

ПРАКТИЧЕСКАЯ РАБОТА

Разработка модулей программного средства

Содержание: название, формулировка темы и цели, время выполнения, необходимое обеспечение, теоретическая часть, практическая часть, технология выполнения, описание заданий, рефлексия самоконтроля, контрольные вопросы

Время выполнения: 2 ч.

Материально-техническое обеспечение: персональный компьютер (системный блок, монитор, клавиатура, мышь)

Программное обеспечение: ОС Windows, интегрированная среда разработки, офисные программы Microsoft Office.

Учебно-методическое обеспечение: конспект лекций, описание работы, встроенная справочная система ОС Windows и сред разработки.

Порядок выполнения практической работы: изучить теоретический материал, выполнить задания, составить отчёт с выводами о проделанной работе, ответить на контрольные вопросы, сдать выполненную работу и отчёт.

Требования к отчёту: отчёт должен быть оформлен в соответствии с «ГОСТ 19.XXX ЕСПД».

Цель: приобретение навыков разработки модулей программного средства.

Теоретическая часть

Парадигмы программирования

- Процедурное (императивное) программирование;
- Декларативное программирование;
- Структурное программирование;
- Структурное программирование;
- Функциональное программирование;
- Логическое программирование;
- Объектно-ориентированное программирование;
- Модульное программирование;
- Обобщённое программирование.

Модульное программирование

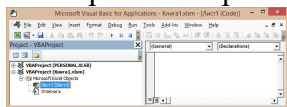
Модульное программирование - это организация программы как совокупности небольших независимых блоков, называемых модулями, структура и поведение которых подчиняются определённым правилам. Использование модульного программирования позволяет упростить тестирование программы и обнаружение ошибок. Аппаратно-зависимые подзадачи могут быть строго отделены от других подзадач, что улучшает мобильность создаваемых программ. Модуль - функционально законченный фрагмент программы. Во многих языках (но далеко не обязательно) оформляется в виде отдельного файла с исходным кодом или поименованной непрерывной её части. Некоторые языки предусматривают объединение модулей в пакеты. Языки модульного программирования: Assembler, Кобол, RPG, ПЛ/1, Ада, D, F (англ.), Фортран, Haskell, Blitz BASIC, OCaml, Паскаль, ML, Модула-2, Оберон, Компонентный Паскаль, Zonnon, Erlang, Perl, Python и Ruby.

Visual Basic for Applications

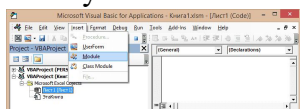
Visual Basic for Applications (VBA) - немного упрощённая реализация языка программирования Visual Basic, встроенная в линейку продуктов Microsoft Office (включая версии для Mac OS), а также во многие другие программные пакеты, такие как AutoCAD, SolidWorks, CorelDRAW, WordPerfect и ESRI ArcGIS. VBA покрывает и расширяет функциональность ранее использовавшихся специализированных макроязыков, таких как WordBasic. VBA является интерпретируемым языком. Как и следует из его названия, VBA близок к Visual Basic. VBA, будучи языком, построенным на COM, позволяет использовать все доступные в операционной системе COM объекты и компоненты ActiveX. По сути, возможно создание приложения на основе Microsoft Word VBA, использующего только средства Corel Draw.

VBA-Excel: создание макроса

Создайте новую книгу Excel и сохраните ее как книгу с поддержкой макросов с расширением .xlsm. В старых версиях Excel по 2003 год – как обычную книгу с расширением .xls. Нажмите сочетание клавиш «левая_клавиша_Alt+F11», которое откроет редактор VBA. С правой клавишей Alt такой фокус не пройдет. Также, в редактор VBA можно перейти по ссылке «Visual Basic» из панели инструментов «Разработчик» на ленте быстрого доступа. Если вкладки «Разработчик» на ленте нет, ее следует добавить в настройках параметров Excel.

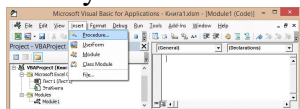


В левой части редактора VBA расположен проводник проекта, в котором отображены все открытые книги Excel. Верхней строки, как на изображении, у вас скорее всего не будет, так как это – личная книга макросов. Справа расположен модуль, в который записываются процедуры (подпрограммы) с кодом VBA. На изображении открыт модуль листа, мы же далее создадим стандартный программный модуль. Нажмите кнопку «Module» во вкладке «Insert» главного меню. То же подменю откроется при нажатии на вторую кнопку после значка Excel на панели инструментов.

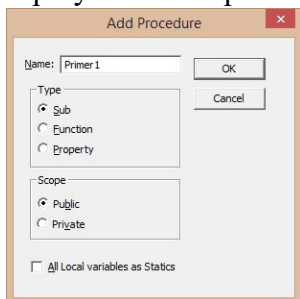


После нажатия кнопки «Module» вы увидите ссылку на него, появившуюся в проводнике слева.

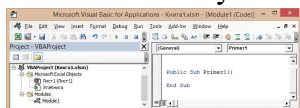
Добавляем на стандартный модуль шаблон процедуры – строки ее начала и завершения, между которыми мы и будем писать свою первую программу (процедуру, подпрограмму). Откройте стандартный модуль двойным кликом по его ссылке в проводнике. Поместите в него курсор и нажмите кнопку «Procedure...» во вкладке «Insert» главного меню. Та же ссылка будет доступна при нажатии на вторую кнопку после значка Excel на панели инструментов.



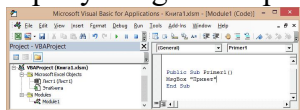
В результате откроется окно добавления шаблона процедуры (Sub).



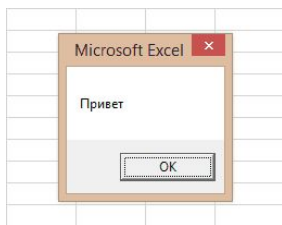
Наберите в поле «Name» имя процедуры: «Primer1», или скопируйте его отсюда и вставьте в поле «Name». Нажмите кнопку «OK», чтобы добавить в модуль первую и последнюю строки процедуры.



Имя процедуры может быть написано, как на латинице, так и на кириллице, может содержать цифры и знак подчеркивания. Оно обязательно должно начинаться с буквы и не содержать пробелы, вместо которых следует использовать знак подчеркивания. Вставьте внутрь шаблона процедуры следующую строку: MsgBox "Привет".



Функция MsgBox выводит информационное сообщение с указанным текстом. В нашем примере – это «Привет». Проверьте, что курсор находится внутри процедуры, и запустите ее, нажав клавишу «F5». А также, запустить процедуру на выполнение можно, нажав на треугольник (на изображении под пунктом меню «Debug») или на кнопку «Run Sub/UserForm» во вкладке «Run» главного меню редактора VBA Excel.



Если Вы увидели такое сообщение, то Вы написали программу.

Работа с переменными

Чтобы использовать в процедуре переменные, их необходимо объявить с помощью ключевого слова «Dim». Если при объявлении переменных не указать типы данных, они смогут принимать любые доступные в VBA Excel значения. Комментарии в тексте процедур начинаются со знака «'» (апостроф).

Пример присвоения переменным числовых значений:

```
Public Sub Primer2()
'Объявляем переменные x, y, z
Dim x, y, z
'Присваиваем значение переменной x
x = 25
'Присваиваем значение переменной y
y = 35
'Присваиваем переменной z сумму
'значений переменных x и y
z = x + y
'Выводим информационное сообщение
'со значением переменной z
MsgBox z
End Sub
```

Пример присвоения переменным строковых значений:

```
Public Sub Primer3()
'Объявляем переменные x, y, z
Dim x, y, z
'Присваиваем строку переменной x
x = "Добрый"
'Присваиваем строку переменной y
y = "день!"
'Присваиваем переменной z строку,
'состоящую из строк x и y
'с пробелом между ними
z = x & " " & y
'Выводим информационное сообщение
'со значением переменной z
MsgBox z
End Sub
```

Скопируйте примеры в стандартный модуль и запустите их на выполнение. Перейдите на активный рабочий лист Excel, чтобы увидеть результат.

Работа с диапазонами

Для обозначения диапазонов, в том числе и отдельных ячеек, в VBA Excel имеется ключевое слово «Range». Ячейке A1 на рабочем листе будет соответствовать выражение «Range("A1")» в коде VBA Excel.

Пример изменения содержимого ячеек:

```
Public Sub Primer4()
'Объявляем переменную x
Dim x
'Присваиваем значение переменной x
x = 125.61
'Присваиваем ячейке A1
'значение переменной x
Range("A1") = x
'Присваиваем значение ячейке B1
Range("B1") = 356.24
'Записываем в ячейку C1
'сумму ячеек A1 и B1
Range("C1") = Range("A1") + Range("B1")
```

End Sub

Скопируйте пример в стандартный модуль и запустите их на выполнение. Перейдите на активный рабочий лист Excel, чтобы увидеть результат.

VBA: опции

В VBA массивы могут начинаться с индекса 1, что для многих странно, поэтому в начале модулей можно прописывать:

OPTION BASE 0

Так же рекомендуется прописать:

OPTION EXPLICIT

Важным оператором является ON ERROR:

ON ERROR RESUME NEXT ' продолжает со следующей строки
ON ERROR GOTO label: ' переходит, в случае ошибки, к метке label:
ON ERROR GOTO 0 ' возвращает обычное поведение.

Часто макросы требуют долгого времени выполнения, которое можно значительно сократить. Для ускорения работы макросов нужно: отключить перерисовку объектов на экране (чтобы ничего не мигало; выключить расчёт; не обрабатывать события; включить отображение границ страниц (почему-то помогает; отключить вывод данных в статусной строке. В начале и в конце каждой ресурсоёмкой функции вызвать Prepare и Ended.

```
Public Sub Prepare()  
    Application.ScreenUpdating = False  
    Application.Calculation = xlCalculationManual  
    Application.EnableEvents = False  
    ActiveSheet.DisplayPageBreaks = False  
    Application.DisplayStatusBar = False  
    Application.DisplayAlerts = False  
End Sub  
  
Public Sub Ended()  
    Application.ScreenUpdating = True  
    Application.Calculation = xlCalculationAutomatic  
    Application.EnableEvents = True  
    ActiveSheet.DisplayPageBreaks = True  
    Application.DisplayStatusBar = True  
    Application.DisplayAlerts = True  
End Sub
```

VBA-Excel: строка подключения к базе данных MS SQL

```
Rem Строка подключения для MS SQL  
Sub Get_MSSQL_Data()  
    Dim db As ADODB.Connection  
    Dim rs As ADODB.Recordset  
    Dim sqlStr As String  
    Set rs = CreateObject("ADODB.Recordset")  
    Set db = New ADODB.Connection  
    db.Open _  
    "DRIVER={SQL Server};SERVER=SName;UID=UserName;PWD=Password;DATABASE=DBName"  
    sqlStr = "SELECT Count(*) as cnt FROM [DBName].[DB].[Table]"  
    rs.Open sqlStr, db  
    While Not rs.EOF  
        str1 = rs.Fields("cnt").Value  
        rs.MoveNext  
    Wend  
    rs.Close  
    db.Close  
End Sub  
"DRIVER={SQL Server};SERVER=SName;UID=UserName;PWD=Password;DATABASE=DBName"
```

Для того, чтобы при выполнении кода не возникало ошибки «Compile error: User-defined type not defined»:


```

r = 0
c = 0
txt = ""
Application.ScreenUpdating = False
Do Until EOF(1)
    Line Input #1, Data
    For i = 1 To Len(Data)
        Char = Mid(Data, i, 1)
        If Char = "," Then 'запятая
            ActiveCell.Offset(r, c) = txt
            c = c + 1
            txt = ""
        ElseIf i = Len(Data) Then 'конец строки
            If Char <> Chr(34) Then txt = txt & Char
            ActiveCell.Offset(r, c) = txt
            txt = ""
        ElseIf Char <> Chr(34) Then
            txt = txt & Char
        End If
    Next i
    c = 0
    r = r + 1
Loop
Close #1
Application.ScreenUpdating = True
End Sub

Function DefaultFilePath()
    DefaultFilePath = Application.DefaultFilePath
End Function

```

VBA-Access: работа с базами данных при помощи ADO

Говоря о доступе к базам данных, нельзя не упомянуть о возможностях, которые предоставляют объекты ADO (ActiveX Data Object). Можно работать с базой данных (СУБД MS Access), как с файлом, но можно выполнять обработку данных, используя так называемый источник данных. Для этого рассмотрим, что такое ODBC. ODBC (Open Database Connectivity) источники соединения с базой. Для работы с ODBC необходимо выполнить следующие действия:

- Если это необходимо, то выполнить установку ODBC драйвера для вашей СУБД. Возможно для этого понадобится установка клиентского ПО на ваш персональный компьютер (все работы выполняются с правами администратора);
- В меню «Пуск/Настройка/Панель управления/Администрирование» открыть иконку «Источники данных (ODBC)», открыть закладку «Пользовательский DSN» и нажать кнопку «Добавить»; в появившемся окне выбрать драйвер Microsoft Access Driver (в зависимости от версии может быть – *.mdb, может быть *.mdb, *.accdb); в поле имя источника данных набрать “con_weather”; в разделе «База данных» нажать кнопку выбрать и в диалоговом окне выбрать базу, например, «...\DBproba1.mdb» (может быть расширение accdb); завершив указанные выше операции, нажать кнопку ОК; перейти в режим редактирования VBA. В меню Tools\References выбрать библиотеку Microsoft ActiveX Data Objects 2.8 Library;
- Набрать и выполнить следующую программу:

```

Sub db_stud()
    Dim MyCon As New Connection
    Dim strSQL As String
    Dim rs As Recordset

    strSQL = "Select data_weather, temperature_weather from weather"
    MyCon.Open "con_weather" Set rs = New Recordset
    rs.Open      strSQL,      MyCon,      adOpenForwardOnly,      adLockReadOnly,      adCmdText
    ActiveWorkbook.Worksheets("Лист1").Activate i = 1

    Do Until rs.EOF()
        Cells(i, 1) = rs!data_weather    Cells(i, 2) = rs!temperature_weather    rs.MoveNext    i = i + 1 Loop rs.Close
    MyCon.Close

```

End Sub

Если вы работаете через ODBC источник, то существует альтернативный рассмотренному выше вариант получения информации о структуре базы данных основанный на использовании метода OpenSchema, объекта «соединение»:

Набор записей = соединение.OpenSchema (QueryType, Criteria, SchemaID)

Параметры схемы таблиц:

QueryType	Criteria
adSchemaTables	TABLE_CATALOG TABLE_SCHEMA TABLE_NAME TABLE_TYPE

Пример получения списка всех таблиц и запросов:

```
Set rs = cn.OpenSchema(adSchemaTables)
While Not rs.EOF Debug.Print rs!TABLE_NAME rs.MoveNext Wend
```

Пример получения списка только таблиц:

```
Set rs = cn.OpenSchema(adSchemaTables, Array(Empty, Empty, Empty, "Table"))
```

Для получения списка только таблиц в базе данных Pubs Microsoft SQL Server, используйте следующую команду:

```
Set rs = cn.OpenSchema(adSchemaTables, Array("Pubs", Empty, Empty, "Table"))
```

Для получения списка полей в таблице можно использовать adSchemaColumns. Параметры схемы полей таблицы:

QueryType	Criteria
adSchemaColumns	TABLE_CATALOG TABLE_SCHEMA TABLE_NAME COLUMN_NAME
Set rs = cn.OpenSchema(adSchemaColumns, Array(Empty, Empty, "Employees")) While Not rs.EOF Debug.Print rs!COLUMN_NAME rs.MoveNext Wend	

Для получения списка имен индексов в таблице adSchemaIndexes. Параметры схемы индексов:

QueryType	Criteria
adSchemaIndexes	TABLE_CATALOG TABLE_SCHEMA INDEX_NAME TYPE TABLE_NAME
Set rs = cn.OpenSchema(adSchemaIndexes, Array(Empty, Empty, Empty, Empty, "Employees")) While Not rs.EOF Debug.Print rs!INDEX_NAME rs.MoveNext Wend	

Чтобы просмотреть список индексов в таблице Authors базы данных Pubs SQL Server с использованием конструкции adSchemaIndexes, можно воспользоваться следующей записью:

```
Set rs = cn.OpenSchema(adSchemaIndexes, Array("Pubs", "dbo", Empty, Empty, "Authors"))
Ниже приведены примеры нескольких программ, работающих со структурой БД
```

```
'Открываем определенное соединение connection.
Dim cn As New ADODB.Connection
Dim rs As New ADODB.Recordset
Private Sub Command1_Click()
'Получаем информацию об определенных полях
Set rs = cn.OpenSchema(adSchemaColumns, Array("pubs", "dbo", "authors"))
While Not rs.EOF
    Debug.Print rs!COLUMN_NAME
```

```

rs.MoveNext
Wend
End Sub
Private Sub Command2_Click()
'Получаем информацию об основном ключе таблицы
Set rs = cn.OpenSchema(adSchemaPrimaryKeys,
    Array("pubs", "dbo", "authors"))
MsgBox rs!COLUMN_NAME
End Sub

Private Sub Command3_Click()
'Получаем информацию обо всех таблицах
Dim criteria(3) As Variant criteria(0) = "pubs" criteria(1) = Empty criteria(2) = Empty criteria(3) = "table"
Set rs = cn.OpenSchema(adSchemaTables, criteria)
While Not rs.EOF
    Debug.Print rs!TABLE_NAME
    rs.MoveNext
Wend
End Sub

```

Приведённый ниже пример программы позволяет получить список таблиц в базе данных. Перед решением необходимо добавить ссылку (tools → references → Microsoft ActiveX Data Objects 6):

```

Sub Test_STRUCT_table()
Dim MyCon As New Connection
MyCon.Open "testr"
Set rs = MyCon.OpenSchema(adSchemaTables)
Workbooks("Example_DB.xls").Worksheets("Table_Structure1").Activate i = 2
While Not rs.EOF
    Cells(i, 1).Value = rs!TABLE_NAME
    Cells(i, 2).Value = rs!TABLE_CATALOG
    Cells(i, 3).Value = rs!TABLE_SCHEMA    Cells(i, 4).Value = rs!TABLE_TYPE    i = i + 1    rs.MoveNext
Wend
MyCon.Close
Cells(1, 1).Value = i - 2
End Sub

```

Следующая программа показывает, как получать информацию о полях таблиц. Перед решением необходимо добавить ссылку (tools → references → Microsoft ActiveX Data Objects 6):

```

Sub Test_STRUCT_columns()
Dim MyCon As New Connection
MyCon.Open "testr"
Workbooks("Example_DB.xls").Worksheets
    ("Table_Structure1").Activate
For ii = 2 To Cells(1, 1).Value
    Set rs = MyCon.OpenSchema(adSchemaColumns,    Array(Empty, Empty, Cells(ii, 1).Text)    i = 2
    While Not rs.EOF
        Cells(ii, 5 + i).Value = rs!COLUMN_NAME    i = i + 1    rs.MoveNext
    Wend
Next ii
MyCon.Close
End Sub

```

Примеры решения задач

Пример. VBA-Access СУБД «Студент»

Задача

Разработать проект работы с БД. Проект демонстрирует взаимодействие пользователя с единственной таблицей, хранящей информацию о студентах. Интерфейс работы с базой данных реализован посредством пользовательской формы. Пользовательский интерфейс позволяет добавлять, удалять, изменять информацию в таблице, просматривать текущие записи и получать отчет о студентах с использованием поискового фильтра по «Фамилии».

Решение

Для начала в базе данных «MyDB1.mdb», находящейся в том же рабочем каталоге, что и рабочая книга с разрабатываемым проектом, создадим таблицу «stud», имеющую следующую структуру:

Имя поля	Тип данных
key_stud	Счетчик
fio_stud	Текстовый
birthday_stud	Дата/время
sex_stud	Логический
date_in_stud	Дата/время
spec_stud	Числовой

Далее переименуем рабочий лист нашей рабочей книги в «INFO_OUT». Следующим шагом создадим специальный класс работы с базой данных. Методы данного класса будут реализовывать основные функции работы с базой данных (некоторые даже не будут использоваться в данном проекте), а также функции взаимодействия с основной пользовательской формой. Для создания класса добавляем новый класс «Stud_Info2» в рабочий проект и пишем следующий код:

```
Dim myWS As Workspace
Dim fio As String
Dim bd As Date
Dim sex As Boolean
Dim date_in As Date
Dim spec As Integer
Public myDB As Database
'Методы, отвечающие за установку свойств класса
Sub set_fio(ff As String)
    If ff = "" Then
        MsgBox "Введите правильную фамилию" Else fio = ff End If
    End Sub

Sub set_birthday(dd As Date)
    If Year(dd) = 1899 Then
        MsgBox "Введите правильную дату рождения" Else bd = dd End If
    End Sub

Sub set_date_in(dd As Date)
    If Year(dd) = 1899 Then
        MsgBox "Введите правильную дату поступления" Else date_in = dd
    End If
End Sub

Sub set_sex(ss As Boolean)
    sex = ss
End Sub

Sub set_spec(sp As Integer)
    If sp <= 0 Then
        MsgBox "Введите правильное значение" Else spec = sp End If
    End Sub

'Метод проверяет корректность начальных установок
Private Function Check_Ok() As Boolean
    Check_Ok = True
    If fio = "" Then Check_Ok = False
    If Year(bd) = 1899 Then Check_Ok = False
    If Year(date_in) = 1899 Then Check_Ok = False
    If spec = 0 Then Check_Ok = False End Function

'Метод выполняет роль «конструктора» класса
Private Sub Class_Initialize() fio = "" bd = 0 sex = False date_in = 0 spec = 0 ok = False End Sub

'Метод проверяет наличие базы данных и открывает ее
Function Check_Open_DB(pathx As String, dbName As String) As Boolean

    Check_Open_DB = True
    On Error GoTo errh

    Set myWS = DBEngine.Workspaces(0)
    If Dir(pathx + dbName) <> "" Then
        Set myDB = myWS.OpenDatabase(pathx + dbName)
        If myDB.Name <> pathx + dbName Then
            Check_Open_DB = False
            MsgBox "You HAVE some problems with DATABASE"
```

```

Exit Function
End If
Else
    Check_Open_DB = False
End If
Exit Function errh:
Check_Open_DB = False
End Function
‘Метод удаляет определенную базу данных
Function Kill_Database(pathx As String, dbName As String) As
Boolean
If Dir(pathx + dbName) <> "" Then
    Kill pathx + dbName
    Kill_Database = True
Else
    MsgBox "Could not find DATABASE " & pathx + dbName
    Kill_Database = False
End If

End Function
‘Метод создает БД с определенным именем и по определённому пути
Function Create_Database(pathx As String, dbName As String) As
Boolean
If Dir(pathx + dbName) = "" Then
    Set myWS = DBEngine.Workspaces(0)
    Set myDB = myWS.CreateDatabase(pathx + dbName,
dbLangCyrillic, dbVersion40)
    Create_Database = True
Else
    MsgBox "DATABASE " & pathx + dbName & " already exists"
    Create_Database = False
End If
End Function
‘Метод проверяет наличие определенной таблицы в открытой БД
Function Check_Table(tableName As String) As Boolean
Dim td As TableDef
Check_Table = False
For Each td In myDB.TableDefs
    If tableName = td.Name Then
        Check_Table = True
        Exit Function
    End If
Next
End Function
‘Метод удаляет определенную таблицу в БД
Function Drop_table(tableName As String) As Boolean
Dim strSQL As String
If Check_Table(tableName) = True Then    strSQL = "drop table " & tableName    myDB.Execute strSQL
    Drop_table = True
Else
    MsgBox "There is no table " & tableName & " in the DATABASE"
    Drop_table = False
End If
End Function
‘Метод создает определенную таблицу в БД
Function Create_Table(tableName As String) As Boolean

Dim strSQL As String
If Check_Table(tableName) = False Then
    strSQL = "Create table " & tableName & " ([key_stud] COUNTER
CONSTRAINT key_stud PRIMARY KEY," & _
        " [fio_stud] TEXT (50), [birthday_stud] DATETIME,
[sex_stud] BIT, [date_in_stud] DATETIME, " & _
        "[spec_stud] LONG)"    myDB.Execute strSQL    Create_Table = True
Else
    MsgBox "The table " & tableName & " in the DATABASE"
    Create_Table = False
End If

```

```

End Function
'Метод проверяет наличие в БД (в таблице Stud) записи с
'определенными характеристиками
Function Check_InsDB() As Boolean
    Check_InsDB = True
    Dim strSQL As String
    Dim rst As Recordset If Check_Ok() = True Then    strSQL = "select * from stud where fio_stud like '" & fio & "' and
    birthday_stud = '" & Format(bd, "\#mm\dd\yyyy\#") & "' and sex_stud = '" & sex & "' and date_in_stud = '" &
    Format(date_in,
    "\#mm\dd\yyyy\#") & "' and spec_stud = '" & spec
    Set rst = myDB.OpenRecordset(strSQL, dbOpenSnapshot)
    If rst.RecordCount = 0 Then
        Check_InsDB = True
        'MsgBox "OK"
    Else
        Check_InsDB = False
        MsgBox "Information ALREADY in the dataBASE"    End If    rst.Close
    Else
        Check_InsDB = False
    End If

End Function

End Function
'Метод реализует добавление информации в БД (в таблицу Stud)
Function Exec_InsDB() As Boolean
    Dim strSQL As String If Check_InsDB() = True Then    strSQL = "insert into stud (fio_stud, birthday_stud,sex_stud,
    date_in_stud, spec_stud) values ('" + fio & "', " & Format(bd, "\#mm\ /dd\ /yyyy\#") & "', " & sex & "', " & Format(date_in,
    "\#mm\dd\yyyy\#") & "', " & spec & "')"    myDB.Execute strSQL
    Exec_InsDB = True
Else
    Exec_InsDB = False
End If
End Function

'Метод реализует добавление информации в БД (в таблицу Stud)
Function Exec_UpdDB(key As Integer) As Boolean
    Dim strSQL As String If Check_InsDB() = True Then
        strSQL = "update stud set fio_stud = '" & fio & "', birthday_stud = '" & Format(bd, "\#mm\dd\yyyy\#") & _
sex_stud = '" & sex & "', date_in_stud = '" & Format(date_in, "\#mm\dd\yyyy\#") & "', spec_stud = '" & spec & "' where
key_stud = '" & key
        MsgBox strSQL    myDB.Execute strSQL
        Exec_UpdDB = True
    Else
        Exec_UpdDB = False
    End If
End Function

'Метод реализует удаление информации из БД (таблицы Stud)
Function Exec_DelDB(key As Integer) As Boolean
    Dim strSQL As String If key <> 0 Then
        strSQL = "Delete from stud where key_stud = '" & key    myDB.Execute strSQL
        Exec_DelDB = True
    Else
        MsgBox "Could not delete key=" & key
        Exec_DelDB = False
    End If
End Function

```

Далее создаем пользовательские формы для взаимодействия с базой данных. Это форма UF_DATABASE (Рис.14.2), форма UF_REPORT (Рис.14.3), форма UF_CALENDAR (Рис.14.4).

Ниже изображения каждой формы приведен текст обработчиков событий элементов управления, связанных с соответствующей формой. Отметим, что данные формы практически не содержат программного кода, т.к. они в данном проекте предназначены только для отображения, ввода и изменения текущей информации, а вся обработка и взаимодействие с базой данных происходит в процедурах модуля (посредством выполнения методов разработанного класса). В форме только устанавливается флаг, соответствующий нажатой кнопке, и определяющий действие, которое надо выполнить в текущий момент.

Public flagtb As Boolean

Public flagAll As Integer

Private Sub CheckBox1_Click()

If CheckBox1.Value = True Then

 CheckBox1.Caption = "Мужской"

Else

 CheckBox1.Caption = "Женский"

End If

End Sub

Private Sub CommandButton1_Click()

 flagAll = 4

 Hide

End Sub

Private Sub CommandButton2_Click()

 flagAll = 1

 Hide

End Sub

Private Sub CommandButton3_Click()

flagAll = 3 Hide

End Sub

Private Sub CommandButton4_Click()

flagAll = 2 Hide

End Sub

Private Sub CommandButton5_Click()

flagAll = 5 Hide

End Sub

Private Sub CommandButton6_Click()

flagAll = 6 Hide

End Sub

Private Sub CommandButton7_Click()

UF_REPORT.Show

Hide

End Sub

Private Sub TextBox2_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single) flagtb = False

UF_CALENDAR.Caption = "Выбор даты рождения"

If TextBox2.Text <> "" Then

 UF_CALENDAR.Calendar1.Value = TextBox2.Text

End If

UF_CALENDAR.Show

End Sub

Private Sub TextBox3_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single) flagtb = True

UF_CALENDAR.Caption = "Выбор даты поступления"

If TextBox3.Text <> "" Then

 UF_CALENDAR.Calendar1.Value = TextBox3.Text

End If

UF_CALENDAR.Show

End Sub

Private Sub UserForm_Initialize()

 CheckBox1.Value = False CheckBox1.Caption = "Женский"

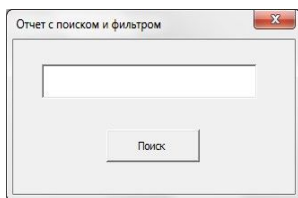
 flagtb = False For i = 1 To 5

 ComboBox1.AddItem (Str(i))

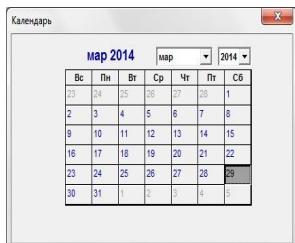
 Next i

 ComboBox1.ListIndex = 0

```
flagAll = 0
End Sub
```



```
Private Sub CommandButton1_Click()
UF_DATABASE.flagAll = 7
Hide
End Sub
```



```
Private Sub Calendar1_DblClick()
If UF_DATABASE.flagtb = False Then
UF_DATABASE.TextBox2.Value = Str(Calendar1.Value) Else
UF_DATABASE.TextBox3.Value = Str(Calendar1.Value)
End If
Hide
End Sub
```

Ниже приведена основная процедура модуля, которая практически выполняет все взаимодействия с пользователем и базой данных:

```
Sub proba_x2()

Dim dbName As String, pathx As String, tableName As String
Dim wsheetName As String
Dim studi As New Stud_Info2
Dim rst As Recordset

wsheetName = "INFO_OUT" dbName = "MyDB1.mdb" tableName = "stud"
pathx = ActiveWorkbook.path + "\"

If studi.Check_Open_DB(pathx, dbName) = False Then
MsgBox "You have some problems with DATABASE " & dbName & vbCrLf
& vbCrLf & "      Good Bye"
Exit Sub
End If

If studi.Check_Table(tableName) = False Then
MsgBox "You have some problems with DATABASE " & dbName & vbCrLf
& vbCrLf & "      Good Bye"
Exit Sub
End If

Set rst = studi.myDB.OpenRecordset("Select * from stud", dbOpenSnapshot)
If rst.RecordCount <> 0 Then rst.MoveLast rst.MoveFirst Call UF_SET(rst)
End If

UF_DATABASE.flagAll = 0
Do Until UF_DATABASE.flagAll = 1
*** До тех пор пока не нажата кнопка выход

UF_DATABASE.Show (1)
```

```

Select Case (UF_DATABASE.flagAll)

Case 4:
*** Добавление информации в Таблицу

    Call CLASS_SET(studi)
    If studi.Exec_InsDB() = True Then      strBuf = UF_DATABASE.TextBox1 & " " &
UF_DATABASE.TextBox2
        MsgBox strBuf & vbLf & " Information has been ADDED into the DATABASE"    End If
        Set rst = studi.myDB.OpenRecordset("Select * from stud", dbOpenSnapshot)    rst.MoveLast    rst.MoveFirst
    Call UF_SET(rst)
Case 5:
*** Изменение информации в Таблице

    Call CLASS_SET(studi)
    If studi.Exec_UpdDB(rst.Fields("key_stud").Value) = True Then
        strBuf = UF_DATABASE.TextBox1 & " " &
UF_DATABASE.TextBox2
        MsgBox strBuf & vbLf & " Information has been UPDATED in the DATABASE"    End If
        Set rst = studi.myDB.OpenRecordset("Select * from stud", dbOpenSnapshot)    rst.MoveLast    rst.MoveFirst
    Call UF_SET(rst)
Case 2:
*** Переход к предыдущей записи

    rst.MovePrevious    If rst.BOF = True Then    rst.MoveLast    End If
    Call UF_SET(rst)
Case 3:
*** Переход к следующей записи

    rst.MoveNext
    If rst.EOF = True Then    rst.MoveFirst    End If
    Call UF_SET(rst)
Case 6:
*** Удаление информации из Таблицы

    If studi.Exec_DelDB(rst.Fields("key_stud").Value) = True Then
        strBuf = UF_DATABASE.TextBox1 & " " &
UF_DATABASE.TextBox2
        MsgBox strBuf & vbLf & " Information has been DELETED from the DATABASE"
    End If
    Set rst = studi.myDB.OpenRecordset("Select * from stud", dbOpenSnapshot)    rst.MoveLast    rst.MoveFirst
    Call UF_SET(rst)
Case 7:
*** Получение отчета по запросу ("с параметром")

    ActiveWorkbook.Worksheets(wsSheetName).Activate    strSQL = "Select * from stud where fio_stud like '" &
UF_REPORT.TextBox1.Value & _
    "' order by fio_stud"
    Set rst = studi.myDB.OpenRecordset(strSQL, dbOpenSnapshot)

    fldc = rst.Fields.Count    Range("A:F").ClearContents
    Range("A1").Value = "Ключ"
    Range("B1").Value = "Фамилия И.О."
    Range("C1").Value = "Дата Рождения"
    Range("D1").Value = "Пол"
    Range("E1").Value = "Дата поступления"
    Range("F1").Value = "Специальность"
    i = 2
    Do Until rst.EOF
        For j = 0 To fldc - 1
            If j = 3 Then
                If rst.Fields(j) = True Then
                    Cells(i, j + 1).Value = "М"                Else
                    Cells(i, j + 1).Value = "Ж"
                End If
            Else
                Cells(i, j + 1).Value = rst.Fields(j)
            End If
        Next j
        i = i + 1
    Loop

```

```

        End If      Next j      i = i + 1      rst.MoveNext
    Loop
    If rst.RecordCount <> 0 Then      rst.MoveFirst      Call UF_SET(rst)
    Else
        Call UF_CLR
    End If
End Select
Loop

MsgBox "GOOD bye"

End Sub
Sub UF_SET(rst As Recordset)
    '*** Заполнение полей пользовательской формы данными из рекордсета

    With UF_DATABASE
        .Caption = "Всего " & rst.RecordCount & " записей"
        .TextBox1.Value = rst.Fields("fio_stud")
        .TextBox2.Value = Str(rst.Fields("birthday_stud"))
        .CheckBox1.Value = rst.Fields("sex_stud")
        .TextBox3.Value = Str(rst.Fields("date_in_stud"))
        .ComboBox1.Value = rst.Fields("spec_stud")
        .TextBox4.Value = rst.Fields("key_stud")
    End With
End Sub

Sub UF_CLR()
    '*** Заполнение полей пользовательской формы

    With UF_DATABASE
        .Caption = "Всего 0 записей"
        .TextBox1.Value = ""
        .TextBox2.Value = ""
        .CheckBox1.Value = False
        .TextBox3.Value = ""
        .ComboBox1.Value = 0
        .TextBox4.Value = 0
    End With
End Sub

Sub CLASS_SET(ByRef studii As Stud_Info2)
    '*** Заполнение свойств класса данными из пользовательской формы

    With UF_DATABASE
        studii.set_fio (.TextBox1.Value)      studii.set_birthday (.TextBox2.Value)      studii.set_date_in (.TextBox3.Value)
        studii.set_sex (.CheckBox1.Value)      studii.set_spec (.ComboBox1.Value)
    End With
End Sub

```

Для запуска проекта надо запустить процедуру proba_x2().

Пример. VBA-Access: СУБД «Поставки»

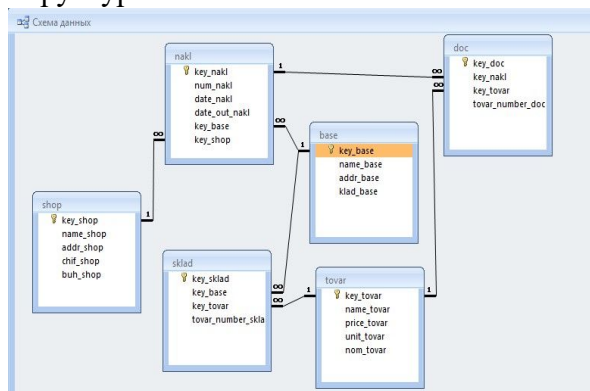
Задача

Данный проект несколько сложнее чем предыдущий. Он предполагает написание программы, которая взаимодействует с базой данных через объект ADO, посредством ODBC источника. Данная программа должна подключиться к базе данных, проверить существуют ли в ней исходные таблицы. Если таблицы существуют, то необходимо сначала удалить связи между отношениями, затем удалить сами таблицы (т. к. существующие таблицы могут носить имена таблиц проекта, но их структура может быть для нас не подходящей). Далее программа должна: создать нужные нам таблицы заново; создать необходимые связи между отношениями. загрузить информацию о товарах, базах хранения товаров и товарных накладных в созданную нами базу данных из текстовых файлов. Базовая часть проекта должна обеспечивать пользователю возможность получать информацию о товарах, находящихся на складах товарных баз и о накладных, по которым данные товары отгружались. Отчеты о товарах должны

выводится в Лист MS Excel, документ Word, список пользовательской формы, текстовый файл по выбору пользователя. Также пользователь должен иметь возможность выводить в окно списка (ListBox) пользовательской формы информацию о накладных. Структура таблиц, которые необходимо создать:

<table><tr><th colspan="2">base</th></tr><tr><th>Имя поля</th><th>Тип данных</th></tr><tr><td>key_base</td><td>Счетчик</td></tr><tr><td>name_base</td><td>Текстовый</td></tr><tr><td>addr_base</td><td>Текстовый</td></tr><tr><td>klad_base</td><td>Текстовый</td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	base		Имя поля	Тип данных	key_base	Счетчик	name_base	Текстовый	addr_base	Текстовый	klad_base	Текстовый																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
base																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
Имя поля	Тип данных																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
key_base	Счетчик																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
name_base	Текстовый																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
addr_base	Текстовый																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
klad_base	Текстовый																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							

Структура связей:



Пользовательская форма:

Решение

Ниже каждой формы приведен код, обработки событий, связанных с элементами управления, размещенными на форме.

```

Private Sub CommandButton1_Click()
    Call Module12.show_sklad
End Sub
Private Sub CommandButton2_Click()
    Unload UF_SKLAD
End Sub
Private Sub OptionButton1_Click()
    Module12.radio_but = "win" End Sub
Private Sub OptionButton2_Click()
    Module12.radio_but = "wsheet" End Sub
Private Sub OptionButton3_Click()
    Module12.radio_but = "word" End Sub
Private Sub OptionButton4_Click()
    Module12.radio_but = "file" End Sub
Private Sub OptionButton5_Click()
    Module12.radio_but_obj = "nakl" End Sub
Private Sub OptionButton6_Click()
    Module12.radio_but_obj = "sklad" End Sub
Private Sub UserForm_Initialize()
    Call Module12.clr_wsheets

```


End Sub

Процедуры модуля будут взаимодействовать с формой через глобальные переменные, также для работы будет нужен пользовательский тип данных. Все эти переменные и структура должны быть объявлены в начале модуля:

```
Public radio_but As String
Public radio_but_obj As String

Type nakl_struct key As Integer num As Integer datex As Date date_outx As Date klad As String buh As String
shop As String base As String
End Type
```

Далее приводим код удаления старых таблиц и добавления новых. Для удобства работы и отладки программы все действия будем протоколировать в текстовый файл:

```
Sub db3004_creation()
Dim mCON As New Connection
Dim strSQL As String
Dim hFileOut As Long
Dim path_x As String Dim fOutName As String fOutName = "ProtocolCreation.txt" path_x = ActiveWorkbook.path +
"\\"

mCON.Open ("db_practic2")
'Проверка и удаление таблиц
' **** Check (drop) TABLES ****
'Открываем текстовый файл для протоколирования действий
hFileOut = FreeFile
If Dir(path_x + fOutName) = "" Then
    Open path_x + fOutName For Output Access Write As #hFileOut
Else
    Open path_x + fOutName For Append As #hFileOut
End If
Print #hFileOut,vbLf & "*START WORKING at " & Now() & "*" & vbLf On Error GoTo err1
'Сначала удаляем связи
strSQL = "alter table nakl drop constraint key_base" mCON.Execute strSQL
Print #hFileOut, strSQL & " -- " & Now()

strSQL = "alter table nakl drop constraint key_shop" mCON.Execute strSQL
Print #hFileOut, strSQL & " -- " & Now()

strSQL = "alter table doc drop constraint key_nakl" mCON.Execute strSQL
Print #hFileOut, strSQL & " -- " & Now()

strSQL = "alter table doc drop constraint key_tovar" mCON.Execute strSQL
Print #hFileOut, strSQL & " -- " & Now()

strSQL = "alter table sklad drop constraint key_base_sklad" mCON.Execute strSQL
Print #hFileOut, strSQL & " -- " & Now()

strSQL = "alter table sklad drop constraint key_tovar_sklad" mCON.Execute strSQL
Print #hFileOut, strSQL & " -- " & Now()
'Открываем схему таблиц, для удаления только тех таблиц, которые в схеме
Set rs = mCON.OpenSchema(adSchemaTables)
While Not rs.EOF
    Debug.Print rs!table_name

    strSQL = ""
    Select Case rs!table_name
        Case "shop"
            strSQL = "drop table shop"
        Case "base"
            strSQL = "drop table base"
        Case "tovar"
            strSQL = "drop table tovar"
        Case "nakl"
            strSQL = "drop table nakl"
        Case "doc"
            strSQL = "drop table doc"
        Case "sklad"
            strSQL = "drop table sklad"
    End Select
```

```

If strSQL <> "" Then      mCON.Execute strSQL
    Print #hFileOut, strSQL & " -- " & Now()
    End If      rs.MoveNext
Wend

' ***** create TABLES *****
'Создаем нужные нам таблицы
strSQL = "CREATE TABLE shop ([key_shop] COUNTER CONSTRAINT key_shop PRIMARY KEY, " & _
"name_shop] TEXT (50), [addr_shop] TEXT (50), [chif_shop] TEXT
(50), [buh_shop] TEXT (50))"
mCON.Execute strSQL MsgBox "Table shop created"
Print #hFileOut, "Table shop created " & " -- " & Now()

strSQL = "CREATE TABLE base ([key_base] COUNTER CONSTRAINT key_base PRIMARY KEY, " & _
"name_base] TEXT (50), [addr_base] TEXT (50), [klad_base] TEXT
(50))"
mCON.Execute strSQL MsgBox "Table base created"
Print #hFileOut, "Table base created " & " -- " & Now()

strSQL = "CREATE TABLE tovar ([key_tovar] COUNTER CONSTRAINT key_tovar PRIMARY KEY, " & _
"name_tovar] TEXT (50), [price_tovar] CURRENCY, [unit_tovar]
TEXT (50), [nom_tovar] LONG)"

mCON.Execute strSQL
MsgBox "Table tovar created "
Print #hFileOut, "Table tovar created " & " -- " & Now()

strSQL = "CREATE TABLE nakl ([key_nakl] COUNTER CONSTRAINT key_nakl PRIMARY KEY, " & _
"[num_nakl] LONG, [date_nakl] DATETIME, [date_out_nakl]
DATETIME, [key_base] LONG, [key_shop] LONG)"

mCON.Execute strSQL
MsgBox "Table nakl created "
Print #hFileOut, "Table nakl created " & " -- " & Now()

strSQL = "CREATE TABLE doc ([key_doc] COUNTER CONSTRAINT key_doc
PRIMARY KEY, " & _
"[key_nakl] LONG, [key_tovar] LONG, [tovar_number_doc] LONG )"

mCON.Execute strSQL
MsgBox "Table doc created"
Print #hFileOut, "Table doc created " & " -- " & Now()

strSQL = "CREATE TABLE sklad ([key_sklad] COUNTER CONSTRAINT key_sklad PRIMARY KEY, " & _
"[key_base] LONG, [key_tovar] LONG, [tovar_number_sklad] LONG)"

mCON.Execute strSQL
MsgBox "Table sklad created"
Print #hFileOut, "Table sklad created " & " -- " & Now()
'Создаем связи между таблицами
strSQL = "ALTER TABLE nakl ADD CONSTRAINT key_base FOREIGN KEY
(key_base) REFERENCES base (key_base)" mCON.Execute strSQL
MsgBox "CONSTRAINT nakl <-> base added"
Print #hFileOut, "CONSTRAINT nakl <-> base added"

strSQL = "ALTER TABLE nakl ADD CONSTRAINT key_shop FOREIGN KEY
(key_shop) REFERENCES shop (key_shop)" mCON.Execute strSQL
MsgBox "CONSTRAINT nakl <-> shop added"
Print #hFileOut, "CONSTRAINT nakl <-> shop added " & " -- " &
Now()
strSQL = "ALTER TABLE doc ADD CONSTRAINT key_nakl FOREIGN KEY
(key_nakl) REFERENCES nakl (key_nakl)" mCON.Execute strSQL
MsgBox "CONSTRAINT doc <-> nakl added"
Print #hFileOut, "CONSTRAINT doc <-> nakl added " & " -- " &
Now()

```

```

strSQL = "ALTER TABLE doc ADD CONSTRAINT key_tovar FOREIGN KEY
(key_tovar) REFERENCES tovar (key_tovar)" mCON.Execute strSQL
MsgBox "CONSTRAINT doc <-> tovar added"
Print #hFileOut, "CONSTRAINT doc <-> tovar added " & " -- " &
Now()
strSQL = "ALTER TABLE sklad ADD CONSTRAINT key_base_sklad FOREIGN KEY (key_base) REFERENCES
base (key_base)" mCON.Execute strSQL
MsgBox "CONSTRAINT sklad <-> base added"
Print #hFileOut, "CONSTRAINT sklad <-> base added " & " -- " &
Now()
strSQL = "ALTER TABLE sklad ADD CONSTRAINT key_tovar_sklad FOREIGN KEY (key_tovar) REFERENCES
tovar (key_tovar)" mCON.Execute strSQL
MsgBox "CONSTRAINT sklad <-> tovar added"
Print #hFileOut, "CONSTRAINT sklad <-> tovar added " & " -- " &
Now()

Print #hFileOut, vbLf & " *** WORKING FINISH at " & " -- " & Now
& " *** " Close (hFileOut) mCON.Close
Exit Sub
'Обработка ошибок (вывод описания ошибки) и продолжении работы
err1:
MsgBox Err.Description
Print #hFileOut, Err.Description & " -- " & Now()
Resume Next

End Sub

```

Далее напишем процедуру, которая загружает данные в созданные таблицы из текстовых файлов. Эта же процедура может пригодится, если есть необходимость подгружать данные, полученные из внешних источников

```

Sub db3004_load()
Dim mCON As New Connection
Dim strSQL As String Dim mas() As String
Dim strBuf As String
Dim fName As String
Dim hFile As Long
Dim path_x As String
Dim strMSG As String
'Получение пути к рабочему каталогу
Dim fOutName As String
path_x = ActiveWorkbook.path + "\"

fOutName = "ProtocolLoading.txt" mCON.Open ("db_practic2")

hFileOut = FreeFile
'Открываем файл протокола для записи
If Dir(path_x + fOutName) = "" Then
    Open path_x + fOutName For Output Access Write As #hFileOut
Else
    Open path_x + fOutName For Append As #hFileOut
End If
Print #hFileOut, vbLf & " *START WORKING at " & Now() & " * " & vbLf "Загрузка информации о магазинах из
текстового файла
'1 ***** fName = "shop.txt"
If Dir(path_x + fName) = "" Then
    strMSG = "File " & fName & " not found!" MsgBox strMSG
    Print #hFileOut, strMSG & " -- " & Now()
    Close (hFileOut)
    Exit Sub
End If

hFile = FreeFile
Open path_x + fName For Input Access Read As #hFile i = 0 k = 0
Do Until EOF(hFile)
    Line Input #hFile, strBuf    mas = Split(strBuf, ";") If UBound(mas) = 3 Then    k = k + 1
    strSQL = "select * from shop where name_shop like '" &

```

```

Trim(mas(0)) & "" Set rst = New Recordset
rst.Open strSQL, mCON, adOpenForwardOnly, adLockReadOnly, adCmdText
If rst.EOF = True Then
    strSQL = "INSERT into shop (name_shop, addr_shop, chif_shop, buh_shop) values ('" & Trim(mas(0)) & "', '" & _
Trim(mas(1)) & "', '" & Trim(mas(2)) & "', '" & Trim(mas(3)) & _
"')"
    mCON.Execute strSQL i = i + 1 End If rst.Close
End If
Loop Close (hFile)
strMSG = i & " records from " & k & " added into shop"
MsgBox strMSG
Print #hFileOut, strMSG & " -- " & Now()
'Загрузка информации о базах из текстового файла
'2 *****

fName = "base.txt"
If Dir(path_x + fName) = "" Then
    strMSG = "File " & fName & " not found!"
    MsgBox strMSG
    Print #hFileOut, strMSG & " -- " & Now()
    Close (hFileOut)
    Exit Sub
End If
hFile = FreeFile
Open path_x + fName For Input Access Read As #hFile i = 0 k = 0
Do Until EOF(hFile) Line Input #hFile, strBuf mas = Split(strBuf, ";") If UBound(mas) = 2 Then k = k + 1
    strSQL = "select * from base where name_base like '" &
Trim(mas(0)) & "" Set rst = New Recordset
rst.Open strSQL, mCON, adOpenForwardOnly, adLockReadOnly, adCmdText
If rst.EOF = True Then
    strSQL = "INSERT into base (name_base, addr_base, klad_base) values ('" & Trim(mas(0)) & "', '" & _
Trim(mas(1)) & "', '" & Trim(mas(2)) & "')" mCON.Execute strSQL
    i = i + 1 End If rst.Close
End If
Loop Close (hFile)
strMSG = i & " records from " & k & " added into base"
MsgBox strMSG
Print #hFileOut, strMSG & " -- " & Now()
'Загрузка информации о товарах из текстового файла
'3 ***** fName = "tovar.txt"

If Dir(path_x + fName) = "" Then
    strMSG = "File " & fName & " not found!"
    MsgBox strMSG
    Print #hFileOut, strMSG & " -- " & Now()
    Close (hFileOut)
    Exit Sub
End If
hFile = FreeFile
Open path_x + fName For Input Access Read As #hFile i = 0 k = 0
Do Until EOF(hFile)
    Line Input #hFile, strBuf mas = Split(strBuf, ";") If UBound(mas) = 3 Then k = k + 1
    strSQL = "select * from tovar where name_tovar like '" &
Trim(mas(0)) & "' and nom_tovar = " & Trim(mas(3)) Set rst = New Recordset
rst.Open strSQL, mCON, adOpenForwardOnly, adLockReadOnly, adCmdText
If rst.EOF = True Then
    strSQL = "INSERT into tovar (name_tovar, price_tovar, unit_tovar, nom_tovar) values ('" & Trim(mas(0)) & "', '" & _
Trim(mas(1)) & "', '" & Trim(mas(2)) & "', '" & Trim(mas(3)) & "')" mCON.Execute strSQL
    i = i + 1 End If rst.Close
End If
Loop
Close (hFile)

strMSG = i & " records from " & k & " added into tovar"
MsgBox strMSG
Print #hFileOut, strMSG & " -- " & Now()
'Загрузка информации о накладных (заголовки) из текстового файла

```

```

' 4 ***** fName = "nakl.txt"

If Dir(path_x + fName) = "" Then
    strMSG = "File " & fName & " not found!"
    MsgBox strMSG
    Print #hFileOut, strMSG & " -- " & Now
    Close (hFileOut)
    Exit Sub
End If
hFile = FreeFile
Open path_x + fName For Input Access Read As #hFile i = 0 k = 0
Do Until EOF(hFile)
    Line Input #hFile, strBuf    mas = Split(strBuf, ";") If UBound(mas) = 4 Then    k = k + 1
    strSQL = "select * from nakl where num_nakl = " &
Trim(mas(0)) & " and date_nakl = " & Format(Trim(mas(3)),
"\#mm\dd\yyyy\#")

    Set rst = New Recordset
    rst.Open strSQL, mCON, adOpenForwardOnly, adLockReadOnly, adCmdText
    If rst.EOF = True Then
        strSQL = "INSERT into nakl (num_nakl, date_nakl,
date_out_nakl, key_base, key_shop) values (" & Trim(mas(0)) & ", " & Format(Trim(mas(1)), "\#mm\dd\yyyy\#") & ",
" &
Format(Trim(mas(2)), "\#mm\dd\yyyy\#") & ", " & Trim(mas(3)) &
_ ", " & Trim(mas(4)) & ")"

        mCON.Execute strSQL
        i = i + 1    End If    rst.Close    End If
Loop Close (hFile)
strMSG = i & " records from " & k & " added into nakl"
MsgBox strMSG
Print #hFileOut, strMSG & " -- " & Now
'Загрузка информации о накладных (товары) из текстового файла
' 5 *****

fName = "doc.txt"
If Dir(path_x + fName) = "" Then
    strMSG = "File " & fName & " not found!"
    MsgBox strMSG
    Print #hFileOut, strMSG & " -- " & Now
    Close (hFileOut)
    Exit Sub
End If
hFile = FreeFile
Open path_x + fName For Input Access Read As #hFile i = 0 k = 0
Do Until EOF(hFile)
    Line Input #hFile, strBuf    mas = Split(strBuf, ";") If UBound(mas) = 2 Then    k = k + 1
    strSQL = "select * from doc where key_nakl = " &
Trim(mas(0)) & " and key_tovar = " & Trim(mas(1))    Set rst = New Recordset
    rst.Open strSQL, mCON, adOpenForwardOnly, adLockReadOnly, adCmdText
    If rst.EOF = True Then
        strSQL = "INSERT into doc (key_nakl, key_tovar, tovar_number_doc) values (" & Trim(mas(0)) & ", " & _
Trim(mas(1)) & ", " & Trim(mas(2)) & ")"

        mCON.Execute strSQL
        i = i + 1    End If    rst.Close
    End If
Loop Close (hFile)
strMSG = i & " records from " & k & " added into doc"
MsgBox strMSG
Print #hFileOut, strMSG & " -- " & Now()

'Загрузка информации о складах из текстового файла
' 6 *****

fName = "sklad.txt"
If Dir(path_x + fName) = "" Then

```

```

strMSG = "File " & fName & " not found!"
MsgBox strMSG
Print #hFileOut, strMSG & " -- " & Now
Close (hFileOut)
Exit Sub
End If
hFile = FreeFile
Open path_x + fName For Input Access Read As #hFile i = 0 k = 0
Do Until EOF(hFile)
    Line Input #hFile, strBuf    mas = Split(strBuf, ";") If UBound(mas) = 2 Then    k = k + 1
    strSQL = "select * from sklad where key_base = " &
Trim(mas(0)) & " and key_tovar = " & Trim(mas(1))    Set rst = New Recordset
rst.Open strSQL, mCON, adOpenForwardOnly, adLockReadOnly, adCmdText
If rst.EOF = True Then
    strSQL = "INSERT into sklad (key_base, key_tovar, tovar_number_sklad) values (" & Trim(mas(0)) & _", " &
Trim(mas(1)) & ", " & Trim(mas(2)) & ")"

    mCON.Execute strSQL    i = i + 1
End If    rst.Close
End If
Loop Close (hFile)
strMSG = i & " records from " & k & " added into sklad"
MsgBox strMSG
Print #hFileOut, strMSG & " -- " & Now()

Print #hFileOut, vbLf & " *** WORKING FINISH at" & " -- " & Now
& " *** " Close (hFileOut) mCON.Close Exit Sub errh:
MsgBox "SOMTHING NOT GOOD"
End Sub

```

Следующим шагом создаем процедуру, работающую с основной формой в немодальном режиме

```

Sub FORM_PREPARE()

Dim mCON As New Connection
Dim strSQL As String
Dim rst As Recordset

Set rst = New Recordset mCON.Open ("db_practic2")

strSQL = "select name_base from base"
rst.Open strSQL, mCON, adOpenForwardOnly, adLockReadOnly, adCmdText
UF_SKLAD.ComboBox1.Clear
Do Until rst.EOF
    UF_SKLAD.ComboBox1.AddItem (rst!name_base)    rst.MoveNext
Loop
UF_SKLAD.ComboBox1.ListIndex = 0 rst.Close
strSQL = "select num_nakl, date_nakl from nakl" rst.Open strSQL, mCON, adOpenForwardOnly, adLockReadOnly,
adCmdText
UF_SKLAD.ComboBox2.Clear
Do Until rst.EOF
    UF_SKLAD.ComboBox2.AddItem (rst!num_nakl & " от " &
Str(rst!Date_nakl))    rst.MoveNext
Loop

UF_SKLAD.ComboBox2.ListIndex = 0 radio_but = "win" radio_but_obj = "sklad" UF_SKLAD.OptionButton1 = True
UF_SKLAD.OptionButton6 = True

UF_SKLAD.Show (0) mCON.Close
End Sub

```

Далее приведен код процедуры отвечающей за вывод информации о товарах на складах и информацию о накладных. Данная процедура выступает в роли своеобразного диспетчера, позволяющего выбрать способ представления информации

```

Sub show_sklad()
Dim mCON As New Connection
Dim strSQL As String

```

```

Dim rst As Recordset
Dim ss As String
Dim nn As Integer
Dim mas() As String
Dim ns As nakl_struct

Set rst = New Recordset mCON.Open ("db_practic2")
Select Case radio_but_obj Case "sklad":
    'Готовим запрос по складам
    strSQL = "select b.name_base, t.name_tovar,
s.tovar_number_sklad as tn, t.unit_tovar, " & _ " t.price_tovar as tp, tn*tp from base as b " & _ "inner join (sklad as s
inner join (tovar as t) on
t.key_tovar=s.key_tovar ) on b.key_base=s.key_base where
b.name_base like '" & UF_SKLAD.ComboBox1.Value & ""

    Case "nakl"
        ss = UF_SKLAD.ComboBox2.Value    mas = Split(ss, " ")    nn = Val(mas(0))

    'Получаем информацию о накладных из представления nakl_q
    'select t.name_tovar, t.price_tovar as pt, d.tovar_number_doc as 'nt, pt*nt as sum_tovar from nakl as n inner join (doc as
d inner
'join tovar as t on t.key_tovar=d.key_tovar) on n.key_nakl =
'd.key_nakl where n.key_nakl = " & ns.key

        strSQL = "select * from nakl_q where num_nakl = " & nn    rst.Open strSQL, mCON, adOpenForwardOnly,
adLockReadOnly, adCmdText
        If rst.EOF <> True Then

            ns.key = rst!key_nakl    ns.num = rst!num_nakl    ns.base = rst!name_base    ns.klad = rst!klad_base
ns.date_outx = rst!date_out_nakl    ns.datex = rst!Date_nakl    ns.shop = rst!name_shop    ns.buh = rst!buh_shop
End If    rst.Close End Select

rst.Open strSQL, mCON, adOpenForwardOnly, adLockReadOnly, adCmdText
UF_SKLAD.ListBox1.Clear

Call clr_wsheets
Select Case radio_but

    Case "win":
        'Вывод информации в окно ListBox
        Select Case radio_but_obj Case "sklad": 'Вывод информации по складу
            Do Until rst.EOF
                ss = rst.Fields(1) & ": " & rst.Fields(2) & " " & rst.Fields(3) & " по " & rst.Fields(4) & " руб. всего на " &
rst.Fields(5) & " руб."
                UF_SKLAD.ListBox1.AddItem (ss)    rst.MoveNext    Loop
            Case "nakl"
                'Вывод информации по накладной
                ss = "Накладная № " & ns.num & " от " & ns.datex
                UF_SKLAD.ListBox1.AddItem (ss)
                'ss = ""
                'UF_SKLAD.ListBox1.AddItem (ss)
                ss = "    Отправитель: " & ns.base    UF_SKLAD.ListBox1.AddItem (ss)
                ss = "    Получатель: " & ns.shop
                UF_SKLAD.ListBox1.AddItem (ss)    ss = ""
                UF_SKLAD.ListBox1.AddItem (ss)    'UF_SKLAD.ListBox1.AddItem (ss)
                ss = "№п.п Наименование цена количество сумма"    UF_SKLAD.ListBox1.AddItem (ss)
                ss = "-----"
                UF_SKLAD.ListBox1.AddItem (ss)
                i = 1
                Do Until rst.EOF
                    ss = rst.Fields(0) & mstr(20 - Len(rst.Fields(0))) & rst.Fields(1) & " руб." & mstr(10 - Len(rst.Fields(1))) &
rst.Fields(2) & mstr(10 - Len(rst.Fields(2))) & rst.Fields(3) &
" руб."
                    UF_SKLAD.ListBox1.AddItem (ss)    i = i + 1    rst.MoveNext    Loop
                    ss = "-----"    UF_SKLAD.ListBox1.AddItem (ss)
                    ss = "Отгрузил: " & ns.klad & "    Принял: " & ns.buh    UF_SKLAD.ListBox1.AddItem (ss)    ss = "Дата
отгрузки: " & ns.datex & "    Дата получения:

```

```

" & ns.date_outx
UF_SKLAD.ListBox1.AddItem (ss)
ss = "Подпись _____ Подпись _____ "
UF_SKLAD.ListBox1.AddItem (ss)
End Select Case "wsheet":
'Вывод информации в рабочий лист
Cells(1, 1).Value = UF_SKLAD.ComboBox1.Value i = 2
Do Until rst.EOF
For j = 1 To 5
Cells(i, j).Value = rst.Fields(j).Value Next j rst.MoveNext i = i + 1
Loop

Case "word":
'Вывод информации в документ MS Word
Call word_exp("rep1.doc", UF_SKLAD.ComboBox1.Value, rst)

End Select
'MsgBox "Information DONE" mCON.Close
End Sub

```

Далее приведен код вспомогательной функции, создающей строку из пробелов нужной длины. при формировании строк для вывода в окно списка пользовательской формы:

```

Function mstr(l As Integer) As String

Dim ss As String ss = "" If l > 0 Then For i = 0 To l ss = ss + " "
Next Else ss = "" End If mstr = ss End Function Sub clr_wsheets()
ActiveWorkbook.Worksheets("info_out").Activate
Range("A:F").Select
Selection.ClearContents
End Sub

```

Следующая процедура формирует документ Word для вывода информации о товарах, находящихся на складе:

```

Sub word_exp(fn As String, cn As String, rstx As Recordset)

Dim oApp As New Word.Application
Dim oDoc1 As Word.Document
Dim TableWord As Word.Table
Dim pathx As String

pathx = ActiveWorkbook.path + "\"

MsgBox "Приступаем к сохранению отчета по фирме" & vbCrLf & cn & vbCrLf & "в файл " & fn

Set oDoc1 = oApp.Documents.Add oApp.Visible = True oDoc1.Activate

oApp.CentimetersToPoints(2)
oDoc1.Application.Selection.PageSetup.LeftMargin = oApp.CentimetersToPoints(2)
oDoc1.Application.Selection.PageSetup.RightMargin = oApp.CentimetersToPoints(1)
oDoc1.Application.Selection.PageSetup.TopMargin = oApp.CentimetersToPoints(2.5)
oDoc1.Application.Selection.PageSetup.BottomMargin = oApp.CentimetersToPoints(1.5)

With oDoc1.Application.Selection
.ParagraphFormat.Alignment = 3
.Font.Bold = True
.Font.Size = 13
.Font.Name = "Times New Roman"
End With
oDoc1.Application.Selection.TypeText ("Отчет от " & Now())
With oDoc1.Application.Selection
.ParagraphFormat.Space15
.TypeText (vbLf & vbCrLf)
.EndOf
End With
With oDoc1.Application.Selection
.ParagraphFormat.Alignment = 1

```



```

.Font.Bold = True
.Font.Size = 20
.Font.Name = "Times New Roman"
.InsertAfter "ООО"
.InsertAfter " "
.InsertAfter """" .InsertAfter cn
.InsertAfter """"
.EndOf
.InsertParagraphAfter
.InsertAfter "Перечень товаров, имеющихся на складе"
.EndOf
End With

With oDoc1.Application.Selection
.ParagraphFormat.Alignment = 1
.Font.Bold = True
.Font.Size = 13
.Font.Name = "Times New Roman"
End With

Set TableWord =
oDoc1.Tables.Add(oDoc1.Application.Selection.Range, 1, 5)

'TableWord.bo
TableWord.Borders(wdBorderTop).Visible = True
TableWord.Borders(wdBorderLeft).Visible = True
TableWord.Borders(wdBorderVertical).Visible = True
TableWord.Borders(wdBorderRight).Visible = True
TableWord.Borders(wdBorderHorizontal).Visible = True
TableWord.Borders(wdBorderBottom).Visible = True
TableWord.Cell(1, 1).Range.Text = "Наименование"
TableWord.Cell(1, 2).Range.Text = "Кол-во"
TableWord.Cell(1, 3).Range.Text = "Единица"
TableWord.Cell(1, 4).Range.Text = "Цена (руб)" TableWord.Cell(1, 5).Range.Text = "Сумма (руб)"
i = 0
Do Until rstx.EOF
oDoc1.Application.Selection.InsertRowsBelow
For j = 1 To 5
If j = 4 Or j = 5 Then
TableWord.Cell(i + 2, j).Range.Text =
Format(rstx.Fields(j).Value, "##,##0.00")
Else
TableWord.Cell(i + 2, j).Range.Text = rstx.Fields(j).Value
End If Next j rstx.MoveNext i = i + 1
Loop
oDoc1.Application.Selection.EndOf

TableWord.Columns(1).Width = oApp.CentimetersToPoints(5)
TableWord.Columns(2).Width = oApp.CentimetersToPoints(3)
TableWord.Columns(3).Width = oApp.CentimetersToPoints(3)
TableWord.Columns(4).Width = oApp.CentimetersToPoints(3) TableWord.Columns(5).Width =
oApp.CentimetersToPoints(3)

Dim fName As String Do
fName = Application.GetSaveAsFilename()
Loop Until fName <> "False" oDoc1.SaveAs fName

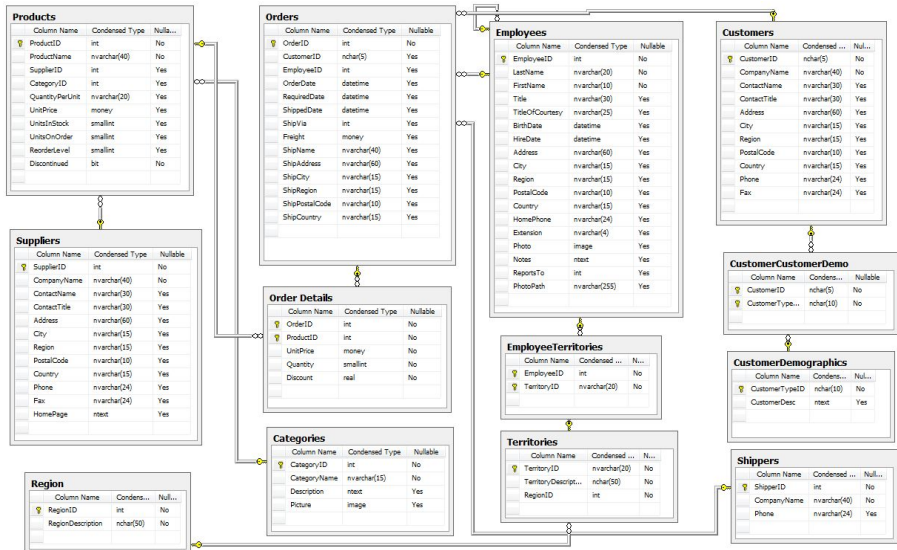
MsgBox "Формирование отчета завершено!" oDoc1.Close oApp.Quit
End Sub

```

Первые две процедуры являются «разовыми» нужны только в начале выполнения проекта, поэтому они запускаются по вызову из среды разработки. Сама форма взаимодействия с пользователем запускается из процедуры FORM_PREPARE.

Пример. VBA-Excel: СУБД «Товары на складе»
Задача

В компании ведется учет товаров, которые имеются на складе, при помощи таблицы Products БД Northwind:



В этой таблице находятся следующие важные столбцы

Product ID	Идентификатор товара
ProductName	Наименование продукта
UnitPrice	Стоимость продукта за единицу
UnitsInStock (На складе)	Количество единиц этого товара на складе
ReorderLevel (Минимальный запас)	Минимально допустимое количество единиц данного товара на складе. Если реальное количество единиц этого товара меньше, чем этот уровень, товар нужно срочно заказывать
Discontinued (Поставки прекращены)	Флаг прекращения работы с товаром. Если в этом столбце стоит «истина», то это значит, что принято решение закупки этого товара больше не производить

Все остальные столбцы для целей этой работы можно игнорировать. Заполнение таблицы Products производится при помощи специализированного приложения, созданного достаточно давно и не предусматривающего некоторых необходимых форм. Требуется создать приложение на основе Excel, которое: производит вставку в лист Excel данные по всем строкам и всем вышеуказанным столбцам этой таблицы; генерирует в Excel дополнительные столбцы:

- Заказать товар, штук - разница между столбцами ReorderLevel (Минимальный запас) и UnitsInStock (На складе) - этот столбец помещается информация о количестве товара в штуках, которое нужно срочно заказать).

Эту информацию нужно генерировать только для тех записей, для которых значение в столбце ReorderLevel (Минимальный запас) больше, чем в столбце UnitsInStock (На складе), и у которых значение столбца Discontinued (Поставки прекращены) установлено в Ложь. Стоимость заказа - определяло бы стоимость такого пополнения склада для каждой строки в таблице. Стоимость заказа рассчитывается как произведение предыдущего столбца и столбца UnitPrice (Цена). Эту информацию также нужно генерировать только для тех записей, для которых значение в столбце ReorderLevel (Минимальный запас) больше, чем в столбце UnitsInStock (На складе).

- Вставляло бы одной строкой ниже полученных записей из базы данных две итоговые строки: общая стоимость товаров на складе - итоговая стоимость всех товаров, которые находятся на складе (как сумма произведений столбцов «На складе» и «Цена для каждой строки»); общая стоимость товаров к заказу - итог по столбцу Стоимость заказа.

Общий вид получившегося приложения (первые строки листа с импортированными данными):

G79									
A	B	C	D	E	F	G	H	I	J
1	Получить данные								
2									
3									
4	ProductID	ProductName	UnitPrice	UnitsInStock	ReorderLevel	Discontinued	Заказать товар, штук	Стоимость заказа	
5	1	Chai	18	39	10	ЛОЖЬ			
6	2	Chang	19	17	25	ЛОЖЬ	8	152,00p.	
7	3	Aniseed Syrup	10	13	25	ЛОЖЬ	12	120,00p.	
8	4	Chef Anton's Cajun Seasoning	22	53	0	ЛОЖЬ			
9	5	Chef Anton's Gumbo Mix	21,35	0	0	ИСТИНА			

Последние строки с итоговыми значениями:

N92												
A	B	C	D	E	F	G	H	I	J	K	L	M
69	Gudbrandsdalsost	36	26	15	ЛОЖЬ							
70	Outback Lager	15	15	30	ЛОЖЬ	15	225,00p.					
71	Flatemyssost	21,3	26	0	ЛОЖЬ							
72	Mozzarella di Giovanni	34,8	14	0	ЛОЖЬ							
73	Rud Kaviar	15	101	5	ЛОЖЬ							
74	Longlife Tofu	10	4	5	ЛОЖЬ	1	10,00p.					
75	Rhinlorau Klosterbier	7,75	125	25	ЛОЖЬ							
76	Lakkalikauri	18	57	20	ЛОЖЬ							
77	Original Frankfurter grune Soße	13	32	15	ЛОЖЬ							
Общая стоимость товаров на складе:			74,050,85p.									
Общая стоимость товаров к заказу:			3,633,45p.									

Решение

Принцип работы: подключение к БД и создание Query Table на основе объекта RecordSet, в котором хранится информация, полученная из БД.

- Создайте новый файл Excel, на вкладке Разработчик в группе Элементы управления нажмите кнопку «Вставить» и среди элементов ActiveX щелкните по элементу управления Кнопка и нарисуйте кнопку на листе Excel. Для определенности будем считать, что созданная кнопка занимает ячейки с A1 ÷ E1 первого листа.
- Щелкните по кнопке Свойства (проверьте, что нажата кнопка Режим конструктора) и в появившемся окне Properties настройте для свойства Caption значение «Получить данные». При необходимости воспользуйтесь свойством Font, чтобы настроить подходящий шрифт для вашей кнопки.
- Щелкните правой кнопкой мыши по созданной вами кнопке и в контекстном меню выберите Исходный текст. Откроется редактор Visual Basic с курсором ввода на месте события Click для вашей кнопки.
- В окне редактора кода в меню Tools выберите References и установите флажок напротив строки Microsoft ActiveX Data Objects 2.1 Library.

Код для события Click вашей кнопки «Получить данные»:

```
Private Sub CommandButton1_Click()
'Вначале - чистим всю книгу от старых данных Cells.Select
Selection.Clear

' Создаем и настраиваем объект Connection Dim cn As New ADODB.Connection
cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Nwind.mdb" cn.Open

'Создаем и настраиваем объект Recordset Dim rs As New ADODB.Recordset
rs.Open "SELECT[ProductID],[ProductName],[UnitPrice],[UnitsInStock],[ReorderLevel]," & _ "[Discontinued] FROM Products", cn
'На основе Recordset создаем объект QueryTable и вставляем его, начиная с 4-й строки Dim QT1 As QueryTable
Set QT1 = QueryTables.Add(rs, Range("A4"))
QT1.Refresh

'Определяем количество записей в QueryTable
Dim nRowCount As Integer
Dim oRange As Range Set oRange = QT1.ResultRange
nRowCount = oRange.Rows.Count

'Формируем столбец "Заказать товара, штук"
Range("G4").Value = " Заказать товара, штук "
Range("G4").Font.Bold = True Range("G4").Columns.AutoFit

'Формируем столбец "Стоимость заказа" Range("H4").Value = " Стоимость заказа"
Range("H4").Font.Bold = True Range("H4").Columns.AutoFit

'Создаем диапазон, который включит в себя столбец G
' "вдоль" QueryTable
Set oRange = Range("G5", "G" & nRowCount + 3)

'Готовим переменные, которые нам потребуются в цикле Dim oCell As Range
Dim sRowNumber As String
Dim cMoney As Currency
Dim cItoMoney As Currency
Dim cItoSklad As Currency

'Проходим циклом по всем ячейкам созданного диапазона For Each oCell In oRange.Cells
'Получаем абсолютный номер строки в виде строковой переменной sRowNumber = Replace(oCell.Address(True),
"$G$", "")
'Проверяем определенные нами условия
If Range("E" & sRowNumber).Value > Range("D" & sRowNumber) And _
```

```

Range("F" & sRowNumber).Value = False Then
'Получаем значение для столбца G (заказ в штуках)
oCell.Value = (CInt(Range("E" & sRowNumber).Value) - CInt(Range("D" &
sRowNumber).Value))
'Получаем значение для столбца H (стоимость заказа)
cMoney = (CInt(Range("E" & sRowNumber).Value) - CInt(Range("D" & sRowNumber).Value)) * CCur(Range("C" &
sRowNumber).Value)
'Записываем его в столбец H
Range("H" & sRowNumber).Value = cMoney
'Сразу плюсуем к итогу в рублях cItogMoney = cItogMoney + cMoney
End If

'И в том же цикле сразу суммируем стоимость товаров на складе cItogSklad = cItogSklad + (Range("C" &
sRowNumber).Value * Range("D" &
sRowNumber).Value)
Next

'Формируем две строки с итогами
Range("B" & nRowCount + 6).Value = " Общая стоимость товаров на складе:" Range("B" & nRowCount +
6).Font.Bold = True
Range("B" & nRowCount + 7).Value = " Общая стоимость товаров к заказу:" Range("B" & nRowCount +
7).Font.Bold = True
Range("D" & nRowCount + 6).Value = cItogSklad Range("D" & nRowCount + 6).Font.Bold = True
Range("D" & nRowCount + 7).Value = cItogMoney
Range("D" & nRowCount + 7).Font.Bold = True

'Для красоты выделяем итоговое значение
Range("D" & nRowCount + 7).Select
'и производим скроллинг
Range("D" & nRowCount + 7).Show

```

Задания для самостоятельной работы

Для варианта задания написать программу СУБД на языке VBA-Access и/или VBA-Excel.

Варианты заданий:

1. АИС «Индивидуальный план преподавателя».
2. АИС «Обслуживание заказов клиентов».
3. АИС «Прохождение преддипломной практики студентами».
4. АИС «Лицензионное программное обеспечение организации».
5. АИС «Арендная плата за нежилые помещения».
6. АИС «Списание основных средств».
7. АИС «Аттестация сотрудников предприятия».
8. АИС «Трудоустройство».
9. АИС «Спортивные сооружения области».
10. АИС «Справочник предприятия».
11. АИС «Паспорт здоровья сотрудника».
12. АИС «Справочник абитуриента».
13. АИС «Платные образовательные услуги населению».
14. АИС «Новостная лента организации».
15. АИС «Анализ продаж».
16. Текстовый процессор Microsoft Word.
17. Табличный процессор Microsoft Excel.
18. Персональный коммуникатор Microsoft Outlook.
19. Приложение для управления базами данных (СУБД) Microsoft Access.
20. Приложение для организации общения Microsoft Communicator (Microsoft Lync).
21. Приложение для подготовки публикаций Microsoft Publisher.
22. Приложение для работы с бизнес-диаграммами Microsoft Visio.

Контрольные вопросы

- 1.Парадигмы программирования.
- 2.Модульное программирование.
- 3.БД: модель данных, предметная область, сущности и атрибуты.
- 4.БД: реляционная модель данных, основные принципы реляционных БД.
- 5.БД: связи между таблицами, типы связей, основные и внешние ключи.
- 6.БД MS Access: создание таблиц, создание представлений.
- 7.БД MS Access: создание форм доступа к таблицам, создание отчетов.
- 8.Архитектура БД для VBA: интерфейс, ядро, хранилище.
- 9.Архитектура БД для VBA: локальные и удаленные БД.
- 10.Создание БД с DAO: объекты Workspace, Database, TableDef, Field, Index, Relation.
- 11.Работа с БД на VBA: подключение библиотек, объявление переменных.
- 12.Работа с БД на VBA: подготовка запроса к БД, операторы выполнения запросов.
- 13.Работа с БД на VBA. Программное получение информации о структуре БД.
- 14.Уплотнение БД. Зачем нужна операция уплотнения?
- 15.Выполнение параметрических запросов из программы.
- 16.Отличия хранимого запроса от запроса, формируемого в программе.