

Лабораторная работа №6  
по дисциплине “Модели данных”

АИС “Школа”

1. Цель работы
2. Создание 4-х скриптов на открытие/закрытие форм/отчетов
3. Создание группы макросов/скриптов
4. Организация работы скриптов по разным событиям
5. Импорт и экспорт данных
6. Автоматизация импорта и экспорта
7. Выводы

МГТУ им. Н.Э. Баумана

Студент: Кучеренко М.А.

Группа: ИУ5-34

Дата: 04.12.2017

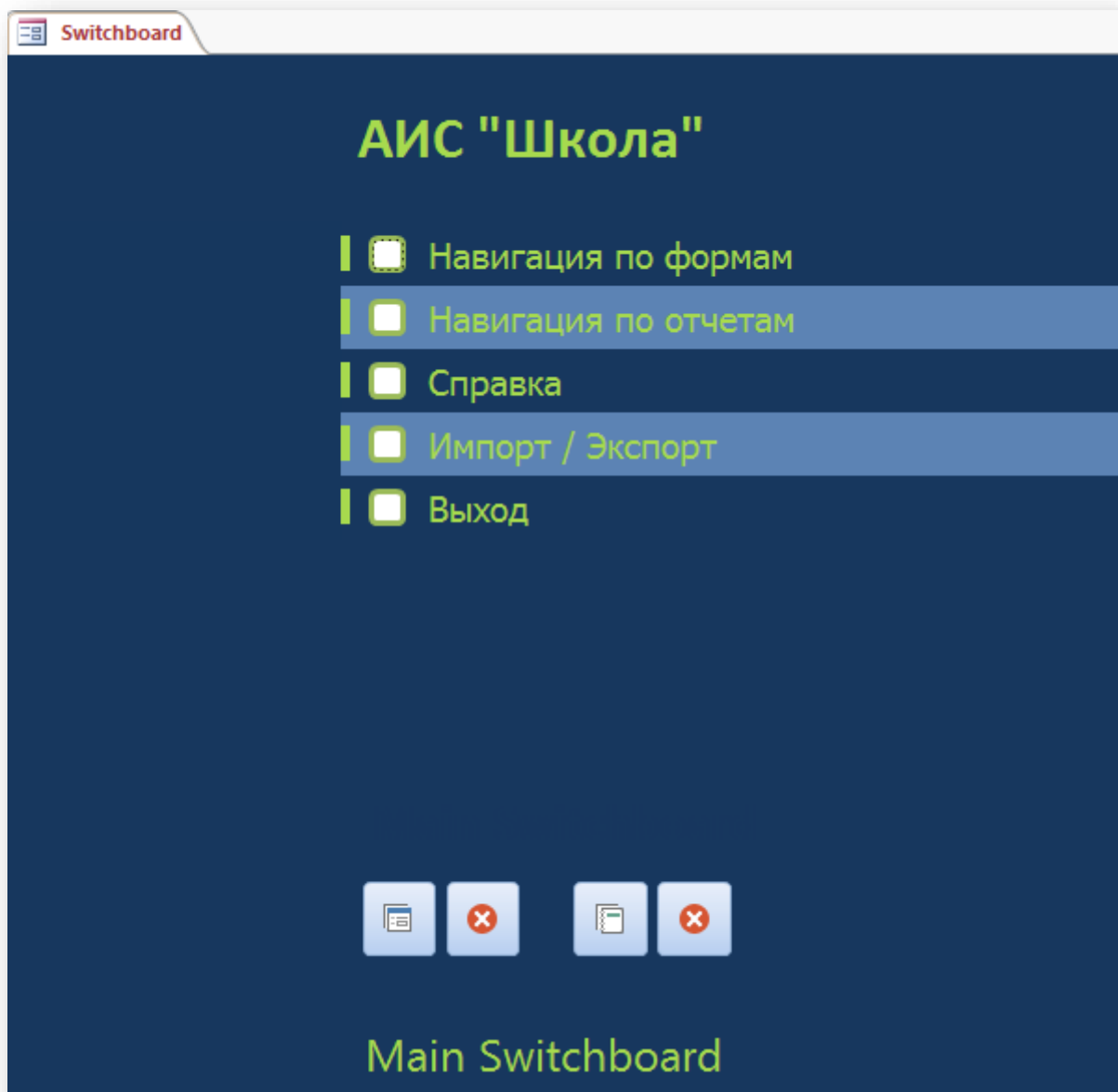
## 1. Цель работы

Изучить методику написания простейших макросов, технологию обмена данными между СУБД MS Access и табличным процессором MS Excel.

## 2. Создание 4-х скриптов на открытие/закрытие форм/отчетов

Вместо создания макросов взят подход написания скриптов на vba, как полностью аналогичный, но более гибкий.

Расположим кнопки, по которым будут вызываться скрипты, на главной кнопочной форме, для удобства.



Создаем в событиях нажатия на кнопки 4 скрипта:

```
Private Sub CloseAll_Click()  
    Dim Obj As AccessObject, dbs As Object  
    Set dbs = Application.CurrentProject  
    For Each Obj In dbs.AllForms  
        If (Obj.IsLoaded) And (Not Obj.name = "АИС ""Школа""") Then  
            DoCmd.Close acForm, Obj.name  
        End If  
    Next Obj  
End Sub
```

---

```
Private Sub CloseAllReport_Click()  
    Dim Obj As AccessObject, dbs As Object  
    Set dbs = Application.CurrentProject  
    For Each Obj In dbs.AllReports  
        If (Obj.IsLoaded) Then  
            DoCmd.Close acReport, Obj.name  
        End If  
    Next Obj  
End Sub
```

---

```
Private Sub OpenAll_Click()  
    Dim Obj As AccessObject, dbs As Object  
    Set dbs = Application.CurrentProject  
    For Each Obj In dbs.AllForms  
        If (Not Obj.IsLoaded) And (Not Obj.name = "LogInForm") Then  
            DoCmd.OpenForm Obj.name, acNormal  
        End If  
    Next Obj  
    DoCmd.OpenForm "АИС ""Школа""", acNormal  
End Sub
```

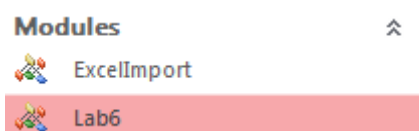
---

```
Private Sub OpenAllReports_Click()  
    Dim Obj As AccessObject, dbs As Object  
    Set dbs = Application.CurrentProject  
    For Each Obj In dbs.AllReports  
        If (Not Obj.IsLoaded) Then  
            DoCmd.OpenReport Obj.name, acViewPreview  
        End If  
    Next Obj  
End Sub
```

То есть просто проходимся по всем элементам (формам или отчетам) и последовательно открываем (закрываем) их.

### 3. Создание группы макросов/скриптов

Вместо группы макросов, опять-таки воспользуемся аналогичным подходом – создадим отдельный модуль и несколько процедур/функций в нем:



#### Option Compare Database

```
Public Sub OpenReportToView(ReportName As String)
    DoCmd.OpenReport ReportName, acViewReport
End Sub
```

```
Public Function MouseDown(ByRef Obj As Image, name As String) As Boolean
    Obj.Picture = "Z:\virtual sharing\МД\datamodels\img\ico\" & name & "_hover.png"
    MouseDown = True
End Function
```

```
Public Function MouseUp(ByRef Obj As Image, name As String) As Boolean
    Obj.Picture = "Z:\virtual sharing\МД\datamodels\img\ico\" & name & "_normal.png"
    MouseUp = True
End Function
```

```
Public Function RecordFocusChanged(ByRef Field As TextBox) As Boolean
    If (Field.Value >= 1000) Then
        Field.BackColor = RGB(247, 168, 171)
    Else
        Field.BackColor = RGB(215, 227, 188)
    End If
    RecordFocusChanged = True
End Function
```

Сразу назовем их по событиям, которые они будут обрабатывать, а также добавим набор картинок в (путь выбран произвольно):

Z:\virtual sharing\МД\datamodels\img\ico\



## 4. Организация работы скриптов по разным событиям

Добавим элементы на форму и напомним для них обработчики:

```

Private Sub AddImg_Click()
    Lab6.OpenReportToView ("Кабинеты - Стоимость")
End Sub

Private Sub AddImg_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Res = Lab6.MouseDown(Me![AddImg], "add")
End Sub

Private Sub AddImg_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Res = Lab6.MouseUp(Me![AddImg], "add")
End Sub

Private Sub Form_Current()
    Res = Lab6.RecordFocusChanged(Me![Text53])
End Sub

Private Sub HelpImg_Click()
    Res = MsgBox("При значении, меньшем 1000, " & _
        "подсвечивает зеленым, иначе красным", _
        vbOKOnly, "Справка")
End Sub

Private Sub HelpImg_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Res = Lab6.MouseDown(Me![HelpImg], "help")
End Sub

Private Sub HelpImg_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Res = Lab6.MouseUp(Me![HelpImg], "help")
End Sub

Private Sub Text53_AfterUpdate()
    Res = Lab6.RecordFocusChanged(Me![Text53])
End Sub

```

Можно заметить, что обращение к процедурам/функциям из модуля происходит аналогично обращению к макросам из группы макросов, то есть через оператор “точки”.

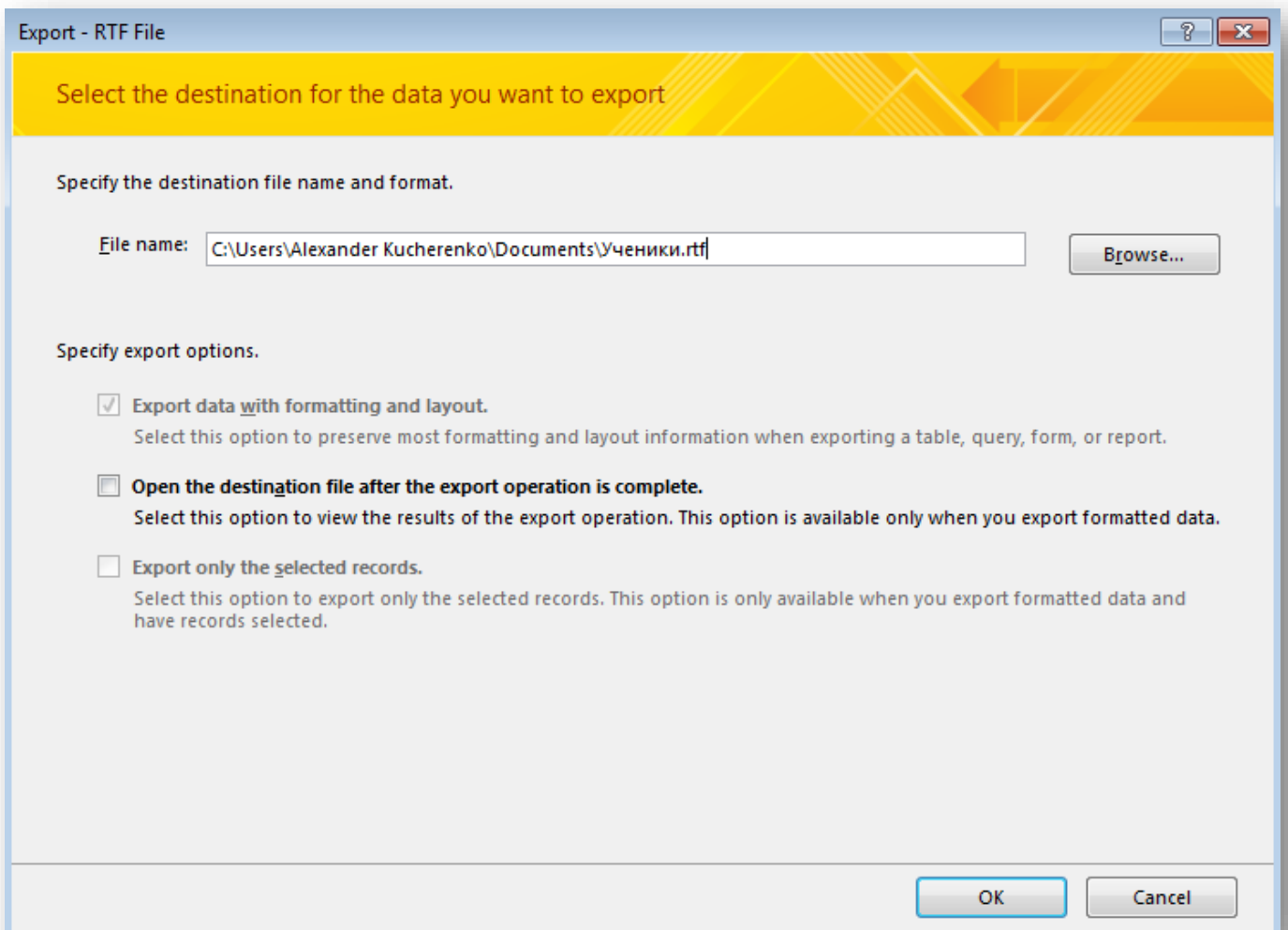
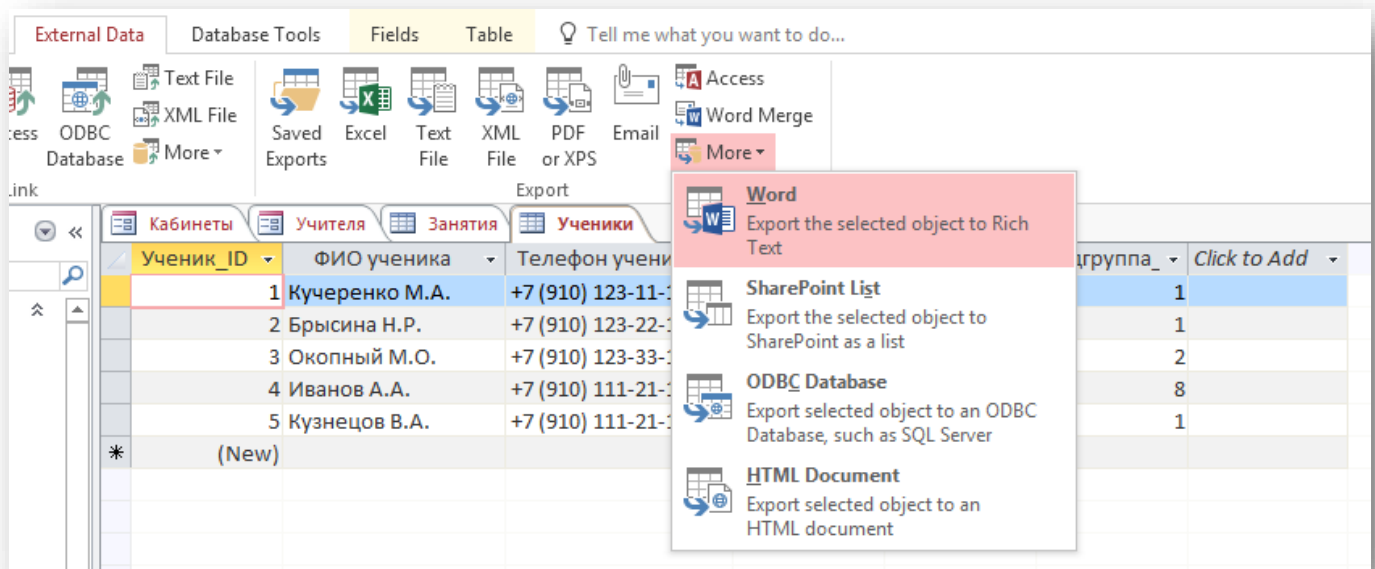
## 5. Импорт и экспорт данных

Сначала проведем тривиальную ручную операцию экспорта из таблицы MS Access в таблицы MS Excel, MS Word, HTML, XML.

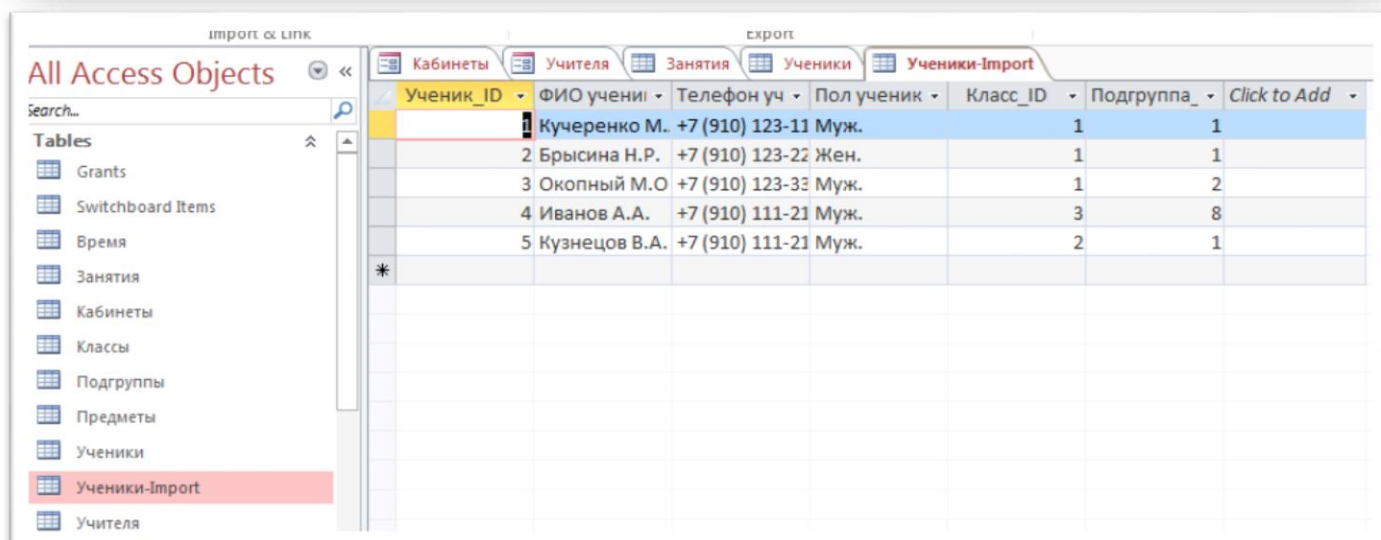
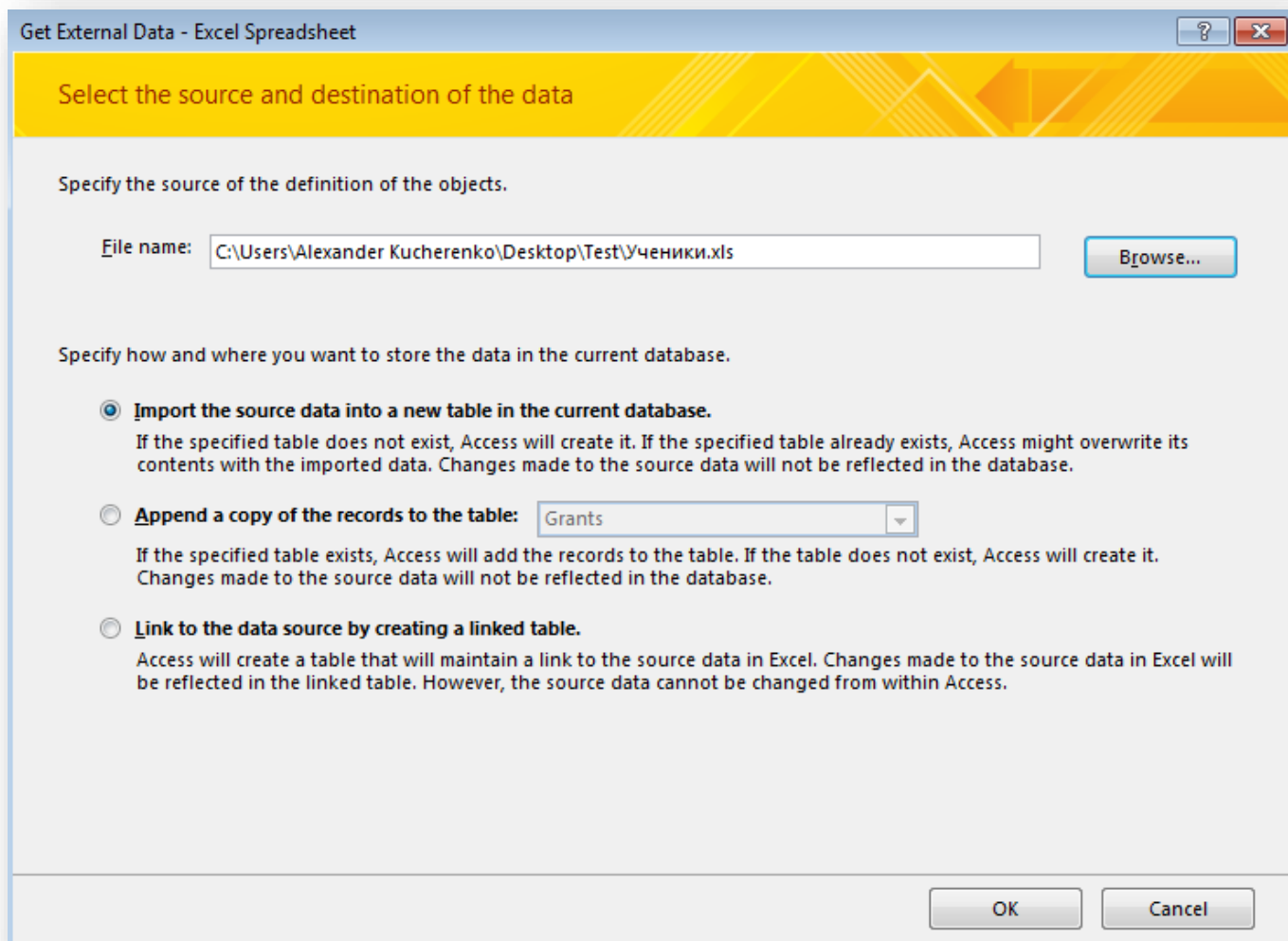
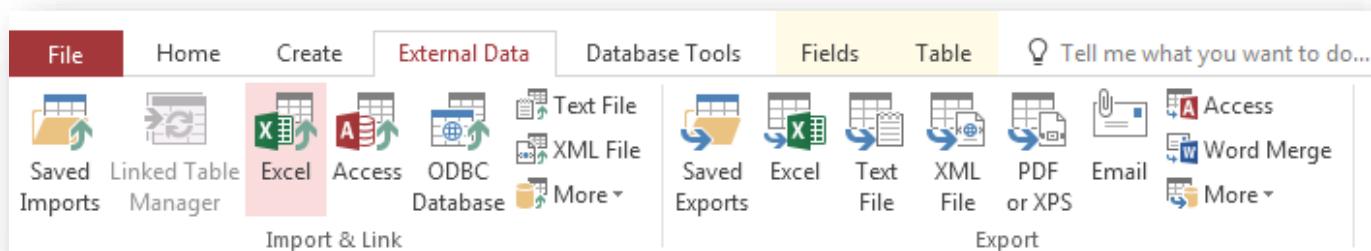
Все эти операции аналогичны, поэтому приведу только пример экспорта в документ MS Word (.rtf)

Затем проведем ручную операцию импорта из вышеприведенных форматов обратно в MS Access.

И также, приведу только импорт из MS Excel, т.к. импорт из других аналогичен.



Ученики.html	11.12.2017 0:57	Firefox HTML Doc...	2 КБ
Ученики.rtf	11.12.2017 0:57	Формат RTF	8 КБ
Ученики.xls	11.12.2017 0:46	Лист Microsoft Ex...	12 КБ
Ученики.xml	11.12.2017 0:54	Документ XML	3 КБ



## 6. Автоматизация импорта и экспорта

Но тривиальные ручные операции экспорта и импорта в 2к17 слишком скучны, поэтому добавим автоматизированный экспорт в наш макет АИС.

Для простоты рассмотрим только автоматизированный экспорт/импорт только для .xls формата (формата MS Excel).

Добавим модуль **ExcelImport**, для надежности там будут храниться проверки.

```
Option Compare Database
Option Explicit
```

```
Public Sub ImportExcelSpreadsheet(fileName As String, tableName As String)
On Error GoTo BadFormat
    DoCmd.TransferSpreadsheet acImport, acSpreadsheetTypeExcel12, tableName, fileName, True
Exit Sub
```

```
BadFormat:
    MsgBox "The file you tried to import was not an Excel spreadsheet."
End Sub
```

Добавим отдельную форму (доступ к ней через кнопочную форму):

ImportExport

Импорт и экспорт данных

Путь до папки: C:\Users\Alexander Kucherenko\Desktop\Test Поиск

Экспорт таблиц в файлы Excel

Имя файла: C:\Users\Alexander Kucherenko\Desktop\Test\Ученики.xls Поиск

Импорт таблицы из файла Excel

Пропишем обработчики для кнопок – вызов диалоговых окон по выбору папки и файла, а также непосредственно сам импорт и экспорт по указанным данным (+ обработка ошибок “на дурака”).

Код см. на след. странице.



```

Private Sub btnBrowse_Click()
    Dim diag As Office.FileDialog
    Dim item As Variant

    Set diag = Application.FileDialog(msoFileDialogFilePicker)
    diag.AllowMultiSelect = False
    diag.Title = "Please select an Excel Spreadsheet"
    diag.Filters.Clear
    diag.Filters.Add "Excel Spreadsheets", "*.xls, *.xlsx"

    If diag.Show Then
        For Each item In diag.SelectedItems
            Me.txtFileName = item
        Next
    End If
End Sub

```

---

```

Private Sub Command2_Click()
    Dim tbl As TableDef

    If Nz(Me.Path, "") = "" Then
        MsgBox "Нащальнике, выбери папку!"
        Exit Sub
    End If

    For Each tbl In CurrentDb.TableDefs
        If Not tbl.name Like "MSys*" And Not tbl.name Like "~*" Then
            DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, _
                tbl.name, Me![Path].Value & "\" & tbl.name & ".xls", True, tbl.name
        End If
    Next
End Sub

```

---

```

Private Sub Command29_Click()
    Dim FSO As New FileSystemObject

    If Nz(Me.txtFileName, "") = "" Then
        MsgBox "Нащальнике, выбери файл!"
        Exit Sub
    End If

    If FSO.FileExists(Nz(Me.txtFileName, "")) Then
        ExcelImport.ImportExcelSpreadsheet Me.txtFileName, FSO.GetFileName(Me.txtFileName)
    Else
        MsgBox "File not found!"
    End If
End Sub

```

---

```

Private Sub Command51_Click()
    Dim diag As Office.FileDialog
    Dim item As Variant

    Set diag = Application.FileDialog(msoFileDialogFolderPicker)









    diag.AllowMultiSelect = False
    diag.Title = "Please select a folder to save data"

    If diag.Show Then
        For Each item In diag.SelectedItems
            Me.Path.Value = item
        Next
    End If
End Sub

```

Прекрасно, теперь можно регулярно выгружать данные из АИС и, например, передавать их на анализ в другие программы или на архивирование в резервные копии, из которых потом снова можно импортировать данные в АИС.

Отличное расширение функционала отказоустойчивости нашей АИС.

 Grants.xls	11.12.2017 0:46	Лист Microsoft Ex...	11 КБ
 Switchboard Items.xls	11.12.2017 0:46	Лист Microsoft Ex...	11 КБ
 Время.xls	11.12.2017 0:46	Лист Microsoft Ex...	11 КБ
 Занятия.xls	11.12.2017 0:46	Лист Microsoft Ex...	14 КБ
 Кабинеты.xls	11.12.2017 0:46	Лист Microsoft Ex...	15 КБ
 Классы.xls	11.12.2017 0:46	Лист Microsoft Ex...	11 КБ
 Подгруппы.xls	11.12.2017 0:46	Лист Microsoft Ex...	11 КБ
 Предметы.xls	11.12.2017 0:46	Лист Microsoft Ex...	13 КБ
 Ученики.html	11.12.2017 0:57	Firefox HTML Doc...	2 КБ
 Ученики.rtf	11.12.2017 0:57	Формат RTF	8 КБ
 Ученики.xls	11.12.2017 0:46	Лист Microsoft Ex...	12 КБ
 Ученики.xml	11.12.2017 0:54	Документ XML	3 КБ
 Учителя.xls	11.12.2017 0:46	Лист Microsoft Ex...	12 КБ

## 7. Выводы

Пользовательские макросы и скрипты позволяют значительно расширить функционал АИС, представленной в MS Access. Автоматизирование операций, улучшение эргономичности пользовательского интерфейса, анализ данных и создание резервных копий – все это возможно реализовать с помощью макросов и скриптов. Объединение в группы/модули же позволяет уменьшить количество дублируемого кода и повысить удобство для программиста-разработчика.