

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ АРХИТЕКТУРНО-
СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ**

И.В.Маланичев, Л.Б.Ермолаева

**РУКОВОДСТВО
ПО ПРИКЛАДНОМУ ПРОГРАММИРОВАНИЮ НА VBA**

Учебное пособие

Казань
2016

УДК 519.85
ББК 32.973.26-018
М18

Маланичев И.В., Ермолаева Л.Б.

М18 Руководство по прикладному программированию на VBA: Учебное пособие / Маланичев И.В., Ермолаева Л.Б. – Казань: Изд-во Казанск. гос. архитектур.-строит. ун-та, 2016. – 83 с.

ISBN 978-5-7829-0504-0

Печатается по решению Редакционно-издательского совета Казанского государственного архитектурно-строительного университета

Учебно-методическое пособие разработано для студентов направлений подготовки 08.03.01 «Строительство», 09.03.02 «Информационные системы и технологии», 23.03.01 «Технология транспортных процессов», а также для магистров и аспирантов.

Пособие предназначено для углубленного изучения программирования на языке Visual Basic for Application как средства расширения возможностей табличного процессора Excel. Материал представлен в форме решения прикладных задач от создания простых макросов, автоматизирующих работу с электронными таблицами, до сложных имитационных моделей, использующих средства объектно-ориентированного и визуального программирования.

Рецензенты

Ведущий специалист Отдела информационных геологических ресурсов и мониторинга геологической среды МЭПР РТ

С.В.Ермолаев

Проректор по организационно-правовым вопросам и информационным технологиям, зав.кафедрой ИТиСАПР КГАСУ

Д.М.Кордончик

УДК 004.92

ББК 32.73-018.2

© Казанский государственный архитектурно-строительный университет, 2016

© Маланичев И.В.,
Ермолаева Л.Б., 2016

Введение

Пособие составлено на основе заданий, используемых на кафедре прикладной математики КГАСУ в процессе преподавания программирования на языке Visual Basic for Application (VBA).

Оно наглядно показывает, что расширение возможностей табличного процессора Microsoft Excel средствами VBA позволяет не только упростить и автоматизировать работу с электронными таблицами путем создания простых макросов, но и превращает Excel в мощный инструмент объектно-ориентированного и визуального программирования для математического моделирования и решения прикладных задач любой сложности.

Глава 1 содержит материалы для лабораторных занятий по теме «Программирование с использованием массивов» (Информатика, 1 курс, направление «Строительство», направление «Технологии транспортных процессов»).

Глава 2 содержит материалы для лабораторных занятий по теме «Структурное программирование. Подпрограммы» (Информатика, 1 курс, направление «Строительство», направление «Технологии транспортных процессов»).

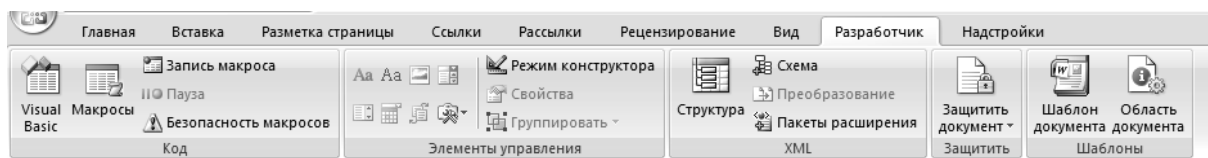
Глава 3 содержит материалы для лабораторных занятий по темам «Функции пользователя», «Разработка макросов» (Технологии программирования, 2 курс, направление «Информационные системы и технологии»).

Главы 4 и 5 содержат материалы для изучения тем «Вычислительный эксперимент и математическое моделирование», «Стохастическое моделирование» (Математическое моделирование, 1 курс (магистры), направление «Строительство»).

Пособие представлено в форме решения прикладных задач, начиная с создания простых макросов, автоматизирующих работу с электронными таблицами и заканчивая сложными имитационными моделями, использующими средства объектно-ориентированного и визуального программирования.

Программирование в Excel – это процесс создания макросов Visual Basic, встроенных в Excel. Макросы (макрокоманды) – это программы, написанные на языке программирования VBA. Для работы с макросами в Excel имеется вкладка «Разработчик», содержащая основные инструменты и среда программирования, вызываемая кнопкой **Редактор Visual Basic** или клавиатурной комбинацией ALT+F11. Если вкладка «Разработчик» отсутствует на ленте, то

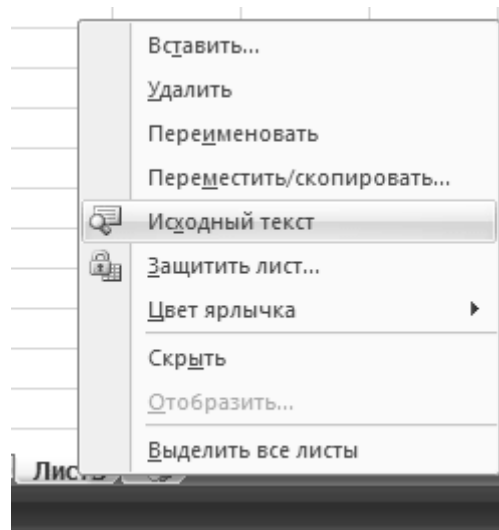
- нажмите (круглую) кнопку **Office** → **Параметры Excel** (справа внизу);
- в открывшемся диалоговом окне **Параметры Excel** в меню (слева) выберите **Основные**; – в разделе **Основные параметры работы с Excel** установите флажок **Показывать вкладку "Разработчик"** на ленте и закрыть окно.



VBA-программа имеет модульную структуру. Модулями в VBA называют структурные элементы, соответствующие основным объектам.

Модули делятся на пять основных типов:

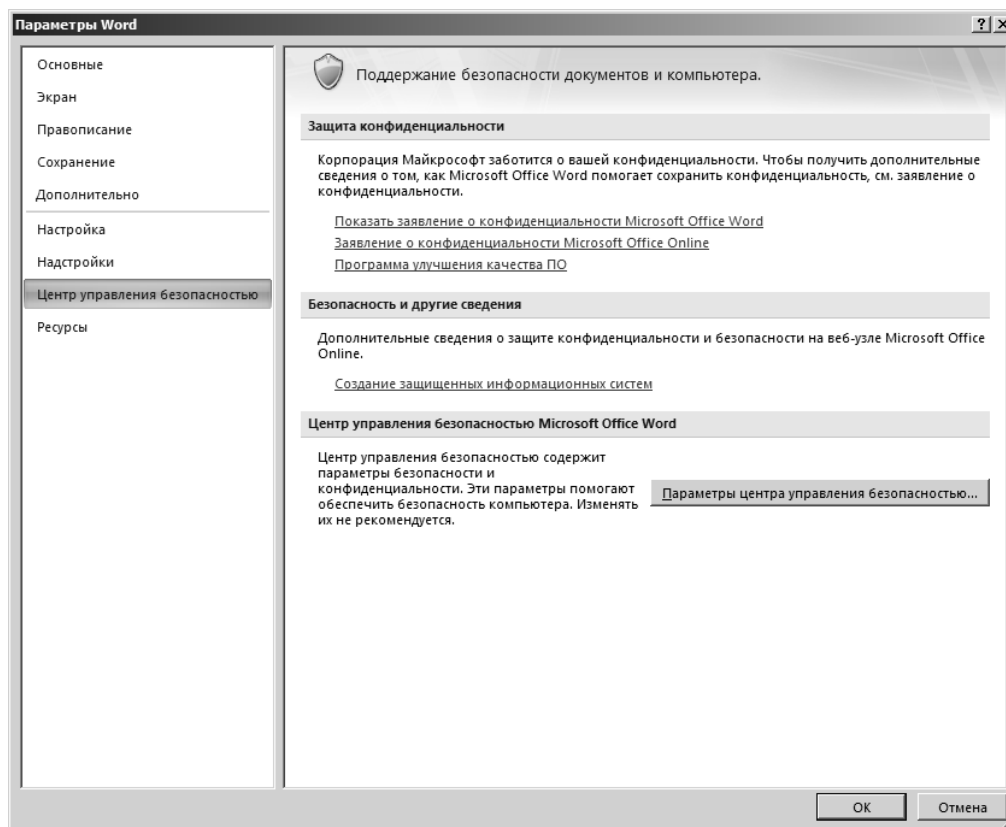
- Компонентные модули – модули основных объектов-компонент, из которых состоит приложение. В Excel это Модуль книги и Модуль листа. Код модуля можно отобразить через проводник объектов (**Project Explorer**) редактора VBA двойным щелчком по объекту в списке. В компонентных модулях содержатся "встроенные" процедуры. В модуль листа можно попасть, если, щелкнув правой кнопкой мыши по ярлычку листа, выбрать из контекстного меню пункт **Исходный текст**
- Стандартные модули. Стандартным называется модуль, не связанный ни с одним объектом. Обычно в стандартных модулях содержатся часто используемые процедуры функции, объявления глобальных переменных. При записи макрорекордером автоматически создаются стандартные модули.
- Модули классов описывают объекты, созданные пользователем.
- Модули пользовательских форм содержат код и графическое описание пользовательских диалоговых окон.



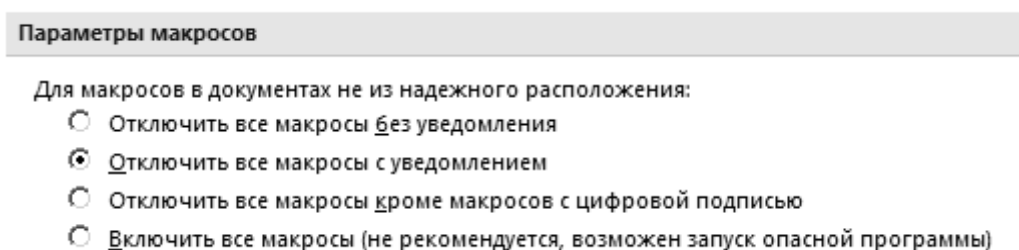
Все модули, кроме компонентных, вставляются в проект VBA через меню **View – Insert** редактора.

Модульная структура VBA-программы не ограничивается делением на модули. Каждый программный модуль, в свою очередь, состоит из подпрограмм: функций и процедур.

Язык VBA позволяет выполнять различные действия с помощью набора макросов, вызываемых автоматически без ведома пользователя. Механизм автоматических макросов используется макровирусами для активизации и распространения. Для защиты от макровирусов предусмотрена возможность выбора режима безопасности:



- нажмите кнопку **Office** → **Параметры Excel**;
- в открывшемся диалоговом окне **Параметры Excel** в меню (слева) выберите **Центр управления безопасностью**; нажмите кнопку **Параметры центра управления безопасностью** и выбрать нужный вариант



- закрыть окно.

1. Составление алгоритмов и программ с использованием сложных структур данных Excel

Задание 1. Решить задачу с использованием динамических массивов.

Массив – совокупность переменных одного типа (целые, с плавающей точкой, строки и т.д.), имеющих общее имя (различаются по номеру – индексу массива) и хранящиеся в одной области памяти. Последнее свойство массива требует обязательного резервирования памяти для хранения массива. Это достигается обязательной операцией объявления массива.

Массив может быть объявлен операторами объявления переменных
`Dim x(5) As Double, y(1 To 10), z(5, 5), a(25)`
`Public c(10,10) as Single`

`n=8`

`Dim b(n) As Integer`

Приведенные выше примеры объявляют статические массивы, размер которых известен до выполнения программы. Если размер массива программа получает как внешнее данные, через операцию ввода или считывая значение из таблицы Excel, то массив должен быть объявлен как динамический

`n = Cells(1, 1)`

`ReDim x(n)`

Размер (но не размерность) динамического массива может быть изменен в процессе работы программы повторным выполнением оператора `ReDim`. При этом значения переменных массива удаляются. Если их необходимо сохранить, то следует использовать оператор с ключевым словом `Preserve`.

`ReDim Preserve x(n)`

Для освобождения памяти от массива служит оператор `Erase`.

`Erase x`

Массивы в VBA допускают только поэлементные операции, организуемые обычно через цикл `For ... Next` по индексу массива. Единственное исключение – передача массива как аргумента функции. Пример функции, вычисляющей среднее значение элементов массива:

```
Function f(x(), n)
    s=0
    for i = 1 To n
        s = s + x(i)
    next i
    f = s / n
End Function
```

Пример 1. Даны n элементов массива a и m элементов массива b .
Дополнить массив a массивом b сначала.

Порядок работы.

Размер массивов не указан в условии задачи. Это означает, что программа должна работать для любых значений, которые выступают как данные задачи.

1) Введем значения массивов (произвольные числа в произвольном количестве) в таблицу Excel.

2) Вставим в ячейку C2 функцию подсчета чисел в столбце таблицы
СЧЁТ(A:A)

и скопируем ее в ячейку D2. Теперь в ячейках C2 и D2 находятся значения размеров массивов a и b .

	A	B	C	D	E
1	a	b	n	m	
2	6	123	10	3	
3	-3,5	-98			
4	8	0,789			
5	17				
6	190				
7	345				
8	-45				
9	0,097				
10	23				
11	-9				
12					
13					
14					

3) Откроем редактор VBA (Alt F11) и в модуль листа с исходными данными введем код программы

```
Sub program1()  
n = Cells(2, 3)      'ввод размера массива a  
ReDim a(n)           'объявление динамического массива a  
  
m = Cells(2, 4)      'ввод размера массива b  
ReDim b(m)           'объявление динамического массива b  
  
'цикл ввода значений массива a  
For i = 1 To n  
    a(i) = Cells(i + 1, 1)  
Next i
```

```

'цикл ввода значений массива b
For j = 1 To m
    b(j) = Cells(j + 1, 2)
Next j

```

```

'расширение массива b с сохранением значений
ReDim Preserve b(m + n)

```

```

'переписывание значений a в b
For i = 1 To n
    b(i + m) = a(i)
Next i

```

```

'расширение массива a
ReDim a(m + n)

```

```

'переписывание значений b в a
For i = 1 To m + n
    a(i) = b(i)
    Cells(i + 1, 6) = a(i)
Next i

```

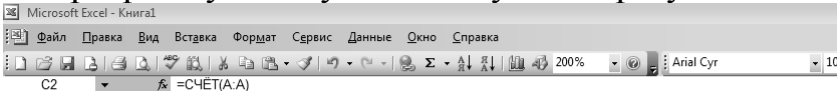
```

'удаление массива b
Erase b

```

End Sub

4) Выполнить программу и получить следующий результат:



	A	B	C	D	E	F
1	a	b	n	m		a
2	6	123	10	3		123
3	-3,5	-98				-98
4	8	0,789				0,789
5	17					6
6	190					-3,5
7	345					8
8	-45					17
9	0,097					190
10	23					345
11	-9					-45
12						0,097
13						23
14						-9
15						

Варианты задания 1.1

1. Из последовательности чисел a_1, a_2, \dots, a_n выбрать отрицательные элементы, подсчитать их число и переписать их подряд в массив X.
2. Вычислить компоненты векторов $a(a_1, a_2, \dots, a_n)$ и $b(b_1, b_2, \dots, b_n)$ по формуле $a_i = \arctg(i+2)/n$; $b_i = i + \cos i$ и вывести их на печать. Из векторов a и b получить вектор $c(a_1 b_1, a_2 b_2, \dots, a_n b_n)$, компоненты которого пронумеровать в порядке от 1 до $2n$ ($n=10$).

Из последовательности чисел y_1, y_2, \dots, y_n выбрать элементы, кратные 3 (т.е. делящиеся на 3 без остатка). Подсчитать их количество и напечатать их порядковые номера.

3. Вычислить значения компонент вектора $x(x_1, x_2, \dots, x_n)$ по формуле

$$x_i = \begin{cases} \arctg(i+2)/(n+3), & \text{если } \lg(ni) < 0 \\ e^{i+\cos n} \cos i, & \text{если } \lg(ni) \geq 0. \end{cases}$$

Выбрать положительные компоненты полученного вектора и переписать их в массив $b(b_1, b_2, \dots, b_k)$, $k=0,1,\dots,5$.

4. Из последовательности чисел a_1, a_2, \dots, a_n выбрать элементы, равные 0, подсчитать их количество, а оставшиеся элементы распечатать в одну строку.

5. Вычислить компоненты вектора $x(x_1, x_2, \dots, x_n)$ по заданной формуле $x_i = a \sin(0,5i) + b \cos(i-1)$, где $a=2,2$; $b=-0,4$ ($i=1,2,\dots,10$). Вывести их на печать. Преобразовать полученный вектор следующим образом: все отрицательные компоненты увеличить на 0,5, а все положительные компоненты заменить на 1. Преобразованный вектор распечатать в столбец.

6. Дана последовательность чисел $x(x_1, x_2, \dots, x_n)$. Вычислить сумму элементов с нечетными номерами и произведение элементов с четными номерами.

7. Дана последовательность чисел $x(x_1, x_2, \dots, x_n)$. Выбрать из нее целые числа и вычислить значение величины $M = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$ Исходные данные: $n=15, X = (0,1; 0,2; 1,3; 0,4; 2; 9; 10,2; 5,5; 2,3; 1,5; 1; 0,7; 15; 3; 4,5)$.

8. Дана последовательность чисел $X = (5,2; -3; 0,7; -2; 0,9; 1,5; 3,4; 0,5)$. Вычислить значения функции

$$f(x_i) = \begin{cases} 2x_i + 5, & \text{если } x_i < 0, \\ x_i^2 - 2,5, & \text{если } 0 \leq x_i \leq 1, \\ x_i - 0,5, & \text{если } x_i > 1, \quad (i=1,2,\dots,8). \end{cases}$$

Найти максимальный элемент, напечатать его и его порядковый номер.

9. Дан вектор $y(y_1, y_2, \dots, y_{10})$. Найти его максимальный элемент, напечатать его и его порядковый номер.

10. Дана последовательность чисел $b(b_1, b_2, \dots, b_n)$. Выбрать положительные элементы и найти среднее арифметическое квадратов этих чисел.

11. Вычислить компоненты вектора $x(x_1, x_2, \dots, x_8)$ по формуле

$$x_i = \begin{cases} e^{\sqrt{i+1}} + 2, & \text{если } \sin(i+a) > 0,5, \\ 3\operatorname{tg}(i+1), & \text{если } \sin(i+a) \leq 0,5. \end{cases}$$

Подсчитать количество отрицательных элементов вектора и вывести их порядковые номера.

12. Дана последовательность чисел $x(x_1, x_2, \dots, x_n)$. Разбить ее на две последовательности, а именно, на последовательность $b(b_1, b_2, \dots, b_k)$ с отрицательными элементами и последовательность $a(a_1, a_2, \dots, a_m)$ с положительными элементами ($n = m + k$).

13. Дан вектор $Y = (2; 0,3; -3,8; 6,4; -1,5; 4,2)$. Вычислить значения функции $d = t^2 \sin t$, если t принимает значения от 1 до 2,5 с шагом 0,3. Значения величин d и t представить в виде векторов и напечатать их элементы. Найти средний модуль разности между элементами векторов d и Y по формуле $L = \frac{1}{6} \sum_{i=1}^6 |d_i - y_i|$

14. Дана последовательность чисел y_i ($i = 1, 2, \dots, n$). Заменить в ней максимальный элемент на 1, а минимальный на -1.

15. Вычислить компоненты вектора $y(y_1, y_2, \dots, y_n)$ по формуле $y_i = 4\sqrt{1+2x_i^2} / (x_i + 1) \sin(\pi x_i / 2)$ ($i = 1, 2, \dots, n$) и вычислить сумму элементов с четными индексами при $n = 8$, $X = (0,1; 0,3; 1; 1,4; 2; 2,5; 3; 7,5)$.

16. Дана последовательность чисел y_k ($k = 1, 2, \dots, n$). Выбрать из них положительные и найти среди них наибольшее.

17. Дан вектор $X = (4; 2,1; 2,5; 2,1; 3,2; 1,5; 0,8; 1)$. Определить компоненты вектора $Z(z_1, z_2, \dots, z_n)$, заданного формулой $z_i = \sum_{k=1}^5 \frac{x_i^k}{(k+1)}$ ($i = 1, 2, \dots, n$) при $n = 7$.

18. Дана последовательность чисел y_k ($k = 1, 2, \dots, n$). Выбрать из нее только целые числа и подсчитать их количество.

19. Дан вектор $A = (4; 2,1; 2,5; 2,1; 3,2; 1,5; 0,8; 1)$. Определить компоненты вектора $Z(z_1, z_2, \dots, z_n)$, заданного формулой $z_i = a_i + \sum_{k=1}^{i-1} z_k$, причем $z_1 = a_1$ ($n = 7$).

20. Из последовательности чисел y_1, y_2, \dots, y_n выбрать элементы,

кратные 5 (т.е. делящиеся на 5 без остатка). Подсчитать их количество и вычислить их сумму.

21. Определить компоненты вектора $Z(z_1, z_2, \dots, z_n)$, заданного формулой $z_i = p_i q_{n+1} - i$, где $p_i = \arctg \frac{i+2}{10}$, $q_i = e^{i+\cos\sqrt{15}}$, $i=1, 2, \dots, n$ при $n=10$.

22. Дана последовательность чисел $y_k (k=1, 2, \dots, 15)$. Выбрать из нее только числа, удовлетворяющие условию $-0,5 < y_k < 8$ и вычислить их среднее арифметическое.

23. Дана окружность $(x-a)^2 + (y-b)^2 = r^2$ и n точек, координаты x и y которых заданы массивами $x(x_1, x_2, \dots, x_n)$ и $y(y_1, y_2, \dots, y_n)$.

Вычислить количество точек, лежащих на окружности, и напечатать их номера, если $a=4$; $b=-6$; $r=7$; $x=(-2; -3; 4; 8,5; 4; 7; 5,5)$; $y=(4; -6; -13; -8; 1; -1; 0)$.

24. Дана последовательность чисел $b_k (k=1, 2, \dots, 10)$. Найти среднее геометрическое положительных чисел этой погрешности, если даны значения элементов вектора $b=(-1; -5; 4; -20; -14; 32; 2; -11; 16; -3)$.

25. Дана последовательность чисел $x_k (k=1, 2, \dots, 10)$. Составить таблицу значений по формуле $x_k (k=1, 2, \dots, 10)$. по формуле $M(x_k) = \sum_{i=1}^8 \frac{(-1)^n x_k^n}{2n-1}$. Из полученных чисел выбрать максимальное, если задан массив $x=(1,9; 2; 1,6; 0,9; 2,3; 7,6; 1; 7,2)$ и $n=8$.

26. Дана последовательность чисел $a_k (k=1, 2, \dots, 10)$. Записать вектор, каждый элемент которого является суммой элементов, равноудаленных от концов данной последовательности.

27. Вычислить значения функции $z_i = \frac{x_i y_i}{x_i + y_i}$ и записать их в виде элементов вектора Z , если элементы y_i массива Y известны, $x_i = a + ih$. Просуммировать элементы z_i вектора Z , стоящие на нечетных местах. Исходные данные: $a=2$; $h=0,5$; $Y=\{2,1; 8,3; -4,5; 3,9; 0,8; -0,4; 1,5; 9,2; 7,3; -6,5\}$.

28. Даны два вектора по 10 элементов в каждом. Найти максимальную разность соответствующих элементов векторов и порядковый их номер.

29. Дан вектор с элементами $a_k (k=1, 2, \dots, n)$. Вычислить компоненты вектора $c_k (k=1, 2, \dots, n)$, заданные формулой $c_k = b_k - a_k^2$, а компоненты вектора b по формуле $b_k = \sqrt{\lg(1,5k - k/2)}$ при $n=6$; $a=(-1; -0,5; 4,2; -20; -1,4; 3,2; 2; -11,5; 1,6; -3)$.

30. Дана последовательность чисел $b_k (k=1, 2, \dots, 10)$. Найти среднее арифметическое отрицательных чисел этой погрешности, если даны значения элементов вектора $b=(-1; -5; 4; -20; -14; 32; 2; -11; 16; -3)$.

Варианты задания 1.2

1. Дан двумерный массив $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Найти произведение элементов массива, стоящих не на главной диагонали, для которых $i + j$ – нечетное число. Вывести на печать все промежуточные значения.
2. Дан двумерный массив $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Найти произведение минимального нечетного и максимального четного элементов, находящихся на главной диагонали. Вывести на печать все промежуточные значения.
3. Дан двумерный массив $C_{ij} (i = \overline{1,8}; j = \overline{1,8})$. Найти произведение максимального отрицательного и минимального положительного элементов на главной диагонали. Вывести на печать все промежуточные значения.
4. Дан двумерный массив $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Найти произведение минимального положительного и максимального нечетного элементов в каждой строке. Вывести на печать все промежуточные значения.
5. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Найти произведение минимального положительного и максимального нечетного элементов в каждом столбце. Вывести на печать все промежуточные значения.
6. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Найти минимальные положительные и максимальные нечетные элементы в каждой строке и определить наименьшее среди них. Вывести на печать все промежуточные значения.
7. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Переставить в обратном порядке все числа каждой строки (последнее число на первое место, предпоследнее на второе и т.д.). Вывести на печать все промежуточные значения.
8. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Найти максимальные нечетные и минимальные положительные элементы в каждой строке и переставить их местами. Вывести на печать все промежуточные значения.
9. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,6}; j = \overline{1,6})$. Найти максимальные отрицательные и минимальные положительные элементы в каждой строке. Заменить на +1 минимальные положительные, на -1 максимальные отрицательные. Вывести на печать все промежуточные и окончательные значения.
10. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Записать числа первой строки в первый столбец, числа второй строки во второй и т.д.. Вывести на печать все промежуточные и окончательные значения.

11. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Найти в каждой строке числа, кратные 3, и заменить их минимальным числом данной строки. Вывести на печать все промежуточные и окончательные значения.
12. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,6}; j = \overline{1,6})$. Найти в каждом столбце числа, кратные 5, и заменить их минимальным числом данного столбца. Вывести на печать все промежуточные и окончательные значения.
13. Дан двумерный массив $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Заменить первой строки элементами главной диагонали, а элементы главной диагонали элементами первой строки. Вывести на печать все промежуточные значения.
14. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Заменить элементы главной диагонали среднеарифметическим значением элементов соответствующей строки. Вывести на печать все промежуточные значения.
15. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,6}; j = \overline{1,6})$. Найти в каждой строке числа, кратные 3, и заменить их минимальным положительным числом этой данной строки. Вывести на печать все промежуточные и окончательные значения.
16. Дан двумерный массив $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Найти произведение максимального отрицательного и минимального положительного элементов каждой строки. Вывести на печать все промежуточные значения.
17. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Переставить в обратном порядке числа каждого столбца (последнее число на первое, предпоследнее на второе и т.д.). Вывести на печать все промежуточные значения и результирующую матрицу.
18. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,6}; j = \overline{1,6})$. Найти максимальные нечетные и минимальные положительные элементы в каждой строке и заменить их на соответствующий элемент главной диагонали. Вывести на печать все промежуточные и окончательные значения.
19. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Заменить элементы главной диагонали произведением максимального отрицательного и минимального положительного элементов соответствующей строки. Вывести на печать все промежуточные и окончательные значения.
20. Дан двумерный массив из целых чисел $C_{ij} (i = \overline{1,5}; j = \overline{1,5})$. Заменить элементы главной диагонали среднеарифметическим значением элементов соответствующей строки. Вывести на печать все промежуточные и

окончательные значения.

21. Дан двумерный массив из целых чисел $C_{ij}(i = \overline{1,6}; j = \overline{1,6})$. Найти максимальные отрицательные и минимальные положительные элементы в каждой строке. Вывести на печать все промежуточные значения.

22. Дан двумерный массив $C_{ij}(i = \overline{1,6}; j = \overline{1,6})$. Заменить в нем все строки и все столбцы с нечетными номерами на строки и столбцы с четными номерами. Вывести на печать все промежуточные и окончательные значения.

23. Дан двумерный массив $C_{ij}(i = \overline{1,5}; j = \overline{1,5})$. Найти сумму элементов массива, стоящих на главной диагонали, для которых $i + j$ – четное число. Вывести на печать все промежуточные значения.

24. Дан двумерный массив $C_{ij}(i = \overline{1,5}; j = \overline{1,5})$. Вычислить максимальные по модулю элементы в каждой строке и минимальные по модулю элементы в каждом столбце. Вывести на печать все промежуточные значения.

25. Дан двумерный массив $C_{ij}(i = \overline{1,5}; j = \overline{1,5})$. Найти минимальные элементы в каждой строке и максимальные в каждом столбце. Вывести на печать все промежуточные значения.

26. Дан двумерный массив $C_{ij}(i = \overline{1,5}; j = \overline{1,5})$. Найти сумму элементов массива, стоящих на главной диагонали, для которых $i + j$ – четное число. Вывести на печать все промежуточные значения.

27. Дан двумерный массив $C_{ij}(i = \overline{1,5}; j = \overline{1,5})$. Найти произведение минимального четного и максимального нечетного элементов, находящихся вне главной диагонали. Вывести на печать все промежуточные значения.

28. Дан двумерный массив $C_{ij}(i = \overline{1,8}; j = \overline{1,8})$. Найти произведение максимального положительного и минимального отрицательного элементов на главной диагонали. Вывести на печать все промежуточные значения.

29. Дан двумерный массив из целых чисел $C_{ij}(i = \overline{1,5}; j = \overline{1,5})$. Найти произведение минимального положительного и максимального четного элементов в каждой строке. Вывести на печать все промежуточные значения.

30. Дан двумерный массив из целых чисел $C_{ij}(i = \overline{1,5}; j = \overline{1,5})$. Найти произведение максимального положительного и минимального нечетного элементов в каждом столбце. Вывести на печать все промежуточные значения.

2. Структурный подход в программировании

Нисходящее проектирование

Одним из приемов снижения трудоемкости процесса программирования является структурный подход. Его основным принципом является модульное программирование, предполагающее разделение основной задачи на отдельные подзадачи и создание для них отдельной автономной программы – модуля. Специально созданная программа объединяет все модули в целое и управляет их работой. Программа на VBA всегда состоит из модулей и подпрограмм.

При разработке модульных программ применяются два метода проектирования – нисходящее и восходящее.

При нисходящем проектировании разработка программного комплекса идет сверху вниз. На первом этапе разработки кодируется, тестируется и отлаживается головной модуль, который отвечает за логику работы программы. Остальные модули заменяются заглушками, имитирующими работу этих модулей. Применение заглушек позволяет на самом раннем этапе проектирования проверить работоспособность программы и локализовать источник ошибок. На последних этапах проектирования все заглушки постепенно заменяются рабочими модулями.

Задание 2. Решить комбинированную задачу с использованием массивов, используя технику нисходящего проектирования.

Пример 2. Вычислить:

$$S = \begin{cases} \sqrt{\sum_{i=1}^{10} (z_i - 3)} & \text{если } \min_{1 \leq i \leq 10} x_i < \max_{1 \leq i \leq 10} y_i \\ \prod_{i=1}^{10} z_i & \text{во всех остальных случаях} \end{cases}$$

где

$$z_i = \begin{cases} x_i + y_i & \text{если } \sum_{i=1}^5 x_i > \frac{my_i}{n} \text{ и } \frac{x_i}{y_i} < 1 \\ x_i & \text{если } \prod_{i=1}^7 \frac{x_i}{y_i} > 1 \text{ или } \frac{m}{n} > 2 \\ y_i & \text{во всех остальных случаях} \end{cases}$$

x_i, y_i ($i = 1, 2, \dots, 10$) задать

m - номер варианта

n - номер группы

Порядок работы:

1) Проанализировать задачу и определить последовательность вычислений:

Сначала необходимо ввести данные: m и n

затем задать или ввести массивы x_i, y_i

вычислить $\sum_{i=1}^5 x_i$ и $\prod_{i=1}^7 \frac{x_i}{y_i}$

вычислить z_i ($i = 1, 2, \dots, 10$)

вычислить $\min_{1 \leq i \leq 10} x_i$ и $\max_{1 \leq i \leq 10} y_i$

вычислить S , при этом

вычислить $\sum_{i=1}^{10} (z_i - 3)$ или $\prod_{i=1}^{10} z_i$

2) Выделить структурные элементы алгоритма (все, что нельзя вычислить одной формулой: суммы, произведения, максимумы, минимумы, вычисление массивов и т.д.), обозначить их результаты как промежуточные переменные:

$$\begin{aligned} \sum_{i=1}^5 x_i &\rightarrow Sx, & \prod_{i=1}^7 \frac{x_i}{y_i} &\rightarrow Pxy, & \min_{1 \leq i \leq 10} x_i &\rightarrow \min X, & \max_{1 \leq i \leq 10} y_i &\rightarrow \max Y, \\ \sum_{i=1}^{10} (z_i - 3) &\rightarrow Sz & \prod_{i=1}^{10} z_i &\rightarrow Pz \end{aligned}$$

3) Составить программу:

Все переменные объявить как глобальные

Сгенерировать массивы как случайные числа в произвольном интервале

Вычисление всех переменных, промежуточных и результатов организовать в виде подпрограмм, вызываемых из основной программы:

Call Find_Sx()

сначала все подпрограммы ввести как "заглушки", т.е. без внутреннего кода:

Sub Find_Sx()

End Sub

4) Составить программы для нахождения промежуточных результатов

Постепенно заполнять "заглушки" содержательным кодом, проверяя их работу

Предусмотреть вывод всех данных, промежуточных и конечных результатов

5) Выполнить программу.

6) Все этапы работы оформить в тетради:

Условие

обозначения переменных

укрупненная блок-схема программы

таблица ввода-вывода

текст программы

	A	B	C	D	E	F	G	H	I	J	K	L
1	m	n	x	y	z	Sx	Pxy	minX	maxY	Sz	Pz	S
2	15	3	72,2818	-18,8862	53,3956	275,143	-5,13556	-12,2999	79,6588	451,253		21,2427
3			28,3798	17,7952	28,3798							
4			98,409	62,488	98,409							
5			54,9898	-4,2319	50,7579							
6			21,0821	34,6933	55,7754							
7			-12,2999	79,6588	-12,2999							
8			33,3455	76,0965	33,3455							
9			18,9105	29,7676	48,6782							
10			81,7983	24,4001	81,7983							
11			43,0133	20,4183	43,0133							

```
'объявление глобальных переменных модуля
Dim x(10), y(10), z(10)
Dim m, n, minX, maxY, Sx, Pxy, Pz, Sz, S
```

```
'основная процедура
Sub Main_Program()
    m = Cells(2, 1)      'ввод параметров
    n = Cells(2, 2)

    'генерация массивов x и y как случайных чисел
    a = -20              'произвольные значения
    b = 100              'границ интервалов

    Randomize
    For i = 1 To 10
        x(i) = a + (b - a) * Rnd()
        y(i) = a + (b - a) * Rnd()
        Cells(i + 1, 3) = x(i)
        Cells(i + 1, 4) = y(i)
    Next i

    'вызов процедуры нахождения суммы элементов массива x
    Call Find_Sx
    Cells(2, 6) = Sx

    'вызов процедуры нахождения произведения элементов
    'массивов x и y
    Call Find_Pxy
    Cells(2, 7) = Pxy

    'вызов процедуры нахождения элементов массива z
    Call Find_Z

    'вызов процедуры нахождения минимального значения
```

```

'элементов массива x
    Call Find_minX
    Cells(2, 8) = minX

'вызов процедуры нахождения максимального значения
'элементов массива y
    Call Find_maxY
    Cells(2, 9) = maxY

'вызов процедуры нахождения S
    Call Find_S
    Cells(2, 12) = S
End Sub      'конец основной процедуры

```

```

Sub Find_Sx()
    Summa = 0
    For i = 1 To 5
        Summa = Summa + x(i)
    Next i
    Sx = Summa
End Sub

```

```

Sub Find_Pxy()
    P = 1
    For i = 1 To 7
        P = P * x(i) / y(i)
    Next i
    Pxy = P
End Sub

```

```

Sub Find_Z()
    For i = 1 To 10
        If Sx > m * y(i) / n And x(i) / y(i) < 1 Then
            z(i) = x(i) + y(i)
        ElseIf Pxy > 1 Or m / n > 2 Then
            z(i) = x(i)
        Else
            z(i) = y(i)
        End If
        Cells(i + 1, 5) = z(i)
    Next i
End Sub

```

```

Sub Find_minX()
    Min = x(1)
    For i = 1 To 10
        If Min > x(i) Then Min = x(i)
    Next i
    minX = Min
End Sub

```

```

Sub Find_maxY()
    Max = y(1)
    For i = 1 To 10
        If Max < y(i) Then Max = y(i)
    Next i
    maxY = Max
End Sub

```

```

Sub Find_S()
    If minX < maxY Then
        Call Find_Sz
        Cells(2, 10) = Sz
        S = Sqr(Abs(Sz))
    Else
        Call Find_Pz
        Cells(2, 11) = Pz
        S = Pz
    End If
End Sub

```

```

Sub Find_Sz()
    Summa = 0
    For i = 1 To 10
        Summa = Summa + z(i) - 3
    Next i
    Sz = Summa
End Sub

```

```

Sub Find_Pz()
    P = 1
    For i = 1 To 10
        P = P * z(i)
    Next i
    Pz = P
End Sub

```

Варианты задания 2

1. Найти значение функции

$$R = \begin{cases} \sum_{i=1}^{10} z_i + \prod_{i=1}^{10} x_i, & \text{если } \min_{1 \leq i \leq 10} x_i > x_i, \\ \sum_{i=1}^{10} z_i - \prod_{i=1}^{10} y_i & \text{в остальных случаях,} \end{cases} \quad \text{где } z_i = \begin{cases} x_i, & \text{если } \frac{x_i}{y_i} > 1 \text{ или } \sum_{i=1}^{10} y_i < 5, \\ y_i, & \text{если } \frac{x_i}{y_i} < 1 \text{ или } \frac{m}{y_i} < 2, \\ nx_i - y_i & \text{в остальных случаях.} \end{cases}$$

2. Найти значение функции

$$V = \begin{cases} \min_{1 \leq i \leq 10} z_i, & \text{если } m \sum_{i=1}^{10} x_i > n \sum_{i=1}^{10} y_i, \\ \max_{1 \leq i \leq 10} z_i & \text{в остальных случаях,} \end{cases} \quad \text{где } z_i = \begin{cases} x_i, & \text{если } \frac{mx_i}{ny_i} < 1 \text{ или } x_i > y_i, \\ y_i, & \text{если } \frac{mx_i}{ny_i} > 1 \text{ и } \frac{m}{n} > 2, \\ x_i - y_i & \text{в остальных случаях.} \end{cases}$$

3. Найти значение функции

$$S = \begin{cases} \sum_{i=1}^{10} (z_i + x_i), & \text{если } \min_{1 \leq i \leq 10} z_i > \sum_{i=1}^{10} x_i, \\ \prod_{i=1}^7 (z_i + y_i) & \text{в остальных случаях,} \end{cases} \quad \text{где } z_i = \begin{cases} x_i, & \text{если } x_i > y_i \text{ и } \frac{m}{n} \sum_{i=1}^{10} y_i < 5, \\ y_i, & \text{если } x_i < y_i \text{ или } \frac{my_i}{n} < 5, \\ x_i - y_i & \text{в остальных случаях.} \end{cases}$$

4. Напечатать порядковые номера положительных элементов массива, заданных формулой

$$z_i = \begin{cases} \frac{x_i + y_i}{m + 2}, & \text{если } \frac{x_i}{y_i} < 1 \text{ и } \max_{1 \leq i \leq 10} y_i > 0,1 \sum_{i=1}^{10} x_i, \\ \frac{m(x_i - y_i)}{n}, & \text{если } \frac{x_i}{y_i} > 1 \text{ и } \max_{1 \leq i \leq 10} x_i > y_i, \\ x_i y_i & \text{в остальных случаях.} \end{cases}$$

5. Найти наименьший отрицательный элемент массива, заданного формулой

$$z_i = \begin{cases} x_i y_i, & \text{если } \frac{m}{n} > 3 \text{ или } \frac{m}{n} < 0,5, \\ \frac{|x_i - y_i|}{m}, & \text{если } 0,5 < \frac{m}{n} < 3, \\ \frac{x_i}{y_i} & \text{в остальных случаях.} \end{cases}$$

6. Найти значение функции

$$S = \sqrt{m \sum_{i=1}^{10} [(z_i^2 + x_i^2) / y_i^2]}, \quad \text{где } z_i = \begin{cases} \frac{(x_i - y_i)^2}{m}, & \text{если } m > n \text{ или } |x_i| < n, \\ y_i, & \text{если } m < \prod_{i=1}^{10} y_i \text{ или } y_i < n, \\ y_i & \text{в остальных случаях.} \end{cases}$$

7. Найти значение функции

$$P = \sqrt[3]{m \prod_{i=1}^{10} \left(\frac{z_i}{x_i + y_i} \right)^2}, \quad \text{где } z_i = \begin{cases} \frac{(x_i + y_i)^2}{m}, & \text{если } m > \sum_{i=1}^{10} x_i \text{ или } n < 10, \\ \frac{(x_i - y_i)^2}{n}, & \text{если } m < 5 \text{ и } y_i^2 < 10, \\ x_i & \text{в остальных случаях.} \end{cases}$$

8. Найти значение функции

$$t = m \sqrt[3]{10 \sum_{i=1}^{10} \left(\frac{z_i - y_i}{x_i + n} \right)^2}, \quad \text{где } z_i = \begin{cases} \frac{(x_i - y_i)^2}{m}, & \text{если } n < \prod_{i=1}^5 x_i^2 \text{ или } |x_i| + y_i > n, \\ \frac{x_i + y_i}{n}, & \text{если } m < \prod_{i=1}^{10} y_i \text{ и } |y_i| > 2, \\ mx_i & \text{в остальных случаях.} \end{cases}$$

9. Найти значение функции

$$Z = m \sqrt[3]{\prod_{i=1}^{10} \left(\frac{z_i}{x_i + y_i} \right)^2}, \quad \text{где } z_i = \begin{cases} x_i + y_i, & \text{если } m > 5 \sum_{i=1}^{10} (x_i - 1) \text{ или } |x_i| > 5, \\ x_i - y_i, & \text{если } |x_i| < 5 \text{ и } n < 10, \\ x_i y_i & \text{в остальных случаях.} \end{cases}$$

10. Вычислить среднеарифметическое значение отрицательных элементов массивов x_i, y_i, z_i ($i = \overline{1,10}$),

$$\text{где } z_i = \begin{cases} \frac{(x_i + y_i)}{m}, & \text{если } \left| \frac{x_i}{y_i} \right| < 1 \text{ и } n |x_i| < 100, \\ \frac{(x_i - y_i)}{n}, & \text{если } |x_i - y_i| < \frac{m}{n}, \\ x_i - y_i & \text{в остальных случаях.} \end{cases}$$

11. Вычислить значение функции

$$R = \begin{cases} \sum_{i=1}^{10} z_i^2, & \text{если } x_i > y_i, \\ \prod_{i=1}^{10} z_i, & \text{если } x_i < y_i, \end{cases} \quad \text{где } z_i = \begin{cases} \frac{x_i + y_i}{m}, & \text{если } \min_{1 \leq i \leq 10} y_i > x_i, \\ \frac{(x_i - y_i)}{n}, & \text{если } \max_{1 \leq i \leq 10} x_i > y_i, \\ \frac{x_i}{y_i} & \text{в остальных случаях.} \end{cases}$$

12. Вывести на печать порядковые номера положительных элементов массива z_i ($i = \overline{1,10}$),

$$\text{где } z_i = \begin{cases} \frac{(x_i + y_i)}{m}, & \text{если } \left| \frac{x_i}{y_i} \right| < 1 \text{ и } n > 5, \\ \frac{|x_i - y_i|}{2}, & \text{если } \frac{|x_i - y_i|}{2} < \frac{m}{n} \text{ и } n < 5 \sum_{i=1}^{10} x_i, \\ x_i + y_i & \text{в остальных случаях.} \end{cases}$$

13. Вычислить значение функции

$$B = \begin{cases} \sum_{i=1}^7 (z_i + x_i)^2, & \text{если } \min_{1 \leq i \leq 10} x_i > \sum_{i=1}^{10} \frac{x_i}{m}, \\ \prod_{i=1}^{10} (z_i - y_i), & \text{в остальных случаях,} \end{cases} \quad \text{где } z_i = \begin{cases} x_i, & \text{если } m y_i > n x_i, \\ y_i, & \text{если } \sum_{i=1}^{10} \frac{x_i}{y_i} < 1 \text{ и } \frac{n}{m} < 2, \\ x_i + y_i & \text{в остальных случаях.} \end{cases}$$

14. Вычислить значение функции

$$A = \begin{cases} \sum_{i=1}^{10} z_i, & \text{если } \left| \frac{x_i + y_i}{z_i} \right| > 2, \\ \max_{1 \leq i \leq 10} z_i, & \text{если } \left| \frac{x_i + y_i}{z_i} \right| \leq 2 \end{cases} \quad \text{где } z_i = \begin{cases} \frac{|x_i^2 - y_i^2|}{2}, & \text{если } \frac{m x_i}{2} > 2 \text{ и } \frac{m y_i}{2} > 2, \\ x_i + y_i, & \text{если } \frac{m x_i}{2} < 2 \text{ или } n_{uu} < 2, \\ x_i y_i & \text{в остальных случаях.} \end{cases}$$

15. Найти значение функции

$$B = \begin{cases} \sum_{i=1}^{10} z_i, & \text{если } \max_{1 \leq i \leq 10} z_i > z_{10}, \\ \prod_{i=1}^{10} (z_i - y_i), & \text{если } \max_{1 \leq i \leq 10} z_i > z_{10}, \end{cases} \quad \text{где } z_i = \begin{cases} x_i, & \text{если } \sum_{i=1}^{10} x_i > 5 y_i, \\ y_i, & \text{если } x_i < y_i \text{ и } m > n, \\ x_i + y_i & \text{в остальных случаях.} \end{cases}$$

16. Вычислить значение функции

$$L = \begin{cases} \sqrt{\sum_{i=1}^{10} (z_i - 3)}, & \text{если } \min_{1 \leq i \leq 10} x_i > \min_{1 \leq i \leq 10} y_i, \\ \prod_{i=1}^{10} z_i, & \text{если } \min_{1 \leq i \leq 10} x_i \leq \min_{1 \leq i \leq 10} y_i, \end{cases}$$

$$\text{где } z_i = \begin{cases} x_i + y_i, & \text{если } \sum_{i=1}^6 x_i > \frac{m y_i}{n} \text{ и } \frac{y_i}{x_i} < 1, \\ y_i, & \text{если } \frac{y_i}{n} > 1 \text{ и } \frac{m}{n} > 2, \\ x_i & \text{в остальных случаях} \end{cases}$$

17. Вычислить среднеарифметическое значение положительных элементов массивов x_i, z_i ($i = \overline{1,10}$),

$$\text{где } z_i = \begin{cases} \frac{x_i + y_i}{m}, & \text{если } \min_{1 \leq i \leq 10} y_i > x_i \text{ и } m x_i > n y_i, \\ \frac{(x_i - y_i)}{n}, & \text{если } \min_{1 \leq i \leq 10} y_i < x_i \text{ или } m x_i < n y_i, \\ \frac{n x_i}{m y_i} & \text{в остальных случаях} \end{cases}$$

18. Вычислить значение функции

$$L = \begin{cases} \sqrt{\sum_{i=1}^{10} z_i^2}, & \text{если } \sum_{i=1}^5 x_i > 10, \\ \prod_{i=1}^{10} \frac{x_i + y_i}{z_i}, & \text{если } \sum_{i=1}^5 x_i \leq 10, \end{cases} \quad \text{где } z_i = \begin{cases} x_i + y_i, & \text{если } m y_i > n x_i \text{ и } \frac{x_i}{n} < 0,1, \\ |x_i - y_i|, & \text{если } m y_i \leq n x_i \text{ или } y_i < 0, \\ \frac{x_i}{y_i} & \text{в остальных случаях} \end{cases}$$

19. Найти значение функции

$$A = \begin{cases} \max_{1 \leq i \leq 10} z_i + \min_{1 \leq i \leq 10} y_i, & \text{если } \sum_{i=1}^{10} x_i > 10, \\ \max_{1 \leq i \leq 10} z_i - \min_{1 \leq i \leq 10} y_i, & \text{если } \sum_{i=1}^{10} x_i \leq 10 \end{cases} \quad \text{где } z_i = \begin{cases} x_i + y_i, & \text{если } \left| \frac{x_i}{y_i} \right| < 1 \text{ или } |y_i| > 5, \\ \frac{|x_i - 2y_i|}{2}, & \text{если } \left| \frac{x_i}{y_i} \right| \geq 1 \text{ и } n < 5 \sum_{i=1}^{10} x_i, \\ \left| \frac{x_i}{y_i} \right| & \text{в остальных случаях} \end{cases}$$

20. Вычислить значение функции

$$U = \begin{cases} \sqrt{\sum_{i=1}^{10} (z_i + y_i)^2}, & \text{если } \sum_{i=1}^5 x_i > \prod_{i=1}^{10} y_i, \\ \prod_{i=1}^{10} z_i - y_i, & \text{если } \sum_{i=1}^5 x_i \leq \prod_{i=1}^{10} y_i, \end{cases} \quad \text{где } z_i = \begin{cases} x_i - y_i, & \text{если } \left| \frac{x_i}{y_i} \right| < 1 \text{ и } |y_i| < 5, \\ \frac{|x_i + 2y_i|}{2}, & \text{если } \left| \frac{x_i}{y_i} \right| \geq 1 \text{ или } n > \sum_{i=1}^{10} x_i, \\ \left| \frac{x_i}{y_i} \right| + 0,25 & \text{в остальных случаях} \end{cases}$$

21. Вычислить значение функции

$$R = \begin{cases} \sum_{i=1}^{10} z_i + \prod_{i=1}^{10} x_i, & \text{если } \min_{1 \leq i \leq 10} x_i > x_i, \\ \sum_{i=1}^{10} z_i - \prod_{i=1}^{10} y_i & \text{в остальных случаях,} \end{cases} \quad \text{где } z_i = \begin{cases} x_i, & \text{если } \frac{x_i}{y_i} > 1 \text{ или } \sum_{i=1}^{10} y_i < 5, \\ y_i, & \text{если } \frac{x_i}{y_i} < 1 \text{ или } \frac{m}{y_i} < 2, \\ nx_i - y_i & \text{в остальных случаях.} \end{cases}$$

22. Найти значение функции

$$V = \begin{cases} \min_{1 \leq i \leq 10} z_i, & \text{если } m \sum_{i=1}^{10} x_i > n \sum_{i=1}^{10} y_i, \\ \max_{1 \leq i \leq 10} z_i & \text{в остальных случаях,} \end{cases} \quad \text{где } z_i = \begin{cases} x_i, & \text{если } \frac{mx_i}{ny_i} < 1 \text{ или } x_i > y_i, \\ y_i, & \text{если } \frac{mx_i}{ny_i} > 1 \text{ и } \frac{m}{n} > 2, \\ x_i - y_i & \text{в остальных случаях.} \end{cases}$$

23. Напечатать порядковые номера положительных элементов массива, заданных формулой

$$z_i = \begin{cases} \frac{x_i + y_i}{m + 2}, & \text{если } \frac{x_i}{y_i} < 1 \text{ и } \max_{1 \leq i \leq 10} y_i > 0,1 \sum_{i=1}^{10} x_i, \\ \frac{m(x_i - y_i)}{n}, & \text{если } \frac{x_i}{y_i} > 1 \text{ и } \max_{1 \leq i \leq 10} x_i > y_i, \\ x_i y_i & \text{в остальных случаях.} \end{cases}$$

24. Найти значение функции

$$S = \begin{cases} \sum_{i=1}^{10} (z_i + x_i), & \text{если } \min_{1 \leq i \leq 10} z_i > \sum_{i=1}^{10} x_i, \\ \prod_{i=1}^7 (z_i + y_i) & \text{в остальных случаях,} \end{cases} \quad \text{где } z_i = \begin{cases} x_i, & \text{если } x_i > y_i \text{ и } \frac{m}{n} \sum_{i=1}^{10} y_i < 5, \\ y_i, & \text{если } x_i < y_i \text{ или } \frac{my_i}{n} < 5, \\ x_i - y_i & \text{в остальных случаях.} \end{cases}$$

25. Найти наименьший отрицательный элемент массива, заданного формулой

$$z_i = \begin{cases} x_i y_i, & \text{если } \frac{m}{n} > 3 \text{ или } \frac{m}{n} < 0,5, \\ \frac{|x_i - y_i|}{m}, & \text{если } 0,5 < \frac{m}{n} < 3, \\ \frac{x_i}{y_i} & \text{в остальных случаях.} \end{cases}$$

26. Найти значение функции

$$S = \sqrt{m \sum_{i=1}^{10} [(z_i^2 + x_i^2) / y_i^2]}, \quad \text{где } z_i = \begin{cases} \frac{(x_i - y_i)^2}{m}, & \text{если } m > n \text{ или } |x_i| < n, \\ y_i, & \text{если } m < \prod_{i=1}^{10} y_i \text{ или } y_i < n, \\ y_i & \text{в остальных случаях} \end{cases}$$

27. Найти значение функции

$$P = \sqrt[3]{m \prod_{i=1}^{10} \left(\frac{z_i}{x_i + y_i} \right)^2}, \quad \text{где } z_i = \begin{cases} \frac{(x_i + y_i)^2}{m}, & \text{если } m > \sum_{i=1}^{10} x_i \text{ или } n < 10, \\ \frac{(x_i - y_i)^2}{n}, & \text{если } m < 5 \text{ и } y_i^2 < 10, \\ x_i & \text{в остальных случаях} \end{cases}$$

28. Вычислить значение функции

$$t = m \sqrt[3]{10 \sum_{i=1}^{10} \left(\frac{z_i - y_i}{x_i + n} \right)^2}, \quad \text{где } z_i = \begin{cases} \frac{(x_i - y_i)^2}{m}, & \text{если } n < \prod_{i=1}^5 x_i^2 \text{ или } |x_i| + y_i > n, \\ \frac{x_i + y_i}{n}, & \text{если } m < \prod_{i=1}^{10} y_i \text{ и } |y_i| > 2, \\ mx_i & \text{в остальных случаях} \end{cases}$$

29. Найти значение функции

$$Z = m \sqrt[3]{\prod_{i=1}^{10} \left(\frac{z_i}{x_i + y_i} \right)^2}, \quad \text{где } z_i = \begin{cases} x_i + y_i, & \text{если } m > 5 \sum_{i=1}^{10} (x_i - 1) \text{ или } |x_i| > 5, \\ x_i - y_i, & \text{если } |x_i| < 5 \text{ и } n < 10, \\ x_i y_i & \text{в остальных случаях} \end{cases}$$

30. Вычислить среднеарифметическое значение отрицательных элементов массивов x_i, y_i, z_i ($i = \overline{1, 10}$),

$$\text{где } z_i = \begin{cases} \frac{(x_i + y_i)}{m}, & \text{если } \left| \frac{x_i}{y_i} \right| < 1 \text{ и } n |x_i| < 100, \\ \frac{(x_i - y_i)}{n}, & \text{если } |x_i - y_i| < \frac{m}{n}, \\ x_i - y_i & \text{в остальных случаях} \end{cases}$$

3. Программирование функций и построение графиков с использованием объектной модели Excel

Объектная модель Excel является основой программирования в Excel на VBA. Приложение Excel состоит из объектов, обеспечивающих функциональность программы. Эти объекты являются «открытыми», т.е. ими можно управлять извне, и их функциональные возможности могут использовать программы VBA. Совокупность объектов Excel, открытых для использования другими приложениями, и называется объектной моделью Excel.

Объект в VBA – это структура данных, представляющая совокупность свойств и методов. Свойства объекта – это информация об объекте в виде значений связанных с ним переменных или других объектов. Методы объекта представляют собой действия, выполняемые объектом и существуют в виде связанных с объектом функций. Обозначение свойств и методов выполняется в стандартном формате (точка – операция доступа к свойствам и методам объекта):

ИмяОбъекта.Свойство

ИмяОбъекта.Метод

Значения свойств объекта задаются оператором присваивания (=).

Параметры в функцию-метод могут передаваться тремя способами:

1) Указание значений параметров в скобках в определенном порядке

ИмяОбъекта.Метод(аргумент1, аргумент2, ...)

2) Указание значений параметров без скобок в определенном порядке

ИмяОбъекта.Метод аргумент1, аргумент2, ...

3) Указание значений именованных аргументов в произвольном и неполном порядке с помощью оператора (:=)

ИмяОбъекта.Метод ИмяАргумента1 := аргумент1, ...

Объектная модель Excel организована в виде иерархической структуры. Верхним узлом является объект Application, представляющий собственно Excel. Все другие объекты являются дочерними, наиболее важный из них – коллекция Workbooks, содержащая все открытые рабочие книги Workbook. Каждый объект Workbook имеет свою коллекцию Sheets, содержащую рабочие листы Sheet и листы диаграмм Chart. Рабочий лист может содержать такие объекты как диапазон ячеек Range и выделение Selection.

Для использования именованного объекта он должен быть объявлен с указанием его типа:

Dim ИмяОбъекта As ТипОбъекта

Объект создается специальным оператором

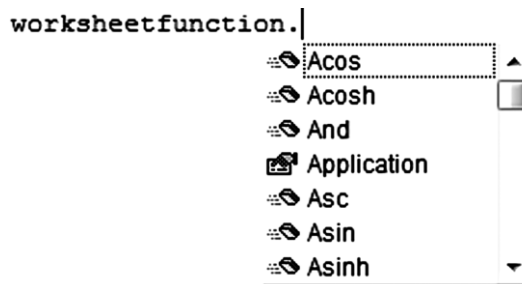
Set ИмяОбъекта = New ТипОбъекта

Элементарное взаимодействие табличного процессора Excel и VBA-кода может включать в себя:

- программирование функций, доступных для табличных вычислений в Excel;
- использование функций Excel кодом программы VBA;
- использование табличных вычислений Excel кодом программы VBA;
- генерация кода VBA табличным процессором при записи макросов.

Функция пользователя, доступная для вычислений в таблице Excel, создается в VBA с помощью кода процедуры-функции, размещенном в стандартном модуле. Стандартный модуль вставляется в проект VBA через меню редактора Insert → Module и по умолчанию имеет имя Module1, которое при необходимости можно изменить. После ввода кода функция появляется в списке Мастера функций в категориях «Определенные пользователем» и «Полный алфавитный перечень». При вставке функции в ячейку открывается диалоговое окно, содержащее поля ввода для каждого аргумента функции.

Код VBA может использовать функции табличного процессора. Эти функции являются методами объекта `Application.WorksheetFunction`. При этом необходимо вводить имя функции в нелокализованной (англоязычной) форме. При вводе точки после `WorksheetFunction` открывается список доступных функций листа. Функции табличного процессора, аналогичные функциям VBA, например, квадратный корень, недоступны и отсутствуют в списке.



Чтобы вставить формулу в ячейку таблицы в процессе выполнения программы VBA, нужно присвоить строку, содержащую запись формулы (как если бы она вводилась в ячейку), свойству `Formula` или `FormulaLocal` (если формула содержит символы, отличные от латинских) соответствующего объекта `Range`. Формула может содержать функции Excel. Результат формулы может быть возвращен программе считыванием значения из ячейки.

```
Range("B2").Formula = "=LOG10(A2)"  
y = Cells(2, 2)
```

Запись макроса позволяет генерировать код VBA при выполнении действий в табличном процессоре. Запись включается при нажатии соответствующей кнопки на вкладке **Разработчик**. После нажатия кнопки «Остановить запись» в проекте VBA появляется модуль, содержащий код. Его можно использовать самостоятельно или скопировать в программу.

Задание 3.1. Создать модуль функции пользователя $f(x)$ и построить ее график в интервале $[x_1; x_2]$. Использовать в модуле функции Excel.

Задание 3.2. Создать макрос построения графика с вводом всех параметров и формулы в процессе его выполнения

Варианты заданий 3.1 и 3.2

варианты	$y = f(x)$	a	b	x_1	x_2	Δx
1	$y = \frac{1 + \sin^2(b^3 + x^3)}{\sqrt[3]{b^3 + x^3}}$	-	2.5	1.28	3.28	0.01
2	$y = \frac{\sqrt[3]{ax+b}}{\lg^2 x}$	1.35	0.98	1.14	4.24	0.01
3	$y = \frac{1 + \lg^2 \frac{x}{a}}{b - e^{\frac{x}{a}}}$	2.0	0.95	1.25	2.75	0.01
4	$y = \sqrt[4]{x^2 - 2,5} + \sqrt[3]{\lg x^2}$	-	-	1.25	3.25	0.01
5	$y = \frac{\lg^2(a^2 + x)}{(a + x)^2}$	- 2.5	3.4	5.5	6.5	0.01
6	$y = \frac{(a + bx)^{2,5}}{1,8 + \cos^3(ax)}$	- 0.25	3.4	5.5	6.5	0.01
7	$y = \frac{a^x - b^x}{\lg \frac{a}{b}} \sqrt[3]{ab}$	0.4	0.8	3.2	6.2	0.01
8	$y = \frac{b^3 + \sin^2 ax}{\arccos(bx) + e^{-x/2}}$	1.2	0.48	0.5	2.0	0.01
9	$y = \frac{\lg(x^2 - 1)}{\ln(ax^2 - b)}$	1.1	0.09	1.2	2.2	0.01
10	$y = \frac{\arccos(x^2 - b^2)}{\arcsin(x^2 - a^2)}$	0.05	0.06	0.2	0.95	0.005
11	$y = \arcsin(x^a) + \arccos(x^b)$	2.0	3.0	0.11	0.36	0.005
12	$y = a^{x^2-1} - \lg(x^2 - 1) + \sqrt[3]{x^2 - 1}$	1.6	-	1.2	3.7	0.01
13	$y = \frac{a\sqrt{x} - b \ln x}{\lg x-1 }$	4.1	2.7	2.2	5.2	0.01
14	$y = \sqrt{\frac{ a-bx }{\lg^3 x}}$	7.2	4.2	1.81	5.31	0.01

15	$y = (\arcsin^2 x + \arccos^4 x)^3$	-	-	0.26	0.66	0.005
16	$y = \frac{\ln b^2 - x^2 }{\sqrt[5]{ x^2 - a^2 }}$	2.0	1.1	0.08	1.08	0.01
17	$y = \frac{a + \operatorname{tg}^2 bx}{b + \operatorname{ctg}^2 ax}$	0.1	0.5	0.15	1.35	0.01
18	$y = \frac{(a + bx)^{2.5}}{1 + \lg(a + bx)}$	2.5	4.6	1.1	3.6	0.01
19	$y = \frac{\lg^2(a + x)}{(a + x)^2}$	2.0	-	1.2	4.2	0.01
20	$y = \frac{\sqrt[3]{(x - a)^2} + \sqrt[5]{ x + b }}{\sqrt[9]{x^2 - (a + b)^2}}$	0.8	0.4	1.23	7.23	0.05
21	$y = (\sin^3 x + \cos^3 x) \ln x$	-	-	0.11	0.36	0.005
22	$y = a^{x^2-1} - \lg(x^2 - 1) + \sqrt[3]{x^2 - 1}$	2.25	-	1.2	2.7	0.01
23	$y = \frac{a\sqrt[3]{x} - b \lg x}{\lg^3(x - 1)}$	4.1	2.7	1.5	1.9	0.01
24	$y = \sqrt[5]{\frac{a + bx}{\lg^3 x}}$	7.2	1.3	1.56	4.71	0.05
25	$y = \sqrt[7]{\arcsin^4 x + \arccos^6 x}$	-	-	0.22	0.92	0.01
26	$y = \frac{\ln(b^2 - x^2)}{\sqrt[3]{ x^2 - a^2 }}$	2.0	4.1	0.77	1.77	0.01
27	$y = \frac{\sqrt[3]{a} + \operatorname{tg}^{4.5} bx}{\sqrt[5]{b} + \operatorname{ctg}^{2.7} ax}$	0.1	0.5	0.33	1.23	0.01
28	$y = \frac{\sin(a + bx)^4}{1 + \cos(a + bx)}$	2.5	4.6	1.15	3.05	0.05
29	$y = \frac{\operatorname{tg}[\lg^3(a + x)]}{\sqrt[7]{(a + x)^2}}$	2.0	-	1.08	1.88	0.01
30	$y = \frac{\sqrt[3]{x - a} + \sqrt[5]{x + b}}{\sqrt[7]{x} - \sqrt[9]{x^2 - (a + b)^2}}$	0.8	0.4	1.42	1.92	0.01

Пример 3.1. Создать модуль функции пользователя $f(x)$ и построить ее график в интервале $[x_1; x_2]$. Использовать в модуле функции Excel.

Исходные данные

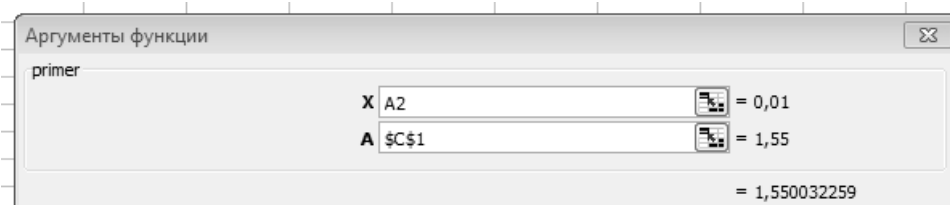
$y = f(x)$	a	b	x_1	x_2	Δx
$y = \sqrt{\arcsin^2 x + a^2}$	1,55	-	0	0,5	0,01

Порядок работы.

- 1) Вставить стандартный модуль через меню редактора Insert → Module.
- 2) В модуль ввести код функции

```
Function primer(x, a)
    primer = Sqr(WorksheetFunction.Asin(x) ^ 2 + a ^ 2)
End Function
```

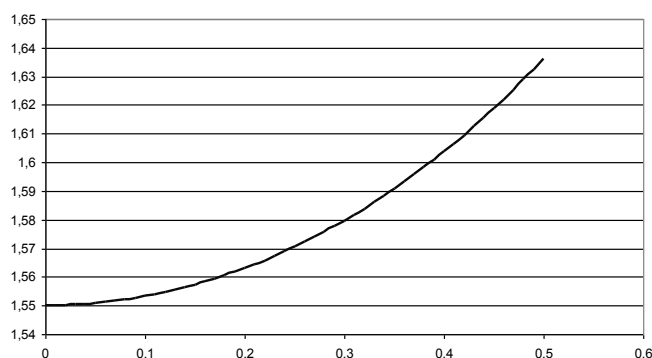
- 3) Открыть лист и ввести в столбец A последовательность значений переменной $x \in [0; 0,5]$ с шагом 0,01.
- 4) Ввести в ячейку C1 значение параметра $a = 1,55$.
- 5) Вставить в ячейку B1 формулу. Откроется мастер функций. Найти в списке функций пользователя primer. В открывшемся диалоговом окне задать ссылки на переменную x (A1) и параметр a (\$C\$1).



- 6) Скопировать формулу в ячейки столбца B.

	A	B	C
1	0	1,55	1,55
2	0,01	1,550032	
3	0,02	1,550129	
4	0,03	1,55029	

- 6) Выделить данные в столбцах AB и вставить точечную диаграмму.



Пример 3.2. Создать макрос построения графика с вводом всех параметров и формулы в процессе его выполнения

Исходные данные

$y = f(x)$	a	b	x_1	x_2	Δx
$y = \frac{x}{\sqrt{x^2 + a^2}}$	2,5	-	0	5	0,5

Алгоритм программы. Все параметры и формула функции $f(x)$ вводятся в процессе выполнения программы. Для упрощения ввода используются именованные ячейки. Ссылки на них в формуле выглядят как имена переменных. Формула функции размещается в ячейке B2. В ячейку B1 заносятся текущие значения переменной x . При этом автоматически происходит вычисление $f(x)$ в ячейке B2. После этого пара значений x , $f(x)$ копируются в очередную строку таблицы. После построения таблицы строится график $f(x)$.

Порядок работы.

1) В модуль листа ввести код и выполнить программу

```
Sub InputAll()
```

```
    'заголовки столбцов
```

```
    Cells(1, 1) = "x"
```

```
    Cells(1, 2) = "y"
```

```
    Cells(1, 3) = "a"
```

```
    'вставка формулы, введенной в окно ввода,
```

```
    'в ячейку таблицы
```

```
    Cells(2, 2).FormulaLocal = InputBox("формула f(x)")
```

```
    'ввод параметров задачи
```

```
    x1 = InputBox("x1=")
```

```
    x2 = InputBox("x2=")
```

```
    h = InputBox("h=")
```

```
    a = InputBox("a=")
```

```
    Cells(2, 3) = a
```

```
    'создание именованных ячеек
```

```
    Range("A2").Name = "x"
```

```
    Range("C2").Name = "a"
```

```
    'определяем число точек
```

```
    Dim n As Integer
```

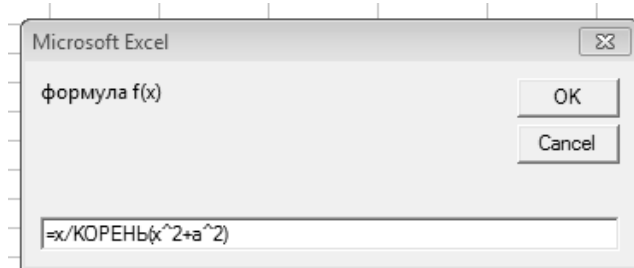
```
    n = (x2 - x1) / h
```

```

'цикл построения таблицы x и f(x)
For i = 0 To n
    x = x1 + h * i      'текущее значение x
    Range("x") = x      'в ячейку "x"
    Cells(i + 3, 1) = x      'копируем x и f(x)
    Cells(i + 3, 2) = Cells(2, 2) 'в строку таблицы
Next i
'код макроса построения графика
,
End Sub

```

2) В процессе выполнения программы в открывающиеся диалоговые окна ввести необходимые данные:



2) Открыть лист с построенной таблицей и записать макрос построения точечной диаграммы без маркеров. Данные для графика расположены с третьей строки. Завершить запись.

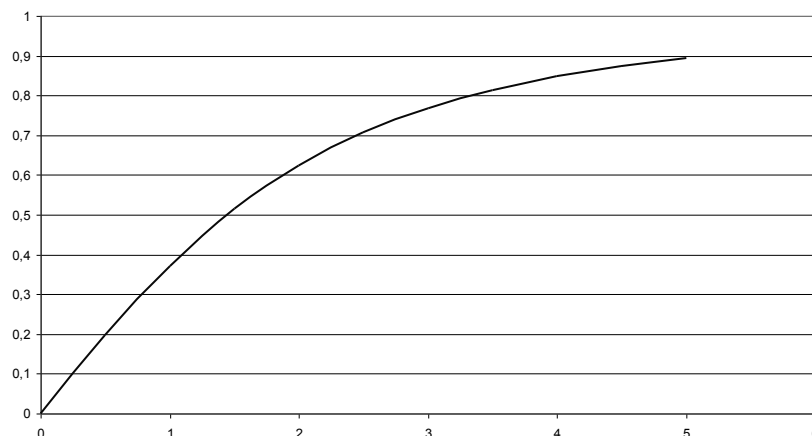
3) Открыть код записанного модуля и отредактировать его, изменив диапазон данных на "A3:B"&n+3. Скопировать код модуля и вставить его в программу.

```

'код макроса построения графика
ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SetSourceData Source := Range("A3:B"&n+3)

```

4) Очистить лист и выполнить макрос.



Задание 3.3. Создать макрос построения графика функции, заданной параметрически.

Пример 3.3. Построить полярные кривые $r = |20 + a| \cos^2 \varphi - \sin^2 \varphi|$, $\varphi \in [0, 2\pi]$, $\Delta\varphi = \pi/36$, для значений параметра $a = 60, 40, 20, 0$.

Задание функции в параметрическом виде $x = x(t)$, $y = y(t)$ позволяет строить сложные кривые, содержащие самопересечения и описывающие многозначные функции. Для построения графика функции, заданной параметрически, в Excel достаточно вычислить декартовы координаты точек графика x , y и вывести их в виде таблицы. График Excel соединяет точки сплайном последовательно. Если график состоит из нескольких ветвей (кривых), между соответствующими частями таблицы должны быть свободные ячейки.

Полярные координаты точки связаны с декартовыми соотношением

$$x = r \cos \varphi$$

$$y = r \sin \varphi$$

Порядок работы:

1) В модуль листа ввести код и выполнить программу

```
Sub Polar()  
    N = 72 'число точек кривой  
    pi = 3.1415927  
    d = 2 * pi / N 'шаг по углу fi  
    i = 1 'номер строки таблицы  
    For a = 60 To 0 Step -20 'цикл по параметру a  
        Cells(i, 3) = a 'вывод значения a  
        For k = 0 To N 'цикл по углу fi  
            fi = k * d 'текущее значение угла fi  
            r = Abs(20 + a * Abs(Cos(fi)^2 - Sin(fi)^2))  
            x = r * Cos(fi)  
            y = r * Sin(fi)  
            Cells(i, 1) = x  
            Cells(i, 2) = y  
            i = i + 1  
        Next k  
        i = i + 1 'пропуск ячейки  
    Next a  
    Call graph("A1:B" & i - 2) 'вызов процедуры  
                                'построения графика  
End Sub
```

```

Sub graph(rng As String)
    'процедура построения графика
    'место для вставки макроса
End Sub

```

2) Открыть лист с построенной таблицей и записать макрос построения точечной диаграммы. Если диаграмма сильно деформирована (масштаб по осям различен), в процессе записи макроса выделить область построения графика и изменить ее. Завершить запись.

3) Открыть код записанного модуля и отредактировать его, установив нужные значения ширины (Height = 250), высоты (Width = 250) и положения левого верхнего угла графика (Top = 15), а также изменив диапазон данных на аргумент процедуры graph(rng). Скопировать код модуля и вставить его в процедуру graph.

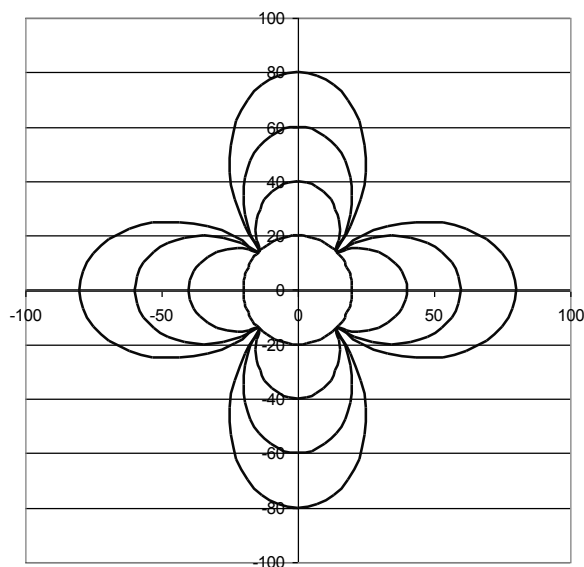
```

ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SetSourceData Source:= Range(rng)
                                                                    'изменить

ActiveChart.HasLegend = False
ActiveChart.PlotArea.Select
Selection.Height = 250
                                                                    'изменить
Selection.Width = 250
                                                                    'изменить
Selection.Top = 15
                                                                    'изменить

```

4) Очистить лист и выполнить макрос.



Варианты задания 3.3

Нечетные варианты (1, 3, 5, ...). Рассчитать x , y –координаты точек кривых, заданных в параметрической форме, для параметра t , изменяющегося на данном отрезке $[t_{нач}; t_{кон}]$ с шагом Δt , при каждом из указанных значений α

№	Параметрическая кривая	Значения аргумента			Значение α
		$t_{нач}$	$t_{кон}$	Δt	
1	$x(t) = \alpha \cos(t)$ $y(t) = \frac{\alpha \sin(t)}{2}$	0	2π	$\pi/36$	30; 40; ...; 60
3	$x(t) = \sin(\alpha t)$ $y(t) = \frac{\sin(4t)}{2}$	0	2π	$\pi/72$	1; 2; ...; 4
5	$x(t) = 40(t - \alpha \sin(t))$ $y(t) = 40(1 - \alpha \cos(t))$	0	2π	$\pi/36$	1; 0,75; ...; 0,25
7	$x(t) = 80 \cos(t)$ $y(t) = 50 \sin(t + \alpha)$	0	2π	$\pi/36$	0; $\pi/8$; ...; $\pi/2$
9	$x(t) = 40 \cos(t) + 20 \alpha \cos(2t)$ $y(t) = 40 \sin(t) + 20 \alpha \sin(2t)$	0	2π	$\pi/72$	0,5; 1; ...; 2
11	$x(t) = 45 \cos(t) + 15 \alpha \cos(3t)$ $y(t) = 45 \sin(t) + 15 \alpha \sin(3t)$	0	2π	$\pi/36$	30; 40; ...; 60
13	$x(t) = 60 \cos(t) + 15 \alpha \cos(4t)$ $y(t) = 60 \sin(t) + 15 \alpha \sin(4t)$	0	2π	$\pi/36$	0,5; 1,0; ...; 3,0
15	$x(t) = 20 \alpha (2 + \cos(t))$ $y(t) = 20 \alpha \sin(t)$	0	2π	$\pi/36$	30; 40; ...; 60
17	$x(t) = (120 - 20 \alpha) \cos(t) - 20 \alpha$ $y(t) = \frac{2}{3} (120 - 20 \alpha) \sin(t)$	0	2π	$\pi/36$	0; 1; ...; 4
19	$x(t) = (4 + \alpha)t + (12 - 3\alpha)t^2 + (2\alpha - 8)t^3$ $y(t) = (4 + \alpha)(1 - t)t$	0	1	0,02	0; 7,5; ...; 30
21	$x(t) = \alpha t - \alpha \sin(t)$ $y(t) = \alpha - \alpha \cos(t)$	0	2π	$\pi/72$	1; 2; ...; 4
23	$x(t) = \alpha (\cos(t) + t \sin(t))$ $y(t) = \alpha (\sin(t) - t \cos(t))$	0	2π	$\pi/72$	1; 2; ...; 4
25	$x(t) = \alpha (t + \pi + \sin(t))$ $y(t) = \alpha (1 - \cos(t))$	$-\pi$	π	$\pi/36$	1; 2; ...; 4

27	$x(t) = \alpha \cos(t)$ $y(t) = \frac{\alpha \sin(t)}{2}$	0	2π	$\pi/36$	30; 40; ...; 60
29	$x(t) = \sin(\alpha t)$ $y(t) = \frac{\sin(4t)}{2}$	0	2π	$\pi/36$	30; 40; ...; 60

Четные варианты (2, 4, 6, ...). Рассчитать координаты точек кривых, заданных в полярных координатах, для полярного угла φ , изменяющегося на данном отрезке $[\varphi_{нач}; \varphi_{кон}]$ с шагом $\Delta\varphi$, при каждом из указанных значений α

№	Кривая в полярных координатах	Значения аргумента			Значение α
		$\varphi_{нач}$	$\varphi_{кон}$	$\Delta\varphi$	
2	$r = \alpha\varphi$	0	2π	$\pi/36$	5; 7,5; ...; 15
4	$r = 30\cos\varphi + \alpha$	0	2π	$\pi/72$	10; 25; ...; 70
6	$r = \alpha\sqrt{2\cos 2\varphi}$	$-\pi/4$	$\pi/4$	$\pi/36$	1; 0,75; ...; 0,25
8	$r = \alpha \sin 2\varphi$	0	2π	$\pi/36$	80; 70; ...; 50
10	$r = 20 + \alpha \cos^2 \varphi - \sin^2 \varphi$	0	2π	$\pi/36$	0; 20; ...; 60
12	$r = \alpha \sin 3\varphi$	0	π	$\pi/36$	80; 70; ...; 50
14	$r = 20 + \alpha \cos^2 \varphi - \sin^2 \varphi$	0	2π	$\pi/36$	60; 40; ...; 0
16	$r = \alpha\varphi$	0	2π	$\pi/36$	20; 40; ...; 80
18	$r = \alpha / \sqrt[4]{\cos^4 \varphi + \sin^4 \varphi}$	0	2π	$\pi/36$	80; 60; ...; 20
20	$r = \alpha(\cos \varphi + 1)$	0	2π	$\pi/36$	20; 40; ...; 80
22	$r = (1 + \sin \varphi)(1 + \alpha \cos 8\varphi) \cdot (1 + (1 - \alpha) \cos 24\varphi)$	0	2π	$\pi/180$	0,5; 0,75; 0,9
24	$r = (1 + \sin \varphi)(1 + 0,9 \cos \alpha\varphi) \cdot (1 + 0,1 \cos 3\alpha\varphi)$	0	2π	$\pi/180$	2; 4; ...; 8
26	$r = \alpha\varphi$	0	2π	$\pi/36$	15; 12,5; ...; 5
28	$r = 30\cos\varphi + \alpha$	0	2π	$\pi/72$	70; 55; ...; 10
30	$r = \alpha\sqrt{2\cos 2\varphi}$	$-\pi/4$	$\pi/4$	$\pi/36$	50; 60; ...; 80

Задание 3.4. Параметрическая сплайн-интерполяция.

Контур детали задан координатами точек.

1. Аппроксимировать контур параметрическим сплайном.
2. Найти площадь, длину контура и координаты центра масс.
3. Построить чертеж, вычислив 20-30 точек контура.
4. Дополнить чертеж изображением центра масс.

Сплайн-интерполяция функции $y = y(x)$, заданной таблицей значений в узлах $(x_i; y_i)$, определяет набор функций-сплайнов $f_i(x)$, аппроксимирующих $y(x)$ на интервалах $x_{i-1} \leq x < x_i$, $i = 0, 1, 2, \dots, n$.

i	x_i	y_i
0	x_0	y_0
1	x_1	y_1
2	x_2	y_2
...
n	x_n	y_n

$$y(x) = \begin{cases} f_1(x) & x_0 \leq x < x_1 \\ f_2(x) & x_1 \leq x < x_2 \\ \dots & \dots \\ f_n(x) & x_{n-1} \leq x < x_n \end{cases}$$

Если применить кубические сплайны, то

$$f_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3.$$

Введем обозначения $h_i = x_i - x_{i-1}$,

тогда в пределах каждого из сплайнов $0 \leq x - x_{i-1} < h_i$.

Из условия непрерывности функции $f(x)$ следует $2n$ уравнений:

$$f_i(x_{i-1}) = y_{i-1}$$

$$f_i(x_i) = y_i \quad i = 0, 1, 2, \dots, n.$$

Или

$$a_i = y_{i-1}$$

$$a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i \quad i = 0, 1, 2, \dots, n.$$

Из условия непрерывности 1-й производной функции $f(x)$ следует $n-1$ уравнений:

$$f'_i(x_i) = f'_{i+1}(x_i) \quad i = 1, 2, \dots, n-1.$$

Т.к. $f'_i = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2$, то

$$b_i + 2c_i(x_i - x_{i-1}) + 3d_i(x_i - x_{i-1})^2 = b_{i+1} \quad \text{или}$$

$$b_i + 2c_i h_i + 3d_i h_i^2 - b_{i+1} = 0 \quad i = 1, 2, \dots, n-1.$$

Из условия непрерывности 2-й производной функции $f(x)$ следует $n-1$ уравнений:

$$f''_i(x_i) = f''_{i+1}(x_i) \quad i = 1, 2, \dots, n-1.$$

Т.к. $f''_i = 2c_i + 6d_i(x - x_{i-1})$, то

$$2c_i + 6d_i(x_i - x_{i-1}) = 2c_{i+1} \quad \text{или}$$

$$c_i + 3d_i h_i - c_{i+1} = 0 \quad i = 1, 2, \dots, n-1.$$

Получаем $2n + (n-1) + (n-1) = 4n-2$ уравнения относительно $4n$ неизвестных. Оставшиеся два уравнения задают, фиксируя значения производных на концах кривой, например так:

$$f''(x_0) = 0 \quad f''(x_n) = 0,$$

или

$$c_1 = 0 \\ c_n + 3d_n h_n = 0.$$

Полученные уравнения представляют собой систему линейных алгебраических уравнений относительно $4n$ неизвестных a_i, b_i, c_i, d_i , ($i = 0, 1, \dots, n$), которую представим в матричной форме:

$$M \cdot X = V,$$

где матрицы имеют следующий вид:

$$\begin{array}{c} a_1 \quad b_1 \quad c_1 \quad d_1 \quad c_2 \quad d_2 \quad c_2 \quad d_2 \quad \dots \quad c_n \quad d_n \\ \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ \dots \\ 2n+1 \\ 2n+2 \\ 2n+3 \\ 2n+4 \\ \dots \\ 4n-1 \\ 4n \end{array} \left[\begin{array}{cccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & h_1 & h_1^2 & h_1^3 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & h_2 & h_2^2 & h_2^3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & 2h_1 & 3h_1^2 & 0 & -1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 3h_1 & 0 & 0 & -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2h_2 & 3h_2^2 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3h_2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 3h_n \end{array} \right] = \left[\begin{array}{c} y_0 \\ y_1 \\ y_1 \\ y_2 \\ \dots \\ 0 \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \end{array} \right] \end{array}$$

Решение системы имеет вид

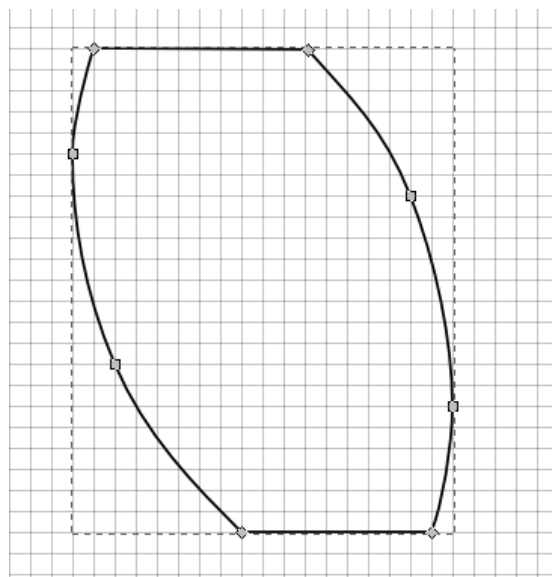
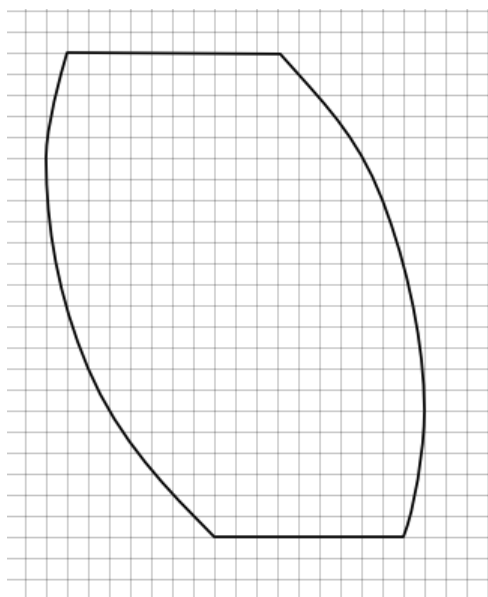
$$X = M^{-1} \cdot V.$$

и может быть найдено с помощью функций МОБР() и МУМНОЖ().

Для параметрических сплайнов необходимо ввести вдоль кривой параметр t , значения которого различны для разных точек кривой, и выполнить интерполяцию для $x = x(t)$ и $y = y(t)$. Для каждой криволинейной дуги контура интерполяция выполняется независимо. Прямолинейные отрезки контура интерполировать не нужно.

Порядок работы:

- 1) Определить по чертежу несколько (5-10) точек фигуры, выбрав начало координат левее и ниже контура.



- 2) Координаты точек представить в виде таблицы.
- 3) Разделить кривую на дуги, разделенные угловыми точками, ввести координаты в таблицу Excel. Для соседних дуг конечная и начальная точки совпадают.
- 4) Ввести вдоль кривой параметр t , значения которого различны для разных точек кривой, например изменяются от 0 с шагом 0,1 для следующей точки.

x	y
8	0
2	8
0	18
1	23
10	23
16	16
18	6
17	0

	A	B	C	D
1	t	x	y	
2		0	8	0
3		0,1	2	8
4		0,2	0	18
5		0,3	1	23
6				
7		0,3	1	23
8		0,4	10	23
9				
10		0,4	10	23
11		0,5	16	16
12		0,6	18	6
13		0,7	17	0
14				
15		0,7	17	0
16		0,8	8	0
17				

5) Вставить в проект VBA стандартный модуль и ввести код функции, вычисляющей координаты точек сплайна, площади, длины и моменты.

*'функция spline имеет 3 аргумента:
'диапазон x0 - значения параметра t в узлах дуги
'диапазон y0 - значения координаты x или y в узлах
'значение параметра t в нужной точке*

```
Function spline(ByVal x0 As Range,  
                ByVal y0 As Range, t)
```

```
n = x0.Rows.Count - 1      'число точек дуги
```

*'объявление массивов:
'a, b, c, d - коэффициенты сплайна отдельной дуги
'x, y - параметр и координата в узле
'h - значение разности параметра в узлах дуги
'M1 - обратная матрица к матрице системы m
'coeff - все коэффициенты сплайнов как решение
'v - вектор свободных членов системы*

```
ReDim a(n), b(n), c(n), d(n), x(n), y(n), h(n)  
Dim M1() As Variant, coeff() As Variant  
ReDim m(1 To 4 * n, 1 To 4 * n), v(1 To 4 * n)
```

'считывание значений элементов массивов

```
For i = 0 To n  
x(i) = x0.Cells(i + 1, 1).Value  
y(i) = y0.Cells(i + 1, 1).Value  
Next i
```

'вычисление h в узлах

```
For i = 1 To n  
h(i) = x(i) - x(i - 1)  
Next i
```

'первоначальное задание матриц системы

```
For i = 1 To 4 * n  
    For j = 1 To 4 * n  
        m(i, j) = 0      'матрица из нулей  
    Next j  
    v(i) = 0              'вектор из нулей  
Next i
```


'коэффициенты матриц для первых 2n уравнений

'k - счетчик сплайнов

For k = 0 To n - 1

i = 2 * k

j = 4 * k

m(i + 1, j + 1) = 1

m(i + 2, j + 1) = 1

m(i + 2, j + 2) = h(k + 1)

m(i + 2, j + 3) = h(k + 1) ^ 2

m(i + 2, j + 4) = h(k + 1) ^ 3

v(i + 1) = y(k)

v(i + 2) = y(k + 1)

Next k

'коэффициенты матриц для следующих 2n-2 уравнений

For k = 0 To n - 2

i = 2 * k + 2 * n

j = 4 * k

m(i + 1, j + 2) = 1

m(i + 1, j + 3) = 2 * h(k + 1)

m(i + 1, j + 4) = 3 * h(k + 1) ^ 2

m(i + 1, j + 6) = -1

m(i + 2, j + 3) = 1

m(i + 2, j + 4) = 3 * h(k + 1)

m(i + 2, j + 7) = -1

Next k

'коэффициенты матриц для последних 2 уравнений

m(4 * n - 1, 3) = 1

m(4 * n, 4 * n - 1) = 1

m(4 * n, 4 * n) = 3 * h(n)

'Решение СЛАУ вызовом функций листа MInverse и MMult

'вектор v должен быть транспонирован

'функцией Transpose:

M1 = WorksheetFunction.MInverse(m())

coeff = WorksheetFunction.MMult(M1,
WorksheetFunction.Transpose(v))

'Определение коэффициентов отдельных сплайнов

For i = 1 To n

k = 4 * (i - 1)

```

a(i) = coeff(k + 1, 1)
b(i) = coeff(k + 2, 1)
c(i) = coeff(k + 3, 1)
d(i) = coeff(k + 4, 1)
Next i

```

```

'Вычисление значения сплайн-интерполяции
'в нужной точке с параметром t
For k = 1 To n
    If x(k - 1) <= t And t <= x(k) Then
        spline = a(k) + b(k) * (t - x(k - 1)) +
                  c(k) * (t - x(k - 1)) ^ 2 +
                  d(k) * (t - x(k - 1)) ^ 3
    End If
Next k

End Function

```

6) Вставить в тот же модуль код функций, вычисляющих площади, длины и моменты.

Для вычисления используются формулы, определяющие вклад в вычисляемые величины двух соседних точек сплайна $x(t_i)$, $y(t_i)$ и $x(t_{i+1})$, $y(t_{i+1})$, $i = 0, 1, \dots, i_{\max}$, где i_{\max} – достаточно большое число.

$$\begin{aligned}
 S_{\Delta} &= \frac{1}{2} |OM_1 \times OM_2| = \frac{1}{2} |x_1 y_2 - x_2 y_1| & S &= \frac{1}{2} \left| \sum_{i=0}^{i_{\max}-1} (x_i y_{i+1} - x_{i+1} y_i) \right| \\
 L &= \sum_{i=0}^{i_{\max}-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} & x_c &= -\frac{M_y}{S} & y_c &= \frac{M_x}{S} \\
 M_x &= \frac{1}{6} \sum_{i=0}^{i_{\max}-1} (x_i - x_{i+1})(y_i^2 + y_i y_{i+1} + y_{i+1}^2) & M_y &= \frac{1}{6} \sum_{i=0}^{i_{\max}-1} (y_i - y_{i+1})(x_i^2 + x_i x_{i+1} + x_{i+1}^2)
 \end{aligned}$$

'Функция Area имеет 3 аргумента:

'диапазон t0 – значения параметра t в узлах дуги

'диапазон x0 – значения координаты x в узлах

'диапазон y0 – значения координаты y в узлах

```

Function Area(ByVal t0 As Range, _
              ByVal x0 As Range, ByVal y0 As Range)

```

'определение минимального и максимального

'значений параметра в пределах дуги

'вызовом функций листа Min и Max

```

tmin = Application.WorksheetFunction.Min(t0)
tmax = Application.WorksheetFunction.Max(t0)
imax = 100      'максимальное число точек
dt = (tmax - tmin) / imax  'шаг параметра
'цикл вычисления площади
Area = 0
For i = 0 To imax - 1
    t1 = tmin + i * dt      'параметр текущей точки
    t2 = t1 + dt            'параметр следующей точки
    x1 = spline(t0, x0, t1)  'координаты
    y1 = spline(t0, y0, t1)  'текущей
    x2 = spline(t0, x0, t2)  'и следующей
    y2 = spline(t0, y0, t2)  'точки
    Area = Area + 0.5 * (x1 * y2 - x2 * y1)
Next i
End Function

```

Код функций для вычисления длины и моментов отличается лишь именем функции и расчетной формулой, т.е. заменой подчеркнутых фрагментов на следующие:

```

Length = Length + Sqr((x2 - x1) ^ 2 + (y2 - y1) ^ 2)
Mx = Mx + ((x2 - x1)*(y1^2 + y1 * y2 + y2^2))/6
My = My + ((y2 - y1)*(x1^2 + x1*x2 + x2^2))/6

```

7) В модуль листа ввести код основной программы, выполняющей вызов функций вычисления площадей, длин, моментов и центра масс.

```

Sub prog()
'Объявление объектов-диапазонов
'для параметра, x-координаты и y-координаты
'их число равно числу дуг контура
Dim t1 As Range, x1 As Range, y1 As Range
Dim t2 As Range, x2 As Range, y2 As Range
Dim t3 As Range, x3 As Range, y3 As Range
Dim t4 As Range, x4 As Range, y4 As Range

'Задание объектов-диапазонов
'для параметра, x-координаты и y-координаты
Set t1 = Range("A2:A5")
Set t2 = Range("A7:A8")
Set t3 = Range("A10:A13")
Set t4 = Range("A15:A16")

```

```

Set x1 = Range("B2:B5")
Set x2 = Range("B7:B8")
Set x3 = Range("B10:B13")
Set x4 = Range("B15:B16")

Set y1 = Range("C2:C5")
Set y2 = Range("C7:C8")
Set y3 = Range("C10:C13")
Set y4 = Range("C15:c16")

'вычисление вклада в площадь для отдельных дуг
s1 = Area(t1, x1, y1)
s2 = Area(t2, x2, y2)
s3 = Area(t3, x3, y3)
s4 = Area(t4, x4, y4)

'вычисление суммарной площади
S = Abs(s1 + s2 + s3 + s4)
Cells(2, 5) = S

'вычисление вклада в длину контура для отдельных дуг
L1 = Length(t1, x1, y1)
L2 = Length(t2, x2, y2)
L3 = Length(t3, x3, y3)
L4 = Length(t4, x4, y4)

'вычисление суммарной длины
L = Abs(L1 + L2 + L3 + L4)
Cells(2, 6) = L

'вычисление координат центра масс через моменты
'отдельных дуг
xc = -(My(t1, x1, y1) + My(t2, x2, y2) +
      My(t3, x3, y3) + My(t4, x4, y4)) / S
yc = (Mx(t1, x1, y1) + Mx(t2, x2, y2) +
      Mx(t3, x3, y3) + Mx(t4, x4, y4)) / S

Cells(2, 7) = xc
Cells(2, 8) = yc

End Sub

```

8) Выполнить программу. Результат в таблице на листе:

	E	F	G	H
	S	L	xc	yc
	328,556	69,1443	9,08935	11,5823

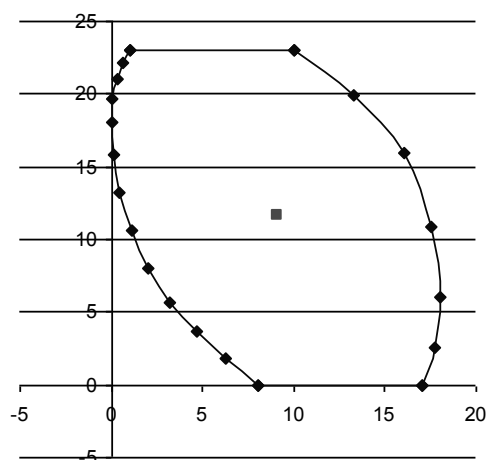
9) Чтобы построить чертеж, вычислив 20-30 точек контура, составить таблицу значений параметра t для каждой дуги контура (на рисунке в ячейках A20:A32, A34:A35 b и т.д.) с меньшим шагом Δt и вычислить соответствующие координаты $x(t)$ и $y(t)$, вставив в ячейки функцию spline. Аргументы функций – диапазоны исходной таблицы (абсолютные ссылки) и значение параметра новой таблицы (относительная ссылка).

	A	B	C	D
20	0	8	0	
21	0,025	6,29688	1,79688	
22	0,05	4,675	3,675	
23	0,075	3,21563	5,71563	
24	0,1	2	8	
25	0,125	1,09063	10,5594	
26	0,15	0,475	13,225	
27	0,175	0,12188	15,7781	
28	0,2	-2,8E-17	18	
29	0,225	0,075	19,7313	
30	0,25	0,3	21,05	
31	0,275	0,625	22,0938	
32	0,3	1	23	
33				
34	0,3	1	23	
35	0,4	10	23	
36				
37	0,4	10	23	
38	0,45	13,325	19,9	
39	0,5	16	16	
40	0,55	17,525	10,925	
41	0,6	18	6	
42	0,65	17,7	2,525	
43	0,7	17	-1,3E-15	
44				
45	0,7	17	0	
46	0,8	8	0	
47				
48		9,08935		11,5823

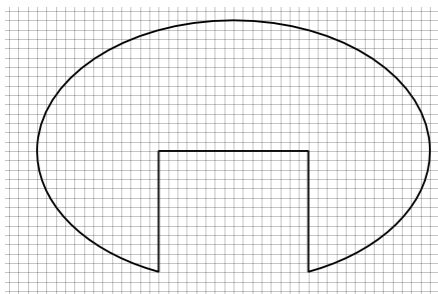
20	0	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A20)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A20)
21	0,025	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A21)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A21)
22	0,05	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A22)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A22)
23	0,075	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A23)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A23)
24	0,1	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A24)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A24)
25	0,125	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A25)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A25)
26	0,15	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A26)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A26)
27	0,175	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A27)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A27)
28	0,2	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A28)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A28)
29	0,225	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A29)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A29)
30	0,25	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A30)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A30)
31	0,275	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A31)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A31)
32	0,3	=spline(\$A\$2:\$A\$5;B\$2:B\$5;\$A32)	=spline(\$A\$2:\$A\$5;C\$2:C\$5;\$A32)
33			
34	0,3	1	23
35	0,4	10	23
36			

10) По вычисленным значениям координат x , y построить точечную диаграмму.

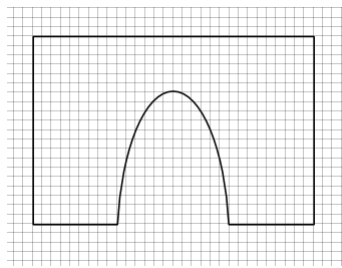
11) Центр масс можно отобразить на той же диаграмме как отдельную точку, дополнив таблицу найденными значениями (ячейки B48 и D48) и расширив область построения диаграммы.



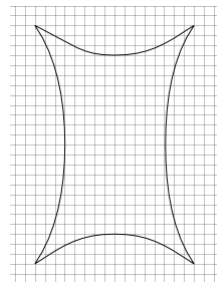
Варианты задания 3.4



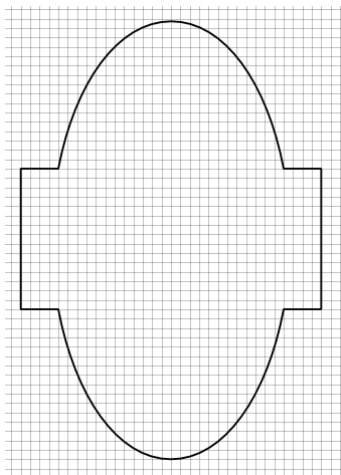
1



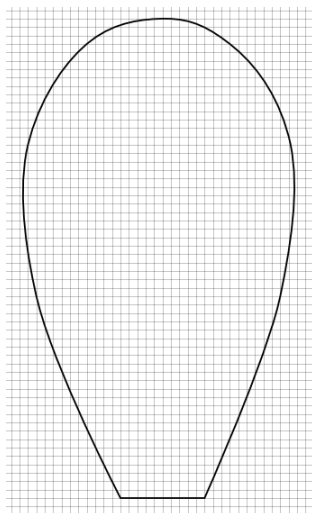
2



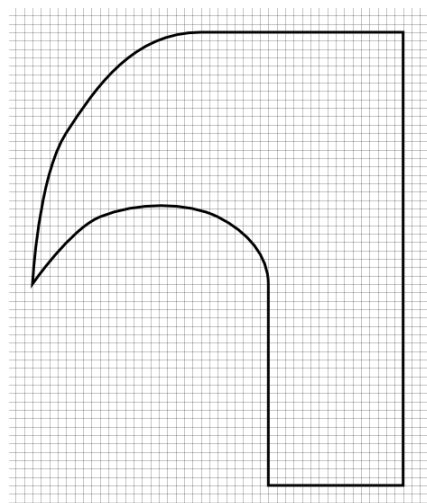
3



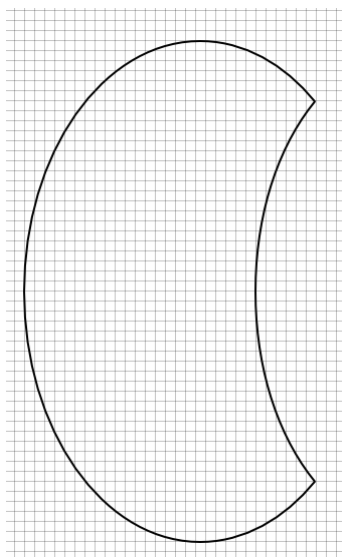
4



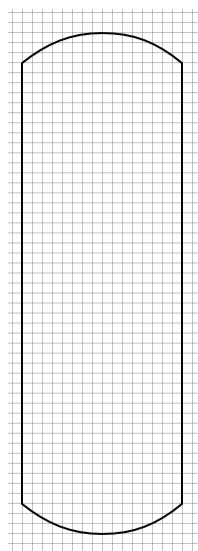
5



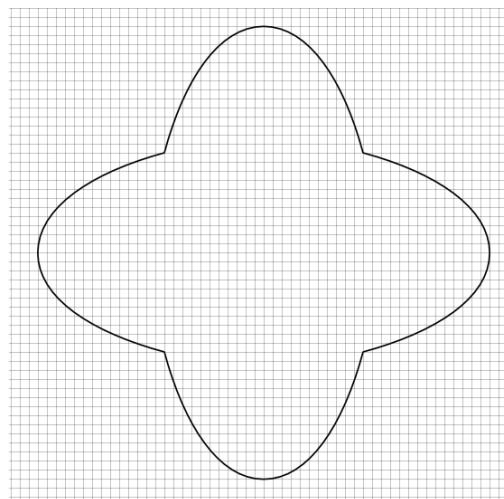
6



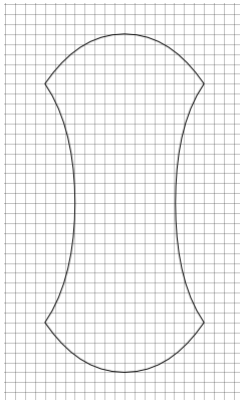
7



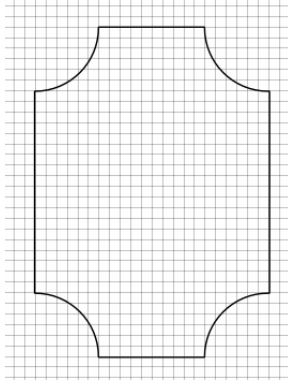
8



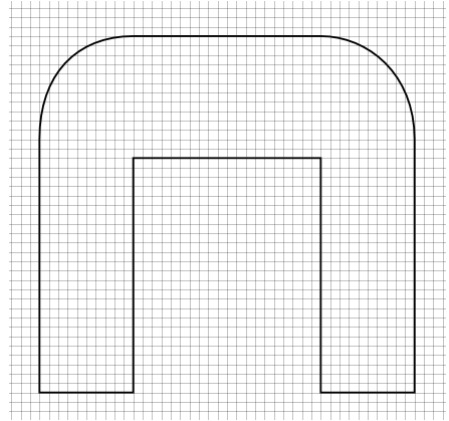
9



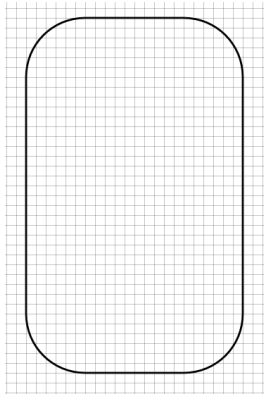
10



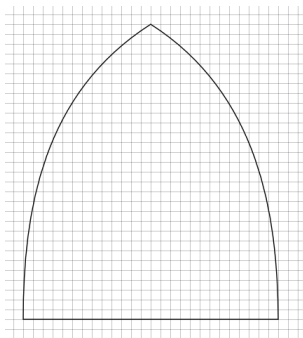
11



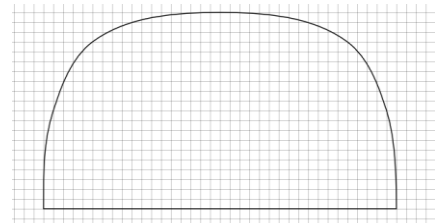
12



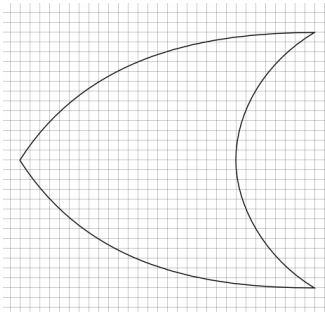
13



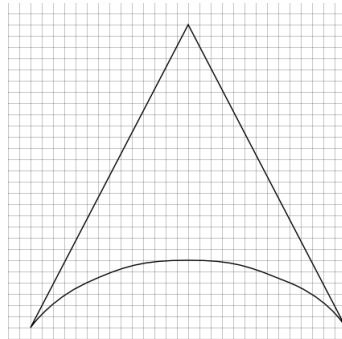
14



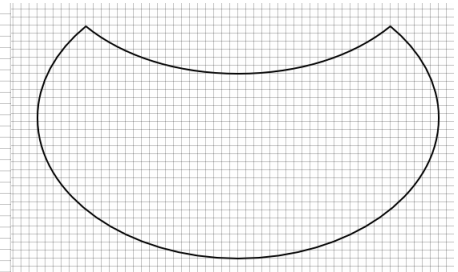
15



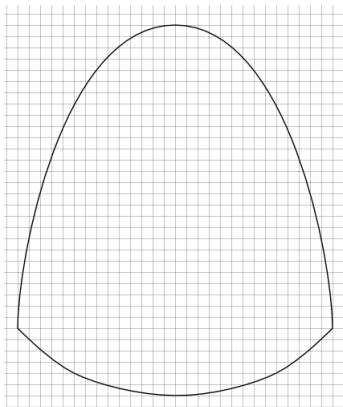
16



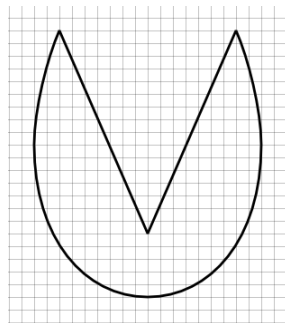
17



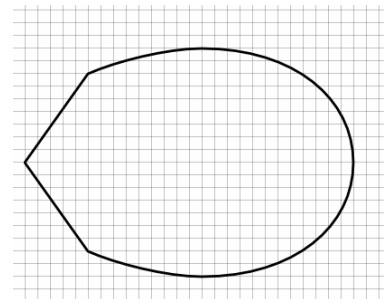
18



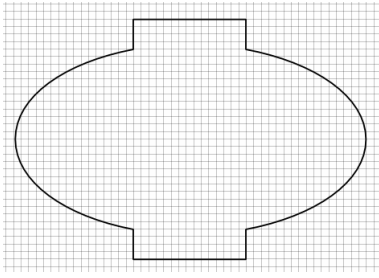
19



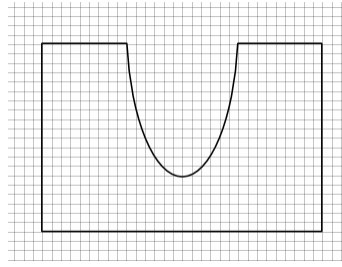
20



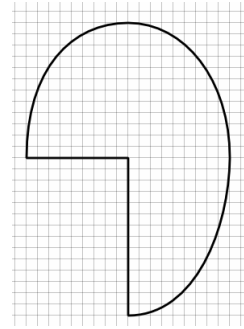
21



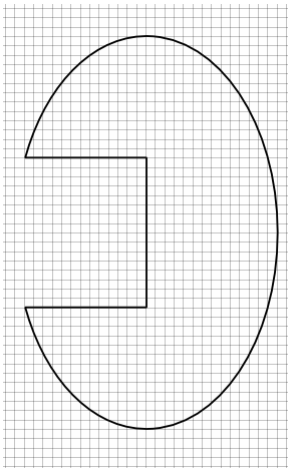
22



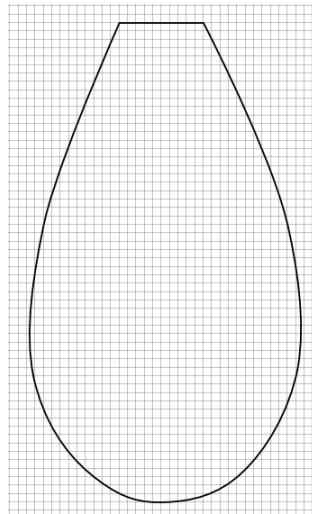
23



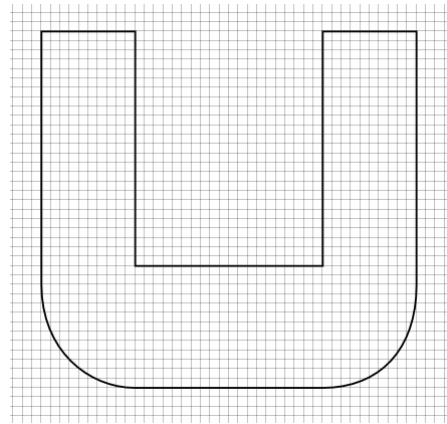
24



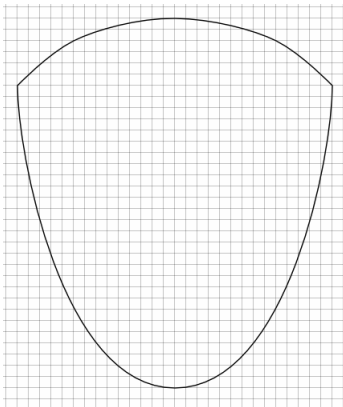
25



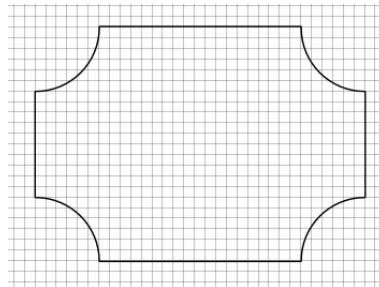
26



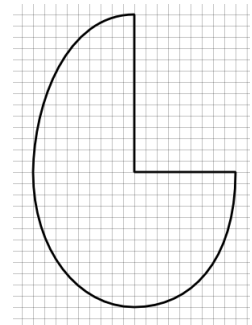
27



28



29



30

4. Объектно-ориентированное программирование в VBA

В VBA наряду с огромным числом встроенных объектов предусмотрена возможность создания пользовательских объектов. Пользовательские объекты являются элементами пользовательских классов. Для создания объекта описывается класс объекта, содержащий всю информацию об объекте (его свойства и методы) как новом типе. Пользовательские классы конструируются в модулях классов. Процесс создания класса состоит из следующей последовательности действий:

1. Вставить в проект VBA через меню редактора модуль класса: Insert → Class Module. Откроется окно нового модуля класса, которое по умолчанию имеет имя Class1;

2. В окне Properties (Свойства) присвоить свойству Name осмысленное имя класса;

3. Объявить переменные класса, которые используются как «значения свойств»;

4. Создать процедуру инициализации класса (конструктор)

```
Private Sub Class_Initialize
```

В этой процедуре указать значения свойств, принимаемые по умолчанию переменными класса при создании объекта класса;

5. При помощи процедур-свойств Property объявить имена свойств. Property Let объявляет имена свойств, значениями которых являются переменные, Property Set объявляет имена свойств, значениями которых являются объекты, Property Get устанавливает возможность считывания значений свойства:

```
Public Property Let свойство (Аргументы)
```

```
    инструкции
```

```
End property
```

```
Private Property Get свойство (Аргументы) As тип
```

```
    инструкции
```

```
End property
```

6. Создать методы класса. Методы создаются при помощи обычных процедур и функций.

7. Для использования объекта в модуле программы он должен быть объявлен с указанием его класса:

```
Dim ИмяОбъекта As Класс
```

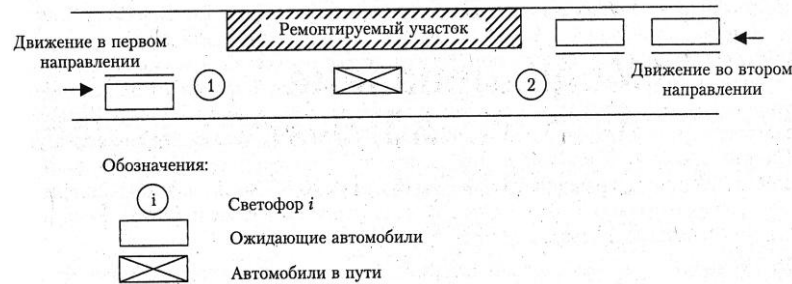
Экземпляр объекта создается и удаляется оператором Set :

```
Set ИмяОбъекта = New Класс
```

```
Set ИмяОбъекта = Nothing
```

Задание 4. Моделирование дорожного движения на ремонтируемом участке дороги. Моделируемая система представляет собой поток транспорта, движущегося по дороге с двусторонним движением, одна сторона которой закрыта в связи с ремонтом.

Светофоры, размещенные на обоих концах одностороннего участка, управляют движением на нем. Светофоры открывают движение на участке в одном из направлений в течение заданного промежутка времени.



Когда загорается зеленый свет, машины следуют по участку с интервалом $t_{\text{въезда}}$. Подъезжающий к участку автомобиль едет по нему без задержки, если горит зеленый свет и перед светофором нет машин, иначе становится в очередь.

Автомобили подъезжают к светофорам через интервалы со средними значениями $t_{\text{прибытия } 1}$, $t_{\text{прибытия } 2}$. Светофор имеет следующий цикл: зеленый в первом направлении в течение $t_{\text{зеленый } 1}$, красный в обоих направлениях, зеленый во втором направлении в течение $t_{\text{зеленый } 2}$, красный в обоих направлениях. Красный свет в обоих направлениях в течение $t_{\text{красный}}$ горит для того, чтобы автомобили, следующие через ремонтируемый участок, могли покинуть его до переключения зеленого света на другое направление.

Определить такие значения зеленых интервалов светофоров, при которых среднее время ожидания автомобилей будет минимальным

Исходные данные

параметры	Светофор 1	Светофор 2	общие
средний интервал прибытия автомобилей	$10 + m_1$	$5 + m_2$	
зеленый интервал	$40 + m$	$30 \div 120$	
красный интервал			$60 - 3n$
время въезда			2
максимальная длина очереди			150
время моделирования (часов)			12

где n - номер группы (последняя цифра); m - номер варианта (по списку группы)

m_1 - число десятков номера варианта; m_2 - число единиц номера варианта

Описание модели.

Объектно-ориентированное программирование предполагает организацию программы не как структуру последовательно выполняющихся процедур и функций, а как систему взаимодействующих объектов.

Рассматриваемая модель требует введения двух классов объектов – это Автомобиль и Светофор. Какие свойства и методы необходимы для описания класса Автомобиль? Очевидно, что не имеют значения тип (автобус, грузовик, легковая машина), число пассажиров и даже скорость движения. Единственная роль объекта Автомобиль в модели – определить время, проведенное в состоянии ожидания, т.е. в очереди. Таким образом, класс автомобиль имеет только два свойства – время ожидания (время_ожидания) и номер в очереди (номер). Единственный метод класса – процедура инициализации объекта `class_initialize()`.

Класс светофор значительно сложнее. Здесь это не просто светофор, а состояние дороги в месте расположения светофора. Его свойствами являются: номер светофора (номер), средний интервал прибытия машин (интервал_прибытия), продолжительность зеленого интервала (зеленый_этого), продолжительность зеленого интервала другого светофора (зеленый_другого), промежутки времени до наступления событий, таких как загорание красного (до_красного) или зеленого света (до_зеленого), прибытие машины (до_прибытия), проезд машины на ремонтируемый участок и освобождение места в очереди (до_проезда), а также сама очередь как массив объектов класса Автомобиль (Очередь), где число занятых мест определяется длиной очереди (длина_очереди), которая не превышает некоторое максимальное значение (длина_очереди_макс). Методы класса:

Создать (интервал_прибытия, зеленый_этого, зеленый_другого, номер) – создание объекта и задание его свойств, включая пустую очередь нулевой длины;
Прибытие() – процедура, обрабатывающая событие, связанное с прибытием машины, которая проезжает или становится в очередь;
Проезд() – процедура, обрабатывающая событие, связанное с проездом машины и освобождением места в очереди;
Красный() – процедура, обрабатывающая событие, связанное с включением красного света;
Зеленый() – процедура, обрабатывающая событие, связанное с включением зеленого света;
Пуск() – процедура, обрабатывающая изменение параметров светофора с течением времени.

Основная программа как процедура Моделирование () организует процесс моделирования системы как цикл по параметру t , представляющему собой модельное время, изменяющееся с шагом 1 в заданных пределах $0 < t < t_max$ после ввода исходных данных (красный_интервал, время_въезда, масштаб_времени, t_max) и создания объектов. На каждом шаге цикла уменьшаются на 1 интервалы времени до наступления событий. Неактивные счетчики времени устанавливаются равными -1. В случае наступления события вызывается соответствующий метод объекта. Процедура Моделирование () ведет учет и вывод результатов в виде оценки значений глобальных переменных прибыло (число прибывших машин), проехало (число проехавших машин), очередь1 (средняя длина очереди левого светофора), очередь2 (средняя длина очереди правого светофора), ожидание (среднее время ожидания).

Из теории систем массового обслуживания известно, что промежутки времени между наступлением события (приездом машины) – это случайные величины, распределенные по экспоненциальному закону. Для вычисления соответствующих значений (до_прибытия) используется генератор экспоненциально распределенных случайных чисел rnd_exp, аргументом которого выступает средний интервал прибытия автомобилей. Если $0 \leq u \leq 1$ – равномерно распределенная случайная величина. Пусть случайная величина t имеет экспоненциальное распределение со средним значением τ . Тогда функция распределения случайной величины t $F(t) = 1 - e^{-t/\tau}$, ($0 \leq t \leq \infty$). Для реализации случайной величины t получаем реализацию случайной величины u и вычисляем t методом обратной функции $t = F^{-1}(u)$, т.е.

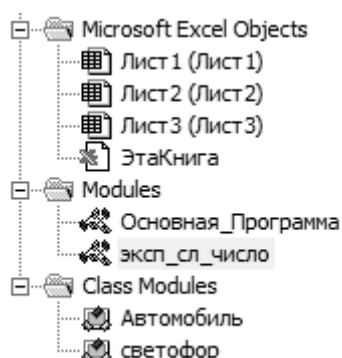
$$t = -\tau \ln(1 - u)$$

Наличие экспоненциальных случайных величин со средними значениями и среднеквадратичными отклонениями порядка нескольких секунд не позволяет использовать в качестве единицы модельного времени секунду, так как погрешность округления случайной величины до целого будет сравнима с самой величиной. Для того, чтобы избежать потери точности при округлении временных интервалов до целых значений, вводится масштабирование по времени с параметром масштаб_времени, равным 10, т.е. шаг по времени t , равный 1, соответствует 0,1 секунды. Масштабируются все временные промежутки при вводе и выводе.

Представленная модель является примером имитационной модели системы массового обслуживания. Так как модель стохастическая, требуется многократные расчеты и последующая статистическая обработка результатов.

Порядок работы:

1. Создать проект следующей структуры и ввести код программы моделирования в соответствующие модули.

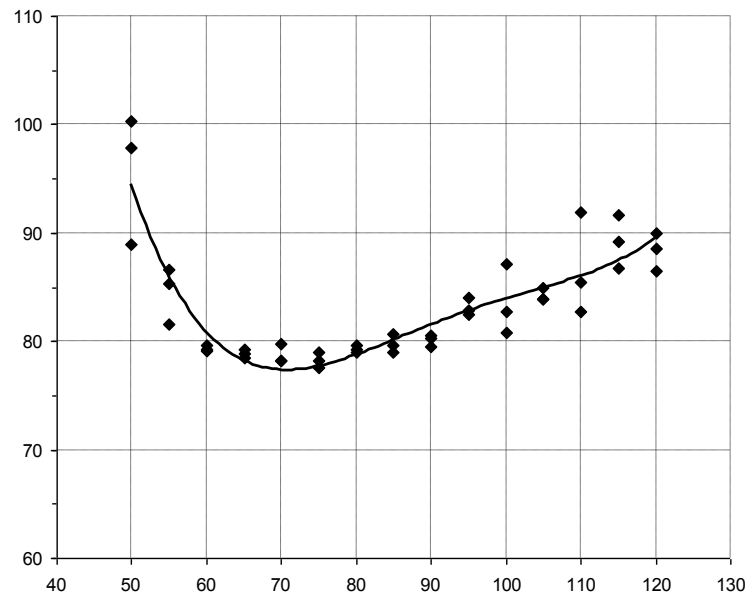


2. Ввести таблицу данных-результатов в Лист 1.

параметры	Светофор 1	Светофор 2	общие
средний интервал прибытия автомобилей	12	9	
зеленый интервал	50	70	
красный интервал			55
время въезда			2
максимальная длина очереди			150
время моделирования (часов)			12
результаты			
прибыло			8448
проехало			8428
средняя длина очереди	7,2	8,1	
среднее время ожидания			78,4

3. Выполнить моделирование, изменяя интервал $t_{\text{зеленый } 2}$ в пределах 40 - 140 сек с шагом 5 сек. Каждый расчет повторить 3-5 раз. Результаты зависимости $t_{\text{ожидания}}$ от $t_{\text{зеленый } 2}$ занести в таблицу.

зеленый интервал	среднее время ожидания
40	549,2
40	606,2
40	581,5
45	207,2
45	134,7
45	345,9
50	88,9
50	97,9
...	...
95	82,9
95	84
100	82,7
100	80,8
...	...



4. Построить точечную диаграмму, добавить полиномиальную линию тренда, и определить графически точку минимума $t_{\text{зеленый } 2}$ с точностью 1 сек.

5. Сформулировать вывод: наименьшее среднее время ожидания равное 77 с достигается при продолжительности зеленого интервала 2-го светофора 70 с.

Код программы.

Класс Автомобиль

```
Public номер, время_ожидания
```

```
Private Sub class_Initialize()
```

```
    номер = 0
```

```
    время_ожидания = 0
```

```
End Sub
```

Класс Светофор

```
Public номер, интервал_прибытия
```

```
Public зеленый_этого, зеленый_другого
```

```
Public до_красного, до_зеленого
```

```
Public до_проезда, до_прибытия, длина_очереди
```

```
Dim Очередь() As Автомобиль
```

```
Public Sub Создать(a, b, c, d)
    ReDim Очередь(длина_очереди_макс) As Автомобиль
    номер = d
    For i = 0 To длина_очереди_макс
        Set Очередь(i) = New Автомобиль
        Очередь(i).номер = i
    Next i
```

```
интервал_прибытия = a
зеленый_этого = b
зеленый_другого = c
    If номер = 1 Then
        до_красного = зеленый_этого
        до_зеленого = -1
    Else
        до_красного = -1
        до_зеленого = красный_интервал + _
                        зеленый_другого _

    End If
    до_поезда = -1
    длина_очереди = 0
    до_прибытия = Int(rnd_exp(интервал_прибытия))
    If до_прибытия = 0 Then до_прибытия = 1
End Sub
```

```
Public Sub Прибытие()
    Dim машина As Автомобиль
    If длина_очереди < длина_очереди_макс Then
        до_прибытия = Int(rnd_exp(интервал_прибытия))
        If до_прибытия = 0 Then до_прибытия = 1
        прибыло = прибыло + 1
        Set машина = New Автомобиль
        машина.номер = прибыло
        If длина_очереди > 0 Then
            длина_очереди = длина_очереди + 1
            Очередь(длина_очереди).номер = машина.номер
            Очередь(длина_очереди).время_ожидания = _
                машина.время_ожидания
        ElseIf до_зеленого > 0 Then
            длина_очереди = 1
            Очередь(длина_очереди).номер = машина.номер
            Очередь(длина_очереди).время_ожидания = _
                машина.время_ожидания
```

```

Else
    проехало = проехало + 1
    ожидание = ожидание * (1# - 1# / проехало)
End If
Set машина = Nothing
End If
End Sub



---


Public Sub Проезд()
    If до_красного = -1 Then
        до_проезда = -1
        Exit Sub
    End If
    проехало = проехало + 1

    tmp = Очередь(1).время_ожидания
    ожидание = ожидание*(1 - 1/проехало) + tmp/проехало

    Очередь(1).номер = 0
    Очередь(1).время_ожидания = 0

    If длина_очереди > 1 Then
        For i = 2 To длина_очереди
            Очередь(i - 1).номер = Очередь(i).номер
            Очередь(i - 1).время_ожидания =
                Очередь(i).время_ожидания
        Next i
        до_проезда = время_въезда
    Else
        до_проезда = -1
    End If
    Очередь(длина_очереди).номер = 0
    Очередь(длина_очереди).время_ожидания = 0
    длина_очереди = длина_очереди - 1
End Sub



---


Public Sub Красный()
    до_красного = -1
    до_зеленого = 2*красный_интервал + зеленый_другого
    до_проезда = -1
End Sub

```

```
Public Sub Зеленый()
    до_зеленого = -1
    до_красного = зеленый_этого
    If длина_очереди > 0 Then Проезд
End Sub
```

```
Public Sub Пуск()
    If до_проезда > 0 Then до_проезда = до_проезда-1
    If до_проезда = 0 Then Проезд
    If до_красного > 0 Then до_красного = до_красного-1
    If до_красного = 0 Then Красный
    If до_зеленого > 0 Then до_зеленого = до_зеленого-1
    If до_зеленого = 0 Then Зеленый
    If до_прибытия > 0 Then до_прибытия = до_прибытия-1
    If до_прибытия = 0 Then Прибытие

    If длина_очереди > 0 Then
        For i = 1 To длина_очереди
            Очередь(i).время_ожидания = _
                Очередь(i).время_ожидания + 1
        Next i
    End If

    If (номер = 1) And (t > 0) Then
        очередь1 = очередь1 + длина_очереди
    ElseIf (номер = 2) And (t > 0) Then
        очередь2 = очередь2 + длина_очереди
    End If
End Sub
```

Модуль эксп_сл_число

```
'генератор экспоненциально распределенных случайных
'чисел
Public Function rnd_exp(tau)
    Randomize
    u = Rnd()
    rnd_exp = - tau * Log(1 - u)
End Function
```

'Основная программа

```
Public масштаб_времени, прибыло, проехало, t
Public длина_очереди_макс
Public очереди1, очередь2, ожидание
Public красный_интервал, время_въезда
Dim t_max As Double

Public Sub Моделирование()
    Dim s1 As светофор, s2 As светофор

    прибыло = 0
    проехало = 0
    очереди1 = 0
    очередь2 = 0
    длина_очереди_макс = Cells(6, "D")
    ожидание = 0
    масштаб_времени = 10 'шаг по времени 0,1 секунды
    t_max = 3600# * масштаб_времени * Cells(7, "D")
    красный_интервал = Cells(4, "D") * масштаб_времени
    время_въезда = Cells(5, "D") * масштаб_времени

    интервал1 = Cells(2, "B") * масштаб_времени
    интервал2 = Cells(2, "C") * масштаб_времени
    зеленый1 = Cells(3, "B") * масштаб_времени
    зеленый2 = Cells(3, "C") * масштаб_времени

    Set s1 = New светофор
    Set s2 = New светофор

    Call s1.Создать(интервал1, зеленый1, зеленый2, 1)
    Call s2.Создать(интервал2, зеленый2, зеленый1, 2)

    For t = 0 To t_max
        s1.Пуск
        s2.Пуск
    Next

    Cells(10, "D") = прибыло
    Cells(11, "D") = проехало
    Cells(12, "B") = очереди1 / t_max
    Cells(12, "C") = очередь2 / t_max
    Cells(13, "D") = ожидание / масштаб_времени
End Sub
```

5. Визуальное программирование в VBA

Визуальное программирование — способ создания программы путём манипулирования графическими объектами вместо написания текста кода. В VBA визуальное программирование используется для расширения интерфейса приложения или создания собственных оконных интерфейсов. Выполняется с помощью создания элементов управления, которые могут размещаться непосредственно на рабочих листах или в пользовательских диалоговых окнах. Элементы управления являются объектами. Поэтому, как любые объекты, они обладают свойствами, методами и событиями — специальными методами (процедурами), связанными с типичными изменениями состояния элемента управления..

Пользовательские диалоговые окна создаются на основе пользовательских форм UserForm следующей последовательностью действий:

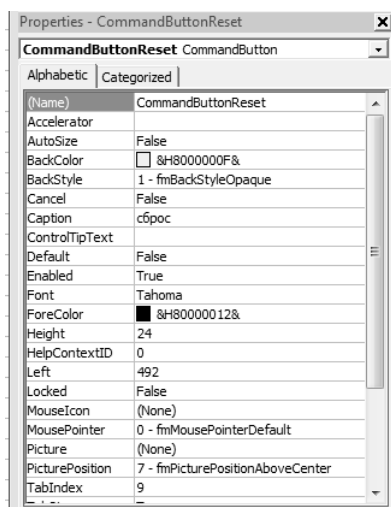
1. Вставить модуль формы UserForm в проект рабочей книги командой меню редактора Insert → UserForm. Формы получают такие имена, как UserForm1, UserForm2 и т.д.

2. Создать процедуру, которая будет отображать форму UserForm. Для этого необходимо вызывать метод Show объекта UserForm. Если объект называется UserForm1, то следующая процедура отобразит это пользовательское диалоговое окно.

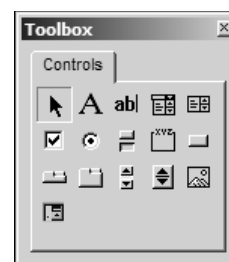
```
Sub ShowDialog()  
    UserForm1.Show  
End Sub
```

Процедура должна располагаться в стандартном модуле VBA, а не в модуле формы.

3. Добавить элементы управления в форму UserForm, используя окно Toolbox. Выбрать элемент, щелкнув на той кнопке в окне Toolbox, которая соответствует



добавляемому элементу управления. После этого щелкнуть внутри диалогового окна для создания элемента управления и, перетаскив его границы в диалоговом окне, задать необходимый размер.



4. Настроить свойства добавленных элементов управления. После добавления нового элемента управления ему назначается имя, которое состоит из названия типа элемента управления и числового кода CommandButton1, Label1, Label2. Рекомендуется

переименовывать элементы управления используя описательные имена, например `cmdStart`, `txtResult`. Чтобы изменить имя элемента необходимо в окне **Properties** редактора выделить необходимый элемент и изменить свойство `Name`. С помощью панели **Properties** обычно настраивают исходные значения свойств элементов. Свойства могут быть установлены и изменены программно.

5. Создать процедуры «обработчики событий» для элементов управления. Эти процедуры добавляются в модуль кода `UserForm` и выполняются при возникновении различных событий (например, при щелчке на кнопке).

Основные элементы управления:

CommandButton – кнопка.

Label – надпись (нередатируемый текст).

TextBox – текстовое поле ввода и вывода данных.

CheckBox – переключатель типа «флаг». Предоставляет возможность выбрать один из двух вариантов: да или нет, включить или выключить и т.д. Если элемент управления `CheckBox` установлен, то он имеет значение `True`, в противном случае значение равно `False`.

OptionButton – переключатель типа «радиокнопка». Применяется при выборе одного варианта настроек из нескольких. Всегда создаются группой. Когда один элемент группы выбран, все остальные автоматически становятся неактивными.

Основные свойства элементов управления:

Enabled – доступность (значения `True` или `False`). Недоступный элемент неактивен.

Visible – видимость (значения `True` или `False`). Невидимый элемент скрыт.

Caption – подпись (текст, размещенный на элементе)

Text – текст в текстовом поле

Font – шрифт подписи, текста (гарнитура, размер, цвет и т.д.)

Value – значение (`True` или `False`)

Основные события элементов управления:

Click() – щелчок (на кнопке).

DblClick() – двойной щелчок (на кнопке).

Change() – изменение (текста, значения).

Текст, вводимый и выводимый в текстовое поле `TextBox`, представляет собой тип данных `String` (строка). Для контроля вводимых данных и преобразования текста в числовые значения применяются следующие функции:

`CDbl (строка)` – преобразование текста в соответствующее числовое значение с двойной точностью.

`CInt (строка)` – преобразование текста в соответствующее целое число.

CStr(выражение) – преобразование результата выражения в текстовую строку.

IsNumeric(строка) – проверка того, что текст строки содержит запись числа. Результат функции – True или False.

Пример 5.1. Создать диалоговое окно для вычисления определенных интегралов от произвольной функции методом Монте-Карло.

Алгоритм интегрирования Монте-Карло. Предположим, требуется вычислить определённый интеграл $\int_a^b f(x)dx$. Среднее значение функции $f(x)$

на интервале $[a; b]$ $f_{cp} = \frac{1}{b-a} \int_a^b f(x)dx$. Таким образом, интеграл выражается как

$$\int_a^b f(x)dx = (b-a)f_{cp}.$$

Но среднее значение функции $f(x)$ на интервале $[a; b]$ можно вычислить как выборочное среднее. Выбираем случайно n точек x_i , равномерно распределённых на $[a; b]$, для каждой точки вычисляем $f(x_i)$.

Затем вычисляем выборочное среднее: $f_{cp} = \frac{1}{n} \sum_{i=1}^n f(x_i)$

В итоге получаем оценку интеграла:

$$\int_a^b f(x)dx \approx \frac{b-a}{n} \sum_{i=1}^n f(x_i)$$

Таким образом, интеграл, вычисленный методом Монте-Карло представляет собой среднее значение результатов вычисления функции в случайно выбранных точках интервала интегрирования. Точность оценки зависит только от количества точек n . Так как оценка интеграла представляет собой случайную величину, ее погрешность зависит от n как ошибка среднего значения, т.е. обратно пропорционально \sqrt{n} . Таким образом, метод Монте-Карло имеет порядок точности $1/2$ – для уменьшения погрешности в 10 раз необходимо увеличить n в 100 раз.

Порядок работы.

- 1) Вставим в проект модуль формы. Insert → UserForm. Изменим свойства: Name – Integral, Caption – Численное интегрирование.
- 2) Добавим элементы управления TextBox со следующими именами :
 - TextBoxA – поле для ввода нижнего предела
 - TextBoxB – поле для ввода верхнего предела
 - TextBoxF – поле для ввода формулы функции
 - TextBoxN – поле для ввода числа точек
 - TextBoxR – поле для вывода результата
- 3) Изменим значение свойства Font элементов TextBox – установим шрифт

Times New Roman 18 полужирный;

4) Добавим элементы управления Label:

Label1 – надпись для символа интеграла \int

Label2 – надпись для символа дифференциала $dx =$

Label3 – надпись для числа точек $n =$

5) Изменим значение свойства Font элементов Label

Label1 – шрифт Symbol 48

Label2 и Label3 – Times New Roman 18 полужирный

6) Изменим значение свойства Caption элементов Label:

Label1 – вставим символ \int в рамку элемента

Label2 – $dx =$

Label3 – $n =$

7) Добавим элементы управления CommandButton со следующими именами :

CommandButtonCalculate – кнопка включения вычислений

CommandButtonReset – кнопка сброса

8) Изменим значение свойства Caption элементов CommandButton:

CommandButtonCalculate – вычислить

CommandButtonReset – сброс

9) Добавим элементы управления OptionButton со следующими именами:

OptionButtonM

OptionButtonP

10) Изменим значение свойства Caption элементов OptionButton:

OptionButtonM – Монте-Карло

OptionButtonP – парабола

11) Изменим значение свойства Enabled элементов OptionButton:

OptionButtonM – True

OptionButtonP – False

12) Добавим элементы управления CheckButton со следующими именами:

CheckButtonAll

CheckButtonResult

13) Изменим значение свойства Caption элементов CheckButton:

CheckButtonAll – все

CheckButtonResult – результат

14) Изменим значение свойства Enabled элементов CheckButton:

CheckButtonAll – True

CheckButtonResult – False

15) Изменим значение свойства Value элементов CheckButton:

CheckButtonAll – True

CheckButtonResult – False

16) Вставим стандартный модуль Insert → Module и введем код процедуры вызова формы

```

Sub ShowDialog()
    Integral.Show
End Sub
17) В модуле формы введем код программы
Private Sub CommandButtonCalculate_Click()
    Worksheets("Интеграл").Activate
    If IsNumeric(TextBoxA.Text) = True
        Then a = CDb1(TextBoxA.Text)
    End If
    If IsNumeric(TextBoxB.Text) = True Then
        b = CDb1(TextBoxB.Text)
    End If
    If IsNumeric(TextBoxN.Text) = True Then
        n = CInt(TextBoxN.Text)
        TextBoxN.Text = n
    End If
    Formula = "=" & CStr(TextBoxF.Text)
    If n > 0 Then
        Range("A1").Name = "x"
        Range("B1").FormulaLocal = Formula
        If OptionButtonM.Value = True Then
            Randomize
            S = 0
            For i = 1 To n
                x = Rnd() * (b - a) + a
                Range("x") = x
                f = Cells(1, "B")
                If IsNumeric(f) = True Then
                    S = S + f
                Else
                    TextBoxF.Text = ""
                End If
            Next i
        End If
        If TextBoxF.Text <> "" Then
            TextBoxR.Text = CDb1(S) * (b - a) / n
        End If
        Range("A1:B1").Clear
    Else
        TextBoxN.Text = ""
        CommandButtonCalculate.Enabled = False
    End If
End Sub

```

```
Private Sub TextBoxF_Change()  
    TextBoxR.Text = ""  
    If TextBoxF.Text <> "" Then  
        CommandButtonCalculate.Enabled = True  
    Else  
        TextBoxF.Text = ""  
    End If  
End Sub
```

```
Private Sub TextBoxN_Change()  
    TextBoxR.Text = ""  
    If IsNumeric(TextBoxN.Text) Then  
        n = CInt(TextBoxN.Text)  
        CommandButtonCalculate.Enabled = True  
    Else  
        TextBoxN.Text = ""  
    End If  
End Sub
```

```
Private Sub TextBoxA_Change()  
    TextBoxR.Text = ""  
    If IsNumeric(TextBoxA.Text) Then  
        a = CDb1(TextBoxA.Text)  
        CommandButtonCalculate.Enabled = True  
    Else  
        TextBoxA.Text = ""  
    End If  
End Sub
```

```
Private Sub TextBoxB_Change()  
    TextBoxR.Text = ""  
    If IsNumeric(TextBoxB.Text) Then  
        b = CDb1(TextBoxB.Text)  
        CommandButtonCalculate.Enabled = True  
    Else  
        TextBoxB.Text = ""  
    End If  
End Sub
```

```
Private Sub CommandButtonReset_Click()  
    If CheckBoxAll.Value = True Then  
        TextBoxA.Text = ""  
        TextBoxB.Text = ""  
        TextBoxN.Text = ""  
        TextBoxF.Text = ""  
        TextBoxR.Text = ""  
    End If  
End Sub
```



```

End If
CommandButtonCalculate.Enabled = False
End Sub

```

Программа использует алгоритм использования табличного процессора кодом VBA, аналогичный заданию 3.2.

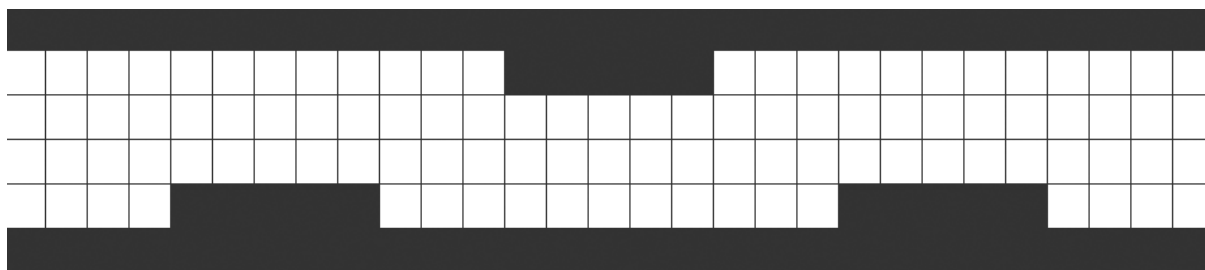
Задание 5.1. Изменить функциональность программы Примера 5.1, добавив возможность вычисления интеграла другими численными методами и модифицировав результаты действия кнопки «сброс»:

Варианты задания 5.1

Варианты			Дополнить программу возможностью вычисления интеграла	Изменить функциональность кнопки «сброс» добавив возможность
1	11	21	Методом левых прямоугольников	удаления результата
2	12	22	Методом левых прямоугольников	удаления функции
3	13	23	Методом правых прямоугольников	удаления n
4	14	24	Методом правых прямоугольников	удаления пределов
5	15	25	Методом центральных прямоугольников	удаления результата
6	16	26	Методом центральных прямоугольников	удаления функции
7	17	27	Методом трапеций	удаления n
8	18	28	Методом трапеций	удаления пределов
9	19	29	Методом парабол	удаление результата
10	20	30	Методом парабол	удаления функции

Большинство элементов управления можно разместить непосредственно на листе Excel. Для этого используется панель инструментов на вкладке Разработчик. При создании и редактировании интерфейса листа Excel специальной кнопкой на этой панели переводится в «режим конструктора». Двойной щелчок на элементе управления в режиме конструктора создает шаблон процедуры события в редакторе VBA. Код элементов управления листа находится в компонентном модуле этого листа.

Задание 5.2. Моделирование дорожного движения с помощью клеточных автоматов. Определить пропускную способность и среднюю скорость движения на участке многополосной магистрали на основе модели клеточных автоматов. Схема участка магистрали:



Описание математической модели

Клеточный автомат – идеализированная физическая система с дискретным пространством и временем, элементы которой могут находиться в конечном числе состояний с заданными правилами переходов между ними.

Математическая модель многополосной автомагистрали на основе клеточного автомата состоит из следующих частей:

- описание модели дорожного полотна;
- описание возможных состояний каждой клетки;
- описание правил перехода;

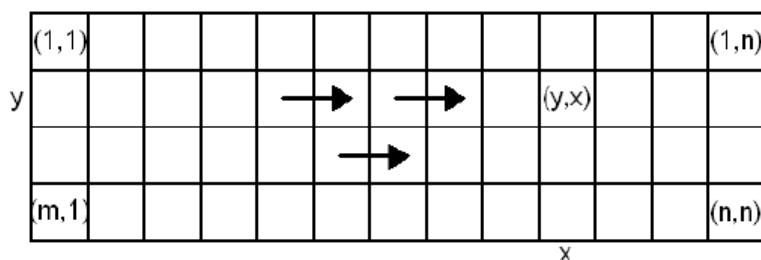
Модель дорожного полотна

При моделировании дорожного движения с помощью клеточных автоматов в качестве модели дорожного полотна используется прямоугольная сетка размера $m \times n$, состоящая из клеток, каждая из которых может находиться в одном из определенных состояний.

Описание каждой клетки представляет собой вектор из четырех переменных: $((y, x), s, t)$, где:

- y – целое число, от 1 до m , описывает номер полосы автомагистрали;
- x – целое число, от 1 до n , описывает текущую координату клетки вдоль направления автомобильного движения;
- s – описывает состояние конкретной клетки в клеточном автомате;
- t – описывает конкретный момент времени с начала моделирования.

Ось x моделирует направление автомагистрали, по которому движутся автомобили, ось y моделирует автомагистраль по ширине, от крайней правой полосы ($y = 1$) – и до крайней левой полосы ($y = m$). Автомобиль может находиться в одной клетке и движется слева направо вдоль оси x . Начало отсчета координат клеточного автомата в левом верхнем углу. Левый верхний угол имеет координаты $(1, 1)$. Левый нижний угол имеет координаты $(m, 1)$. Правый нижний угол и правый верхний угол клеточного автомата имеют координаты (n, m) и $(n, 1)$ соответственно.



Возможные состояния для клетки

Множество допустимых состояний для каждой клетки автомата в конкретный момент времени представлено тремя состояниями:

- 0 – свободна. Клетка в этом состоянии может изменить свое состояние только на состояние 1, в случае если автомобиль переместился в данную клетку.
- 1 – занята автомобилем. Движущийся автомобиль либо освобождает клетку, занимая при этом одну из трех клеток, соседних справа, либо переходит в состояние 2.
- 2 – препятствие. Число клеток в состоянии 2 не может уменьшаться.

Правила перехода

Изменения состояния автомобилей могут быть учтены в математической модели при помощи двух правил.

Правило 1 определяет вероятность p_n , появления на каждом новом шаге нового автомобиля в состоянии 1 для каждой свободной клетки из левого ряда. Для всех клеток с координатами $(y, 1)$ таких, что $s(y, 1) = 0$, происходит генерация случайного числа $r(y, 1)$ в диапазоне от 0 до 1. Если это случайное число оказывается больше чем p_n , то клетка с координатами $(y, 1)$ остается в состоянии 0. Если $r(y, 1) < p_n$, то в клетке с координатами $(y, 1)$ появляется новый автомобиль, который начинает движение: $s(y, 1) = 0$.

Правило 2 определяет механизм, изменения состояния движущегося автомобиля, находящегося в клетке с состоянием 1 в виде функции перехода f .

Учитываются состояния только трех клеток впереди по ходу движения (то есть справа) от данной клетки. Для клетки с координатами (y, x) это будут три соседних клетки с координатами $(y - 1, x + 1)$, $(y, x + 1)$

и $(y + 1, x + 1)$. Каждая из этих клеток может находиться или в одном из трех состояний: 0, 1 или 2.

Множество разрешенных в модели состояний для трех клеток справа по ходу движения 28. Для каждой из возможных конфигураций функция f задается при помощи списка вариантов, каждый из которых состоит из двух элементов

p_i – число от 0 до 1 – определяет вероятность срабатывания именно правила i при вызове функции f ;

$cell_i$ – 0, 1, 2 или 3 – определяет номер клетки, в которой автомобиль оказывается после срабатывания правила i :

0: автомобиль остается в текущей клетке;

1: автомобиль в движении перестраивается в левую полосу;

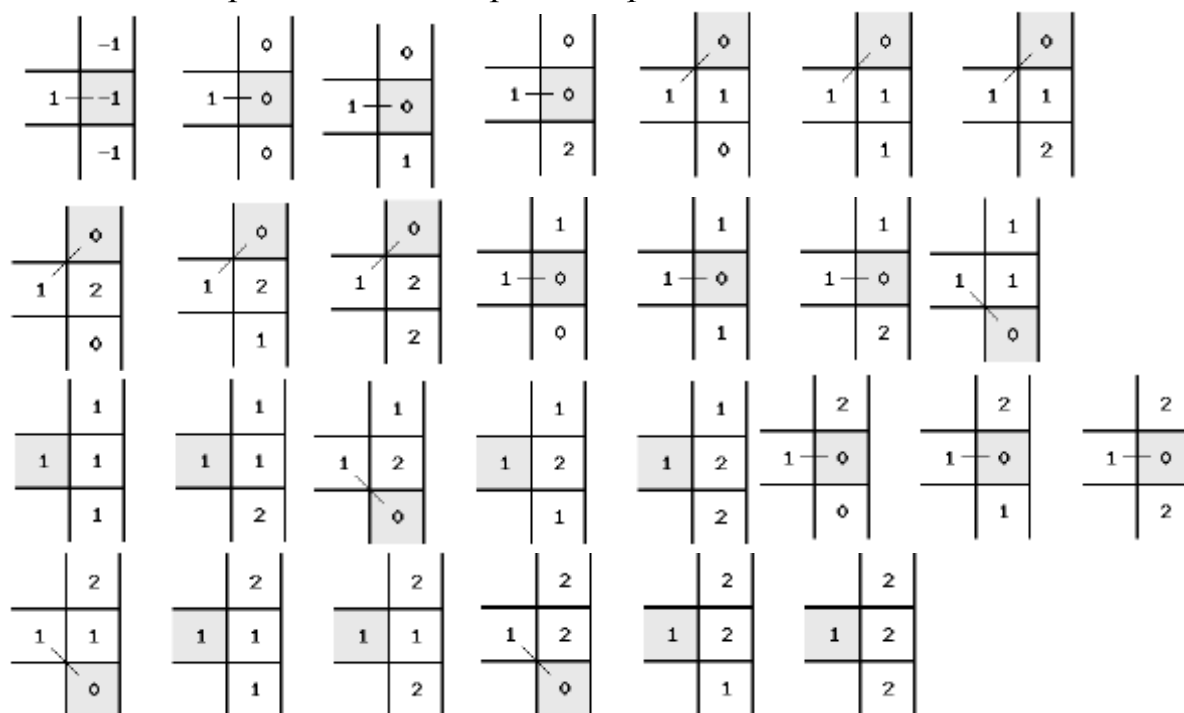
2: автомобиль движется вперед;

3: автомобиль в движении перестраивается в правую полосу;

		1
0		2
		3

Движущиеся автомобили могут исчезать из зоны видимости дорожного полотна, переходя в прямоугольной сетке в самый правый ряд. Это происходит при вызове функции $f(-1, -1, -1)$.

Иллюстрация базовых правил перехода в автомате:



Описание программы

Рассматриваемая модель требует введения двух классов объектов – это `CarPlace` (клетка) и `Model` (модель).

Свойства объекта `CarPlace` – `state` (состояние 0, 1 или 2) и `car` (наличие машины в клетке, 1 или 0). Метод – конструктор `Init()`.

Объект `Model` описывает дорогу в целом. Свойства объекта `Model` – массив объектов `CarPlace` (`car()`) и переменные `pn` (вероятность появления машины), `auto` (количество машин на дороге), `entered` (количество въехавших машин), `moved` (количество движущихся машин), `vt` (скорость). Методы объекта `Model` – создание модели с заданной диапазоном ячеек схемой и заданной вероятностью появления машины `Init`, пошаговое изменение состояния `Step`, вызывающее последовательно методы `Arrival` (прибытие машины), `Departure` (убытие машины), `Rule` (правила перехода).

Элементы управления

Кнопка **Setup** – включает режим настройки. Становятся доступны другие кнопки.

Кнопка **Default** устанавливает параметры модели и схему дороги по умолчанию.

Кнопка **Road** создает схему дороги, где выделенная на листе ячейка задает правый нижний угол схемы.

Кнопка **Block** отмечает выделенную клетку как препятствие (синим цветом).

Кнопка **Car** отмечает выделенную клетку как автомобиль (красным цветом).

Кнопка **Free** стирает выделенную клетку.

Кнопка **Clear** очищает весь лист.

Поле ввода справа предназначено для ввода значения вероятности появления машины. По умолчанию там стоит 0,1.

Кнопка **Ready** завершает режим настройки. Становятся доступны новые кнопки, а кнопки настройки блокируются (кроме **Setup**). Настройку можно повторить.

Кнопка **Start** включает режим моделирования. Все кнопки настройки блокируются.

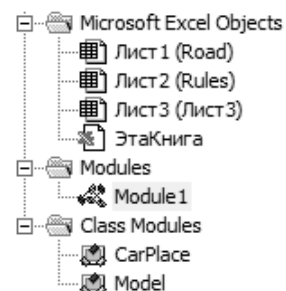
Кнопка **Step** выполняет пошагово моделирование движения. Номер шага отображается в соседнем текстовом поле.

Кнопка **Finish** завершает моделирование и отображает результаты. После нажатия блокируются все кнопки, кроме **Setup**.

Порядок работы:

1. Переименовать Лист 1 в Road, Лист 2 в Rules. Установить стиль ссылок R1C1.

2. Создать проект следующей структуры и ввести код программы моделирования в соответствующие модули.

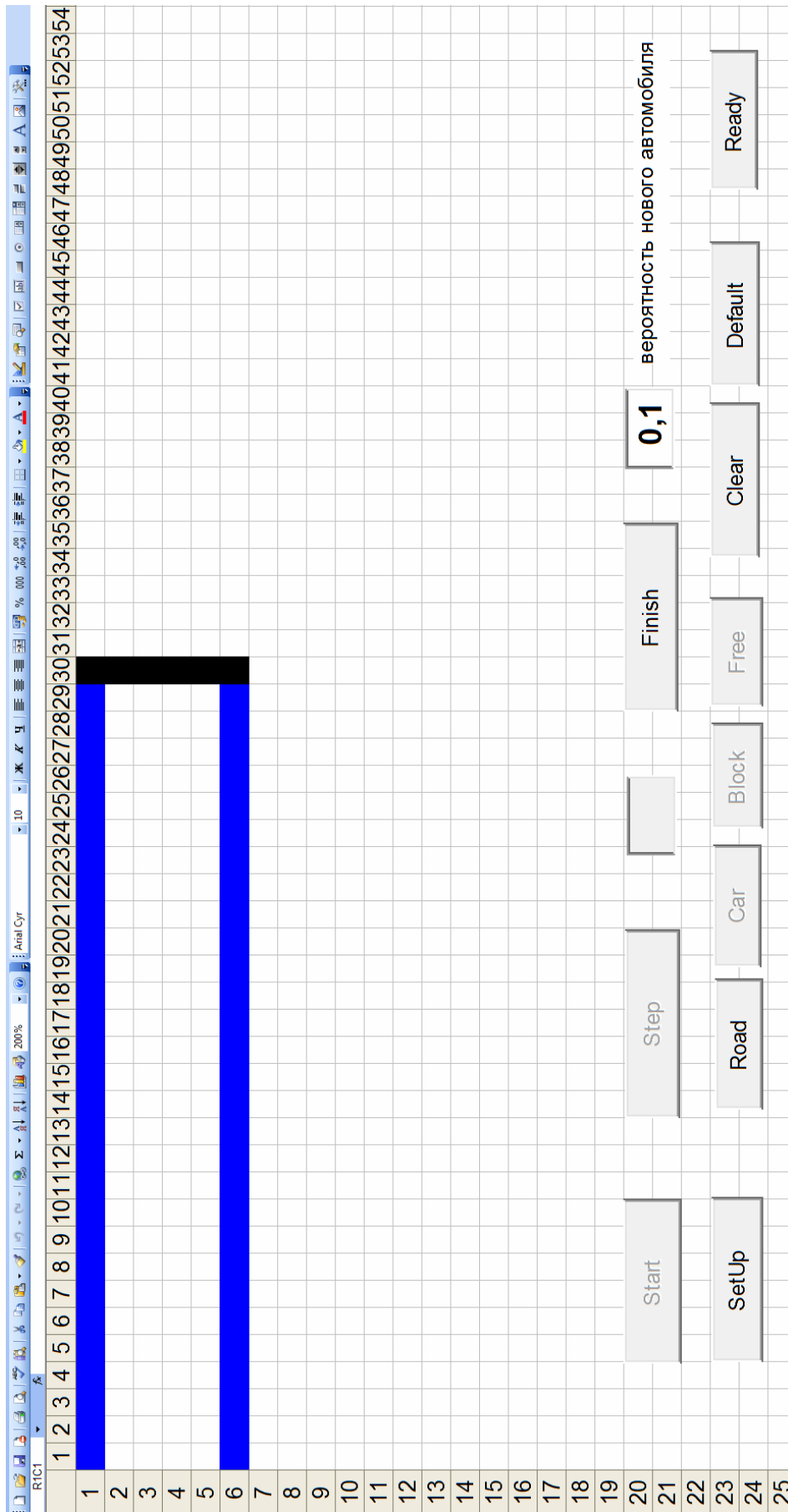


3. Ввести в лист Rules таблицу правил перехода:

	1	2	3	4	5	6	7	8	9	10	11
1	-1	-1	-1	1	2						
2	0	0	0	0,4	2	0,05	1	0,05	3	0,5	0
3	0	0	1	0,4	2	0,1	1	0,5	0		
4	0	0	2	0,4	2	0,1	1	0,5	0		
5	0	1	0	0,3	1	0,2	3	0,5	0		
6	0	1	1	0,5	1	0,5	0				
7	0	1	2	0,5	1	0,5	0				
8	0	2	0	0,3	1	0,2	3	0,5	0		
9	0	2	1	0,5	1	0,5	0				
10	0	2	2	0,5	1	0,5	0				
11	1	0	0	0,4	2	0,1	3	0,5	0		
12	1	0	1	0,5	2	0,5	0				
13	1	0	2	0,5	2	0,5	0				
14	1	1	0	0,5	3	0,5	0				
15	1	1	1	1	0						
16	1	1	2	1	0						
17	1	2	0	0,5	3	0,5	0				
18	1	2	1	1	0						
19	1	2	2	1	0						
20	2	0	0	0,4	2	0,1	3	0,5	0		
21	2	0	1	0,5	2	0,5	0				
22	2	0	2	0,5	2	0,5	0				
23	2	1	0	0,5	3	0,5	0				
24	2	1	1	1	0						
25	2	1	2	1	0						
26	2	2	0	0,5	3	0,5	0				
27	2	2	1	1	0						
28	2	2	2	1	0						

4. Установить на листе Road ширину столбцов равной 1 см (квадратные ячейки) и масштаб отображения 200%:

5. Создать на листе Road элементы управления



Код программы

Module1

```
Public t, m, n  
Public mdl As Model
```

'class CarPlace

```
Public Car As Integer  
Public state As Integer
```

```
Private Sub Init()  
    state = 0  
    Car = 0  
End Sub
```

'class Model

```
Dim c() As CarPlace  
Public pn, auto, entered, moved, vt
```

```
Public Sub Init(ByVal road0 As Range, pn0)  
    pn = pn0  
    m = road0.Rows.Count  
    n = road0.Columns.Count  
    vt = 0  
    entered = 0  
    ReDim c(m, n)  
    For y = 1 To m  
        For x = 1 To n  
            Set c(y, x) = New CarPlace  
            color = road0.Cells(y, x).Interior.ColorIndex  
            If color = 1 Then  
                c(y, x).state = -1  
            ElseIf color = xlNone Then  
                c(y, x).state = 0  
            ElseIf color = 3 Then  
                c(y, x).state = 1  
                c(y, x).Car = 1  
            ElseIf color = 5 Then  
                c(y, x).state = 2  
            End If  
        Next x  
    Next y  
End Sub
```

```

Public Sub Step()
    auto = 0
    moved = 0
    For x = n - 1 To 1 Step -1
        For y = 2 To m - 1
            If c(y, x).state = -1 Then
                c(y, x).state = -1
                c(y, x).Car = 0
            ElseIf c(y, x).state = 2 Then
                c(y, x).state = 2
            ElseIf c(y, x).Car = 1 Then
                left = c(y - 1, x + 1).state
                midle = c(y, x + 1).state
                right = c(y + 1, x + 1).state
                Call Rule(left, midle, right, y, x)
            End If
        Next y
    Next x
    Call Arrival
    Call Departure

    For x = 1 To n
        For y = 2 To m - 1
            If c(y, x).state = -1 Then
                color = 1
            ElseIf c(y, x).state = 0 Then
                color = xlNone
            ElseIf c(y, x).state = 1 Then
                color = 3
                auto = auto + 1
            ElseIf c(y, x).state = 2 Then
                color = 5
            End If
            Workbooks(1).Sheets("Road").Cells(y,
                x).Interior.ColorIndex = color
        Next y
    Next x
    Debug.Print auto
    Workbooks(1).Sheets("Road").Cells(1, 50) = auto
    If auto > 0 Then vt = vt + moved / auto
    Workbooks(1).Sheets("Road").Cells(3, 50) = vt
End Sub

```

```

Public Sub Arrival()
    For y = 2 To m - 1
        If c(y, 1).state = 0 Then
            r = Rnd()
            If r < pn Then
                c(y, 1).state = 1
                c(y, 1).Car = 1
                entered = entered + 1
            End If
        End If
    Next y
    Workbooks(1).Sheets("Road").Cells(2, 50) = entered
End Sub

```

```

Public Sub Departure()
    For y = 2 To m - 1
        c(y, n).state = -1
        c(y, n).Car = 0
    Next y
End Sub

```

```

Public Sub Rule(left, midle, right, y, x)
    r = Rnd()
    For i = 1 To 28
        With Workbooks(1).Sheets("Rules")
            lf = .Cells(i, 1)
            md = .Cells(i, 2)
            rt = .Cells(i, 3)

            If lf = left And md = midle And rt = right Then
                k = 2
                p = 0
                Do
                    k = k + 2
                    p = p + .Cells(i, k)
                Loop Until r < p

                d = .Cells(i, k + 1)
                If d = 0 Then
                    c(y, x).state = 1
                    c(y, x).Car = 1
                ElseIf d = 1 Then
                    c(y, x).state = 0
                End If
            End If
        End With
    Next i
End Sub

```

```

        c(y, x).Car = 0
        c(y - 1, x + 1).state = 1
        c(y - 1, x + 1).Car = 1
        moved = moved + 1
    ElseIf d = 2 Then
        c(y, x).state = 0
        c(y, x).Car = 0
        c(y, x + 1).state = 1
        c(y, x + 1).Car = 1
        moved = moved + 1
    ElseIf d = 3 Then
        c(y, x).state = 0
        c(y, x).Car = 0
        c(y + 1, x + 1).state = 1
        c(y + 1, x + 1).Car = 1
        moved = moved + 1
    End If
Exit For
End If
End With
Next i
Workbooks(1).Sheets("Road").Activate
End Sub

```

Лист1 (Road)

```

Private Sub cmdCar_Click()
    c = ActiveCell.Column
    r = ActiveCell.Row
    If c < n And r < m And c > 1 And r > 1 Then
        ActiveCell.Interior.ColorIndex = 3
    End If
End Sub

```

```

Private Sub cmdBlock_Click()
    c = ActiveCell.Column
    r = ActiveCell.Row
    If c < n And r < m And c > 1 And r > 1 Then
        ActiveCell.Interior.ColorIndex = 5
    End If
End Sub

```

```
Private Sub cmdSet_Click()  
    cmdRoad.Enabled = True  
    cmdClear.Enabled = True  
    cmdDefault.Enabled = True  
End Sub
```

```
Private Sub cmdRoad_Click()  
    m = Selection.Row  
    n = Selection.Column  
    If m >= 3 And n >= 3 Then  
        Range(Cells(1,1),Cells(1,n)).Interior.ColorIndex=5  
        Range(Cells(m,1),Cells(m,n)).Interior.ColorIndex=5  
        Range(Cells(1,n),Cells(m,n)).Interior.ColorIndex=1  
        cmdRoad.Enabled = False  
        cmdCar.Enabled = True  
        cmdBlock.Enabled = True  
        cmdFree.Enabled = True  
        cmdReady.Enabled = True  
    Else  
        cmdClear_Click  
    End If  
End Sub
```

```
Private Sub cmdFinish_Click()  
    cmdStart.Enabled = False  
    cmdSet.Enabled = True  
    cmdStep.Enabled = False  
    cmdReady.Enabled = False  
    cmdRoad.Enabled = False  
    cmdClear.Enabled = False  
    cmdDefault.Enabled = False  
    cmdCar.Enabled = False  
    cmdBlock.Enabled = False  
    cmdFree.Enabled = False  
End Sub
```

```
Private Sub cmdStep_Click()  
    Call mdl.Step  
    t = t + 1  
    TextBoxSteps.Text = t  
    TextBoxAuto.Text = auto  
End Sub
```

```

Private Sub cmdFree_Click()
    c = ActiveCell.Column
    r = ActiveCell.Row
    If c < n And r < m And c > 1 And r > 1 Then
        ActiveCell.Interior.ColorIndex = xlNone
    End If
End Sub

```

```

Private Sub cmdStart_Click()
    cmdStart.Enabled = False
    cmdSet.Enabled = False
    cmdStep.Enabled = True

    Randomize
    Set mdl = New Model
    If IsNumeric(TextBoxP.Text) = True And
        CDb1(TextBoxP.Text) > 0 And
        CDb1(TextBoxP.Text) <= 1 Then
        pn = CDb1(TextBoxP.Text)
    Else
        pn = 0.1
        TextBoxP.Text = CStr(pn)
    End If
    t = 0
    auto = 0
    Workbooks(1).Sheets("Road").Activate
    Dim road As Range
    Set road = Range(Cells(1, 1), Cells(m, n))
    Call mdl.Init(road, pn)
End Sub

```

```

Private Sub cmdClear_Click()
    Cells.Select
    Selection.Interior.ColorIndex = xlNone
    Range("A1").Select
    cmdRoad.Enabled = True
    TextBoxSteps.Text = ""
    TextBoxP.Text = ""
End Sub

```

```

Private Sub cmdDefault_Click()
    Cells.Select
    Selection.Interior.ColorIndex = xlNone
    Range("A1").Select
    m = 6
    n = 30
    pn = 0.1
    Range(Cells(1,1), Cells(1,n)).Interior.ColorIndex=5
    Range(Cells(m,1), Cells(m,n)).Interior.ColorIndex=5
    Range(Cells(1,n), Cells(m,n)).Interior.ColorIndex=1
    cmdRoad.Enabled = True
    cmdReady.Enabled = True
    TextBoxP.Text = CStr(pn)
End Sub

```

```

Private Sub cmdReady_Click()
    cmdReady.Enabled = False
    cmdRoad.Enabled = False
    cmdClear.Enabled = False
    cmdDefault.Enabled = False
    cmdCar.Enabled = False
    cmdBlock.Enabled = False
    cmdFree.Enabled = False
    cmdStart.Enabled = True
End Sub

```

6. Создать на листе Road схему магистрали

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1																															
2																															
3																															
4																															
5																															
6																															
7																															

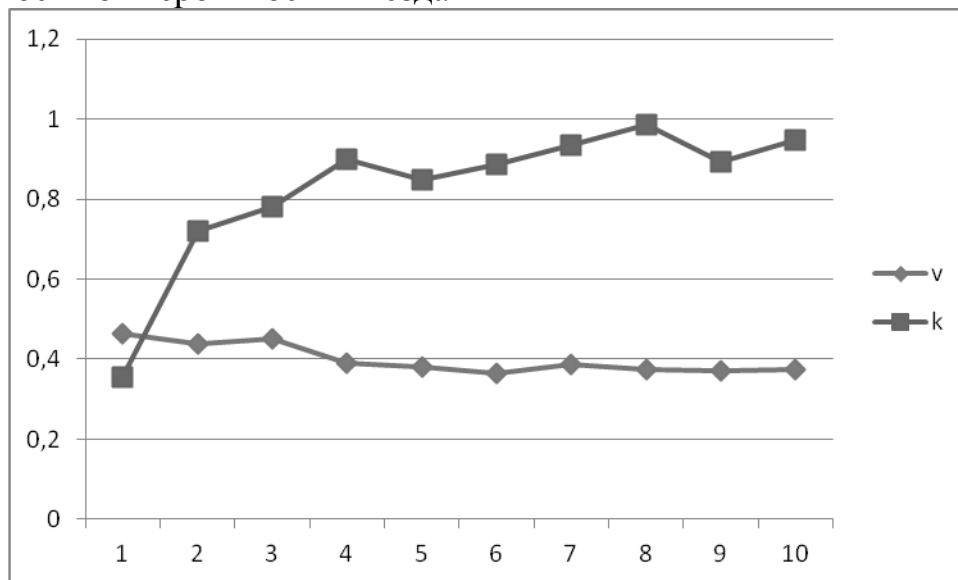
7. Установить вероятность въезда $p = 0,1$ и выполнить моделирование в пошаговом режиме, сделав не менее 150 шагов. Записать результаты в таблицу.

8. Повторить моделирование, изменяя значение вероятности въезда p до 1 с шагом 0,1. Составить таблицу.

Зависимость средней скорости и пропускной способности от вероятности въезда

p	v	k
0,1	0,465065	0,353333
0,2	0,438888	0,72
0,3	0,452567	0,78
0,4	0,390701	0,9
0,5	0,379546	0,846667
0,6	0,364097	0,886667
0,7	0,38544	0,933333
0,8	0,372525	0,986667
0,9	0,372445	0,893333
1	0,37311	0,946667

9. Построить график зависимости средней скорости и пропускной способности от вероятности въезда



V-Зависимость средней скорости от вероятности въезда

K- Зависимость пропускной способности от вероятности въезда

10. По графику, используя вставку линии тренда определить предельные значения средней скорости и пропускной способности.

11. Сформулировать результат моделирования:

Выводы.

1. Средняя скорость при полной загрузке участка – 0,39 клеток/шаг, что составляет 84% от максимальной.
2. Предельная пропускная способность участка – 0,89 машин/шаг – 89% от максимальной.

Варианты задания 5.2

1 16		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	1																														
	2																														
	3																														
	4																														
	5																														
	6																														
2 17		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	1																														
	2																														
	3																														
	4																														
	5																														
	6																														
3 18		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	1																														
	2																														
	3																														
	4																														
	5																														
	6																														
4 19		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	1																														
	2																														
	3																														
	4																														
	5																														
	6																														
5 20		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	1																														
	2																														
	3																														
	4																														
	5																														
	6																														
6 21		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	1																														
	2																														
	3																														
	4																														
	5																														
	6																														
7 22		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	1																														
	2																														
	3																														
	4																														
	5																														
	6																														

8	23		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		1																														
		2																														
		3																														
		4																														
		5																														
		6																														

9	24		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		1																														
		2																														
		3																														
		4																														
		5																														
		6																														

10	25		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		1																														
		2																														
		3																														
		4																														
		5																														
		6																														

11	26		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		1																														
		2																														
		3																														
		4																														
		5																														
		6																														

12	27		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		1																														
		2																														
		3																														
		4																														
		5																														
		6																														

13	28		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		1																														
		2																														
		3																														
		4																														
		5																														
		6																														

14	29		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		1																														
		2																														
		3																														
		4																														
		5																														
		6																														

15	30		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
		1																														
		2																														
		3																														
		4																														
		5																														
		6																														

* * *

Таким образом, расширение возможностей табличного процессора Excel средствами языка программирования Visual Basic for Application позволяет не только упростить и автоматизировать работу с электронными таблицами путем создания простых макросов, но и превратить Excel в мощный инструмент объектно-ориентированного и визуального программирования для решения прикладных задач любой сложности и математического моделирования.

Литература

1. Основы программирования в VBA: Методические указания по курсу «Информатика» для лабораторных и контрольных работ для студентов всех специальностей и направлений подготовки / Сост.: Ф.Г.Ахмадиев, И.Г.Бекбулатов, Ф.Г.Габбасов. – Казань: Изд-во КГАСУ, 2013. – 36 с.
2. Слепцова Л.Д. Программирование на VBA в Microsoft Office 2010. – М. : ООО «И.Д. Вильямс», 2010. – 432 с.
3. Уокенбах Д. Excel 2010: профессиональное программирование на VBA. – М. : ООО «И.Д. Вильямс», 2012. – 944 с.
4. Эйткен П. Интенсивный курс программирования в Excel за выходные. – М. : Издательский дом «Вильямс», 2004. – 432 с.
5. Труб И.И. Объектно-ориентированное моделирование на C++: Учебный курс. – СПб.: Питер, 2006. – 411 с.
6. Nagel K., Schreckenberg M. A cellular automaton model for freeway traffic // J. Physique I France. 1992. 2, pp. 2221–2229.

Оглавление

Введение	3
1. Составление алгоритмов и программ с использованием сложных структур данных Excel.	6
Варианты задания 1.1.....	9
Варианты задания 1.2.....	12
2. Структурный подход в программировании. Нисходящее проектирование.....	15
Варианты задания 2.....	20
3. Программирование функций и построение графиков с использованием объектной модели Excel.....	26
Варианты заданий 3.1 и 3.2.....	28
Варианты задания 3.3.....	35
Варианты задания 3.4.....	46
4. Объектно-ориентированное программирование в VBA.....	49
Задание 4. Моделирование дорожного движения на ремонтируемом участке дороги.....	50
5. Визуальное программирование в VBA.....	59
Варианты задания 5.1.....	65
Варианты задания 5.2.....	80