

Практическая работа № 8.

Тема: *Решение задач в Excel с помощью VBA. Использование встроенных функций.*

Цель: Научиться выполнять вычисления с использованием функций и операторов VBA, организовывать ввод и вывод данных, научиться выполнять отладку программ.

Время: 40 мин.

Задание: Изучите основные функции VBA для работы с различными типами данных, составьте программу для расчёта краткосрочной ссуды.

Литература: 1. Ривкинд Й.Я., Лысенко Т.И., Черникова Л.А., Шакотько В.В. Информатика. 11 класс. Учебник для общеобразовательных учебных заведений. Академический уровень. - К.: Генеза, 2010
2. Берков Н.А. программирования на VISUAL BASIC: Учебное пособие. - М: МГИУ, 2001.
3. Воробьева, Ф.И. Приемы программирования в среде VISUAL BASIC for APPLICATION: учебное пособие / Ф.И. Воробьева, Е.С. Воробьев. - Казань: Изд-во Казан. гос. технол. ун-та; 2010.

Интернет-ссылки:

1. http://msk.edu.ua/ivk/informatika_2k_1s.php

Последовательность выполнения работы:

1. Включите компьютер, войдите в систему и запустите табличный процессор Microsoft Excel.
2. Переименуйте Лист1 в «Ссуда», сохраните рабочую книгу в папке «Мои документы» в формате «Книга Excel с поддержкой макросов»:

Имя файла:	
Тип файла:	Книга Excel с поддержкой макросов (*.xlsm)

В имени файла укажите свою фамилию и номер практической работы, например: «Иванов, пр13».

3. На первом листе («Ссуда») подготовьте таблицу и кнопку-рисунок по образцу:

Ссуда (потребительский кредит)	
Размер ссуды	
Дата выдачи ссуды	
Дата возврата ссуды	
Годовая процентная ставка	
Возвращаемая сумма	
Потери (прибыль банка)	



Рассчитать


Ввод исходных данных	Очистить
-----------------------------	-----------------

Другие элементы управления (кнопки) добавятся чуть позже.

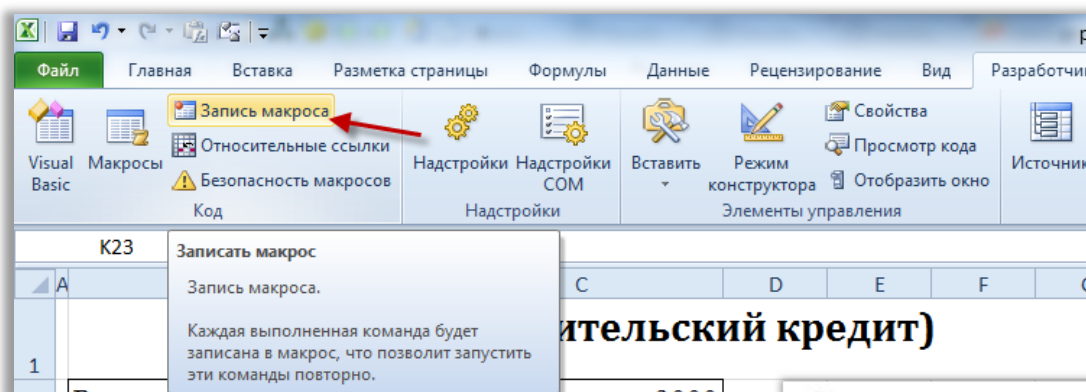
– Для таблицы используйте шрифт *Times New Roman*, размер – 14;

– Рисунок можно использовать свой, или как в образце, его можно [скачать с сайта](#).

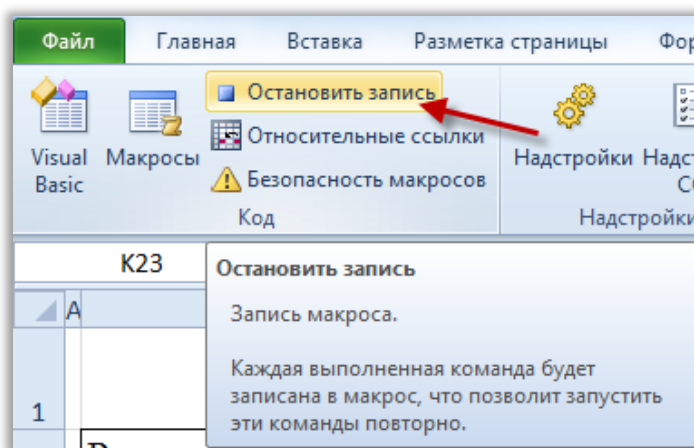
4. Ячейки C2:C7 (см. рисунок ниже) заполните произвольными данными и с помощью макрорекодера создайте макрос, очищающий этот диапазон ячеек. Добавьте кнопку «Очистить» и назначьте этой кнопке созданный макрос.

	A	B	C	D	E	F	G
1		Ссуда (потребительский кредит)					
2		Размер ссуды	3000	 Рассчитать			
3		Дата выдачи ссуды	22.01.2013				
4		Дата возврата ссуды	22.07.2013				
5		Годовая процентная ставка	36				
6		Возвращаемая сумма	3494,15				
7		Потери (прибыль банка)	494,15				
8							
9							
10		Ввод исходных данных		Очистить			
11							
12							

- Для создания макроса воспользуйтесь вкладкой «Разработчик», командой «Запись макроса»:



- Просто нажмите на кнопку и выполняйте нужные действия. Макрорекодер автоматически запишет все действия в виде программного кода.
 – Выделите нужные ячейки и нажмите клавишу «Delete». Переместите курсорную рамку куда-нибудь в сторону и остановите запись.



- После небольшой коррекции у вас должно получиться следующее:

```
Sub Очистить ()
'
' Очистить Макрос
'
    Range ("C2:C7") .ClearContents
End Sub
```

5. Следующая программа берёт данные из таблицы, обрабатывает и записывает результат в ту же таблицу:

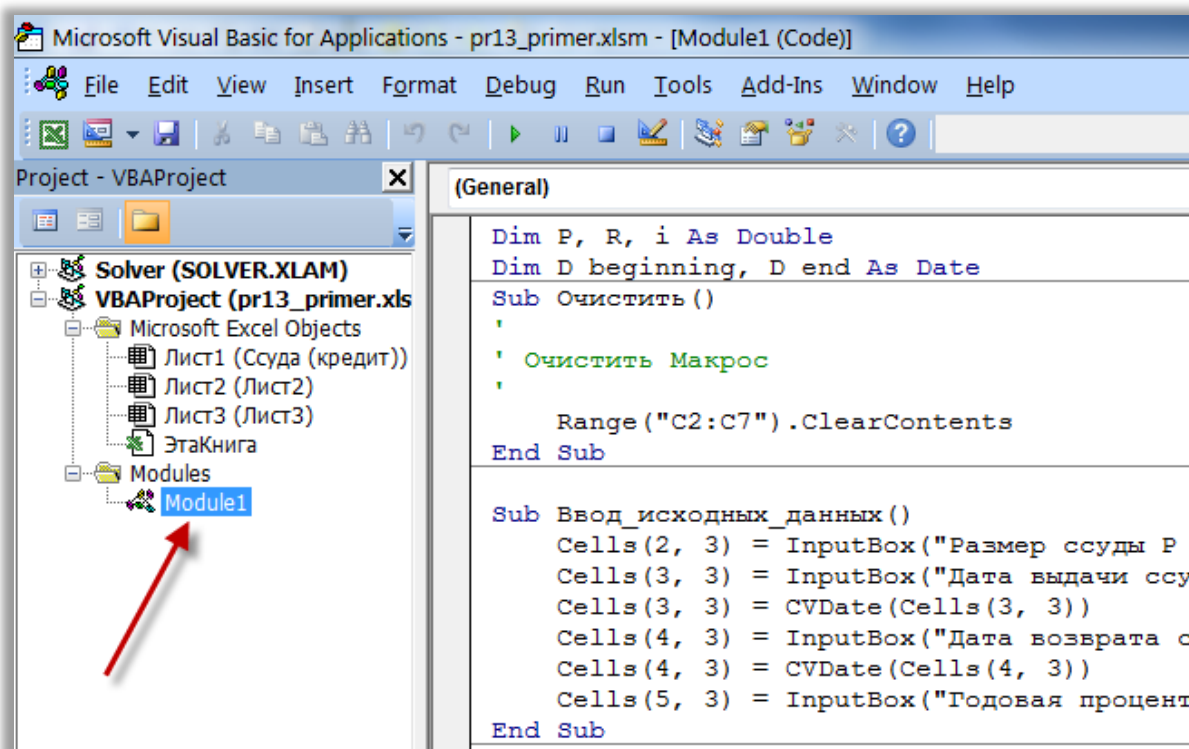
```

Sub Ссуда ()
Dim P, R, i As Double
Dim D_beginning, D_end As Date
P = CDBl(Cells(2, 2))
D_beginning = CDate(Cells(3, 2))
D_end = CDate(Cells(4, 2))
i = CDBl(Cells(5, 2)) / 100
If D_end < D_beginning Then
MsgBox "Ошибка в датах", _
vbExclamation, "Расчет ссуды"
Exit Sub
End If
R = P * (1 + i) ^ ((D_end - D_beginning) / 365)
'R = Format(R, "Fixed")
Cells(6, 2) = CStr(R)
End Sub

```

Часто для заполнения таблицы используют специальные формы или программный код. Так меньше вероятность занесения ошибочных данных. Для ввода текстовых данных используется функция **InputBox**. Создайте процедуру *Ввод_исходных_данных()* с использованием этой функции.

- Синтаксис функции **InputBox** выглядит так:
InputBox (prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])
- Параметры в квадратных скобках – необязательные.



- Процедура создаётся вручную в окне модуля. Можно создать новый модуль (Insert → Module), а можно использовать и тот, который был создан автоматически макрорекодером (Module1);

- Текст программы (процедуры) записывается внутри блока **Sub <имя_процедуры> () ... End Sub**. Сравните вашу процедуру с образцом:

```

Sub Ввод_исходных_данных()
Cells(2, 3) = InputBox("Размер ссуды P = ", "Расчет ссуды", 3000)
Cells(3, 3) = InputBox("Дата выдачи ссуды: ", "Расчет ссуды", "22.01.2013")
Cells(3, 3) = CVDate(Cells(3, 3))
Cells(4, 3) = InputBox("Дата возврата ссуды: ", "Расчет ссуды", "22.07.2013")

```

```
Cells(4, 3) = CDate(Cells(4, 3))
```

```
Cells(5, 3) = InputBox("Годовая процентная ставка, % ", "Расчет ссуды", 36)
```

```
End Sub
```

– *Функция CDate преобразует текст, содержащий дату, в дату (в тип variant)*

6. Добавьте на лист «Ссуда» ещё одну кнопку и назначьте ей макрос «Ввод исходных данных».
7. Составьте программу для расчёта краткосрочной ссуды, полученный макрос свяжите с кнопкой-рисунком на листе «Ссуда» рабочей книги Excel.

Образец:

```
Dim P, R, i As Double
Dim D_beginning, D_end As Date
Sub Очистить()
'
' Очистить, Макрос
'
Range("C2:C7").Select
Selection.ClearContents
Range("C2").Select
End Sub

Sub Ссуда()
If Cells(2, 3) = Empty Then           'Проверка, заполнена ли ячейка
MsgBox "Не введена сумма ссуды", vbExclamation, "Расчет ссуды"
Exit Sub
End If

P = CDBl(Cells(2, 3))
D_beginning = CDate(Cells(3, 3))
D_end = CDate(Cells(4, 3))
i = CDBl(Cells(5, 3)) / 100

If D_end < D_beginning Then
MsgBox "Ошибка в датах", _
vbExclamation, "Расчет ссуды"
'Если оператор слишком длинный,
'можно разместить его в двух или более строках,
'добавив в конце каждой из строк
'(кроме последней) символ подчеркивания ( _ )

Exit Sub
End If
R = P * (1 + i) ^ ((D_end - D_beginning) / 365)
R = CDBl(Format(R, "Fixed"))
Cells(6, 3) = R
Cells(7, 3) = R - P: Cells(7, 3).Font.Bold = True
'Можно сделать и наоборот —
'разместить несколько операторов в одной строке программного кода.
'Для этого применяется знак :
Debug.Print "D_beginning = "; D_beginning
Debug.Print "D_end = "; D_end
Debug.Print "D_end - D_beginning = "; (D_end - D_beginning)
Debug.Print "i = "; i
Debug.Print "R = "; R

End Sub
```

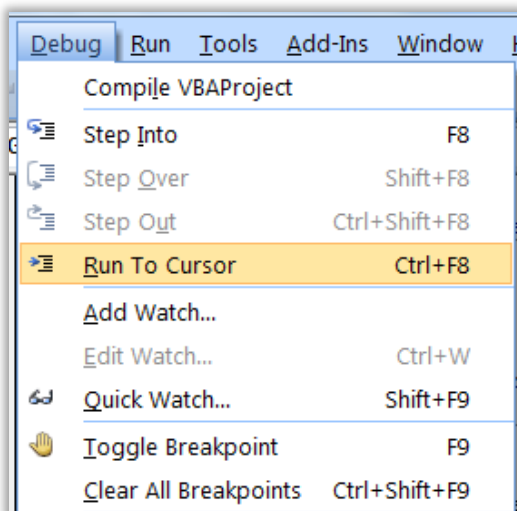
- Этот текст (из образца) можно просто скопировать в модуль VBA Excel. Транслятор VBA проверит код и преобразует его к виду:

```
Sub Ссуда()  
If Cells(2, 3) = Empty Then 'Проверка, заполнена ли ячейка  
    MsgBox "Не введена сумма ссуды", vbExclamation, "Расчет ссуды"  
    Exit Sub  
End If  
  
P = CDb1(Cells(2, 3))  
D_beginning = CDate(Cells(3, 3))  
D_end = CDate(Cells(4, 3))  
i = CDb1(Cells(5, 3)) / 100  
  
If D_end < D_beginning Then  
    MsgBox "Ошибка в датах", _  
        vbExclamation, "Расчет ссуды" 'Если оператор слишком длинный,  
        'можно разместить его в двух или более строках,  
        'добавив в конце каждой из строк  
        '(кроме последней) символ подчеркивания (_)  
    Exit Sub  
End If  
  
R = P * (1 + i) ^ ((D_end - D_beginning) / 365)  
R = CDb1(Format(R, "Fixed"))  
Cells(6, 3) = R  
Cells(7, 3) = R - P: Cells(7, 3).Font.Bold = True 'Можно сделать и наоборот –  
        'разместить несколько операторов в одной строке программного кода.  
        'Для этого применяется знак :  
Debug.Print "D_beginning = "; D_beginning  
Debug.Print "D_end = "; D_end  
Debug.Print "D_end - D_beginning = "; (D_end - D_beginning)  
Debug.Print "i = "; i  
Debug.Print "R = "; R  
  
End Sub
```

- Синим цветом выделяются служебные слова VBA, встроенные функции, стандартные операторы и инструкции языка.

Debug.Print служит для отображения значения выражения в специальном окне **Immediate** в процессе выполнения программы.

Выводить в это окно можно любые допустимые выражения, включая свойства объектов. Для того чтобы вывести значение переменной, нужно ввести знак "?" и имя переменной и нажать клавишу <Enter>. Значение переменной отобразится на следующей строке. При этом программа должна выполняться в режиме отладки:



Высокий уровень (дополнительное задание):

8. Составьте программу для расчёта потребительского кредита, предлагаемого Приват-банком:

- 1) Первые 55 дней – беспроцентный кредит;
- 2) Ставка – 30% годовых;
- 3) Ежемесячное обязательное погашение (после 55 дней) – не менее 7% от оставшейся суммы (2,5% - банку, остальное – погашение долга).
- 4) 30-го числа каждого месяца – день расчёта, когда начисленные проценты добавляются к сумме долга. А начисляются они каждый день на ту сумму, которую должен клиент. Их можно рассчитать по формуле

$$R = P * (1 + i)^{((D_end - D_beginning) / 365)},$$

где $i=30$ (% ставка),

P - долг, взятая сумма,

D_end - 30-е число,

$D_beginning$ - 30 число предыдущего месяца или дата взятия кредита.

Если происходит платёж, этот интервал делится на два, в формуле для второго интервала вместо P будет фигурировать $P - P_1$, где P_1 – сумма платежа.

Рассчитайте, сколько вы заплатите сверх взятой на один год суммы, если взяли 8000 гривен в начале года и погашаете кредит ежемесячно, в начале каждого месяца, начиная со второго (скажем, 10-го числа), путём перечисления фиксированной суммы 700 гривен. Когда вы полностью выплатите кредит? Каким будет последний платёж?

Краткая справка:

Применение операторов ветвления в VBA.

Ветвление реализуется с помощью оператора **IF** и инструкции **Select Case**.

Оператор ветвления IF.

Блочные структуры **IF** – наиболее эффективные из структур логического ветвления. Их можно применять для создания логических структур практически любой сложности.

В общем виде эта структура выглядит следующим образом:

```
IF логическое_выражение THEN  
    блок операторов  
END IF
```

Когда *логическое выражение* принимает значение *истина* (true), то выполняется блок операторов, заключенный между операторами **IF** и **END IF**. Если *логическое выражение* принимает значение *ложь* (false), то этот блок не выполняется и управление переходит к оператору, следующему за оператором **END IF**.

Структура **IF** может содержать оператор **ELSE**:

```
IF логическое_выражение THEN  
    блок операторов  
ELSE  
    блок операторов  
END IF
```

Управление переходит к блоку операторов, следующему за **ELSE** в том случае, если *логическое выражение* принимает значение *ложь* (false).

Это был пример одноблочной структуры оператора **IF**. Одноблочная структура оператора **IF** преобразуется в многоблочную путем добавления оператора **ElseIf**. В этом случае структура выглядит следующим образом:

```
IF логическое_выражение_1 THEN  
    блок операторов 1  
ElseIf логическое_выражение_2 THEN  
    блок операторов 2
```

```
ElseIf логическое_выражение_3 THEN  
    блок операторов 3  
ELSE  
    блок операторов4  
END IF
```

Когда встречается многоблочная структура IF, VB определяет значение логического выражения 1. Если оно равно **true**, то выполняется блок операторов 1, если **false**, то блок операторов 1 полностью пропускается и проверяется значение *логического выражения 2* и т.д. Если ни одно из поставленных условий не выполняется, то управление передается блоку операторов 4.

В приведенном ниже примере показано использование как блоковой, так и однострочной форм инструкции If...Then...Else:

```
Dim Number, Digits, MyString  
Number = 53 ' Инициализирует переменную.  
If Number < 10 Then  
    Digits = 1  
ElseIf Number < 100 Then  
    ' Условие является истинным, поэтому выполняется следующая инструкция.  
    Digits = 2  
Else  
    Digits = 3  
End If  
' Использует однострочную форму для присвоения значения.  
If Digits = 1 Then MyString = "Один" Else MyString = "Больше 1"
```