



Отчеты в Visual Studio

Средства создания отчетов

1. Обычные офисные средства, такие, как Microsoft Office, с интегрированными возможностями для извлечения данных. (VSTO или технология COM)
2. Microsoft Reporting Services
3. Crystal Reports for Microsoft Visual Studio .NET



Технология СОМ

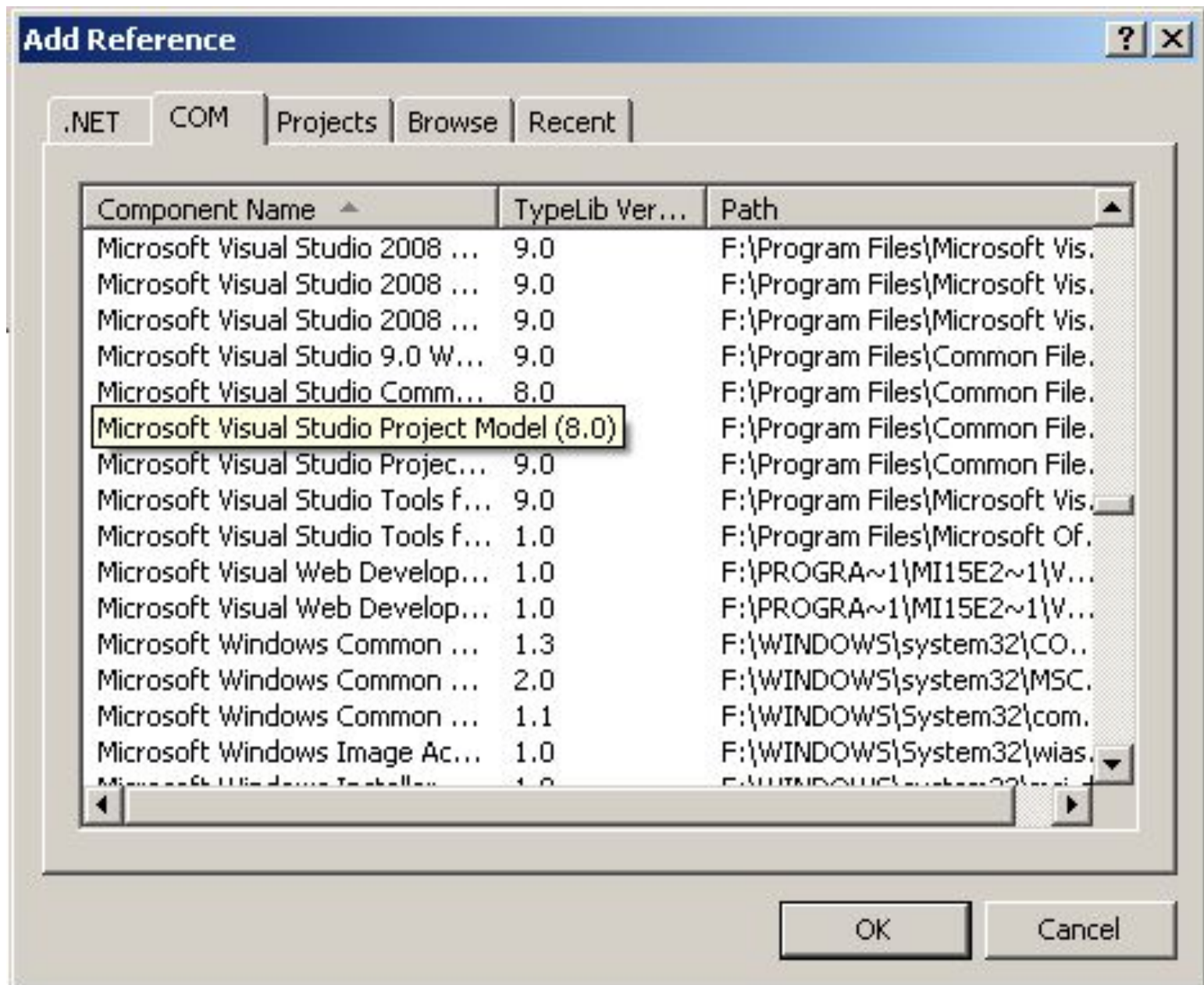
Технология COM

- (Component Object Model — компонентная объектная модель), разработанная компанией Microsoft и применяемая в Visual Basic, позволяет использовать в разработке приложений возможности других программ, поддерживающих данную модель, таких, например, как программы, входящие в пакет Microsoft Office.
- Из приложений, созданных в Visual Basic, можно передавать данные в Excel или Word для формирования документов и их печати, построения диаграмм и графиков, а также получать данные из этих программ.

1. Подключение библиотек

Project □ Add Reference □ закладка COM □
выбрать необходимую библиотеку, например,
Microsoft Excel 11.0 Object Library

(Цифра 11 обозначает установленную на
компьютере версию Microsoft Excel (11
соответствует версии Microsoft Office 2003).)

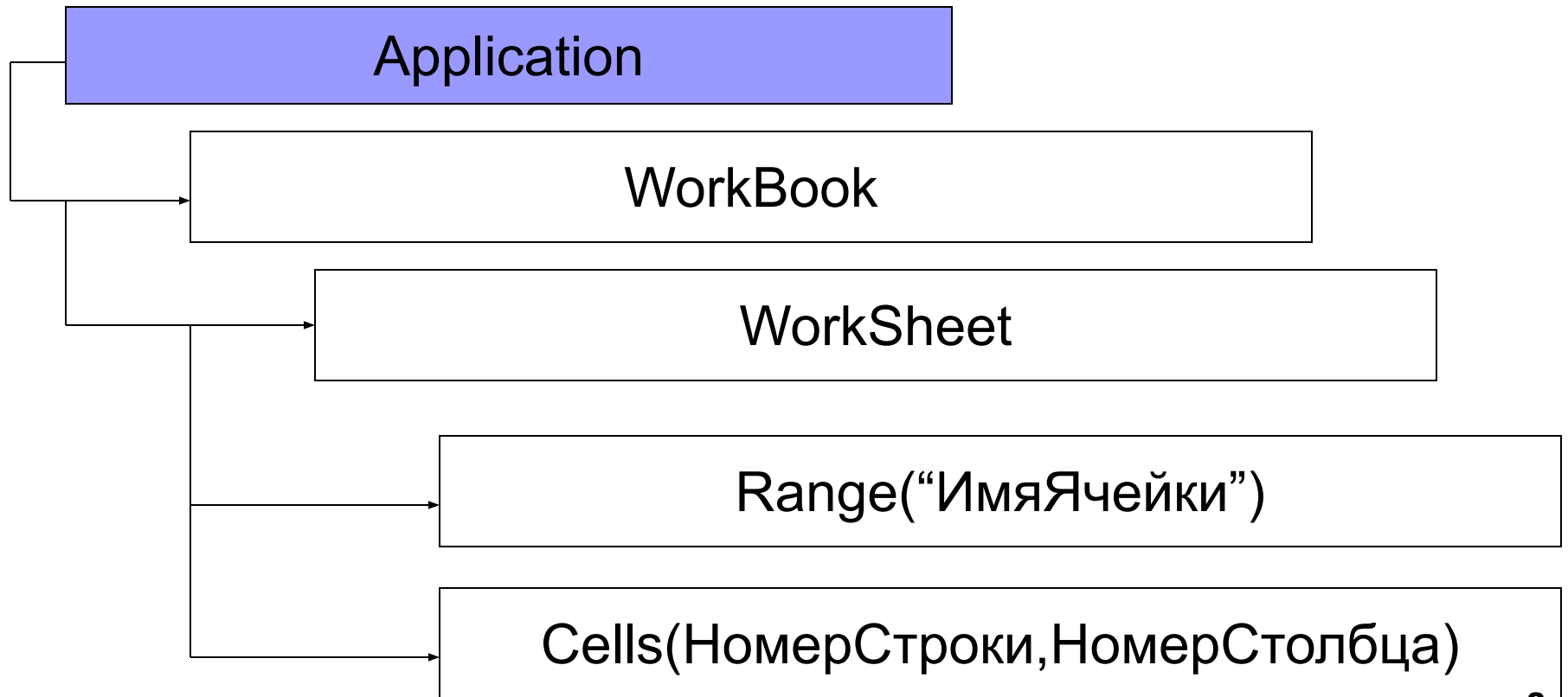


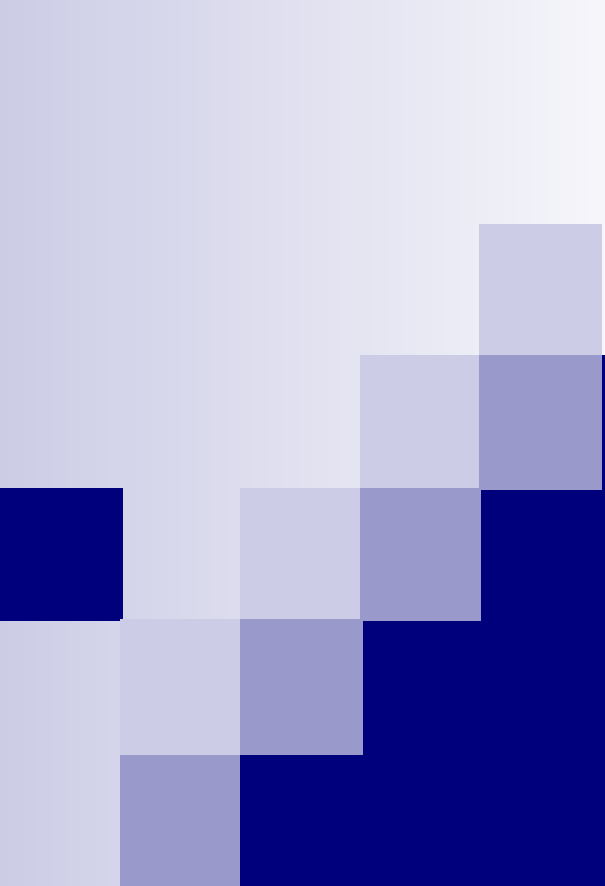
2. Ссылка на пространство имен (для удобства)

- Imports Microsoft.Office.Interop,
где **Interop** (от англ.Interoperation)

3. Создание необходимых объекты – экземпляров класса

- Объектная модель Excel





Объект Application – приложение Excel

Объект Application – свойства

Caption

Возвращает текст в строке имени главного окна Excel. Установка свойства равным Empty возвращает заголовок, используемый по умолчанию. В следующем примере первая инструкция устанавливает в качестве заголовка окна приложения текст Отчет за 1999 год, а вторая возвращает имя окна, используемое по умолчанию, т. е. Microsoft Excel:

```
Application.Caption = "Отчет за 1999 год" Application.  
Caption = Empty
```

DisplayAlerts

Допустимые значения: True (отображаются встроенные предупреждения о работе программы) и False (предупреждения не отображаются)

Height

Высота окна приложения в пунктах

Width

Ширина окна приложения в пунктах

Объект Application –методы

Quit	Закрывает приложение. Например: <code>Application.Quit</code>
------	---

Объект Application - Программно в VisualStudio

```
Dim ЗадайтеИмяПеременнойExcel As  
New Excel.Application
```

```
ИмяПеременнойExcel.Visible = True
```



Объект Workbook – рабочая книга

WorkBook – свойства

Saved

Допустимые значения: True (если не производились изменения в документе со времени его последнего сохранения), False (в противном случае)

WorkBook – методы

Методы	Выполняемые действия
Activate	<p>Активизирует рабочую книгу так, что ее первый рабочий лист становится активным.</p> <p>Например:</p> <pre>Workbook. Activate</pre>
Add	<p>Создает новый объект для семейства <code>Workbooks</code>.</p> <p>Синтаксис:</p> <pre>Add (Template)</pre> <ul style="list-style-type: none">• <code>Template</code> — задает шаблон, на основе которого создается новая рабочая книга. Допустимые значения: <code>xlWBATChart</code>, <code>xlWBATExcel14IntlMacroSheet</code>, <code>xlWBATExcel14MacroSheet</code> или <code>xlWBATWorksheet</code>. <p>Если аргумент <code>Template</code> опущен, то создается новая рабочая книга с количеством листов, заданных свойством <code>SheetsInNewWorkbook</code></p>

Workbook – методы

SaveAs

Сохранение рабочей книги в другом файле.

Синтаксис:

```
SaveAs (Filename)
```

- `Filename` — строка, указывающая имя файла, в котором будет сохранена рабочая книга

В следующем примере активная рабочая книга сохраняется в файле с именем НоваяВерсия:

```
ActiveBook. SaveAs Filename :=  
"НоваяВерсия"
```


Объект Workbook - Программно в VisualStudio

Dim *ЗадайтеИмяПеременнойРабочейКниги*
As Excel.Workbook

ИмяПеременнойРабочейКниги =
ИмяПеременнойExcel.Workbooks.Add



Объект WorkSheet – рабочий лист

WorkSheet – Свойства

Add

Создает новый рабочий лист.

Синтаксис:

Add (Before, After, Count, Type)

- Before — указывает лист, перед которым будет размещен новый рабочий лист
- After — указывает лист, после которого будет размещен новый рабочий. Если аргументы Before и After опущены, то новый лист размещается перед активным листом
- Count — число добавляемых листов, по умолчанию имеет значение 1
- Type — указывает тип добавляемого листа.
Допустимые значения: xlWorksheet (по умолчанию),

xlExcel14MacroSheet и xlExcel14
IntlMacroSheet.

Например:

```
ActiveWorkbook . Worksheets . Add
```

— вставляется новый лист перед активным листом активной рабочей книги

WorkSheet – Свойства

Delete

Удаляет рабочий лист. Например:

Worksheets (1) . Delete - удаляется первый рабочий лист из активной рабочей книги

Объект WorkSheet - Программно в VisualStudio

```
Dim ЗадайтеИмяПеременнойРабочегоЛиста  
As New Excel.Worksheet
```

```
ИмяПеременнойРабочегоЛиста =  
ИмяПеременнойРабочейКниги.Worksheets(1)
```

```
ИмяПеременнойРабочегоЛиста.Activate()
```



Объекты Range, Cell - диапазон ячеек и ячейка

Range – Свойства

Value	<p>Возвращает значение из ячейки или в ячейки диапазона. В данном примере переменной x присваивается значение из ячейки C1 :</p> <p><code>x = Range ("C1") .Value</code> В следующем примере в диапазон A1 : B2 введена 1 :</p> <p><code>Range ("A1:B2") .Value = 1</code></p>
Count	<p>Возвращает число объектов в наборе. В данном примере переменной x присваивается значение, равное числу строк диапазона A1 : B2 :</p> <p><code>x = Range ("A1 :B2") .Rows . Count</code></p>
CurrentRegion	<p>Возвращает число строк текущего диапазона. Текущим является диапазон, ограниченный пустыми строками и столбцами и содержащий данный элемент. В следующем примере переменной y присваивается значение, равное числу строк в текущем диапазоне, содержащем ячейку A1 :</p> <p><code>y = Range ("A1") . CurrentRegion . Rows . Count</code></p>

Range – Свойства

WrapText

Позволяет переносить текст при вводе в диапазон. Допустимые значения True и False. В следующем примере в ячейку B2 вводится текст длинный текст и в этой ; ячейке устанавливается режим ввода текста с переносом:
With Range ("B2") .Value = "Длинный текст" .WrapText = True
End With

Range – Свойства

Font Возвращает объект Font (шрифт). Объект Font имеет следующие свойства:

- **Name** — строка, указывающая имя шрифта, например "Arial Cyr"
- **FontStyle** — СТИЛЬ, возможен Regular (обычный), Bold (ПОЛУЖИРНЫЙ), Italic(курсив), Bold italic (полужирный курсив)
- **size** - размер
- **strikethrough** — допустимы два значения: True (буквы имеют линию по центру, как будто они перечеркнуты) и False (не имеют линии по центру)
- **Superscript** — допустимы два значения: True (текст используется как верхний индекс) и False (не используется как верхний индекс)
- **Subscript** — допустимы два значения: True (текст используется как нижний индекс) и False (не используется как нижний индекс)
- **Underline**-допустимыми являются значения:
 - xlNone (нет подчеркивания)
 - xlSingle (одинарное, по значению)

Объекты Range, Cell - Программно в VisualStudio

ИмяПеременнойРабочегоЛиста.Cells(1, 1) = "Заказы"

*ИмяПеременнойРабочегоЛиста.Range("a1").Value =
"Заказы"*

*ИмяПеременнойРабочегоЛиста.Range("a1:c2").Font.
Bold = True*

*ИмяПеременнойРабочегоЛиста.Range("a1:c2").Font.
Size = 16*

Итого “Создание отчета БД в Excel”

1. Подключить библиотеки Microsoft Excel 11.0 Object Library (Project □ Add Reference □ закладка COM)
2. Добавить ссылку на пространство имен
3. Объявить и создать объекты Excel.Application, Excel.WorkBook, Excel.WorkSheet

4. Программно или мастером создать автономный или подключенный объект ADO для получения данных из БД.
5. Программно выгрузить из объекта ADO данные в Excel-ячейки рабочего листа объекты Range или Cells.
6. Если необходимо вызвать команду для завершения приложения (если вы не хотите, чтобы оно продолжало работать). Например: **xlApp.Quit()**



Пример 1 -

Использованные функции Excel
в приложении VS

Форма

Закладная

Вычисление выплат с помощью Excel:

Проценты

Месяцы

Основной долг

Вычислить

Подключить библиотеки, добавить ссылку на пространство имен

1. Подключить библиотеки Microsoft Excel 11.0 Object Library, System.Math

2. Добавить в код

```
Imports Microsoft.Office.Interop.Excel
```

Обработчик на кн. «Вычислить»

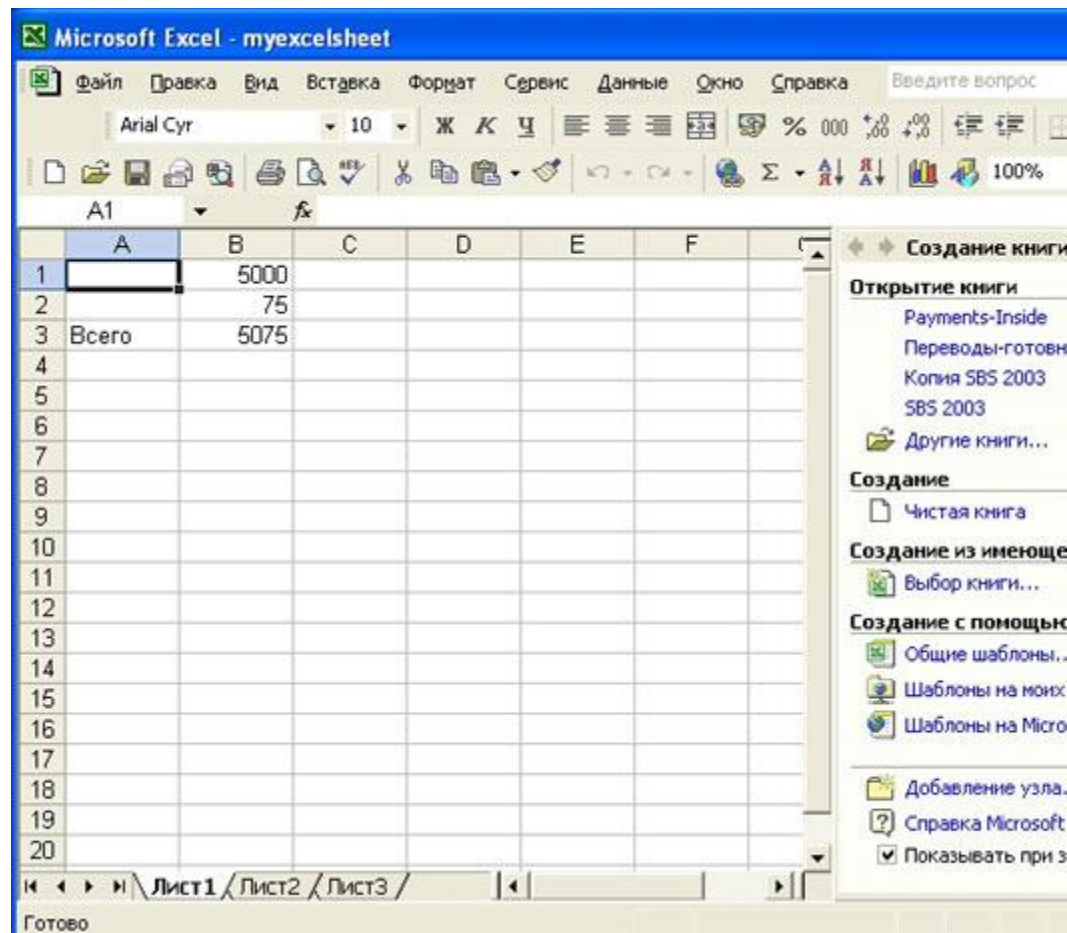
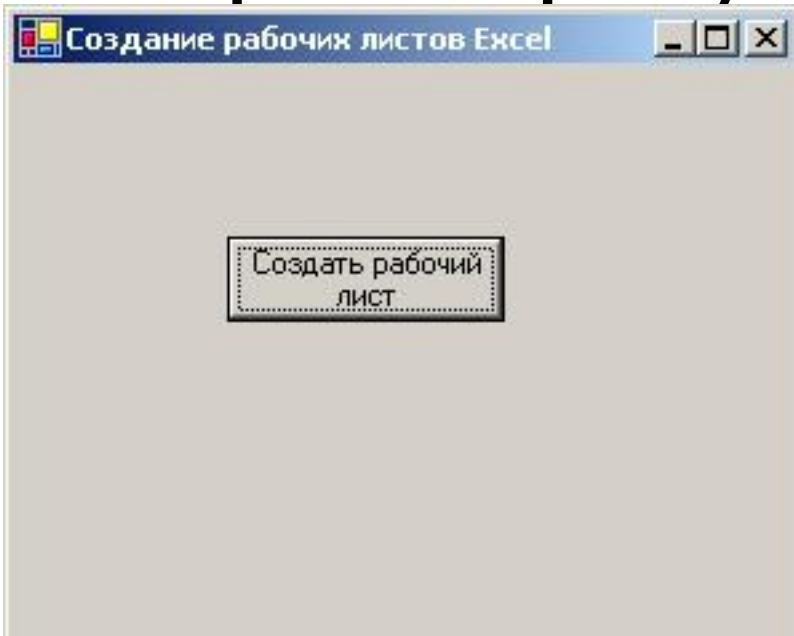
```
Dim xlApp As New Excel.Application
Dim L As Single
L = xlApp.WorksheetFunction.Pmt _
(txtbox1.Text / 12, txtbox2.Text, txtbox3.Text)
MsgBox("Платеж за месяц =" & CStr(L))
xlApp.Quit()
```




Пример 2

Работа с рабочими листами

Форма и результат выполнения



Обработчик на кнопку

```
Dim xlApp As New Excel.Application  
xlApp.Visible = True
```

```
Dim xlBook As Excel.Workbook
```

```
xlBook = apExcel.Workbooks.Add
```

```
Dim xlSheet As New Excel.Worksheet
```

```
xlSheet = xlBook b.Worksheets(1)
```

```
xlSheet.Activate()
```

Обработчик на кнопку

'Добавляем данные

```
xlSheet.Cells(1, 2) = 5000
```

```
xlSheet.Cells(2, 2) = 75
```

```
xlSheet.Cells(3, 1) = "Всего"
```

'Вводим в ячейку B3 формулу Sum

```
xlSheet.Range("B3").Formula = "=Сумм(B1:B2)"
```

'Форматируем ячейку B3 жирным шрифтом

```
xlSheet.Range("B3").Font.Bold = True
```

'Отображаем лист

```
xlSheet.Application.Visible = True
```

'Сохраняем лист в папке c:\vbnet

```
xlSheet.SaveAs("C:\vbnet\myexcelsheet.xls")
```



Пример 3

Создание отчета для
таблицы Модели в Excel

Форма и результат

Form1

Отчет таблица Модели
в Excel

Microsoft Excel - Книга4

Файл Правка Вид Вставка Формат

G8 fx

	А	В	С
1	Таблица модель		
2	Код модели	Название	Цена
3		2 модель01	100
4		3 модель02	200
5		5 модель03	150
6			
7			

Обработчик

```
Dim xlApp As New Excel.Application
```

```
xlApp.Visible = True
```

```
Dim xlBook As Excel.Workbook
```

```
xlBook = apExcel.Workbooks.Add
```

```
Dim xlSheet As New Excel.Worksheet
```

```
xlSheet = xlBook b.Worksheets(1)
```

```
xlSheet.Activate()
```

Обработчик

```
xlSheet.Cells(1, 1).Value = "Таблица модель"
```

'Форматируем ячейку жирным шрифтом

```
xlSheet.Range("a1").Font.Bold = True
```

```
xlSheet.Cells(2, 1) = "Код модели"
```

```
xlSheet.Cells(2, 2) = "Название"
```

```
xlSheet.Cells(2, 3) = "Цена"
```


Обработчик

```
Dim cn As New SqlConnection
    Dim strsql As String
    cn.ConnectionString = " ..."
    strsql = " select * from модель"
    cn.Open()
    Dim cmd As New SqlCommand(strsql, cn)
Dim rdr As SqlDataReader = cmd.ExecuteReader()
```

Обработчик

```
Dim i As Integer = 3
```

```
While rdr.Read
```

'Добавляем данные

```
xlSheet.cells(i, 1) = rdr("код_модели")
```

```
xlSheet.cells(i, 2) = rdr("название_модели")
```

```
xlSheet.cells(i, 3) = rdr("цена")
```

```
    i = i + 1
```

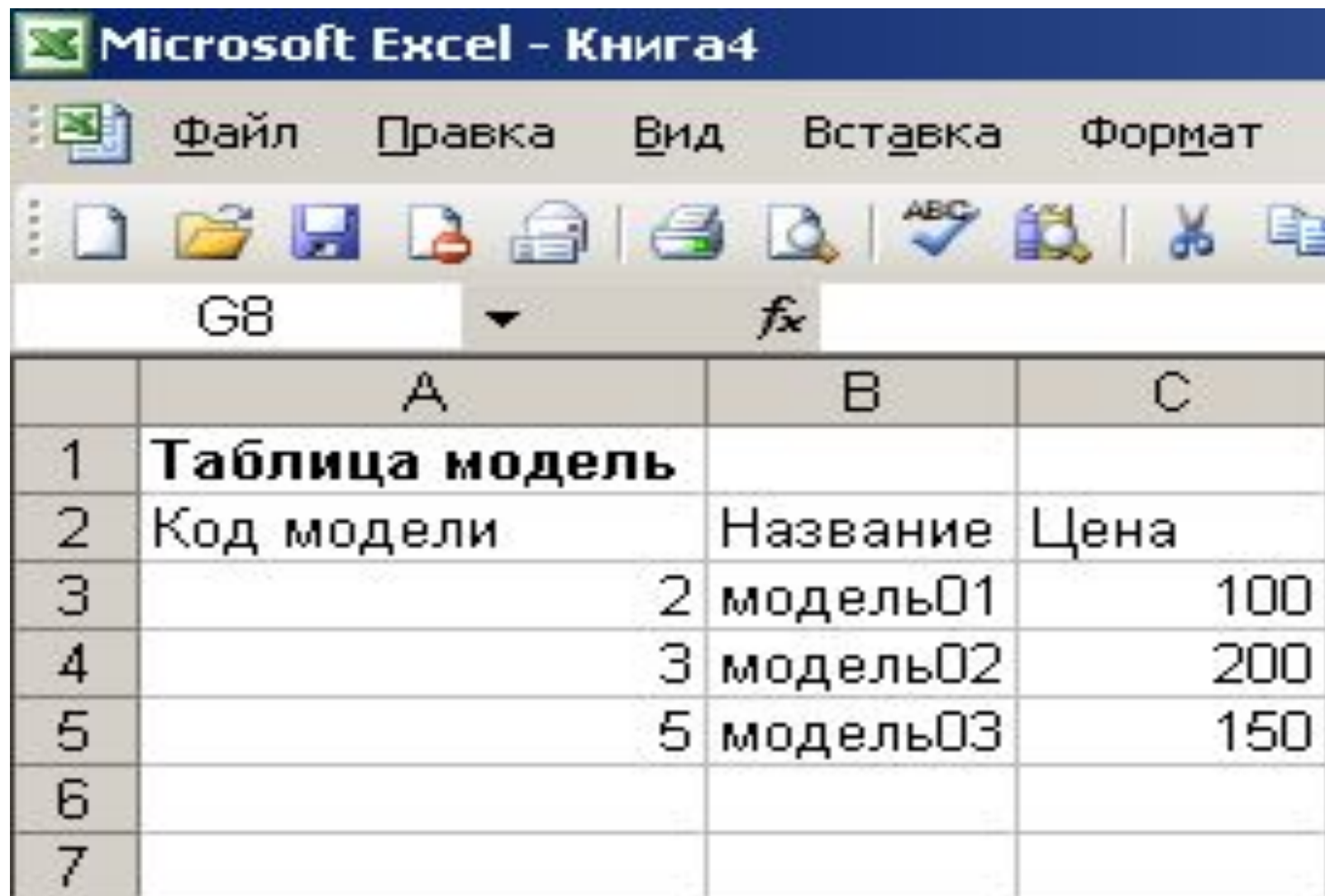
```
End While
```

Обработчик

'Отображаем лист

```
xlSheet.Application.Visible = True  
cn.Close()
```

Результат



Microsoft Excel - Книга4

Файл Правка Вид Вставка Формат

G8 fx

	A	B	C
1	Таблица модель		
2	Код модели	Название	Цена
3	2	модель01	100
4	3	модель02	200
5	5	модель03	150
6			
7			



КОНЕЦ