

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

Кафедра информационных
технологий и компьютерных
систем

**ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ VBA
В СРЕДЕ MS EXCEL 2007**

Методические указания
к выполнению лабораторных работ
по дисциплине «Информатика»
для студентов очной формы обучения
направления 38.03.01 «Экономика»
профили подготовки
«Бухгалтерский учет, анализ и аудит»,
«Финансы и кредит»

Севастополь
2017

УДК 004.912

МЕТОДИЧЕСКИЕ УКАЗАНИЯ к выполнению лабораторных работ «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ VBA В СРЕДЕ MS EXCEL 2007» по дисциплине «Информатика» для студентов очной формы обучения по направлению 38.03.01 «Экономика» профиль подготовки «Бухгалтерский учет, анализ и аудит», «Финансы и кредит» / Сост. доцент кафедры ИТиКС Е.В. Козлова, ст. преподаватель кафедры ИТиКС М.А.Лебедева, ассистент кафедры ИТиКС А.Г.Герцовский – Севастополь: Изд-во СГУ, 2017. – 32 с.

Целью методических указаний является оказание помощи студентам в выполнении лабораторных работ по программированию по дисциплине «Информатика».

Приведены теоретические сведения, необходимые для выполнения лабораторных работ, варианты заданий, примеры выполнения, рекомендации по выполнению, требования к оформлению работ.

Методические указания рассмотрены и утверждены на заседании кафедры Информационных технологий и компьютерных систем, протокол №9 от 19 апреля 2017г.

Рецензенты: И.А. Балакирева, доцент кафедры Информационных технологий и компьютерных систем, В.В. Альчаков, доцент кафедры Информатики и управления в технических системах

СОДЕРЖАНИЕ

1 Основы программирования на языке VBA.....	4
2 Лабораторная работа №1. Функции пользователя в MS EXCEL	11
3 Лабораторная работа №2. Разработка программ линейной структуры.....	13
4 Лабораторная работа №3. Разработка программ разветвляющейся структуры	20
5 Лабораторная работа №4. Разработка программ циклической структуры	24
6 Лабораторная работа №5. Одномерные массивы	27
Библиографический список.....	32

1 ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ VBA

1.1 Понятие алгоритма

Написанию программы всегда предшествует разработка некоторого плана (алгоритма) решения задачи.

Алгоритм – последовательность действий, позволяющая за конечное число шагов на основании исходных данных получить искомый результат.

Свойства алгоритма:

- дискретность — алгоритм состоит из отдельных шагов;
- детерминированность — каждый шаг алгоритма однозначно определен;
- массовость — алгоритм применяется к целому классу задач, при решении конкретной задачи исходные данные могут меняться в определенных пределах;
- результативность — за конечное число шагов достигается некоторый результат;
- понятность — алгоритм должен содержать команды, которые может выполнить конкретный исполнитель;
- конечность — алгоритм должен завершаться после конечного числа шагов.

Существуют различные формы записи алгоритмов, отличающиеся друг от друга наглядностью, компактностью, степенью формализации и другими показателями: словесный, графический (в виде блок-схем), язык программирования.

Графический способ записи алгоритмов является самым наглядным и предполагает использование определенных геометрических фигур - блоков. Каждому блоку ставятся в соответствие определенные действия (рис.1.1)

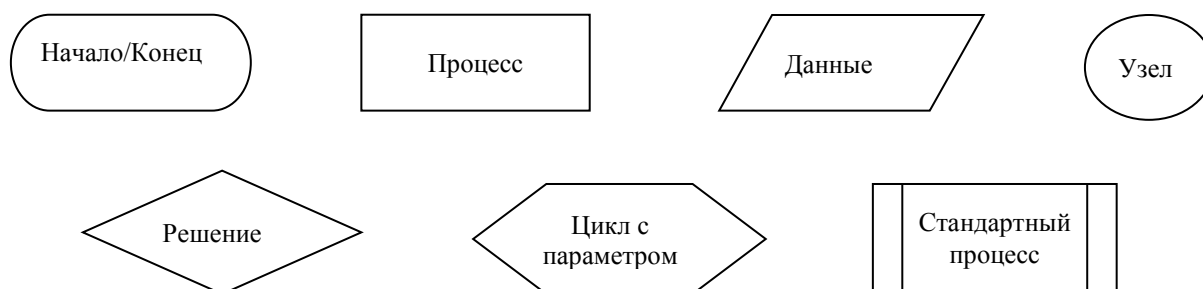


Рис. 1.1 - Элементы блок-схемы

Блок «Начало/Конец» отображает вход из внешней среды или выход из нее (наиболее частое применение – начало и конец программы). Внутри фигуры записывается соответствующее действие.

Блок «Процесс» отображает выполнение одной или нескольких операций. Внутри фигуры записывают непосредственно сами операции, например, операцию присваивания: $a = 10 * b + c$.

Блок «Данные» содержит список имен вводимых или выводимых данных.

Блок «Узел» отображает выход в часть схемы и вход из другой части этой схемы. Используется для обрыва линии и продолжения ее в другом месте, например, разделение блок-схемы, не помещающейся на листе.

Блок «Решение» содержит условие, например, $x > 3$, так как любое условие может быть только истинно или ложно, у блока один вход и два выхода, соответствующие действиям, когда условие истинно или ложно

Блок «Цикл с параметром» служит для изображения циклов, как правило, у него два входа (первый и повторный вход в цикл) и один выход, соответствующий завершению циклического процесса.

Блок «Стандартный процесс» отображает вызов стандартных или написанных пользователем подпрограмм.

При построении блок-схем блоки соединяются линиями в порядке выполнения операций. Блок схема алгоритма начинается блоком «Начало» и заканчивается блоком «Конец». Всегда должен существовать путь, соединяющий начальную и конечную вершины.

Пример блок-схемы алгоритма для вычисления значения выражения $y = \frac{x+2}{x}$ приведен на рис.1.2.

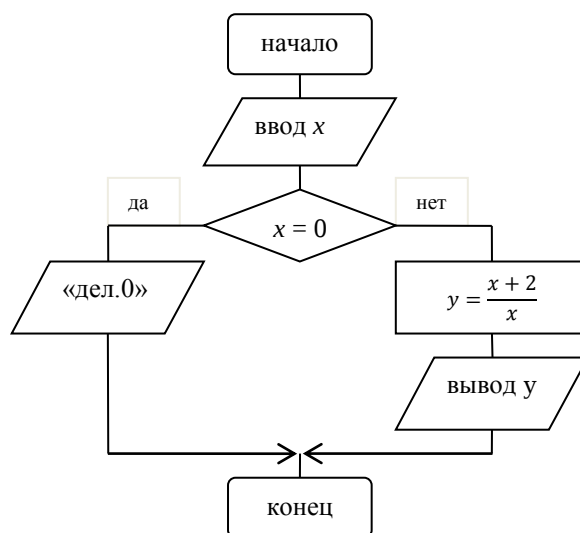


Рис. 1.2 - Блок-схема алгоритма

1.2. Основные понятия языка VBA

Программа - это *реализация* алгоритма на конкретном языке программирования. Язык VBA (Visual Basic for Applications) – язык визуального и событийно-управляемого программирования,

встроенный в приложения MS Office. Он используется как для решения вычислительных задач, так и для автоматизации деятельности, связанной с обработкой различных типов документов. Распространенной причиной использования VBA в Excel является автоматизация повторяющихся задач.

Алфавит языка VBA включает следующий набор символов:

1. прописные и строчные буквы латинского алфавита и русского алфавита A-Z, a-z;
2. арабские цифры 0,1,2,...,9;
3. знаки арифметических действий: *,-,+/, \ (целочисленное деление), ^ (возведение в степень);
4. знаки операций отношений =, <>, и т.д.
5. знаки препинания и разделители: ., : ; - ! ? «» _
6. (), [], пробел, подчеркивание
7. символы объявления типа: %, &; !, #, \$

Программирование на языке VBA основывается на основных понятиях: объект, данные, операторы, подпрограммы и модули.

Объектами в Excel является все то, с чем работает пользователь электронной таблицы - рабочие книги, электронные таблицы, ячейки и т.д. Каждый объект обладает *свойствами* - признаками, описывающими объект. К свойствам ячейки относятся содержимое ячейки, формула и различные параметры форматирования ячейки.

При обращении к какому-либо свойству конкретного объекта используется следующий синтаксис: **<Объект>.<Свойство объекта>**. Например, чтобы прочитать формулу, содержащуюся в текущей ячейке, нужно написать ActiveCell.Formula (ActiveCell - объект, а Formula - его свойство).

Кроме свойств, у объектов есть ряд методов. Метод - это действие, применяемое к объекту. Например, одним из методов для объекта Range (Диапазон) является ClearContents (Очистить содержимое). Этот метод позволяет очистить содержимое диапазона. При обращении к какому-либо методу конкретного объекта используется следующий синтаксис: **<Объект>.<Метод объекта>**: Range("A1:A10").ClearContents.

Данные являются элементарными конструкциями языка программирования и включают переменные, константы, структурированные данные.

Константы – величины, значения которых задаются заранее и не изменяются при выполнении программы. Переменные – величины, которые в процессе выполнения программы могут менять свое значение, под переменной понимают поименованную область памяти компьютера, в которой хранится значение этой переменной.

Имена переменных и констант должны удовлетворять следующим правилам:

- первым символом всегда должна быть буква;
- в составе имени нельзя использовать символы: !, @, &, \$, #, пробел;

- в качестве имени нельзя использовать ключевые (зарезервированные) слова, входящие в конструкции языка VBA;
- длина имени не может быть более 255 символов.

Данные имеют тип, указывающий, какие значения они могут хранить и какие операции над данными можно выполнять (таблица 1.1)

Таблица 1.1. Типы переменных и констант

Тип данных	Описание	Диапазон
Byte	Целое число	от 0 до 255
Integer	Целое число	от -32 768 до 32767
Long	Длинное целое число	от -2 147 483 648 до 2 147 483 647
Single	Число с плавающей запятой одинарной точности	а) для отрицательных чисел: от -3,402823E38 до -1,401298E-45 б) для положительных чисел: от 1,401298E-45 до 3,402823E38
Double	Число с плавающей запятой двойной точности	а) для отрицательных чисел: от -1,79769313486231E308 до -4,94065645841247E-324 б) для положительных чисел: от 4,94065645841247E-324 до 1,79769313486231E308
Currency (денежный)	Число с фиксированной десятичной точкой	от -922.337.203.685.477.5808 до 922.337.203.685.477.5807
String	Строка символов	от 0 до 147483647 символов
Variant	Универсальный	Значения любого типа
Date	Дата	от 1.01.100 года до 31.12. 9999 года
Object	Объект	Ссылка на любой объект

Операторы используются для описания объектов и для выполнения организации вычислительного процесса.

Описание данных имеет вид: Dim | Private | Public Данные as Тип. Объявление связывает имя переменной или константы с ее типом, заданным конструкцией As. Когда она опущена, переменная получает тип Variant.

Пример определения константы: Public Const pi As Double = 3.141593.

Переменные и константы, объявленные как Private, сохраняют своё значение только во время выполнения блока кода, в котором они определены.

Переменные и константы, объявленные как **Public**, сохраняют своё значение и вне блока кода, в котором они определены, то есть до конца выполнения программы.

Переменные и константы, объявленные без ключевых слов сохраняют своё значение согласно месту их объявления (описания).

Операторы, используемые для организации вычислительного процесса, производят действия с объектами и переменными. К ним относят операторы присваивания, условные операторы, операторы цикла, операторы ввода-вывода данных.

Программа VBA представляет собой совокупность процедур и функций, размещенных, в зависимости от особенностей решаемой задачи, в одном или нескольких модулях. Каждый модуль имеет две области: общую область и область подпрограмм. В общей области помещаются операторы описания переменных, которые являются общими для всех процедур и функций этого модуля. В области подпрограмм помещается код программы. В VBA программный код, реализующий какие-либо действия, оформляется в виде процедур и функций.

Различают следующие типы процедур:

- процедуры обработки событий;
- процедуры макросов;
- процедуры (функций) пользователя.

1.3 Среда программирования VBA в EXCEL

Проекты VBA создаются с помощью среды разработки, создаваемой редактором Visual Basic. Для запуска среды на вкладке *Разработчик* в группе *Код* необходимо нажать кнопку **Visual Basic**, либо нажать клавиши <Alt>+<F11>.

Интерфейс VBA состоит из следующих основных компонентов: окно проекта, окно свойств, окно редактирования кода, окна форм, меню и панели инструментов.(рис.1.3)

Окно проекта в редакторе Visual Basic активизируется выбором команды *View→Project Explorer* (Вид→Окно проекта). В этом окне для каждой открытой рабочей книги создаётся проект VBA (VBA Project). Проект VBA – это набор всех объектов и модулей VBA, привязанных к текущей книге. В проекте автоматически создается модуль для каждого рабочего листа *Лист* и для всей книги *ЭтаКнига*. Стандартные модули, модули классов и пользовательских форм добавляются с помощью команды главного меню *Insert*. При создании нового кода VBA необходимо придерживаться правил:

- код, который относится к рабочей книге, вводится в объект *ЭтаКнига* ;
- код, который относится к рабочему листу, должен быть введён в объект *Лист*;

- код более общего характера должен быть введен в *Module*;
- код для нового объекта должен быть введен в *Class Module*;
- для создания диалогового окна для взаимодействия с пользователем используют *Userform*.

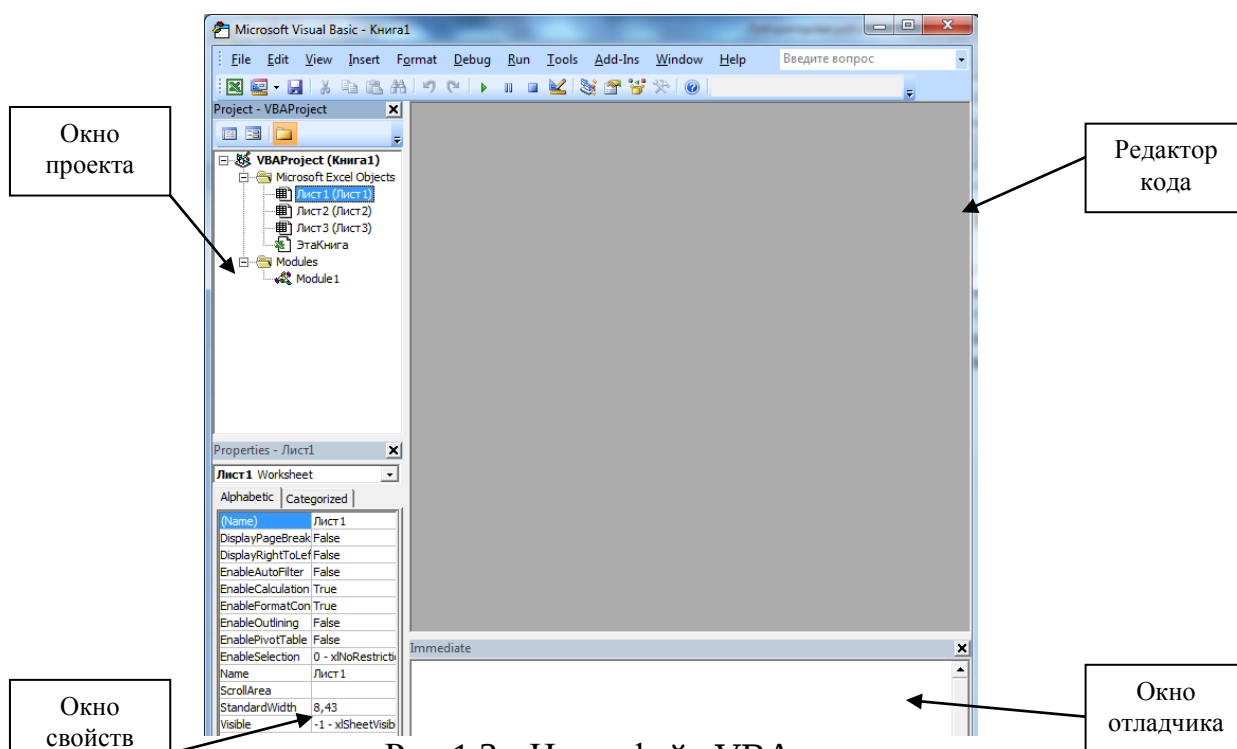


Рис.1.3 - Интерфейс VBA

Окно свойств активизируется выбором команды *View→Properties Windows* (Вид→Окно свойств). Используя это окно, можно просмотреть свойства объекта, выделенного в окне проекта, и изменить их установки. Окно свойств состоит из двух частей: в верхней части окна свойств располагается раскрывающийся список, из которого можно выбрать объект, рабочая область состоит из двух вкладок: «По алфавиту» и «По категориям», отображающих набор свойств в алфавитном порядке и по категориям.

Редактор кода используется для ввода и изменения кода процедур и функций приложения. Окно редактора кода для модуля соответствующего объекта открывается двойным щелчком на значке объекта в окне проекта. Для создания нового модуля необходимо выполнить команду *Insert – Module*. По мере ввода кода VBA в окно *Code*, редактор *Visual Basic* следит за правильностью ввода, ищет ошибки в коде и выделяет код, который требует исправления. Редактор *Visual Basic* является обычным текстовым редактором, обладающим возможностью редактирования текста (копирование, перемещение, удаление). В окне редактирования можно просматривать текст всех процедур модуля (кнопка *Full Module View*) или одну выбранную процедуру (кнопка *Procedure View*).

Окно отладчика можно отобразить в редакторе Visual Basic через меню *View > Immediate Window*. Это окно помогает при отладке кода. Оно выполняет роль области вывода для отладки выражений и позволяет вычислять отдельные выражения или выполнять строки кода по одной.

Окно переменных можно отобразить через меню нажмите *View > Locals*. В этом окне отображаются все переменные, объявленные в текущей процедуре. Окно делится на столбцы, в которых содержатся имя, значение и тип каждой переменной, и эта информация обновляется автоматически в ходе выполнения программы. Окно *Locals* очень полезно при отладке кода VBA.

Окно редактирования форм (UserForm) служит для создания диалоговых окон, разрабатываемых приложений в VBA. Редактор форм является одним из основных инструментов визуального программирования. Форма в проект добавляется с помощью команды *Insert > Form* (Вставка - Форма). В результате на экран выводится незаполненная форма с панелью инструментов Панель элементов (Toolbox). Используя панель инструментов Панель элементов из незаполненной формы, можно сконструировать любое требуемое для приложения диалоговое окно.

При создании программы на VBA в книге Excel на вкладке *Разработчик* в группе *Код* нажать кнопку *Безопасность макросов*, в появившемся окне установить переключатель *Включить все макросы*. Для сохранения модуля, созданного в редакторе VBA, выполнить команду *File > Save* (Файл – Сохранить), в окне «Сохранение документа» в поле «Имя файла» ввести имя, в поле «Тип файла» выбрать тип «Книга Excel с поддержкой макросов».

2 ЛАБОРАТОРНАЯ РАБОТА № 1

Функции пользователя в MS Excel

2.1 Цель работы: Получение практических навыков создания функций пользователя в EXCEL с использованием Visual Basic for Application (VBA), применение функций пользователя для вычислений на рабочем листе

2.2 Теоретические сведения

2.2.1 Понятие функции пользователя

Функции предназначены для вычисления единственного результата, зависящего от входных данных, называемых аргументами функции. В MS Excel имеется большое количество встроенных функций. При необходимости пользователь может создавать собственные функции с помощью языка VBA. По умолчанию все пользовательские

функций попадают в категорию *Определённые пользователем*. Вставка функций, определенных пользователем, на рабочий лист осуществляется так же, как и любых стандартных функций – с помощью *Мастера функций*. Аргументы функции можно вводить в ячейки рабочего листа либо задавать в полях ввода диалогового окна Мастера функций.

Функция – это специальным образом оформленный блок программного кода, который выполняет действия в пределах своего блока и возвращает значение. Пользовательские функции строятся в стандартном модуле редактора VBA.

Синтаксис функции на языке VBA:

[Private | Public] Function имя_функции (аргументы) As Тип
список операторов VBA

имя = выражение

End Function

Каждая функция начинается ключевым словом *Function*, за которым следует имя функции, задаваемое пользователем, и заканчивается оператором *End Function*. Аргументы – список переменных, представляющих значения, которые переданы этой функции для обработки. Аргументы могут отсутствовать.

[Private | Public] – необязательные ключевые слова, определяющие область видимости функции. После зарезервированного слова *As* указывается тип результата, хранящегося в имени функции. Если часть *As Тип* отсутствует, результат имеет тип *Variant*.

Операторы в функции обрабатывают текст, управляют входными данными или вычисляют значения. Оператор *имя_функции = выражение* говорит о том, что имени, определённому в первой строке функции, присваивается значение выражения, которое и становится возвращаемым значением функции. В теле функции этот оператор должен присутствовать обязательно.

2.2.2. Пример использования функции пользователя

Постановка задачи: построить функцию пользователя, вычисляющую значение функции в заданных точках $y = \frac{e^{x-2}}{\sin(\pi \cdot x)}$, где $x = 0,8$.

На вкладке *Разработчик* в группе *Код* нажать кнопку Visual Basic. В появившемся окне среды разработки в окне проектов выделить объект *ЭтаКнига* и выполнить команду *Insert – Module*. Для вставки шаблона функции можно выполнить команду *Insert – Procedure*, в появившемся окне *Add Procedure* установить переключатель *Function*, в поле *Name* ввести имя функции – *y*. Затем ввести код функции:

Function y(x As Single) As Single

y = Exp(x - 2) / Sin(Application.Pi() * x)

End Function

`Application.Pi()` – вычисление числа пи (3,1417) с помощью стандартной функции Excel. Объект `Application` представляет само приложение Excel и позволяет вызывать более встроенные функции рабочего листа с помощью конструкции:

`Application.ФункцияРабочегоЛиста(Аргументы).`

После ввода программы сохранить книгу, указав тип «с поддержкой макросов». Переключиться на рабочий лист, в ячейки A1 и B1 ввести соответственно имена «x» и «y». В ячейку A2 ввести значение переменной x - 0,8. Выделить ячейку B2 и вставить функцию с помощью мастера функций, на первом шаге выбрать категорию «Определенные пользователем», имя функции, на втором шаге задать аргумент (рис. 2.1)

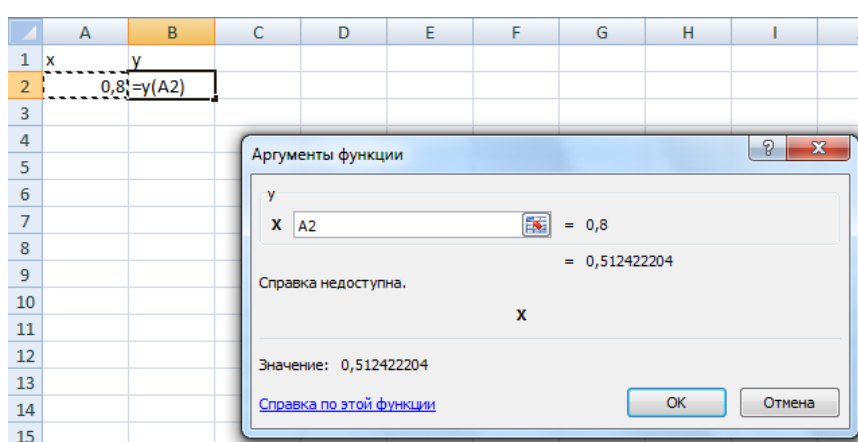


Рис.2.1 - Вставка функции пользователя на рабочий лист

2.3 Последовательность выполнения работы

2.3.1 В рабочей книге MS Excel создать функцию пользователя на языке VBA в соответствии с вариантом (таблица 2.1).

2.3.2 Вставить функцию на рабочий лист, задав значения аргументов, получить значения функции

2.3.3 Оформить отчет

Таблица 2.1. Варианты заданий

Вариант	Функция	Аргументы
1	$z = \cos^2(x) - \sin^2(y) + 2 \cdot \sin(x) \cdot \cos(y)$	$x=1,12$ $y=0,67$
2	$z = \tan(0,2 \cdot x) - \sin^2(y)$	$x=1,25$ $y=0,825$
3	$z = \log(x + 6,5) + \sin(y)$	$x=0,335$ $y=0,625$
4	$z = \frac{1}{ \sin(2 \cdot y) + 0,5 } + 2 \cdot \cos^2(x)$	$x=2,215$ $y=0,823$
5	$z = \cos(\cos(x)) + \sin(\sin(x))$	$x=1,432$ $y=0,98$
6	$z = e^{x \cdot \sin(y)}$	$x=-1,72$ $y=0,55$
7	$z = e^{2 \cdot \sin(x) - \cos(y)}$	$x=2,03$ $y=1,817$
8	$z = \cos^2(x) + 2 \cdot \sin(x) \cdot \cos(y)$	$x=2,014$ $y=1,16$

9	$z = \tan(0,2 \cdot x^2) + \sin(y)$	$x=3,125 \quad y=1,24$
10	$z = \log(x+y) + \sin(2 \cdot y)$	$x=1,83 \quad y=1,359$
11	$z = \frac{1}{ \sin(2 \cdot y^3) + 0,2 } + 2 \cdot \cos(x)$	$x=1,83 \quad y=1,359$
12	$z = \cos(\cos(x)) + 4 \cdot \sin(x)$	$x=0,767 \quad y=1,13$
13	$z = x + e^{x \cdot \sin(3 \cdot y)}$	$x=1,99 \quad y=0,122$
14	$z = e^{2 + \sin(x) - \cos(y)}$	$x=0,543 \quad y=1,18$
15	$z = e^{\sin(y)} + \cos(x)$	$x=3,43 \quad y=0,621$

2.3 Содержание отчета

Цель работы, постановка задачи, краткие теоретические сведения, текст процедуры на VBA с пояснениями, анализ результатов, выводы

2.5 Контрольные вопросы

1. Какова структура функции пользователя?
2. Как в тексте функции записываются встроенные функции рабочего листа?
3. Сколько результатов можно вычислить с помощью функции?
4. Как вставить функцию пользователя на рабочий лист?
5. Из чего состоит заголовок функции?
6. Может ли список аргументов быть пустым?

3 ЛАБОРАТОРНАЯ РАБОТА № 2

Разработка программ линейной структуры

3.1 Цель работы: Получение практических навыков разработки программ линейной структуры в EXCEL с использованием Visual Basic for Application (VBA),

3.2 Теоретические сведения

Линейные алгоритмы – алгоритмы, в котором команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий. Программы линейной структуры, как правило, содержат операторы присваивания и операторы, осуществляющие ввод исходных данных и вывод результата.

3.2.1 Оператор присваивания

Оператор присваивания используют, чтобы присвоить переменной, константе или свойству объекта нужное значение.

Синтаксис: *Имя_Переменной = Выражение*

Операция присваивания предписывает выполнить выражение, заданное в его правой части, и присвоить результат имени переменной, имя которой указано в левой части.

Выражение – это последовательность операндов, объединенных между собой знаками операций. В качестве операнда могут быть использованы: константы, переменные, функции или заключенные в круглые скобки выражения. В зависимости от типа операндов и используемых операций выражения делятся на: арифметические, логические и строковые (текстовые).

Арифметическим называется выражение, содержащее операнды только арифметического (вещественного и/или целого) типа и знаки математических операций:

+ (сложение), - (вычитание), * (умножение), ^ (возведение в степень) – используются для операндов и вещественного, и целого типа
/ (деление) используются для операндов вещественного типа
\ (целочисленное деление), mod (операция нахождения остатка от деления) - используются для операндов только целого типа.

Выражение, содержащее хотя бы один знак логической операции, называется **логическим**. Результатом выполнения такого выражения является логическая величина со значением **True** (истина), либо **False** (ложь). Логическими операциями могут быть:

операции отношения (сравнения). Операнды этих операций могут иметь любой тип, кроме логического типа. К этим операциям относятся: > (больше), < (меньше), >= (больше или равно), <= (меньше или равно), = (равно), <> (не равно)

специальные логические операции. Операнды этих операций могут иметь только логический тип. К этим операциям относятся: AND (логическое умножение), OR (логическое сложение), NOT (логическое отрицание).

Строковое (текстовое) выражение может содержать операнды только символьного (текстового или строкового) типа. В VBA, имеется строковая операция, которая носит название **конкатенация** или **слияние подстрок в строку**. Запись этой операции можно производить как знаком “+”, так и знаком “&” (принято использовать знак “&”).

VBA выполняет операции в соответствии с их приоритетами (Таблица 3.1). Для изменения приоритета операция используют скобки

Таблица 3.1. Приоритеты операций

Приоритет	Операция
1	Вызов функции и выражения в скобках
2	^ (возведение в степень)
3	- (смена знака числа)
4	* (умножение), / (деление), \ (целочисленное деление), Mod (остаток от деления)

5	+ и – (сложение и вычитание)
6	<, >, >=, <=, <>, =
7	Not
8	And
9	Or
10	Xor

В VBA имеется большой набор встроенных функций, использование которых существенно упрощает программирование (таблица 3.2). Эти функции можно разделить на следующие основные категории: математические функции; функции обработки строк; функции проверки, определения и преобразования типов данных; функции времени и даты; финансовые функции и др. Функция задается с помощью указателя функции, который записывается в виде имени, и аргументов, заключенных в круглые скобки: *Имя_функции(аргументы)*. Аргументами могут быть константы, переменные, функции, арифметические выражения.

Таблица 3.2. Встроенные функции VBA

Математическая запись	Функция VBA	Описание	Пример
$\sin x$	Sin(x)	Синус угла в радианах	$\sin^2 x \rightarrow \sin(x)^2$
$\cos x$	Cos(x)	Косинус угла в радианах	$\cos x^2 \rightarrow \cos(x^2)$
$\operatorname{tg} x$	Tan(x)	Тангенс угла в радианах	$\operatorname{Tg}^2 x \rightarrow \tan(2 * x)$
$\operatorname{Arctg} x$	Atn(x)	Арктангенс	$2 \operatorname{Arctg} x \rightarrow 2 * \operatorname{Atn}(x)$
$ x $	Abs(x)	Модуль числа	$ 2x + 6 \rightarrow \operatorname{Abs}(2 * x + 6)$
$\sqrt{x}$	Sqr(x)	Корень квадратный	$\sqrt{x^2 + 3} \rightarrow \operatorname{Sqr}(x^2 + 3)$
e^x	Exp(x)	Экспонента	$5e^{-3x} \rightarrow 5 * \operatorname{Exp}(-3 * x)$
$\ln x$	Log(x)	Натуральный логарифм	$\operatorname{Lg} x \rightarrow \operatorname{Log}(x) / \operatorname{Log}(10)$
	Int(x)	Округление до ближайшего меньшего целого	$\operatorname{Int}(10,9)=10$
	Fix(x)	Отбрасывание дробной части числа	$\operatorname{Fix}(10,9)=10$
	Sgn(x)	Знак числа	
	Val(x)	Преобразование строки в число	
	Str(x)	Преобразование числа в строку	

3.2.1 Ввод и вывод данных с помощью диалоговых окон

В VBA ввод и вывод информации (для взаимодействия с пользователем) можно осуществлять в диалоговых окнах. Диалоговое окно ввода значений реализуется встроенной функцией InputBox. В окне

ввода, реализованное функцией `InputBox`, отображается поле для ввода значения переменной. Далее пользователь должен нажать кнопку ОК.

Функция `InputBox()` имеет следующий синтаксис:

Имя_Переменной = InputBox(Prompt, [Title], [Default], [XPos], [YPos])

где аргументы: `Prompt` или Сообщение - обязательный аргумент, который задает в диалоговом окне информационное сообщение. Все остальные аргументы являются необязательными.

`Title` - строковая константа или переменная, значение которой будет отображаться в заголовке окна

`Default` - строковая константа или переменная, значение которой будет отображаться в поле ввода;

`xPos`, `yPos` – положение окна ввода соответственно по горизонтали и вертикали.

Функция ввода выводит на экран диалоговое окно, содержащее сообщение, после ввода и две кнопки ОК и Cancel. Устанавливает режим ожидания ввода текста пользователем и нажатия кнопки <ОК> или <Отмена>. Если пользователь выбрал <ОК>, окно ввода возвращает значение типа `String`, находящийся в поле ввода. Если пользователь выбрал <Отмена>, окно ввода возвращает пустую строку (`Empty`).

Для вывода информации применяются диалоговые окна сообщений, реализуемые оператором **`MsgBox`** или функцией **`MsgBox()`**. `MsgBox` осуществляет вывод информации в диалоговом окне и устанавливает режим ожидания нажатия кнопки пользователем.

Оператор `MsgBox` имеет следующий синтаксис:

`MsgBox Prompt, [Buttons], [Title], [HelpFile], [Context]`

где аргументы: `Prompt` или Сообщение - обязательный аргумент, задающий в окне выводимое информационное сообщение. Все остальные аргументы являются необязательными.

`Buttons` - Кнопки, которые можно использовать в диалоговом окне вывода сообщений. В окне сообщений могут применяться различные кнопки (ОК, Отмена и т.д.). Если не указывать, какие кнопки необходимо отображать в окне сообщений, то по умолчанию отображается кнопка ОК. Кроме того, в диалоговых окнах вывода сообщений можно использовать различные значки (`vbQuestion` - значок вопросительного знака, `vbExclamation` - значок восклицательного знака и т.д.).

`Title` - строковое выражение, которое будет отображаться в строке заголовка окна сообщения.

3.2.2 Создание программ в VBA

Программы на языке VBA оформляются в виде процедур. Процедура представляет собой именованную часть кода, выполняющую определенные действия: ввод и вывод данных, вычисление значения выражения и т.д. Процедура может иметь параметры и в результате

выполнения последовательности инструкций изменять их значения. Синтаксис:

[Private | Public] Sub ИмяПроцедуры (СписокПараметров) As Тип
Тело процедуры (строки кода)
End Sub

Перед заголовком процедуры Sub могут быть необязательные ключевые слова, означающие область видимости данной процедуры: Public(устанавливается по умолчанию) –может быть вызвана из любой части программы, Private –можно вызывать только из того модуля, в котором она расположена. Список параметров процедуры может быть пустым.

Тело процедуры содержит: объявления переменных и констант и последовательность инструкций (операторов). Инструкции представляют собой последовательность логических строк. Обычно в каждой строке вводят только один оператор. Если в одной строке записывают подряд несколько операторов, то друг от друга их отделяют двоеточием: $a = 2 : b = a + 3$

Для пояснения хода выполнения программы рекомендуется использовать комментарии. Они вводятся в программу с помощью оператора REM или одинарной кавычки (апострофа). Часть строки после оператора комментария не выполняется.

Для создания процедуры запустить VBA нажатием кнопки Visual Basic. В появившемся окне среды разработки выполнить команду *Insert – Module*. В окне редактора кода ввести текст процедуры. При вводе ключевые слова VBA показываются синим цветом, комментарии – зеленым. Для запуска программы на выполнение установить курсор в любом месте программы и нажать F5 либо в главном меню выбрать команду Run или использовать кнопку  на панели инструментов.

3.2.4 Пример разработки программы линейной структуры

Постановка задачи: разработать алгоритм и составить программу решения задачи вычисления значения переменной z :
 $z = \sin^2 y \cdot x^{y/x} - \sqrt[3]{y/x}$ для исходных данных: $x=1,325$; $y=8,166$.

Блок схема алгоритма приведена на рис. 3.1.

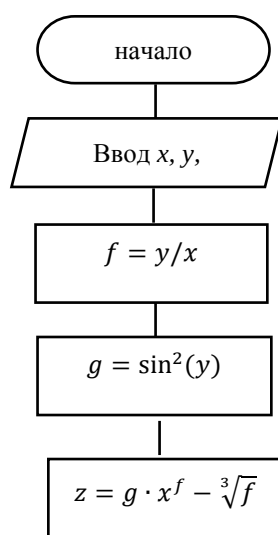


Рис. 3.1 - Блок-схема алгоритма

Текст программы:

```
Sub LINPROGR()
    X = Val(InputBox("Введите X"))    ' ввод X
    Y = Val(InputBox("Введите Y"))    ' ввод Y
    MsgBox "X=" & Str(X) & "Y=" & Str(Y) ' вывод
    F = Y / X
    G = Sin(y)^2
    Z = G * X ^ F - F ^ (1 / 3)
    MsgBox "Z=" & Str(Z)              ' вывод результата
End Sub
```

Функция **Val(строка)** - возвращает числа, содержащиеся в строке, как числовое значение соответствующего типа. В качестве разделителя целой и дробной части распознается *только* точка (.)

Функция **Str(число)** - возвращает значение типа Variant (String), являющееся строковым представлением числа.

Набрать программу в окне редактора VBA и запустить на выполнение командой RUN. В появившихся диалоговых окнах ввести значения исходных данных x и y (рис.3.2) и нажать ОК. В качестве разделителя целой и дробной части в VBA используется точка.

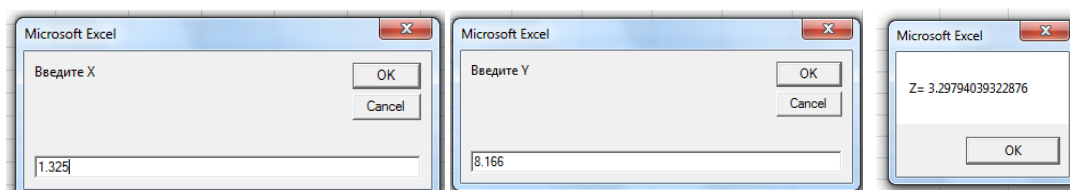


Рис. 3.2 - Диалоговые окна ввода и вывода

3.3 Последовательность выполнения работы

3.3.1 Разработать блок-схему решения задачи вычисления значений переменных в соответствии с вариантом (таблица 3.1).

3.3.2 В рабочей книге MS Excel создать процедуру на языке VBA для решения задачи

3.3.3 Выполнить программу для заданных исходных данных

3.3.4 Оформить отчет

Таблица 3.1 Варианты задания

Вариант задания	Расчетные формулы	Значения исходных данных
1	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}; b = 1 + \frac{z^2}{3+z}$	$x=1,42 \quad y=-1,2$ $z=3,5$
2	$\gamma = x^{y/x} - \sqrt[3]{y/x}$ $\psi = (y-x) \frac{y-z(y-x)}{1+(y-x)}$	$x=1,825$ $y=18,225$ $z=-3,29$
3	$s = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$ $\psi = x(\sin x^3 + \cos^2 y)$	$x=0,335$ $y=0,025$
4	$y = e^{-bt} \sin(at+b) - \sqrt{ bt+a }$ $s = b \sin(at^2 \cos 2t) - 1$	$a=-0,5 \quad b=1,7$ $t=0,44$
5	$\omega = \sqrt{x^2+b} - b^2 \sin^3(x+a)/x$ $y = \cos^2 x^3 - x/\sqrt{a^2+b^2}$	$a=1,5 \quad b=15,5$ $x=-2,9$
6	$s = x^2 \operatorname{tg}^2(x+b)^2 + a/\sqrt{x+6}$ $Q = \frac{bx^2 - a}{e^{bx} - 1}$	$a=16,5$ $b=3,4$ $x=0,61$
7	$R = x^2(x+1)/b - \sin^2(x+a)$ $s = \sqrt{xb/a} + \cos^2(x+b)^3$	$a=0,7 \quad b=0,05$ $x=0,5$
8	$y = \sin^3(x^2+a)^2 - \sqrt{x/6}$ $z = \frac{x^2}{a} + \cos(x+b)^3$	$a=1,1 \quad b=0,004$ $x=0,2$
9	$f = \sqrt{mtgt} + c \sin t $ $z = m \cos(bt \sin t) + c$	$m=2, \quad c=-1$ $t=1,2, \quad b=0,7$
10	$y = btg^2 x - \frac{a}{\sin^2(x/a)}$ $d = ae^{\sqrt{a}} * \frac{x + \sqrt{x+a}}{x - \sqrt{ x-b }}$	$a=3,2$ $b=17,5$ $x=4,8$
11	$f = \ln(a+x^2) + \sin^2(x/b)$ $z = e^{-ex} * \frac{x + \sqrt{x+a}}{x - \sqrt{ x-b }}$	$a=10,2 \quad b=9,2$ $x=2,2$ $c=0,5$
12	$y = \frac{a^{2x} + b^{-x} \cos(a+b)x}{x+1}$ $R = \sqrt{x^2+b} - b^2 \sin^3(x+a)/x$	$a=0,3 \quad b=0,9$ $x=0,61$

13	$z = \sqrt{ax \sin 2x + e^{-2x}(x+b)}$ $\omega = \cos^2 x^3 - x / \sqrt{a^2 + b^2}$	$a=0,5 \quad b=3,1$ $x=1,4$
14	$U = \frac{a^2 x + e^{-x} \cos bx}{bx - e^{-x} \sin bx + 1}$ $f = e^{2x} \ln(a+x) - b^{3x} \ln(b-x)$	$a=0,5 \quad b=2,9$ $x=0,3$
15	$z = \frac{\sin x}{\sqrt{1+m^2 \sin^2 x}} - c^* m^* \ln(mx)$ $s = e^{-ax} \sqrt{x+1} + e^{-bx} \sqrt{x+1,5}$	$m=0,7 \quad a=0,5$ $b=1,08 \quad x=1,7$ $c=2,1$

3.4 Содержание отчета

Цель работы, постановка задачи, краткие теоретические сведения, текст процедуры на VBA с пояснениями, анализ результатов, выводы

3.5 Контрольные вопросы

1. Какова структура процедуры в VBA?
2. Из каких операторов состоит программа линейной структуры?
3. Приведите примеры встроенных функций VBA.
4. Какой функцией реализуется диалоговое окно ввода?
5. Как в программе осуществить вывод данных в диалоговое окно вывода?
6. Правила записи комментариев.

4.ЛАБОРАТОРНАЯ РАБОТА № 3

Разработка программ разветвляющейся структуры

4.1 Цель работы: Получение практических навыков разработки программ разветвляющейся структуры в EXCEL с использованием Visual Basic for Application (VBA),

4.2 Теоретические сведения

Ветвление - это управляющая операция языка программирования, позволяющая изменить порядок выполнения алгоритма программы.

Для реализации ветвлений в программах на VBA используют: условный оператор If; оператор выбора Select Case; оператор безусловного перехода Goto.

Условный оператор If позволяет проверить некоторое условие и в зависимости от результатов проверки выполнить то или иное действие. В VBA существует два типа условного оператора: линейный и блочный.

1. Линейный условный оператор используется для того, чтобы выполнить какой-либо один оператор, если некоторое условие будет истинным. Синтаксическая конструкция линейного оператора имеет две формы: безальтернативную, альтернативную.

Структура **безальтернативного** **условного** **оператора** (сокращенный вариант): **If <условие> Then <оператор 1>**

<условие> - выражение логического типа;

<оператор 1>, <оператор 2> - операторы языка VBA.

<Оператор 1> выполняется, если <условие> истинно. Если условие ложно, осуществляется переход к следующему за **If** оператору.

Структура **альтернативного** **условного** **оператора** (полный вариант): **If <условие> Then <оператор 1> Else <оператор 2>**

где If, Then, Else - зарезервированные слова (если, то, иначе);

Если <условие> истинно, выполняется <оператор 1>, если условие ложно, выполняется <оператор 2>. Линейные условные операторы записываются на одной строке. Пример: IF x>0 Then y=sqr(x) Else y=x^2.

2. Блочный условный оператор используется, если в случае истинности условия необходимо выполнить несколько программных операторов (блок операторов). Каждая из ветвей условного оператора может содержать произвольное количество операторов, записываемых с новой строки. Блочный оператор имеет две формы: безальтернативный, альтернативный.

Структура безальтернативного блочного оператора (сокращенный вариант)

If <условие> Then
 <оператор1>

End If

Структура альтернативного блочного оператора:

If <условие1> Then
 <оператор1>

Elseif <условие2>
 <оператор2>

Else
 <оператор3>

End If

где End If - указывает на окончание блока оператора If.

Конструкция: If...Then...ElseIf применяется в случаях, когда необходимо рассмотреть еще несколько условий в дополнение к исходному, и выбрать один из нескольких вариантов (количество вариантов больше 2).

4.2.2. Пример разработки программы

Постановка задачи: разработать алгоритм и составить программу решения задачи вычисления значения переменной y :

$$y = \begin{cases} \sqrt{2+x^2}; x \leq -3 \\ \sin^3 x; -3 < x \leq 3 \\ x^2 - 3; x > 3 \end{cases}$$

Блок схема алгоритма приведена на рис. 3.1

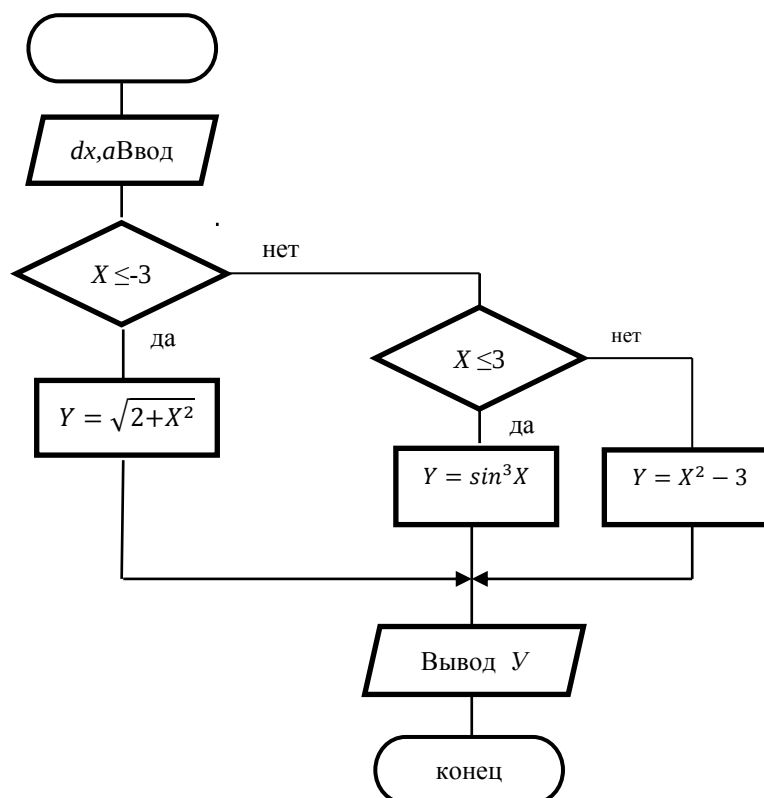


Рис. 3.1. Блок-схема алгоритма

Текст программы

```
Sub RazvetvlPr()
```

```
X = Val(InputBox("Введите X"))
```

```
If X <= -3 Then
```

```
    Y = Sqr(2 + X ^ 2)
```

```
ElseIf X < 3 Then
```

```
    Y = Sin(X) ^ 3
```

Else

$$Y = X^2 - 3$$

End If

MsgBox "X=" & Str(X) & "Y=" & Str(Y)

End Sub

Набрать программу в окне редактора VBA и запустить на выполнение командой RUN.

4.3 Последовательность выполнения работы

4.3.1 Разработать блок-схему решения задачи вычисления значения переменной y в соответствии с вариантом (таблица 3.1).

4.3.2 В рабочей книге MS Excel создать процедуру на языке VBA для решения задачи

4.3.3 Выполнить программу для исходных данных из каждого диапазона

4.3.4 Оформить отчет

Таблица 4.1. Варианты заданий

№ вар	Функция	№ вар	Функция
1	$y = \begin{cases} 2-x ^{\frac{1}{3}}, & x \geq 1 \\ -x + 2e^{-2x}, & 0 < x < 1 \\ \frac{1+x}{\sqrt[3]{1+x^2}}, & x \leq 0 \end{cases}$	8	$y = \begin{cases} x + \frac{1}{x}, & x < 2 \\ x^3 \sin x, & 2 \leq x < 3 \\ x + 3, & x \geq 3 \end{cases}$
2	$y = \begin{cases} \frac{1+\ln x}{x}, & x > 1 \\ \sqrt{1-x}, & 0 < x \leq 1 \\ -x^2, & x \leq 0 \end{cases}$	9	$y = \begin{cases} x\sqrt{1-x}, & x < 0 \\ \frac{x^2-1}{x^2+1}, & 0 \leq x \leq 1 \\ \sin x, & x > 1 \end{cases}$
3	$y = \begin{cases} x\sqrt{1-x}, & x < 0 \\ \frac{x^2-1}{x^2+1}, & 0 \leq x < 1 \\ \sin x, & x \geq 1 \end{cases}$	10	$y = \begin{cases} x\sqrt{1-x}, & x < 0 \\ \frac{2x}{1-x^3}, & 0 \leq x < 1 \\ \sin x, & x \geq 1 \end{cases}$
4	$y = \begin{cases} x\sqrt{1-x}, & x < 0 \\ \frac{2x}{1-x^3}, & 0 \leq x < 1 \\ \sin x, & x \geq 1 \end{cases}$	11	$y = \begin{cases} \frac{\alpha\beta}{x-\pi} \cdot e^x, & -1 \leq x \leq 3 \\ \sqrt[3]{x^3\beta \sin x}, & -3 < x \leq -2 \\ \alpha^x, & \text{в остальных случаях} \end{cases}$

5	$y = \begin{cases} x + \frac{1}{x}, x < 2 \\ x^3 \sin x, 2 \leq x < 3 \\ x + 3, x \geq 3 \end{cases}$	12	$y = \begin{cases} \sqrt[3]{1 - e^x \cdot \cos x}, -2 \leq x \leq 5 \\ x^{2.5} - \operatorname{tg}\left(\frac{1}{x}\right), 5 < x \leq 6 \\ 2.5, \text{ в остальных случаях} \end{cases}$
6	$y = \begin{cases} (x-2)^2, x < 0 < 0,5 \\ \frac{x}{1 + \cos x}, 0,5 \leq x < 1,5 \\ x^2 + 1, x \geq 1,5 \end{cases}$	13	$y = \begin{cases} x^{4/5} \cdot \cos^{3/2} x, 0.5 \leq x \leq 1 \\ \lg(\sqrt[3]{x} + e^{x-1}), x \geq 1 \\ 0, \text{ в остальных случаях} \end{cases}$
7	$y = \begin{cases} 2-x ^{1/3}, x \geq 1 \\ -x + 2e^{-2x}, 0 < x < 1 \\ \frac{1+x}{\sqrt[3]{1+x^2}}, x \leq 0 \end{cases}$	14	$y = \begin{cases} x^2 - 2 \cdot \ln \sin(x) ; x \leq 1 \\ \arctg^3 \frac{a+b}{x-a}; x = 5 \\ 5 \text{ в остальных случаях} \end{cases}$

4.4 Содержание отчета.

Цель работы, постановка задачи, краткие теоретические сведения, текст процедуры на VBA с пояснениями, анализ результатов, выводы

4.5. Контрольные вопросы

1. Назначение условного оператора.
2. Какова структура линейного условного оператора?
3. Чем отличается полный условный оператор от сокращенного?
4. В каком случае используют блочный условный оператор?
5. Какой тип имеет условие?
6. В каком случае используют конструкцию If...Then...ElseIf?

5 ЛАБОРАТОРНАЯ РАБОТА № 4

Разработка программ циклической структуры

5.1 Цель работы: Получение практических навыков разработки программ разветвляющейся структуры в EXCEL с использованием Visual Basic for Application (VBA),

5.2 Теоретические сведения

5.2.1 Понятие цикла

Циклический алгоритм – алгоритм, предусматривающий многократное повторение одних и тех же действий, называемых телом цикла. Блок операторов, находящийся между началом и концом цикла называется "тело цикла". Процесс однократного выполнения тела цикла называется итерацией цикла.

Если количество повторов заданного блока операторов известно заранее, используют цикл с заданным числом повторений **For..Next**.

Синтаксис:

For *счетчик* = *нач_значение* **To** *конеч_значение* [**Step** *шаг*]
 [*операторы*]
Exit For
 [*операторы*]
Next [*счетчик*]

Параметр *счетчик* – это числовая переменная, автоматически изменяющая свое значение на величину *шаг* после каждого повтора цикла. Цикл выполняется до тех пор, пока $\text{счетчик} \leq \text{конеч_значение}$ при положительном значении параметра *шаг* или до тех пор, пока $\text{счетчик} \geq \text{конеч_значение}$ при отрицательном значении параметра *шаг*. Если параметр **Step** (*шаг*) опущен, то по умолчанию приращение переменной-счетчика полагается равным 1. При выходе из цикла переменная $\text{Счетчик} = \text{Конец} + \text{Шаг}$. Оператор **Exit For** завершает выполнение цикла и передает управление оператору, следующему за ключевым словом **Next**.

5.2.2 Вывод данных на Рабочий лист

В модели объектов Excel имеются более 100 объектов, которые позволяют манипулировать всеми его возможностями и данными. Одним из наиболее широко используемых объектов в Excel является объект Range, представляющий собой ссылку на ячейку, строку, столбец, несколько расположенных как рядом, так и отдельно ячеек, или трехмерный диапазон.

Для вывода данных на рабочий лист используется объект Range (диапазон) или его свойство Cells (ячейки). Возможны 3 способа вывода:

- Range (<диапазон>) = <выражение>
- Cells (№ строки, № столбца) = <выражение>
- Cells (№ строки, заголовок столбца) = <выражение>

Выражение может иметь числовой или строковый тип.

Пример;

1) Для того чтобы переменной x присвоить значение из ячейки B2 записывается один из операторов: x = cells (1,2) или x= range («B2»)

2) Для того чтобы в ячейку A2 поместить число 10 надо записать: cells (2,1)=10 или range («A2»)=10

5.2.3. Пример разработки программы циклической структуры

Постановка задачи: разработать алгоритм и составить программу решения задачи вычисления значения функции y для заданных значений аргумента x: $y = \frac{ax^2}{\sqrt{x+a}}$, x изменяется от $x_n=15$ до $x_k=25$ с шагом $\Delta x=1,5$.

Таблицу значений вывести на рабочий лист.

Блок схема алгоритма приведена на рис. 5.1

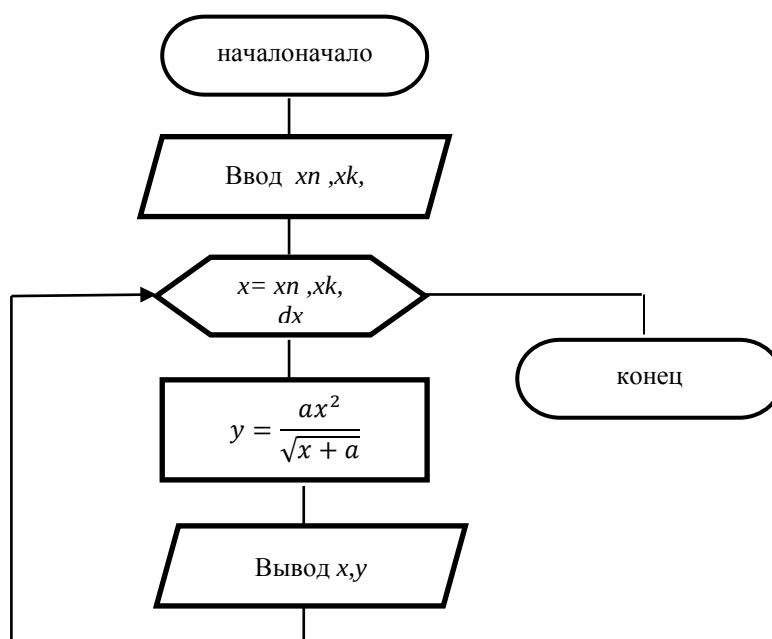


Рис. 5.1. Блок-схема алгоритма

Текст программы:

```

Sub Tabul()
    A = Val(InputBox("ÂâÄ"))
    XN = Val(InputBox("ÂâÄXN"))
    XK = Val(InputBox("ÂâÄXK"))
    DX = Val(InputBox("ÂâÄDX"))
    I = 1
    Cells(I, 1) = "X"
    Cells(I, 2) = "Y"
    For X = XN To XK Step DX
        I = I + 1
        Y = A * X ^ 2 / Sqr(X + YA)
        Cells(I, 1) = X
        Cells(I, 2) = Y
    Next X
End Sub

```

5.3 Последовательность выполнения работы

5.3.1 Разработать блок-схему решения задачи табулирования функции y в соответствии с вариантом (таблица 3.1). Вычислить и напечатать значения функции y для значений аргумента, изменяющихся от x_n до x_k с шагом Δx

5.3.2 В рабочей книге MS Excel создать процедуру на языке VBA для решения задачи, предусмотреть вывод таблицы на рабочий лист Excel

5.3.3 Выполнить программу

5.3.4 Оформить отчет

Таблица 5.1. Варианты заданий

Варианты задания	Расчетные формулы	Исходные данные				
		a	b	x_n	x_k	Δx
1	$y = \arctg bx / (1 + \sin^2 x)$	-	0,75	1,35	6,5	0,8
2	$y = \sqrt{a + bx} / \ln^2 x$	19,6	7,6	14,6	34,8	6
3	$y = a \ln^2 x / b + \sqrt{x}$	1,38	-1,26	60	100	10
4	$y = s \ln^2 x / \sqrt{x} + bx$	-	1,68	1,2	2,4	0,2
5	$y = \ln^2 (x - b) / a \sqrt{x}$	0,36	5,5	10	50	6
6	$y = e^{ax} + b / 1 + \cos^2 x$	0,9	1,85	0	1,2	0,15
7	$y = a + \sqrt[3]{x} / \sin^2 bx$	1,24	0,67	10,2	12,4	0,45
8	$y = a \sqrt{x} - bx / \ln^2 x$	2,8	0,45	40	60	4,5
9	$y = \sqrt{ax} - b / tg^2 x$	20,2	7,65	3,5	4	0,1
10	$y = e^{-x^2} / (a + bx / \ln^2 x)$	4,6	2,5	0,75	1,8	0,3

11	$y = (tg^2 ax - b) / e^{ax}$	0,55	0,78	4,2	5,8	0,25
12	$y = arctg bx / (1 + \sqrt[3]{ax})$	7,83	0,3	9	12	0,35
13	$y = \sin^2 ax / (ax + b)$	0,28	1,35	1,2	7,5	0,5
14	$y = e^{-ax} / (b + \cos^3 ax)$	0,9	0,66	2,3	8,9	1,3
15	$y = \ln^2 \sqrt{x} / (a\sqrt{x})$	0,85	-	17,2	24,6	2

5.4 Содержание отчета

Цель работы, постановка задачи, краткие теоретические сведения, текст процедуры на VBA с пояснениями, анализ результатов, выводы

5.5. Контрольные вопросы

1. Назначение оператора цикла.
2. Какова структура цикла с заданным числом повторений?
3. Как задать количество повторений цикла?
4. Как вывести значение в ячейку рабочего листа?
5. Может ли цикл с заданным числом повторений не выполниться ни разу?
6. Из каких элементов состоит заголовок цикла?

6 ЛАБОРАТОРНАЯ РАБОТА № 5. Одномерные массивы

6.1 Цель работы: получение навыков обработки массивов данных и разработки программ циклической структуры

6.2 Теоретические сведения.

Массивы – это упорядоченные конечные последовательности элементов одного и того же типа. Массив может быть одномерным (линейным) или многомерным. Двумерные массивы называют матрицами. В памяти компьютера массиву отводят блок памяти, в котором элементы массива располагаются друг за другом. Номер элемента массива называется его индексом. Индексы представляют собой выражения любого скалярного типа, кроме вещественного.

Все элементы массивы имеют одно и то же имя. Доступ к элементу массива осуществляется по его порядковому номеру в последовательности. Таким образом, чтобы получить значение, например, третьего элемента некоторого массива нужно указать имя массива, и номер (индекс) элемента. Например, A(3) – третий элемент массива A.

Пусть переменная В содержит массив пяти целых чисел: 24, 8, 6, 77, 3 (рис.5.1). Тогда В(3) – имя элемента со значением 6, т.е. В(3) = 6 (рис. 6.1)

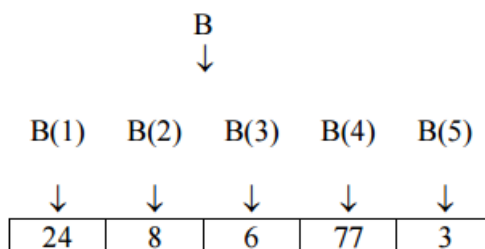


Рис. 6.1 – Одномерный массив из пяти элементов

Если выполнить присваивание, например, В(3) = В(5), то значением элемента В(3), как и В(5), будет число 3, прежнее значение 6 «сотрется». Обозначение В(і) можно считать именем і-го элемента массива В.

Массивы должны быть объявлены до их использования в программе. Для объявления массива используется оператор **Dim**. Синтаксис:

Dim | Public | Private Имя_Массива (Размерность_Массива) As Тип_Данных

где Dim | Public | Private – ключевые слова, декларирующие массив и область его видимости;

Имя_Массива – идентификатор, определяющий имя массива

Размерность_Массива – значение нижней и верхней границы значения индекса (номера) соответствующей размерности.

As – ключевое слово, предваряющее указание типа элементов массива;

Тип_Данных – базовый или созданный пользователем тип данных.

Например, описание линейного массива А, состоящего из десяти элементов – целых чисел выглядит так: Dim A(10) As Integer.

В VBA нумерация элементов массива начинается с 0, поэтому массив А состоит из 11 элементов. Для того, чтобы номер первого элемента массива был равен 1 используют описание Dim A(1 To 10) As Integer или перед описанием массива записывают директиву Option Base 1.

Иногда в процессе выполнения программы требуется изменять размер массива. В этом случае первоначально массив объявляют как динамический. Для этого при объявлении массива не надо указывать его размерность: Dim R () As Single.

Затем в программе следует вычислить необходимое количество элементов массива и сохранить его в некоторой переменной, например n, а затем изменить размер динамического массива с помощью оператора ReDim:

ReDim R (1 To 3).

Синтаксическая конструкция переопределения массива:

ReDim [Preserve] Имя_Массива (Размерность_Массива) As Тип_Данных

где ReDim – ключевое слово, указывающее, что переопределяются размеры массива;

Preserve – необязательное ключевое слово, используемое для сохранения данных в существующем массиве при изменении значения последней размерности. Если оно не используется, то данные в массиве при изменении размерности не сохраняются.

Имя_Массива – имя массива;

Размерность_Массива - значение нижней и верхней границы значения индекса (номера) соответствующей размерности. Если отсутствует инструкция Option Base, нижняя граница индекса по умолчанию равна 0.

Поскольку массив состоит из нескольких элементов, то ввод, вывод и обработка элементов массива производится в цикле. Количество элементов массива известно заранее, поэтому используют цикл For с числом шагов, равным длине (количеству элементов) массива

Ввод и вывод массива фиксированной длины можно осуществить с клавиатуры или с листа Excel (таблица 6.1).. Пусть, например, требуется ввести и вывести массив M из 10 целых чисел.

Таблица 6.1 – Ввод и вывод элементов массива

Ввод массива с клавиатуры	For i = 1 To 10 M(i) = InputBox (“Введите элемент массива”) Next i
Ввод массива с листа Excel (массив расположен в диапазоне A1:A10)	For i = 1 To 10 M(i) = Cells(i,1) ‘номер строки меняется в цикле Next i
Вывод массива в окно вывода	For i = 1 To 10 MsgBox Str(M(i)) Next i
Вывод массива на лист Excel (в диапазон B1:B10)	For i = 1 To 10 Cells(i,2) = M(i) ‘номер строки меняется в цикле Next i

6.3 Пример обработки массива

Постановка задачи: В массиве из 10 целых чисел найти максимальное значение.

Блок схема алгоритма имеет вид:

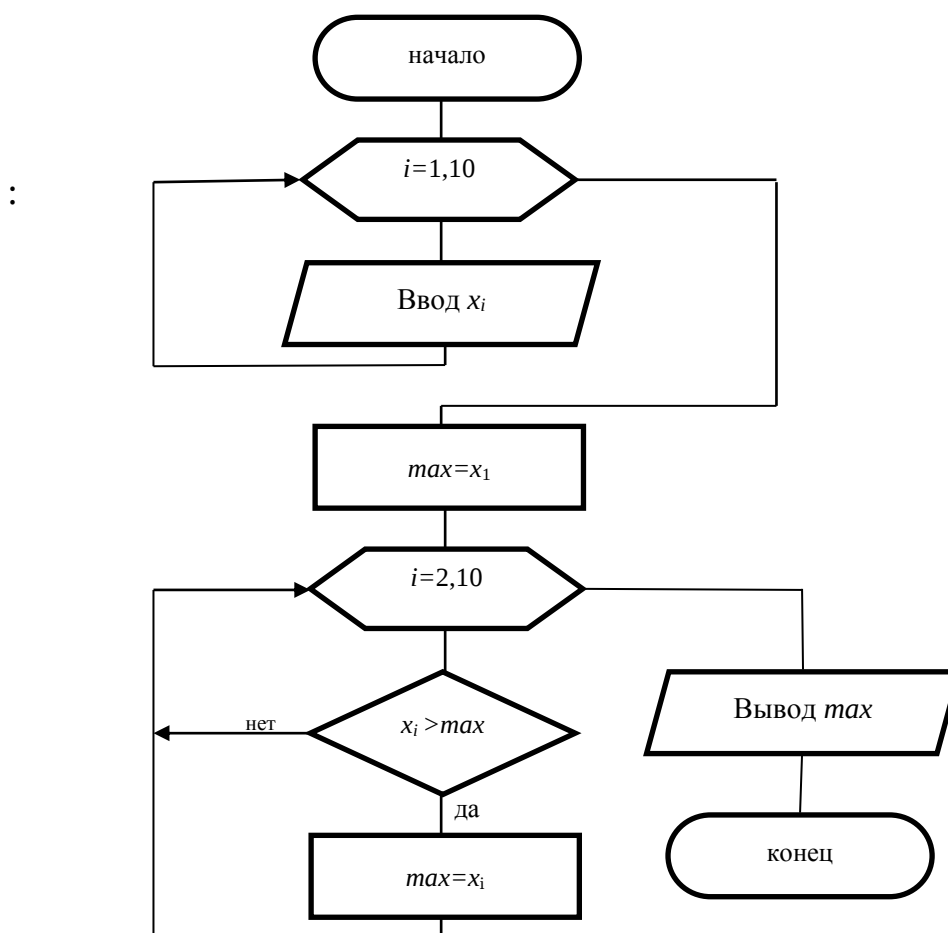


Рис. 6.2 – Схема алгоритма

Текст программы на VBA:

```

Public Sub maximum()
Dim X(1 To 10), i, max As Integer
For i = 1 To 10
X(i) = Val(InputBox("Введите элемент массива")) 'Ввод элемента
массива с клавиатуры
Cells(i, 2) = X(i) 'Вывод элемента массива в ячейку второго
столбца (B)
Next i
max = X(1)
For i = 2 To 10
If X(i) > max Then max = X(i)
Next i
  
```

MsgBox "максимальный элемент" & Str(max) 'Вывод результата в
окно
Cells(1, 3) = max 'Вывод результата в ячейку C1

6.4 Задание к работе

Таблица 5.2. – Индивидуальные задания

Вариант	Задание
1	Найти сумму элементов массива, больших заданного числа С
2	Найти произведение элементов массива, больших заданного числа С
3	Найти минимальный элемент массива и его порядковый номер
4	Найти среднее арифметическое элементов массива
5	Преобразовать массив, заменив в нем положительные элементы на 1, а отрицательные на -1
6	Определить количество положительных элементов массива
7	Найти количество элементов массива, кратных 5
8	Найти сумму положительных элементов массива
9	В массиве поменять местами первый и максимальный элемент
10	В массиве поменять местами последний и минимальный элемент
11	Найти сумму элементов массива, которые являются четными числами
12	Найти произведение элементов массива, которые являются нечетными числами
13	Найти наибольшее из положительных элементов массива
14	Вычислить произведение элементов массива, не равных нулю
15	Найти номер элемента массива с минимальным значением

6.5 Содержание отчета

Цель работы краткие теоретические сведения, задание к работе, схема алгоритма, текст процедуры на VBA с пояснениями, анализ результатов, выводы.

6.6 Контрольные вопросы

1. Что такое массив ?
2. Как осуществляется доступ к элементу массива?
3. Как объявить массив?
4. Как вывести массив на рабочий лист?
5. Каков по умолчанию номер первого элемента массива?
6. В каком случае используют динамические массивы?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гарбер Г.З. Основы программирования на Visual Basic и VBA в Excel 2007./ Г.З. Гарбер. - М.: «Солон – Пресс», 2008 г.- 191 с.
2. Гетц, К. Программирование в Microsoft Office для пользователя : учеб. пособие / К. Гетц, М. Джилберт. - К. : BHV, 2000. - 384 с. : ил.
3. Истомин, Е. П. Информатика и программирование: Pascal и VBA: учеб. для студ. вузов, обуч. по спец. "Прикладная информатика (по областям)" и др. междисциплинар. спец. / Е. П. Истомин, А. М. Власовец ; Рос. гос. гидрометеоролог. ун-т. - СПб. : Андреевский издательский дом, 2010. - 294 с. : ил.
4. Каммингс С. VBA для "чайников": простой и неумолимый способ усовершенствования бизнес - приложений, в том числе Office 2000, с помощью Visual Basic for Application : учеб. пособие / С. Каммингс. - 2-е изд. - К. ; М. ; СПб. : Диалектика, 2000. - 382 с. : ил.
5. Михеев, Р. VBA и программирование в MS OFFICE для пользователей / Р. Михеев. - Санкт-Петербург : БХВ - Петербург, 2006. - 388 с. : ил.
6. Слепцова, Л. Д. Программирование на языке VBA : самоучитель / Л. Д. Слепцова. - М. ; СПб. ; К. : Диалектика, 2004. - 384 с. : ил.