МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ЭКОНОМИКИ И МЕНЕДЖМЕНТА

ЗАДАЧИ ПО ПРОГРАММИРОВАНИЮ в Microsoft Excel

Учебно-методическое пособие
по курсу «Информатика»
для студентов института экономики и менеджмента
направлений подготовки
38.03.01 – Экономика и 38.03.02 – Менеджмент

Томск
2017

РАССМОТРЕНО И УТВЕРЖДЕНО методической комиссией института экономики и менеджмента.
Протокол № 5 от 17 февраля 2017 г.
Председатель МК ИЭМ В.В. Маковеева

Пособие составлено в соответствии с тематикой практических занятий и программой курса «Информатика» для студентов института экономики и менеджмента направлений подготовки 38.03.01 – Экономика и 38.03.02 – Менеджмент.

В данном пособии приведены задачи для освоения программирования в Microsoft Excel 2013 на языке Visual Basic for Application (VBA). Для некоторых типичных задач приведены соответствующие алгоритмы и программы.

Предназначено для преподавателей, аспирантов, студентов и магистрантов дневной, очно-заочной и заочной формы обучения.

СОСТАВИТЕЛЬ: **Лещинский Борис Семенович**, к.т.н., доцент кафедры информационных технологий и бизнес-аналитики ИЭМ НИ ТГУ.

ПРЕДИСЛОВИЕ

В данном пособии приведены задачи для освоения основных возможностей программирования на языке Visual Basic for Application (VBA) для Microsoft Excel.

В первом разделе даются алгоритмы и программы типичных задач, а также комментарии к ним. Во втором разделе приведены контрольные задачи. Для каждой из предложенных задач следует написать программу в универсальной форме. Это означает, что программа должна обеспечивать проведение расчетов при любых допустимых исходных данных. Для контроля правильности написания программ в описании каждой задачи указаны конкретные исходные данные и в приложении А приведены соответствующие ответы. В приложении Б указаны основные операторы языка, которые требуются для решения предлагаемых задач.

Описание каждой задачи состоит из следующих разделов:

– *исходные данные*. В этом разделе указаны, какие данные являются входными для программы и каковы их условные обозначения, используемые в расчетных формулах;

– *расчет*. Здесь приведены расчетные формулы, с помощью которых можно получить требуемый результат;

– *вывести*. В этом разделе указаны величины, которые должны быть вычислены и выведены на экран;

– *конкретные исходные данные для проверки*. Здесь указаны конкретные значения входных данных, что позволяет проверить правильность написания программы.

1 ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ

Разработчик программы, как правило, сталкивается с двумя проблемами: понимание логики решения задачи и описание этой логики средствами конкретного языка программирования. Попытка одновременного решения этих проблем непосредственно при написании текста программы существенно затрудняет разработку. Предпочтительно решать их последовательно. На первом этапе разрабатывается алгоритм решения задачи. При этом, внимание уделяется описанию последовательности действий, которая гарантирует получение правильного результата, безотносительно к языку программирования. На втором этапе осуществляется написание текста программы. Здесь уже требуется записать готовый алгоритм таким образом, чтобы текст удовлетворял синтаксическим и семантическим правилам конкретного языка программирования.

В примерах решения задач для описания алгоритмов использован структурный способ (в виде блок-схемы).

1.1 Диалоговый ввод и вывод данных с помощью стандартных функций

Диалоговый ввод (или вывод) означает, что для этих целей используются диалоговые окна, в которых пользователь читает сообщения или указывает данные, которые хочет ввести. В VBA имеются соответствующие разнообразные возможности. Проще всего для этого использовать стандартные функции ***InputBox*** (диалоговый ввод данных) и ***MsgBox*** (диалоговый вывод).

Пример 1. Ввести элемент одномерного массива вещественного типа (объем массива – 10 элементов), номер которого ввел пользователь. Вычислить квадрат этого элемента и вывести результат (с указанием номера этого элемента) в следующем виде:

Квадрат …-го элемента (…): …

Решение.

Прежде, чем писать текст программы, проанализируем условие задачи.

Заметим, что при вводе номера элемента (обозначим его *i*) пользователь может ошибиться, указав недопустимое число (меньше 1 или боль-

ше 10), что приведет к невозможности дальнейших действий. Поэтому необходимо проанализировать полученное от пользователя число и, если оно недопустимо, вывести на экран сообщение об ошибке и прекратить выполнение программы. Эта часть алгоритма представляет собой ветвление (рис. 1):

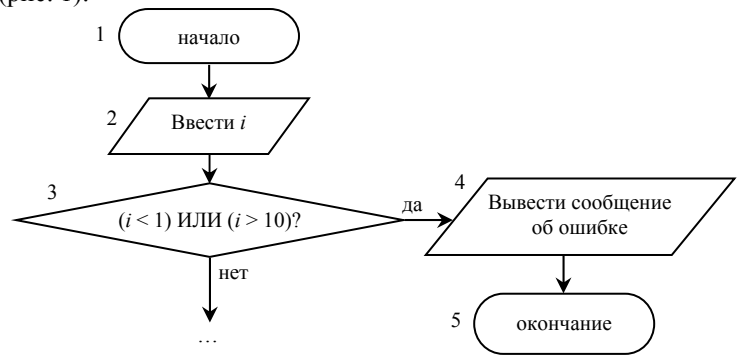

Рис. 1. Проверка корректности введенного числа.

Текст программы:

```
Sub Example_1()                                          ' блок 1
  Dim a(1 To 10) As Single, i As Integer      ' определение переменных
  i = InputBox("Введите номер элемента", "Ввод данных", 1) ' блок 2
  If (i < 1) Or (i > 10) Then                            ' блок 3
    MsgBox "Недопустимый номер элемента", _
           vbOKOnly + vbCritical, "Ошибка"               ' блок 4
    End                                                  ' блок 5
  End If
  a(i) = InputBox("Введите " + CStr(i) + "-й элемент", _
           "Ввод данных", 0)
  MsgBox "Квадрат " + CStr(i) + "-го элемента (" + _
           CStr(a(i)) + "): " + CStr(a(i) ^ 2), _
           vbOKOnly + vbInformation, "Вывод результата"
End Sub
```

Здесь с помощью оператора ***Dim*** описан массив *a* вещественного типа (***As Single***) объемом 10 элементов и простая переменная *i* целого типа (***As Integer***). Затем с помощью функции ***InputBox*** осуществляется ввод номера элемента с присваиванием переменной *i* полученной константы. Ар-

гументами этой функции являются текстовый комментарий, название диалогового окна и константа 1, которая должна быть указана в поле ввода при выводе окна на экран. В результате, во время выполнения программы на экране появится окно ввода данных (рис. 2):

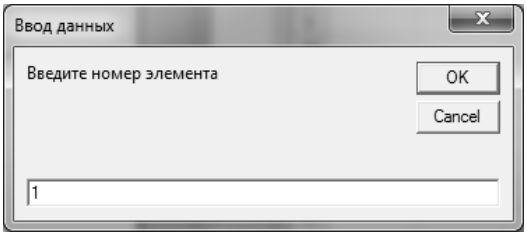

Рис. 2. Диалоговое окно ввода данных.

Пользователь может сразу подтвердить ввод (если его устраивает число 1) или изменить число в поле ввода. В результате, значением функции ***InputBox*** будет константа, указанная в поле ввода.

В следующей части программы производится проверка значения переменной *i* на корректность (с помощью оператора ***If***). Если это значение удовлетворит условию (i < 1) Or (i > 10), то (***Then***) с помощью функции ***MsgBox*** будет выведено на экран сообщение об ошибке (рис. 3):

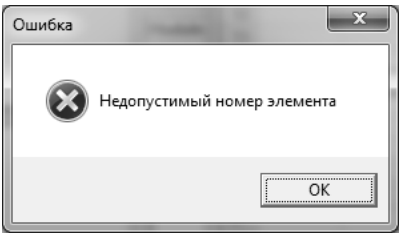

Рис. 3. Сообщение об ошибке ввода числа.

и выполнение программы закончится (оператор ***End***).

Далее следует оператор ввода числа с присваиванием его элементу *a*(*i*) массива *a*. Первый аргумент (текстовый комментарий) функции ***InputBox*** составлен (с помощью операции «конкатенация») из трех частей: строковая константа "*Введите* ", значение переменной *i*, преобразо-

ванное в строковый тип с помощью функции ***CStr***, и строковая константа "-й элемент". В результате, во время выполнения программы, если пользователь введет, например, номер элемента 3, на экране появится окно (рис. 4):

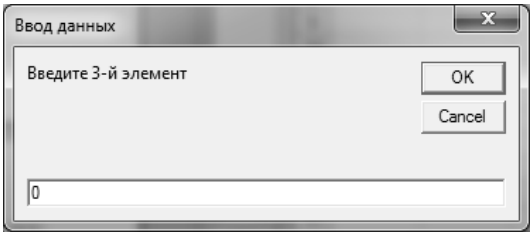

Рис. 4. Конкатенация комментария для окна ввода данных.

Результат вычисления выводится на экран с помощью функции ***MsgBox***. Первый аргумент этой функции, как и в предыдущем случае, также составлен из нескольких частей. При этом, имея в виду, что это аргумент строкового типа, числовые данные преобразовываются в строковые с помощью функции ***CStr***. Заметим, что в качестве одной из частей этого аргумента указано и возведение в квадрат $a(i)$.

В итоге, например, если пользователь введет значение элемента 13,5, то получим (рис. 5):

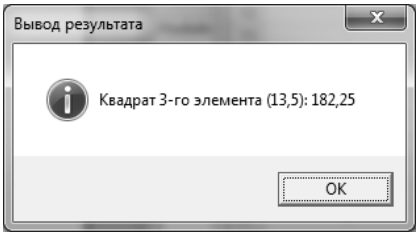

Рис. 5. Окно вывода результатов расчета.

1.2 Цикл

Пример 2. Расчет относительной экономии (перерасхода) материальных ресурсов.

Исходные данные (вводятся пользователем):

n – количество наименований продукции;

r_i – нормативный расход материала на единицу i-й (i = 1,..., n) продукции;

r_{fi} – фактический расход материала на единицу i-й (i = 1,..., n) продукции;

q_i – выпуск i-й (i = 1,..., n) продукции.

Расчет.

Экономия (перерасход) материальных ресурсов (в %) вычисляется по формуле

$$F_r = \frac{R_f - R}{R} \cdot 100 \text{, где}$$

$R_f = \sum_{i=1}^{n} r_{fi} \cdot q_i$ – общий фактический расход материалов;

$R = \sum_{i=1}^{n} r_i \cdot q_i$ – общий нормативный расход материалов.

Вывести полученные результаты (R, R_f, F_r).

Конкретные исходные данные для проверки:

n = 3;

r_1 = 230.6; r_2 = 59.4; r_3 = 145.4;

r_{f1} = 214.3; r_{f2} = 68.5; r_{f3} = 154.6;

q_1 = 154; q_2 = 316; q_3 = 261

Решение.
Блок-схема алгоритма приведена на рис. 6.

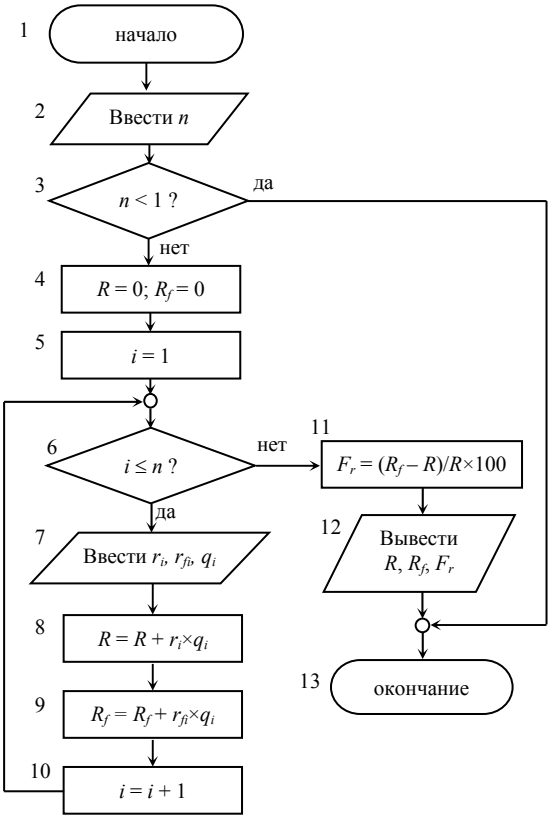

Рис. 6. Блок-схема алгоритма для примера 2.

В этой задаче мы имеем массивы r, r_{fi}, и q_i, причем объемы этих массивов заранее не известны. Поэтому в программе мы сначала объявим их (с помощью ***Dim***) динамическими (без указания количества элементов), а после ввода n доопределим с помощью оператора ***ReDim***.

Текст программы:

```
Sub Example_2()                                          ' блок 1
  Dim r() As Single, rf() As Single, q() As Integer, _
      Rrf As Single, Rr As Single, Fr As Single, _
      n As Integer, prompt As String          ' описание переменных
  n = InputBox("Введите количество наименований продукции", _
           "Расход материалов", 1)                        ' блок 2
  If n < 1 Then End                                      ' блок 3
  ReDim r(1 To n), rf(1 To n), q(1 To n)      ' доопределение массивов
  Rr = 0                                                 ' блок 4
  Rrf = 0
  For i = 1 To n                                      ' блоки 5 и 6
    r(i) = InputBox("Введите нормативный расход для " + _
           CStr(i) + "-й продукции", _                   ' блок 7
           "Расход материала на одну единицу продукции", 0)
    rf(i) = InputBox("Введите фактический расход для " + _
           CStr(i) + "-й продукции", _
           "Расход материала на одну единицу продукции", 0)
    q(i) = InputBox("Введите количество " + CStr(i) + _
           "-й продукции", "Выпуск продукции", 0)
    Rr = Rr + r(i) * q(i)                                ' блок 8
    Rrf = Rrf + rf(i) * q(i)                             ' блок 9
  Next                                                  ' блок 10
  Fr = (Rrf - Rr) / Rr * 100                            ' блок 11
  prompt = "ОБЩИЙ РАСХОД МАТЕРИАЛОВ:" + Chr(13) + _
      "    - нормативный " + CStr(Rr) + Chr(13) + _
      "    - фактический " + CStr(Rrf) + Chr(13) + Chr(13) + _
      "Экономия (перерасход) материальных ресурсов: " + _
      CStr(Fr) + "%"
  MsgBox prompt, vbOKOnly + vbInformation, _            ' блок 12
         "Расход материалов"
End Sub                                                 ' блок 13
```

Обратите внимание, что переменные R и R_f, используемые в описании задачи, в тексте программы переименованы – в *Rr* и *Rrf* соответственно. Это пришлось сделать, т.к. в языке программирования изменение величины знаков не изменяет их смысла (в отличие от обычного языка). Поэтому имена *r* и *R* (*rf* и *Rrf*) считаются эквивалентными, что противоречит

условию нашей задачи. В ней *r* и *rf* – это имена массивов расхода материала для *n* продукций, а *R* и R_f – это имена переменных, имеющих смысл общего расхода материалов. Кроме этого, заметьте, что текстовый комментарий для вывода результатов сформирован в переменной *prompt*, которая затем используется в функции ***MsgBox***. Такой способ позволяет вывести все результаты в одном окне.

Результат выполнения программы представлен на рис. 7:

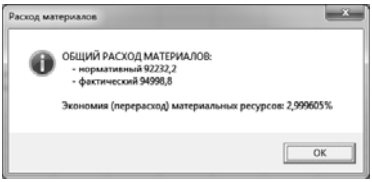

Рис. 7. Результат выполнения программы для примера 2.

Пример 3. Для любой введенной пользователем двумерной таблицы вычислить суммы элементов каждого столбца.

Исходные данные (вводятся пользователем):

n – количество строк,

m – количество столбцов,

a_{ij} – элемент, находящийся на пересечении *i*-й ($i = 1,..., n$) строки и *j*-го ($j = 1,..., m$) столбца.

Расчет.

Сумма элементов *j*-го столбца вычисляется следующим образом

$$S_j = \sum_{i=1}^{n} a_{ij}$$

Вывести полученные результаты (S_j для $j = 1,..., m$).

Конкретные исходные данные для проверки:

$n = 3$; $m = 4$;

$a_{11} = 26$; $a_{12} = 159$; $a_{13} = 376$; $a_{14} = 24$;

$a_{21} = 32$; $a_{22} = 83$; $a_{23} = 348$; $a_{15} = 93$;

$a_{31} = 460$; $a_{32} = 12$; $a_{33} = 18$; $a_{34} = 273$

Решение.

При решении этой задачи мы сделаем так, чтобы пользователь был вынужден корректно ввести *n* (количество строк) и *m* (количество столбцов. Для этого используем операторы цикла с постусловием ***Do*** и ***Loop Until***.

Блок-схема алгоритма приведена на рис. 8.

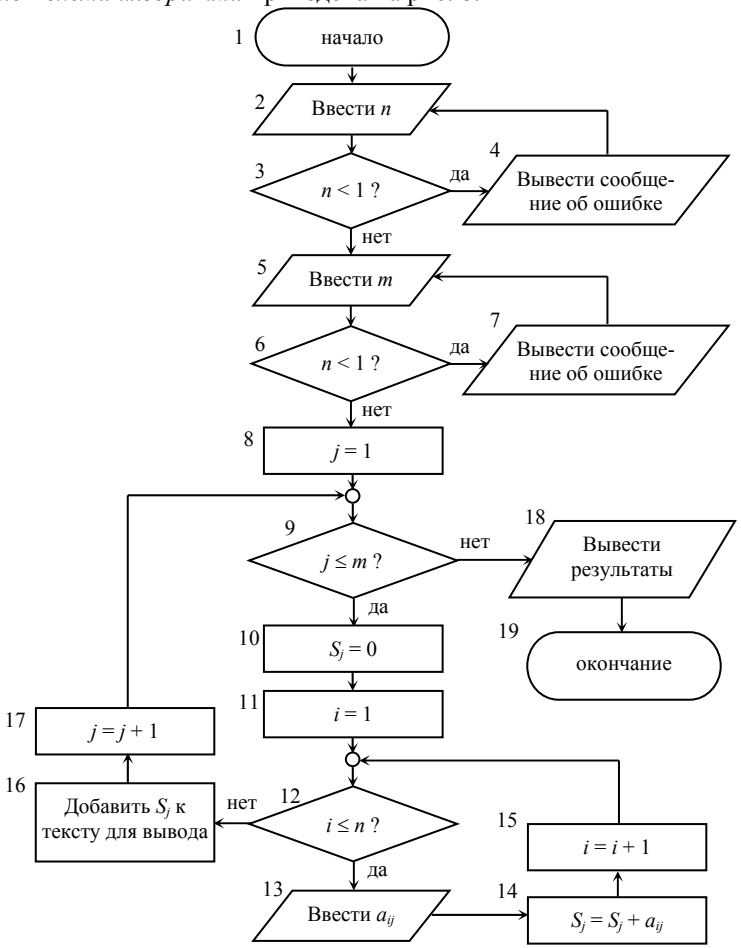

Рис. 8. Блок-схема алгоритма для примера 3

В данном случае мы имеем цикл по *i* (номер строки), вложенный в цикл по *j* (номер столбца). Во внешнем цикле задается номер столбца *j*, начальное значение переменной S_j, которая используется для накопления суммы элементов этого столбца, и формируется текст для вывода результатов. Во внутреннем цикле производится ввод очередного элемента a_{ij} и накопление суммы элементов *j*-го столбца.

Текст программы:

```
Sub Example_3()                                          ' блок 1
    Dim a() As Single, n As Integer, m As Integer, S() As Single, _
        prompt As String                    ' определение переменных
    Do
        n = InputBox("Введите количество строк", _
                "Ввод данных", 1)                        ' блок 2
        If n < 1 Then _                                  ' блок 3
            MsgBox "Недопустимое количество строк", _
                    vbOKOnly + vbCritical, "Ошибка"      ' блок 4
    Loop Until n > 0
    Do
        m = InputBox("Введите количество столбцов", _
                "Ввод данных", 1)                        ' блок5
        If m < 1 Then _                                  ' блок 6
            MsgBox "Недопустимое количество столбцов", _
                    vbOKOnly + vbCritical, "Ошибка"      ' блок 7
    Loop Until m > 0
    ReDim a(1 To n, 1 To m), S(1 To m)        ' доопределение массива
    prompt = "Сумма элементов"
    For j = 1 To m                                       ' блоки 8 и 9
        S(j) = 0                                         ' блок 10
        For i = 1 To n                                   ' блоки 11 и 12
            a(i, j) = InputBox("Введите " + CStr(i) + "-й элемент " + _
                CStr(j) + "-го столбца", "Ввод данных", 0)   ' блок 13
            S(j) = S(j) + a(i, j)                        ' блок 14
        Next                                             ' блок 15
        prompt = prompt + Chr(13) + "   " + CStr(j) + _  ' блок 16
            "-го столбца: " + CStr(S(j))
    Next                                                 ' блок 17
    MsgBox prompt, vbOKOnly + vbInformation, _           ' блок 18
            "Вывод результатов")
```

End Sub ' *блок 19*

Результат выполнения программы представлен на рис. 9:

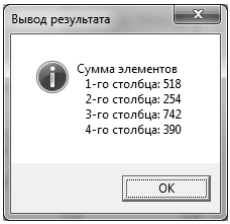

Рис. 9. Результат выполнения программы для примера 3.

1.3 Ветвление

Пример 4. Для любой введенной пользователем двумерной таблицы найти максимальные значения элементов каждой строки и сумму этих значений.

Исходные данные (вводятся пользователем):

n – количество строк,

m – количество столбцов,

a_{ij} – элемент, находящийся на пересечении i-й (i = 1,..., n) строки и j-го (j = 1,..., m) столбца.

Расчет.

Сумма максимальных элементов, находящихся в строках, вычисляется следующим образом

$$S_{max} = \sum_{i=1}^{n} R_i \text{ , где}$$

$R_i = \max\limits_{j=1,...,m} a_{ij}$ – максимальный элемент i-й строки.

Вывести полученные результаты (S_{max} и все R_i для i = 1,..., n).

Конкретные исходные данные для проверки:

n = 3; m = 4;

a_{11} = 172; a_{12} = 159; a_{13} = 176; a_{14} = 324;

a_{21} = 234; a_{22} = 813; a_{23} = 378; a_{15} = 593;

a_{31} = 530; a_{32} = 182; a_{33} = 188; a_{34} = 241

Решение.

Блок-схема алгоритма приведена на рис. 10:

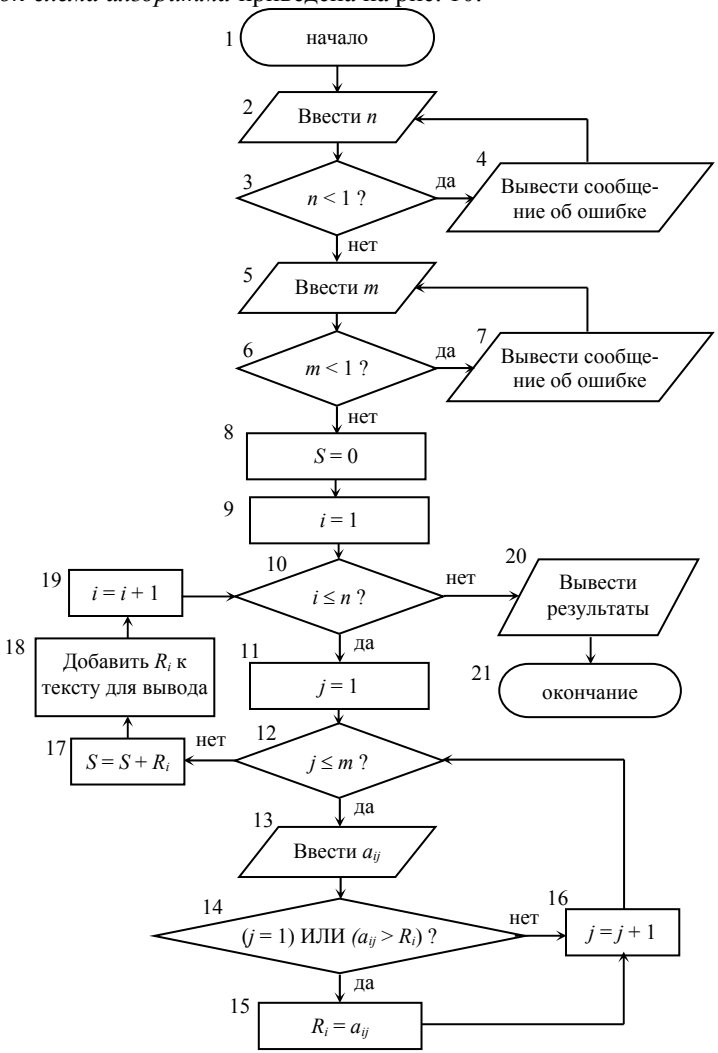

Рис. 10. Блок-схема алгоритма для примера 4.

Текст программы:

```
Sub Example_4()                                          ' блок 1
    Dim a() As Single, n As Integer, m As Integer, prompt As String, _
        R() As Single, S As Single          ' определение переменных
    Do
        n = InputBox("Введите количество строк", _
                "Ввод данных", 1)                        ' блок 2
        If n < 1 Then _                                  ' блок 3
            MsgBox "Недопустимое количество строк", _
                    vbOKOnly + vbCritical, "Ошибка")     ' блок 4
    Loop Until n > 0
    Do
        m = InputBox("Введите количество столбцов", _
                "Ввод данных", 1)                        ' блок5
        If m < 1 Then _                                  ' блок 6
            MsgBox "Недопустимое количество столбцов", _
                    vbOKOnly + vbCritical, "Ошибка"      ' блок 7
    Loop Until m > 0
    ReDim a(1 To n, 1 To m), R(1 To n)       ' доопределение массивов
    prompt = "Максимальные элементы:"
    S = 0                                                ' блок 8
    For i = 1 To n                                       ' блоки 9 и 10
        For j = 1 To m                                   ' блоки 11 и 12
            a(i, j) = InputBox("Введите " + CStr(j) + "-й элемент " + _
                CStr(i) + "-ой строки", "Ввод данных", 0)    ' блок 13
            If (j = 1) Or (a(i, j) > R(i)) _             ' блок 14
                Then R(i) = a(i, j)                      ' блок 15
        Next                                             ' блок 16
        S = S + R(i)                                     ' блок 17
        prompt = prompt + Chr(13) + "   в " + CStr(i) + _     ' блок 18
                "-й строке: " + CStr(R(i))
    Next                                                 ' блок 19
    MsgBox prompt + Chr(13) + " Сумма этих элементов:" + _
            CStr(S), vbOKOnly + vbInformation, _
            "Вывод результатов"                          ' блок 20
End Sub                                                  ' блок 21
```

Результат выполнения программы представлен на рис. 11:

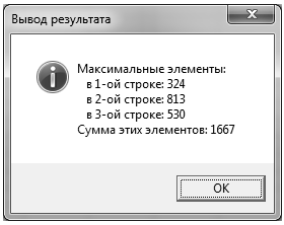

Рис. 11. Результат выполнения программы для примера 4.

Пример 5. В любой введенной пользователем двумерной таблице с равным количеством строк и столбцов найти минимальный элемент среди диагональных, т.е. находящихся на пересечении строк и столбцов с одинаковыми номерами.

Исходные данные (вводятся пользователем):

n – количество строк и столбцов,

a_{ij} – элемент, находящийся на пересечении i-й строки (i = 1,..., n) и j-го (j = 1,..., n) столбца.

Расчет.

Среди всех a_{ii} (i = 1,..., n) необходимо найти элемент A_{min}, удовлетворяющий следующему условию:

$$A_{\min} = \min_{i=1,\ldots,n} a_{ii}$$

Вывести полученный результат (A_{min}) с указанием номеров строки и столбца, на пересечении которых оно находится.

Конкретные исходные данные для проверки:

n = 4;

a_{11} = 72; a_{12} = 15; a_{13} = 16; a_{14} = 23;
a_{21} = 34; a_{22} = 81; a_{23} = 38; a_{15} = 87;
a_{31} = 50; a_{32} = 12; a_{33} = 68; a_{34} = 12
a_{41} = 98; a_{42} = 47; a_{43} = 79; a_{44} = 93

Решение.

Особенность этой задачи в том, что вводятся все элементы таблицы, но используются в расчетах только диагональные. Поэтому во внутрен-

нем цикле (для j-го столбца) после ввода элемента a_{ij} дальнейшие действия производятся, если номер столбца (j) равен номеру строки (i).

Блок-схема алгоритма приведена на рис. 12.

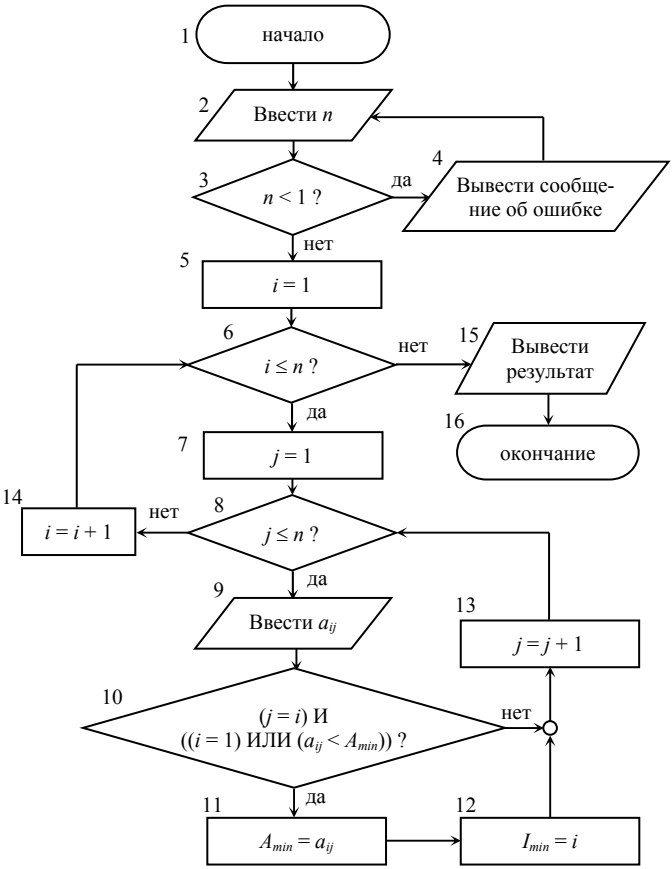

Рис. 12. Блок-схема алгоритма для примера 5.

Текст программы:

```
Sub Example_5()                                    ' блок 1
    Dim a() As Single, n As Integer, _             ' определение переменных
        Amin As Single, Imin As Integer
```

```
    Do
      n = InputBox("Введите количество строк (столбцов)", _
            "Ввод данных", 1)                                    ' блок 2
      If n < 1 Then _                                            ' блок 3
        MsgBox "Недопустимое количество строк (столбцов)", _
                vbOKOnly + vbCritical, "Ошибка"                  ' блок 4
    Loop Until n > 0

    ReDim a(1 To n, 1 To n)                    ' доопределение массивов
    For i = 1 To n                                               ' блок 5 и 6
      For j = 1 To n                                             ' блок 7 и 8
        a(i, j) = InputBox("Введите " + CStr(j) + "-й элемент " + _
            CStr(i) + "-й строки", "Ввод данных", 0)             ' блок 9
        If (j = i) And (i = 1 Or a(i, i) < Amin) Then            ' блок 10
          Amin = a(i, i)                                         ' блок 11
          Imin = i                                               ' блок 12
        End If
    Next j, i                                                    ' блок 13 и 14
MsgBox "Минимальный элемент: " + CStr(Amin) + Chr(13) + _
          "Он находится на пересечении " + CStr(Imin) + _
          "-й строки и " + CStr(Imin) + "-го столбца", _
          vbOKOnly + vbInformation, "Вывод результата"           ' блок 15
End Sub                                                          ' блок 16
```

Обратите внимание, что для окончания двух циклов использован один оператор ***Next*** с переменными *j* и *i*.

Результат выполнения программы представлен на рис. 13:

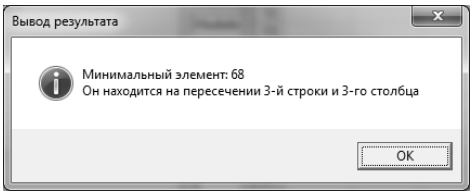

Рис. 13. Результат выполнения программы для примера 5.

1.4 Основы объектно-ориентированного программирования

Пример 6. Написать макрос, который создает новую рабочую книгу, добавляет в нее лист после первого листа с присваиванием нового имени и сохраняет эту книгу в файл с каким-либо именем.

Решение.

Решение этой задачи включает две части:

1) добавление новых элементов в коллекцию ***Workbooks*** (все открытые рабочие книги) и ***Sheets*** (все листы рабочей книги);

2) сохранение рабочей книги в файл.

Для первой части нам понадобится метод ***Add***, который добавляет новый элемент в коллекцию. Сохранение рабочей книги выполняет метод ***SaveAs***, его аргумент ***Filename*** предназначен для задания спецификации файла.

Текст программы:

```
Sub СохранениеМакроса()
    Workbooks.Add.Sheets.Add(, Sheets(1)).Name = "new"
    ActiveWorkbook.SaveAs Filename := "C:\MyBook.xlsx"
End Sub
```

Заметим, что метод ***Add*** для коллекции ***Workbooks*** записан без аргументов, а для ***Sheets*** – с аргументами, помещенными в скобки. В результате, ***Add*** не только выполнит добавление нового элемента, но и возвратит ссылку на этот объект, что позволяет продолжать оператор, применяя действия к этому объекту.

Пример 7. По результатам сдачи экзамена, находящимся в таблице на рабочем листе, вычислить количество студентов, получивших оценки *неуд*, *уд*, *хор* и *отл*.

Диапазон ячеек с результатами сдачи экзамена вводится пользователем в диалоговом окне во время выполнения программы.

Результаты вывести в диалоговом окне.

Конкретные исходные данные для проверки представлены на рис. 14.

	A	B	C
1	**Результаты сдачи экзамена по информатике**		
2			
3	№ п/п	Ф.И.О.	Результат
4	1	Краюхина П.Р.	хор
5	2	Вышнева Ю.Б.	хор
6	3	Кросов П.И.	уд
7	4	Филина К.Е.	неуд
8	5	Мышкин О.Д.	отл
9	6	Артюхова К.О.	отл
10	7	Мальцев В.С.	уд
11	8	Ващеев Л.К.	хор

Рис. 14. Исходная таблица с результатами сдачи экзамена.

Решение.

Решение этой задачи включает три этапа:

– получение от пользователя ссылки на диапазон ячеек, в котором находятся результаты сдачи экзамена,

– вычисление количества студентов, получивших оценки «неуд», «уд», «хор» и «отл»,

– вывод результатов расчета.

Блок-схема алгоритма приведена на рис. 15.

Сначала с помощью диалогового окна ввода получаем от пользователя ссылку на диапазон ячеек, в котором находятся результаты сдачи экзамена. Сразу же проверяем, не отказался ли пользователь от ввода кнопкой ***Отмена***, и, если отказался, выполнение программы прекращается.

Затем подготавливаем переменные для вычисления требуемых количеств. После этого циклически для каждой ячейки диапазона проверяем ее значение и в зависимости от этого производим вычисление соответствующей величины.

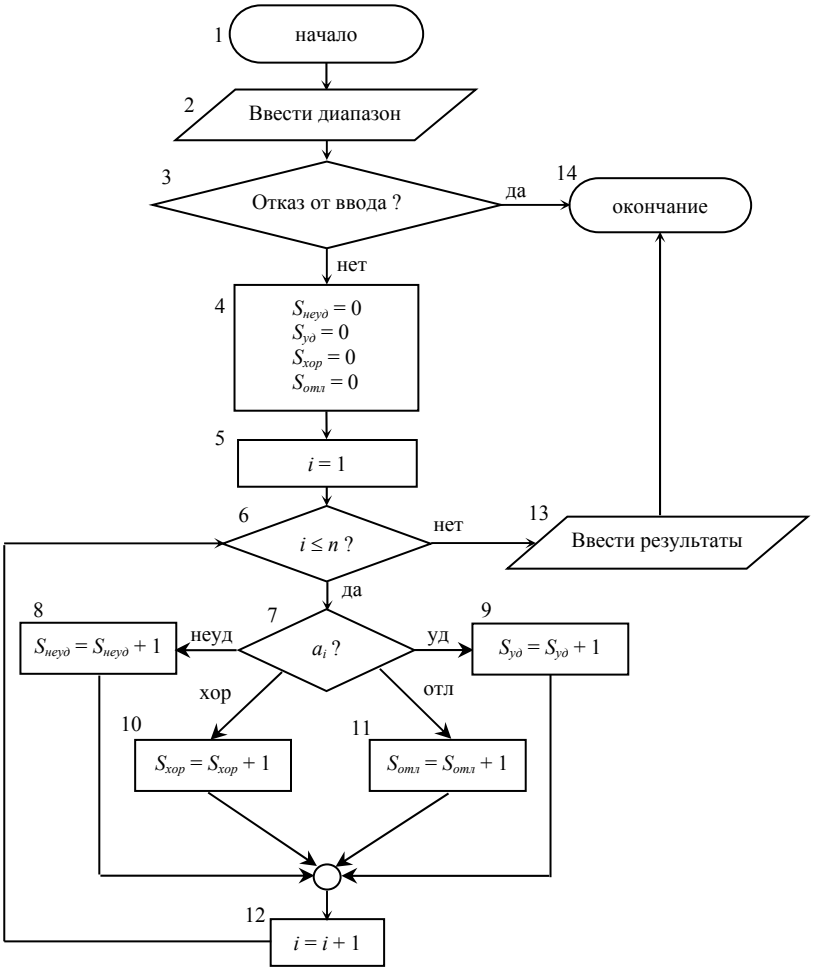

Рис. 15. Блок-схема алгоритма для примера 7.

Текст программы:

```
Sub Example_6()                                    ' блок 1
    Dim UsRange As Range, _
```

```
    Sнеуд As Integer, Sуд As Integer, Sхор As Integer, Sотл As Integer
    On Error Resume Next
    Set UsRange = Application.InputBox( _                    ' блок 2
                  "Укажите диапазон для вычислений", _
                  "Ввод", ActiveCell.Address, Type := 8)
    If UsRange Is Nothing Then End                           ' блок 3
    Sнеуд = 0: Sуд = 0: Sхор = 0: Sотл = 0                   ' блок 4
    For i = 1 To UsRange.Count                          ' блоки 5 и 6
        Select Case UsRange.Cells(i).Value                   ' блок 7
            Case "неуд": Sнеуд = Sнеуд + 1                   ' блок 8
            Case "уд": Sуд = Sуд + 1                         ' блок 9
            Case "хор": Sхор = Sхор + 1                     ' блок 10
            Case "отл": Sотл = Sотл + 1                     ' блок 11
        End Select
    Next                                                     ' блок 12
        MsgBox "Результаты сдачи экзамена:" + Chr(13) + _    ' блок 13
               "   неуд получили " + CStr(Sнеуд) + " студ." + Chr(13) + _
               "   уд    получили " + CStr(Sуд) + " студ." + Chr(13) + _
               "   хор   получили " + CStr(Sхор) + " студ." + Chr(13) + _
               "   отл   получили " + CStr(Sотл) + " студ." + _
               vbOKOnly + vbInformation, "Результаты"
End Sub                                                      ' блок 14
```

Главная особенность этой программы состоит в том, что в ней используется объектная переменная (*UsRange*) типа ***Range*** (область рабочего листа). В связи с этим:

1) для ввода ссылки на диапазон ячеек используется не обычная функция VBA ***InputBox*** (она подходит для данных, а не для объектов), а *метод* ***InputBox***, который применяется к объекту ***Application*** (MS Excel);

2) перед оператором, использующим метод ***InputBox***, помещен оператор *On Error Resume Next*, который обеспечивает игнорирование ошибки пользователя, если он щелкнет по кнопке ***Отмена***;

3) для присваивания объектной переменной ссылки на диапазон используется специальный оператор ***Set***.

Результат выполнения программы представлен на рис. 16:

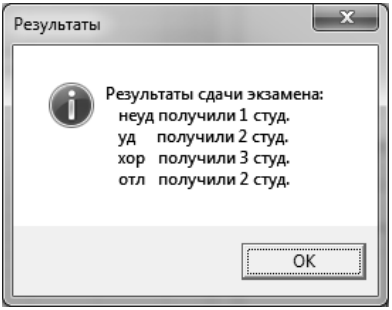

Рис. 16. Результат выполнения программы для примера 7.

Пример 8. Расчет продолжительности ренты при разной величине ренты (400, 600, 800 и 1000) и разном количестве начислений процентов в год (1, 4 и 12).

Исходные данные (вводятся пользователем):
R – член ренты,
j – годовая ставка процентов.

Расчет.
Продолжительность ренты (n) рассчитывается по формуле:

$$n = \frac{-\ln(1 - \frac{A}{R} \cdot ((1 + \frac{j}{m})^m - 1))}{m \cdot \ln(1 + \frac{j}{m})}.$$

Здесь
A – современная величина ренты;
R – член ренты;
j – годовая ставка процентов;
m – количество начислений процентов в год.

Результаты представить в табличной форме на активном рабочем листе активной рабочей книги. Название оформить в соответствии со следующими параметрами шрифта: размер 13 пт, полужирный курсив, цвет синий, двойное подчеркивание. Для отображения значения продолжительности ренты установить ограничение: 2 знака после разделителя целой и дробной части.

Конкретные исходные данные для проверки:
$R = 100$; $j = 0{,}08$.

Решение.

Программа применяется к активному листу активной рабочей книги, поэтому в операторах мы можем их не указывать. Текст будет состоять из трех блоков: первый блок – ввод данных, второй– ввод формулы в ячейку таблицы и копирование ее в остальные ячейки, третий – оформление таблицы (выравнивание и проведение границ).

Первые два блока получим в автоматическом режиме создания текста программы. Для этого с помощью команды ***Разработчик ⇒ Запись макроса*** (кнопка ) запустите этот режим и выполните следующие действия:

1) введите:

– в A1 текст *Расчет продолжительности ренты*
– в A3 текст *Современная величина ренты*
– в B3 текст *Количество начислений процентов в год*
– в B4:D4 числа *1*; *4*; *12*
– в A5:A8 числа *400*; *600*; *800* *1000*
– в B5 формулу

```
=–LN(1-$A5/$B$10*((1+$B$11/B$4)^B$4–1))/(B$4*LN(1+$B$11/B$4))
```

– в A10 текст *Член ренты:*
– в A11 текст *Годовая ставка процентов:*
– в B10 число *100*
– в B11 число *0,08*

2) в диалоговом окне ***Формат ячеек*** оформите название таблицы (в ячейке A1) в соответствии с требуемыми параметрами и установите требуемое ограничение для отображения значения в ячейке B5;

3) автозаполнением скопируйте формулу из B5 в B5:D5, а затем – из B5:D5 в B5:D8;

4) остановите режим автоматического создания макроса командой ***Разработчик ⇒ Остановить запись*** (кнопка).

В результате, на рабочем листе получим следующую таблицу (рис. 17):

	A	B	C	D	E
1	***Расчет продолжительности ренты***				
2					
3	Современ	Количество начислений процентов в год			
4		1	4	12	
5	400	5,01	5,05	5,06	
6	600	8,50	8,61	8,64	
7	800	13,27	13,60	13,68	
8	1000	20,91	21,96	22,22	
9					
10	Член рент	100			
11	Годовая с	0,08			

Рис. 17. Таблица, полученная в результате автоматического создания макроса.

Перейдите в окно VBA и в автоматически созданном модульном листе и увидите полученный текст программы. Посмотрите, какой он огромный. На самом деле, такой большой текст не нужен. Его можно сократить в несколько раз.

Первая пара операторов

```
Range("A1").Select
ActiveCell.FormulaR1C1 = "Расчет продолжительности ренты"
```

появилась в результате ввода названия таблицы в ячейку *A1*. Мы были вынуждены сначала выделить ячейку, а затем ввести текст. Поэтому в первом операторе к объекту, на который указывает ***Range***, применен метод ***Select*** (выделить). Во втором операторе для объекта ***ActiveCell*** (активная ячейка) свойству ***FormulaR1C1*** (содержимому в формате R1C1) присваивается текстовая константа. Здесь можно обойтись без Select…ActiveCell и записать

```
Range("A1").FormulaR1C1 = "Расчет продолжительности ренты"
```

Но и это еще не все. Поскольку в подобном случае для ***Range*** предусматривается умолчание свойства, можно просто присвоить ему текстовую константу. В итоге, получаем

```
Range("A1") = "Расчет продолжительности ренты"
```

Аналогично можно сократить и остальные операторы, посвященные вводу данных. Кроме этого, некоторые из них удобнее записать в одной строке. Таким образом, получаем

```
Range("A1") = "Расчет продолжительности ренты"
Range("A3") = "Современная величина ренты "
Range("B3") = "Количество начислений процентов в год"
Range("B4") = 1:        Range("C4") = 4:        Range("D4") = 12
```

```
    Range("A5") = 400:    Range("A6") = 600:    Range("A7") = 800
    Range("A8") = 1000
    Range("A10") = "Член ренты:"
    Range("B10") = 100
    Range("A11") = "Годовая ставка процентов"
    Range("B11") = 0.08
```

В следующей части программы производятся изменения свойств шрифта (объект ***Font***) для содержимого ячейки *A1* (объект Range("A1")). Параметры шрифта производились во вкладке ***Шрифт*** диалогового окна ***Формат ячеек***, поэтому в программе присутствуют операторы для всех свойств, в том числе и для тех, которые мы не изменяли. Эти операторы можно удалить. В результате останется

```
    With Range("A1").Font
        .FontStyle = "полужирный курсив"
        .Size = 13
        .Underline = xlUnderlineStyleDouble
        .Color = 12611584
    End With
```

Далее следуют операторы выполнения действий с формулами (ввод, изменение формата отображения числа, копирование):

```
Range("B5").Select
ActiveCell.FormulaR1C1 =
  "=-LN(1-RC1/R10C2*((1+R11C2/R4C)^R4C-1))/(R4C*LN(1+R11C2/R4C))"
Selection.NumberFormat = "0.00"
Selection.AutoFill Destination := Range("B5:D5"), Type:=xlFillDefault
Range("B5:D5").Select
Selection.AutoFill Destination := Range("B5:D8"), Type:=xlFillDefault
Range("B5:D8").Select
```

Часть текста, посвященную ячейке *B5*, переоформляем с применением конструкции ***With–End With*** (текст станет более ясным), для метода ***AutoFill*** удаляем аргумент ***Type*** (его значение установлено по умолчанию), в предпоследней паре операторов удаляем часть Select… Selection и, наконец, удаляем последний оператор. В результате, получаем

```
With Range("B5")
    .FormulaR1C1 = _
        "=-LN(1-RC1/R10C2*((1+R11C2/R4C)^R4C-1))/(R4C*LN(1+R11C2/R4C))"
    .NumberFormat = "0.00"
```

```
    .AutoFill Destination := Range("B5:D5")
End With
Range("B5:D5").AutoFill Destination := Range("B5:D8")
```

В итоге, наш текст программы приобрел следующий вид:

```
Sub Макрос1()
    Range("A1") = "Расчет продолжительности ренты"
    Range("A3") = "Современная величина ренты "
    Range("B3") = "Количество начислений процентов в год"
    Range("B4") = 1:      Range("C4") = 4:      Range("D4") = 12
    Range("A5") = 400:  Range("A6") = 600:  Range("A7") = 800
    Range("A8") = 1000
    Range("A10") = "Член ренты:"
    Range("B10") = 100
    Range("A11") = "Годовая ставка процентов"
    Range("B11") = 0.08
    With Range("A1").Font
        .FontStyle = "полужирный курсив"
        .Size = 13
        .Underline = xlUnderlineStyleDouble
        .Color = 12611584
    End With
    With Range("B5")
        .FormulaR1C1 = _
          "=-LN(1-RC1/R10C2*((1+R11C2/R4C)^R4C-1))/(R4C*LN(1+R11C2/R4C))"
        .NumberFormat = "0.00"
        .AutoFill Destination:=Range("B5:D5")
    End With
    Range("B5:D5").AutoFill Destination:=Range("B5:D8")
End Sub
```

Запустите на новом рабочем листе полученную программу и убедитесь, что она работает правильно.

Остальные операторы (ввод исходных данных, объединение ячеек, выравнивание, проведение границ) мы добавим в текст программы уже в редакторе VBE.

Исходными данными у нас являются два параметра: член ренты и годовая ставка процентов. Сделаем так, чтобы эти данные вводились в диалоговых окнах, а не указывались вручную на рабочем листе. Для этого в нашей программе заменим операторы

```
    Range("B10") = 100
```
и
```
    Range("B11") = 0.08
```
на операторы соответственно
```
    Range("B10") = _
        CInt(InputBox("Введите член ренты", "Ввод данных"))
    Range("B11") = _
        CDbl(InputBox("Введите годовую ставку процентов", _
                      "Ввод данных"))
```

Здесь функции ***CInt*** и ***CDbl*** понадобились для преобразования текстовых данных в числовые, ведь функция ***InputBox*** возвращает введенное данное в текстовом формате, а необходимые нам данные – числовые (член ренты – целое число, а годовая ставка процентов – вещественное).

Название таблицы и общее название для второго, третьего и четвертого столбцов установим по центру с объединением соответствующих диапазонов ячеек, в остальных ячейках – по центру:

```
    Range("A1:D1,B3:D3").HorizontalAlignment = xlCenterAcrossSelection
    Range("A3:A8,B4:D8").HorizontalAlignment = xlCenter
```

Название первого столбца установим в объединенной области *A3*:*A4* (с помощью метода ***Merge***) с вертикальным выравниванием по центру (с помощью свойства ***VerticalAlignment***):

```
        With Range("A3:A4")
            .Merge
            .VerticalAlignment = xlCenter
        End With
```

Далее добавим операторы изменения размера столбцов: для столбца *A* применим метод ***AutoFit*** (автоподгонка), а для диапазона столбцов *B*:*D* с помощью свойства ***ColumnWidth*** установим размер 12 знаков:

```
        Columns("A").AutoFit
        Columns("B:D").ColumnWidth = 12
```

Осталось провести границы между ячейками внутри таблицы и обрамление вокруг нее:

```
        With Range("A5:A8,B4:D8")
            .Borders(xlLeft).Weight = xlThin
            .Borders(xlTop).Weight = xlThin
        End With
        Range("A3:A4,A5:A8,B3:D4,B5:D8").BorderAround _
            Weight:=xlMedium
```

В итоге получаем следующий текст программы:

```
Sub Макрос1()
    Range("A1") = "Расчет продолжительности ренты"
    Range("A3") = "Современная величина ренты "
    Range("B3") = "Количество начислений процентов в год"
    Range("B4") = 1:      Range("C4") = 4:      Range("D4") = 12
    Range("A5") = 400:  Range("A6") = 600:  Range("A7") = 800
    Range("A8") = 1000
    Range("A10") = "Член ренты:"
    Range("B10") = _
        CInt(InputBox("Введите член ренты", "Ввод данных"))
    Range("A11") = "Годовая ставка процентов"
    Range("B11") = _
        CDbl(InputBox("Введите годовую ставку процентов", _
                      "Ввод данных"))
    With Range("A1").Font
          .FontStyle = "полужирный курсив"
          .Size = 13
          .Underline = xlUnderlineStyleDouble
          .Color = 12611584
    End With
    With Range("B5")
        .FormulaR1C1 = _
          "=-LN(1-RC1/R10C2*((1+R11C2/R4C)^R4C-1))/(R4C*LN(1+R11C2/R4C))"
        .NumberFormat = "0.00"
        .AutoFill Destination:=Range("B5:D5")
    End With
    Range("B5:D5").AutoFill Destination:=Range("B5:D8")
    Range("A1:D1,B3:D3").HorizontalAlignment = xlCenterAcrossSelection
    Range("A3:A8,B4:D8").HorizontalAlignment = xlCenter
    With Range("A3:A4")
        .Merge
        .VerticalAlignment = xlCenter
    End With
    Columns("A").AutoFit
    Columns("B:D").ColumnWidth = 12
    With Range("A5:A8,B4:D8")
        .Borders(xlLeft).Weight = xlThin
        .Borders(xlTop).Weight = xlThin
```

```
    End With
    Range("A3:A4,A5:A8,B3:D4,B5:D8").BorderAround _
          Weight := xlMedium
End Sub
```

Запустите программу на выполнение, введите исходные данные и, в результате, получите следующую таблицу (рис. 18):

	A	B	C	D
1	***Расчет продолжительности ренты***			
2				
3	Современная величина ренты	Количество начислений процентов в год		
4		1	4	12
5	400	5,01	5,05	5,06
6	600	8,50	8,61	8,64
7	800	13,27	13,60	13,68
8	1000	20,91	21,96	22,22
9				
10	Член ренты	100		
11	Годовая ставка процентов	0,08		

Рис. 18. Результат выполнения программы для примера 8.

2 КОНТРОЛЬНЫЕ ЗАДАЧИ

2.1 Цикл

Задача 1. Расчет темпа изменения объема товарной продукции (работ, услуг).

Исходные данные (вводятся пользователем):

n – количество наименований продукции;

q_{0i}, q_{1i} – количество произведенной i-й (i = 1,..., n) продукции (работ, услуг) соответственно в предыдущем и анализируемом периоде;

p_{1i} – цена единицы i-й (i = 1,..., n) продукции (работ, услуг) в анализируемом периоде.

Расчет.

Темп изменения объема товарной продукции (работ, услуг) вычисляется следующем образом (в %)

$$T = \frac{F_1}{F_{01}} \cdot 100 \text{ , где}$$

$F_{01} = \sum_{i=1}^{n} q_{0i} \cdot p_{1i}$ – общая стоимость продукции (работ услуг) в предыдущем периоде в ценах анализируемого периода;

$F_1 = \sum_{i=1}^{n} q_{1i} \cdot p_{1i}$ – общая стоимость продукции (работ, услуг) в анализируемом периоде.

Вывести полученные результаты (F_{01}, F_1, T).

Конкретные исходные данные для проверки:

$n = 4$;

$q_{01} = 459$; $q_{02} = 764$; $q_{03} = 68$; $q_{04} = 573$;

$q_{11} = 562$; $q_{12} = 328$; $q_{13} = 80$; $q_{14} = 724$;

$p_{11} = 2080.5$; $p_{12} = 931.4$; $p_{13} = 4271.8$; $p_{14} = 1261.6$

Задача 2. Расчет относительного изменения нормативного расхода материальных ресурсов.

Исходные данные (вводятся пользователем):

n – количество наименований продукции;

r_{0i}, r_{1i} – нормативный расход материала на единицу i-й (i = 1,..., n) продукции (в натуральных единицах) соответственно в предыдущем и анализируемом периодах;

q_{1i} – выпуск i-й (i = 1,..., n) продукции (в натуральных единицах) в анализируемом периоде.

Расчет.

Относительное изменение нормативного расхода материалов вычисляется по формуле (в %)

$$P_r = \frac{R_1 - R_0}{R_0} \cdot 100 \text{ , где}$$

$R_0 = \sum_{i=1}^{n} r_{0i} \cdot q_{1i}$ – расход материалов на весь выпуск продукции в анализируемом периоде по нормам предыдущего периода;

$R_1 = \sum_{i=1}^{n} r_{1i} \cdot q_{1i}$ – расход материалов на весь выпуск продукции в анализируемом периоде.

Вывести полученные результаты (R_0, R_1, P_r).

Конкретные исходные данные для проверки:

$n = 3$;
$r_{01} = 84.6$; $r_{02} = 124.7$; $r_{03} = 53.8$;
$r_{11} = 91.7$; $r_{12} = 107.9$; $r_{13} = 61.3$;
$q_{11} = 137$; $q_{12} = 261$; $q_{13} = 84$

Задача 3. Оценка влияния изменения расхода материалов на изменение удельной материалоемкости изделия.

Исходные данные (вводятся пользователем):

P_1 – цена всего изделия в анализируемом периоде;

n – количество групп материалов, используемых для выпуска изделия;

q_{0i}, q_{1i} – количество материалов i-й ($i = 1,..., n$) группы соответственно в предыдущем и анализируемом периоде (в натуральных единицах);

p_{1i} – цена единицы материала i-й ($i = 1,..., n$) группы в анализируемом периоде.

Расчет.

Влияние изменения расхода материалов на изменение удельной материалоемкости изделия оценивается следующим образом

$$D_q = \frac{Z_1 - Z_{01}}{P_1}, \text{ где}$$

$Z_{01} = \sum_{i=1}^{n} q_{0i} \cdot p_{1i}$ – материальные затраты в себестоимости изделия в предыдущем периоде в ценах анализируемого периода;

$Z_1 = \sum_{i=1}^{n} q_{1i} \cdot p_{1i}$ – материальные затраты в себестоимости изделия в анализируемом периоде.

Вывести полученные результаты (Z_{01}, Z_1, D_q).

Конкретные исходные данные для проверки:

$n = 3$; $P_1 = 2452.1$

$q_{01} = 138$; $q_{02} = 84$; $q_{03} = 265$;

$q_{11} = 183$; $q_{12} = 76$; $q_{13} = 242$;

$p_{11} = 1.8$; $p_{12} = 7.5$; $p_{13} = 3.6$

Задача 4. Оценка относительного изменения затрат на заработную плату в себестоимости изделия.

Исходные данные (вводятся пользователем):

n – количество операций изготовления изделия;

q_{0i}, q_{1i} – годовая тарифная ставка i-й ($i = 1,..., n$) операции соответственно в предыдущем и анализируемом периодах;

t_{0i}, t_{1i} – трудоемкость (час.) i-й ($i = 1,..., n$) операции соответственно в предыдущем и анализируемом периодах.

Расчет.

Относительное изменение затрат на заработную плату в себестоимости изделия оценивается следующим образом (в %)

$$dZ_{отн} = \frac{Z_1 - Z_0}{Z_0} \cdot 100 \text{ , где}$$

$Z_0 = \sum_{i=1}^{n} q_{0i} \cdot t_{0i}$ – затраты на заработную плату в себестоимости изделия в предыдущем периоде;

$Z_1 = \sum_{i=1}^{n} q_{1i} \cdot t_{1i}$ – затраты на заработную плату в себестоимости изделия в анализируемом периоде.

Вывести полученные результаты (Z_0, Z_1, $dZ_{отн}$).

Конкретные исходные данные для проверки:

$n = 3$;

$t_{01} = 6.8$; $t_{02} = 11.4$; $t_{03} = 9.5$;

$t_{11} = 7.2$; $t_{12} = 9.3$; $t_{13} = 7.8$;

$q_{01} = 351.4$; $q_{02} = 217.6$; $q_{03} = 287.5$;

$q_{11} = 286.9$; $q_{12} = 223.5$; $q_{13} = 269.8$

Задача 5. Оценка влияния изменения тарифных ставок на изменение затрат на заработную плату в себестоимости изделия.

Исходные данные (вводятся пользователем):

n – количество операций изготовления изделия;

q_{0i}, q_{1i} – годовая тарифная ставка i-й (i = 1,..., n) операции соответственно в предыдущем и анализируемом периодах;

t_{1i} – трудоемкость (час.) i-й (i = 1,..., n) операции в анализируемом периоде.

Расчет.

Влияние изменения тарифных ставок на изменение затрат на заработную плату в себестоимости изделия оценивается следующим образом

$$P_q = \frac{Z_1}{Z_{01}}, \text{ где}$$

$Z_1 = \sum_{i=1}^{n} q_{1i} \cdot t_{1i}$ – затраты на заработную плату в себестоимости изделия в анализируемом периоде;

$Z_{01} = \sum_{i=1}^{n} q_{0i} \cdot t_{1i}$ – затраты на заработную плату в себестоимости изделия в предыдущем периоде, исходя из часовых тарифных ставок анализируемого периода.

Вывести полученные результаты (Z_1, Z_{01}, P_q).

Конкретные исходные данные для проверки:

n = 3;

q_{01} = 265.3; q_{02} = 97.5; q_{03} = 135.4;

q_{11} = 234.1; q_{12} = 118.5; q_{13} = 141.6

t_{11} = 9.7; t_{12} = 12.4; t_{13} = 6.8;

Задача 6. Оценка изменения объема товарооборота и влияния изменения цен товаров на изменение объема товарооборота.

Исходные данные (вводятся пользователем):

n – количество групп товаров;

t_{0i}, t_{1i} – объем товарооборота i-й группы товаров (i = 1,..., n) соответственно в предыдущем и анализируемом периодах;

p_{0i}, p_{1i} – цена единицы товара i-й группы (i = 1,..., n) соответственно в предыдущем и анализируемом периодах.

Расчет.

Изменение объема товарооборота оценивается индексом объема товарооборота, вычисляемого по формуле

$$J_{tp} = \frac{T_1}{T_0}, \text{ где}$$

$T_0 = \sum_{i=1}^{n} t_{0i}$ – объем товарооборота в предыдущем периоде;

$T_1 = \sum_{i=1}^{n} t_{1i}$ – объем товарооборота в анализируемом периоде.

Влияние изменения цен товаров на изменение объема товарооборота оценивается ценовым индексом объема товарооборота

$$J_p = \frac{T_1}{T_{10}}, \text{ где}$$

$T_{10} = \sum_{i=1}^{n} \frac{t_{1i}}{r_i}$ – объем товарооборота в анализируемом периоде в ценах предыдущего периода;

$r_i = \frac{p_{1i}}{p_{0i}}$ – индекс цены i-й группы товара.

Вывести полученные результаты (T_0, T_1, T_{10}, J_{tp}, J_p).

Конкретные исходные данные для проверки:

$n = 5$;

$t_{01} = 8564.3$; $t_{02} = 24384.5$; $t_{03} = 17562.4$; $t_{04} = 354.7$; $t_{05} = 15458.4$;

$t_{11} = 12835.4$; $t_{12} = 25856.7$; $t_{13} = 18124.2$; $t_{14} = 425.6$; $t_{15} = 17681.3$;

$p_{01} = 43.5$; $p_{02} = 168.3$; $p_{03} = 361.2$; $p_{04} = 15.6$; $p_{05} = 261.5$;

$p_{11} = 51.5$; $p_{12} = 215.4$; $p_{13} = 415.2$; $p_{14} = 18.4$; $p_{15} = 165.2$

Задача 7. Вычисление суммарного темпа инфляции для нескольких последовательных периодов.

Исходные данные (вводятся пользователем):

n – количество периодов;

a_i – темп инфляции в i-ом периоде (i = 1,..., n).

Расчет.

Суммарный темп инфляции для нескольких последовательных периодов вычисляются следующим образом

$$A = \prod_{i=1}^{n} (1 + a_i) - 1$$

Вывести полученный результат вычисления суммарного темпа инфляции.

Конкретные исходные данные для проверки:

$n = 6$;

$a_1 = 0.08$; $a_2 = 0.06$; $a_3 = 0.04$; $a_4 = 0.07$; $a_5 = 0.07$; $a_6 = 0.09$

Задача 8. Вычисление наращенной суммы долга с начислением простых процентов при переменной процентной ставке.

Исходные данные (вводятся пользователем):

D – размер кредита;

m – количество периодов;

n_i – длительность i-го ($i = 1,..., m$) периода (годы);

p_i – процентная ставка в течение i-го периода ($i = 1,..., m$).

Расчет.

Наращенная сумма долга с начислением простых процентов при переменной процентной ставке вычисляется по формуле

$$S = D \cdot (1 + \sum_{i=1}^{m} n_i \cdot p_i)$$

Вывести полученный результат вычисления наращенной суммы долга.

Конкретные исходные данные для проверки:

$m = 4$; $D = 150000$;

$n_1 = 0.5$; $n_2 = 1$; $n_3 = 2$; $n_4 = 2.5$;

$p_1 = 0.4$; $p_2 = 0.3$; $p_3 = 0.15$; $p_4 = 0.2$

Задача 9. Вычисление наращенной суммы долга с начислением простых процентов при процентной ставке, изменяющейся на фиксированную величину.

Исходные данные (вводятся пользователем):

m – количество платежей;

D – размер кредита;

t_1 – длительность первого периода (годы);

t – длительность i-го ($i = 2,..., m$) периода (годы);

p_1 – процентная ставка в течение первого периода;

d_p – величина изменения процентной ставки.

Расчет.

Наращенная сумма долга с начислением простых процентов при процентной ставке, изменяющейся на фиксированную величину, вычисляется по формуле

$$S = D \cdot (1 + \sum_{i=1}^{m} n_i \cdot p_i), \text{ где}$$

$n_1 = t_1$;

$n_i = t$, $p_i = p_{i-1} + d_p$ для $i = 2,\ldots, m$

Вывести полученный результат вычисления наращенной суммы долга.

Конкретные исходные данные для проверки:

$m = 5$; $D = 200000$;

$t_1 = 1$; $p_1 = 0.2$;

$t = 0.5$; $d_p = 0.01$

Задача 10. Начисление простых процентов при изменении суммы вклада и переменной процентной ставке.

Исходные данные (вводятся пользователем):

m – количество изменений остатков средств на счете;

D_i – остаток средств на счете после очередного i-го ($i = 1,..., m$) поступления или снятия;

n_i – срок хранения i-й суммы ($i = 1,..., m$) до нового поступления (снятия) средств на счет (годы);

p_i – процентная ставка i-го ($i = 1,..., m$) периода.

Расчет.

Сумма, полученная владельцем счета, с начислением простых процентов при изменении суммы вклада и переменной процентной ставке вычисляется по формуле

$$S = D_m + \sum_{i=1}^{m} (D_i \cdot n_i \cdot p_i)$$

Вывести полученный результат вычисления суммы, полученной владельцем счета.

Конкретные исходные данные для проверки:

$m = 4$;

$D_1 = 8000$; $D_2 = 1000$; $D_3 = 5000$; $D_4 = 10000$;

$n_1 = 0.1$; $n_2 = 0.5$; $n_3 = 1$; $n_4 = 0.75$;

$p_1 = 0.2$; $p_2 = 0.1$; $p_3 = 0.13$; $p_4 = 0.15$.

Задача 11. Вычисление наращенной суммы долга с начислением сложных процентов при переменной процентной ставке.

Исходные данные (вводятся пользователем):

D – размер кредита;

m – количество периодов;

n_i – длительность i-го периода ($i = 1,..., m$) начисления процентов (годы);

p_i – процентная ставка в течение i-го периода ($i = 1,..., m$).

Расчет.

Наращенная сумма долга с начислением сложных процентов при переменной процентной ставке вычисляется по формуле

$$S = D \cdot \prod_{i=1}^{m} (1 + p_i)^{n_i}$$

Вывести полученный результат вычисления наращенной суммы долга.

Конкретные исходные данные для проверки:

$m = 4$; $D = 350000$;

$n_1 = 0.5$; $n_2 = 1$; $n_3 = 2$; $n_4 = 1.5$;

$p_1 = 0.2$; $p_2 = 0.15$; $p_3 = 0.12$; $p_4 = 0.17$

Задача 12. Вычисление средней ставки простых процентов нескольких операций с разными суммами ссуд и процентными ставками.

Исходные данные (вводятся пользователем):

m – количество операций;

D_i – размер i-й ссуды ($i = 1,..., m$);

p_i – процентная ставка для i-й ссуды ($i = 1,..., m$).

Расчет.

Средняя ставка простых процентов нескольких операций с разными суммами ссуд и процентными ставками вычисляется по формуле

$$P = \frac{\sum_{i=1}^{m} D_i \cdot p_i}{\sum_{j=1}^{m} D_j}$$

Вывести полученный результат вычисления средней ставки.

Конкретные исходные данные для проверки:

$m = 4$;

$D_1 = 20500$; $D_2 = 15000$; $D_3 = 25000$; $D_4 = 10000$;

$p_1 = 0.15$; $p_2 = 0.2$; $p_3 = 0.14$; $p_4 = 0.25$

Задача 13. Вычисление среднего взвешенного срока консолидированного платежа.

Исходные данные (вводятся пользователем):

m – количество платежей;

n_i – период i-го ($i = 1,..., m$) платежа (годы);

Q_i – размер i-го платежа ($i = 1,..., m$).

Расчет.

Вычисление среднего взвешенного срока консолидированного платежа производится по формуле

$$N = \frac{\sum_{j=1}^{m} Q_j \cdot n_j}{\sum_{k=1}^{m} Q_k}$$

Вывести полученный результат вычисления среднего взвешенного срока консолидированного платежа.

Конкретные исходные данные для проверки:

$m = 5$;

$n_1 = 0.2$; $n_2 = 0.12$; $n_3 = 0.17$; $n_4 = 0.15$; $n_5 = 0.3$;

$Q_1 = 18350$; $Q_2 = 25545$; $Q_3 = 46000$; $Q_4 = 12600$; $Q_5 = 25000$

Задача 14. Определение срока консолидированного платежа с применением простых процентов.

Исходные данные (вводятся пользователем):

S – размер консолидированного платежа;

p – процентная ставка;

m – количество платежей;

D_i – размер i-го платежа ($i = 1,..., m$);

n_i – длительность i-го периода ($i = 1,..., m$) платежа (годы).

Расчет.

Срок консолидированного платежа с применением простых процентов вычисляется по формуле

$$T = \frac{1}{p} \cdot \left(\frac{S}{\sum_{j=1}^{m} D_j \cdot (1 + n_j \cdot p)^{-1}} - 1 \right)$$

Вывести полученный результат вычисления среднего взвешенного срока консолидированного платежа.

Конкретные исходные данные для проверки:

$S = 350000$; $p = 0.23$ $m = 4$;

$n_1 = 0.25$; $n_2 = 0.5$; $n_3 = 0.25$; $n_4 = 0.75$;

$D_1 = 45600$; $D_2 = 82350$; $D_3 = 36580$; $D_4 = 120600$

Задача 15. Вычисление суммы средних объемов запасов в группе складов.

Исходные данные (вводятся пользователем):

m – количество складов;

n_i – количество товаров, хранящихся в i-м ($i = 1,..., m$) складе;

V_{ij} – объем запасов j-го ($j = 1,..., n_i$) товара, хранящегося в i-м ($i = 1,..., m$) складе.

Расчет.

Вычисление суммы средних объемов запасов в группе складов вычисляется по формуле

$$S_V = \sum_{i=1}^{m} s_i \text{ , где}$$

$s_i = \frac{1}{n_i}\sum_{j=1}^{n_i} V_{ij}$ – средний объем запасов по группе товаров, хранящихся в i-м ($i = 1,..., m$) складе.

Вывести полученные результаты (s_i для $i = 1,..., m$ и S_V).

Конкретные исходные данные для проверки:

$m = 3$;
$n_1 = 5$; $n_2 = 3$; $n_3 = 4$;
$V_{11} = 126$; $V_{12} = 59$; $V_{13} = 76$; $V_{14} = 294$; $V_{15} = 193$;
$V_{21} = 325$; $V_{22} = 283$; $V_{23} = 38$;
$V_{31} = 46$; $V_{32} = 312$; $V_{33} = 184$; $V_{34} = 73$

2.2 Ветвление

Задача 16. Вычисление количества групп продукции с убыточно проданными изделиями.

Исходные данные (вводятся пользователем):

n – количество групп продукции;
m – количество изделий в каждой группе;
S_{ij} – прибыль от продажи j-го ($j = 1,..., m$) изделия (положительная, если цена продажи больше себестоимости, и отрицательная в противном случае) i-й ($i = 1,..., n$) группы.

Расчет.

Количество групп продукции с убыточно проданными изделиями вычисляется по формуле

$$R = \sum_{i=1}^{n} \delta_i \text{ , где}$$

$$\delta_i = \begin{cases} 1, \text{ если среди } S_{ij} \text{ (для } j = 1,...,m \text{) есть отрицательные} \\ 0, \text{ если среди } S_{ij} \text{ (для } j = 1,...,m \text{) нет отрицательных} \end{cases}$$

Вывести полученный результат (R).

Конкретные исходные данные для проверки:

$n = 5$; $m = 4$;
$S_{11} = 342$; $S_{12} = 290$; $S_{13} = 39$; $S_{14} = 167$;
$S_{21} = 126$; $S_{22} = 653$; $S_{23} = -45$; $S_{24} = 324$;
$S_{31} = 24$; $S_{32} = 94$; $S_{33} = 191$; $S_{34} = 64$;
$S_{41} = -12$; $S_{42} = 284$; $S_{43} = 275$; $S_{44} = -95$;
$S_{51} = 383$; $S_{52} = 194$; $S_{53} = 63$; $S_{54} = 278$

Задача 17. Вычисление количества групп продукции с безубыточно проданными изделиями.

Исходные данные (вводятся пользователем):

m – количество групп продукции;

n – количество изделий в каждой группе;

V_{ij} – прибыль от продажи j-го (j = 1,..., n) изделия (положительная, если цена продажи больше себестоимости, и отрицательная в противном случае) i-й (i = 1,..., m) группы.

Расчет.

Количество групп продукции с безубыточно проданными изделиями вычисляется по формуле

$$T = \sum_{i=1}^{m} k_i \text{ , где}$$

$$k_i = \begin{cases} 1, \text{ если среди } V_{ij} \text{ (для } j = 1,...,n \text{) нет отрицательных} \\ 0, \text{ если среди } V_{ij} \text{ (для } j = 1,...,n \text{) есть отрицательные} \end{cases}$$

Вывести полученный результат (T).

Конкретные исходные данные для проверки:

m = 4; n = 6;

V_{11} = 142; V_{12} = 74; V_{13} = 532; V_{14} = 326; V_{15} = –57; V_{16} = 364;
V_{21} = 462; V_{22} = 352; V_{23} = 76; V_{24} = 89; V_{25} = 253; V_{26} = 573;
V_{31} = 235; V_{32} = –34; V_{33} = 196; V_{34} = –92; V_{35} = 472; V_{36} = 428;
V_{41} = 96; V_{42} = 174; V_{43} = 214; V_{44} = 68; V_{45} = 225; V_{46} = 47

Задача 18. Вычисление общей суммы баллов по результатам тестирования студента.

Исходные данные (вводятся пользователем):

n – количество результатов тестирования;

t_i – результат i-го (i = 1,..., n) тестирования (равен 0, если на соответствующий вопрос был дан неверный ответ, или 1, если ответ правильный).

Расчет.

Общая сумма баллов вычисляется исходя из следующих условий оценивания результатов тестирования:

– за неправильный ответ (t_i = 0) дается нуль баллов;

– правильные ответы (t_i = 1) оцениваются следующим образом:

– за первые два ($i = 1, 2$) по 50 баллов,
– за следующие три ($i = 3, 4, 5$) по 25 баллов,
– за остальные ($i \geq 6$) по 5 баллов.

Вывести результат вычисления общей суммы баллов (S_B).

Конкретные исходные данные для проверки:

$n = 12$;
$t_1 = 0$; $t_2 = 1$; $t_3 = 0$; $t_4 = 1$; $t_5 = 1$; $t_6 = 0$;
$t_7 = 0$; $t_8 = 1$; $t_9 = 1$; $t_{10} = 1$; $t_{11} = 0$; $t_{12} = 0$

Задача 19. Вычисление размера консолидированного платежа с применением простой процентной ставки.

Исходные данные (вводятся пользователем):

N – срок консолидированного платежа;
p – процентная ставка;
m – количество платежей;
D_i – размер i-го платежа ($i = 1,..., m$);
n_i – срок i-го ($i = 1,..., m$) платежа.

Расчет.

Размер консолидированного платежа с применением простых процентов вычисляется по формуле

$$R = \sum_j D_j \cdot (1 + r_j \cdot p) + \sum_k D_k \cdot (1 + r_k \cdot p)^{-1}, \text{ где}$$

D_j – размеры объединяемых платежей со сроками $n_j \leq N$;
D_k – размеры объединяемых платежей со сроками $n_k > N$;

$r_j = N - n_j$;
$r_k = n_k - N$.

Вывести полученный результат вычисления размера консолидированного платежа.

Конкретные исходные данные для проверки:

$N = 1.4$; $p = 0.25$; $m = 8$;

$D_1 = 2000$;	$D_2 = 3000$;	$D_3 = 3500$;	$D_4 = 4000$;
$D_5 = 4500$;	$D_6 = 5000$;	$D_7 = 6500$;	$D_8 = 8000$;
$n_1 = 0.25$;	$n_2 = 0.5$;	$n_3 = 0.75$;	$n_4 = 1.25$;
$n_5 = 1.3$;	$n_6 = 1.5$;	$n_7 = 2.5$;	$n_8 = 2.75$

Задача 20. Начисление простых процентов при изменении суммы вклада и фиксированной процентной ставке.

Исходные данные (вводятся пользователем):

m – количество изменений остатков средств на счете;

Q_i – очередное i-е ($i = 1,..., m$) поступление средств на счет (вводится положительное число) или снятие средств со счета (вводится отрицательное число);

n_i – срок хранения i-й суммы ($i = 1,..., m$) до нового поступления (снятия) средств на счет (годы);

p – процентная ставка.

Расчет.

Сумма, полученная владельцем счета, с начислением простых процентов при изменении суммы вклада и фиксированной процентной ставке вычисляется по формуле

$$S = D_m + \sum_{i=1}^{m} (D_i \cdot n_i \cdot p) \text{ , где}$$

D_i – остаток средств на счете после очередного i-го ($i = 1,..., m$) поступления или снятия средств, вычисляемый следующим образом

$D_1 = Q_1$,

$D_i = D_{i-1} + Q_i$ для $i = 2,\ldots, m$

Вывести полученный результат вычисления суммы, полученной владельцем счета.

Конкретные исходные данные для проверки:

$m = 5$; $p = 0.14$;

$Q_1 = 10000$; $Q_2 = 5000$; $Q_3 = -7000$; $Q_4 = -3000$; $Q_5 = 6000$;

$n_1 = 0.5$; $n_2 = 0.1$; $n_3 = 0.3$; $n_4 = 1$; $n_5 = 0.4$

Задача 21. Вычисление последнего платежа в счет погашения кредита с помощью правила торговца.

Исходные данные (вводятся пользователем):

D – размер кредита (тыс. руб.);

n – срок ссуды (годы);

m – количество произведенных ранее платежей;

R_i – сумма i-го ($i = 1,..., m$) частичного платежа (тыс. руб.);

n_i – длительность i-го ($i = 1,..., m$) периода (годы);

p – процентная ставка.

Расчет.

Размер последнего платежа (остатка долга на конец срока) с помощью правила торговца вычисляется следующим образом

$$Q = D \cdot (1 + n \cdot p) - \sum_{i=1}^{m} R_i \cdot (1 + r_i \cdot p)\text{, где}$$

r_i – промежуток времени от момента i-го платежа (i = 1,..., m) до конца срока ссуды, вычисляемый следующим образом

$r_1 = n - n_1$,

$r_i = r_{i-1} - n_i$ для i = 2,..., m

Вывести полученный результат вычисления размера последнего платежа.

Конкретные исходные данные для проверки:

D = 1500; n = 0.75;
m = 3; p = 0.2
R_1 = 500; R_2 = 400; R_3 = 300;
n_1 = 0.3; n_2 = 0.2; n_3 = 0.15

Задача 22. Вычисление коэффициента ритмичности реализации товаров в течение анализируемого периода (любой период не более 1 месяца).

Исходные данные (вводятся пользователем):

n – общее количество дней анализируемого периода ($n \le 31$);
a_i – планируемый товарооборот за i-й день (i = 1,..., n);
b_i – фактический товарооборот за i-й день (i = 1,..., n).

Расчет.

Коэффициент ритмичности рассчитывается по формуле:

$$K_n = \frac{\sum_{i=1}^{n} \min(a_i, b_i)}{\sum_{i=1}^{n} a_i}\text{, где}$$

$\min(a_i,b_i)$ – наименьшее значение из a_i и b_i (i = 1,..., n).

Вывести полученное значение коэффициента ритмичности.

Конкретные исходные данные для проверки:

n = 7;
a_1 = 238.6; a_2 = 325.4; a_3 = 281.3; a_4 = 410.4;
a_5 = 350.8; a_6 = 273.9; a_7 = 290.5;
b_1 = 254.3; b_2 = 270.5; b_3 = 268.4; b_4 = 398.6;
b_5 = 374.5; b_6 = 308.6; b_7 = 274.9

Задача 23. Выявление изделий с одинаковой себестоимостью.

Исходные данные (вводятся пользователем):

m – количество групп продукции;

n – количество изделий в каждой группе;

d_{ij} – себестоимость j-го ($j = 1,..., n$) изделия i-й ($i = 1,..., m$) группы.

Расчет.

Выяснить для каждого i ($i = 1,..., m$), имеются ли среди d_{ij} ($j = 1,..., n$) одинаковые числа.

Вывести в зависимости от результата сообщение о наличии или отсутствии изделий с одинаковой себестоимостью для каждой группы продукции.

Конкретные исходные данные для проверки:

$m = 3$; $n = 6$

$d_{11} = 34$; $d_{12} = 52$; $d_{13} = 251$; $d_{14} = 524$; $d_{15} = 52$; $d_{16} = 52$;
$d_{21} = 126$; $d_{22} = 75$; $d_{23} = 362$; $d_{24} = 174$; $d_{25} = 52$; $d_{26} = 34$;
$d_{31} = 423$; $d_{32} = 79$; $d_{33} = 75$; $d_{34} = 284$; $d_{35} = 359$; $d_{36} = 75$

Задача 24. Выявление изделий с заданной стоимостью в группах продукции.

Исходные данные (вводятся пользователем):

n – количество групп продукции;

m – количество изделий в каждой группе;

d_{ij} – стоимость j-го ($j = 1,..., m$) изделия i-й ($i = 1,..., n$) группы;

s – стоимость, заданная пользователем.

Расчет.

Среди всех d_{ij} ($i = 1,..., n$; $j = 1,..., m$) найти элементы, равные числу s.

Вывести для каждого найденного элемента номер группы продукции и номер изделия. Если таких элементов нет, вывести сообщение об их отсутствии.

Конкретные исходные данные для проверки:

$n = 3$; $m = 5$; $s = 49{,}1$;

$d_{11} = 39.1$; $d_{12} = 65.3$; $d_{13} = 84.8$; $d_{14} = 49.1$; $d_{15} = 64.1$;
$d_{21} = 32.7$; $d_{22} = 94.2$; $d_{23} = 24.6$; $d_{24} = 36.9$; $d_{25} = 75.8$;
$d_{31} = 14.5$; $d_{32} = 49.1$; $d_{33} = 34.5$; $d_{34} = 22.3$; $d_{35} = 49.1$

Задача 25. Выявление в нескольких складах наибольших среди минимальных запасов товаров.

Исходные данные (вводятся пользователем):

n – количество складов;

m – количество товаров;

V_{ij} – объем запасов j-го ($j = 1,..., m$) товара, хранящегося в i-м ($i = 1,..., n$) складе.

Расчет.

Выявление товара с наибольшим объемом запасов среди товаров с минимальным объемом запасов, хранящихся на n складах, заключается в поиске среди всех V_{ij} ($i = 1,..., n$; $j = 1,..., m$) такого элемента S, для которого выполняется условие

$$S = \max_{i=1,...,n}\left(\min_{j=1,...,m} V_{ij} \right)$$

Вывести полученный результат (S) с указанием номера склада и номера товара.

Конкретные исходные данные для проверки:

$n = 3$; $m = 4$;

$V_{11} = 26$; $V_{12} = 78$; $V_{13} = 52$; $V_{14} = 81$;

$V_{21} = 74$; $V_{22} = 63$; $V_{23} = 49$; $V_{24} = 38$;

$V_{31} = 35$; $V_{32} = 48$; $V_{33} = 321$; $V_{34} = 120$

Задача 26. Выявление товаров с наименьшей и наибольшей стоимостью.

Исходные данные (вводятся пользователем):

n – количество поставщиков;

m – количество товаров;

S_{ij} – стоимость j-го ($j = 1,..., m$) товара, поставляемого i-м ($i = 1,..., n$) поставщиком.

Расчет.

Выявление товара с наименьшей стоимостью заключается в поиске среди всех S_{ij} ($i = 1,..., n$; $j = 1,..., m$) такого элемента $S_{\min}$, для которого выполняется условие

$$S_{\min} = \min_{i=1,...,n}\left(\min_{j=1,...,m} S_{ij} \right)$$

Выявление товара с наибольшей стоимостью заключается в поиске среди всех S_{ij} (i = 1,..., n; j = 1,..., m) такого элемента $S_{\max}$, для которого выполняется условие

$$S_{\max} = \max_{i=1,\ldots,n}\left(\max_{j=1,\ldots,m} S_{ij} \right)$$

Вывести полученные результаты ($S_{\min}$, $S_{\max}$,) и количество найденных товаров как с наименьшей ($N_{\min}$), так и наибольшей ($N_{\max}$) стоимостью.

Конкретные исходные данные для проверки:

n = 4; m = 5;

S_{11} = 459;	S_{12} = 764;	S_{13} = 638;	S_{14} = 793;	S_{15} = 764;
S_{21} = 387;	S_{22} = 129;	S_{23} = 300;	S_{24} = 129;	S_{25} = 567;
S_{31} = 286;	S_{32} = 793;	S_{33} = 459;	S_{34} = 790;	S_{35} = 793;
S_{41} = 129;	S_{42} = 427;	S_{43} = 327;	S_{44} = 793;	S_{45} = 459

2.3 Основы объектно-ориентированного программирования

Задача 27. Написать макрос, который выполняет удаление листа с названием *Лист2* из неактивной рабочей книги.

Задача 28. Написать макрос, который с помощью метода ***Add*** добавляет новый лист для диаграммы после первого в неактивную рабочую книгу. Аргументы в метод ***Add*** передавать обращением по имени. Умалчиваемые аргументы не указывать.

Задача 29. Написать макрос, который с помощью метода ***Add*** добавляет новый лист перед активным в активную рабочую книгу. Аргументы в метод ***Add*** передавать обращением по порядку. Умалчиваемые аргументы не указывать.

Задача 30. Написать макрос, который выделяет вторую строку диапазона B3:D8 в активном листе.

Задача 31. Написать макрос, который очищает рабочий лист от форматирования (приводит все свойства листа к значениям по умолчанию). Назвать макрос *Очистка* и запустить его на рабочем листе.

Задача 32. По данным о начислении налога на доходы физических лиц, находящимся в таблице на рабочем листе, вычислить количество

лиц с размерами налогов 1) менее 100 000, 2) от 100 000 до 400 000 (включительно) и 3) более 400 000.

Диапазон ячеек с размерами налога вводится пользователем в диалоговом окне во время выполнения программы.

Результаты вывести на рабочий лист в диапазон, указанный пользователем. Форма вывода показана на рис. 19:

менее 100 000:	...
от 100 000 до 400 000:	...
400 000 и более:	...

Рис. 19. Форма вывода для задачи 32.

Конкретные исходные данные для проверки представлены на рис. 20.

Налоги физических лиц

№ п/п	Ф.И.О.	Налог
1	Бардин Я.М.	414 365
2	Выборов П.Г.	93 781
3	Григорьянц А.К.	216 382
4	Доронин Л.Н.	12 794
5	Мешков С.П.	165 360
6	Неутехин Р.О.	51 562
7	Орлова И.П.	74 285
8	Осколкова Д.В.	512 743
9	Привалова Е.Н.	388 379
10	Шапошникова Ю.А.	46 829

Рис. 20. Исходный данные для задачи 32.

Задача 33. Расчет коэффициента приведения ренты при разной величине количества выплат ренты в год (1, 2 и 4) и разном количестве начислений процентов в год (1, 3 и 6).

Исходные данные (вводятся пользователем):
n – срок ренты,
j – годовая ставка процентов.

Расчет.
Коэффициент приведения ренты (a) рассчитывается по формуле:

$$a = \frac{1 - (1 + \frac{j}{m})^{-mn}}{p \cdot [(1 + \frac{j}{m})^{\frac{m}{p}} - 1]}$$

Здесь

m – число начислений процентов в год;

p – число выплат ренты в год;

n – срок ренты;

j – годовая ставка процентов.

Результаты представить в табличной форме на активном рабочем листе активной рабочей книги. Название таблицы оформить в соответствии со следующими параметрами шрифта: размер 13 пт, полужирный, двойное подчеркивание. Для отображения значения коэффициента приведения ренты установить ограничение: 2 знака после разделителя целой и дробной части.

Конкретные исходные данные для проверки:

$n = 10$; $j = 0{,}25$.

ЛИТЕРАТУРА

1. *Слепцова Л.Д.* Программирование на VBA в Microsoft Excel 2010. М. : ООО "Вильямс", 2010. – 432 с.

2. *Культин Н.Б.* Visual Basic: Освой самостоятельно. СПб : БХВ-Петербург, 2013. – 496 с.

3. *Кузьменко В.Г.* VBA. Эффективное использование. М. : «Бином. Лаборатория знаний», 2015. – 624 с.

4. *Лебедев В.М.* Программирование на VBA в MS Excel. Учебное пособие для академического бакалавриата. М. : Юрайт, 2017. – 272 с.

5. *Уокенбах Джон.* Excel 2013: Профессиональное программирование на VBA. М. : Диалектика/Вильямс, 2014. – 960 с.

Приложение А

ОТВЕТЫ К КОНТРОЛЬНЫМ ЗАДАЧАМ

1. $F_{01} = 2679918.3$; $F_1 = 2729883.6$; $T = 101.8644$
2. $R_0 = 48656.1$; $R_1 = 45874$; $P_r = –5.717881$
3. $Z_{01} = 1832.4$; $Z_1 = 1770.6$; $D_q = –0.0252$
4. $Z_0 = 7601.41$; $Z_1 = 6248.67$; $dZ_{отн} = –17.79591$
5. $Z_1 = 4703.05$; $Z_{01} = 4703.13$; $P_q = 0.999983$
6. $T_0 = 66324$; $T_1 = 74923.2$; $T_{10} = 75160.45$; $J_{tp} = 1.129649$; $J_p = 0.9968434$
7. $A = 0.4857887$
8. $S = 345000$
9. $S = 330000$
10. $S = 11985$
11. $S = 699957$
12. $S = 0.1712766$
13. $S = 0.1878144$
14. $T = 1.63326$
15. $s_1 = 149.6$; $s_2 = 215.3333$; $s_3 = 153.75$; $S_V = 518.6833$
16. $R = 2$
17. $T = 2$
18. $S_B = 115$
19. $R = 35038.65$
20. $S = 13562$
21. $S = 454$
22. $K_n = 0.9561473$
23. В 1 и 3 группах есть, во 2 группе нет
24. $d(1,4)$; $d(3,2)$; $d(3,5)$
25. $S = V(2, 4) = 38$
26. $S_{min} = 129$; $N_{min} = 3$; $S_{max} = 793$; $N_{max} = 4$
32. менее 100 000: 5
    от 100 000 до 400 000: 3
    400 000 и более: 2

33. Результаты представлены на рис. 21.

Расчет коэффициента приведения ренты

Число выплат ренты в год	Число начислений процентов в год		
	1	3	6
1	3,57	3,35	3,29
2	3,78	3,56	3,51
4	3,89	3,67	3,62

Срок ренты:	10
Годовая ставка процентов:	0,25

Рис. 21. Результаты решения задачи 33.

СПИСОК ОСНОВНЫХ ОПЕРАТОРОВ VBA

1. **Sub** <*имя*>([<*список*>]) – начало программы или подпрограммы. Здесь <*список* > – список переменных для приема входных данных и переменных для возвращения результатов выполнения подпрограммы. Для программы <*список* > не указывается.

2. **End Sub** – окончание программы или подпрограммы.

3. **Function** <*имя*>([<*список*>]) – начало функции. Здесь <*список* > – список переменных для приема входных данных.

4. **End Function** – окончание функции.

5. **Call** <*имя*>[(<*список*>)] – вызов программы или подпрограммы.

6. **Оформление комментариев**. Комментарий указывается после оператора в той же строке или в отдельной строке. Перед ним должен быть указан знак апострофа (‘). Если комментарий указан в отдельной строке, вместо апострофа можно использовать ключевое слово ***Rem***.

7. <*имя*> = <*выражение*> – оператор присваивания. Здесь <*имя*> – идентификатор переменной, <*выражение*> – арифметическое, логическое или текстовое выражение, которое, в частности, может быть переменной или константой.

8. **Dim** <*список*> – оператор описания переменных. Здесь <*список*> – это список имен переменных и массивов. Для указания типа после имени в списке указывается ключевое слово ***As*** и обозначение типа: ***Integer*** – целое обычное; ***Long*** – длинное целое; ***Single*** – вещественное с одинарной точностью; ***Double*** – вещественное с двойной точностью; ***String*** – строковый тип; ***Object*** – объектный тип.

9. **ReDim** <*список*>– оператор переопределения динамического массива, т.е. массива, объявленного в операторе ***Dim*** без указания размерности.

10. Цикл с постусловием ***Do – Loop Until***.

```
Do
   …
Loop Until <условие>
```

Операторы ***Do*** и ***Loop Until*** выполняют указанную между ними последовательность операторов до тех пор, пока логическое выражение <*условие*> равно ***False***.

11. Цикл с постусловием ***Do – Loop While***.

Do

…

Loop While <*условие*>

Операторы ***Do*** и ***Loop While*** выполняют указанную между ними последовательность операторов до тех пор, пока логическое выражение <*условие*> равно ***True***.

12. Цикл с предусловием ***While – Wend***.

While <*условие*>

…

Wend

Операторы ***While*** и ***Wend*** выполняют указанную между ними последовательность операторов до тех пор, пока логическое выражение <*условие*> равно ***True***.

13. Цикл со счетчиком ***For – Next***.

For <*имя*> = <*начало*> **To** <*конец*> [**Step** <*шаг*>]

…

Next [<*список*>]

Операторы ***For*** и ***Next*** повторяют выполнение указанной между ними последовательности операторов заданное число раз.

Здесь <*имя*> – имя переменной цикла, <*начало*> – ее начальное значение, <*конец*> – ее последнее значение, <*шаг*> – шаг изменения значений переменной цикла. Переменная цикла должна быть целого типа. Величины <*начало*>, <*конец*> и <*шаг*> могут быть заданы константами, переменными и арифметическими выражениями. Для ***Next*** можно указать список имен переменных циклов, вложенных друг в друга. Эти имена указываются в порядке от последнего к первому по вложенности циклов.

14. **GoTo** <*метка*> – переход на строку, обозначенную меткой. Такая метка может быть задана номером или именем, указанным в начале строки. Если метка задана номером, то после него должен быть указан пробел, а если именем, то – знак двоеточия.

15. Альтернативное ветвление

If <*условие*> **Then**

<*группа* 1>

[**Else**

<*группа* 2>]

End If

Здесь

<*условие*> – логическое выражение; <*группа* 1> – группа операторов, которые выполняются, если <условие> выполняется, т.е. результат выражения равен ***True***; <*группа* 2> – группа операторов, которые выполняются, если <условие> не выполняется, т.е. результат выражения равен ***False***; ***End If*** – окончание ветвления.

Если <*группа* 1> состоит из одного оператора и <*группа* 2> также состоит из одного оператора, то оператор можно записывать в одной строке без указания ***End If***:

If <*условие*> **Then** <*оператор* 1> [**Else** <*оператор* 2>]

Вместо этой конструкции можно использовать функцию ***IIf***:

IIf (<*условие*>, <*оператор* 1>, <*оператор* 2>)

Эта функция возвращает результат, полученный с помощью <*оператор* 1>, если <*условие*> равно ***True***, или результат, полученный с помощью <*оператор* 2>, если <*условие*> равно ***False***.

16. **Множественный выбор**.

Этот оператор оформляется с помощью ключевых слов ***Select Case*** (описание проверяемого условия), ***Case*** (описание ветви), ***Case Else*** (ветвь «*иначе*») и ***End Select*** (окончание оператора множественного выбора).

Select Case <*выражение*>
  Case <*результат* 1>
    <*группа* 1>
  …….
  Case <*результат N*>
    <*группа N*>
  [**Case Else**
    <*группа Else*>]
End Select

Здесь после ключевых слов ***Case*** указываются возможные результаты выражения и в следующих за ними строках – группы операторов, которые должны при этом выполняться. ***Case Else*** используется, если надо выполнить группу операторов, когда полученный результат не совпадает ни с одним из предусмотренных в ***Case***.

17. **Exit Sub** – прерывание программы или подпрограммы с возвращением в вызывающую процедуру.

18. **Exit Do** – прерывание циклов ***Do–Loop Until*** и ***Do–Loop While***.

19. **Exit For** – прерывание цикла ***For – Next***.

20. **End** – оператор прекращения выполнения вычислений.

ОБЪЕКТЫ, МЕТОДЫ И СВОЙСТВА

1. Типы объектных переменных:
   Object – объектная переменная,
   Workbook – рабочая книга,
   Worksheet – рабочий лист,
   Range – область рабочего листа.
2. Коллекции:
   Workbooks – *открытые* рабочие книги;
   Sheets – *все листы* открытой рабочей книги;
   Worksheets – *рабочие листы* рабочей книги;
   Charts – *листы диаграмм* рабочей книги;
   DialogSheets – *диалоговые листы* рабочей книги.
3. Операторные скобки ***With – End With***.

   With <*объект*>
   <*действие* 1>
   .....
   <*действие N*>
   End With

Используется, если необходимо произвести несколько действий с одним и тем же объектом.

4. Цикл ***For Each – Next***.

   For Each <*элемент*> **In** <*коллекция*>
   <*оператор* 1>
   ……..
   <*оператор N*>
   Next [<*элемент*>]

Используется, когда надо применить последовательность операторов ко всем элементам коллекции. Здесь <*коллекция*> – коллекция, к элементам которой надо применить операторы; <*элемент*> – имя объектной переменной, которая будет использоваться в операторах для ссылки на очередной элемент коллекции.

5. Свойства, которые *указывают на объект в другом объекте*:
   ActiveWorkbook – активная рабочая *книга*;

ActiveSheet – активный *лист* в коллекции листов;
ActiveCell – активная *ячейка* в области рабочего листа;
Selection – выделенная *область* рабочего листа;
Font – *шрифт* в диапазоне ячеек;
Interior – *заливка* в диапазоне ячеек.

6. Метод ***Delete***.

Удаляет объект, если он может быть удален (например, лист в коллекции листов рабочей книги). Если объект не может быть удален, этот метод выполняет действия в зависимости от объекта, к которому применяется. Например, применительно к столбцу рабочего листа этот метод приводит к перемещению влево содержимого ячеек всех столбцов, расположенных справа от него.

7. Метод ***Add***.

Добавляет новый элемент в коллекцию. Имеет разные аргументы в зависимости от того, к какому объекту (коллекции) применяется. Применительно к коллекции листов ***Sheets*** спецификация имеет следующий вид

Sheets.Add [*before*], [*after*], [*count*], [*type*]

В данном случае ***Add*** добавляет в ***Sheets*** новый лист типа ***type*** в количестве ***count*** перед листом ***before*** или после листа ***after***. По умолчанию новый лист добавляется перед активным, а ***count*** равен единице. Для аргумента ***type*** предусмотрены следующие значения:

xlWorkSheet – рабочий лист,
xlChart – лист для диаграммы,
xlDialogSheet – диалоговый лист,
xlModule – модульный лист.

По умолчанию этот аргумент имеет значение ***xlWorksheet***.

8. Свойства объекта ***WorkSheet*** (рабочий лист).

ActiveCell – активная ячейка,
Selection – выделенная область,
Range[(<ссылка>)] – диапазон ячеек, строк или столбцов,
Columns[(<ссылка>)] – столбцы,
Rows[(<ссылка>)] – строки,
Cells[(<арг_1>,<арг_2>)] – ячейка, находящаяся на пересечении строки и столбца, номера которых заданы соответственно <*арг*_1> и <*арг*_2>.

9. **Свойства Range-объекта**.

Address – адрес ячейки,
Count – количество элементов в диапазоне,
Name – имя,
Value – значение,
Formula – содержимое в A1-формате,
FormulaR1C1 – содержимое в R1C1-формате,
ColumnWidth – размер столбцов,
RowHeight – размер строк,
Range[(<ссылка>)] – диапазон ячеек, строк или столбцов,
Columns[(<ссылка>)] – столбцы,
Rows[(<ссылка>)] – строки,
Cells[(<арг_1>,<арг_2>)] – ячейка, находящаяся на пересечении строки и столбца, номера которых заданы соответственно <*арг*_1> и <*арг*_2>.

HorizontalAlignment – тип выравнивания по горизонтали. Для него предусмотрены значения:

xlLeft – по левому краю,
xlRight – по правому краю,
xlCenter – по центру,
xlCenterAcrossSelection – по центру объединенной области нескольких ячеек.

VerticalAlignment – тип выравнивания по вертикали. Для него предусмотрены значения:

xlTop – по верхнему краю,
xlBottom – по нижнему краю,
xlCenter – по центру.

WrapText – перенос слов (True – установить, False – отменить).

EntireColumn и ***EntireRow*** – указывают соответственно на все столбцы и строки рабочего листа, ячейки которых принадлежат Range-объекту.

CurrentRegion – указывает на прямоугольный диапазон ячеек, окруженный пустыми строками и столбцами, внутри которого находится данный Range-объект.

Borders(<*аргумент*>) – проведение границ для ячеек внутри Range-объекта. Тип линии определяется аргументом с помощью следующих значений: xlLeft – слева, xlRight – справа, xlTop – сверху, xlBottom – сни-

зу, xlInsideVertical – между ячейками по вертикали, xlInsideHorizontal – между ячейками по горизонтали. Это свойство можно использовать и для проведения границ рядом с областью. В этом случае аргумент должен иметь значения: xlEdgeLeft (слева), xlEdgeRight (справа), xlEdgeTop (сверху) или xlEdgeBottom (снизу).

<*объект*>.***Offset***(<*смещ_стр*>, <*смещ_столбца*>) – указывает на ячейку относительно левой верхней ячейки Range-объекта (<*объект*>). Первый аргумент (<*смещ_стр*>) задает смещение строки, а второй аргумент (<*смещ_столбца*>) – смещение столбца.

10. **Методы Range-объекта**.

Select – выделение области;

Clear – удаление содержимого в ячейках области;

Delete – удаление области;

End(<*арг*>) – определение граничной непустой ячейки в столбце или строке относительно заданной ячейки. <арг> задает направление поиска: xlDown (вниз), xlUp (вверх), xlToLeft (влево), xlToRifht (вправо);

Copy <*диапазон*> – копирование в <*диапазон*>;

Cut <*диапазон*> – перемещение в <*диапазон*>;

AutoFill <*диапазон*> – автозаполнение в <*диапазон*>;

AutoFit – автоподгонка;

Merge – объединение ячеек;

BorderAround – проведение обрамления вокруг Range-объекта.

11. Метод ***InputBox***.

Это метод объекта ***Application***. Он обеспечивает ввод ссылки на Range-объект в диалоговом окне. Спецификация:

InputBox(*Prompt* [, *Title*] [, *Default*] [, *Left*] [, *Top*] [, *HelpFile*>, *HelpContextID*] [, *Type*])

Здесь *Prompt* – текст комментария для вывода в окне; *Title* – текст заголовка окна; *Default* – значение, которое выводится в поле ввода при открытии окна (для “умолчания”); *Left* и *Top* – координаты верхнего левого угла окна; *HelpFile* и *HelpContextID* – файл и раздел справочной системы, прилагаемой к программе; *Type* – тип возвращаемого значения: 0 – формула, 1 – число, 2 – текстовая строка, 4 – логическое значение (True или False), 8 – ссылка на область рабочего листа, 16 – значение ячейки, 64 – массив значений области листа.

СОДЕРЖАНИЕ

Издание подготовлено в авторской редакции

Отпечатано на участке цифровой печати
Издательского Дома Томского государственного университета

Заказ № 2430 от «15» марта 2017 г. Тираж 120 экз.