

Задание и методические указания по автоматизации выпуска счетов-фактур

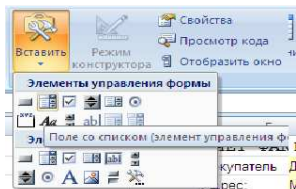
В среде Excel необходимо разработать приложение позволяющее выпускать счета-фактуры в автоматизированном режиме. Приложение должно включать три листа: «Организации», «Счёт-фактура» и «База-счетов». В первом листе указывается список организаций и их реквизитов, для которых составляются счета-фактуры. Во втором листе задан шаблон счёта-фактуры и элементы управления, позволяющие автоматизировать создание и сохранение счёта-фактуры. Третий лист предназначен для сохранения данных счетов фактур.

Работа с приложением должна соответствовать следующему сценарию. В лист «Организации» заносятся реквизиты предприятий. Таким образом создаётся база сведений о предприятиях контрагентах. На листе «Счёт-фактура» находится шаблон документа и следующие элементы управления: поле со списком, позволяющее выбрать требуемую организацию; формулы ссылок на реквизиты организации по выбранной из списка организации; расчёт суммы счёта-фактуры и вывод суммы прописью; кнопка сохранения данных счёта-фактуры в базу на листе «База-счетов»; кнопка переноса данных счёта-фактуры из базы в шаблон для просмотра. В соответствии с этой структурой приложения работа пользователя должна строиться следующим образом. На листе «Счёт-фактура» с помощью поля со списком пользователь выбирает организацию. Если её нет в списке, она добавляется в список на листе «Организации». После её выбора в шаблон счёта-фактуры автоматически вставляются реквизиты выбранной организации и нажимается кнопка «Новый» которая создаёт новый номер счёта-фактуры на 1 больше предыдущего и заносит текущую дату на место даты в шаблоне. Далее заполняется содержимое шаблона. После заполнения должна автоматически появиться сумма и сумма прописью. По нажатию кнопки «Сохранить» данные из счёта-фактуры должны быть перенесены в базу на листе «База-счетов». И, наконец, по нажатию кнопки «Просмотр» должно открыться окно со списком ранее введённых счетов-фактур для выбранной в поле со списком организации. В этом списке должны отражаться номер и дата счёта-фактуры из базы. В этом списке может быть выбрана счёт-фактура и по нажатию кнопки «Просмотреть фактуру» данные этой фактуры должны быть перенесены в шаблон.

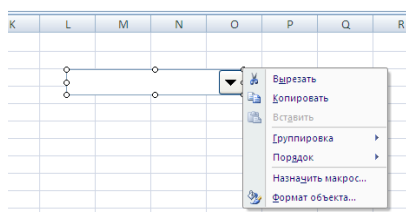
Для выполнения данного задания воспользуйтесь следующими методическими указаниями.

Создадим Excel-книгу и присвоим ей имя «Фактура». Переименуем первый лист как «Организации», второй – «Счёт-фактура» и третий – «База фактур». В листе «Организации» создадим список реквизитов примерно, так как показано на рисунке. Здесь НДС это константа, которая будет использоваться для расчёта суммы НДС в счёте-фактуре. Всё остальное реквизиты организаций, для которых выписываются счета фактуры.

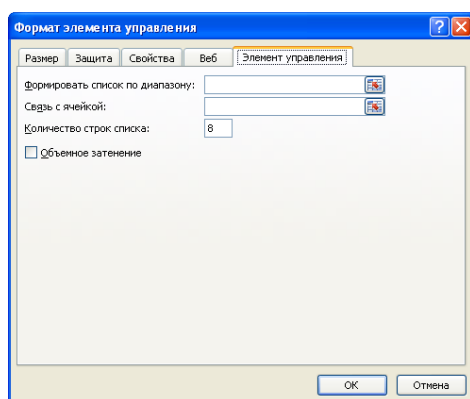
Автоматизацию расчётов начнём с выбора реквизитов покупателя. Выбор покупателя удобно осуществлять по его номеру в списке на листе «Организации». Поэтому в любой свободной ячейке можно выделить место для хранения текущего номера организации. В данном примере это ячейка J1. После этого добавим на лист «Счёт-фактура» элемент управления *поле со списком*. В Excel 2007 это делается на панели «Разработчик» выбором элемента управления «Поле со списком», как показано на рисунке и перетаскиванием его на лист «Счёт-фактура».



В результате получившийся фрагмент листа может выглядеть следующим образом.



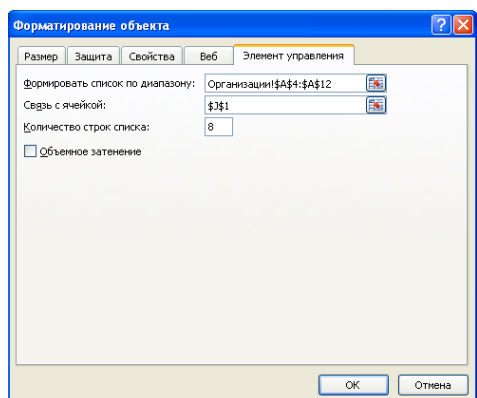
Теперь нужно настроить вставленный элемент управления. Для этого нужно кликнуть правой клавишей мыши на одной из точек объекта. В результате появляется выпадающий список, как показано на рисунке. В нём нужно выбрать «Формат объекта». В результате открывается следующее окно настроек.



Кликнем кнопку выбора в строке «Формировать список по диапазону», при появившемся элементе управления перейдём на лист «Организации» и выделим диапазон ячеек с названиями организаций (с возможностью расширения списка). В результате появится примерно следующая ситуация.

	A	B	C	D
1				
2				
3	Организация	Адрес:	Телефон	Расчетный счет N
4	ДРЭУ Кировского района	Майков переулок д 8	252-61-18	230508504 в Кировском ф-ле ОАО ПС
5	МРЭУ Выборгского района	Верхний пер. 17	345-67-89	170345000000043 в Выборгском
6				
7				
8				
9				
10				
11				
12				
13				
14				

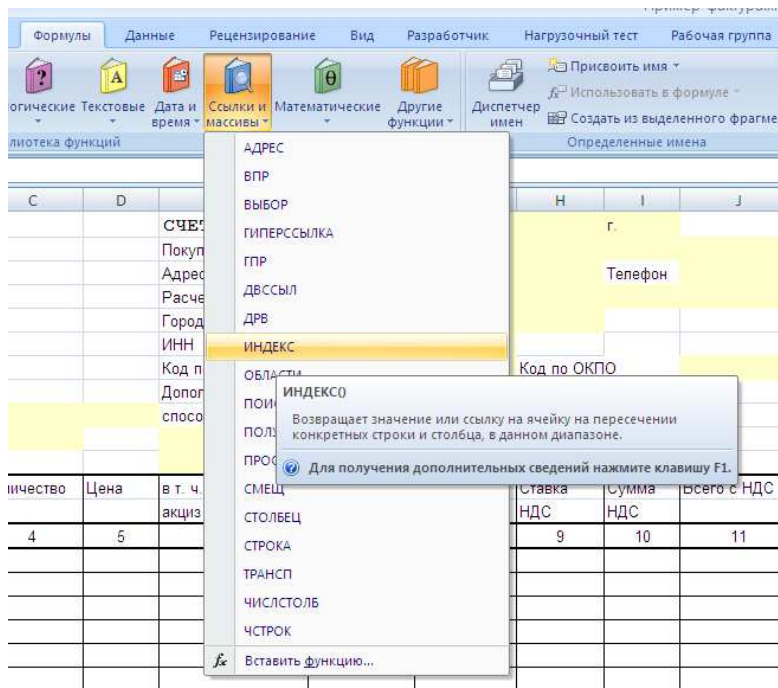
Для возврата в окно настроек, вновь нажмём кнопку выбора на всплывающем окне «Формат элемента управления». Так же настроим диапазон в строке окна «Связь с ячейкой», но в качестве диапазона выберем объявленную ранее ячейку J1. В результате окно настроек должно принять следующий вид.



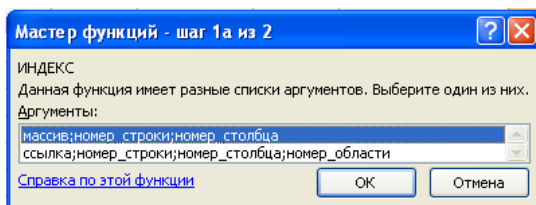
После нажатия кнопки «ОК» элемент правления «Поле со списком» настроен. Теперь он позволяет выбирать организацию из списка и заносить её порядковый номер в ячейку J1. Результат его работы показан ниже.

J	K	L	M	N	O
2					

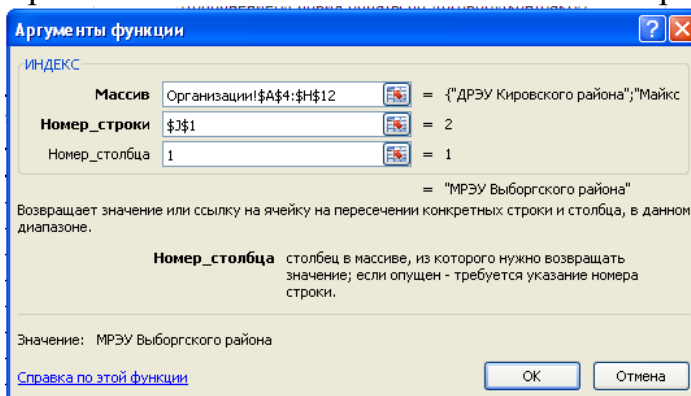
Следующим этапом является автоматизация вставки реквизитов в шаблон документа в соответствии с выбранным номером организации. Первым вставим в ячейку F2 название организации из списка реквизитов листа «Организации». Для этого используем функцию ИНДЕКС. Для Excel 2007 её можно найти в панели инструментов «Формулы» на кнопке «Ссылки и массивы», как показано на рисунке.



После выбора формулы нужно провести настройки функции. Они выполняются с помощью мастера функций.



При нажатии «ОК» появляется окно настроек.

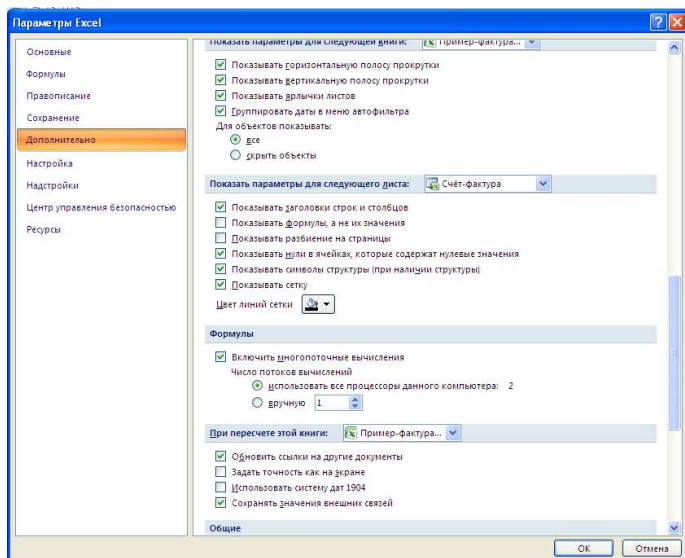


Используемая функция имеет три параметра: первый «Массив» указывает диапазон ячеек с реквизитами организаций; второй «Номер строки» указывает адрес ячейки, где хранится номер строки ячейки, значение которой нужно выбрать; третий «Номер столбца» содержит номер столбца указанной ячейки (он может быть задан не адресом ячейки, а конкретным числом). Работа функции осуществляется по следующей схеме: в таблице, определяемой массивом, заданным в первом параметре, выбирается содержимое ячейки, номер строки и столбца которой заданы во втором и

третьем параметрах и помещается в ячейку, где находится функция. Значение номера строки содержится в ячейке J1, которое, как было описано ранее, заносится с помощью поля со списком.

Аналогичные функции необходимо проставить в ячейки шаблона соответствующие подписям реквизитов. В них будут меняться только номера столбцов (Почему?). Для того, чтобы можно было копировать функцию, созданную в первый раз (действительно функции для всех реквизитов отличаются только номером столбца), необходимо закрепить значения интервала ячеек и номера ячейки номером строки интервала. Это необходимо для того, чтобы при копировании функции не выполнялась автоматически подстройка адресов параметров. Закрепление выполняется путём выделения адреса (интервала адресов) и нажатием клавиши F4. В результате перед номерами столбцов и строк ставятся знаки \$ и при копировании настройка адресов не происходит. После копирования можно проверить работу функций. В поле со списком выберите организацию – в ячейках, где стоят функции должны появиться правильные значения реквизитов для выбранной организации.

На следующем этапе заполним содержимое таблицы формулами, чтобы при заполнении счёта-фактуры автоматически выполнялись расчёт столбцов «Сумма», «НДС» и «Всего с НДС». Напомню, что ввод формул начинается со знака =, и далее курсором выделяются ячейки операнды, разделяемые знаками арифметических (и других) операций. Для указанных столбцов формулы первой строки фактуры могут выглядеть так. Ячейка F15:=C15*D15, ячейка I15: =F15*H15, ячейка J15: =F15+I15. Предполагаю, что копирование формул на весь документ известно студентам. В результате копирования в столбцах содержащих формулы при незаполненном документе отображаются нули. Для того чтобы скрыть нулевые значения в Excel 2007 необходимо снять галочку в позиции «Показывать нули в ячейках, которые содержат нулевые значения» параметров Excel, как показано на рисунке.

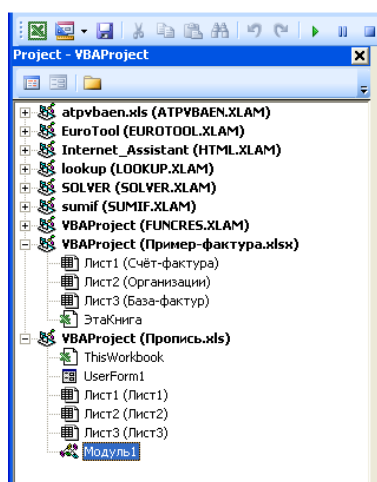


В ячейку в конце документа необходимо вставить формулу подсчёта суммы. Для нашего примера она может выглядеть так: =СУММ(J15:J35).

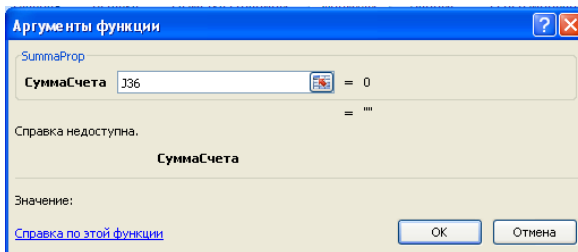
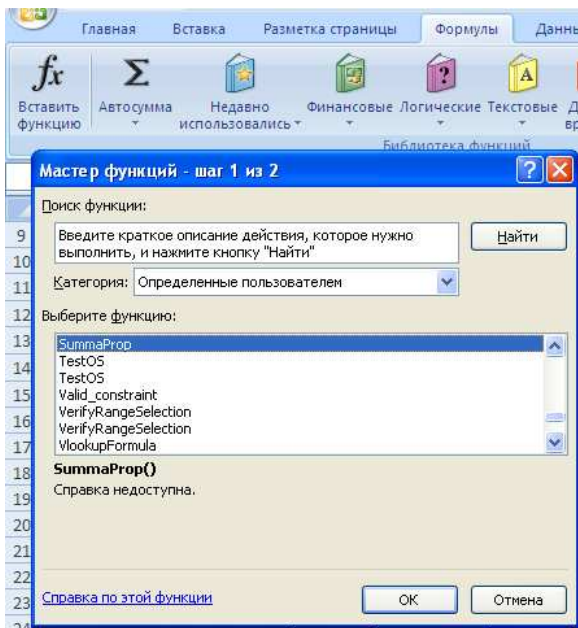
Любой финансовый документ должен содержать сумму прописью, то есть значение суммы должно быть записано словами. Для преобразования числовой суммы в сумму прописью существует много программ. В прилагаемом Excel файле «Пропись.xls» содержится один из существующих вариантов. Вам предлагается скопировать её к себе в книгу и использовать для получения суммы прописью.

Внимание задание! После копирования программы как показано ниже, составьте краткое словесное описание этой программы.

Для копирования программы откройте книгу «Пропись.xls», затем редактор VisualBasic. В окне проектов будут видны структуры обеих книг. Примерно так как показано ниже.



В проекте существует программный модуль «Модуль1», а в проекте «Пример-фактура.xls» его нет. Создайте его и скопируйте содержимое «Модуль1» проекта «Пропись.xls» в «Модуль1» проекта «Пример-фактура.xls». После этого программы суммы прописью будут доступны и вашей книге. Главная подпрограмма этого комплекса программ SummaProp, параметром которой является числовое значение суммы, которую нужно отобразить прописью. Для этого в выбранную ячейку (например B37) необходимо вставить ссылку на функцию SummaProp, передав ей в качестве параметра значение суммы документа хранящегося в ячейке J36. Вставка функции производится из панели «Формулы» кнопкой «Вставить функцию». При этом выбирается список функций определённых пользователем как показано на рисунке. После этого появится окно с запросом суммы счёта. Введите в него адрес ячейки J36 или выберите эту ячейку курсором (следующий рисунок) и нажмите «ОК».



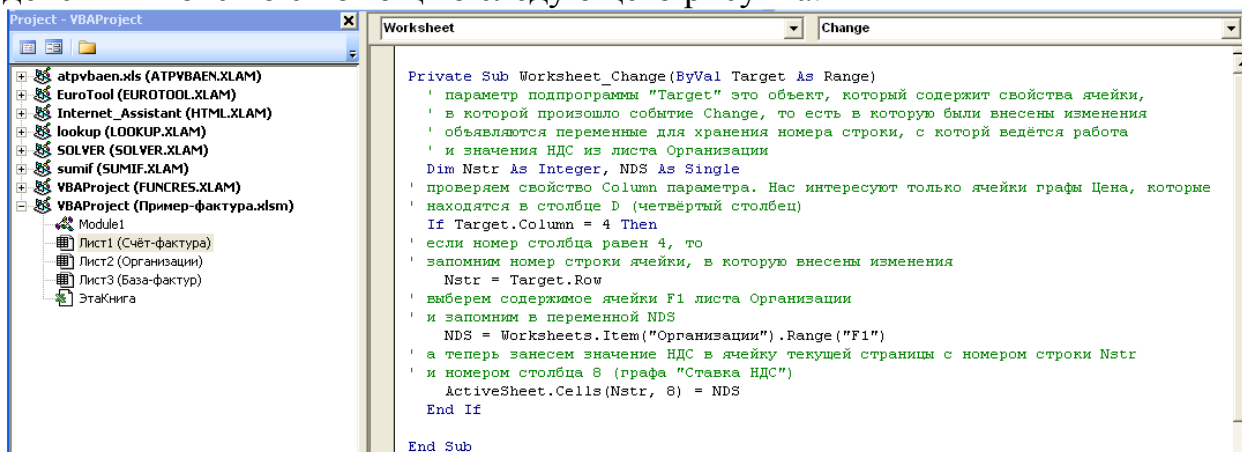
Эта часть работы с счётом-фактурой закончена. Теперь следует проверить работу с документом. Попробуйте занести в него придуманные вами данные и посмотреть результаты. Я занёс в документ две строки и получил следующий результат.

Наименование	Ед. изм.	Количество	Цена	в т. ч. акциз	Сумма	в т. ч. акциз	Ставка НДС	Сумма НДС	Всего с НДС
2	3	4	5	6	7	8	9	10	11
Видео плеер	шт.	15	1500.00		22500				22 500.00
Телефон Samsung	шт.	12	6200.00		74400				74 400.00
									96 900.00
Сумма прописью:	Девяносто шесть тысяч девятьсот рублей 00 копеек								

Анализируя получившийся документ, можно заметить, что в нём отсутствует ставка и сумма НДС. Как было отмечено ранее, ставка НДС задана на листе «Организации», а формула расчёта введена в строки графы «Сумма НДС». Если заранее занести ставку НДС во все строки, то документ будет выглядеть некрасиво, поэтому было бы правильнее заносить ставку в графу во время заполнения строки. В каждую строку счёта-фактуры пользователь заносит наименование, единицу измерения, количество и цену. Остальные графы рассчитываются по формулам. По логике вещей, ставка НДС потребуется для расчёта, когда будет введена цена товара, то есть при заполнении столбца D.

В этот момент необходимо перенести значение НДС из ячейки F1 листа «Организации» в текущую строку графы «Ставка НДС». При решении данной задачи воспользуемся возможностями объектно-ориентированного подхода к программированию в Excel. Наиболее употребительными объектами в Excel являются Application (Приложение), Workbook (Рабочая книга), Worksheet (Рабочий лист) и Range (Диапазон). Каждый из них имеет набор свойств, методов и событий. Для системного изучения работы с этими и другими объектами направляю вас к лекциям **Интернет-Университет Информационных Технологий** <http://www.INTUIT.ru> - **VBA в MS Office 2007**.

Здесь же мы рассмотрим только их использование применительно к решению поставленной задачи. Итак, задача: внести значение НДС хранящееся в ячейке F1 листа «Организации» в ячейку графы «Ставка НДС» в момент, когда заполнена ячейка с ценой товара. Для этого используем механизм обработки события *Change* объекта *Worksheet*. Последовательность действий поясню с помощью следующего рисунка.



В левом окне выбираем «Лист1(Счёт-фактура)», в левом поле со списком вверху выбираем объект *Worksheet*, а в правом событие *Change*. Автоматически создаётся процедура *Private Sub Worksheet_Change(ByVal Target As Range)*, предназначенная для обработки события, которое возникает при изменении данных ячейки. В ней пишем программный код (смотри рисунок), в котором заполняем значение НДС в графе «Ставка НДС». Обращаю внимание, что свойства объекта отделяются точкой. Так свойство *Column* возвращает номер столбца объекта, а *Row* – номер строки. Свойства могут уточняться, как это сделано в операторе: *Worksheets.Item("Организации").Range("F1")*. Здесь из класса объектов *Worksheets* выбирается экземпляр - лист "Организации", а уже из него ячейка F1. Остальное должно быть понятно из прошлых работ.

На этом подготовка счёта-фактуры закончена. Однако задача ставилась шире. Созданную фактуру необходимо: сохранить, при необходимости посмотреть, создать новую счёт-фактуру. Для хранения данных ранее созданных счетов-фактур используем лист «База фактур». На листе будут располагаться две таблицы: «Список счетов-фактур» и «Содержимое счетов-фактур» со следующими заголовками столбцов.

A	B	C	D	E
Номер счёта-фактуры	Организация	Дата создания	Код организации	Границы блока

G	H	I	J
---	---	---	---

Наименование Ед. изм. Количество Цена

Теперь требуется написать фрагменты программ, которые будут заносить данные из счёта-фактуры в базу при сохранении документа и из базы в счёт-фактуру при необходимости просмотра и подготовку документа к вводу нового счёта-фактуры. Эти фрагменты привяжем к соответствующим кнопкам с названиями: «В базу», «Посмотреть» и «Новая». Первый из фрагментов реализуем в виде следующей процедуры.

Public Sub CopyToDB()

' процедура копирования данных из счёта-фактуры в базу

' переменные предназначены для хранения номера первой пустой строки

' различных диапазонов ячеек

Dim NextRow1 As Long, NextRow2 As Long, NextRow3 As Long

' здесь будет формироваться адрес диапазона ячеек

Dim AdrFact As String

' это объектные переменные для хранения диапазонов ячеек

Dim obj_ZaglFact As Range, obj_ContentFact As Range, obj_ContentBase As Range

' выбирается диапазон непустых ячеек листа "База-фактур" прилегающих к ячейке A1
Set obj_ZaglFact = Sheets("База-фактур").Range("A1").CurrentRegion

' процедура Duplicat проверяет есть ли в объекте obj_ZaglFact значение

' содержащееся в ячейке с адресом F1, т.е. сохранена ли уже в базе фактура

' с номером содержащимся в ячейке F1

If Duplicat(obj_ZaglFact, Range("F1")) Then

' если в базе данных о счёте-фактуре нет, то сохраняем в объектной переменной

' непустой интервал ячеек, прилегающих к ячейке G1 листа "База-фактур"

Set obj_ContentBase = Sheets("База-фактур").Range("G1").CurrentRegion

' так же в объектной переменной сохраняем интервал ячеек для хранения данных
"Счёта-фактуры"

Set obj_ContentFact = Sheets("Счёт-фактура").Range("A15:D23")

' теперь определяем номер первой пустой строки в базе

' данная конструкция более подробно описана в методических указаниях

NextRow1 = obj_ZaglFact.Cells(Rows.Count, 1).End(xlUp).Row + 1

' копирование ячеек заголовка счёта-фактуры в область базы фактур

Sheets("Счёт-фактура").Range("F1").Copy obj_ZaglFact.Cells(NextRow1, 1)

Sheets("Счёт-фактура").Range("H1").Copy obj_ZaglFact.Cells(NextRow1, 3)

Sheets("Счёт-фактура").Range("J1").Copy obj_ZaglFact.Cells(NextRow1, 4)

' формируем и вставляем формулу с названием организации по её номеру

obj_ZaglFact.Cells(NextRow1, 2).Formula = "=INDEX(Организации!\$A\$4:\$A\$44," +

obj_ZaglFact.Cells(NextRow1, 4).Address + ",1)"

' формируем адрес диапазона счёта-фактуры содержащего данные для отправки в базу
AdrFact = "\$A\$15:" & Cells(MyLastRow(obj_ContentFact), 4).Address

' определяем первую свободную строку в базе данных для занесения содержимого счёта-фактуры

```

NextRow2 = obj_ContentBase.Cells(Rows.Count, 1).End(xlUp).Row + 1
' копируем содержимое счёта-фактуры в буфер
Sheets("Счёт-фактура").Range(AdrFact).Copy
' вставляем данные из буфера в базу, начиная с первой свободной строки
Sheets("База-фактур").Cells(NextRow2, 7).PasteSpecial Paste:=xlPasteValues
Set obj_ContentBase = Sheets("База-фактур").Range("G1").CurrentRegion
' определяем последнюю строку содержимого базы после переноса данных из счёта-
фактуры
NextRow3 = obj_ContentBase.Cells(Rows.Count, 1).End(xlUp).Row
' формируем и запоминаем адрес диапазона с сохранёнными данными
Sheets("База-фактур").Cells(NextRow1, 5) = Sheets("База-фактур").Cells(NextRow2,
7).Address & ":" & Sheets("База-фактур").Cells(NextRow3, 10).Address
Else
' сообщение, если в базе уже сохранены данные этого счёта-фактуры
MsgBox ("Счёт-фактура с таким номером в базе уже есть")
End If
End Sub

```

В теле процедуры приведены краткие описания операторов. Здесь же поясню общий алгоритм её работы. В таблицу «Список счетов-фактур» нужно занести общие данные о счёте-фактуре (в дальнейшем СФ): номер СФ, дату СФ, код организации и адрес диапазона таблицы «Содержимое счетов-фактур» листа «База-фактур», куда будут скопированы данные СФ. Кроме того в эту таблицу добавлен столбец, который содержит функцию «ИНДЕКС» определяющую организацию по её номеру. Эта формула подробно рассмотрена выше. В таблицу «Содержимое счетов-фактур» заносятся данные СФ, которые нужно сохранить. Перед копированием СФ нужно проверить, есть ли в таблице «Список счетов-фактур» с таким номером. Если есть, выдать об этом сообщение, если нет, выполнить копирование. При копировании сначала в таблицу «Список счетов-фактур» добавляется строка с общими данными СФ. Затем в таблицу «Содержимое счетов-фактур» копируются данные строк «Наименование», «Ед. изм.», «Количество» и «Цена» из СФ.И, наконец, в ячейку столбца «Границы блока» таблицы «Список счетов-фактур» заносится адрес диапазона ячеек таблицы «Содержимое счетов-фактур», в который было произведено копирование. Этот адрес позволит в дальнейшем копировать данные СФ из базы в документ.

К данной процедуре приведу следующие пояснения. Метод *CurrentRegion* позволяет определить объект включающий интервал непустых ячеек окружающих текущую. Так выражение *Sheets("База-фактур").Range("A1").CurrentRegion* задаёт диапазон непустых ячеек окружающих ячейку A1 на листе «База-фактур». Функция *Duplicat(obj_ZaglFact, Range("F1"))* проверяет наличие в базе номера СФ, которую собираются скопировать. Она имеет следующий вид.

```

Public Function Duplicat(BlockCells As Range, MyCell As Range) As Boolean
' функция проверяет наличие в объекте (BlockCells) номера счёта фактуры (MyCell)

```

```

' значение функции устанавливается - ИСТИНА
Duplicat = True
' циклострокамаобъекта
For Each CellRowInBlockCells.Rows
    ' если в первой ячейке строки номер счёта-фактуры совпадает с
    ' переданным в качестве параметра номером
    If CellRow.Cells(1, 1).Value = MyCell.Value Then
        ' значит счёт-фактура с таким номером в базе есть
        ' и функция принимает значение ЛОЖЬ
        Duplicat = False
    ' выходизцикла
    Exit For
End If
Next CellRow
End Function

```

Все пояснения приведены в теле функции.

Следующим моментом, который требует пояснения, это использование выражения `obj_ZaglFact.Cells(Rows.Count, 1).End(xlUp).Row + 1`. Оно находит номер первой свободной строки в диапазоне ячеек. В данном случае объекта `obj_ZaglFact`. Для сравнения приведу функцию, которая делает то же самое.

```

Public Function MyLastRow(BlockCellsAsRange) AsLong
' Функция определения последней строки объекта диапазона ячеек
' цикл по строкам объекта - диапазон ячеек, переданному в качестве параметра
For Each CellMyInBlockCells.Rows
' если первая ячейка строки пуста
If IsEmpty(CellMy.Cells(1, 1)) Then
' вернуть в качестве значения функции номер строки предшествующей пустой
MyLastRow = CellMy.Row - 1
' выйти из цикла
Exit For
End If
Next CellMy
End Function

```

Здесь интерес представляет цикл `ForEach` (Для каждого). Это цикл, который перебирает все экземпляры указанного объекта. Так строку функции `For Each CellMyInBlockCells.Rows` следует понимать так. Объект `BlockCells`, который передан функции в качестве параметра, имеет свойство `Rows` (строки). В указанном цикле переменная `CellMy` это объект, который принимает значение каждой из строк `BlockCells`. Поэтому ссылка `CellMy.Cells(1, 1)` это ссылка на первый столбец первой и единственной строки объекта. Такие циклы широко распространены в объектном программировании.

Вернёмся к процедуре `CopyToDB`. Одним из основных операторов, используемых в процедуре является оператор копирования. Пример из процедуры:

```

Sheets("Счёт-фактура").Range("F1").Copyobj_ZaglFact.Cells(NextRow1, 1)

```

Ячейка F1 листа «Счёт-фактура» копируется в колонку 1 первой пустой строки объекта *obj_ZaglFact*. Недостатком такого способа является то, что копируется не только содержимое ячейки, но и её формат, оформление и т.д. Чтобы скопировать только значения приходится пользоваться таким способом:

```
Sheets("Счёт-фактура").Range(AdrFact).Copy  
Sheets("База-фактур").Cells(NextRow2, 7).PasteSpecial Paste:=xlPasteValues
```

Здесь содержимое диапазона, записанного в переменную *AdrFact* копируется в буфер. Следующим оператором содержимое буфера вставляется (метод *PasteSpecial*) в ячейку столбца 7 первой свободной строки на листе "База-фактур". Последнее пояснение для этой процедуры сделаю к следующему, казалось бы, непонятному оператору.

```
Sheets("База-фактур").Cells(NextRow1, 5) = Sheets("База-  
фактур").Cells(NextRow2,7).Address & ":" & Sheets("База-фактур").Cells(NextRow3,  
10).Address
```

Здесь *NextRow1* переменная с номером строки в таблице «Список счетов-фактур», в которую заносится заголовок копируемого СФ, *NextRow2* – номер строки начала диапазона, *NextRow3* – номер строки конца диапазона таблицы «Содержимое счетов-фактур», в который был занесён скопированный СФ. В результате вычисления выражения справа от оператора присваивания получается символьная строка с адресом диапазона. Например как эта: $\$G\$2:\$J\4 (напомню, что G это столбец 7, а J – 10). Теперь можно создать кнопку и привязать к ней созданную процедуру.

Следующим функционалом нашего проекта должен быть механизм, который позволяет выбрать из базы данные ранее сохранённые СФ и скопировать их в шаблон счёта-фактуры на листе «Счёт-фактура». Этот механизм включает две составляющих: интерфейс позволяющий просмотреть список СФ, хранящихся в базе, и выбрать интересующий; процедуру, которая переносит данные из базы в макет СФ. Интерфейс может быть организован по-разному. Одним из вариантов может быть конструкция из трёх элементов управления «Список», аналогичных полю со списком, описанному ранее. Параметрами списка являются диапазон просматриваемых ячеек и адрес ячейки, в которой указывается номер выбранной строки списка. Фрагмент листа интерфейса приведён ниже. Номер выбранной строки сохраняется в ячейке Q1. Свойства элемента управления – «Список» позволяют выводить на экран только один столбец данных. В то же время для однозначной идентификации СФ необходимы номер фактуры, название организации и дата создания. Поэтому приходится вставлять три списка и привязывать их к одной ячейке – Q1.

О	Р	Q
База фактур		
1		
№	Организация	Дата
3	ДРЭУ Кировского района	5 апр 12
4	ДРЭУ Кировского района	6 апр 12
5	МРЭУ Выборгского района	7 апр 12

Процедура переноса данных в макет СФ выглядит следующим образом.

Public Sub FromBase()

' процедура предназначена для заполнения счёта-фактуры данными из базы

' определены переменные: диапазон списка счетов фактур в базе,

' переменная адреса содержимого счетов-фактур в базе

' и переменная номера строки

Dim obj_ZaglFact As Range, adrCont As String, CountRow As Integer

Dim i As Integer

' установит объектную переменную obj_ZaglFact в диапазон списка счетов-фактур базы

Set obj_ZaglFact = Sheets("База-фактур").Range("A1").CurrentRegion

' запомнить адрес содержимого счетов-фактур в базе (5 столбец таблицы диапазона)

' из строки указанной в ячейке Q1 листа "Счёт-фактура"

adrCont = obj_ZaglFact.Cells(Sheets("Счёт-фактура").Range("Q1") + 1, 5)

' определить число строк в объекте obj_ZaglFact, которое равно числу заполненных строк

' минус одна строка заголовка таблицы

CountRow = obj_ZaglFact.Rows.Count - 1

' теперь очистим содержимое счёта-фактуры

Sheets("Счёт-фактура").Range("A15:D23").ClearContents

Sheets("Счёт-фактура").Range("H15:H23").ClearContents

' и скопируем из базы в документ соответствующие строки ячеек

Sheets("База-фактур").Range(adrCont).Copy

Sheets("Счёт-фактура").Range("A15").PasteSpecial Paste:=xlPasteValues

' запомнить код организации счёта-фактуры в базе (4 столбец таблицы диапазона)

' из строки указанной в ячейке Q1 листа "Счёт-фактура"

adrCont = obj_ZaglFact.Cells(Sheets("Счёт-фактура").Range("Q1") + 1, 4).Address

' скопировать код организации счёта-фактуры из базы в документ

Sheets("База-фактур").Range(adrCont).Copy

Sheets("Счёт-фактура").Range("J1").PasteSpecial Paste:=xlPasteValues

' запомнить адрес номера счёта-фактуры в базе (1 столбец таблицы диапазона)

' из строки указанной в ячейке Q1 листа "Счёт-фактура"

adrCont = obj_ZaglFact.Cells(Sheets("Счёт-фактура").Range("Q1") + 1, 1).Address

```

' скопировать номер счёта-фактуры из базы в документ
Sheets("База-фактур").Range(adrCont).Copy
Sheets("Счёт-фактура").Range("F1").PasteSpecial Paste:=xlPasteValues
' перенести процент НДС из ячейки F1 листа "Счёт-фактура"
' в столбец Ставка НДС (в столбец ячеек, сколько строк в документе)
For i = 15 To 15 + CountRow - 1
    Sheets("Счёт-фактура").Cells(i, 8) = Sheets("Организации").Range("F1")
Next i
End Sub

```

Если вы разобрались с первой процедурой, то здесь придётся затратить гораздо меньше труда. Она привязывается к кнопке «Посмотреть».

И, наконец, ещё один функционал - создание новой СФ. Он должен выполнять следующие действия. Автоматически создавать новый номер счёта-фактуры, вводить текущую дату и очищать макет СФ от предыдущих записей. Процедура имеет следующий вид.

```

Public Sub NewData(BlockBase As Range, BlockFact As Range, BlockNDS As Range,
NumFact As Range, DateCell As Range)
' процедура содания нового счёта-фактуры.
' Параметры: список счётов-фактур в базе (BlockBase),
' диапазон ячеек данных счёта-фактуры (BlockFact)
' диапазон ячеек данных счёта-фактуры с НДС(BlockNDS)
' адрес ячейки счёта-фактуры с номерм счёта (NumFact)
' адрес ячейки счёта-фактуры с датой счёта (DateCell)
' номер счёта заполняется с помощью функции MaxNum
' как максимальный номер в базе плюс 1
NumFact.Value = MaxNum(BlockBase) + 1
' занесение текущей даты в ячейку
DateCell.Value = Date
' очистка содержимого диапазонов данных счёта-фактуры
BlockFact.ClearContents
BlockNDS.ClearContents
End Sub

```

Как видно, процедура имеет несколько формальных параметров, смысл которых указан в комментариях. Для того, чтобы использовать эту процедуру нужно её вызвать с указанием фактических параметров. Это делается с помощью процедуры вызова, имеющей следующий вид.

```

Public Sub NewShet()
' процедура обращения к процедуре NewData с передачей ей соответствующих
параметров
NewDataSheets("База-фактур").Range("A1").CurrentRegion, Sheets("Счёт-
фактура").Range("A15:D23"), Sheets("Счёт-фактура").Range("H15:H23"), Range("F1"),
Range("H1")
End Sub

```

В качестве параметров задаются диапазоны, которые нужно очистить или заполнить в вызываемой процедуре. Эта процедура привязывается к соответствующей кнопке. В процедуре используется функция определения максимального номера счёта -фактуры, которая имеет следующий вид.

```

Public Function MaxNum(BlockNum As Range) As Long
    ' функция находит и возвращает максимальный номер счёта-фактуры
    ' из диапазона, передаваемого в качестве параметра (BlockNum)
Dim Nomer As Long
Nomer = 0
    ' Для каждой строки диапазона BlockNum
For Each CellRow In BlockNum.Rows
    ' проверяем: является ли первая ячейка строки (номер счёта-фактуры) числом
If IsNumeric(CellRow.Cells(1, 1).Value) Then
    ' если да, то больше ли это число того, которое записано в переменной Nomer
If CellRow.Cells(1, 1).Value > Nomer Then
        ' если это так, то в переменную Nomer заносим это число
        Nomer = CellRow.Cells(1, 1).Value
    End If
End If
Next CellRow
    ' возвращаем в качестве значения функции величину из переменной Nomer
MaxNum = Nomer
End Function

```