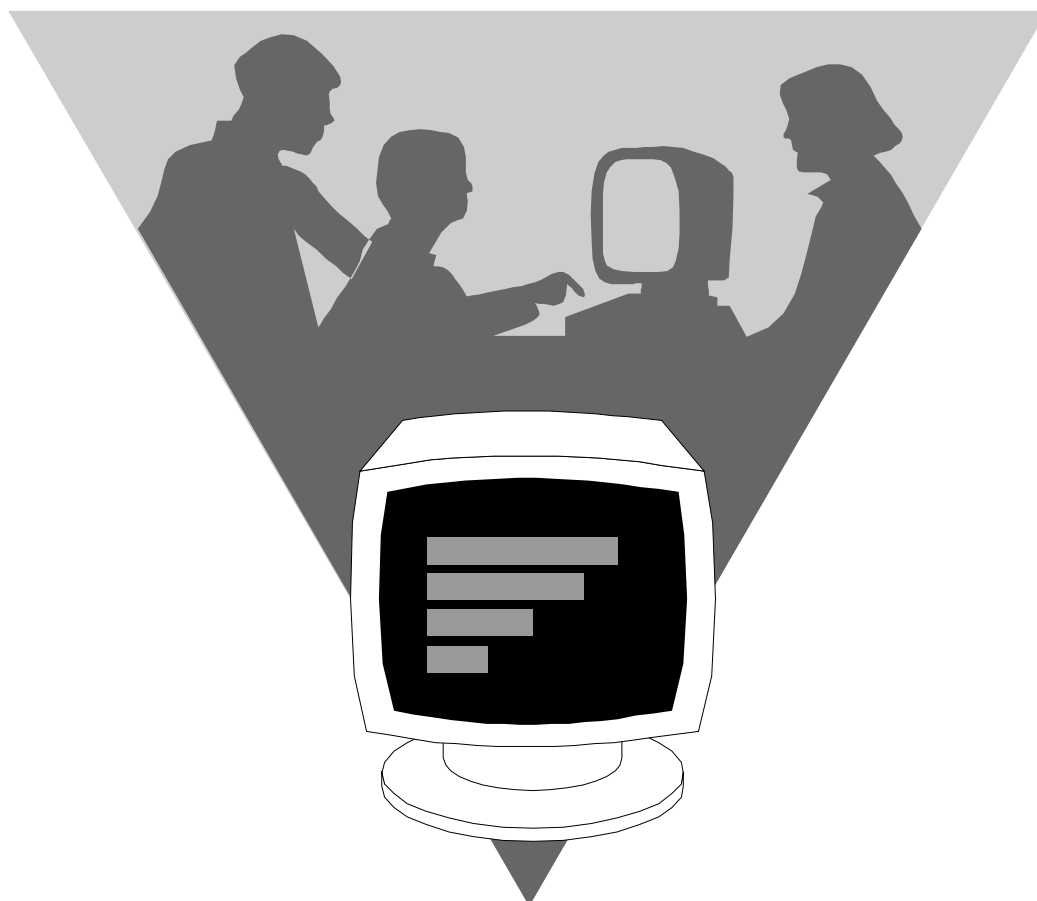


**Министерство образования и науки РФ
Московский государственный университет геодезии и картографии**

«Программирование в среде MS EXCEL»

**Учебно-методическое пособие
по курсу
«Технология программирования»**



Москва 2016 г.

Составитель: Кудлаев А.А.

Программирование в среде MS EXCEL. Учебно-методическое пособие по курсу «Технология программирования». М., МИИГАиК, 2016, 35 стр.

Учебно-методическое пособие составлено в соответствии с утвержденной программой курса «Технология программирования» для студентов факультета прикладной космонавтики, рекомендовано кафедрой вычислительной техники и автоматизированной обработки аэрокосмической информации.

Учебно-методическое пособие содержит примеры и практические задания по основам языка VBA и его практическому применению в среде MS EXCEL.

Библиография: 8 названий.

Рецензенты:

Проф. кафедры ВТиАОАИ МИИГАиК Гаврилова В.В.

Член-кор. Академии информатизации, с.н.с. НТИЦ МЭИ Макальский Л.М.

ОГЛАВЛЕНИЕ:

Задание №1. Функции пользователя	01
Задание №2. Процедуры	06
Задание №3. Ветвления	07
Объекты, свойства и методы VBA	10
Задание №4. Циклы	11
Задание №5. Стандартные функции для работы с массивами	17
Задание №6. Решение системы линейных уравнений	22
Задание №7. Решение оптимизационных задач	24
Задание №8. Метод спирального координатного спуска при решении задач оптимизации	29
Задание №9. Формы	32
Задание №10. Создание сценариев	33
Задание №11. Диалоговые окна	35
Описание программы	37
Литература	37

Задание №1

Функции пользователя

Простейшей задачей, решаемой на VBA, является создание функций пользователя, имеющих имена. Общий вид функции пользователя:

```
Function имя_функции(список_параметров)
    тело_функции
End Function
```

Создадим функцию нахождения угла треугольника по его сторонам:

$$A = \arccos((b^2 + c^2 - a^2) / 2bc)$$

Функции пользователя создаются в модуле, после чего с ними можно работать с помощью мастера функций.

Для создания модуля VBA необходимо:

- Установить на панели инструментов Visual Basic;
- вызвать редактор Visual Basic;
- в листе модуля ввести текст программы:

```
Function AngleA(a, b, c)
    AngleA = Application.ACOS((b ^ 2 + c ^ 2 - a ^ 2) / (2 * b * c))
End Function
```

Поскольку функция ACOS() не является внутренней функцией VBA, то ее записывают в форме Application.ACOS().

По умолчанию созданная функция при ее выборе с помощью мастера функций относится к категории **Определенные пользователем**. Можно изменить категорию принадлежности функции, а также добавить необходимые комментарии в мастере функций.

Для этого необходимо выполнить следующие действия:

- при активном листе модуля, на котором введена функция, нажать клавишу **<F2>**, на экране появится окно **Просмотр объектов**;
- в окне **Классы** окна **Просмотр объектов** необходимо выбрать рассматриваемый **Модуль**, при этом активизируются окно **Компоненты модуля**;
- для активного окна **Компоненты модуля** нажать правую кнопку мыши и выбрать **Свойства**;
- в **Свойствах** можно задать описание заданной функции.

Геометрические расчеты, векторные операции и преобразования координат

Самостоятельно: Для заданного примера из списка задач создайте module VBA под именем «Задание01». В этом модуле запрограммируйте функции, отладьте их для указанных входных данных и результаты представьте на рабочем листе под именем «Задание01».

1. Решение треугольника по 3 сторонам a , b и c :

$$A = \arccos((b^2 + c^2 - a^2) / 2bc), \quad B = \arccos((a^2 + c^2 - b^2) / 2ac),$$

$$C = \pi - (A + B), \quad S = (ab \sin C) / 2.$$

Для $a = 4$, $b = 2$, $c = 5$ получим

$$S = 3,80, \quad A = 49,46^\circ, \quad B = 22,33^\circ, \quad C = 108,21^\circ.$$

2. Решение треугольника по двум сторонам a, b и углу C между ними:

$$c = \sqrt{a^2 + b^2 - 2ab \cos C}, \quad A = \arccos((b^2 + c^2 - a^2)/2bc), \quad B = \pi - (A + C),$$

$$S = (ac \sin B)/2.$$

Для $a = 4$, $b = 2$, $C = 108,21^\circ$ получим

$$S = 3,80, \quad A = 49,46^\circ, \quad B = 22,33^\circ, \quad c = 5,0$$

3. Решение треугольника по двум сторонам c, b и углу B:

$$C = \arcsin (c \sin B/b), \quad A = \pi - (B + C), \quad a = b \sin A/\sin B, \quad S = (ac \sin B)/2.$$

При $b < c$ существует и второе решение:

$$C' = \pi - C, \quad A' = \pi - (B + C'), \quad a' = (b \sin A')/\sin B, \quad S' = (a'c \sin B)/2.$$

Для $b = 2$, $c = 2,5$ и $B = \pi/4$ рад получим $S = 2,39$, $a = 2,70$, $A = 1,27$ рад,
 $C = 1,08$ рад, $S' = 0,74$, $a' = 0,83$, $A' = 0,30$ рад, $C' = 2,06$ рад.

4. Решение треугольника по двум углам B, C и стороне a между ними:

$$A = \pi - (B + C), \quad b = a \sin B/\sin A, \quad c = a \sin C/\sin A, \quad S = ac \sin B/2.$$

Для $a = 50$, $B = 1$ рад и $C = 1,5$ рад получим $S = 1753,14$, $A = 0,64$ рад,
 $b = 70,30$ и $c = 83,34$.

5. Решение треугольника по двум углам A и C и стороне против одного из них:

$$B = \pi - (A + C), \quad b = (a \sin B)/\sin A, \quad c = (a \sin C)/\sin A, \quad S = (bc \sin A)/2.$$

Для $a = 50$, $A = 1$ рад и $C = 1,5$ рад получим $S = 886,80$, $B = 0,64$ рад,
 $b = 35,56$ и $c = 59,27$.

6. Вычисление ориентированного расстояния от точки (x_0, y_0) до прямой $y = a + bx$ на плоскости.

$$d = (a + bx_0 - y_0)/\text{SQRT}(b^2 + 1)$$

Для $a = 2$, $b = 3$ и $x_0 = 4$, $y_0 = 5$ получим $d = 2,85$.

7. Вычисление ориентированного расстояния от точки (x_0, y_0, z_0) в пространстве до плоскости $Ax + By + Cz + D = 0$:

$$d = (Ax_0 + By_0 + Cz_0 + D)/\sqrt{A^2 + B^2 + C^2}$$

Для $A = 2$, $B = 2$, $C = 2$, $D = 1$ получим $d = 8,95$ при $z_0 = 6$, $y_0 = 5$, $x_0 = 4$.

8. Расстояние между двумя точками (x_1, y_1) и (x_2, y_2) , находящимися на одной плоскости:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

При $x_1 = -2,3$, $y_1 = 4$, $x_2 = 8,5$ и $y_2 = 0,7$ получим $d = 11,29$.

9. Перерасчет координат x и y точки из одной системы декартовых координат в другую с центром в точке (x_0, y_0) и углом поворота α :

$$x' = (x - x_0) \cos \alpha + (y - y_0) \sin \alpha,$$

$$y' = (x - x_0) \sin \alpha + (y - y_0) \cos \alpha.$$

Для $x_0 = 30$, $y_0 = 40$, $\alpha = 0,5$ рад и $x = 50$, $y = 50$ получим $x' = 22,34$ и $y' = 18,36$.

Visual Basic for Applications

10. Вычисление координат (x, y) точки, делящей отрезок с концевыми точками A(x₁, y₁) и B(x₂, y₂) на плоскости в отношении m₁/m₂:

$$x = (m_2x_1 + m_1x_2)/(m_1 + m_2) \qquad y = (m_2y_1 + m_1y_2)/(m_1 + m_2)$$

Для $m_1 = 2$, $m_2 = 3$, $y_1 = -4$, $x_1 = 6$, $y_2 = 0$ и $x_2 = 0$ получим $x = 3,6$ и $y = -2,4$.

11. Вычисление длины отрезка:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

заданного точками или векторами $A_1(x_1, y_1, z_1)$ и $A_2(x_2, y_2, z_2)$, выходящими из центра декартовой системы координат;

вычисление координат точки, делящей этот отрезок в отношении m_1/m_2 ,

$$x = (m_2x_1 + m_1x_2)/(m_1 + m_2), y = (m_2y_1 + m_1y_2)/(m_1 + m_2), z = (m_2z_1 + m_1z_2)/(m_1 + m_2)$$

и вычисление угла между векторами:

$$\varphi = \arccos((x_1x_2 + y_1y_2 + z_1z_2) / (\sqrt{x_1^2 + y_1^2 + z_1^2} * \sqrt{x_2^2 + y_2^2 + z_2^2}))$$

Для $m_1 = 2$, $m_2 = 3$, $A_1 = (2; 4; -1)$ и $A_2 = (-3; -1; 6)$ получим $d = 9,95$, $x = 0$, $y = 2$, $z = 1,8$ и $\varphi = 2,1115608$ рад.

12. Вычисление длины вектора, заданного координатами конца (x, y, z),

$$d = \sqrt{x^2 + y^2 + z^2},$$

и вычисление углов между заданным вектором и осями декартовой системы координат: $A = \arccos(x/d)$, $B = \arccos(y/d)$ и $C = \arccos(z/d)$.

Для $x = 2$, $y = -2$, $z = -1$ получим $d = 3$, $A = 48,19^\circ$, $B = 131,81^\circ$ и $C = 109,47^\circ$.

13. Преобразование декартовых координат точки (x, y) на плоскости в полярные (с длиной радиуса-вектора A и углом φ):

$$A = \sqrt{x^2 + y^2}, \quad \varphi = \begin{cases} \arccos(x/A) \text{ при } y \geq 0 \\ -\arccos(x/A) \text{ при } y < 0 \end{cases}$$

Для $x = y = 1$ получим $A = 1,41$ и $\varphi = 45,0^\circ$.

14. Преобразование полярных координат (A, φ) точки в декартовы (x, y):

$$x = A \cos \varphi \quad \text{и} \quad y = A \sin \varphi.$$

Для $A = \sqrt{2}$ и $\varphi = 45^\circ$ получим $x = y = 1,0$.

15. Вычисление полярных координаты прямой Ax + By + C = 0, лежащей на плоскости:

$$P = C/\sqrt{A^2 + B^2}, \quad \cos \alpha = \pm A/\sqrt{A^2 + B^2}, \quad \sin \alpha = \pm B/\sqrt{A^2 + B^2},$$

верхний знак соответствует $D < 0$, а нижний $-D \geq 0$.

Для прямой $3x - 4y + 10 = 0$ получим $P = 2$, $\cos \alpha = -0,6$, $\sin \alpha = 0,8$,

для прямой $3x - 4y - 10 = 0$ получим $P = 2$, $\cos \alpha = 0,6$ и $\sin \alpha = -0,8$.

Задание №2**Процедуры**

Процедура - это совокупность операторов VBA, выполняющих определенное действие. Процедуры имеют стандартное оформление:

```
Sub имя_процедуры(список_параметров)
    тело_процедуры
End Sub
```

Кроме выполнения определенных действий процедура может возвращать значения, которые присваиваются параметрам внутри процедуры. Процедуры нельзя вызывать из ячеек рабочего листа.

Программа VBA состоит из одного или нескольких модулей. Модуль - это лист с текстом программы, вставленный в рабочую книгу. Обычно текст программы начинается с опций, которые управляют описанием переменных и т.д. Затем следует объявление глобальных переменных или констант для данного модуля. Далее располагается текст функций пользователя и процедур, составляющих программу.

Пример организации модуля:

```
Option Explicit
Function AngleA(a, b, c)
    AngleA = Application.Acos((b ^ 2 + c ^ 2 - a ^ 2) / (2 * b * c))
End Function
```

```
Function AngleB(a, b, c)
    AngleB = Application.Acos((a ^ 2 + c ^ 2 - b ^ 2) / (2 * a * c))
End Function
```

```
Function AngleC(a, b, c)
    AngleC = Application.Pi() - (AngleA(a, b, c) + AngleB(a, b, c))
End Function
```

```
Sub Rezult()
    Dim a, b, c, AC, S As Single
    a = 4: b = 2: c = 5
    MsgBox ("AngleA= " & AngleA(a, b, c))
    MsgBox ("AngleB= " & AngleB(a, b, c))
    MsgBox ("AngleC= " & AngleC(a, b, c))
    AC = AngleC(a, b, c)
    S = a * b * Sin(AC) / 2
    MsgBox ("S= " & CStr(S))
End Sub
```

Инструкция Option Explicit указывает на необходимость описания типов всех переменных, используемых на данном листе модуля. Функция CStr переводит числовой формат в строковый. Процедура MsgBox выводит текстовую информацию в окно вывода.

Самостоятельно: для полученного примера из работы №1 создайте module под именем «Задание02». В этом модуле используйте созданные ранее вами функции, осуществите их вызов из процедуры, входные данные задайте в теле процедуры, вывод результатов выполните с помощью процедуры MsgBox.

Задание №3

Функции пользователя с операторами условного перехода

Рассмотрим пример функции пользователя

```
Function G(y)
    If y<=0 Then G=Y^2 Else G=Y^3
End Function
```

вычисляющей следующую функцию с двумя условиями:

$$g = \begin{cases} y^2, & y \leq 0, \\ y^3, & y > 0. \end{cases}$$

В функции **G** использовался условный оператор **If-Then-Else**
Синтаксис:

```
If УСЛОВИЕ Then ОПЕРАТОР_1 Else ОПЕРАТОР_2
```

В операторе условия **ОПЕРАТОР_1** выполняется, если **УСЛОВИЕ ИСТИННО**, в противном случае выполняется **ОПЕРАТОР_2**.

УСЛОВИЕ - это выражение логического типа. Выражение может быть простым и сложным. При записи простых условий используются операции отношения: {=, <>, >, <, >=, <=}.

Сложные условия образуются из простых путем применения логических операций {Not, And, Or} и круглых скобок.

В условном операторе допустимо использование блока операторов вместо любого из операторов:

```
If УСЛОВИЕ Then БЛОК_ОПЕРАТОРОВ_1 Else БЛОК_ОПЕРАТОРОВ_2 End If
```

Ветвь **Else** в условном операторе является необязательной. В операторе условия после **Then** можно разместить несколько операторов. В этом случае они должны располагаться в одну строчку и быть разделены двоеточием. В условном операторе может проверяться несколько условий.

Функции пользователя с несколькими операторами условного перехода

Рассмотрим пример функции пользователя, вычисляющей значения следующей функции с тремя условиями:

$$z = \begin{cases} 1 + x, & x \leq -1 \\ 2 \ln(1 + x), & x \in (-1, 0) \\ (1 + x)^2, & x \geq 0 \end{cases}$$

Функцию **z** можно вычислить с помощью следующей функции пользователя:

```
Function z(x)
    If x<=-1 Then z=1+x
    If x>-1 And x<0 Then z=2*Application.Ln(1+x)
    If x>=0 Then z=(1+x)^2
End Function
```

Функции пользователя с оператором выбора Select Case

Оператором выбора удобно пользоваться, когда в зависимости от значения некоторого выражения, имеющего конечное множество допустимых значений, необходимо выполнить разные действия.

Синтаксис:

Select Case ТЕСТИРУЕМОЕ_ВЫРАЖЕНИЕ

Case УСЛОВИЕ_ВЫБОРА_1
БЛОК_ОПЕРАТОРОВ_1

... ..

Case УСЛОВИЕ_ВЫБОРА_n
БЛОК_ОПЕРАТОРОВ_n

Case Else
БЛОК_ОПЕРАТОРОВ

End Select

После каждого оператора Case может находиться произвольное количество других операторов, и все они будут выполняться, если условие оператора Case истинно.

Пример вычисления комиссионных на основе оператора выбора Select Case:

	А	В	С
1	Объем продаж	Премия	
2	1000	=Premia(A2)	
3	20000	=Premia(A3)	
4	30000	=Premia(A4)	
5	33000	=Premia(A5)	
6	45000	=Premia(A6)	
7	73000	=Premia(A7)	

Function Premia (prodano)

Select Case prodano

Case 0 To 9999

Premia = 0.08*prodano

Case 10000 To 39999

Premia = 0.1*prodano

Case Is > = 40000

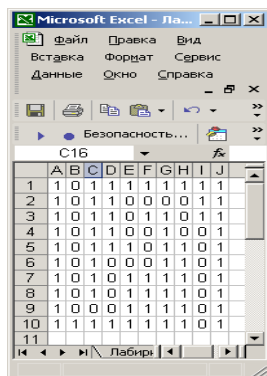
Premia = 0.14*prodano

End Select

End Function

Самостоятельно: создайте module под именем «Задание03», в этом модуле запрограммируйте функции с операторами выбора, рассмотренные выше, отладьте их для указанных входных данных, результаты представьте на рабочем листе под именем «Задание03».

Алгоритм трассировки



```
Option Base 1
Sub LABIRINT1()
Dim R, L, U, D, F As Labels
```

```
Dim lab(10, 10) As Integer
```

```
For i = 1 To 10
  For j = 1 To 10
    lab(i, j) = Cells(i, j)
  Next j
Next i
```

```

i = 1: j = 2: lab(i, j) = 8          ' Точка старта на C3
R:
If lab(i, j + 1) <> 0 Then GoTo D   ' Проверка справа
j = j + 1: lab(i, j) = 8           ' Шаг вправо
If (j - 10) >= 0 Then GoTo F       ' Проверка правой границы
GoTo R
D:
If lab(i + 1, j) <> 0 Then GoTo L   ' Проверка вниз
i = i + 1: lab(i, j) = 8           ' Шаг вниз
If (i - 10) >= 0 Then GoTo F       ' Проверка нижней границы
GoTo R
L:
If lab(i, j - 1) <> 0 Then GoTo U   ' Проверка влево
j = j - 1: lab(i, j) = 8           ' Шаг влево
If (j) <= 1 Then GoTo F            ' Проверка левой границы
GoTo R
U:
If lab(i - 1, j) <> 0 Then GoTo F   ' Проверка вверх
i = i - 1: lab(i, j) = 8           ' Шаг вверх
If i <= 1 Then GoTo F              ' Проверка верхней границы
GoTo R
F:
For i = 1 To 10
  For j = 1 To 10
    Cells(i, j) = lab(i, j)
    If lab(i, j) = 8 Then
      Cells(i, j).Font.ColorIndex = 3
    End If
  Next j
Next i
End Sub
```

Самостоятельно: в module «Задание03» запрограммируйте алгоритм трассировки. Отладьте программу. Результаты представьте на рабочем листе с именем «Трассировка». Модифицируйте алгоритм:

1. исключите использование массива lab(i, j);
2. обеспечьте «нечувствительность» алгоритма к коду трассы;
3. для визуализации работы алгоритма трассировки обеспечьте приемлемую «задержку».

Объекты, свойства и методы VBA

Объект - это то, чем управляют с помощью программы на VBA: диалог, рабочий лист, диапазон ячеек и др. Каждый объект обладает *свойствами*. Например, диалог может быть видимым и невидимым (свойство **Visible**). Объект содержит также список методов, которые к нему применимы. Методы - это то, что можно сделать с объектом (показать или убрать диалог можно с помощью методов **Show** и **Hide**). Объект является элементарной единицей построения программ VBA.

В VBA имеется много встроенных объектов:

Range	Диапазон ячеек
Cells	Ячейка
Sheet	Лист
WorkSheet	Рабочий лист
DialogSheet	Диалоговое окно

Большинство объектов принадлежит к группе подобных объектов - наборам: все рабочие листы рабочей книги образуют набор - **Worksheets**. Наборы используются одним из двух способов: либо действие совершается над всеми объектами набора, либо со ссылкой на набор выбирается конкретный объект для работы с ним. Наборы:

Sheets	Листы
DialogSheets	Диалоговые окна

Изменяя свойства, можно изменять характеристики объекта. Установка значений свойств - это один из способов управления объектами. Для установки свойства вводят имя объекта, затем точку и за ней - имя свойства, далее знак равенства и значение свойства:

Объект.свойство=Выражение

Range("Данные").Value=0.1 для свойства **Value** диапазона ячеек **Данные** устанавливается значение 0.1

Range("A2").Formula="СУММ(A1:C1)" в ячейку **A2** вставляется формула путем изменения свойства **Formula**.

Некоторые свойства являются неизменяемыми, т.е. свойства можно узнать, но нельзя изменить. Синтаксис чтения свойств выглядит следующим образом:

Переменная = Объект.Свойство

Процентная_ставка=Range("A1").Value	переменной Процентная ставка
Процентная_ставка=Cells(1,1).Value	присваивается значение из ячейки A1 текущего рабочего листа

Синтаксис вызова объекта: **Объект.Метод**

Range("Данные").Clear метод **Clear** позволяет очистить содержимое диапазона ячеек

Range("A10:B12").Select выбирает диапазон ячеек

В EXCEL некоторые объекты содержат другие объекты, например, рабочая книга - рабочие листы - диапазон ячеек и т.д. Объектом самого высокого уровня является **Application** (приложение). Точка после имени объекта указывает на то, что далее следует имя свойства или метода. Но после точки можно указать и имя объекта для перехода от одного объекта к другому. Ссылки на объекты могут быть слишком громоздкими. Если объект часто используется в программе, то создают объектную переменную с помощью команды **Set**:

Dim R As Object

Set R=Application.WorkBooks("Отчет").Sheets("Май").Rows(2).Cells(1)

R.Value="Да"

Задание №4

Оператор цикла FOR - NEXT

Для многократного выполнения одного оператора или их группы служит оператор цикла FOR - NEXT. Цикл FOR - NEXT обеспечивает многократное выполнение блока операторов при последовательном изменении счетчика от начального до конечного значения с указанным шагом изменения. По оператору Exit For можно выйти из оператора цикла досрочно.

Рассмотрим пример использования оператора цикла. Нужно вычислить:

$$s = \frac{2 \sum_{i=1}^n x_i + \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} c_{ij} \right)^2}{1 + \sum_{i=1}^n x_i^2},$$

где x - вектор из n компонент, b и c - матрицы размерности $m \times m$, причем $n=3$, $m=2$ и

$$x = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}, \quad b = \begin{pmatrix} 2 & 1 \\ 2 & 5 \end{pmatrix}, \quad c = \begin{pmatrix} 1 & 1 \\ 4 & 5 \end{pmatrix}$$

На рабочем листе «Задание04» введите компоненты вектора x и компоненты матриц b и c . Найдите значение S .

Первый способ:

Option Base 1

Function S_one(x, b, c As Variant) As Double

Dim sx, sx2, sbc As Double, i, j, n, m As Integer

n = x.Rows.Count

m = b.Rows.Count

sx = 0

For i = 1 To n

sx = sx + x(i)

Next i

sbc = 0

For i = 1 To m

For j = 1 To m

sbc = sbc + b(i, j) * c(i, j)

Next j

Next i

sxx = 0

For i = 1 To n

sxx = sxx + x(i) ^ 2

Next i

S_one = (2 * sx + sbc ^ 2) / (1 + sxx)

End Function

Второй способ:

Function S_two(x, b, c As Variant) As Double

Dim sx, sxx, sbc As Double

sx = Application.Sum(x)

sbc = Application.SumProduct(b, c)

sxx = Application.SumSq(x)

S_two = (2 * sx + sbc ^ 2) / (1 + sxx)

End Function

Самостоятельно:**Вариант 1**

$$s = \frac{2 \sum_{i=1}^n x_i y_i + \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right)^2}{3 + \sum_{i=1}^n x_i},$$

где x, y - вектора с n компонентами, b - матрица размерности $m \times m$, причем $n = 4, m=2$,

$$x = (3, 1, 2, 3), \quad y = (1, 7, 2, 3), \quad b = \begin{pmatrix} 4 & 1 \\ 2 & 5 \end{pmatrix}.$$

Вариант 2

$$s = \frac{2 \sum_{i=1}^m a_i + \left(\sum_{i=1}^n \sum_{j=1}^n c_{ij} \right)^2}{\left(1 + \sum_{i=1}^m a_i \right) \left(1 + \sum_{i=1}^m a_i^2 \right)},$$

где a - вектора из m компонентов, c - матрица размерности $n \times n$, причем $n = 3, m=4$,

$$a = (3, 1, 2, 3), \quad c = \begin{pmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{pmatrix}.$$

Вариант 3

$$s = \frac{\sum_{i=1}^n x_i + 2 \sum_{i=1}^n y_i^2 + 5 \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right)^3}{3 + \sum_{i=1}^n y_i},$$

где x, y - вектора из n компонентов, b - матрица размерности $m \times m$, причем $n = 4, m=2$,

$$x = (1, 2, 7, 4), \quad y = (1, 7, 2, 3), \quad b = \begin{pmatrix} 4 & 1 \\ 2 & 5 \end{pmatrix}.$$

Вариант 4:

$$s = 3 \sum_{i=1}^m a_i^2 + 7 \sum_{i=1}^n \sum_{j=1}^n c_{ij} - \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij} \right)^2.$$

где a - вектора из m компонентов, c - матрица размерности $n \times n$, причем $n = 3, m=4$,

$$a = (3, 1, 2, 3), \quad c = \begin{pmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{pmatrix}.$$

Вариант 5

$$s = \sum_{i=1}^n x_i + 2 \sum_{i=1}^n y_i^2 + \left(\sum_{i=1}^m \sum_{j=1}^m b_{ij} \right) \left(2 + \sum_{i=1}^n x_i \right) - 2 \left(1 + \sum_{i=1}^n x_i y_i \right),$$

где x, y - вектора из n компонентов, b - матрица размерности $m \times m$, причем $n = 4, m=2$,

$$x = (1, 2, 7, 4), \quad y = (1, 7, 2, 3), \quad b = \begin{pmatrix} 4 & 1 \\ 2 & 5 \end{pmatrix}.$$

Visual Basic for Applications

Вариант 6

$$s = \left(\sum_{i=1}^m a_i \right)^2 + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2 - \left(3 + \sum_{i=1}^n \sum_{j=1}^n c_{ij} \right) \left(1 + \sum_{i=1}^m a_i^2 \right),$$

где a - вектора из m компонентов, c - матрица размерности $n \times n$, причем $n = 3$, $m=4$,

$$a = (3, 1, 2, 3), \quad c = \begin{pmatrix} 2 & 2 & 4 \\ 2 & 4 & 6 \\ 2 & 5 & 3 \end{pmatrix}.$$

Вариант 7

$$s = \left(2 \sum_{i=1}^n x_i + \sum_{i=1}^n x_i y_i \right) \left(2 - \sum_{i=1}^n x_i \right) + 3 + \sum_{i=1}^n x_i^2,$$

где x , y - вектора из n компонентов, причем $n = 4$, $m=2$,

$$x = (1, 2, 7, 4), \quad y = (1, 7, 2, 3).$$

Вариант 8

$$s = \left(1 + \sum_{i=1}^m a_i \right)^2 \left(1 + \sum_{i=1}^m a_i^2 \right) - \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij} \right) \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2 \right),$$

где a - вектора из m компонентов, c - матрица размерности $n \times n$, причем $n = 2$, $m=4$,

$$a = (1, 4, 1, 3), \quad c = \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix}.$$

Вариант 9

$$s = \left(\sum_{i=1}^n x_i^2 + 5 \sum_{i=1}^n x_i y_i \right) \left(1 + \sum_{i=1}^n x_i + \sum_{i=1}^n y_i \right) - 3 + \sum_{i=1}^n x_i^2,$$

где x , y - вектора из n компонентов, причем $n = 4$,

$$x = (7, 5, 7, 4), \quad y = (2, 4, 2, 3).$$

Вариант 10

$$s = \left(1 + \sum_{i=1}^m a_i \right)^2 \left(1 + \sum_{i=1}^n \sum_{j=1}^n c_{ij}^2 \right) - 1 - \sum_{i=1}^m a_i^2 + 4 \sum_{i=1}^n \sum_{j=1}^n c_{ij},$$

где a - вектора из m компонентов, c - матрица размерности $n \times n$, причем $n = 3$, $m=4$,

$$a = (2, 1, 1, 3), \quad c = \begin{pmatrix} 5 & 2 & 4 \\ 1 & 4 & 2 \\ 7 & 3 & 5 \end{pmatrix}.$$

Самостоятельно: для заданного примера создайте module под именем «Задание04», в этом модуле запрограммируйте функцию, отладьте ее для указанных входных данных и результаты представьте на рабочем листе с именем «Задание04». Задание выполните двумя способами.

Оператор цикла For-Each-Next

Используется для перебора объектов из группы подобных объектов (ячеек из диапазона или элементов массива):

For Each Элемент In Группа
БЛОК ОПЕРАТОРОВ

Next Элемент

Например, процедура ZNAK меняет все положительные числа диапазона ячеек A1:B2 знаком «+», все отрицательные числа – знаком «-», а нули оставляет без изменения:

Option Explicit

Sub Znak()

Dim c As Object

For Each c In Worksheets("Лист1").Range("A1:B2")

If IsNumeric(c.Value) Then

If c.Value > 0 Then c.Value = "+"

End If

If IsNumeric(c.Value) Then

If c.Value < 0 Then c.Value = "-"

End If

If IsNumeric(c.Value) Then

If c.Value = 0 Then c.Value = 0

End If

Next c

End Sub

Функция IsNumeric(выражение) возвращает True, если выражение – числовое, и False – в противном случае.

Процедура CVET в зависимости от содержимого ячейки выделенной области (возвращается методом Selection) изменяет цвет ее фона, если содержимое ячейки положительно, то изменяется цвет, размер и тип шрифта.

Sub Cvet()

Dim a As Object

For Each a In Selection

If IsNumeric(a.Value) Then

If a.Value > 0 Then

a.Interior.ColorIndex = 8

a.Font.Bold = True

a.Font.ColorIndex = 5

a.Font.Size = 20

End If

End If

If IsNumeric(a.Value) Then

If a.Value < 0 Then a.Interior.ColorIndex = 4

End If

If IsNumeric(a.Value) Then

If a.Value = 0 Then a.Interior.ColorIndex = 6

End If

Next a

End Sub

Font – объект. Этот объект имеет следующие свойства:

Size Размер шрифта

Bold Жирный шрифт

Italic Курсивный шрифт

ColorIndex Цвет символов (1-56)

UnderLine Подчеркнутый шрифт: (xlNone - отсутствие подчеркивания; xlSingle - одинарное подчеркивание; xlDouble - двойное подчеркивание)

Interior объект, характеризующий фон

Visual Basic for Applications

Оператор цикла While – Wend

Оператор цикла While–Wend используется для организации цикла с неизвестным заранее числом шагов. Цикл While–Wend обеспечивает многократное выполнение блока операторов пока УСЛОВИЕ принимает значение True.

Синтаксис:

```
While УСЛОВИЕ
    БЛОК_ОПЕРАТОРОВ
Wend
```

Следующая процедура вычисляет значение числа π с задаваемой с клавиатуры точностью. Вычисление значения числа π основано на том, что сумма ряда $1 - 1/3 + 1/5 - 1/7 + 1/9...$ приближается к значению $\pi/4$ при достаточно большом количестве членов ряда.

Option Base 1

Option Explicit

Sub VP()

```
    Rem Pi/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9...
```

```
    Dim Pi, elem, t As Double
```

```
    Dim n As Integer
```

```
    Pi = 0: n = 1: elem = 1
```

```
    t = InputBox("Задайте точность вычислений")
```

```
    While elem >= t
```

```
        elem = 1 / (2 * n - 1)
```

```
        If (n Mod 2) = 0 Then Pi = Pi - elem Else Pi = Pi + elem
```

```
        n = n + 1
```

```
    Wend
```

```
    Pi = Pi * 4
```

```
    MsgBox ("Значение ПИ равно " & CStr(Pi))
```

```
    MsgBox ("Просуммировано " & CStr(n) & " членов ряда")
```

End Sub

В VBA для организации циклов с неизвестным заранее числом шагов используются и другие операторы цикла: Do While-Loop, Do Until-Loop, Do-Loop While и Do-Loop Until.

Оператор Do While-Loop обеспечивает многократное выполнение блока операторов до тех пор пока УСЛОВИЕ соблюдается, а оператор Do Until-Loop – пока УСЛОВИЕ не соблюдается. Операторы Do-Loop While и Do-Loop Until отличаются от перечисленных выше двух операторов тем, что сначала блок операторов выполняется по крайней мере один раз, а потом проверяется УСЛОВИЕ.

Синтаксис:

Do While УСЛОВИЕ БЛОК_ОПЕРАТОРОВ Loop	Do Until УСЛОВИЕ БЛОК_ОПЕРАТОРОВ Loop	Do БЛОК_ОПЕРАТОРОВ Loop While УСЛОВИЕ	Do БЛОК_ОПЕРАТОРОВ Loop Until УСЛОВИЕ
---	---	---	---

Задание: отладьте представленные программы в module «Задание04», результаты представьте на рабочем листе «Задание04».

Самостоятельно: в module «Задание04» напишите программу, вычисляющую сумму ряда, описывающего заданную функцию с заданной точностью. С клавиатуры вводятся вещественное число x (аргумент) и вещественное число e (точность), которое задает условие окончания суммирования ряда: если очередное слагаемое по модулю меньше e , то это слагаемое надо учесть в сумме и суммирование прекратить. Программа должна выдать значение заданной функции и вычисленное значение суммы ряда, расположив их друг под другом.

Вариант 1

$$-\ln(1-x) = x + \frac{x^2}{2} + \frac{x^3}{3} + \dots + \frac{x^n}{n} + \dots \quad \text{для } -1 \leq x < 1$$

Вариант 2

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + (-1)^{n-1} \frac{x^n}{n} + \dots \quad \text{для } -1 \leq x < 1$$

Вариант 3

$$\frac{1}{2} \ln\left(\frac{1+x}{1-x}\right) = x + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n-1}}{2n-1} + \dots \quad \text{для } -|x| < 1$$

Вариант 4

$$\operatorname{arctg}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + (-1)^{n-1} \frac{x^{2n-1}}{2n-1} + \dots \quad \text{для } -|x| \leq 1$$

Вариант 5

$$\frac{1}{(x-1)^2} = 1 + 2x + 3x^2 + \dots + nx^{n-1} + \dots \quad \text{для } -|x| < 1$$

Вариант 6

$$\frac{1-x^2}{(1+x^2)^2} = 1 - 3x^2 + 5x^4 - \dots + (-1)^{n-1} (2n-1)x^{2n-2} + \dots \quad \text{для } -|x| < 1$$

Вариант 7

$$\frac{2}{(1-x)^3} = 1 \times 2 + 2 \times 3x + 3 \times 4x^2 + \dots + n(n+1)x^{n-1} + \dots \quad \text{для } -|x| < 1$$

Вариант 8

$$\frac{x}{(x-1)^2} = \frac{1}{x} + \frac{2}{x^2} + \frac{3}{x^3} + \dots + \frac{n}{x^n} + \dots \quad \text{для } -|x| < 1$$

Вариант 9

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots \quad \text{для } -1 \leq x < 1$$

Вариант 10

$$\sin\left(x + \frac{\pi}{4}\right) = \frac{\sqrt{2}}{2} \left(1 + x - \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} - \dots + (-1)^{\frac{n^2-n}{2}} \frac{x^n}{n!} + \dots \right) \quad \text{для } -1 \leq x < 1$$

Задание №5

Стандартные функции для работы с массивами

Используются следующие функции для работы с массивами:

Count	Количество чисел в массиве (СЧЕТ)
CountA	Количество элементов массива (СЧЕТЗ)
Sum	Сумма элементов массива (СУММ)
SumProduct	Сумма произведений элементов массива (СУММПРОИЗВ)
SumSq	Сумма квадратов элементов массива (СУММКВ)
SumXmY2	Сумма квадратов разностей элементов двух массивов (СУММКВРАЗН)
SumX2mY2	Сумма разностей квадратов элементов двух массивов (СУММРАЗНКВ)

Напишем функцию, вычисляющую коэффициент корреляции. Коэффициент корреляции двух последовательностей $x_i, y_i, i \in [1, n]$ вычисляется по формуле:

$$R = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{\left(n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2 \right) \left(n \sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i \right)^2 \right)}}$$

Function RR(x As Variant, y As Variant) As Double

Dim n As Integer

Dim sx, sy, sxy, sx2, sy2 As Double

n = Application.Count(x)

sx = Application.Sum(x)

sy = Application.Sum(y)

sxy = Application.SumProduct(x, y)

sx2 = Application.SumSq(x)

sy2 = Application.SumSq(y)

RR = (n*sxy-sx*sy)/((n*sx2-sx^2)*(n*sy2-sy^2))^(1/2)

End Function

Пусть, например, необходимо вычислить коэффициент корреляции двух последовательностей: 1, 2, 3, 4, 5, 6 и 5, 8, 11, 12, 18, 21, записанных в диапазоны ячеек A1:A6 и B1:B6, соответственно. Результат необходимо поместить в ячейку C1. Для этого в ячейку C1 необходимо ввести формулу: = R(A1:A6;B1:B6). Коэффициент корреляции можно вычислить и непосредственно функцией КОРРЕЛ.

Самостоятельно: в module «Задание05» запрограммируйте функцию вычисления коэффициента корреляции. Решите задачу первым способом по образцу задания №4. Отладьте программы. Результаты представьте на рабочем листе «Задание05». Для заданных последовательностей сравните результат со встроенной функции EXCEL КОРРЕЛ.

Работа с массивамиПростейшие операции над массивами

- Умножение массива на константу

На рабочем листе выделяется область такого же размера, как и массив множимое. Затем вводится формула `=Массив1*5`. Ввод формулы заканчивается нажатием клавиш `<Ctrl>+<Shift>+<Enter>`, таким образом выполняются операции над массивами.

- Сумма (разность) массивов: `=Массив1 + Массив2`
- Поэлементное произведение (деление) массивов: `=Массив1 * Массив2`
- Массив, каждый элемент которого связан посредством функции с соответствующим элементом первоначального массива: `=sin(Массив1)`

При работе с массивами формула действует на все ячейки диапазона. Нельзя изменять отдельные ячейки в операндах формулы.

Встроенные функции для работы с матрицами

В EXCEL имеются следующие специальные функции для работы с матрицами:

МОБР	(MINVERSE)	Обратная матрица
МОПРЕД	(MDETERM)	Определитель матрицы
МУМНОЖ	(MMULT)	Матричное произведение двух матриц
ТРАНСП	(TRANSPOSE)	Транспонированная матрица

При работе с матрицами перед вводом формулы надо выделить область на рабочем листе, куда будет выведен результат вычислений.

Пример 1: решить систему линейных уравнений с двумя неизвестными.

Решение линейной системы $AX=B$, где A - матрица коэффициентов, B - столбец (вектор) свободных членов, X - столбец (вектор) неизвестных, имеет вид $X=A^{-1}B$, где A^{-1} - обратная матрица.

Для решения этой системы уравнений введем формулу:
`=МУМНОЖ(МОБР(МассивА);МассивВ)`

$$A = \begin{pmatrix} 2 & 1 \\ 4 & 5 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Пример 2: решить систему линейных уравнений $A^2X=B$

Для решения этой системы уравнений введем формулу:

`=МУМНОЖ(МОБР(МУМНОЖ(МассивА; МассивА)); МассивВ)`

$$A = \begin{pmatrix} 7 & 2 \\ 1 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

Пример 3: вычислить квадратичную формулу $z=X^TAX$, где A - квадратная матрица, X - вектор, для вычисления z введем формулу:

`=МУМНОЖ(МУМНОЖ(ТРАНСП(МассивХ); МассивА); МассивХ)`

$$A = \begin{pmatrix} 2 & 4 \\ 1 & 3 \end{pmatrix}, \quad X = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Пример 4: вычислить квадратичную формулу $z=Y^T A^T A Y$. Для вычисления z введем формулу:

`=МУМНОЖ(ТРАНСП(МассивУ);МУМНОЖ(ТРАНСП(МассивА);`

`МУМНОЖ(МассивА; МассивУ)))`

$$A = \begin{pmatrix} 7 & 2 \\ 1 & 4 \end{pmatrix}, \quad Y = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

Visual Basic for Applications

Пример 5: решить систему m линейных уравнений с n неизвестными ($m > n$).

$$A = \begin{pmatrix} 3 & 2 \\ 4 & -5 \\ 3 & 3 \end{pmatrix} \quad B = \begin{pmatrix} 7 \\ 40 \\ 3 \end{pmatrix}$$

Решение может быть получено отбором n любых линейно независимых уравнений и применением решения для системы n линейных уравнений с n неизвестными (Пример 1). При этом полученное решение будет удовлетворять и остальным $m-n$ уравнениям.

Более общий подход – метод наименьших квадратов. Для этого обе части матричного уравнения системы умножают слева на транспонированную матрицу A^T .

$$A^T A X = A^T B$$

Затем обе части уравнения умножают слева на матрицу $(A^T A)^{-1}$. Если эта матрица существует, то система определена. С учетом того, что $(A^T A)^{-1} * (A^T A) = E$, получаем

$$X = (A^T A)^{-1} A^T B$$

Стандартные функции для работы с матрицами

Используются следующие функции для работы с матрицами:

Mmult	Произведение двух матриц
Minverse	Обратная матрица
Transpose	Транспонированная матрица
Mdeterm	Определитель матрицы

Напишем функцию, решающую систему линейных уравнений $A X = B$ (пример 1), где A - матрица коэффициентов, B - столбец свободных членов, X - столбец неизвестных.

```
Function RLS(A As Variant, B As Variant) As Variant
    RLS=Application.Mmult(Application.Minverse(A),B)
End Function
```

Напишем еще одну функцию, вычисляющую квадратичную форму $z = X^T A X$ (пример 3), где A - матрица коэффициентов, X - столбец неизвестных.

```
Function KF(A As Variant, Z As Variant) As Variant
    KF=Application.Mmult(Application.Mmult(Application.Transpose(Z), A), Z)
End Function
```

Самостоятельно:**Вариант 1**

Решить системы линейных уравнений $AX = B$, $A^3X = B$ и вычислить значение квадратичной формы $z = Y^T A^T A^2 Y$, где

$$A = \begin{pmatrix} 9 & 5 & 4 & 7 \\ 4 & 6 & 8 & 7 \\ 5 & 8 & 7 & 6 \\ 5 & 6 & 8 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 6 \\ 3 \\ 7 \end{pmatrix}, \quad Y = \begin{pmatrix} 2 \\ 6 \\ 2 \\ 3 \end{pmatrix}.$$

Вариант 2

Решить системы линейных уравнений $AX = B$, $A^2 A^T X = B$ и вычислить значение квадратичной формы $z = Y^T A^3 Y$, где

$$A = \begin{pmatrix} 9 & 6 & 3 & 8 \\ 4 & 6 & 7 & 4 \\ 2 & 3 & 5 & 3 \\ 4 & 8 & 3 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix}, \quad Y = \begin{pmatrix} 3 \\ 5 \\ 1 \\ 3 \end{pmatrix}.$$

Вариант 3

Решить системы линейных уравнений $AX = B$, $AA^T AX = B$ и вычислить значение квадратичной формы $z = Y^T A^T A^3 Y$, где

$$A = \begin{pmatrix} 2 & 3 & 5 & 2 \\ 5 & 2 & 7 & 5 \\ 4 & 2 & 1 & 7 \\ 7 & 5 & 4 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 2 \\ 3 \\ 2 \end{pmatrix}, \quad Y = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 2 \end{pmatrix}.$$

Вариант 4

Решить системы линейных уравнений $AX = B$, $A^2 A^T AX = B$ и вычислить значение квадратичной формы $z = Y^T A^T AA^T Y$, где

$$A = \begin{pmatrix} 1 & 4 & 2 & 5 \\ 4 & 4 & 5 & 3 \\ 1 & 2 & 6 & 8 \\ 3 & 7 & 3 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 8 \\ 1 \\ 7 \end{pmatrix}, \quad Y = \begin{pmatrix} 1 \\ 2 \\ 5 \\ 1 \end{pmatrix}.$$

Вариант 5

Решить системы линейных уравнений $AX = B$, $AA^T A^2 X = B$ и вычислить значение квадратичной формы $z = Y^T A^3 A^T Y$, где

$$A = \begin{pmatrix} 9 & 6 & 3 & 8 \\ 4 & 6 & 7 & 4 \\ 2 & 3 & 5 & 3 \\ 4 & 8 & 3 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix}, \quad Y = \begin{pmatrix} 3 \\ 5 \\ 1 \\ 3 \end{pmatrix}.$$

Visual Basic for Applications

Вариант 6

Решить системы линейных уравнений $AX = B$, $A^3A^T X = B$ и вычислить значение квадратичной формы $z = Y^T A^2 A^T A Y$, где

$$A = \begin{pmatrix} 2 & 4 & 7 & 4 \\ 4 & 1 & 6 & 2 \\ 8 & 3 & 6 & 7 \\ 6 & 3 & 5 & 7 \end{pmatrix}, \quad B = \begin{pmatrix} 2 \\ 0 \\ 3 \\ 1 \end{pmatrix}, \quad Y = \begin{pmatrix} 2 \\ 1 \\ 4 \\ 1 \end{pmatrix}.$$

Вариант 7

Решить системы линейных уравнений $AX = B$, $A^T A^3 X = B$ и вычислить значение квадратичной формы $z = Y^T A A^T A^2 Y$, где

$$A = \begin{pmatrix} 3 & 3 & 4 & 5 \\ 2 & 6 & 4 & 6 \\ 3 & 4 & 5 & 5 \\ 1 & 9 & 3 & 6 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 4 \\ 0 \\ 3 \end{pmatrix}, \quad Y = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \end{pmatrix}.$$

Вариант 8

Решить системы линейных уравнений $AX = B$, $AA^T A^2 X = B$ и вычислить значение квадратичной формы $z = Y^T A^2 A^T A Y$, где

$$A = \begin{pmatrix} 2 & 1 & 5 & 2 \\ 5 & 2 & 2 & 6 \\ 2 & 2 & 1 & 2 \\ 1 & 3 & 3 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 3 \\ 0 \\ 2 \end{pmatrix}, \quad Y = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 1 \end{pmatrix}.$$

Вариант 9

Решить системы линейных уравнений $AX = B$, $A^T A A^T X = B$ и вычислить значение квадратичной формы $z = Y^T A A^T A A^T Y$, где

$$A = \begin{pmatrix} 7 & 6 & 2 & 7 \\ 4 & 9 & 5 & 5 \\ 2 & 3 & 4 & 9 \\ 1 & 5 & 6 & 9 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 2 \\ 0 \\ 2 \end{pmatrix}, \quad Y = \begin{pmatrix} 1 \\ 1 \\ 3 \\ 3 \end{pmatrix}.$$

Вариант 10

Решить системы линейных уравнений $AX = B$, $A^2 A^T A X = B$ и вычислить значение квадратичной формы $z = Y^T A A^T A A^T Y$, где

$$A = \begin{pmatrix} 3 & 6 & 5 & 2 \\ 4 & 6 & 3 & 5 \\ 2 & 3 & 2 & 6 \\ 2 & 4 & 3 & 6 \end{pmatrix}, \quad B = \begin{pmatrix} 3 \\ 0 \\ 4 \\ 3 \end{pmatrix}, \quad Y = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

Самостоятельно: для выбранного примера реализуйте его двумя способами:

1. с помощью встроенных функций EXCEL
2. с помощью стандартных функций VBA для работы с матрицами.

Результаты оформите в module «Задание05» и на рабочем листе «Задание05».

Работу выполните пошагово.

Задание №6**Решение системы линейных уравнений методом Гаусса**

Определить решение системы линейных алгебраических уравнений (СЛАУ) с помощью метода Гаусса:

$$2x_1 + 3x_2 + 7x_3 + 6x_4 = 1,$$

$$3x_1 + 5x_2 + 3x_3 + x_4 = 3,$$

$$5x_1 + 3x_2 + x_3 + 3x_4 = 4,$$

$$3x_1 + 3x_2 + x_3 + 6x_4 = 5.$$

Такие задачи решаются точными и итерационными методами. Первые дают точное решение за конечное число операций, если они выполняются без погрешности. Решение, получаемое при помощи итерационных методов, зависит от заданной погрешности вычислений и от начального приближения. Метод Гаусса основан на приведении расширенной матрицы путем последовательного исключения неизвестных, к треугольному виду. Расширенная матрица нашей системы имеет вид A1:E4:

	A	B	C	D	E	F	G	H	I	J	K	L
1	2	3	7	6	1		1	0	0	0	0,185714	0,185714
2	3	5	3	1	3		0	1	0	0	0,778571	0,778571
3	5	3	1	3	4		0	0	1	0	-0,63571	-0,63571
4	3	3	1	6	5		0	0	0	1	0,457143	0,457143
5												
6	2	3	7	6	1							
7	0	0,5	-7,5	-8	1,5		1 =		1		0,185714	
8	0	-4,5	-16,5	-12	1,5		3 =		3		0,778571	
9	0	-1,5	-9,5	-3	3,5		4 =		4		-0,63571	
10							5 =		5		0,457143	
11	2	3	7	6	1						0,785714	
12	0	0,5	-7,5	-8	1,5							
13	0	0	-84	-84	15							
14	0	0	-32	-27	8							
15												
16	2	3	7	6	1							
17	0	0,5	-7,5	-8	1,5							
18	0	0	-84	-84	15							
19	0	0	0	0	5	2,285714						

1. Прямой ход метода Гаусса (исключение неизвестных)

1. Содержимое ячеек A1:E1 копируем в ячейки A6:E6, A11:E11 и A16:E16
2. В диапазон ячеек A7:E7 вводим формулу (умножаем первую строку на A2 и делим ее на \$A\$1, вычитаем ее из второй, обращая в нуль коэффициенты при x_1 во втором уравнении системы): $\{=A2:E2-\$A\$1:\$E\$1*(A2/\$A\$1)\}$ {Ctrl+Shift+Enter}
3. Заполняем этим уравнением диапазон A7:E9 для обращения в нуль коэффициентов при x_1 в третьем и четвертом уравнениях системы.
4. Копируем значения из диапазона ячеек A7:E7 в диапазоны A12:E12 и A17:E17 (для копирования значений без формул используется команда **Правка, Специальная вставка** переключатель установить в положение **Значения**).
5. В диапазон ячеек A13:E13 вводим формулу: $\{=A8:E8-\$A\$7:\$E\$7*(B8/\$B\$7)\}$
6. Заполняем диапазон A13:E14. Это обратит в нуль коэффициенты при x_2 в третьем и четвертом уравнениях системы.
7. Копируем значения из A13:E13 в A18:E18.
8. В диапазон ячеек A19:E19 вводим формулу: $\{=A14:E14-\$A\$13:\$E\$13*(C14/\$C\$13)\}$ которая обращает в нуль коэффициент при x_3 четвертого уравнения системы.

Visual Basic for Applications

2. Обратный ход метода Гаусса (нахождение неизвестных).

Вводим в диапазоны G4:K4, G3:K3, G2:K2 и G1:K1 формулы:

{=A19:E19/D19}

{=(A18:E18-G4:K4*D18)/C18}

{=(A17:E17-G4:K4*D17-G3:K3*C17)/B17}

{=(A16:E16-G4:K4*D16-G3:K3*C16-G2:K2*B16)/A16}

В диапазоне ячеек K1:K4 получено решение системы, которое совпадает с решением, полученным с помощью **Поиска решений**.

```
Sub Gauss()  
    Dim i, j, k, n As Integer  
    Dim h As Single  
    Dim A(10, 10), B(10), X(10) As Single  
    n = InputBox("Введите количество уравнений")  
    Rem Заполнение матрицы коэффициентов и  
    Rem вектора свободных членов  
    For i = 1 To n  
        For j = 1 To n  
            A(i, j) = Cells(i, j)  
        Next j  
        B(i) = Cells(i, 5)  
    Next i  
    Rem Прямой ход метода Гаусса  
    For i = 1 To n - 1  
        For j = i + 1 To n  
            A(j, i) = -A(j, i) / A(i, i)  
            For k = i + 1 To n  
                A(j, k) = A(j, k) + A(j, i) * A(i, k)  
            Next k  
            B(j) = B(j) + A(j, i) * B(i)  
        Next j  
    Next i  
    Rem Обратный ход метода Гаусса  
    X(n) = B(n) / A(n, n)  
    For i = n - 1 To 1 Step -1  
        h = B(i)  
        For j = i + 1 To n  
            h = h - X(j) * A(i, j)  
        Next j  
        X(i) = h / A(i, i)  
    Next i  
    Rem Вывод результатов  
    For i = 1 To n  
        Cells(i, 12) = X(i)  
    Next i  
End Sub
```

Самостоятельно: на рабочем листе «Задание06» средствами EXCEL решите методом Гаусса систему линейных уравнений из примеров задания №5. В module «Задание06» запрограммируйте алгоритм решения системы линейных уравнений. Отладьте программу. Внесите в нее изменения по заданию преподавателя.

Задание №7Решение оптимизационных задач с помощью «Поиска решений» табличного процессора EXCEL**Пример 1:** задача об оптимальном использовании ресурсов.

Фирма производит две модели книжных полок: А и В. Их производство ограничено наличием сырья и временем машинной обработки. Для каждого изделия модели А требуется 3 м² досок, а для изделия модели В – 4 м². Фирма может получить до 1700 м² досок в неделю. Для каждого изделия модели А требуется 12 минут машинного времени, а для изделия модели В – 30 минут. В неделю можно использовать 160 часов машинного времени. Сколько изделий каждой модели следует фирме выпускать в неделю, если каждое изделие модели А приносит 2 рубля прибыли, а В – 4 рубля прибыли?

Для решения этой задачи необходимо построить математическую модель:

- Для определения каких величин (переменных) строится модель?
Переменными являются: А - объем производства полок А, В - объем производства полок В.
- В чем состоит цель выбора оптимальных переменных?
Цель – определить среди всех допустимых значений А и В такие, которые максимизируют прибыль Р. Прибыль от производства А и В равна: $P = 2 \cdot A + 4 \cdot B$
- Ограничения неизвестных?
Объем производства положительный и счетный: $A, B \geq 0$, А, В = целые.
Расход материала не может превосходить его запас: $3 \cdot A + 4 \cdot B \leq 1700$.
Временные затраты на изготовление изделий не могут превышать лимита машинного времени: $12 \cdot A + 30 \cdot B \leq 9600$.

Исходные данные, искомые величины, функция цели и ограничения заполняются на рабочем листе EXCEL:

	А	В	С	Д	Е
1		Сырье	Обработка	Прибыль	Кол-во?
2	А	3	12	2	300
3	В	4	30	4	200
4		1700	9600		
5					
6		Ф-ция цели			
7		=D2*E2+D3*E3			
8		Ограничения			
9		=B2*E2+B3*E3	<=	=B4	
10		=C2*E2+C3*E3	<=	=C4	

Visual Basic for Applications

В меню «Сервисе/Поиск решения» выполняется заполнение форм ввода окна:

В результате будет получено следующее решение, при котором будут оптимально выполнены все условия и ограничения:

	A	B	C	D	E	F	G	H	I	J
1		Сырье	Обработка	Прибыль	Кол-во?					
2	А	3	12	2	300					
3	В	4	30	4	200					
4		1700	9600							
5										
6		Ф-ция цели								
7		1400								
8		Ограничения								
9		1700 <=		1700						
10		9600 <=		9600						
11										

Элементы диалогового окна Поиск решения

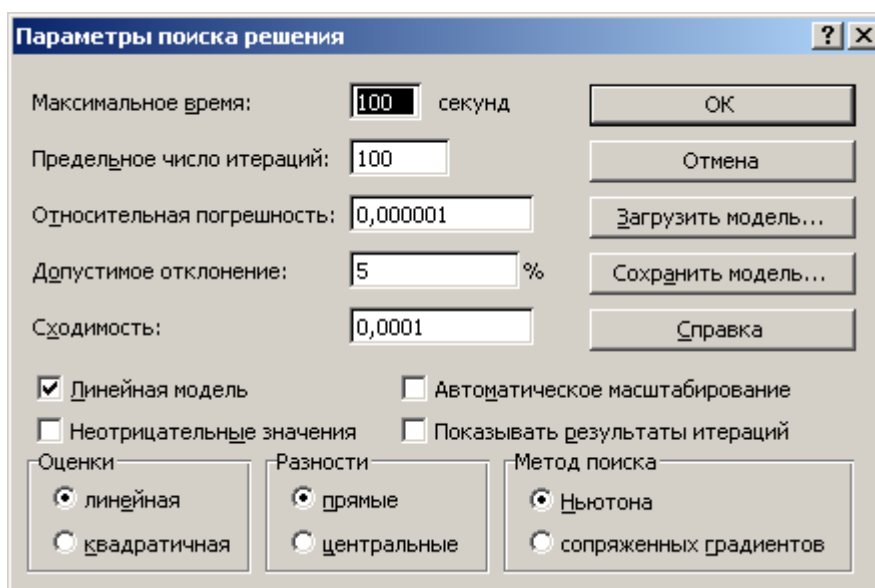
В поле **Установить целевую ячейку** дается ссылка на ячейку с функцией, для которой будет находиться максимум, минимум или заданное значение. Тип взаимосвязи между решением и целевой ячейкой задается переключателем в группе **Равной** (максимальному значению, минимальному значению или значению).

В поле **Изменяя ячейки** указываются ячейки, которые должны изменяться в процессе поиска решения задачи, т.е. ячейки, отведенные под переменные.

Ограничения, налагаемые на переменные, отображаются в поле **Ограничения**. Средство поиска решений допускает ограничения в виде равенств, неравенств, а также позволяет ввести требование целочисленности.

Ограничения добавляются по одному в диалоговом окне **Добавления ограничений**. В поле **Ссылка на ячейку** вводится левая часть на ограничение, в поле **Ограничение** – правая часть. С помощью раскрывающегося списка вводится отношение между левой и правой частями ограничения. Нажатие кнопки **Ок** завершает ввод ограничений.

С помощью кнопки **Параметры** диалогового окна **Поиск решения** вызывается диалоговое окно **Параметры поиска решения**, где можно изменять условия и варианты поиска решения задачи.



Элементы окна **Параметры поиска решения**:

Поле **Максимальное время** служит для задания ограничения времени. Поле **Предельное число итераций** ограничивает число промежуточных вычислений. Поля **Относительная погрешность** и **Допустимое отклонение** служат для задания точности поиска решения. Флажок **Линейная модель** служит для поиска решения линейной задачи оптимизации или нелинейной аппроксимации нелинейной задачи. В случае нелинейной задачи этот флажок должен быть сброшен, в случае линейной задачи – установлен. Флажок **Показывать результаты итераций** служит для приостановки поиска решений и просмотра результатов. Флажок **Автоматическое масштабирование** служит для включения автоматической нормализации входных и выходных значений. Группа **Оценка** служит для выбора метода экстраполяции. Группа **Разности** служит для выбора метода численного дифференцирования. Группа **Метод поиска** служит для выбора алгоритма оптимизации.

Отчет о результатах решения задачи выбирается в диалоговом окне **Результаты поиска решения: Результаты, Устойчивость, Пределы**.

Visual Basic for Applications

Пример 2: Задача о составе смеси.

Фирма занимается разработкой нового типа автомобильных покрышек для дорог Европы, предъявляющей свои требования:

	Требования ЕС	А	В	С
Износостойкость	6	5	6	8
Сцепление	10	10	12	8
Прочность	5	8	5	3

Для производства покрышек используются три полимера: А, В и С - с характеристиками в условных и безразмерных единицах, указанными выше. В каких пропорциях необходимо смешать эти полимеры, чтобы износостойкость оказалась максимальной?

Решения задачи проводится по аналогии с предыдущей задачей:

- Для определения каких величин (переменных) строится модель?
Переменными являются: А – доля полимера А, В – доля полимера В и С – доля полимера С в смеси.
- В чем состоит цель выбора оптимальных переменных?
- Цель – определить среди всех допустимых значений А, В и С такие, которые максимизируют износостойкость покрышек. Износостойкость равна: $5 \cdot A + 6 \cdot B + 8 \cdot C$
- Ограничения неизвестных?

Доли компонентов смеси положительны:

$$A, B, C \geq 0;$$

Сумма долей компонентов должна составлять 100% :

$$A + B + C = 1.$$

Износостойкость должна соответствовать:

$$5 \cdot A + 6 \cdot B + 8 \cdot C \geq 6$$

Сцепление должно соответствовать:

$$10 \cdot A + 12 \cdot B + 8 \cdot C \geq 10$$

Прочность должна соответствовать:

$$8 \cdot A + 5 \cdot B + 3 \cdot C \geq 5$$

После подстановки исходных данных, условий и ограничений будет получено решение:

	А	В	С	Д	Е
1		Требования	А	В	С
2	И	6	5	6	8
3	С	10	10	12	8
4	П	5	8	5	3
5			0,25	0,375	0,375
6	ФЦ	=СУММПРОИЗВ(С2:Е2;С5:Е5)			
7	Ограничения				
8		И	=СУММПРОИЗВ(С2:Е2;\$C\$5:\$E\$5)	>=	=B2
9		С	=СУММПРОИЗВ(С3:Е3;\$C\$5:\$E\$5)	>=	=B3
10		П	=СУММПРОИЗВ(С4:Е4;\$C\$5:\$E\$5)	>=	=B4
11		Состав	=C5+D5+E5	=	1

Пример 3: решение оптимизационной задачи средствами VBA

Производятся два вида продукции: столы и стулья. Для изготовления стула требуется 3 м^2 древесины, стола – 7 м^2 , на складе запасено 420 м^2 древесины. Время на изготовление стула уходит 2 часа, стола – 8 часов, лимит рабочего времени равен 400 часам. Прибыль от продажи стула равна 50 рублям, стола – 150 рублям. Сколько стульев и столов можно изготовить, чтобы не превысить ограничений по материалам и времени, и при этом прибыль была бы максимальной?

Можно искать требуемые значения X (стулья) и Y (столы), перебирая все значения которые они могут принимать. Для использования метода перебора искомые величины должны быть ограничены снизу и сверху. Нижние границы определены условиями неотрицательности и целостности значений, т.е. $X \geq 0$ и $Y \geq 0$. А верхние границы определяются из ограничений:

$$\begin{array}{llll} 2 * X + 8 * Y \leq 400 & Y \leq (400 - 2 * X) / 8 & \Rightarrow & Y \leq 50 \\ 3 * X + 7 * Y \leq 420 & Y \leq (420 - 3 * X) / 7 & \Rightarrow & Y \leq 60 \\ 2 * X + 8 * Y \leq 400 & X \leq (400 - 8 * Y) / 2 & \Rightarrow & X \leq 200 \\ 3 * X + 7 * Y \leq 420 & X \leq (420 - 7 * Y) / 3 & \Rightarrow & X \leq 140 \end{array}$$

Sub Solver()

Rem Поиск оптимального решения

Dim P(140, 50) As Integer

Dim stul, stol, d, t, i, j, max As Integer

For stul = 1 To 140

For stol = 1 To 50

d = 3 * stul + 7 * stol

t = 2 * stul + 8 * stol

' Проверка ограничений

If d <= 420 And t <= 400 Then

' Заполнение массива целевой функции

P(stul, stol) = 50 * stul + 150 * stol

Else: P(stul, stol) = 0

End If

Next stol

Next stul

' Поиск максимума целевой функции

max = P(1, 1)

For i = 1 To 140

For j = 1 To 50

If P(i, j) > max Then max = P(i, j): stul = i: stol = j

Next j

Next i

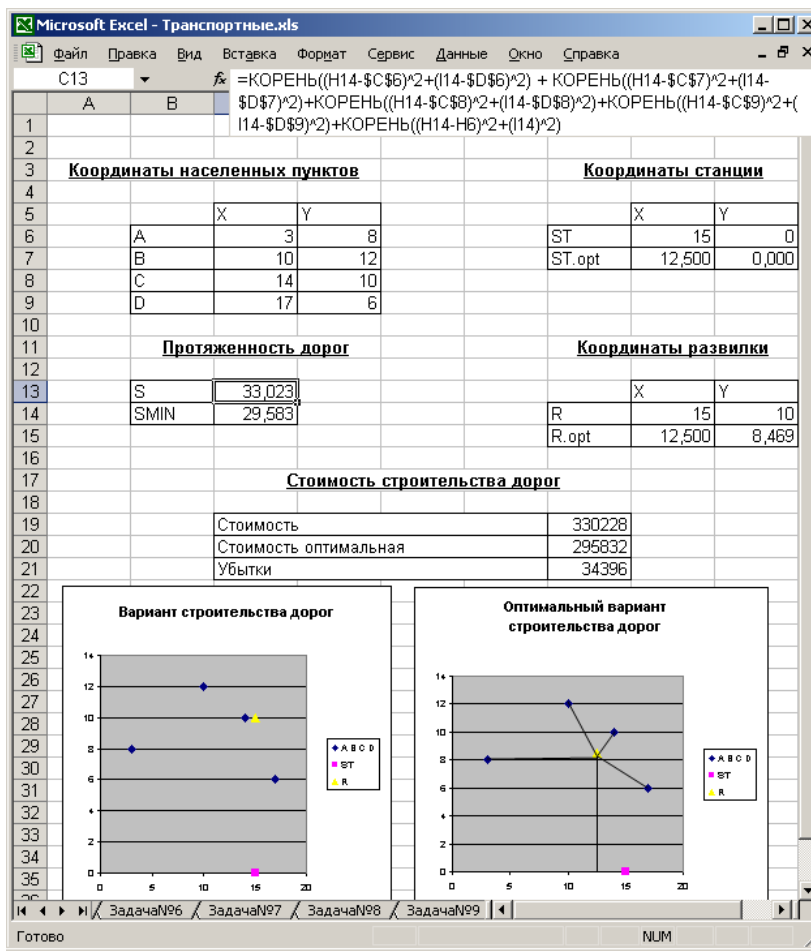
Cells(6, 3) = max: Cells(2, 6) = stol: Cells(3, 6) = stul

End Sub

Самостоятельно: из пособия «Экономико-математические методы и моделирование. Поиск оптимальных решений» решите оптимизационную задачу средствами EXCEL **Поиск решения** и с помощью программы на VBA. Результаты оформите в module «Задание07» и на рабочем листе «Задание07».

Задание №8

Метод спирального координатного спуска при решении задач оптимизации



Постановка задачи

Заданы координаты (X, Y) четырех населенных пунктов. Требуется определить место строительства ж.д. станции ST и развилки R, соединяющей станцию с населенными пунктами, так чтобы затраты на строительство работ были минимальны.

Предполагается, что ж.д. проходит по прямой Y=0. Развилка имеет координаты (X₀, Y₀).

Затраты на строительство дорог будут минимальными, если протяженность всех дорог будет наименьшей. Общую длину дорог можно записать как функцию переменных X₀, Y₀, X, где (X₀, Y₀) - координаты развилки, (X, 0) – координаты станции:

$$F(X_0, Y_0, X) = \sqrt{(X_0 - 3)^2 + (Y_0 - 8)^2} + \sqrt{(X_0 - 10)^2 + (Y_0 - 12)^2} + \sqrt{(X_0 - 14)^2 + (Y_0 - 10)^2} + \sqrt{(X_0 - 17)^2 + (Y_0 - 6)^2} + \sqrt{(X_0 - X)^2 + Y_0^2}$$

Необходимо найти такую тройку чисел {X₀, Y₀, X}, чтобы она обеспечивала минимум целевой функции F(X₀, Y₀, X).

Пояснение к выполнению задачи средствами EXCEL:

1. Подготовить таблицы для внесения данных.
2. Заполнить данными таблицу «Координаты населенных пунктов».
3. Построить точечную диаграмму для отображения населенных пунктов на координатной плоскости.
4. Выбрать возможные (приближенные) координаты станции ST (15, 0) и развилки R (15, 10).

5. В ячейку C13 занести формулу для вычисления протяженности дорог S по проекту

$$=КОРЕНЬ((H14-СC\$6)^2+(I14-ДD\$6)^2) + КОРЕНЬ((H14-СC\$7)^2+(I14-ДD\$7)^2) +$$

$$КОРЕНЬ((H14-СC\$8)^2+(I14-ДD\$8)^2)+КОРЕНЬ((H14-СC\$9)^2$$

$$+ (I14-ДD\$9)^2)+КОРЕНЬ((H14-H6)^2+(I14)^2)$$
6. Скопировать формулу в ячейку C14
7. Используя функцию EXCEL **Поиск решения** получить оптимальный вариант расположения станции и развилки: установите целевую ячейку C14, равной минимальному значению, изменяя ячейки H7, H15, I15.

Программа, реализующая алгоритм спирального координатного спуска:

Function fns(x, y, z)

Rem вычисление общей длины дорог

f1 = Sqr((x - z) ^ 2 + y ^ 2)

f2 = Sqr((x - 3) ^ 2 + (y - 8) ^ 2)

f3 = Sqr((x - 10) ^ 2 + (y - 12) ^ 2)

f4 = Sqr((x - 14) ^ 2 + (y - 10) ^ 2)

f5 = Sqr((x - 17) ^ 2 + (y - 6) ^ 2)

fns = f1 + f2 + f3 + f4 + f5

End Function

Sub z8()

Dim a(3) As Single

a(1) = 15: Rem X развилки

a(2) = 10: Rem Y развилки

a(3) = 15: Rem X станции

h = 1: e = 0.001

r = fns(a(1), a(2), a(3)): s = r

Do: Rem метод спирального координатного спуска

For i = 1 To 3

Do

a(i) = a(i) + h

s1 = s

s = fns(a(1), a(2), a(3))

Loop While s < s1

Next i

h = -h / 5

Loop While Abs(h) > e

MsgBox ("Оптимальная координата станции X " & a(3))

MsgBox ("Оптимальная координата развилки X " & a(1))

MsgBox ("Оптимальная координата развилки Y " & a(2))

MsgBox ("Протяженность дорог " & s)

MsgBox ("Протяженность дорог по оптимальному проекту " & r)

End Sub

Задание: решите оптимизационную задачу средством EXCEL **Поиск решения** и с помощью программы на VBA. Результаты оформите в module «Заданиек08» и на рабочем листе «Задание08».

Задание №9

Формы

Панель инструментов вкладка **Разработчик/Элементы управления формы** содержит следующие объекты: **Кнопка, Поле со списком, Флажок, Счетчик, Список, Переключатель, Группа, Подпись, Полоса прокрутки, Поле, Поле с раскрывающимся списком**

Создайте рабочую книгу с пятью рабочими листами: «Оглавление», «Май», «Июнь», «Июль», «Вспомогательный».

Пример 1

На лист модуля ввести следующие три процедуры:

```
Sub Май()  
    Sheets("Май").Activate  
End Sub  
  
Sub Июнь()  
    Sheets("Июнь").Activate  
End Sub  
  
Sub Июль()  
    Sheets("Июль").Activate  
End Sub
```

Метод **Activate** активизирует объект, в данном случае – рабочий лист.

С помощью объекта **Кнопка** создайте на рабочем листе «Оглавление» три кнопки с именами «Май», «Июнь», «Июль». Чтобы изменить имя кнопки нужно воспользоваться командой **Изменить текст** контекстного меню, которое вызывается нажатием правой кнопки мыши на выбранной кнопке. С помощью команды **Назначить макрос...** в поле ввода **Имя макроса** диалогового окна выбранной кнопке назначается соответствующая процедура.

Пример 2

На рабочем листе «Вспомогательный» в ячейки A1:A3 ввести: Май, Июнь, Июль. Этому диапазону присвоить имя «Список», ячейке A6 – «Номер», ячейке A7 – «Лист».

В ячейку A7 ввести формулу: =ИНДЕКС(Список;Номер;1)

Функция **ИНДЕКС** возвращает значение ячейки диапазона «Массив» с заданными номерами строки и столбца:

```
=ИНДЕКС (Массив; Номер_строки; Номер_столбца)
```

На листе модуля ввести процедуру:

```
Sub AA_1()  
    Dim s As String  
    s = Range("Лист").Value  
    Sheets(s).Select  
End Sub
```

Процедура AA_1 считывает значение из ячейки «Лист» в строковую переменную s и активизирует лист с именем, записанным в эту переменную.

С помощью объекта **Поле со списком** на листе «Оглавление» создать раскрывающийся список, назначив ему процедуру AA_1. В контекстном меню для раскрывающегося списка выполнить команду **Формат объекта...** В диалоговом окне этой команды на вкладке **Элемент управления** в поле **Формировать список по диапазону** ввести «Список», в поле **Помещать результат в ячейку** – «Номер», а в поле **Количество строк списка** – 3.

Пример 3

С помощью объекта **Список** на рабочем листе «Оглавление» создать список. Назначить этому списку процедуру AA_1. С помощью команды **Формат объекта...** в поле **Формировать список по диапазону** ввести «Список», в поле **Помещать результат в ячейку** – «Номер», переключатель установить в положение **Только одного значение**.

Пример 4

С помощью объекта **Кнопка** на рабочем листе «Оглавление» создать кнопку, а рядом с помощью объекта **Счетчик** – счетчик. Ячейке рабочего листа «Оглавление», расположенной рядом со счетчиком присвоить имя «Первый» и ввести в нее формулу:

```
=ИНДЕКС (Список;Номер;1)
```

На листе модуля ввести процедуру:

```
Sub AA_2()  
    Dim s As String  
    s = Range("Первый").Value  
    Sheets(s).Activate  
End Sub
```

Назначить эту процедуру кнопке. Для счетчика в диалоговом окне **Форматирование объекта** в поле **Начальное значение** задать 2, в поле **Минимальное значение** – 1, в поле **Максимальное значение** – 3, в поле **Шаг изменения** – 1, а в поле **Помещать результат в ячейку** – «Номер».

Пример 5

С помощью объекта **Кнопка** на рабочем листе «Оглавление» создать кнопку, на поверхности которой написать «Перейти». С помощью объекта **Рамка** создать группу, а с помощью объекта **Переключатель** внутри группы – три положения переключателя.

На листе модуля ввести процедуру:

```
Sub AA_4()  
    Dim s As String  
    Set NS = Sheets("Оглавление")  
    If NS.OptionButtons(1).Value = xlOn Then s = "Май"  
    If NS.OptionButtons(2).Value = xlOn Then s = "Июнь"  
    If NS.OptionButtons(3).Value = xlOn Then s = "Июль"  
    Sheets(s).Select  
End Sub
```

Назначить этой процедуре кнопку «Перейти».

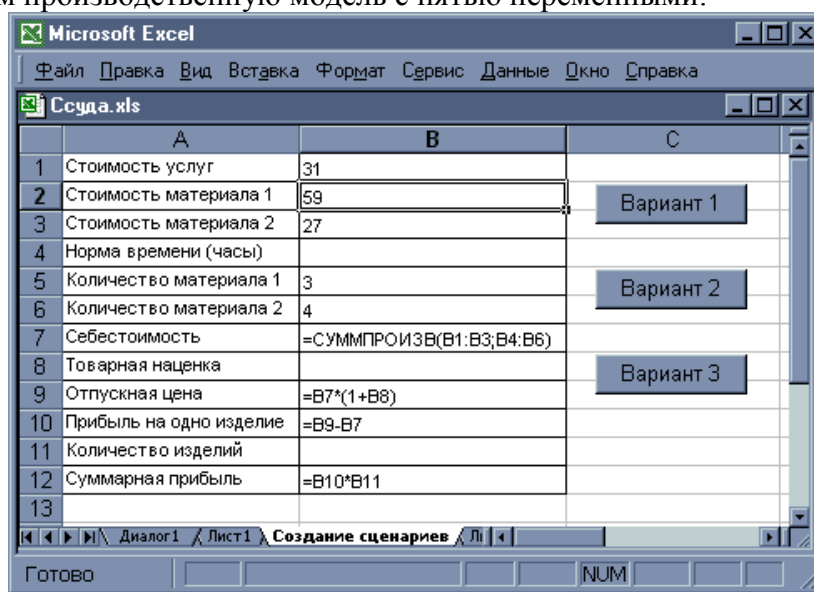
OptionButtons(номер)	Объект управления - кнопка переключателя с указанным номером
Свойство Value объекта OptionButtons	Описывает положение переключателя: выбрана данная кнопка или нет.
Постоянная xlOn, равная 1	Используется, чтобы описать ситуацию, когда кнопка переключателя выбрана.

Задание: выполните примеры по образцу. Примените данные элементы управления для вызова программ из задания №4.

Задание №10

Создание сценариев

Рассмотрим производственную модель с пятью переменными:



Создадим процедуры **Вариант_1**, **Вариант_2**, **Вариант_3**, связанные с соответствующими кнопками, которые осуществляют ввод требуемых значений в ячейки таблицы:

<pre>Sub V_1() Range("C_Y") = 23 Range("C_M_1") = 62 Range("C_M_2") = 32 Range("K_M_1") = 5 Range("K_M_2") = 3 End Sub</pre>	<pre>Sub V_2() Range("C_Y") = 29 Range("C_M_1") = 55 Range("C_M_2") = 24 Range("K_M_1") = 3 Range("K_M_2") = 6 End Sub</pre>	<pre>Sub V_3() Range("C_Y") = 31 Range("C_M_1") = 59 Range("C_M_2") = 27 Range("K_M_1") = 3 Range("K_M_2") = 4 End Sub</pre>
--	--	--

При запуске этих процедур в ячейки будут введены значения, соответствующие этим вариантам:

C_Y	Ячейка B1
C_M_1	Ячейка B2
C_M_2	Ячейка B3
K_M_1	Ячейка B5
K_M_2	Ячейка B6

Задание: выполните пример по образцу. Результаты оформите в module «Задание10» и на рабочем листе «Задание10».

Начисление комиссионных (смотри задание №3)

Комиссионные начисляют по следующему правилу:

- Если объем продаж ≥ 100000 руб., то p_1 % от объема продаж
- Если $50000 \leq$ объем продаж ≤ 100000 руб., то p_2 % от объема продаж
- Если объем продаж ≤ 50000 руб., то p_3 % от объема продаж

Сценарий 1: $p_1=4\%$, $p_2=3\%$, $p_3=2\%$

Сценарий 2: $p_1=6\%$, $p_2=4\%$, $p_3=2\%$

Комиссионные, начисленные по максимальному проценту, будут окрашены красным цветом, по наименьшему - черным, по среднему - синим.

```
Dim ff As Integer
```

```
Function Re(Sale, p1, p2, p3 As Double, ff As Integer)
```

```
Dim s1, s2, s3 As Double
```

```
s1 = 0: s2 = 0: s3 = 0
```

```
If Sale  $\geq$  100000 Then
```

```
s1 = p1: ff = 1
```

```
End If
```

```
If Sale  $\leq$  100000 And Sale  $\geq$  50000 Then
```

```
s2 = p2: ff = 2
```

```
End If
```

```
If Sale  $\leq$  50000 Then
```

```
s3 = p3: ff = 3
```

```
End If
```

```
Re = (s1 + s2 + s3) * Sale
```

```
End Function
```

```
Sub Ta(p1, p2, p3 As Double)
```

```
Dim i As Double, j, n As Integer
```

```
n = Application.Count(Sheets("Задание03").Range("B:B"))
```

```
For j = 2 To n + 1
```

```
Sale = Cells(j, 2).Value
```

```
With Cells(j, 3)
```

```
.Value = Re(Sale, p1, p2, p3, ff)
```

```
Select Case ff
```

```
Case Is = 1
```

```
.Font.ColorIndex = 3
```

```
Case Is = 2
```

```
.Font.ColorIndex = 5
```

```
Case Is = 3
```

```
.Font.ColorIndex = 1
```

```
End Select
```

```
End With
```

```
Next j
```

```
End Sub
```

```
Sub Kom_One()
```

```
Rem Передача процедуре Та параметров по сценарию №1
```

```
Ta 0.04, 0.03, 0.02
```

```
End Sub
```

```
Sub Kom_Two()
```

```
Rem Передача процедуре Та параметров по сценарию №1
```

```
Ta 0.06, 0.04, 0.02
```

```
End Sub
```

Диапазон "B:B" обозначает столбец B.

Application.Count(Sheets("Сценарии_2").Range("B:B")) вычисляет число непустых ячеек в столбце B.

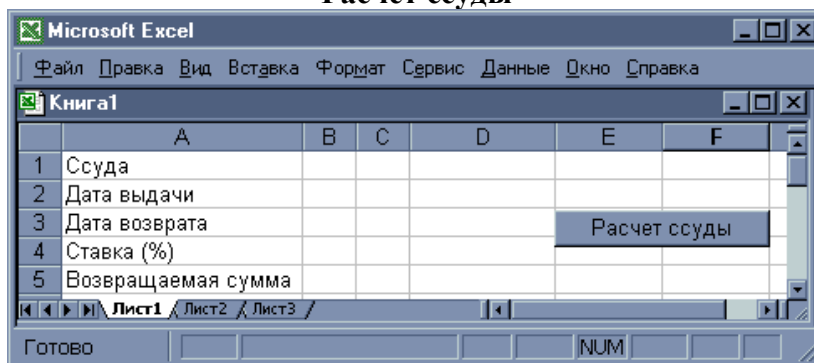
Задание №11

Диалоговые окна

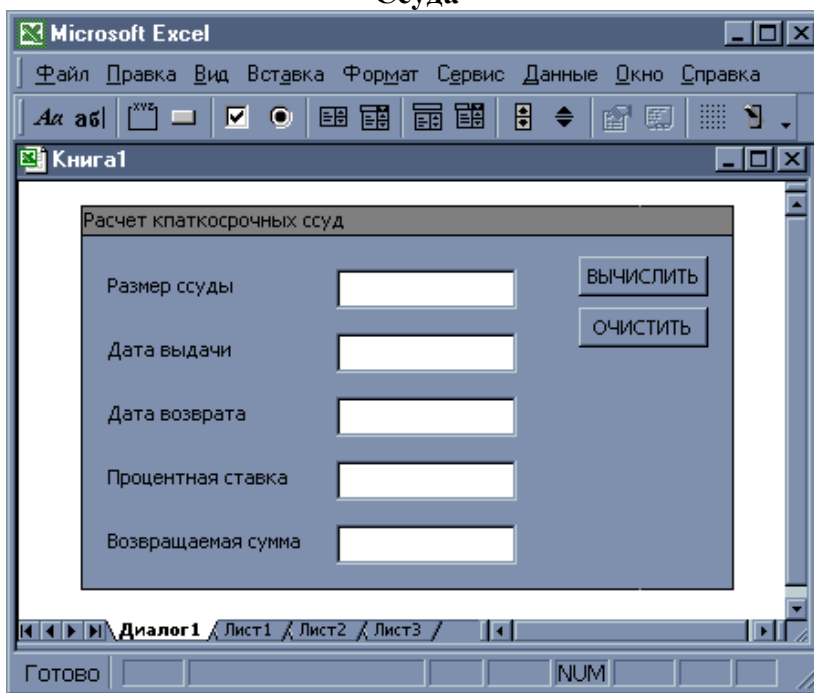
Рассмотрим пример оформления расчета краткосрочной ссуды с использованием диалогового окна.

Создадим рабочую книгу, состоящую из трех листов: рабочего «Расчет ссуды» диалогового – «Ссуда», и листа модуля.

Расчет ссуды



Ссуда



```
Sub DD()  
    DialogSheets("Ссуда").EditBoxes(5).Enabled = False  
    DialogSheets("Ссуда").Show  
End Sub
```

```
Sub SS()  
    Dim p, R, i As Double  
    Dim D_beg, D_end As Date  
    With DialogSheets("Ссуда")  
        p = CDb1(.EditBoxes(1).Text)  
        D_beg = CDate(.EditBoxes(2).Text)  
        D_end = CDate(.EditBoxes(3).Text)  
        i = CDb1(.EditBoxes(4).Text) / 100  
    End With
```

```

If D_end < D_beg Then
    MsgBox "Error in date", vbExclamation, "Calculate ss"
    Exit Sub
End If
R = p * (1 + i) ^ ((D_end - D_beg) / 365)
R = Format(R, "Fixed")
DialogSheets("Ссуда").EditBoxes(5).Text = CStr(R)
Range("D1").Value = p
Range("D2").Value = D_beg
Range("D3").Value = D_end
Range("D4").Value = i
Range("D5").Value = R
End Sub

Sub Cl()
    With DialogSheets("Ссуда")
        .EditBoxes(1).Text = ""
        .EditBoxes(2).Text = ""
        .EditBoxes(3).Text = ""
        .EditBoxes(4).Text = ""
        .EditBoxes(5).Text = ""
    End With
    Range("D1:D5").Clear
End Sub

```

Назначим процедуры следующим кнопкам:

DD Кнопке «Расчет ссуды» рабочего листа «Расчет ссуды»
SS Кнопке «Вычислить» диалогового окна «Ссуда»
CI Кнопке «Очистить» диалогового окна «Ссуда»

Свойство **Enabled** определяет возможность внесения изменений в объекты управления (изменение положения переключателя, ввод текста в поле ввода и т.д.). Задание этого свойства равным **TRUE** позволяет вносить изменения в объект, а **FALSE** - не позволяет. В примере в пятом поле выводится число, вычисляемое программой. Чтобы его невозможно было изменить, в процедуре DD включена команда:

```
DialogSheets("Ссуда").EditBoxes(5).Enabled = False
```

Задание: выполните пример по образцу. Результаты оформите в module «Задание11» и на рабочем листе «Задание11».

Рекомендуемая форма ОПИСАНИЯ ПРОГРАММЫ при оформлении работ:

- **Общие сведения:**
Обозначение и наименование программы. Программное обеспечение, необходимое для функционирования программы. Языки программирования, на которых написана программа.
- **Функциональное назначение:**
Классы решаемых задач. Назначение программы.
- **Описание логической структуры:**
Алгоритм программы. Используемые методы. Структура программы с описанием функций составных частей и связи между ними. Связи программы с другими программами.
- **Используемые технические средства;**
- **Вызов и загрузка:**
Способ вызова программы. Входные точки в программу.
- **Входные данные:**
Характер, организация и предварительная подготовка входных данных. Формат, описание и способ кодирования входных данных.
- **Выходные данные:**
Характер и организация выходных данных. Формат, описание и способ кодирования выходных данных.
- **Примеры работы программы:**
Приводится пример работы программного комплекса, на котором проводится его отладка и тестирование.

ЛИТЕРАТУРА:

1. Булычев Е.В., Кудлаев А.А. Обработка данных с помощью табличного редактора EXCEL. Методические указания.- М., МИИГАиК, 2000, 34 стр.
2. Гарнаев А.Ю. Использование MS Excel и VBA в экономике и финансах.- СПб.: БХВ-Санкт-Петербург, 2000, 336 с.
3. Гарнаев А.Ю. Самоучитель VBA.-СПб.: БХВ - Санкт-Петербург, 2000, 512 с.
4. Гельман В.Я. Решение математических задач средствами EXCEL.- СПб.: Питер, 2003.- 240 с.
5. Гусева О.П., Гусев Е.В., Миронова Н.Н. Одна задача, два решения. М.: Информатика и образование, 2000.- 96 с.
6. Лавренев С.М. EXCEL: Сборник примеров и задач.- М.: Финансы и статистика, 2000. – 336 с.
7. Биллиг В.А. Основы офисного программирования и язык VBA - <http://www.intuit.ru/department/se/vba2000/>
8. Биллиг В.А. Основы офисного программирования и документы EXCEL - <http://www.intuit.ru/department/se/vbaexcel/>