КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ

Р. Г. ТАХАВУТДИНОВ

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

ЧАСТЬ III:
ПРОЦЕДУРЫ, МАССИВЫ И ЦИКЛЫ

Методические указания к лабораторным работам, практическим занятиям, расчетному заданию и самостоятельной работе студентов по дисциплинам «Информатика», «Программные и аппаратные средства информатики», «Алгоритмические языки и программирование», «Вычислительная техника и программирование»

Казань 2008

УДК 516.62(077)
ББК 31.31
М37

Тахавутдинов Р.Г.

Основы алгоритмизации и программирования. Часть III: Процедуры, массивы и циклы. Методические указания к лабораторным работам, практическим занятиям, расчетному заданию и самостоятельной работе студентов по дисциплинам «Информатика», «Программные и аппаратные средства информатики», «Алгоритмические языки и программирование», «Вычислительная техника и программирование». Казань: Казан. гос. энерг. ун-т, 2008.

Содержатся рекомендации по выполнению компьютерного практикума, который предусмотрен Государственным образовательным стандартом. Предназначены для студентов всех специальностей при изучении дисциплин «Информатика», «Программные и аппаратные средства информатики», «Алгоритмические языки и программирование», «Вычислительная техника и программирование», а также других информационно-ориентированных дисциплин. Могут быть использованы в ходе лабораторных работ, практических занятий, при выполнении расчетного задания и для самостоятельной работы студентов.

ВВЕДЕНИЕ

Современное состояние развития программных средств и вычислительной техники позволяет решать многие практически важные задачи в самых различных областях с использованием готовых программных комплексов. При этом зачастую пользователю нет необходимости самому составлять методику решения, алгоритм и программу на каком-либо искусственном языке, как это было десятилетие назад. Вместе с тем, нередко встречаются ситуации, когда эти программные комплексы не предоставляют стандартных опций для решения каких-либо специфических задач пользователя, или же эти стандартные опции являются неудобными для решения конкретных проблем. Поэтому практически во всех программных комплексах предусмотрена возможность их дополнения собственными модулями пользователя, написанными на каком-либо языке программирования. Таким образом, эти программные комплексы, как правило, имеют свою интегрированную среду программирования или же свой язык макрокоманд, с использованием которых пользователь может ***автоматизировать*** работу с этими программными комплексами применительно к собственным задачам, или же ***расширить возможности*** программных комплексов, вводя туда свои собственные модули. Поэтому наряду с умением применять готовые программные продукты, современные пользователи должны обладать также навыками алгоритмизации и программирования, и понимать основные принципы алгоритмизации, которые являются общими для всех искусственных языков и программных сред.

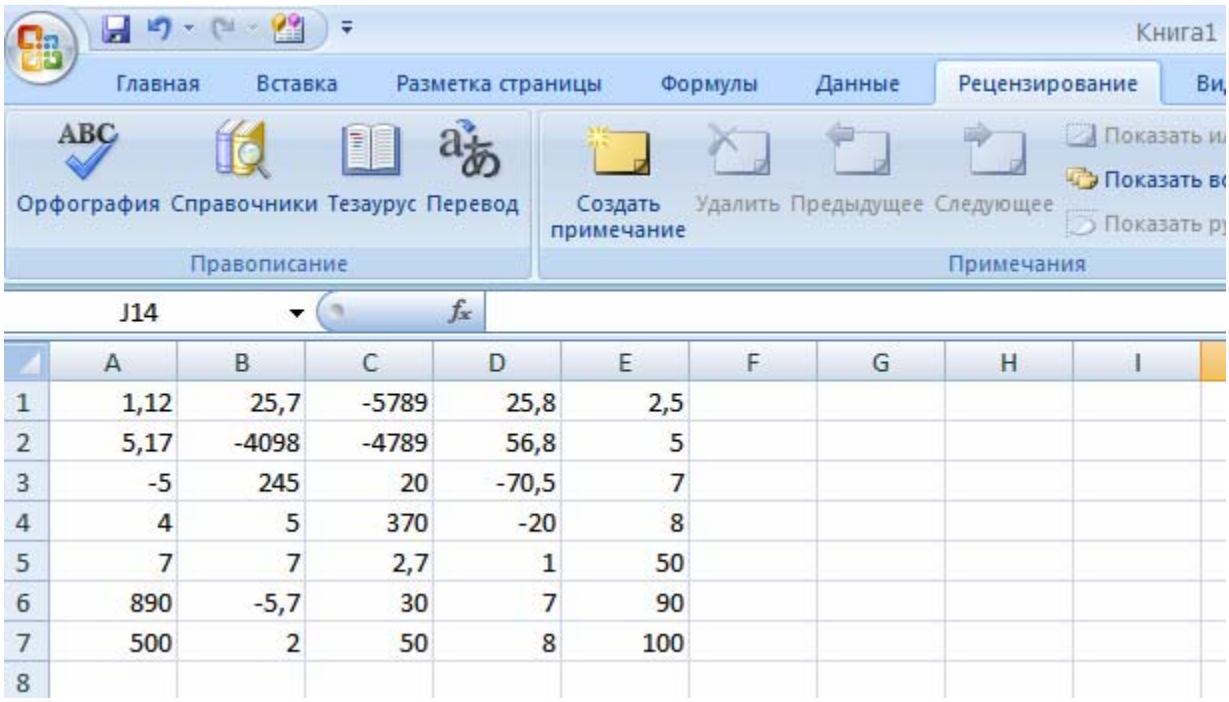
Основной ***целью*** работ, которые изложены в данных методических указаниях, является содействие студентам в приобретении основных навыков алгоритмизации и программирования. При этом также решаются ***задачи*** овладения основами автоматизации работы с документами и расширения возможностей программных продуктов путем создания дополнительных модулей на алгоритмическом языке программирования.

В данном компьютерном практикуме предусмотрено обучение основам алгоритмизации и программирования в рамках программного комплекса Microsoft Office с использованием интегрированного с ним языка программирования V Basic for Applications (сокращенно общепринята аббревиатура VBA). Таким образом, при прохождении компьютерного практикума как бы стирается грань между использованием готового программного комплекса и программированием: пользователи программируют, находясь в рамках готового программного комплекса, и

используют программирование для расширения возможностей этого программного комплекса и автоматизации работы с документами. Данный подход, вне всякого сомнения, является передовым и полностью соответствует современному состоянию развития вычислительной техники и программных средств. Несомненным достоинством принятого подхода является также то, что преподавание компьютерного практикума по дисциплине «Информатика» и другим информационно-ориентированным дисциплинам осуществляется в единой канве, включая как изучение офисных программных средств, так и алгоритмизацию и программирование.

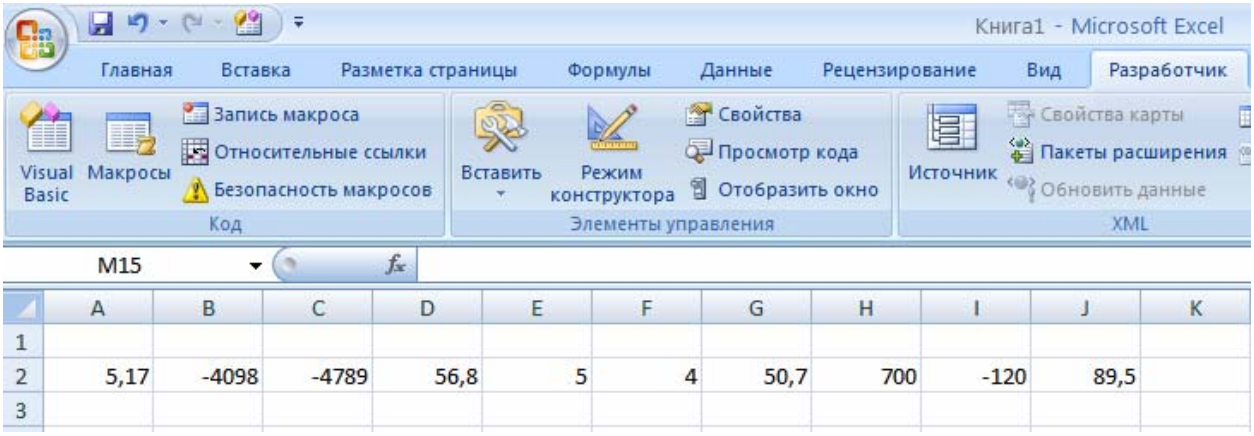
МАССИВЫ

В программировании под *массивом* понимают упорядоченную структуру однородных данных. Наиболее простым примером реализации массива является обычная таблица в листе Excel:



	A	B	C	D	E	F	G	H	I
1	1,12	25,7	-5789	25,8	2,5				
2	5,17	-4098	-4789	56,8	5				
3	-5	245	20	-70,5	7				
4	4	5	370	-20	8				
5	7	7	2,7	1	50				
6	890	-5,7	30	7	90				
7	500	2	50	8	100				
8									

Приведенная выше таблица представляет собой *двумерный массив*, каждая ячейка которого имеет числовой номер строки и буквенный указатель столбца. Если же заполняется лишь одна строка, или же столбец, то это будет иллюстрацией *одномерного массива*:



В модулях VBA вводимые пользователем массивы должны быть объявлены в программе посредством указания размерности массива оператором *Dim*. Например, *Dim a(78)* является объявлением одномерного массива *a*, состоящего из 78 элементов; *Dim a(15,10)* является объявлением двумерного массива, состоящего из 15 строк и 10 столбцов.

Естественно, что массивы могут быть не только одномерными и двумерными. Каждый элемент многомерного массива $a_{i,j,k,l,p,\ldots}$ нумеруется значением его целочисленных индексов *i, j, k, l, p,…*Поэтому массивы можно трактовать как *индексированные переменные*.

ПРОЦЕДУРЫ *Sub*

Мы уже знакомы с процедурой типа *Function* на примерах создания пользовательских функций. Корме процедуры *Function*, в VBA имеется процедура типа *Sub*, представляющая собой именованную часть кода, выполняющую определенные действия. Процедура данного типа пишется в стандартном модуле редактора VBA и имеет синтаксис:

Sub *name* ([arglist])
[Statements]
End Sub

Где *name*- имя процедуры, arglist- список аргументов, Statements – группа инструкций. (Перед словом Sub может находиться указатель, например, Public, Private и др., определяющий область видимости, но в рамках данного лабораторного практикума такие задания не рассматриваются). Важной особенностью является обязательное наличие открывающейся и закрывающейся скобки соответственно до и после перечня аргументов arglist, если даже список аргументов пуст, то есть процедура не имеет аргументов.

ПЕРЕДАЧА ДАННЫХ МЕЖДУ МОДУЛЕМ VBA И ЛИСТОМ EXCEL

Мы уже знакомы с автоматической передачей данных между листом Excel и модулем VBA при вызове функции, определенной пользователем. Если же используется процедура *Sub*, то для записи вычисляемых при выполнении процедуры значений переменных в ячейки листа Excel можно воспользоваться методом *Cells*.

Например:

Cells(i,j)=var - передача значения переменной *var* из модуля VBA в ячейку активного листа Excel, расположенную в *i*-той строке и в *j*-ом столбце;

ActiveCell=var - передача значения переменной *var* из модуля VBA в *активную ячейку* листа Excel.

Абсолютно аналогично осуществляется чтение данных из листа Excel в модуль VBA:

var=Cells(i,j) - присвоение содержимого *i,j*-ой ячейки активного листа Excel переменной *var*;

var=ActiveCell - присвоение содержимого активной ячейки листа Excel переменной *var* модуля VBA.

(Другим средством, используемым для этих целей, является метод *range*, но в данном лабораторном практикуме задания с использованием метода *range* не рассматриваются)

Эти и другие особенности использования массивов, процедур и передачи данных между модулем VBA и листом Excel станут более понятны при выполнении заданий. Поэтому некоторые задания приведены с подробным ходом решения, и студенты должны лишь воспроизвести его на персональном компьютере. Другие задания даются без описания хода решения, или же с частичными пояснениями, поэтому студенты должны выполнить их самостоятельно в аудитории.

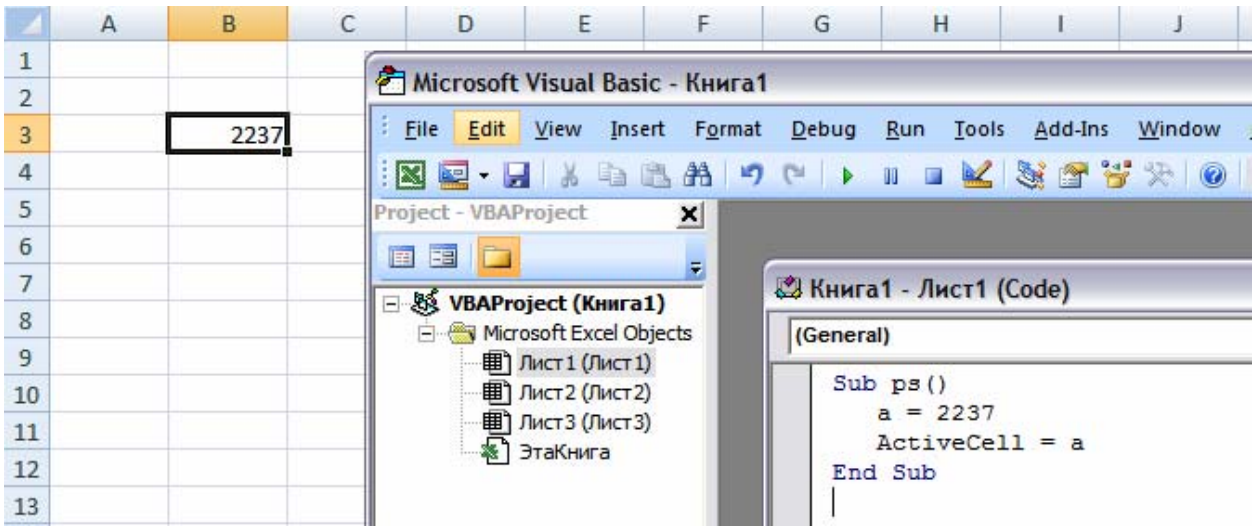
ЗАДАНИЕ 1. Создайте модуль на VBA, представляющий собой процедуру *Sub*, внутри которого переменной *a* присваивается значение 2237, а затем значение переменной *a* передается в активную ячейку листа Excel.

РЕШЕНИЕ ЗАДАНИЯ 1: Вызовем Excel и создадим новую книгу. Выберем щелчком мыши любую ячейку в качестве активной. Активную ячейку легко узнать по жирному обрамлению ее границ.

Вызовем редактор VBA: ***Разработчик> Visual Basic***, или же нажатие комбинации клавиш Alt и F11. Добавим стандартный модуль: ***Вставка> Модуль (Insert>Module)***. В появившемся окне напишем следующий текст программы:

```
Sub ps()
  a = 2237
  ActiveCell = a
End Sub
```

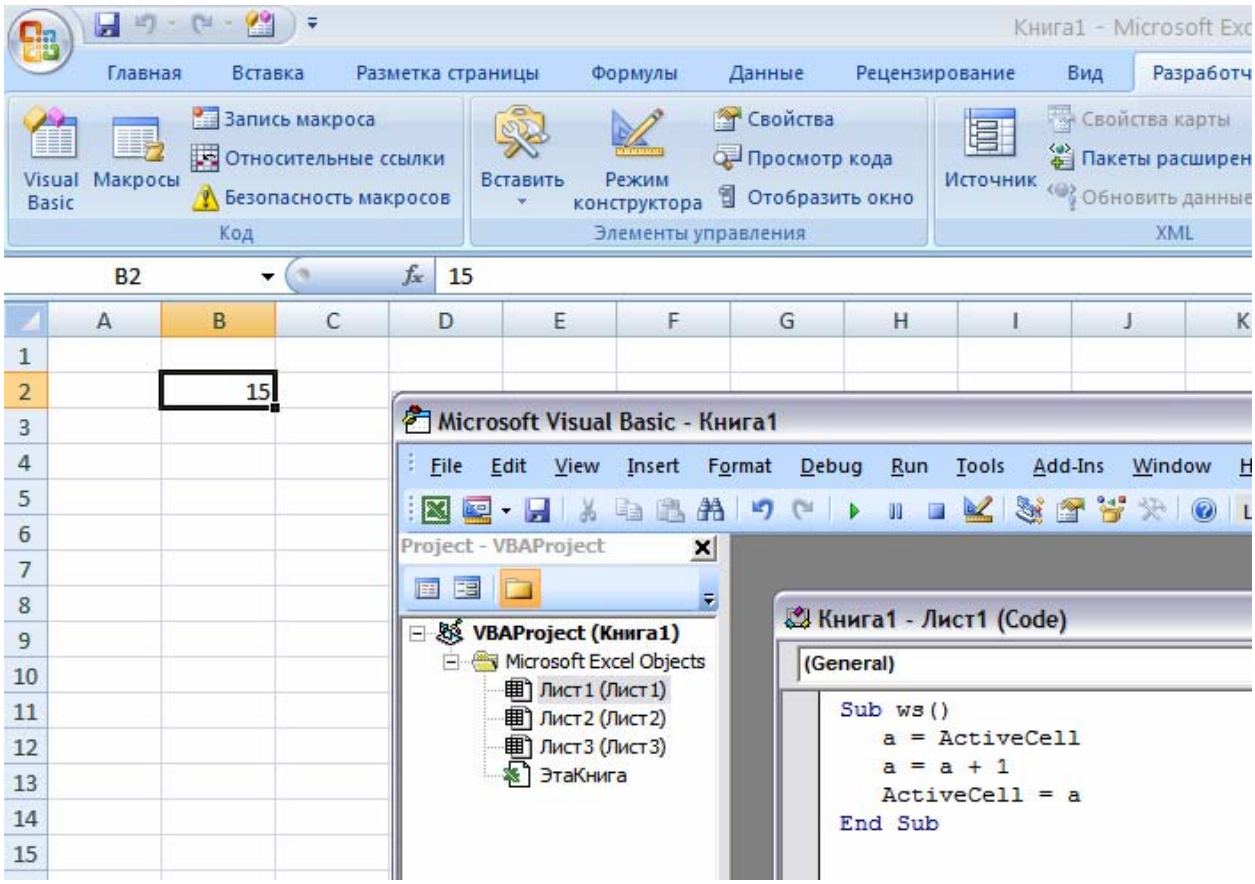
Теперь необходимо запустить выполнение данной программы. Для этого достаточно нажать на клавиатуре функциональный клавиш *F5*, или же мышью на треугольник . После исполнения программы активная ячейка листа Excel будет заполнена переданным из модуля VBA значением переменной *a,* то есть в нашем случае в ячейку *C2* внесется число 2237, и лист примет следующий вид:



Проверьте правильность выполненного Вами задания.

ЗАДАНИЕ 2. Создайте модуль на VBA, представляющий собой процедуру *Sub*, которая вначале присваивает переменной *a* число, содержащееся в активной ячейке листа Excel, затем значение переменной *a* увеличивает на 1, и наконец, передает это новое значение переменной *a* в активную ячейку листа Excel.

РЕШЕНИЕ ЗАДАНИЯ 2. Вид окон:



После нажатия клавиша *F5* и исполнения программы число 15 в активной ячейке увеличится на 1 и станет равным 16. Запускайте эту программу многократно, каждый раз нажимая F5, и убедитесь в том, что значение числа в активной ячейке каждый раз увеличивается на 1, то есть последовательно принимает значения ряда 16, 17, 18, 19, 20 и т.д.

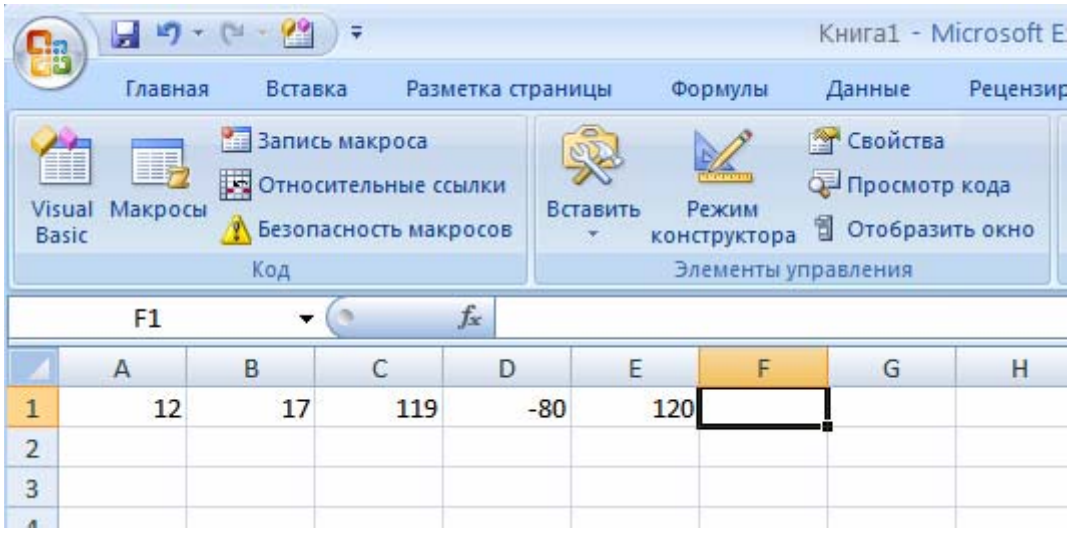
ЗАДАНИЕ 3. Создайте модуль на VBA, представляющий собой процедуру *Sub*, внутри которого переменной *a* присваивается значение 2237, а затем значение переменной *a* передается в ячейку листа Excel, расположенную в 5-ой строке 7-ом столбце.

РЕШЕНИЕ ЗАДАНИЯ 3. Программу модуля можно написать в следующем виде:

```
Sub ws()
  a = 2237
  Cells(5, 7) = a
End Sub
```

Запустите данный модуль и проверьте, что в результате его исполнения число 2237 занеслось в ячейку листа Excel, расположенную в 5-ой строке 7-ом столбце.

ЗАДАНИЕ 4. Создайте в Excel строку из 5 чисел. В модуле VBA составьте программу, которая объявляет массив *a* размерности *Dim a(5)*, считывает его значения из только что созданной таблицы Excel, преобразовывает значение каждого элемента по формуле *a(j)=a(j)+1*, и затем преобразованный массив передает во вторую строку таблицы Excel.
РЕШЕНИЕ ЗАДАНИЯ 4. Введем в ячейки первой строки следующие числа:



В стандартном модуле VBA напишем следующую процедуру:

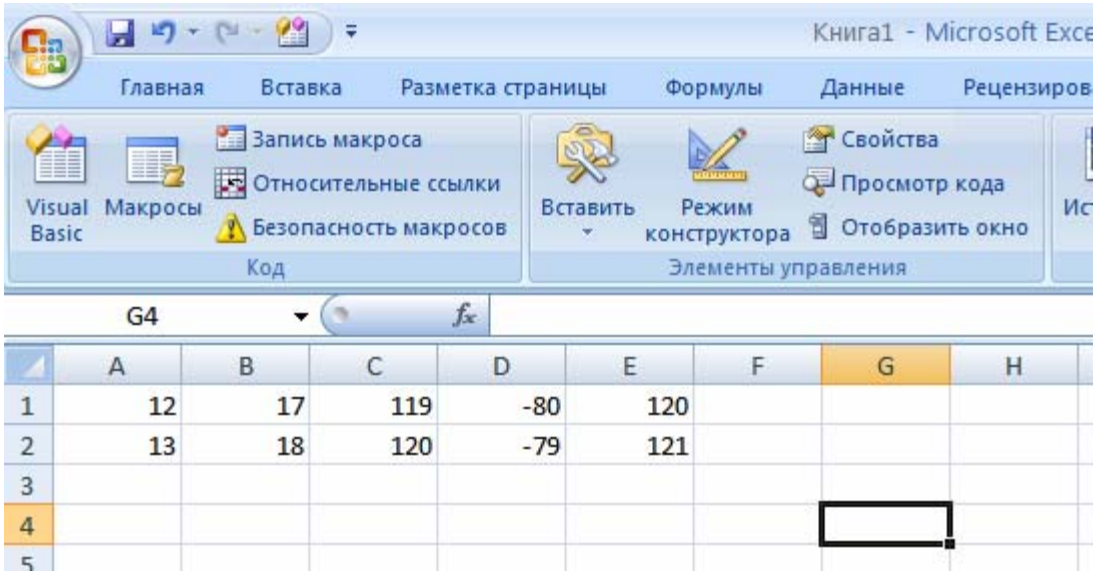
```
Sub fs()

  Dim a(5) 'Объявление одномерного массива из пяти элементов

'Считывание элементов массива из 1-ой строки листа Excel
  a(1) = Cells(1, 1)
  a(2) = Cells(1, 2)
  a(3) = Cells(1, 3)
  a(4) = Cells(1, 4)
  a(5) = Cells(1, 5)

'Преобразование элементов массива
  a(1) = a(1) + 1
  a(2) = a(2) + 1
  a(3) = a(3) + 1
  a(4) = a(4) + 1
  a(5) = a(5) + 1

'Передача элементов массива во 2-ую строку листа Excel
  Cells(2, 1) = a(1)
  Cells(2, 2) = a(2)
  Cells(2, 3) = a(3)
  Cells(2, 4) = a(4)
```

Cells(2, 5) = a(5)

End Sub

Для разъяснения структуры программы здесь написаны комментарии после символа верхнего апострофа ', которые по умолчанию автоматически закрашиваются в зеленый цвет. Символы, расположенные после верхнего апострофа ', при трансляции и исполнении кода фактически не воспринимаются, и лишь помогают пользователям разобраться в тексте программы. Вы можете по своему усмотрению ставить любые удобные для Вас комментарии, а можете и вовсе обходиться без них при составлении несложных программ.

После исполнения программы таблица Excel примет следующий вид:



Удостоверьтесь в том, что значения соседних ячеек первой и второй строки отличаются на 1.

ЗАДАНИЕ 5. Аналогично предыдущему заданию, создайте в Excel столбец из 7 произвольно выбранных чисел. В модуле VBA составьте программу, которая объявляет массив *a* размерности *Dim a(7)*, считывает его значения из только что созданной таблицы Excel, преобразовывает значение каждого элемента по формуле *a(i)=a(i)+5*, и затем преобразованный массив передает в лист Excel *вместо* исходной таблицы. Нажимайте несколько раз на клавиш F5 и удостоверьтесь в том, что каждое исполнение модуля приводит к увеличению элементов столбца на 5.

ЗАДАНИЕ 6. Решите предыдущее задание с использованием оператора условного перехода *If…Then* и безусловного перехода *GoTo*.

РЕШЕНИЕ ЗАДАНИЯ 6. В ходе решения предыдущих заданий Вы должны были заметить, что при работе с массивами в программе чрезвычайно неудобно оперировать с каждым элементом в отдельности. Вместо этого

целесообразно задать переменные индексы элемента и организовать работу по циклу. Это можно сделать следующим образом:

```
Sub fsah()
    Dim a(7)
    i = 1
metka: a(i) = Cells(i, 1)
    a(i) = a(i) + 5
    Cells(i, 1) = a(i)
    i = i + 1
    If  i <= 7  Then  GoTo  metka
End Sub
```

Нажимая несколько раз *F5*, удостоверьтесь в том, что программа работает верно, и его исполнение приводит к тому же результату, что и в предыдущем задании. При выполнении этого задания мы при помощи операторов условного перехода *If...Then* и безусловного перехода *GoTo* фактически организовали цикл. Но при этом мы пока не использовали специальных операторов цикла, которые разъяснены ниже.

ЦИКЛЫ

Необходимость организации циклов в программировании возникает часто, поэтому есть специальные операторы, которые значительно упрощают запись блоков программы, соответствующих работе по циклу. Часто используемая конструкция цикла в алгоритмических языках Basic – это сочетание *For….Next*. При этом организуется автоматическое повторение группы инструкций заданное число раз. Форма синтаксиса следующая:

```
For  i=M  To  N  Step  K
  [statements]
Next  i
```

Оператор *For….Next* повторяет выполнение группы инструкций statements, пока счетчик i изменяет свое значение от начального M до конечного N с шагом K. Нижеследующие задания необходимо выполнить с применением оператора цикла *For….Next*.

ЗАДАНИЕ 7. Выполните ЗАДАНИЕ 5 с использованием оператора цикла *For….Next.*

РЕШЕНИЕ ЗАДАНИЯ 7: Программа модуля будет выглядеть следующим образом:

```
Sub  fs()
Dim  a(7 )
```

```
  For  i = 1  To  7  Step  1
     a(i) = Cells(i, 1)
     a(i) = a(i) + 5
     Cells(i, 1) = a(i)
  Next  i
End  Sub
```

Запустите данный модуль и убедитесь в том, что результат его исполнения совпадает с результатом двух предыдущих заданий. Следует также заметить, что в данном случае в третьей строке данной программы можно было обойтись без фрагмента текста с указанием шага (*Step 1)*, так как если шаг не указан, то он по умолчанию принимается равным единице. Также можно было не указывать имя счетчика *i* в предпоследней строке программы, напечатав вместо этого просто слово *Next*.

ЗАДАНИЕ 8. Заполните шесть первых ячеек десятой строки Excel произвольными числами. В модуле VBA составьте программу, которая объявляет массив *a* размерности *Dim a(6)*, считывает его значения из только что созданной таблицы Excel, преобразовывает значение каждого элемента по формуле *a(j)=a(j)+25*, и затем преобразованный массив передает в первые шесть ячеек пятнадцатой строки таблицы Excel. При выполнении данного задания используйте оператор *For…Next*.

ЗАДАНИЕ 9. Заполните таблицу из пяти строк и семи столбцов в Excel произвольными числами. В модуле VBA составьте программу, которая объявляет массив *a* размерности *Dim a(5,7)*, считывает его значения из только что созданной таблицы Excel, преобразовывает значение каждого элемента по формуле *a(i,j)=a(i,j)+5*, и затем преобразованный массив передает вместо исходной таблицы Excel.

РЕШЕНИЕ ЗАДАНИЯ 9. Здесь уместно использовать вложенные циклы *For…Next.* Внутренний цикл перебирает элементы фиксированной строки поочередно по столбцам, а внешний цикл обеспечивает переход на следующую строку после того, как внутренний цикл закончит перебор всех элементов фиксированной строки:

```
Sub  hs()
  Dim  a(5,7 )
   For  i = 1  To  5  Step  1
     For j = 1  To  7  Step  1
      a(i,j) = Cells(i, j)
      a(i,j) = a(i,j) + 5
```

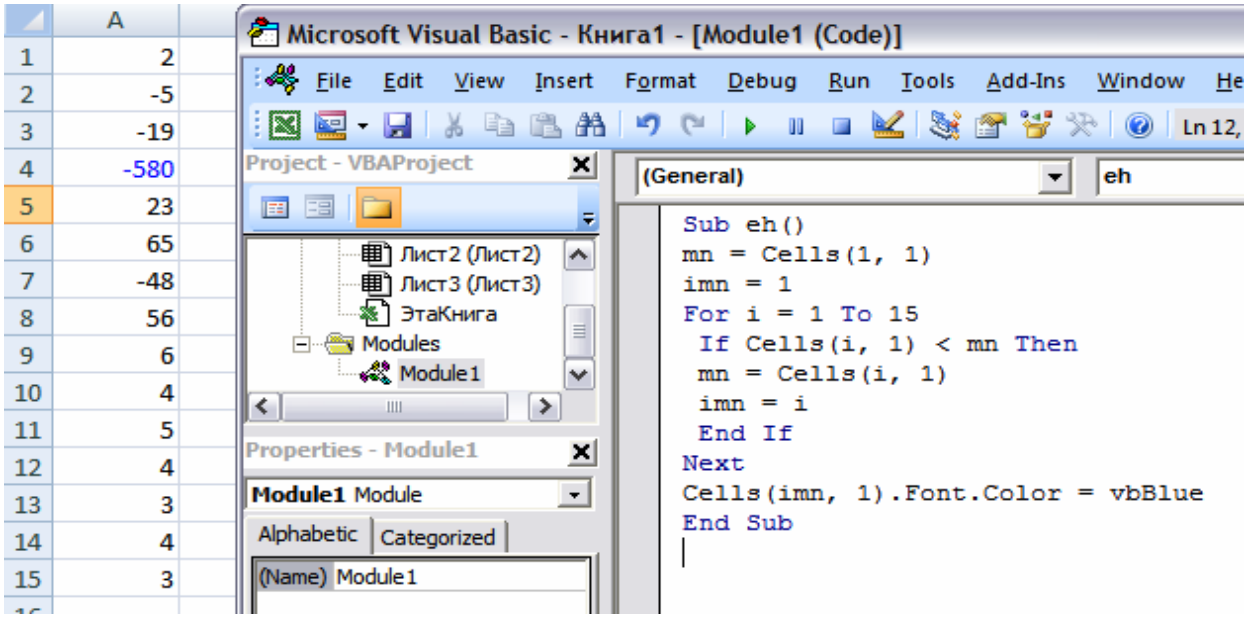
```
      Cells(i, j) = a(i,j)
    Next   j
  Next   i
End  Sub
```

Исполните данный модуль и удостоверьтесь в том, что это приводит к требуемому результату.

ЗАДАНИЕ 10. Заполните таблицу из пяти строк и шести столбцов в Excel произвольными числами. В модуле VBA составьте программу, которая считывает значения массива из только что созданной таблицы Excel, преобразовывает значение каждого элемента массива по формуле *a(i,j)=a(i,j)+5*sin(a(i,j))*, и затем преобразованный массив передает в другой диапазон ячеек листа Excel так, что бы на одном листе отображались и исходная, и преобразованная таблицы.

ЗАДАНИЕ 11. Напишите модуль на VBA для поиска наименьшего элемента столбца и выделения его синим цветом. С его использованием отыщите наименьший элемент столбца листа Excel из 15 элементов.

РЕШЕНИЕ ЗАДАНИЯ 11. В результате выполнения задания окна должны иметь следующий вид. Мы видим, что после исполнения модуля наименьший элемент автоматически окрасился в синий цвет.



ЗАДАНИЕ 12. Напишите модуль на VBA для поиска наибольшего элемента столбца и выделения его красным цветом. С его использованием отыщите наибольший элемент столбца листа Excel из 15 элементов. (Для обозначения красного цвета используйте vbRed. Если у Вас при наборе в листе Excel шрифт в ячейке A4 остается зеленым после выполнения предыдущего

задания, то перед выполнением данного задания Вы можете выделить необходимый для Вас диапазон ячеек, далее мышью выберите *Формат ячеек…* и выберите необходимый для Вас цвет. Вместо этого можно также перейти на другой лист книги Excel, который имеет свойства, принятые по умолчанию, например, на Лист2. Для этого необходимо мышью выбрать соответствующую вкладку в нижнем левом углу)

ЗАДАНИЕ 13. Заполните произвольными числами таблицу Excel, состоящую из 5 строк и 7 столбцов, при этом среди них должны быть как положительные, так и отрицательные числа. Напишите модуль на VBA, в результате исполнения которого все отрицательные элементы данной таблицы Excel окрашиваются в синий цвет, а все положительные – в красный. Исполните программу и проверьте результат.

ЗАДАНИЕ 14. Заполните шесть первых ячеек десятой строки Excel произвольными числами. В модуле VBA составьте программу, которая объявляет массив *a* размерности *Dim a(6)*, считывает его значения из только что созданной таблицы Excel, преобразовывает значение каждого элемента по формуле *a(j)=a(j)+5*, и затем преобразованный массив передает в первые шесть ячеек десятой строки таблицы Excel, окрашивая при этом все элементы в красный цвет.

VBA КАК ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ

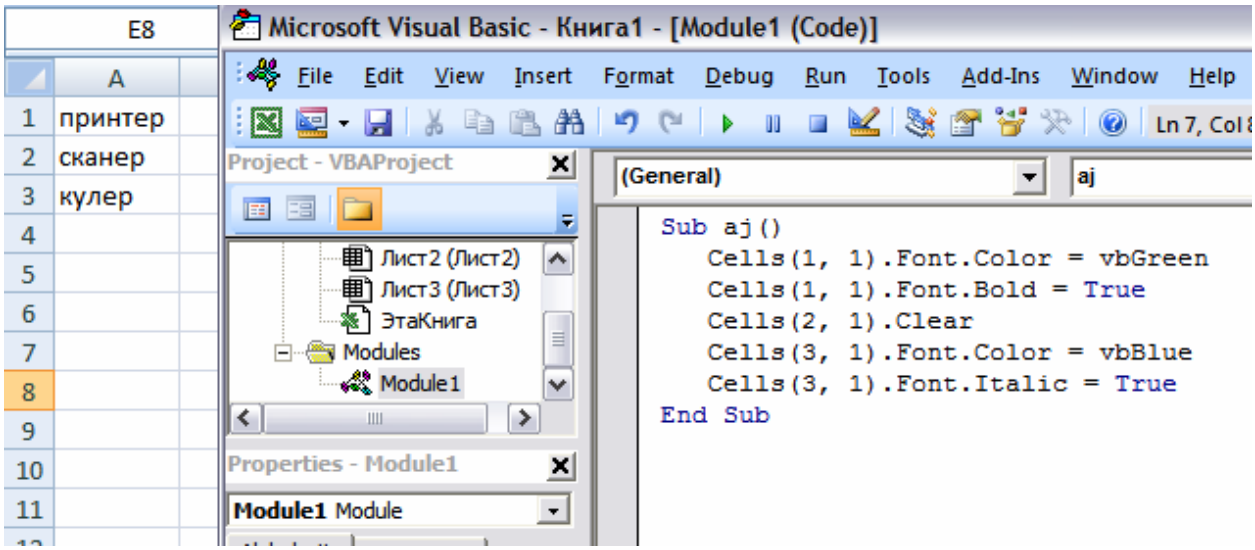
При выполнении всех заданий данного практикума мы, возможно, сами того не замечая, фактически использовали *объектно-ориентированное программирование*. Основные парадигмы объектно-ориентированного программирования – это объект, свойство, метод, событие, класс, семейство объектов. *Объект* можно трактовать как объединение данных с кодом для их переработки. Используемый нами Лист Excel является примером объекта. Ячейка листа и диапазон ячеек также являются объектами. *Семейство* представляет собой объект, содержащий в себе несколько объектов, как правило, одного типа. *Класс* обычно определяется как проект, на основе которого в дальнейшем будет создан конкретный объект. Поэтому любой объект представляет собой экземпляр класса. Сам по себе объект не представляет ценности, если не определены методы работы с ним. *Метод* представляет собой действие, выполняемое над объектом. *Событие* – это извещение, которое генерируется в результате действия пользователя или работы программы. *Свойство* – это атрибуты объекта, определяющие его характеристики. Цвет шрифта, который менялся с черного на синий или красный при выполнении модулей в предыдущих заданиях – это пример свойства *Color* объекта *Font*. Действительно, при выполнении заданий мы составляли программы, выполняющие действия над листами и ячейками Excel, представляющими собой объекты, и изменяли свойства этих объектов.

Выполните еще два задания, в ходе которых кроме свойства *Color* объекта *Font* Вам будет необходимо изменять его логические свойства, которые устанавливают, является ли шрифт полужирным или курсивом.

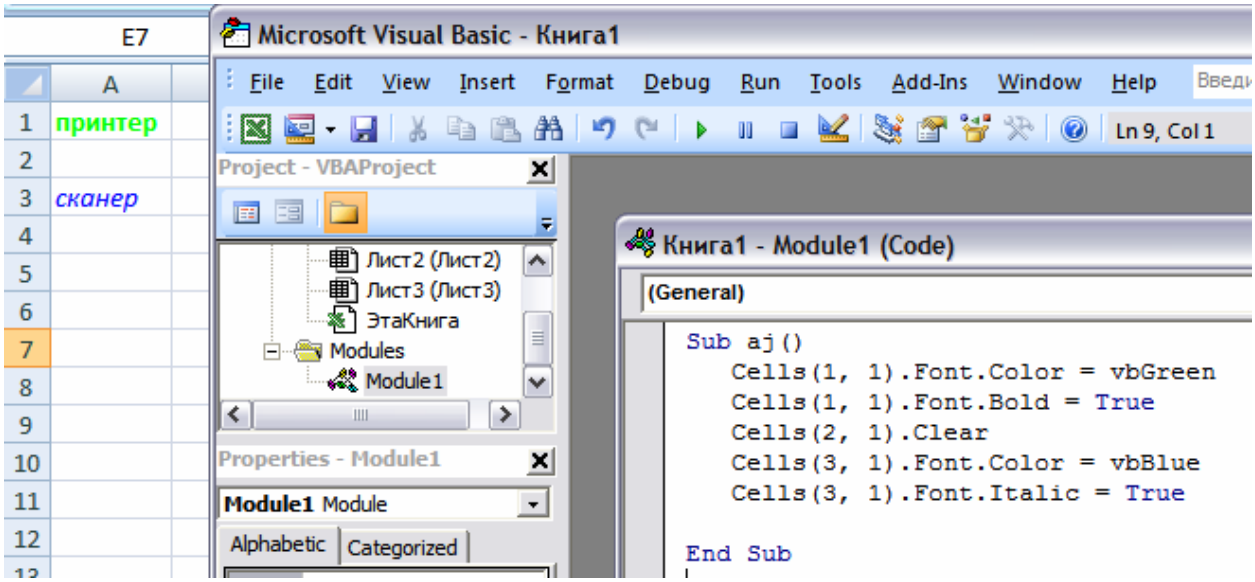
ЗАДАНИЕ 15. Заполните три первых ячейки первого столбца листа Excel произвольным текстом. Создайте и запустите модуль на VBA, в результате исполнения которого шрифт текста в первой ячейке станет зеленым полужирным, в третьей ячейке - синим курсивом, а вторая ячейка очистится.

РЕШЕНИЕ ЗАДАНИЯ 15. Если после выполнения предыдущих заданий сохранился цвет шрифта в ячейках, то проще всего перейти на другой лист книги Excel, например, на Лист 3, который активизируется при выборе мышью соответствующей вкладки, расположенной по умолчанию в нижнем левом углу.

Вид окон до исполнения программы:



После исполнения программы:



ЗАДАНИЕ 16. Заполните произвольными числами таблицу Excel, состоящую из 6 строк и 7 столбцов, при этом среди них должны быть числа как меньше 1000, так и больше 1000. Создайте модуль на VBA, в результате исполнения которого ячейки с числами меньше 1000 очистятся, а шрифт чисел в остальных ячейках станет зеленым полужирным курсивом. Исполните программу и проверьте результат.

ЗАДАНИЕ 17. Заполните произвольными числами таблицу Excel, состоящую из 15 строк и 15 столбцов. Напишите модуль на VBA, в результате исполнения которого происходит транспонирование этой квадратной матрицы (то есть замена элементов по принципу переворота относительно главной диагонали). Элементы транспонированной матрицы, находящиеся выше главной диагонали, в таблице Excel должны окраситься в синий цвет. Элементы транспонированной матрицы, находящиеся ниже главной диагонали, в таблице Excel должны окраситься в красный цвет. Элементы главной диагонали должны иметь прежний цвет, который они имели до транспонирования. Исполните программу и проверьте результат.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1.Гарнаев А.Ю. Excel, VBA, Internet в экономике и финансах. – Спб.:БХВ-Петербург, 2002, - 816 с.
2.Информатика. Базовый курс. Под. ред. С.В.Симоновича: СПб, «Питер», 2002, 640с.

Рустам Гумерович Тахавутдинов

Основы алгоритмизации и программирования. Методические указания к лабораторным работам, расчетному заданию и самостоятельной работе студентов
(Кафедра Информатики и информационных управляющих систем КГЭУ)

Изд. лиц. № 03480 от 8.12.00 КГЭУ 2008 г.
Подписано к печати Формат 60 х 84/16
Гарнитура “Times” Вид печати РОМ Бумага “Business”
Физ. печ. л. Усл. печ. л. Уч-изд. л.
Тираж 200 Заказ

Издательский отдел КГЭУ
420066, Казань, Красносельская, 51

Типография КГЭУ
420066, Казань, Красносельская, 51