

В данном пособии представлены материалы к семинарским занятиям по курсу «Языки управления приложениями». Задачи решаются на языке VBA встроенном в пакет Microsoft Office.

## **Семинар 1 Изучение базового синтаксиса и основных конструкций языка VBA. Изучение свойств простейших объектов приложения Microsoft Excel. Изменение формата ячеек с помощью макросов.**

Материалы семинара:  
основные конструкции языка VBA

### **For ... Next**

```
For counter = start To stop Step step  
операторы  
Next counter
```

где *counter* - счетчик цикла, *start* - начальное значение счетчика, *stop* - конечное значение счетчика, *step* - шаг приращения счетчика (указывать необязательно, по умолчанию = 1)

**Exit For** - преждевременное прерывание цикла

### **If ... Then ... Else**

```
If condition Then  
block1  
Else  
block2  
End If
```

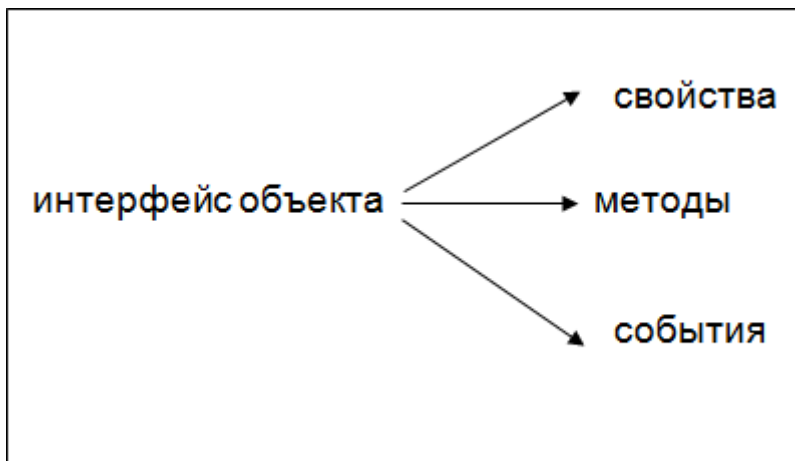
Где *condition* - условное выражение, если *condition* истина, то выполняется *block1*, если *condition* ложь, то выполняется *block2*

**Else** и второй блок операторов необязательны:

```
If condition Then  
block1  
End If
```

возможна проверка нескольких условий:

```
If condition1 Then  
block1  
Elseif condition2 Then  
block2  
Elseif condition3 Then  
block3  
...  
Else  
block4  
End If
```



**Свойства** - способ хранения объектами своих данных  
(excel: value - число, FormatNumber - способ отображения)

Свойства могут быть:

- доступны для чтения
- доступны для чтения и записи
- ссылаться на другие объекты - иерархия

**Методы** - действия над объектом  
(Document - объект, PrintOut - метод)

**События** - реакция объекта на внешние события. Например: нажатие клавиши, щелчок мыши, открытие файла, изменение данных  
(Form - объект, Open - событие, можно написать функцию, которая обеспечивает вывод формы всегда в центр экрана независимо от разрешения монитора)

**Некоторые свойства объекта ячейка:**

**Cells(i, j).Value** - значение ячейки

**Cells(i, j).Font** - шрифт текста ячейки

**Cells(i, j).Font.ColorIndex** - цвет текста ячейки

**Cells(i, j).Interior.ColorIndex** - цвет фона ячейки

Свойство ColorIndex может принимать значения от 1 до 56, если у ячейки отсутствует фон, то значение ColorIndex равно -4142.

### Задание семинара:

написать макрос, который в столбец А заносит числа от 1 до 56, цвет фона в этих ячейках должен совпадать с записанным числом. В соседнем столбце В требуется изменить заливку фона ячейки цветом, совпадающим с числом, записанным в соседней ячейке А (см. Рисунок 1).

#### Пример кода:

```
Sub ttt()  
For i = 1 To 56  
    Cells(i, 1).Value = i  
    Cells(i, 2).Interior.ColorIndex = i  
    Cells(i, 1).Font.ColorIndex = i  
Next i  
End Sub
```

	A	B	C
1	1	Black	
2			
3	3	Red	
4	4	Green	
5	5	Blue	
6	6	Yellow	
7	7	Magenta	
8	8	Cyan	
9	9	Brown	
10	10	Dark Green	
11	11	Dark Blue	
12	12	Olive	
13	13	Purple	
14	14	Teal	
15	15	Grey	
16	16	Dark Grey	
17	17	Light Purple	
18	18	Dark Red	
19	19	Light Yellow	
20	20	Light Cyan	
21	21	Dark Purple	
22	22	Light Red	
23	23	Dark Blue	
24	24	Light Purple	
25	25	Dark Blue	

Рисунок 1

### Задание для самостоятельной работы:

Модифицировать написанный цикл таким образом, чтобы пара столбцов цифра-цвет располагалась в прямоугольнике высотой 10 и шириной 12 (см. Рисунок 2).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1	Black	11	Dark Blue	21	Dark Red	31	Dark Green	41	Light Blue	51	Dark Green	
2			12	Olive	22	Light Red	32	Blue	42	Cyan	52	Dark Olive	
3	3	Red	13	Purple	23	Blue	33	Cyan	43	Light Green	53	Brown	
4	4	Green	14	Teal	24	Light Purple	34	Light Cyan	44	Yellow	54	Dark Purple	
5	5	Blue	15	Grey	25	Dark Blue	35	Light Green	45	Orange	55	Dark Blue	
6	6	Yellow	16	Grey	26	Magenta	36	Light Yellow	46	Orange	56	Dark Grey	
7	7	Magenta	17	Light Purple	27	Yellow	37	Light Blue	47	Dark Purple			
8	8	Cyan	18	Dark Red	28	Cyan	38	Light Red	48	Grey			
9	9	Brown	19	Light Yellow	29	Dark Purple	39	Light Purple	49	Dark Blue			
10	10	Dark Green	20	Light Cyan	30	Dark Red	40	Light Orange	50	Dark Green			
11													
12													

Рисунок 2

## Семинар 2. Диалоги, создания форм. Использование элементов управления, их возможности и особенности. Обработка событий.

Материалы семинара:

Зачастую при написании программы, в нашем случае макроса, требуется взаимодействие с пользователем. Задав вопрос и получив на него ответ, мы делаем выбор дальнейшего развития событий. Для осуществления диалога с пользователем будем пользоваться таким объектом как *диалог*. Диалог это некоторое окно, которое может появляться, исчезать, реагировать на некоторые события и содержать в себе некоторую информацию и элементы, с помощью которых может быть осуществлен диалог с пользователем.

Рассмотрим процесс создания формы.

Откроем окно редактора Visual Basic. В левой части экрана мы видим список объектов, из которых на данный момент состоит наш проект (см Рисунок 3).

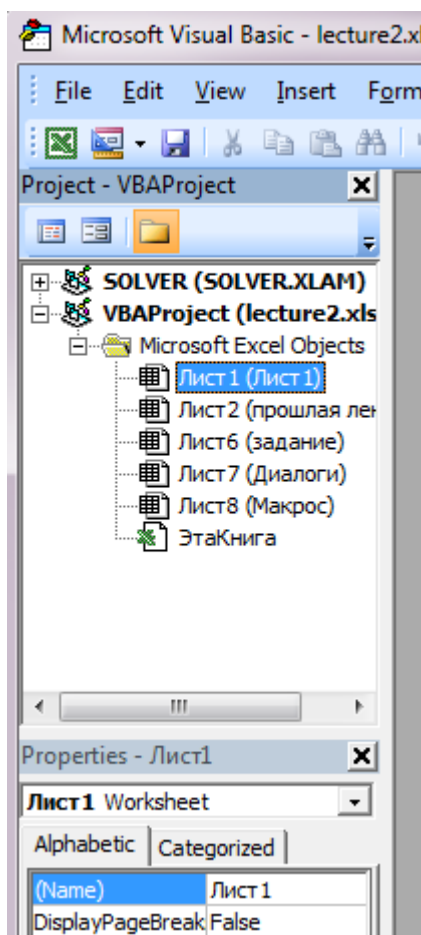


Рисунок 3

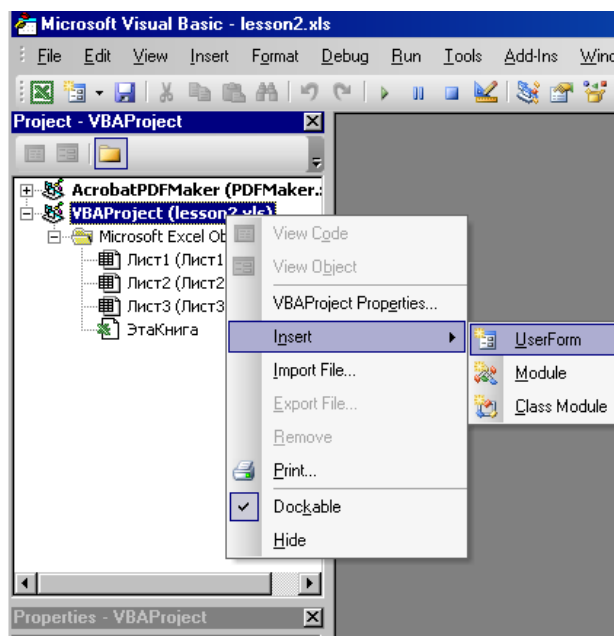


Рисунок 4

Кликнув правой кнопкой мыши на экране, мы получаем возможность добавить к нашему проекту новый объект. В нашем случае выбираем UserForm (см. Рисунок 4). Модуль добавляется к проекту автоматически. Автоматически создаются все заголовки и создается элемент типа Form. В нем уже реализованы многие функции, такие как отрисовка, изменение размеров, закрытие и т.д. Есть возможность добавления на форму элементов управления. Перечень этих элементов мы видим в отдельном окне выбора Toolbox (см. Рисунок 5).

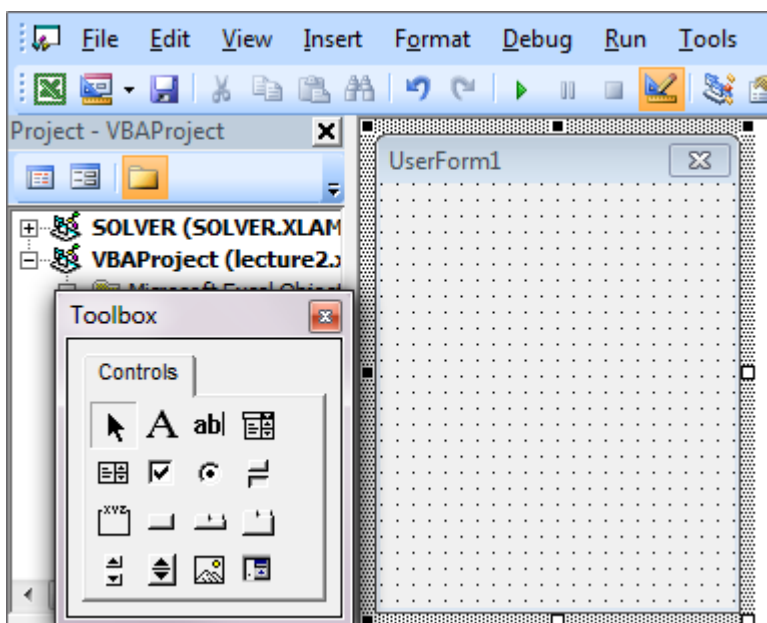


Рисунок 5

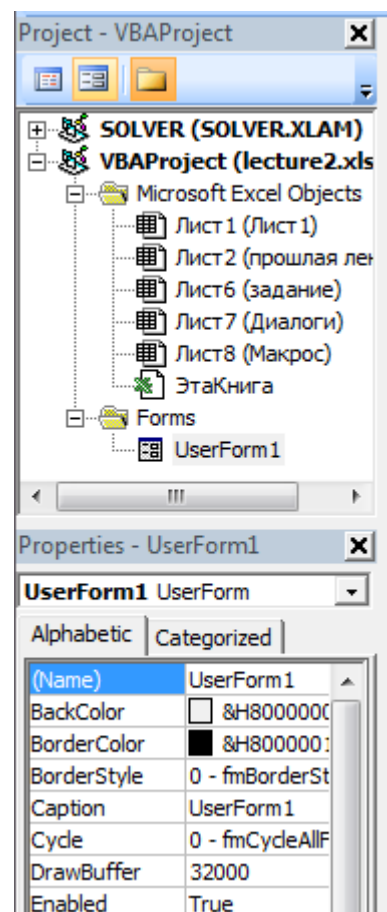


Рисунок 6

Слева, в разделе Properties, мы можем изменять атрибуты объекта (см. Рисунок 6).

Например, для изменения заголовка формы на task1 мы должны в разделе Appearance в строке Caption изменить текст на наше новое название task1.

В разделе Position можно изменить размер окна нашего диалога с пользователем (Left, Top, Height, Width, StartUpPosition – Center Owner, Center Screen, Windows Default). Раздел Position находится ниже, поэтому нам придется воспользоваться скроллингом, расположенным справа от разделов.


Рассмотрим назначение и принцип работы некоторых элементов управления.

**A** - **Label** – надпись.


Используется, когда на экран надо вывести заголовок или поясняющий текст. Для изменения появляющейся на экране надписи надо в Properties в разделе Appearance в строке Caption сделать соответствующие изменения. Пользователь не может редактировать эту надпись в запущенном окне приложения.

**abl** - **Text Box** – поле с текстом.


Этот элемент управления предназначен для показа текста пользователю. Пользователь может менять значение этого поля во время выполнения программы. Этим соответственно можно пользоваться для организации интерфейса с пользователем, например, запроса данных. Изменить текст этого окна можно в разделе Appearance в строке Value. Там же можно посмотреть данные, которые ввел пользователь в процессе выполнения программы.

 - **Check Box** – переключатель типа ДА/НЕТ.

Этот элемент управления позволяет пользователю сделать выбор типа ДА/НЕТ. Если в поле Check Box стоит галочка, то значение свойства Value станет true, в противном случае оно будет false. Можно установить значение по умолчанию, для этого в разделе Appearance в строке Value ставим значение true или false в зависимости от того, что нам нужно показать пользователю. Там же в процессе выполнения программы можно посмотреть, изменил ли пользователь положение переключателя или нет. Справа от переключателя имеется надпись, которая позволяет сделать пояснение для пользователя. Изменить этот текст можно в Properties в разделе Appearance в строке Caption.

 - **Radio Button** – переключатель по вариантам.

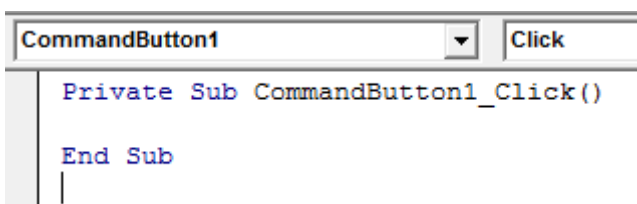
Работа этого элемента управления очень похожа на работу Check Box. Отличие заключается в том, что выбор делается между несколькими вариантами. Из этого следует, что элементов управления типа Radio Button на форме должно быть несколько. Как только пользователь выбирает один из вариантов, выделение с другого варианта снимается автоматически. В свойство Caption заносится пояснение пользователю, а в свойстве Value значение поля по умолчанию – true или false в зависимости от того, что мы хотим показать пользователю в начале работы приложения. В этом же поле в ходе выполнения программы можно посмотреть, сделал ли пользователь выбор этого варианта или нет.

 - **Frame** -- группировка.

Служит для логического объединения различных элементов управления. Пользователь видит в окне тоненькую рамочку. Такое объединение полезно, когда нужно объединить переключатели Radio Button. У этого элемента управления есть заголовок, который по умолчанию показывается в верхней левой части рамки. Значение заголовка можно изменить в разделе Appearance в строке Caption.

 - **CommandButton** - кнопка

Кнопка – без этого элемента управления не обходится практически ни одно приложение. Самые известные примеры кнопок – OK и Exit. После размещения этого элемента управления на нашей форме мы можем поменять надпись, которая будет показываться пользователю на кнопке. Для этого надо в Properties в разделе Appearance в строке Caption сделать соответствующие изменения. По умолчанию кнопки получают имя CommandButton1, CommandButton2 и т.д. На событие Click (нажатие на кнопку) можно написать соответствующий заданию обработчик событий. Например, нажатию кнопки Close можно сопоставить обработчик End - что соответствует окончанию работы макроса. Для этого надо дважды кликнуть на нашу кнопку. После этого откроется окно для редактирования кода программы:



```
CommandButton1 Click
Private Sub CommandButton1_Click()
End Sub
```

Курсор при этом находится внутри процедуры `Private Sub CommandButton1_Click()`, которая будет запускаться каждый раз при нажатии на нашу кнопку. Наша кнопка Exit призвана останавливать работу макроса, следовательно вписываем строку End и возвращаемся к конструктору форм. Аналогично можно написать обработчик на клик на любом элементе управления. Далее мы будем подробно разбирать, какие события существуют у элементов управления и как писать обработчики к ним. Пока вернемся к разбору видов элементов управления.



- ScrollBar – горизонтальная и вертикальная прокрутка

Эти элементы управления позволяют реализовать режим прокрутки на нашей форме. Из свойств отдельно отметим Value – позиция ползунка при запуске программы; Max и Min – максимальное и минимальное значение, которое может принимать элемент прокрутки. LargeChange и SmallChange – изменение позиции ползунка при нажатии на стрелку прокрутки и на полосу прокрутки соответственно. Позже мы подробнее поговорим о работе с этими элементами управления.



- Image -картинка

Элемент управления, который используют для отображения картинки. Легко менять его месторасположение программным путем, т.е. легко двигать картинку в процессе выполнения программы. Сама картинка вставляется из файла, который указывается в свойстве Picture.

Надо заметить, что практически у каждого элемента управления, рассмотренного нами, есть такие свойства как BorderStyle – позволяющий выделять элементы управления рамками; Font - позволяющий изменять стиль надписей элемента; BackColor– позволяющий выбирать цвет фона; Visible – позволяющий скрывать и вновь показывать пользователю элемент управления и многие другие.

### **Активная ячейка и ее свойства:**

**ActiveCell** - текущая активная ячейка

**ActiveCell.Value** - значение текущей активной ячейки

**ActiveCell.Interior.ColorIndex** - цвет фона активной ячейки

**ActiveCell.Next.Activate** - сделать активной следующую ячейку

**ActiveCell.Row** - номер строки активной ячейки

**ActiveCell.Column** - номер столбца активной ячейки

**Cells(1,1).Activate** - сделать ячейку (1,1) активной

### **Последовательность случайных чисел их использование в программе.**

**Randomize n**, где n любое числовое выражение - инициализировать последовательность псевдослучайных чисел [0 , 1). Если вызывать функцию без параметра, то последовательность будет инициализироваться с параметром, зависящим от системного времени компьютера. Пример вызова с использованием секунд текущего времени компьютера в качестве параметра вызова.

#### **Randomize Second(Now)**

Для получения случайного числа нужно использовать функцию Rnd без параметров. Каждый новый вызов будет выдавать следующее число из последовательности. Например:

**Col = Int((Rnd \* 56) + 1)**, при каждом вызове этой строки переменной Col будет присваиваться случайное целое число в диапазоне [1,56].

## Задание семинара:

Написать макрос, который выводит форму с тремя кнопками (см. Рисунок 7).

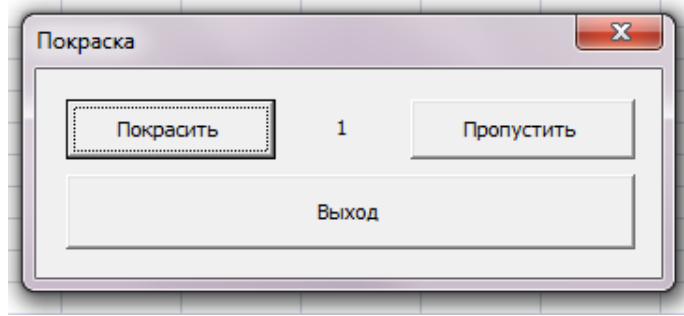


Рисунок 7

Кнопка Покрасить – красит активную ячейку в синий цвет и переходит к следующей ячейке.

Кнопка Пропустить – переходит к следующей ячейке, ничего не делая с текущей

Кнопка Выход – заканчивает работу макроса.

### Пример кода:

```
Код макроса:  
Sub test1()  
UserForm1.Show  
End Sub
```

```
Код Формы:  
Private Sub CommandButton1_Click()  
ActiveCell.Interior.ColorIndex = 11  
ActiveCell.Next.Activate  
End Sub
```

```
Private Sub CommandButton2_Click()  
ActiveCell.Next.Activate  
End Sub
```

```
Private Sub CommandButton3_Click()  
End  
End Sub
```

## Задание для самостоятельной работы:

Изменить макрос, написанный в первом задании следующим образом: ячейки требуется красить не в синий цвет, а в случайный, причем в ячейку требуется вписывать номер этого цвета. Предлагаемый цвет для покраски показывается на форме в элементе Label. При каждом нажатии на кнопку Покрасить или Пропустить случайный цвет меняется (см Рисунок 8).

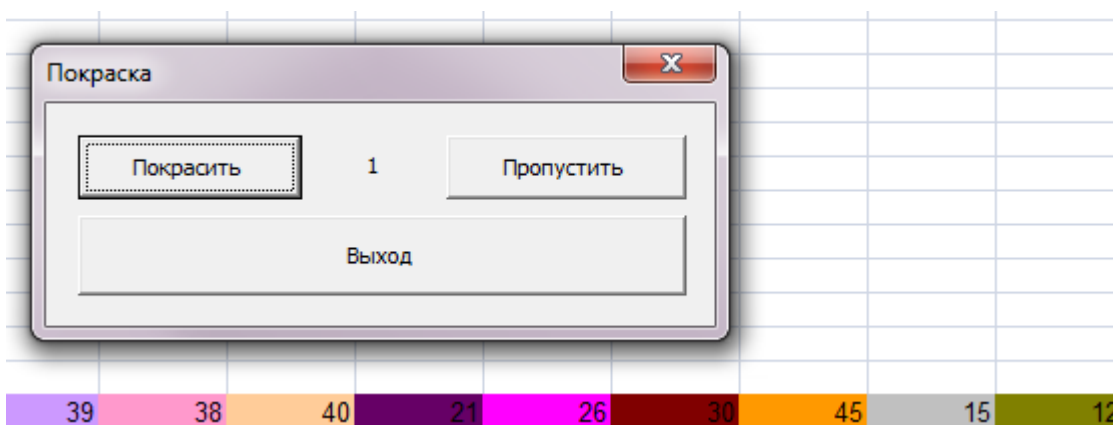


Рисунок 8



## Семинар 3. Collection, Range, Selection. Построение гистограмм.

### Collection

Объект Collection позволяет динамически группировать семейство объектов, идентифицируя их по индексу. Имеет одно свойство Count и три метода add, item, remove.

#### Add item, key, before, after

**Item** – обязательный параметр – добавляемый объект

**Key** – необязательный параметр – может использоваться вместо индекса для поиска

**Before** – необязательный параметр – указывает элемент, перед которым надо вставить добавляемый объект

**After** – необязательный параметр – указывает элемент, после которого надо вставить добавляемый объект

**Remove index** – удаляет объект из коллекции. Если удаляли первый элемент, то второй становится первым, третий – вторым и т.д., а количество элементов уменьшается на единицу

**Item(index)** – возвращает член коллекции по числовому индексу или по key.

Для последних двух методов item может представлять собой как число, так и текст. Если item является числом, то удаляется элемент, находящийся на позиции равной index. При этом значение index должно быть от 1 до числа, равного количеству элементов в коллекции на момент удаления элемента. Если index не является числом, то он воспринимается как key и удаляется элемент, названный этим ключом.

Объявление новой коллекции:

```
Dim mycoll As Collection  
Set my coll = New Collection
```

### Range

Range – диапазон – является одним из основных объектов VBA. Объект Selection может возникнуть двумя способами: результат работы метода Select или при вызове свойства selection. Тип получаемого объекта зависит от типа выделенного объекта. Чаще всего объект Selection принадлежит к классу Range, и при работе с ним можно использовать свойства и методы объекта Range. Объект Range возвращается либо как элемент семейства Range или Cells, либо свойствами Range, Cells, Offset, либо методами ActiveCell, Intersect и Union.

Приведем несколько примеров.

```
Cells(1,2)=1 'в ячейку A2 записали число 1
```

```
Range("A2").Interior.ColorIndex=3 'всем ячейкам диапазона сделать 3 фон, диапазон состоит из одной ячейки  
A2
```

```
Range("B2:D5").Select 'выделение диапазона ячеек, ограниченного прямоугольником от ячейки A2 до ячейки  
D5
```

```
Selection.Cells(2,2).Value=2 'в выделенном диапазоне в ячейку с адресом 2,2 записать число 2. Позиция 2,2  
отсчитывается от верхнего левого края выделенного диапазона. Если был выделен диапазон  
B2:D5, то цифра 2 будет записана в ячейку C3.
```

```
s=Selection.Address 'в строку s будет записана строка "$B$2:$D$5"
```

```
x=Selection.Count 'в переменную x будет записано количество ячеек в выделенном диапазоне, в данном  
случае их 12
```

```
y = Selection.Column 'столбец, с которого начинается выделенный диапазон, в нашем случае 2
```

```
Z = Selection.Row ' строка, с которой начинается выделенный диапазон, в нашем случае 2
```

`q = Selection.Cells(3).Column` 'в выделенном диапазоне отсчитать 3ю ячейку. Счет ведется слева направо и сверху вниз. В переменную `q` записать номер столбца, в котором находится найденная ячейка. В нашем случае это ячейка D2. А номер столбца равен 4

`w = Selection.Rows.Count` 'в переменную `w` записать количество строк в выделенном диапазоне, в нашем случае их 4

`r = Selection.Columns.Count` ' в переменную `r` записать количество столбцов в выделенном диапазоне, в нашем случае их 3

`u = Selection.Cells(2).Value` 'в переменную `u` поместить значение второй ячейки выделенного диапазона, в нашем случае значение ячейки C2

Циклы по всем элемента диапазона

### **For Each ... Next**

***For Each** item in collection*

*операторы*

***Next** item*

Для каждого элемента коллекции выполнить операторы цикла. Цикл выполнится столько раз, сколько элементов было в коллекции в момент начала работы цикла.

Например, требуется очистить коллекцию `mycoll`, т.е. удалить все ее элементы

**For Each** c in mycoll

mycoll.Remove 1

**Next**

При удалении все элементы коллекции сдвигаются в ее начало. В нашем цикле мы удаляем первый элемент коллекции столько раз, сколько их было в начале работы цикла. Т.е. удалятся все элементы. Коллекция становится пустой.

### **Построение гистограмм**

Стандартное определение:

Dim ch As Excel.ChartObject

Set ch = ActiveSheet.ChartObjects.Add(Left, Top, Width, Height)

При этом `Left`, `Top`, `Width`, `Height` задаются в пикселях, `Left`, `Top` отсчитывается от верхнего левого края листа.

Добавление категории:

ch.Chart.SeriesCollection.NewSeries

ch.Chart.SeriesCollection(i).Name = i

ch.Chart.SeriesCollection(i).Values = mycoll.Count

Легенда внизу, нет подписей по оси x:

ch.Chart.SetElement (msoElementLegendBottom)

ch.Chart.SetElement (msoElementPrimaryCategoryAxisWithoutLabels)

### **Массивы**

Определение массива:

Dim cc(8) As Integer 'массив из 9 целых чисел. Первый элемент массива `c(0)`

Dim c(1 To 9) As Integer 'массив из 9 целых чисел. Первый элемент массива `c(1)`

Dim r() As Single 'определение динамического массива

ReDim r(1 To 9) 'определение размерности динамического массива

## Задание семинара:

Пользователь заполняет ячейки цифрами от 1 до 9. Потом выделяет произвольный фрагмент получившейся таблицы и запускает макрос. Макрос должен в выделенной области пройти по всем ячейкам, составить коллекцию ячеек, значение которых равно 1 и покрасить их в желтый цвет (см. Рисунок 9).

### Пример кода:

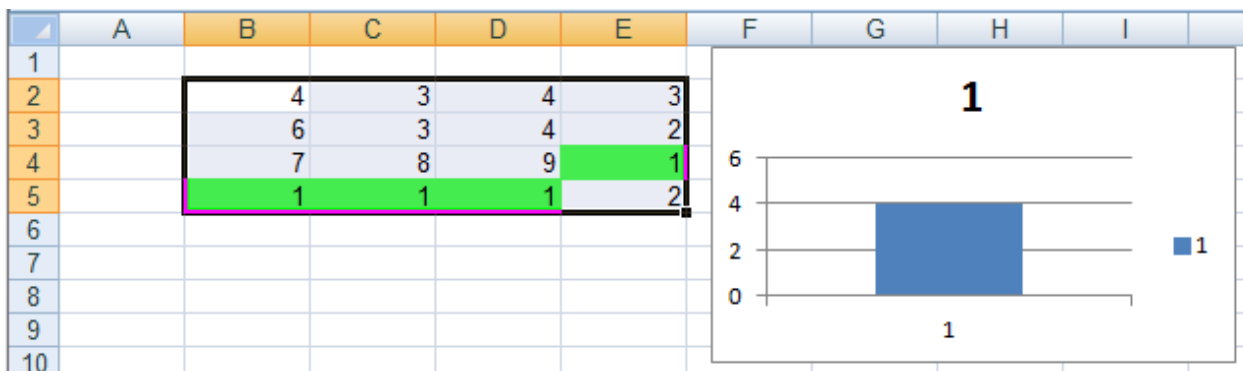
```
Sub col1()  
Dim mycoll As Collection  
Set mycoll = New Collection  
For Each c In Selection.Cells  
    If c.Value = 1 Then  
        mycoll.Add c  
        c.Interior.ColorIndex = 6  
    End If  
Next c  
End Sub
```

	A	B	C	D	E
1					
2		4	3	4	3
3		6	3	4	2
4		7	8	9	1
5		1	1	1	2
6					

Рисунок 9

Модифицируем алгоритм следующим образом: ячейки будем красить не в желтый цвет, а в тот цвет, ColorIndex которого равен количеству ячеек с единицей. Рядом построим гистограмму, отражающую

количество



встретившихся единиц (см. Рисунок 10).

Рисунок 10

### Пример кода:

```
Sub col11()  
Dim mycoll As Collection  
Set mycoll = New Collection  
For Each c In Selection.Cells  
    If c.Value = 1 Then  
        mycoll.Add c  
    End If  
Next c  
For Each c In mycoll  
    c.Interior.ColorIndex = mycoll.Count  
Next c
```

```

Dim ch As Excel.ChartObject
Set ch = ActiveSheet.ChartObjects.Add(250, 1, 250, 150)
ch.Chart.SeriesCollection.NewSeries
ch.Chart.SeriesCollection(1).Name = 1
ch.Chart.SeriesCollection(1).Values = mycoll.Count
End Sub

```

### Задание для самостоятельной работы:

Пользователь заполняет ячейки цифрами от 1 до 9. Потом выделяет произвольный фрагмент таблицы и запускает макрос. Макрос должен в выделенной области Selection пройтись по всем ячейкам и выбрать те, которые содержат цифру 1 и составить коллекцию этих ячеек. Методом count понять, сколько было всего единиц, и покрасить все эти ячейки в ColorIndex, совпадающий с количеством единиц. Аналогично для остальных цифр от 2 до 9. Соответственно, если каких-то цифр в выделенном фрагменте было одинаковое количество, то ячейки с этими цифрами в итоге будут одинакового цвета.

На активном листе построить две гистограммы, показывающие распределение цифр в выделенном диапазоне. В первой гистограмме должно быть 9 категорий, в каждой по одному числу. Во второй гистограмме должна быть одна категория, состоящая из 9 чисел (см. Рисунок 11).

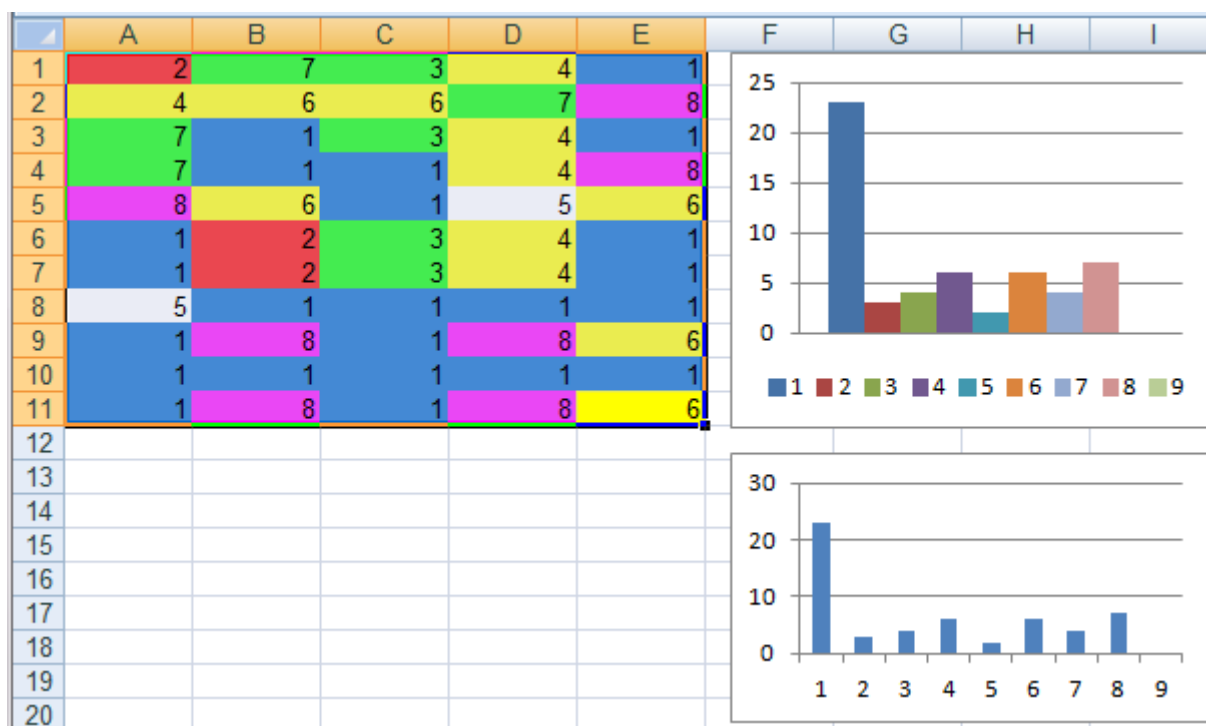


Рисунок 11

## Семинар 4. Реакция на события.

Не обязательно вручную запускать какой-то макрос, чтобы осуществить обработку данных. Можно перехватывать стандартные функции по обработке событий и дополнять обработчики своими действиями. Опишем некоторые события, которые существуют у книги (Workbook):

- Workbook\_Open() – активируется сразу после открытия книги
- Workbook\_BeforeClose(Cancel As Boolean) - активируется при закрытии книги
- Workbook\_BeforePrint(Cancel As Boolean) – активируется при передач и документа на печать
- Workbook\_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean) – активируется при попытке сохранить книгу
- Workbook\_Deactivate()
- Workbook\_Activate()
- Workbook\_SheetBeforeDoubleClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)
- Workbook\_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)
- Workbook\_SheetSelectionChange(ByVal Sh As Object, ByVal Target As Range)

Опишем события, которые существуют у листа(Worksheet):

- Worksheet\_Activate()
- Worksheet\_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
- Worksheet\_Calculate()
- Worksheet\_Change(ByVal Target As Range)
- Worksheet\_Deactivate()
- Worksheet\_FollowHyperlink(ByVal Target As Hyperlink)
- Worksheet\_PivotTableUpdate(ByVal Target As PivotTable)
- Worksheet\_SelectionChange(ByVal Target As Range)

### Задание семинара:

На первом листе книги при клике на ячейку она меняет цвет: из белой в черную, из черной в белую.

#### Пример кода:

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
If ActiveCell.Interior.ColorIndex = -4142 Then
    'отдельно рассматриваем случай, когда ячейку еще ни разу не красили
    ActiveCell.Interior.ColorIndex = 1
Else
    ActiveCell.Interior.ColorIndex = (ActiveCell.Interior.ColorIndex + 1) Mod 2
End If
End Sub
```

Немного изменим задание: пусть на втором листе книги ячейки меняют цвета не только на черный и белый, а добавим туда еще два цвета. Для оптимизации кода создадим процедуру, которую поместим в код объекта ЭтаКнига:

```
Sub ColCh(Num As Integer)
If ActiveCell.Interior.ColorIndex = -4142 Then
    ActiveCell.Interior.ColorIndex = 1
Else
    ActiveCell.Interior.ColorIndex = (ActiveCell.Interior.ColorIndex + 1) Mod Num
End If
End Sub
```

Теперь из первого листа обратимся к функции:

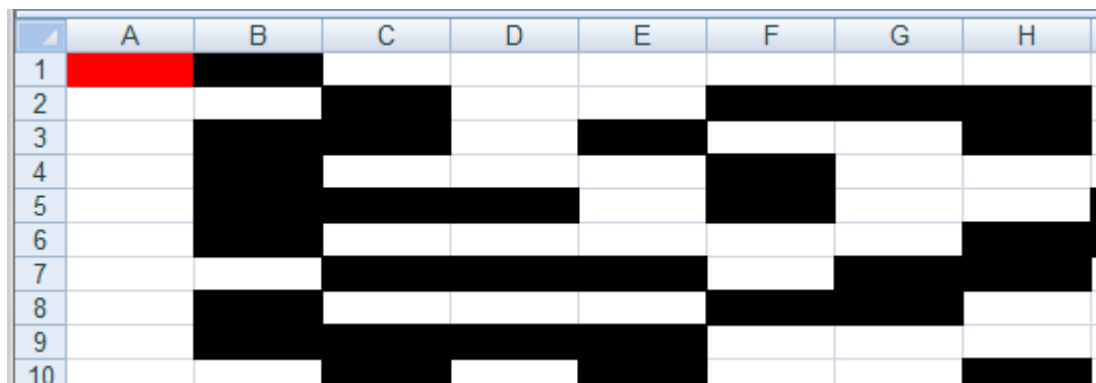
```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
ЭтаКнига.ColCh 2
End Sub
```

А из второго:

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
ЭтаКнига.ColCh 4
End Sub
```

### Задание для самостоятельной работы:

В объекте ЭтаКнига написать обработчик событий Workbook\_SheetBeforeDoubleClick, который будет отменять стандартный обработчик события двойного клика на ячейке, а вместо этого в текущем активном листе менять цвет ячейки (1,1) циклично от 1 до 56 (см. Рисунок 12).



	A	B	C	D	E	F	G	H
1	Red	Black	White	White	White	White	White	White
2	White	Black	Black	White	White	Black	Black	Black
3	White	Black	White	White	Black	White	White	Black
4	White	Black	White	White	Black	Black	White	White
5	White	Black	Black	Black	White	Black	White	Black
6	White	Black	White	White	White	White	Black	Black
7	White	Black	Black	Black	Black	White	Black	White
8	White	Black	White	White	Black	Black	White	White
9	White	Black	Black	Black	White	Black	White	Black
10	White	Black	White	White	Black	Black	White	Black

Рисунок 12

## Семинар 5. WORD. Объектная модель. Изменения свойств объектов из макросов.

### Выделение текста целиком

```
Selection.Start = ActiveDocument.Range.Start
```

```
Selection.End = ActiveDocument.Range.End
```

### Выделение с 1 по 3 абзац

```
Selection.Start = ActiveDocument.Paragraphs(1).Range.Start
```

```
Selection.End = ActiveDocument.Paragraphs(3).Range.End
```

### Объявление коллекции

```
Dim coll As Collection
```

```
Set coll = New Collection
```

### Работа с формой

```
Form.Show
```

```
Form.Hide
```

### Задание семинара:

Во всем тексте найти слова, начинающиеся на букву "а". Перед текстом вставить абзац, состоящий из найденных слов. Слова пронумеровать по шаблону: 1:ага 2:аба... .

#### 1 шаг. Собираем коллекцию слов, начинающихся с буквы "а".

```
Selection.Start = ActiveDocument.Range.Start
```

```
Selection.End = ActiveDocument.Range.End
```

```
Dim coll As Collection
```

```
Set coll = New Collection
```

```
For Each w In Selection.Words
```

```
If w.Characters(1) = "a" Then
```

```
coll.Add w
```

```
w.Font.Bold = True
```

```
End If
```

```
Next
```

#### 2 шаг. Перед текстом вставляем абзац, в который по заданному шаблону записываем все слова из коллекции.

```
Selection.Move wdParagraph, -1 'перемещаем курсор на 1 параграф назад
```

```
Dim i As Integer
```

```
i = 1
```

```
For Each w In coll
```

```
Selection.TypeText i & ":" & w & " " 'печатаем слова по заданному шаблону
```

```
i = i + 1
```

```
Next w
```

```
Selection.InsertParagraph 'добавляем символ конца абзаца
```

```
'Selection.InsertBreak wdPageBreak так можно добавить разрыв страницы
```

### Пример кода:

```
Sub Searchmini()  
Dim coll As Collection  
Set coll = New Collection  
Selection.Start = ActiveDocument.Range.Start  
Selection.End = ActiveDocument.Range.End  
For Each w In Selection.Words  
    If w.Characters(1) = "a" Then  
        coll.Add w  
        w.Font.Bold = True  
    End If  
Next  
Selection.Move wdParagraph, -1  
Dim i As Integer  
i = 1  
For Each w In coll  
    Selection.TypeText i & ":" & w & " "  
    i = i + 1  
Next  
Selection.InsertParagraph  
End Sub
```

### Задание для самостоятельной работы:

Во всем тексте найти слова, начинающиеся на буквы, заданные пользователем в диалоге (см. Рисунок 13). Перед текстом вставить абзацы, состоящие из найденных слов. Слова пронумеровать по шаблону: 1:ага 2:аба... .

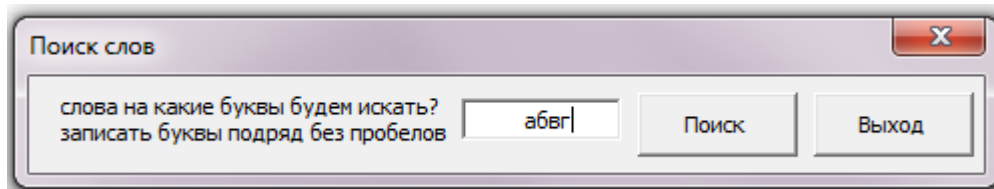


Рисунок 13



## Семинар 6. Поиск с помощью подстановочных знаков.

### Использование подстановочных знаков для поиска и замены

Например, звездочка служит для замены строки знаков (по шаблону «к\*й» будут найдены слова «круглый» и «квадратный»).

1. В меню **Правка** выберите команду **Найти** или **Заменить**.
2. Если флажок **Подстановочные знаки** не отображен на экране, нажмите кнопку **Больше**.
3. Установите флажок **Подстановочные знаки**.
4. Введите подстановочный знак в поле **Найти**. Выполните одно из следующих действий.
  - Чтобы выбрать подстановочный знак из списка, нажмите кнопку **Специальный**, выберите подстановочный знак, а затем в поле **Найти** введите дополнительный текст.
  - Введите подстановочный знак непосредственно в поле **Найти**.
5. Если необходимо заменить элемент, то в поле **Заменить на** введите заменяющий элемент.
6. Нажмите кнопку **Найти далее**, **Заменить** или **Заменить все**.  
Чтобы прервать процесс поиска, нажмите клавишу ESC.

### Примечания

Если флажок **Подстановочные знаки** установлен, выполняется поиск текста, в точности совпадающего с указанным. Обратите внимание, что флажки **Учитывать регистр** и **Только слово целиком** автоматически устанавливаются и становятся недоступными, поэтому они не могут быть сняты пользователем.

Чтобы найти знак, являющийся одним из подстановочных знаков, введите перед ним обратную косую черту (\). Например, введите \?, чтобы найти вопросительный знак.

### Подстановочные знаки для элементов, которые необходимо найти и заменить

#### Знак

Любой знак

Введите ?.

Например, если ввести «г?д», то будут найдены слова «гад», «гид» и «год».

Любая последовательность знаков

Введите \*.

Например, если ввести «г\*д», то будут найдены слова «год» и «город».

Начало слова

Введите <.

Например, если ввести «<(горо)», то будут найдены слова «город» и «гороскоп», а слово «огород» найдено не будет.

Конец слова

Введите >.

Например, если ввести «(ор)>», то будут найдены слова «ор» и «упор», а слово «оратория» найдено не будет.

Один из указанных знаков

Введите [ ].

Например, если ввести «кр[ea]н», то будут найдены слова «крен» и «кран».

Любой знак из указанного диапазона

Введите [-].

Например, если ввести «[в-н]оль», то будут найдены слова «голь» и «ноль». Диапазон должен быть сформирован по возрастанию.

Любой знак, кроме знаков, которые находятся в диапазоне, указанном в скобках

Введите [!x-z].

Например, если ввести «ко[!e-p]а», то будут найдены слова «кода» и «коса», а не слова «кожа» и «кора».

Ровно *n* раз повторенный предыдущий знак

Введите {*n*}.

Например, если ввести «ранен{2}ый», то будет найдено слово «раненный», а не слово «раненый».

Не менее  $n$  раз повторенный предыдущий знак или выражение

Введите {n,}.

Например, если ввести «ранен{1,}ый», то будут найдены слова «раненый» и «раненный».

От  $n$  до  $m$  раз повторенный предыдущий знак или выражение

Введите {n,m}.

Например, если ввести «10{1,3}», то будут найдены «10», «100» и «1000».

Один или более раз повторенный предыдущий знак или выражение

Введите @.

Например, если ввести «в@ерх», то будут найдены слова «верх» и «вверх».

### Примечания

Используйте круглые скобки для группировки подстановочных знаков и текста, а также указания порядка обработки знаков. Например, введите <(при)\*(ние)>, чтобы найти слова «призвание» и «приключение».

Для поиска выражения и последующего изменения порядка его компонентов используйте подстановочный знак \n. Например, введите (Годунов) (Борис) в поле **Найти** и \2 \1 в поле **Заменить на**, чтобы найти словосочетание «Годунов Борис» и заменить его на «Борис Годунов».

## Разрешение вопросов, связанных с поиском и заменой текста или других элементов

### Не удается найти или заменить некоторые рисунки и объекты

В Microsoft Word отсутствует возможность поиска и замены перемещаемых объектов, объектов WordArt, фигурного текста, подложек и графических объектов. Однако, если перемещаемый объект сделать встроенным объектом (Встроенный объект. Рисунок или другой объект, который помещается непосредственно в текст документа Microsoft Word в место вставки.), то появится возможность его поиска и замены.

### Поиск с использованием подстановочных знаков выполняется неправильно

Чтобы использовать подстановочные знаки, в меню **Правка** выберите команду **Найти** или команду **Заменить**, затем нажмите кнопку **Больше** и установите флажок **Подстановочные знаки**. Если поиск выполняется неправильно, попробуйте выполнить следующие действия.

Указывайте точное написание слова. При использовании подстановочных знаков поиск текста выполняется с учетом регистра букв. Например, при поиске **к\*т** будет найдено слово «кот», а не «Кот» или «КОТ». Если необходимо найти все комбинации букв (независимо от регистра), то используйте квадратные скобки. Например, результатами поиска **[Кк]\*[Тт]** будут слова «кот», «Кот» или «КОТ».

### Поиск специальных знаков с использованием подстановочных знаков выполняется неправильно

Если установлен флажок **Подстановочные знаки**, то Microsoft Word не распознает в поле **Найти** коды следующих знаков: знак обычной и концевой сноски, поля, знаки абзаца, разрывы разделов или пробел. Для поиска этих знаков их следует заменить в поле **Найти** перечисленными ниже кодами. (Необходимо отметить, что для полей такая замена невозможна.)

Чтобы найти знак обычной или концевой сноски:

введите ^2.

Необходимо отметить, что знаки обычных и концевых сносок не различаются.

Чтобы найти знак абзаца:

введите ^13.

Чтобы найти разрыв раздела:

введите ^12.

Разрывы страниц и разрывы разделов не различаются.

Чтобы найти пробел:

введите *space* {1,}.

## Использование встроенной функции поиска

Selection.Find.MatchWholeWord = True 'поиск слова целиком

Selection.Find.Text = "к" 'поиск вхождения

Selection.Find.MatchWildcards = True 'поиск по шаблону

Selection.Find.Replacement.Text = "к" 'заменит именно так, как было, с учетом регистра

' Selection.Find.Replacement.Text = "" если сам текст надо оставить как есть

Selection.Find.Replacement.Font.Bold = True, курсив - Italic

Selection.Find.Replacement.Font.ColorIndex = wdBlue 'можно цифрами - всего 16 цветов

Selection.Find.Execute Replace:=wdReplaceAll 'заменять все найденные

## Задание семинара:

Заменяем все предлоги к на латинскую букву k, напишем ее жирным шрифтом и покрасим 4 цветом палитры

```
Sub ZamenaK()
```

```
Selection.Start = ActiveDocument.Range.Start
```

```
Selection.End = ActiveDocument.Range.End
```

```
Selection.Find.MatchWholeWord = True
```

```
Selection.Find.Text = "к"
```

```
Selection.Find.Replacement.Text = "k"
```

```
Selection.Find.Replacement.Font.Bold = True
```

```
Selection.Find.Replacement.Font.ColorIndex = 4
```

```
Selection.Find.Execute Replace:=wdReplaceAll
```

```
End Sub
```

## Задание для самостоятельной работы:

В первых 3 абзацах текста найти все слова, начинающиеся на букву "в", отметить такие слова курсивом и окрасить их в красный цвет.

# Варианты первого практического задания по курсу «Языки управления приложениями» Microsoft Word

- 1. PCO.** Дан текст в Microsoft Word. Написать макрос, выделяющий (цветом или стилем шрифта) определенные слова исходного текста. В первом абзаце выделяются все слова, начинающиеся с буквы, заданной пользователем, во-втором – с буквы, следующей за ней по алфавиту и т.д. В диалоге с пользователем уточняются: 1) тип выделения (различные цвета и/или стиль шрифта), 2) буква алфавита, используемая для выделения слов первого абзаца. Следует помнить, что палитра цветов *ColorIndex* ограничивается 16 цветами, а букв в алфавите 33. Если текст большой, то после 33-й буквы алфавита и после 16-го цвета палитры следует продолжать выделение, используя 1 букву алфавита и 1 цвет палитры.
- 2. SB.** Дан текст в Microsoft Word. Написать макрос, который по заданному тексту собирает коллекции слов, начинающихся с одной и той же буквы. В диалоге с пользователем уточняется диапазон букв, используемых при составлении коллекций, например **б-к**. Исходный текст заменяется на новый, в котором в первом абзаце расположены все слова исходного текста, начинающиеся с первой буквы заданного диапазона, во втором – со следующей буквы диапазона и т.д. Например, для диапазона **б-к** первый абзац будет состоять из слов, начинающихся на букву **б**, во втором – на **в**, всего получится 11 абзацев.
- 3. PCM.** Дан текст в Microsoft Word. Написать макрос, собирающий коллекции из всех слов, начинающихся с заданных пользователем букв. Результатом работы макроса является исходный текст, в котором выделены (цветом или шрифтом) те слова, коллекция которых оказалась наибольшей. В диалоге с пользователем уточняются: 1) тип выделения (различные цвета и/или стиль шрифта), 2) начальные буквы слов для сбора коллекций.
- 4. SC.** Дан текст в Microsoft Word. Написать макрос, собирающий коллекции слов в зависимости от их длины, например коллекция слов длины 7. В диалоге с пользователем задается диапазон интересующих его длин, например 3-5 (должна быть предусмотрена возможность выбора «все длины») Исходный текст заменяется на новый, в котором в первом абзаце расположены слова, длина которых равна первому числу диапазона (в нашем примере 3), во втором абзаце слова, длина которых на единицу больше (в нашем примере 4) и т.д.
- 5. PZP.** Дан текст в Microsoft Word. Написать макрос, выделяющий (цветом или стилем шрифта) те слова, после которых стоит один из заданных пользователем знаков препинания, а так же подсчитывающий количество знаков препинания разного вида. В диалоге с пользователем уточняется тип выделения (различные цвета и/или стиль шрифта), а так же выделяемые и подсчитываемые знаки препинания.
- 6. PCZ.** Дан текст в Microsoft Word. Написать макрос, выделяющий (цветом или стилем шрифта) те слова, которые написаны с большой буквы, но не являются началом предложения (перед этим словом не стоит точка, вопросительный знак, восклицательный знак или многоточие). В диалоге с пользователем уточняется тип выделения (различные цвета и/или стиль шрифта).
- 7. PBL.** Дан текст на русском языке в Microsoft Word. Написать макрос, который находит и выделяет (цветом или стилем шрифта) латинские буквы или цифры, входящие в этот текст. Особо надо выделить слова, целиком состоящие из латинских букв или цифр. В диалоге с пользователем уточняется тип выделения (различные цвета и/или стиль шрифта).
- 8. ZMX.** Дан текст в Microsoft Word. Написать макрос, изменяющий случайным образом порядок букв каждого слова, при этом первые N и последние M букв слова остаются неизменными. Например, при начальных параметрах N=2, M=1 слово “случайным” преобразуется в “слйауынчм”. В диалоге с пользователем уточняются параметры N и M.

9. **PBM.** Дан текст в Microsoft Word. Написать макрос, который подсчитывает количество вхождений каждой буквы в текст и выделяет (цветом или стилем шрифта) те слова, в которые входит самая часто встречающаяся буква. В диалоге с пользователем уточняется тип выделения (различные цвета и/или стиль шрифта).
10. **PBI.** Дан текст в Microsoft Word. Написать макрос, подсчитывающий сколько раз каждая буква, из числа заданных пользователем, встретилась в тексте и подсвечивающий во всем тексте эти буквы цветом, который зависит от частоты их встречаемости. Например, если какая-то буква встретилась в тексте 14 раз, то она должна быть выделена во всем тексте цветом с  $\text{ColorIndex} = 14$ , если буква встретилась 18 раз, то цвет должен иметь индекс  $18 \bmod 16 = 2$ . В диалоге с пользователем уточняется диапазон подсвечиваемых букв, например **б-е**.
11. **ZBZ.** Дан текст в Microsoft Word. Написать макрос, подсчитывающий частоту встречаемости букв в тексте. Те буквы, частота встречаемости которых превышает некоторое заданное пользователем число, должны быть заменены на аналогичные им большие (заглавные) и выделены (цветом или стилем шрифта). В диалоге с пользователем уточняется порог частоты встречаемости и способ выделения.
12. **ZCF.** Дан текст в Microsoft Word. Написать макрос, находящий все числа в тексте и подсчитывающий значение некоторой функции от этих чисел, а затем увеличивающий найденные числа на подсчитанное значение. Функция выбирается пользователем в диалоге из 6-8 возможных, например: сумма чисел, их произведение, минимум, максимум, среднее арифметическое и т.д.
13. **ZCM.** Дан текст в Microsoft Word. Написать макрос, просматривающий заранее выделенный пользователем фрагмент текста и собирающий коллекцию встретившихся в нем букв. Затем во всем тексте выделяются слова, удовлетворяющие заданному пользователем условию, например: выделяются слова которые состоят только из букв коллекции, или в которых есть буквы не входящие в собранную коллекцию и т.д. В диалоге с пользователем ему предоставляются на выбор 3-4 условия.
14. **ZAF.** Дан текст в Microsoft Word. Написать макрос, просматривающий заранее выделенный пользователем фрагмент текста и подсчитывающий значение некоторой функции от встретившихся в этом фрагменте цифр. Функция выбирается пользователем в диалоге из 6-8 возможных, например: сумма чисел, их произведение, минимум, максимум, среднее арифметическое и т.д. Подсчитанное значение используется для преобразования исходного текста, оно определяет максимальное число слов в абзаце: слова, выходящие за эту границу, из исходного текста удаляются.
15. **ZAU.** Дан текст в Microsoft Word. Написать макрос, подсчитывающий количество слов в указанном пользователем абзаце и оставляющий в тексте только те абзацы, количество слов в которых удовлетворяет некоторому условию. Условие выбирается пользователем в диалоге из 5-6 возможных, например: количество слов не должно превышать подсчитанное число, количество слов равно подсчитанному числу и др...
16. **SA.** Дан текст в Microsoft Word. Написать макрос, подсчитывающий длину каждого абзаца (количество слов в нем) и сортирующий в исходном тексте эти абзацы в зависимости от их длины. В диалоге с пользователем уточняется способ сортировки, например: по возрастанию количества слов, по убыванию количества слов и т.д.



# ШАШКИ (сложность высокая)

С помощью макроса реализовать в Excel интерфейс игры **русские шашки**.

## Правила игры:

Игра проходит на доске размером 8x8 только на чёрных клетках. Играют два игрока обычно чёрным и белым цветом. У каждого игрока в начальной позиции по 12 фишек — так называемых «простых» шашек, которые занимают первые три ряда с каждой стороны. В ходе игры шашки движутся по черным полям и могут вставать только на незанятые клетки. «Простая» шашка может ходить по диагонали вперёд на одну клетку или бить вперед или назад. При бое шашка движется по диагонали на две клетки, перепрыгивая через шашку или дамку соперника, которая при этом снимается с доски (съедается). Если из нового положения бьющей шашки можно побить другую шашку соперника, то ход продолжается, и т. д. При прохождении любого поля из последнего от игрока горизонтального ряда, «простая» шашка превращается в дамку и продолжает бой по правилам дамки. Дамка может ходить по диагонали в любом направлении на любое число полей. Игрок на своём ходе обязан побить шашку противника, если у него есть такая возможность. Если есть несколько вариантов хода с боем, то игрок выбирает любой из них. Пропуск хода не допускается. Цель игры — съесть или «запереть» (лишить возможности хода) все шашки противника.

## Постановка задачи:

	a	b	c	d	e	f	g	h		
1										
2									игроку1	счет:
3									игрок1	12
4									игрок2	12
5										
6										
7									ходит:	игрок1
8										

На активном листе должно быть подсвечено игровое поле 8\*8, на котором в каждый момент времени отображается текущее состояние игры. Вместо фишек используется соответствующая раскраска клеток игрового поля.

Перед началом игры в диалоге уточняются имена игроков и цвета, которыми они хотят играть. В начале игры должна быть выставлена начальная игровая позиция (см. рисунок). В течение игры рядом с полем показывается текущий счет и имя игрока, который ходит в данный момент. По окончании игры должно быть выведено сообщение с именем победителя.

Ходы осуществляются кликом мышкой в ячейки игрового поля: клик в шашку означает ее

выделение, клик в пустое поле обозначает ячейку, в которую игрок хочет переместить выделенную шашку. Автоматически проверяется допустимость данного хода.

## Ним (сложность низкая)

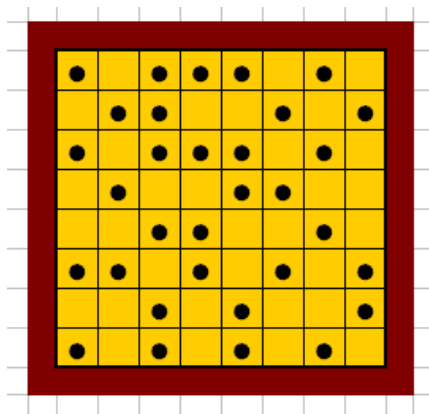
С помощью макроса реализовать в Excel интерфейс игры **ним**.

### Правила игры:

При игре в «ним» два соперника поочередно берут фишки с игрового поля, на котором в начальный момент находится несколько рядов фишек, причем в каждом ряду имеется хотя бы одна фишка. При каждом ходе игрок может взять произвольное количество фишек из одного ряда (но не менее одной). Выигрывает тот игрок, кто берет последнюю фишку.

### Постановка задачи:

На активном листе должно быть подсвечено игровое поле, на котором в каждый момент времени отображается текущее состояние игры. До начала игры в диалоге пользователь указывает размер игрового поля. В начале игры случайным образом расставляются фишки, количество которых зависит от размера поля. В течение игры рядом с полем показывается текущий счет и имя игрока, который ходит в данный момент. По окончании игры должно быть выведено сообщение с именем победителя. Ходы осуществляются кликом мышкой в ячейки игрового поля. Клик в фишку выделяет ее, клик в выделенную фишку отменяет выделение. После выделения игроком фишек, которые он хочет взять на своём ходе, тем или иным образом подтверждает свой ход, после чего автоматически проверяется его допустимость



## Тик-такс (сложность низкая)

С помощью макроса реализовать в Excel интерфейс игры **тик-такс**.

Правила игры в «тик-такс» отличаются от правил игры в «ним» лишь тем, что на каждом ходе можно брать только рядом стоящие фишки одного ряда.

Требование к интерфейсу см. в варианте для игры «ним».

## Нижняя левая (сложность низкая)

С помощью макроса реализовать в Excel интерфейс игры **нижняя левая**.

Имеется прямоугольное поле  $N \times N$ , ( $5 \leq N \leq 7$ ) из клеток, заполненных фишками. Каждый из двух игроков по очереди выбирает фишку из оставшихся на поле и убирает ее вместе со всеми фишками, расположенными правее и выше выбранной. Проигрывает тот, кто забирает фишку, стоящую в нижнем левом углу.

Требование к интерфейсу см. в варианте для игры «ним».



## Цветовая экспансия (сложность средняя)

С помощью макроса реализовать в Excel интерфейс игры **цветовая экспансия**.

### Правила игры:

Игра проходит на поле размером  $N \times N$  ( $N \geq 2$ ). В начале игры все клетки поля раскрашены в  $M$  цветов ( $M \geq 3$ ) так, что соседние клетки имеют разные цвета. При этом необходимо, чтобы клетки с координатами  $(1,1)$  и  $(N,N)$  были окрашены в разные цвета. Играют два игрока, строя свои «империи» из углов поля, т.е. клеток с координатами  $(1,1)$  первый игрок и  $(N,N)$  — второй. В начале игры «империя» каждого игрока состоит из одной угловой клетки. Игроки ходят по очереди. Ход игрока заключается в выборе одного из  $M$  цветов, при этом выбранный цвет должен отличаться от текущего цвета игрока (цвета его угловой клетки) и от текущего цвета противника. После того, как цвет выбран, вся «империя» игрока перекрашивается в выбранный цвет. При этом клетки, граничащие с «империей» игрока и имеющие выбранный на текущем ходу цвет, присовокупляются к ней. После каждого хода подсчитываются очки каждого игрока — размеры их «империй». Игра заканчивается, когда на поле не останется клеток, входящих в какую-либо империю. Выигрывает тот, чья «империя» к концу игры будет больше. При равенстве объявляется ничья.



### Постановка задачи:

Перед началом игры в диалоге уточняются имена игроков, размер поля и количество цветов, после чего в соответствии с правилами игры генерируются начальные цвета ячеек и на активном листе появляется игровое поле. В течение игры рядом с полем показывается текущий счет и имя игрока, который ходит в данный момент.

Ходы могут осуществляться кликом мышкой в ячейки, расположенные рядом с игровым полем и раскрашенные цветами, использующимися в игре. После каждого хода игрока проверяется его допустимость. В случае недопустимого хода игроку выдается сообщение об ошибке и предоставляется возможность сделать ход еще раз. Если игрок сделал допустимый ход, то после него происходит смена цвета «империи» и ход переходит к противнику.

## Тренировка памяти (сложность низкая)

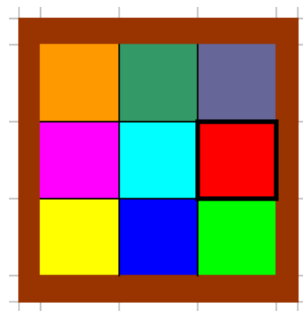
С помощью макроса реализовать в Excel интерфейс игры **тренировка памяти**.

### Правила игры:

Игра проходит на поле размером  $N \times M$  ( $3 \leq N, M \leq 7$ ). Каждая ячейка игрового поля покрашена в свой цвет, цвета не повторяются. Играет один человек. Компьютер показывает игроку некоторую последовательность ячеек (по очереди выделяет их). Игрок должен запомнить и повторить эту последовательность. После трех ошибок игра останавливается.

### Постановка задачи:

Перед началом игры в диалоге уточняется размер поля и длина последовательности  $K$  ( $N \geq 3$ ), с которой игрок хочет начать. После этого компьютер случайным образом выбирает  $K$  ячеек игрового поля и запоминает их последовательность. Затем демонстрирует ее (сначала жирным обводится первая ячейка, потом выделение убирается, затем вторая и т.д.). После этого ход переходит к игроку. Он должен, кликая по ячейкам, выделять их в той же последовательности, как показывал компьютер. Если это удастся, то ход считается успешным. В этом случае компьютер добавляет еще одну ячейку в конец последовательности и снова демонстрирует ее. Если игрок в свой ход ошибся, то появляется сообщение и компьютер демонстрирует последовательность еще раз, не изменяя ее. Игра останавливается после трех ошибок. Результатом игры является длина последней успешно повторенной последовательности.



## Прямоугольники (сложность средняя)

С помощью макроса реализовать в Excel интерфейс игры **прямоугольники**.

### Правила игры:

На поле  $N \times M$  ( $N \geq 2$ ,  $M \geq 2$ ) положили  $K$  ( $K \geq 2$ ) прямоугольников один поверх другого в случайном порядке. Длины сторон прямоугольников выражаются целым числом клеток. Прямоугольники не выходят за границы поля.

Получившуюся ситуацию записали в таблицу чисел (каждой клетке поля соответствует клетка таблицы). Если клетка поля не закрыта прямоугольником, то в соответствующую клетку таблицы записали число 0. Если же клетка закрыта одним или несколькими прямоугольниками, то в соответствующую клетку таблицы записали число, соответствующее самому верхнему прямоугольнику, закрывающему эту клетку. Два разных прямоугольника могут состоять из ячеек с одинаковыми числами.

0	2	2	2	2	0	0
0	2	2	2	2	0	0
1	1	2	4	4	4	4
1	1	0	4	4	4	4
1	1	0	0	0	0	0
1	1	0	0	0	0	0

По содержимому таблицы требуется определить положение и размеры прямоугольников.

### Постановка задачи:

Перед началом игры в диалоге уточняются размер поля и количество прямоугольников, после чего в соответствии с правилами на игровом поле расставляются цифры.

В течение игры пользователь выделяет прямоугольники, которые, по его мнению, были загаданы компьютером, и, после подтверждения хода, этот прямоугольник убирается с доски. В этот момент требуется проверить, что ход игрока корректен, т.е. выделенные ячейки содержат одинаковые числа. Если это так, то выделенные числа убираются с поля, а освободившиеся ячейки заполняются числами, которые стояли бы в них, если убранного прямоугольника на поле вообще не было (т.е., если убранный прямоугольник перекрывал собой другие числа, теперь они станут видны, если под ним чисел не было, ячейки заполняются нулями). После этого игрок может делать следующий ход. Если ход, сделанный игроком, некорректен, то показывается сообщение об ошибке и дается возможность сделать новый ход. В ходе игры рядом с полем показывается, сколько прямоугольников осталось отгадать. Цель игры убрать с поля все цифры большие нуля.

Игра считается успешно законченной, если игровое поле заполнено нулями, а количество отгаданных игроком прямоугольников не больше количества загаданных компьютером в начале игры.

## Найди пару (сложность низкая)

С помощью макроса реализовать в Excel интерфейс игры **найди пару**.

### Правила игры:

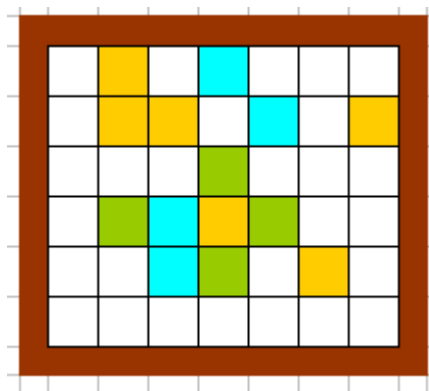
Игра проходит на поле размером  $N \times M$  ( $N \geq 2$ ,  $M \geq 2$ ,  $M$  - четное). В начале игры каждой клетке поля случайным образом приписывается свой цвет (но игроку цвета не показываются). Количество полей одинакового цвета – четно. Количество цветов  $K$  ( $K \geq 2$ ) определено заранее. Игрок за ход может посмотреть цвета двух ячеек. Если они совпадут, то цвет ячеек сохранится, если нет – то цвет ячеек опять скроется. Задача игрока за минимальное количество ходов открыть цвета всех ячеек.

### Постановка задачи:

Перед началом игры в диалоге уточняются размер поля и количество цветов, в которые будут краситься ячейки. После этого игроку показывается игровое поле белого цвета и предоставляется возможность сделать ход. После каждого хода происходит проверка открытых на этом ходе цветов. Если цвета не совпали, то ячейки опять становятся белыми. В ходе игры рядом с полем показывается, сколько ходов было сделано на текущий момент.

### Усложнение задачи (плюс один балл):

Ввести ограничение по времени: пользователю дается 5 минут на игру. Если игрок открыл все поле, а время еще не истекло, то создается новое игровое поле, в котором количество участвующих в раскраске цветов увеличивается на один. При успешном открытии пары время, оставшееся игроку на игру, увеличивается на 10 секунд. При истечении времени игра останавливается. Результат игрока - количество пар, которые удалось отгадать за все время игры.



## Прятки (сложность низкая)

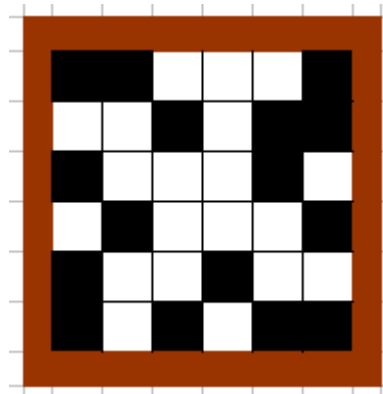
С помощью макроса реализовать в Excel интерфейс игры **прятки**.

### Правила игры:

Игра проходит на поле размером  $N \times M$  ( $M, N \geq 6$ ). Играют двое, ходят по очереди. В начале игры поле пустое. В свой ход игрок может перекрасить одну белую клетку. Клетка считается «видной» по горизонтали, если в одной строке с ней на поле уже есть хотя бы одна перекрашенная клетка. Аналогичным образом клетка может быть «видна» по вертикали и обоим диагоналям. Если игрок в свой ход покрасил ячейку, которая «не видна» по всем четырем направлениям, он получает 4 балла. Если клетка видна по одному направлению, то количество начисленных баллов будет 3, если по двум, то 2, если по трем, то 1. Если клетка «видна» во всех четырех направлениях, то она «плохая», ее перекрашивать нельзя. Игра идет до тех пор, пока все белые ячейки не станут «плохими». Выигрывает тот, кто в процессе игры набрал большее количество баллов.

### Постановка задачи:

Перед началом игры в диалоге уточняются имена игроков, размер поля и цвет, которым будут закрашиваться ячейки. После этого игроки делают ходы по очереди. В свой ход игрок кликает на клетку, которую хочет перекрасить. Программа проверяет, не является ли клетка «плохой». Если клетка «плохая», то выдается сообщение и право хода переходит к сопернику, баллы игроку не добавляются. Если клетку перекрасить можно, то программа добавляет игроку баллы исходя из правил игры, перекрашивает ячейку, проверяет, остались ли поля, которые могут быть закрашены в следующий ход. Если такие поля есть, то передает право хода сопернику. Если нет, то игра заканчивается и сообщается имя победителя.



## Крестики-нолики (сложность низкая)

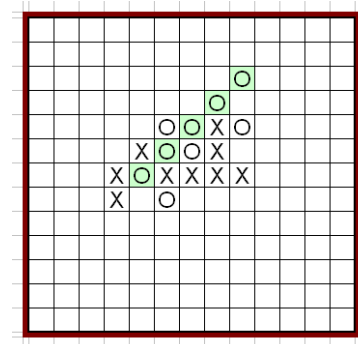
С помощью макроса реализовать в Excel интерфейс игры **крестики-нолики**.

### Правила игры:

Имеется прямоугольное поле  $N \times N$  ( $10 \leq N \leq 30$ ). Играют двое, ходят по очереди. Фишки одного игрока - «крестики» (или поля одного одного цвета), а другого – «нолики» (или поля другого цвета). В начале игры все клетки доски пустые.

При своем ходе игрок ставит свою фишку в любую свободную клетку.

Выигрывает тот, кто первым выстроил 5 своих фишек в (сплошной) ряд по горизонтали, вертикали или диагонали. Возможна и ничья, если никто не выиграл и на доске не осталось пустых клеток.



### Постановка задачи:

На активном листе должно быть подсвечено игровое поле, на котором в каждый момент времени отображается текущее состояние игры. До начала игры в диалоге пользователь указывает размер игрового поля. В течение игры рядом с полем показывается имя игрока, который ходит в данный момент. По окончании игры должно быть выведено сообщение с именем победителя. Ходы осуществляются кликом мышкой в ячейки игрового поля. После того, как игрок сделал ход, проверяется, не стал ли он победителем в результате этого хода. Если нет, то ход переходит другому игроку.

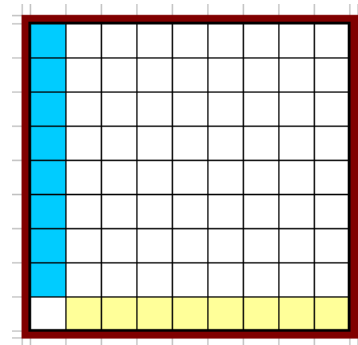
## Додж (to dodge – увертываться) (сложность средняя)

С помощью макроса реализовать в Excel интерфейс игры **додж**.

### Правила игры:

Имеется прямоугольное поле  $N \times N$  ( $4 \leq N \leq 10$ ). Играют двое, ходят по очереди. Один игрок играет желтыми фишками, другой синими. Вместо фишек используется соответствующая раскраска клеток игрового поля. В начале игры на доске уже выставлено по  $(N-1)$  фишке каждого цвета – так, как показано на рисунке.

При каждом ходе желтая фишка может перейти в соседнюю свободную клетку влево, вверх или вправо (но не вниз), а синяя – вверх, вправо или вниз (но не влево). Подойдя к верхнему (правому) краю доски, желтая (синяя) фишка может уйти с доски. Запрещается «запирать» фишки противника (не давать ему ходить); тот, кто сделает это, сразу проигрывает.



Выигрывает тот игрок, который быстрее уведет свои фишки с доски.

### Постановка задачи:

На активном листе должно быть подсвечено игровое поле, на котором в каждый момент времени отображается текущее состояние игры. До начала игры в диалоге пользователь указывает размер игрового поля. В течение игры рядом с полем показывается имя игрока, который ходит в данный момент. По окончании игры должно быть выведено сообщение с именем победителя. Ходы осуществляются кликом мышкой в ячейки игрового поля. После того, как игрок сделал ход, проверяется, не стал ли он победителем в результате этого хода. Если нет, то ход переходит другому игроку.

## Уголки (сложность низкая)

С помощью макроса реализовать в Excel интерфейс игры **уголки**.

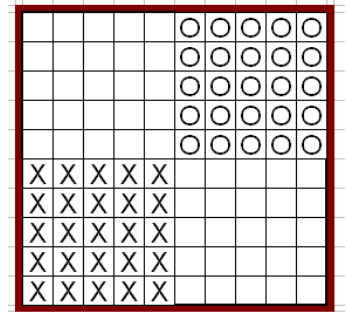
### Правила игры:

Имеется прямоугольное поле  $N \times N$  ( $6 \leq N \leq 16$ ). Играют двое, ходят по очереди. Фишки одного игрока - «крестики» (или поля одного одного цвета), а другого – «нолики» (или поля другого цвета). В начале игры фишки одного игрока уже выставлены в его «базе» (в квадрате из  $M \times M$  клеток), расположенной в левом нижнем углу доски, а фишки другого игрока – в другой базе, расположенной в правом верхнем углу ( $M$  заранее фиксировано и должно равняться примерно  $N/2$ ).

При своем ходе игрок может:

- сделать «шаг» - передвинуть любую свою фишку в любую соседнюю (по горизонтали или вертикали) свободную клетку;
- сделать «прыжок» - переставить любую свою фишку через любую соседнюю (свою или чужую) фишку, если за ней находится свободная клетка;
- сделать любое число «прыжков» подряд, если это возможно.  
(Замечание: нельзя в одном ходе сочетать «шаги» и «прыжки».)

Выигрывает тот, кто быстрее переведет свои фишки в базу противника.



## Игра «Жизнь» (сложность средняя)

С помощью макроса реализовать в Excel интерфейс игры **жизнь**.

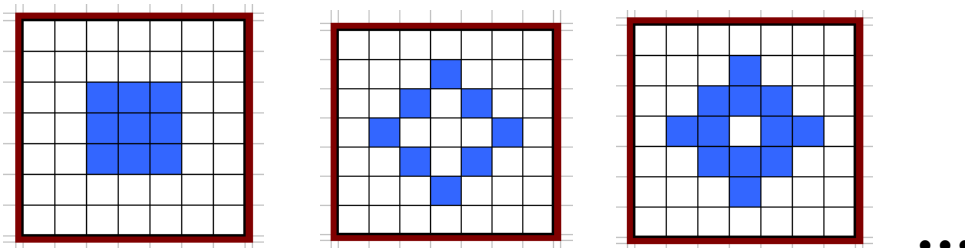
Место действия этой игры — размеченная на клетки поверхность, ограниченная или замкнутая. Каждая клетка на этой поверхности может находиться в двух состояниях: быть живой или быть мёртвой. Клетка имеет восемь соседей. Распределение живых клеток в начале игры называется первым поколением. Каждое следующее поколение рассчитывается на основе предыдущего по таким правилам:

- пустая (мёртвая) клетка рядом с тремя живыми клетками-соседями оживает;
- если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае (если соседок меньше двух или больше трёх) клетка умирает (от «одиночества» или от «перенаселённости»).

Игрок не принимает прямого участия в игре, а лишь расставляет «живые» клетки, которые взаимодействуют согласно правилам уже без его участия.

### Постановка задачи:

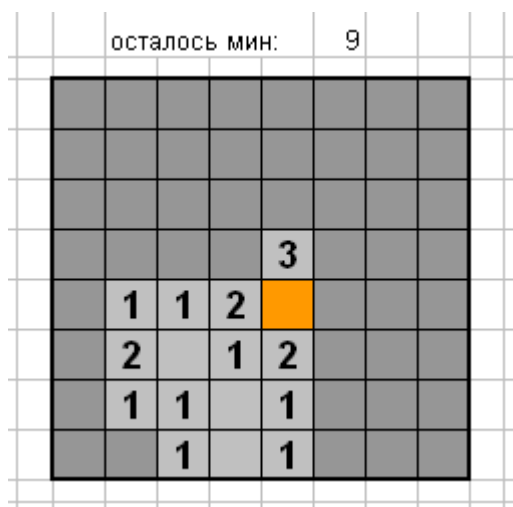
На активном листе должно быть подсвечено игровое поле, на котором в каждый момент времени отображается текущее состояние игры. До начала игры в диалоге пользователь указывает размер игрового поля и его тип (замкнутое или нет), после чего ему дается возможность отобразить на заданном поле первое поколение. В процессе игры пользователь должен иметь возможность просмотра развития колонии как по шагам, так и непрерывного развития на несколько шагов вперед. В случае гибели колонии игра останавливается.





## САПЕР (minesweeper) (сложность средняя)

С помощью макроса реализовать в Excel интерфейс игры сапер.



### Правила игры:

Игра проходит на прямоугольном поле произвольного размера, часть клеток которого заминировано. Количество мин зависит от размера поля. В начале игры расположение мин игроку неизвестно, он должен последовательно открывать клетки. При попадании на заминированную клетку игра останавливается и игроку засчитывается поражение. При попадании на клетку, в которой нет мины, пользователю показывается число, означающее, сколько мин находится в соседних 8 ячейках. Игрок имеет дополнительную возможность, не открывая содержимое клетки, пометить ее как заминированную. Цель игры - определить, в каких ячейках поля находятся мины. Когда все заминированные клетки помечены, игра останавливается. В случае, если заминированные ячейки определены неверно, игроку засчитывается поражение.

### Постановка задачи:

На активном листе должно быть подсвечено игровое поле, на котором в каждый момент времени отображается текущее состояние игры. Размер игрового поля указывается пользователем в диалоге перед началом игры. Клетки, в которых расположены мины, определяются случайным образом. Рядом с игровым полем показывается, сколько мин осталось обнаружить. Когда количество клеток, помеченных игроком как заминированные, становится равным количеству мин, автоматически проверяется, совпадают ли помеченные клетки с теми, в которых реально расположены мины. Пользователь получает сообщение о результате игры и причинах ее окончания.

## Змейка (сложность средняя)

С помощью макроса реализовать в Excel интерфейс игры змейка.



### Правила игры:

Игра проходит на прямоугольном поле произвольного размера. Игрок управляет движением змейки на игровом поле. На поле определенным цветом выделена бонусная ячейка. При прохождении змейки через бонусную ячейку длина змейки увеличивается, а бонусная ячейка меняет свои координаты. При увеличении длины змейки ее скорость увеличивается. Игра заканчивается, когда змейка наткнется на границу игрового поля или когда она напозает на саму себя.

### Постановка задачи:

На активном листе должно быть подсвечено игровое поле, на котором в каждый момент времени отображается текущее состояние игры. До начала игры в диалоге пользователь указывает размер игрового поля и начальную длину змейки. Начальная длина не должна превышать половину длины или ширины игрового поля. В ходе игры рядом с полем показывается текущая длина змейки и ее скорость. Координаты бонусной ячейки определяются случайным образом. Игрок с помощью клавиатуры изменяет направление движения змейки. При окончании игры пользователю должны быть показаны итоговая длина змейки, ее скорость и причина окончания игры.

## Морской бой (сложность средняя)

С помощью макроса реализовать в Excel интерфейс игры **морской бой**.

### Правила игры:

	1	2	3	4	5	6	7	8	9	10	
а			•	•	•						а
б			•	X	•			X			б
в			•	X	•			X			в
г			•	X	•						г
д			•	•	•						д
е		•	•	•		•	•				е
ж		•	X	•	•	•			•		ж
з		•	•	•							з
и		•						•			и
к											к
	1	2	3	4	5	6	7	8	9	10	

Игра проходит на поле размером 10x10. До игры на поле расставляются корабли: 1 четырехпалубный, 2 трехпалубных, 3 двухпалубных и 4 однопалубных. При размещении корабли не могут касаться друг друга даже углами. Расстановка кораблей игроку не показывается. Игрок последовательно делает выстрелы по полю (открывая клетки поля), при этом проверяется, попал ли игрок в корабль. Если да, то клетка помечается крестом, и, если при этом все остальные клетки корабля тоже помечены крестом, то корабль считается подбитым. Если при выстреле корабль не был поражен, то в это поле ставится точка (промах). Цель игры подбить все корабли за как можно меньшее число выстрелов.

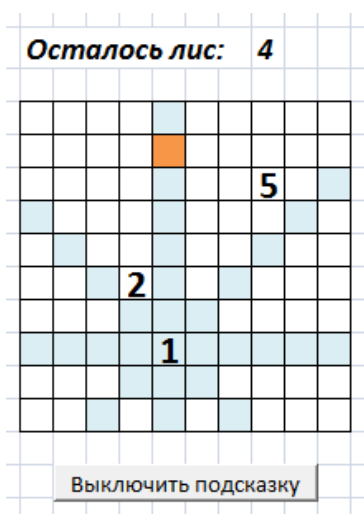
### Постановка задачи:

На активном листе должно быть подсвечено игровое поле 10\*10, на котором в каждый момент времени отображается текущее состояние игры. До начала игры случайным образом создается расстановка кораблей. Пользователь делает выстрелы кликом на клетках поля. При гибели корабля автоматически вокруг него все поля помечаются точками (промахами), а сам он подсвечивается особым образом. По окончании игры показывается, сколько выстрелов сделал игрок. Желательно указывать рядом с полем во время игры сколько и каких кораблей осталось подбить.

## Охота на лис (сложность средняя)

С помощью макроса реализовать в Excel интерфейс игры **охота на лис**.

### Правила игры:



Игра проходит на поле размером 10x10. В пяти клетках поля спрятаны пять лис, по одной в клетке. Цель игры: найти всех лис. Для поиска игрок указывает ячейку поля и получает ответ:

- поле окрашивается рыжим цветом (т.е. лиса найдена)
- в поле появляется цифра от 0 до 5 в зависимости от того, сколько лис видно из этой ячейки.

Лиса считается видимой из ячейки, если она находится в одной вертикали, горизонтали или диагонали от нее.

### Постановка задачи:

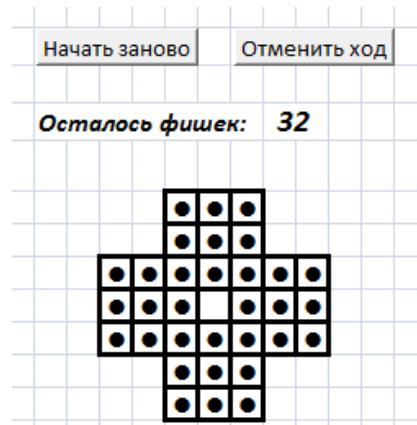
На активном листе должно быть подсвечено игровое поле 10\*10, на котором в каждый момент времени отображается текущее состояние игры. До начала игры случайным образом создается расстановка лис. Пользователь делает ходы кликом на клетках поля. При нахождении лисы поле подсвечивается рыжим цветом, в противном случае появляется число видимых лис. Требуется предусмотреть подсказку: если по текущему состоянию игры можно точно сказать, что в каких-то ячейках лисы быть не может, такие ячейки требуется подсветить другим цветом. Подсказку можно включать и выключать кнопкой, расположенной на игровом поле. По окончании игры показывается, сколько ходов сделал игрок.



## Solitaire (сложность низкая)

С помощью макроса реализовать в Excel интерфейс головоломки **solitaire**.

### Правила игры:



Исходное положение головоломки показано на рисунке. Ход заключается в перепрыгивании по горизонтали или по вертикали некоторой выбранной фишки через соседнюю фишку на свободное поле, при этом фишка, через которую перепрыгнули, снимается с поля. Головоломка считается решенной, если осталась одна фишка в центре игрового поля.

### Постановка задачи:

На активном листе должно быть подсвечено игровое поле, на котором в каждый момент времени отображается текущее состояние игры. До начала игры фишки расставляются таким образом, как показано на рисунке. Ходы осуществляются кликом мышкой в ячейки игрового поля: клик в поле с фишкой означает ее выделение, клик в пустое поле обозначает ячейку, в которую игрок хочет переместить выделенную фишку. Автоматически проверяется допустимость данного хода. Если ход возможен, то фишка перемещается, а с поля снимается фишка, через которую был произведен прыжок. Головоломка считается решенной, если осталась одна фишка и она расположена в центре игрового поля. Рядом с полем необходимо указывать текущее количество фишек. Желательно, чтобы на поле были кнопки, позволяющие отменить последние N ходов или начать игру заново.

## Сумма (сложность средняя)

С помощью макроса реализовать в Excel интерфейс игры **сумма**.

### Правила игры:

Текущий счет:		-10				
ходит игрок1						
1	-13	2	-22	17	-3	-11
22	-5	-7	3	-25	-7	11
14	44	-15	-7	-22	-4	-22
21	-7	5	0	-14	-12	17
2	-17	3	-7	-13	20	-5
-1	-4	14	-9	25	3	-8
22	14	35	-22	-6	-5	-2

Игра проходит на поле размером  $N \times N$ , где  $N$  - нечетно. В центральной клетке расположено число 0, сумма всех чисел равна нулю. В центре находится фишка. Играют два игрока, поочередно делая ходы. За один ход игрок может переместить фишку в любую соседнюю клетку, в которую еще не было сделано ходов. После хода в ячейку она закрывается и в нее больше нельзя ходить. При каждом ходе вычисляется счет игры, который равен текущему счету плюс значение ячейки, в которую был сделан ход. Игра заканчивается, когда нельзя сделать ни одного хода. Если в результате игры счет получился отрицательным, то побеждает первый игрок. Если положительный, то второй.

### Постановка задачи:

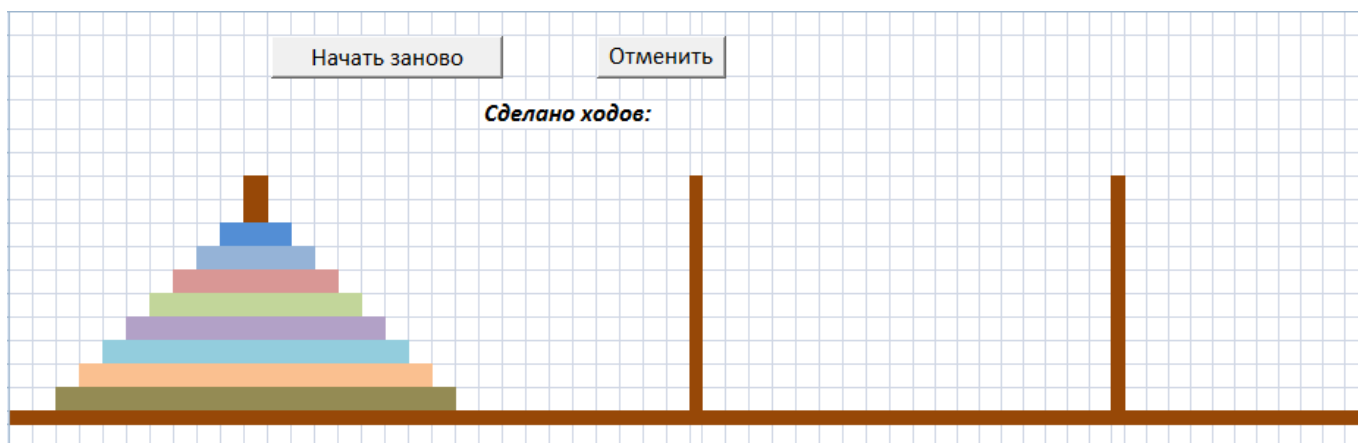
На активном листе должно быть подсвечено игровое поле, на котором в каждый момент времени отображается текущее состояние игры. До начала игры в диалоге пользователь указывает размер игрового поля. В начале игры в центр поля записывается число 0 и туда ставится игровая фишка (поле можно подсветить каким-то цветом). Остальные цифры расставляются случайным образом, сумма всех цифр поля должна быть равна 0. В течение игры рядом с полем показывается текущий счет и имя игрока, который ходит в данный момент. Ход осуществляется кликом в ячейку игрового поля, которая находится по соседству с текущим положением фишки, после чего автоматически проверяется допустимость этого хода и меняется текущий счет игры. Если допустимых ходов нет, то объявляется победитель.

## Ханойская башня (сложность низкая)

С помощью макроса реализовать в Excel интерфейс головоломки **ханойская башня**.

### Правила игры:

Даны три стержня, на один из которых нанизаны восемь колец, причем кольца отличаются размером и лежат меньшее на большем. Кольца можно переносить с одного стержня на другой. За один раз разрешается переносить только одно кольцо, причём нельзя класть большее кольцо на меньшее. Задача состоит в том, чтобы перенести пирамиду из восьми колец с одного стержня на другой за наименьшее число ходов.



### Постановка задачи:

На активном листе должно быть подсвечено игровое поле, на котором в каждый момент времени отображается текущее состояние игры. До начала игры все кольца находятся на самом левом стержне. Пользователь перемещает кольца, кликая сначала на тот стержень, с которого снимает кольцо, а потом на тот стержень, куда хочет кольцо переместить. Программа проверяет, допустим ли такой ход по правилам игры, и, если да, то отрисовывает кольцо на новом месте. Головоломка считается решенной, как только на втором или третьем стержне соберется пирамида из восьми колец. Рядом с полем необходимо указывать, какое количество ходов произведено на текущий момент. Желательно, чтобы на поле были кнопки, позволяющие отменить последние N ходов или начать игру заново.

## Lines (сложность высокая)

С помощью макроса реализовать в Excel интерфейс игры lines.

Текущий счет:	22
Сделано ходов:	7
Собрано линий	3

время на ход	1:45
--------------	------

### Правила игры:

Игра проходит на поле размером 9x9. В игре участвуют фишки семи цветов. До игры на поле расставляются три фишки произвольного цвета. За один ход игрок может передвинуть одну фишку, выделив ее и указав ее новое местоположение. Для совершения хода необходимо, чтобы между начальной и конечной клетками существовал путь из свободных клеток. Цель игры состоит в удалении максимального количества фишек, которые исчезают при выстраивании фишек одного цвета по пять и более в ряд (по горизонтали, вертикали или диагонали). При исчезновении ряда фишек новые три фишки не выставляются. В остальных случаях каждый ход выставляются новые три фишки. Игрок может видеть заранее три фишки, которые появятся в следующем ходу. Чем длиннее собранная линия, тем больше очков получает игрок. Чем больше очков у игрока, тем меньше времени ему даётся на ход. Игра заканчивается, когда поле полностью заполнится фишками.

### Постановка задачи:

На активном листе должно быть подсвечено игровое поле 9\*9, на котором в каждый момент времени отображается текущее состояние игры. До начала игры случайным образом на поле выставляются три фишки. Пользователь делает ход, выбирая фишку, которую он хочет переместить, а затем поле, куда фишка должна попасть. Если такой ход возможен, то фишка перемещается, и происходит проверка, не собралась ли линия. Если нет, то программа выставляет на поле еще 3 фишки и опять проверяет, не собралась ли линия. Количество очков, которые добавляются игроку за собранную линию, зависит от длины этой линии. В ходе игры рядом с игровым полем показывается, сколько ходов сделал игрок, сколько линий собрал и сколько очков получил на текущий момент.

При желании можно выбрать свою игру, но выбор **ОБЯЗАТЕЛЬНО** обсудить с преподавателем.

## **Составление уникальных идентификаторов** (сложность низкая)

В ячейках Excel расположены данные студентов: фамилия, имя, отчество, группа. Все данные записаны на русском языке, данные об одном учащемся занимают одну ячейку. Необходимо для каждого студента сгенерировать уникальный идентификатор (логин). При этом требуется:

- 1 разбить данные по столбцам с заголовками: фамилия, имя, отчество, группа;
2. для каждого учащегося сгенерировать уникальный идентификатор, состоящий из латинских букв, для этого первую букву его имени и фамилию записать буквами латинского алфавита;
3. проверить уникальность каждого идентификатора; в случае возникновения одинаковых идентификаторов для разных студентов, необходимо их усложнить, добавляя при необходимости номера групп, буквы отчества и т.д. При этом следует минимизировать число добавленных знаков. Например, если два В.Смирновых учатся в одной группе, то нет смысла усложнять идентификаторы номерами групп.

Сгенерированные идентификаторы должны быть устроены единообразно, т.е. если при усложнении потребовалось добавить номер группы, то он добавляется в логины всех студентов.

## **Составление графика занятости аудиторий** (сложность низкая)

В ячейках Excel записано расписание нескольких групп. Расписание можно скопировать на сайте <http://profkomvmk.ru/schedule>. По расписанию составить график занятости аудиторий и показать результат в виде таблицы.

## **Составление расписания для преподавателей** (сложность средняя)

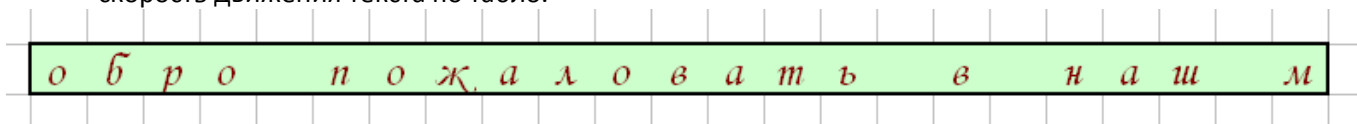
В ячейках Excel записано расписание нескольких групп. Расписание можно скопировать на сайте <http://profkomvmk.ru/schedule>. По расписанию учебных групп составить расписание преподавателей. Расписание каждого преподавателя располагается на отдельном листе файла и разделено по дням недели. По каждой паре показывается следующая информация: время начала пары, время ее окончания, аудитория и учебная группа.

## Бегущая информационная строка (сложность низкая)

С помощью макроса реализовать в Excel эмуляцию бегущей строки на информационном табло (подобная строка есть в некоторых автобусах и в вагонах метро). Эмуляция реализуется путем циклического сдвига текста строки справа налево, при этом исчезающие слова появляются через некоторое время справа.

Перед началом работы в диалоге пользователь имеет возможность задать:

- ширину табло (количество символов на нем);
- стиль и цвет шрифта;
- текст рекламного сообщения произвольной длины;
- скорость движения текста по табло.

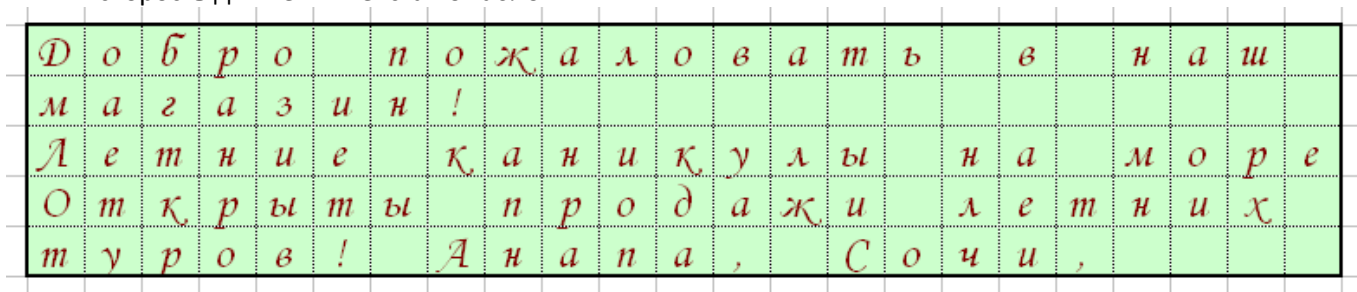


## Рекламное табло (сложность средняя)

С помощью макроса реализовать в Excel эмуляцию показа текста на рекламном табло из нескольких строк. Показ текста осуществляется путем циклического сдвига строк снизу вверх, при котором верхняя строка табло исчезает, но появляется через некоторое время снизу.

Перед началом работы в диалоге пользователь имеет возможность задать:

- ширину табло (количество символов в строке и количество строк на нем);
- стиль и цвет шрифта;
- рекламный текст произвольной длины;
- скорость движения текста по табло.

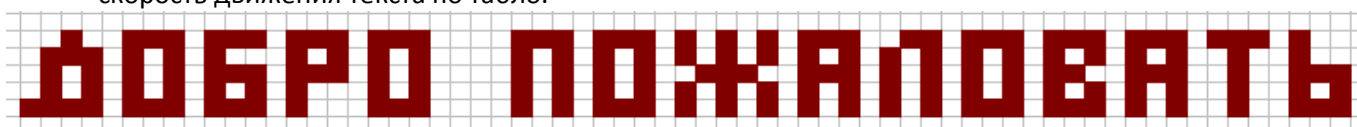


## Световая матрица (сложность высокая)

С помощью макроса реализовать в Excel эмуляцию показа текста в виде бегущей строки на информационном табло. Показ осуществляется сдвигом подсветки ячеек справа на лево, при этом исчезающий слева текст появляется через некоторое время справа. Каждая буква изображается с помощью нескольких подсвечивающихся ячеек табло, на фиксированной для нее матрице (на рисунке для букв А и О размер матрицы 5x3, а для буквы Ж 5x5).

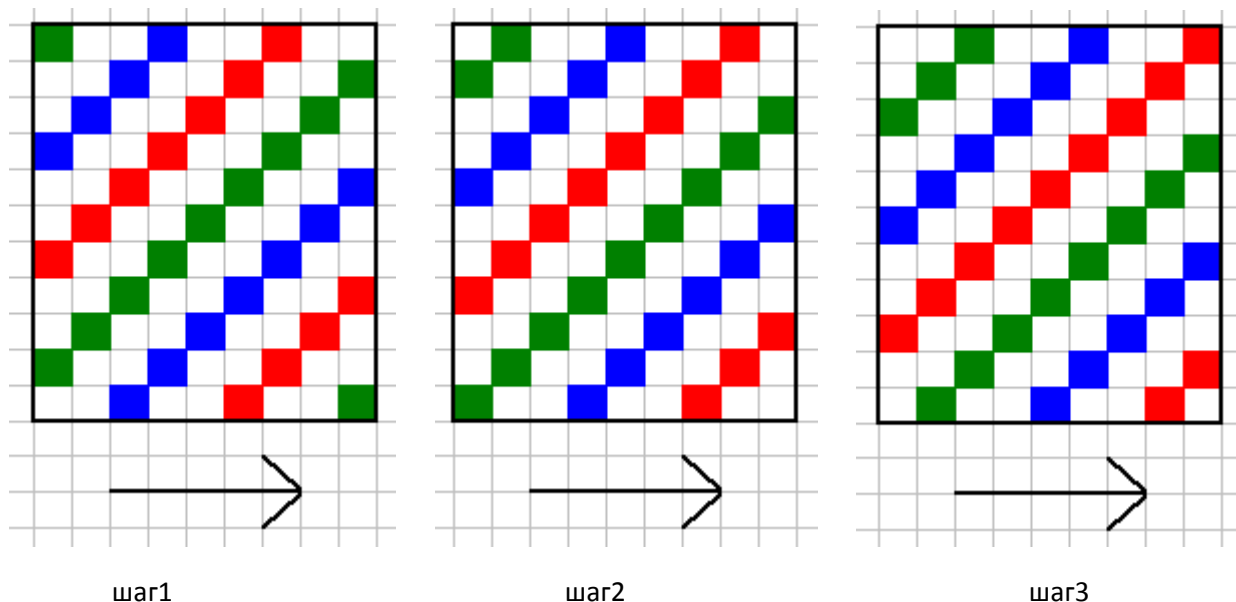
Перед началом работы в диалоге пользователь имеет возможность задать:

- ширину табло (количество ячеек в табло);
- текст информационного сообщения;
- скорость движения текста по табло.



## Иллюминация (сложность низкая)

С помощью макроса реализовать в Excel эмуляцию праздничной иллюминации. Перед началом работы в диалоге пользователь задает размер светового табло (количество ячеек по горизонтали и вертикали). После этого на поле соответствующего размера ему дается возможность с помощью стандартных средств Excel создать рисунок (изменить цвет некоторых ячеек) и выбрать направление движения рисунка (по вертикали, по горизонтали или по диагонали). По мере исчезновения рисунка с одной стороны он должен появляться с другой. На примере показано последовательное движение рисунка по табло в направлении слева направо.

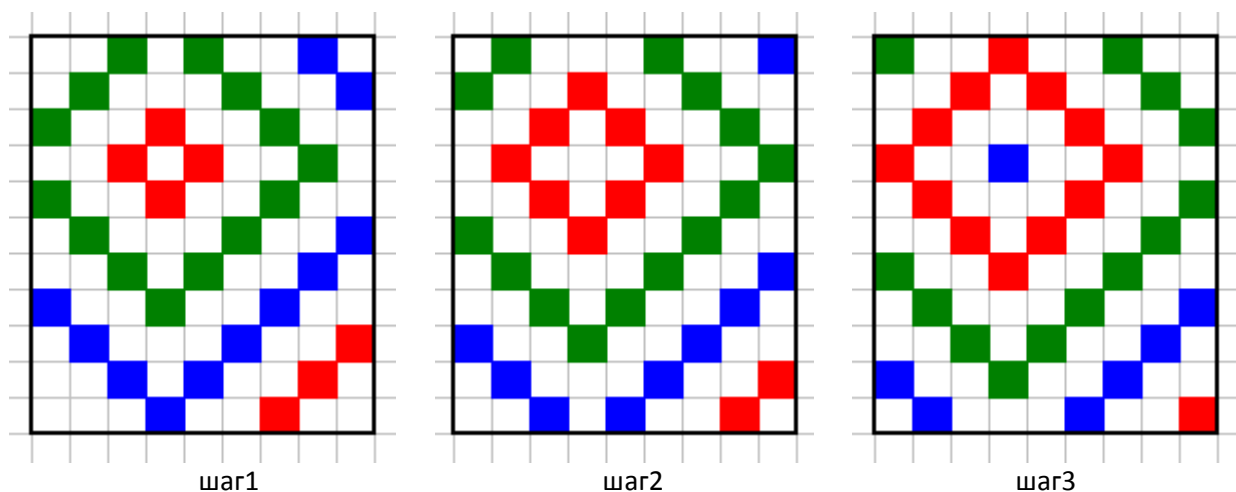


## Иллюминация в разбег (сложность высокая)

С помощью макроса реализовать в Excel эмуляцию праздничной иллюминации. Движение рисунка происходит от внутренней точки внутри светового табло к его краям. Перед началом работы пользователь выбирает:

- размер светового табло;
- тип рисунка из 5-6 предложенных на выбор (ромбы, круги, квадраты, лучи т .д.);
- количество используемых цветов;
- координату точки, в которой будет расположен центр рисунка.

На примере показано движение ромбов от точки с координатой (4,4) с использованием 3-х цветов.



## Список литературы

1. Король В.И. Visual Basic 6.0, Visual Basic for Application 6.0. Язык программирования. Справочник с примерами. – М.:КУДИЦ-ОБРАЗ, 2000
2. Гарнаев А.Ю. Самоучитель VBA. – СПб.: БХВ-Петербург, 2003
3. Эйткен, Питер. Разработка приложений на VBA в среде Office XP.: Пер. с англ. – М.:Издательский дом «Вильямс», 2003
4. Visual Basic for Applications Language Reference  
[[http://msdn.microsoft.com/en-us/library/ee441138\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/ee441138(v=office.12).aspx)]

Приложение А. Решения задач для самостоятельной работы.

### Вариант решения задания семинара 1

```
For i = 0 To 10 Step 2
    For j = 1 To 10
        k = i / 2 * 10 + j
        Cells(j, i + 1).Value = k
        Cells(j, i + 2).Interior.ColorIndex = k
        Cells(j, i + 1).Font.ColorIndex = k
        If k = 56 Then
            Exit For
        End If
    Next j
Next i
```

### Вариант решения задания семинара 2

Код макроса:

```
Sub dr()
    Randomize Second(Now)
    UserDialog.Show
End Sub
```

Код формы:

```
Private Sub CollorButton_Click()
    ActiveCell.Interior.ColorIndex = CollorLabel.Caption
    ActiveCell.Value = CollorLabel.Caption
    NextButton_Click
End Sub
```

```
Private Sub ExitButton_Click()
    End
End Sub
```

```
Private Sub NextButton_Click()
    ActiveCell.Next.Activate
    Col = Int(Rnd * 56)+1
    CollorLabel.Caption = Col
End Sub
```

```
Private Sub UserForm_Initialize()
    CollorLabel.Caption = Int(Rnd * 56) + 1
End Sub
```



### Вариант решения задания семинара 3

```
Sub col3()
Dim cc(8) As Integer 'массив для хранения количества каждой цифры в диапазоне
Dim mycoll As Collection
Set mycoll = New Collection
Dim ch As Excel.ChartObject ' создаем гистограмму и располагаем ее в позиции 250,1
Set ch = ActiveSheet.ChartObjects.Add(250, 1, 180, 140)

For i = 1 To 9
    'цикл по всем возможным числам
    For Each c In Selection.Cells
        'просматриваем выделенный диапазон и,
        'если значение ячейки равно i, то добавляем ее в коллекцию
        If c = i Then
            mycoll.Add c
        End If
    Next

    For Each c In mycoll
        'меняем фон у ячеек, собранных в коллекцию
        c.Interior.ColorIndex = mycoll.Count
    Next

    cc(i - 1) = mycoll.Count 'добавляем в массив cc количество очередной цифры
    ch.Chart.SeriesCollection.NewSeries 'добавляем в гистограмму новую категорию
    ch.Chart.SeriesCollection(i).Name = i
    ch.Chart.SeriesCollection(i).Values = mycoll.Count

    For Each c In mycoll 'очищаем коллекцию для работы со следующим числом
        mycoll.Remove 1
    Next
Next i

'располагаем легенду внизу гистограммы, убираем подписи горизонтальной оси
ch.Chart.SetElement (msoElementLegendBottom)
ch.Chart.SetElement (msoElementPrimaryCategoryAxisWithoutLabels)

'строим вторую гистограмму, в которой только одна категория
'в качестве данных даем ей массив cc, в котором собрали нужные нам значения
Dim chh As Excel.ChartObject
Set chh = ActiveSheet.ChartObjects.Add(250, 151, 180, 100)
chh.Chart.SeriesCollection.NewSeries
chh.Chart.SeriesCollection(1).Values = cc
chh.Chart.HasLegend = False

End Sub
```

#### Вариант решения задания семинара 4

```
Private Sub Workbook_SheetBeforeDoubleClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)
If ActiveSheet.Cells(1, 1).Interior.ColorIndex = -4142 Then
    ActiveSheet.Cells(1, 1).Interior.ColorIndex = 1
Else
    ActiveSheet.Cells(1, 1).Interior.ColorIndex = (ActiveSheet.Cells(1, 1).Interior.ColorIndex + 1) Mod 57
End If
Cancel = True 'отменяем стандартные действия двойного клика
End Sub
```

#### Вариант решения задания семинара 5

```
Sub symb()
StartForm.Show
End Sub

Sub Search()
Dim coll As Collection
Set coll = New Collection
For b = 1 To Len(StartForm.SymbLabel.Text)
    Selection.Start = ActiveDocument.Paragraphs(b).Range.Start
    Selection.End = ActiveDocument.Range.End
    For Each w In Selection.Words
        If w.Characters(1) = Mid(StartForm.SymbLabel.Text, b, 1) Then
            coll.Add w
            w.Font.Bold = True
        End If
    Next w
    Selection.Move wdParagraph, -1 'перемещаем курсор на один абзац назад
    Dim i As Integer
    i = 1
    For Each w In coll
        Selection.TypeText i & ":" & w & " " 'печатаем слова по заданному шаблону
        i = i + 1
    Next
    Selection.InsertParagraph 'добавляем символ конца абзаца
    For Each w In coll
        coll.Remove (1)
    Next
Next
End Sub
```

## Вариант решения задания семинара 6

```
Sub ZamenaK()  
Selection.Start = ActiveDocument.Paragraphs(1).Range.Start  
Selection.End = ActiveDocument.Paragraphs(3).Range.End  
  
Selection.Find.MatchWildcards = True  
Selection.Find.Text = "<B*>"  
  
Selection.Find.Replacement.Text = ""  
Selection.Find.Replacement.Font.Bold = True  
Selection.Find.Replacement.Font.ColorIndex = 6  
Selection.Find.Execute Replace:=wdReplaceAll  
End Sub
```