создайте маску ввода. Для добавления пропущенных полей щелкните по кнопке **Добавить строки** на панели инструментов конструктора таблиц. Для ускорения ввода данных в таблицу отдельным полям можно назначить значение по умолчанию. Это значение будет появляться в соответствующем поле каждой новой записи и его можно заменить другим значением. Для поля **"Средний балл"** задайте ограничение на вводимые значения, при нарушении которых запрещается ввод и выводится текст, заданный свойством **Сообщение об ошибке**.

- Выйдите из режима конструктора с сохранением структуры БД, без указания ключевых полей.

- Откройте созданную БД и заполните ее двадцатью произвольными записями. Вводимые в поля таблицы значения должны соответствовать типам данных и накладываемым на них ограничениям. Проверьте действие ограничений на вводимые значения. Результаты работы покажите преподавателю.

- Завершите работу с Microsoft Access.

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

Задание для создания таблицы

Список полей БД "Студент": Номер, ФИО, Пол, Дата рождения, Домашний адрес, Факультет, Телефон_деканата, ФИО_декана, Курс, Группа, Количество_человек_в_группе, Средний балл, Размер_стипендии, Военнообязан.

Основы программирования на Visual Basic

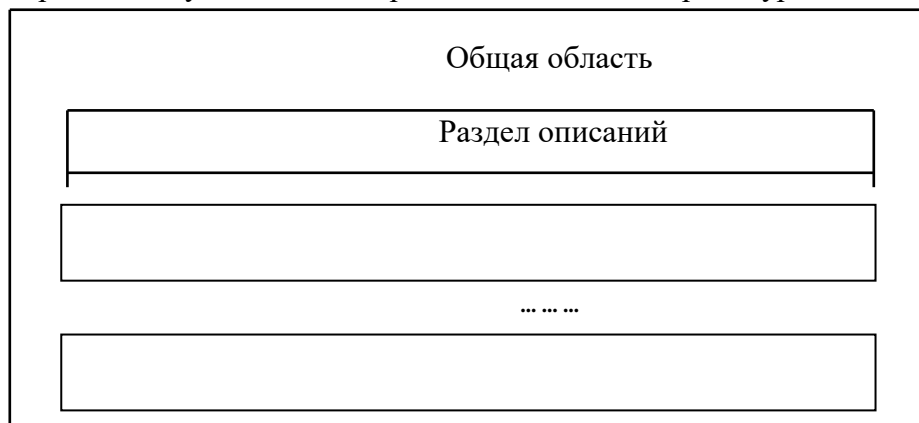
Основная теория:

Основой программы на Visual Basic является процедуры, состоящие из инструкций, которые выполняют необходимые действия и вычисления. Процедуры хранятся в модулях. Модули подразделяются на стандартные и модули форм и отчетов.

Стандартные модули представлены в окне базы данных. Вызов этих процедур может осуществляться из процедур обработки событий, процедур других стандартных модулей, макросов и просто из выражений.

Модули форм и отчетов создается автоматически при создании формы или отчета и является частью их описания.

Стандартный модуль состоит из раздела описаний и процедур:



В стандартном модуле могут храниться обычные процедуры типа Sub и Function. Sub-процедура - это подпрограмма, которая не возвращает значения. Function - процедура-функция, которая возвращает значение. При объявлении требуется соблюдать правила описания процедур.


Формат описания для процедуры Sub

```
Sub имя ([ аргумент])
    инструкции
end sub
```

Необязательный аргумент позволяет передавать в процедуру требуемые значения. Процедура **Sub** вызывается инструкцией: **Call имя**. Синтаксис описания функции очень похож на синтаксис процедуры Sub, но есть и отличия:

```
Function имя ([ аргумент]) [ as тип]
    инструкции
    имя = выражение
end function
```

Параметр **as тип** позволяет явно задать тип данных, который возвращает функция. Если он опущен, то по умолчанию возвращается значение типа **VARIANT**. Если в разделе описания

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

модуля помещена инструкция **Option Explicit**, то неявное описание запрещено. Функцию можно вызвать из выражения.

Константы и переменные могут быть описаны как в разделе описания модуля, тогда они доступны во всех процедурах модуля, так и внутри процедур, тогда они действуют только в ее пределах.

Формат описания переменных:

Dim Имя переменной [**as** тип данных]

Основные типы данных, которые используются при описании: **Boolean, Byte, Integer, Long, Currency, Single, Double, Date, String, Variant, Object**. Если тип данных не указан, по умолчанию переменная получает тип **Variant**. Данные этого типа интерпретируются в зависимости от операции, в которой они используются, т.е. тип может меняться.

Формат описания констант:

Const Имя константы = значение

Константы может определять пользователь и можно использовать стандартные константы, определенные системой. Первые две буквы в имени встроенных констант являются префиксом. Константы из библиотеки VBA имеют префикс "vb". Константы библиотеки Access имеют префикс "as".

В выражениях могут использоваться стандартные функции языка. Перечислим некоторые из них: математические: Sin(x), Sqr(x), Exp(x), Log(x), Tan(x), Int(x) и пр.; преобразования: Str(s), Chr(x), Val(s); системные функции: InputBox- функция ввода данных через системное окно, MsgBox - функция выдачи сообщений в системное окно. Эти функции удобны для организации ввода - вывода в процессе работы программы.

Формат функции ввода:

Переменная=**InputBox**(Приглашение[, заголовок][, начальное значение])

Функция InputBox возвращает значение типа String. Например:

Dim c as string

Dim g as integer

c=InputBox("Введи наименование товара:")

g=Val(InputBox("Введи его цену:",0))

Формат функции вывода сообщений:

[Переменная=]**MsgBox**(Текст[,опция][,Заголовок])

Опция OP определяет внешний вид окна. Значение OP=OP1 + OP2

OP1-определяет вид пиктограммы выводимой в окне. Значения этой опции приведена в следующей таблице:


Константа	Значение	Описание
VbCritical	16	Используется значок <i>Критическое сообщение</i>
VbQuestion	32	Используется значок <i>Предупреждающий запрос</i>
VbExclamation	48	Используется значок <i>Предупреждение</i>
VbInformation	64	Используется значок <i>Информационное сообщение</i>

Например:

x=MsgBox("Возможна ошибка!",VbExclamation)

или

x=MsgBox("Возможна ошибка!",48)

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

OP2 - опция определяет набор кнопок выводимых в окне. Значения этой опции приводится в следующей таблице:

Константа	Значение	Описание
VbOKOnly	0	Отображается только кнопка ОК
VbOKCancel	1	Отображаются кнопки ОК и <i>Отмена</i>
VbAbortRetryIgnore	2	Отображаются кнопки <i>Прервать</i> , <i>Повторить</i> и <i>Пропустить</i>
VbYesNoCancel	3	Отображаются кнопки <i>Да</i> , <i>Нет</i> и <i>Отмена</i>
VbYesNo	4	Отображаются кнопки <i>Да</i> , <i>Нет</i>
VbRetryCancel	5	Отображаются кнопки <i>Повторить</i> и <i>отмена</i>

Например:

```
x=Msgbox("Функция="+str(f),VbOKOnly,"Результаты расчета")
```

```
x=MsgBox("Будете еще вводить?", VbYesNo+VbQuestion, "Запрос")
```

или

```
x=MsgBox("Будете еще вводить?..36","Запрос")
```

Функция MsgBox возвращает значение, которое зависит от того, какая кнопка была нажата в окне. Значения, возвращаемые функцией приведены в следующей таблице:

Константа	Значение	Описание
VbOk	1	Ок
VbCancel	2	Отмена
VbAbort	3	Прервать
VbRetry	4	Повторить
VbIgnore	5	Пропустить
VbYes	6	Да
VbNo	7	Нет

Например:

```
x=MsgBox("Распечатать документ?..36)
```

```
if x=6 then Call PrintRep
```

Каждая инструкция в процедуре записывается с новой строки. Чтобы записать несколько инструкций в одной строке их надо разделять символом двоеточие. Если инструкция слишком длинна, то в конце строки вводят символ продолжения строки - подчеркивание (_) - ф затем нажимают клавишу <Enter> и продолжают инструкцию на следующей строке.

В языке Visual Basic Имеется целый набор инструкций, управляющих порядком выполнения команд. Рассмотрим некоторые из них.


1) Инструкция If ...Then ... Else

Формат:

```
if условие Then
    [инструкции]
```

```
[else
    инструкции]
```

```
End if
```

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

Например, разработаем процедуру вычисления функции $f=y/c$, где $y=5$, а c - вводится с клавиатуры. Если будет введено $c=0$, должно выводиться сообщение.

```

Sub prim()
Dim f,c As Single
Dim y As Integer
y=5
c=Val(InputBox("Введи C:"))
if c=0 Then
    MsgBox("Функция не существует!")
else
    f=y/c
    MsgBox("функция="+str(f))
end if
end sub

```

2) Инструкция Select Case

Формат:

```

Select Case выражение
    Case списокВыражений-1
        инструкции
    ...
    Case списокВыражений-n
        инструкции
    [Case else
        инструкции]
End Select

```

End Select

Например, необходимо определить значение скидки, обработав значение переменной UnOrder следующим образом:

```

Select Case UnOrder
    Case 1
        Discount=0
    Case 2, 3
        Discount=0.1
    Case 4 To 6
        Discount=0.2
    Case Is>6
        Discount=0.5
End Select

```

3) Цикл For ...Next

Формат:

```


For счетчик=начало To конец [Step шаг]
    инструкции

```

Next

Счетчик - любая числовая переменная.

Например, вычислим функцию $f=a*x^2$, где $a=3$, а переменная $x=[1,5]$, шаг=1

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

```

Sub prim2()
Dim a,x,f As Integer
a=3
For x=1 To 5
    f=a*x^2
    MsgBox("X="+str(x)+" F="+str(f))
Next
End Sub

```

4) Цикл While ... Wend

Формат:

```

While условие
    инструкции

```

Wend

Реализуем предыдущий пример, но таблица выводится по-другому:

Tabvivoda="" ' Переменная, в которой формируется таблица для вывода

x=1

```

While x<=5

```

```

    f=a*x^2

```

```

    Tabvivoda=Tabvivoda+"X="+str(x)+" F="+str(f)+Chr(13)

```

```

    x=x+1

```

Wend

```

R=MsgBox(Tabvivoda, 0, "ТАБЛИЦА РАСЧЕТОВ")

```

5) Цикл Do ... Loop

Эта инструкция имеет четыре варианта синтаксиса:

■ Синтаксис 1:

```

Do While условие нахождения в цикле
    инструкции

```

```

Loop

```

■ Синтаксис 2:

```

Do Until условие выхода из цикла
    инструкции

```

```

Loop

```

■ Синтаксис 3:

```

Do
    инструкции
Loop While условие нахождения в цикле

```

■ Синтаксис 4:

```

Do
    инструкции
Loop Until условие выхода из цикла

```

Инструкция применяется аналогично While ...Wend. Например, вычислим сумму значений последовательности, вводимой с клавиатуры:

```


Sub Prim3()

```

```

Dim a,s As Single ' a- элемент последовательности,_

```


Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

```

s- вычисляемая сумма
Dim rez As Integer
s=0
Do
a=Val(InputBox("Введи элемент последовательности:"))
s=s+a
rez=MsgBox("Будешь ещё вводить?", VbYesNo, "Нажми кнопку")
Loop Until rez=VbNo
rez=MsgBox("Сумма="+Str(s) , 0 , "Результаты расчета")
End Sub


```

Порядок выполнения:

1. Откройте базу данных.
2. Создайте в модуле процедуру **Sub** для вычисления таблицы значений функции $F=a|x^3|+a^2x$ для $a=3$, x изменяется на интервале от -4 до 4 с шагом 2 . Для этого:
 - в окне базы данных перейти в закладку **Модули** и щелкнуть кнопку **Создать**. Access откроет окно проектирования модуля.
 - Вызовите команду *процедура* из меню **Вставка**. В появившемся диалоговом окне нажмите селективную кнопку **Подпрограмма** и введите имя создаваемой процедуры. Введите текст программы для решения поставленной задачи.
 - проведите компиляцию загруженного модуля. Если при компиляции будут выявлены ошибки, то их надо исправить.
 - После успешной компиляции выполните созданную процедуру и просмотрите результаты.
3. Подключите созданную процедуру к кнопке в форме. Для этого:
 - перейдите в закладку **Формы** и создайте в режиме **Конструктор** новую форму.
 - В ней создайте кнопку. В окне **создания кнопки** не выбирайте действий, связанных с ее нажатием, щелкните кнопку **Отмена**.
 - В свойствах для созданной кнопки перейдите в закладку **События** и для опции **Нажатие кнопки** выберите значение **Процедура обработки событий**.
 - В текст процедуры обработки событий вставьте вызов процедуры Sub.
 - проверьте действие кнопки.
4. Скорректируйте созданную процедуру для расчета трех таблиц значений функции **f** при разных параметрах $a=-2,4,7$. Для этого организуйте внешний цикл ввода параметра **a**, отладьте скорректированную процедуру. Рекомендуется вывод таблиц организовать в окно **Msgbox** в виде таблиц целиком.
5. Создайте в модуле функцию для обработки последовательности вводимой с клавиатуры. Задание выберите в соответствии с вариантом.
6. В созданную ранее форму вставьте поле и организуйте вывод результата созданной функции в это поле.

Варианты задания

1. Определить максимальный элемент среди отрицательных.
2. Вычислить среднее арифметическое отрицательных чисел.
3. Определить количество чисел, значение которых кратно 3.
4. Вычислить сумму положительных четных чисел.
5. Определить минимальный элемент среди положительных.

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

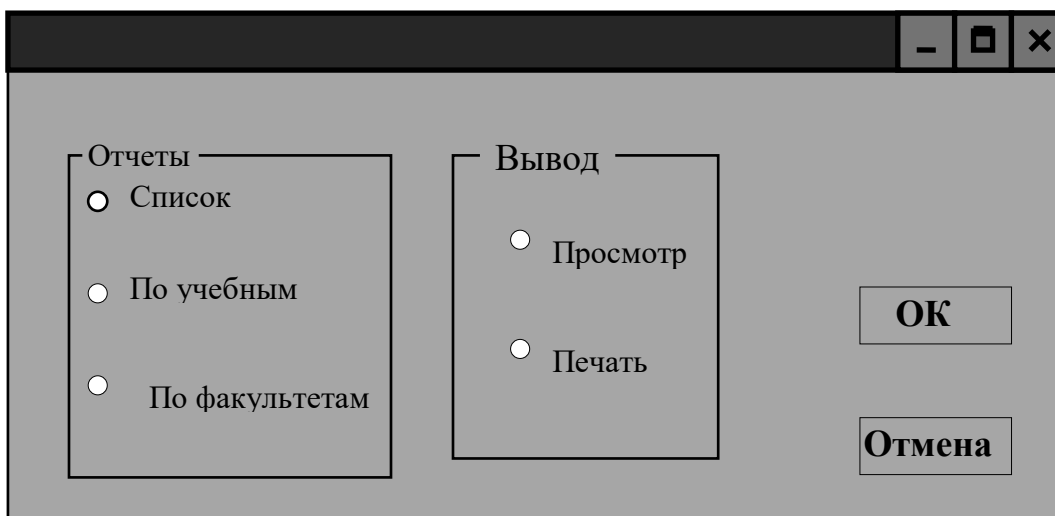
6. Определить количество отрицательных чисел.
7. Определить среднее арифметическое положительных чисел.
8. Определить сколько четных чисел в последовательности.
9. Определить сумму максимального и минимального элементов последовательности.
10. Вычислить сумму чисел, значение которых кратно 5.

Разработка средств вывода отчетов


Порядок выполнения

1. Откройте базу данных.
2. В разрабатываемых автоматизированных системах возникает необходимость просматривать и выводить на печать отчеты различного содержания. Используя группы переключателей в форме можно с легкостью создать программу, которая позволяет выбирать отчеты и направление их вывода. Для этого:
 - Создайте форму выбора отчета:
 - 1) Для этого в окне базы данных раскройте вкладку **Формы** и нажмите кнопку *Создать*. Войдите в режим конструктора, не задавая таблицу или запрос, являющийся источником данных для формы.
 - 2) Создайте группу переключателей, используя мастера групп, т.е. сначала создается группа, а в ней переключатели для каждого отчета. Введите имена отчетов в качестве надписей переключателей.
 - 3) Создайте другую группу переключателей для печати и просмотра (можно еще и для вывода в файл).
 - 4) Создайте кнопки **ОК** и **Отмена**.

Возможный внешний вид полученной в результате вышеприведенных действий формы показан ниже:



The screenshot shows a window with a title bar containing minimize, maximize, and close buttons. The main area contains two groups of radio buttons and two buttons. The first group is titled "Отчеты" and contains three radio buttons labeled "Список", "По учебным", and "По факультетам". The second group is titled "Вывод" and contains two radio buttons labeled "Просмотр" and "Печать". To the right of these groups are two buttons labeled "ОК" and "Отмена".

Министерство образования и науки РФ Ульяновский государственный университет	Форма	
Ф-Методические указания		

- Создайте процедуру обработки события *Нажатие кнопки* для кнопки **ОК**. Для этого в режиме конструктора перейдите к тексту программы, например, вызвав команда **Вид / Программа**. Процедуру нажатия кнопки **ОК** скорректируйте приблизительно следующим образом:

```
Private Sub Кнопка10_Click()
Dim Nreport, Mode As Integer
Dim strdocname As String
Nreport = Группа2.Value
Select Case Nreport
Case 1
    strdocname = "отчет1"
Case 2
    strdocname = "отчет2"
Case 3
    strdocname = "отчет3"
End Select
Mode= Группа10
Select Case Mode
Case 1    'просмотр
    DoCmd.OpenReport strdocname, acPreview
Case 2    'печать
    DoCmd.OpenReport strdocname, acNormal
End Select
End Sub
```

При создании программы обратите внимание на номера групп в форме (можно было предварительно имя группам задать идентификаторами).

3. Сохранить программы в форме и проверить работу этой формы.