

Министерство науки и высшего образования Российской Федерации
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Г.В. САБЛИНА, Д.С. ХУДЯКОВ

ИНФОРМАТИКА

Учебное пособие
для студентов 1-го курса АВТФ направлений
27.03.04 – «Управление в технических системах»
и 09.03.01 – «Информатика и ВТ»

НОВОСИБИРСК
2020

УДК 004(075.8)
С 122

Рецензенты:

канд. техн. наук, доцент *Е.Л. Веретельникова*
канд. техн. наук, доцент *И.Л. Еланцева*

Утверждено Редакционно-издательским советом университета
в качестве учебно-методического пособия

Саблина Г.В.

С 122 Информатика: учебно-методическое пособие / Г.В. Саблина,
Д.С. Худяков. – Новосибирск: Изд-во НГТУ, 2020. – 82 с.

ISBN 978-5-7782-3171-9

В работе изложен теоретический материал, который может быть полезен всем, кто изучает основы программирования на объектно-ориентированном языке Visual Basic for Application (VBA) под управлением электронных таблиц Excel. Приведено описание восьми практических заданий, в результате выполнения которых можно научиться программировать различные виды алгоритмов, решать системы алгебраических уравнений, выполнять матричные операции, обрабатывать текстовую информацию и работать с файлами.

Пособие будет полезно для студентов, изучающих информатику и интересующихся вопросами программирования на VBA.

УДК 004(075.8)

ISBN 978-5-7782-3171-9

© Саблина Г.В., Худяков Д.С., 2020
© Новосибирский государственный
технический университет, 2020

ОГЛАВЛЕНИЕ

Введение	5
1. Основы работы с электронными таблицами MS Excel	6
1.1. Основные понятия.....	6
1.2. Главное окно MS Excel	8
1.3. Ввод, редактирование, форматирование данных в ячейках.	11
1.4. Ввод, редактирование и форматирование текста	14
1.5. Автоматический ввод значений	17
1.6. Перемещение и копирование содержимого ячеек и диапазонов	18
2. Язык VBA под управлением электронных таблиц Excel	19
2.1. Написание первой программы	20
2.2. Объявление переменных	21
2.3. Преобразование типов	23
2.4. Область действия переменных.....	24
2.5. Диалоговые окна ввода/вывода.....	26
2.6. Операции	27
2.7. Массивы	29
2.8. Управляющие конструкции.....	30
2.9. Циклы	33
2.10. Символьные данные. Операции и функции для работы с сим- вольными данными	35
2.11. Работа с текстовыми файлами.....	37
3. Практические задания	39
Практическое задание № 1. Ввод данных и корректировка информации в электронной таблице.....	39
Практическое задание № 2. Линейные, разветвляющиеся, циклические и итерационные алгоритмы.....	46

Практическое задание № 3. Функции комплексного переменного, матрицы, СЛАУ.....	57
Практическое задание № 4. Линейные, разветвляющиеся, циклические алгоритмы в VBA	62
Практическое задание № 5. Функции комплексного переменного и матрицы в VBA	63
Практическое задание № 6. Строковые данные. Подпрограммы-функции.....	64
Практическое задание № 7. Строковые массивы. Подпрограммы-процедуры	67
Практическое задание № 8. Ввод/вывод в файлы. Элементы управления	69
4. Расчетно-графическое задание.....	71
Библиографический список	75

ВВЕДЕНИЕ

Ценность любой информации в значительной мере определяется качеством ее организации и, более того, существенная доля трудовых затрат на обработку информации связана с ее логическим структурированием. Отображать и обрабатывать данные наиболее целесообразно в виде таблиц.

Компьютер позволяет представить таблицы в электронном виде, что дает возможность не только отображать данные, но и автоматизировать их обработку. Класс программ, используемых для этих целей, называется **электронными таблицами**. Применение электронных таблиц упрощает работу с данными и позволяет получать результаты без проведения расчетов вручную или специального программирования.

Наиболее широко электронные таблицы применяются в экономических и бухгалтерских расчетах, а также при решении научно-технических задач. Например:

- для подготовки табличных документов;
- проведения однотипных расчетов над большим объемом данных с использованием мощного аппарата функций и формул;
- исследования влияния различных факторов на результаты;
- моделирования ситуации «что если»;
- табулирования функций, решения задач путем подбора значений параметров;
- статистического анализа данных;
- решения задач оптимизации (поиска оптимальных параметров);
- получения выборки данных, удовлетворяющих определенным критериям;
- прогноза;
- построения графиков и диаграмм по имеющимся данным и т. д.

Одно из наиболее широко используемых средств работы с документами, имеющими табличную структуру – программа Microsoft Excel (MS Excel). В данном пособии описывается работа с версией MS Office 2007: принципиальных отличий более поздних версий в плане рассматриваемого функционала нет, основные изменения касаются интерфейса.

1. ОСНОВЫ РАБОТЫ С ЭЛЕКТРОННЫМИ ТАБЛИЦАМИ MS EXCEL

1.1. ОСНОВНЫЕ ПОНЯТИЯ

Документ MS Excel называется **рабочей книгой**. Каждая книга имеет имя и хранится на диске в отдельном файле. По умолчанию новой рабочей книге присваивается имя «Книга1», которое при сохранении целесообразно изменить. При записи на диск к имени книги добавляется расширение **.xls**. В Excel допускается работа с несколькими рабочими книгами одновременно, при этом каждая открывается в собственном окне. Активной (текущей) может быть только одна из них. Для активизации рабочей книги можно, например, использовать кнопки на **Панели задач**.

Для того чтобы все окна рабочей книги были видны на экране:

1) выберите команду **Окно/Расположить**. На экране появится диалоговое окно **Расположение окон**;

2) в группе **Расположить** отметьте нужный переключатель: **рядом, сверху, вниз, слева направо, каскадом**.

Рабочая книга представляет собой набор **рабочих листов**, каждый из которых имеет табличную структуру и может содержать одну или несколько таблиц, диаграммы, а также внедренные или связанные объекты других приложений (например, MS Word). Каждый лист имеет **имя**, которое отображается на **ярлычке листа**. По умолчанию листы имеют имена «Лист1», «Лист2» и т. д. Эти имена командой **Формат/Лист/Переименовать** можно изменить или, дважды щелкнув по ярлычку, ввести новое имя.

Рабочий лист MS Excel разделен сеткой на **строки** и **столбцы** (рис. 1), на пересечении которых находятся **ячейки**. Обычно столбцы обозначаются прописными латинскими буквами (A, B, C, ..., Z), а далее двухбуквенными сочетаниями (AA, AB, ..., BA, BB, ...). Всего рабо-

чий лист может содержать до 256 столбцов (с именами от А до IY). Строки последовательно нумеруются от 1 до 65536.

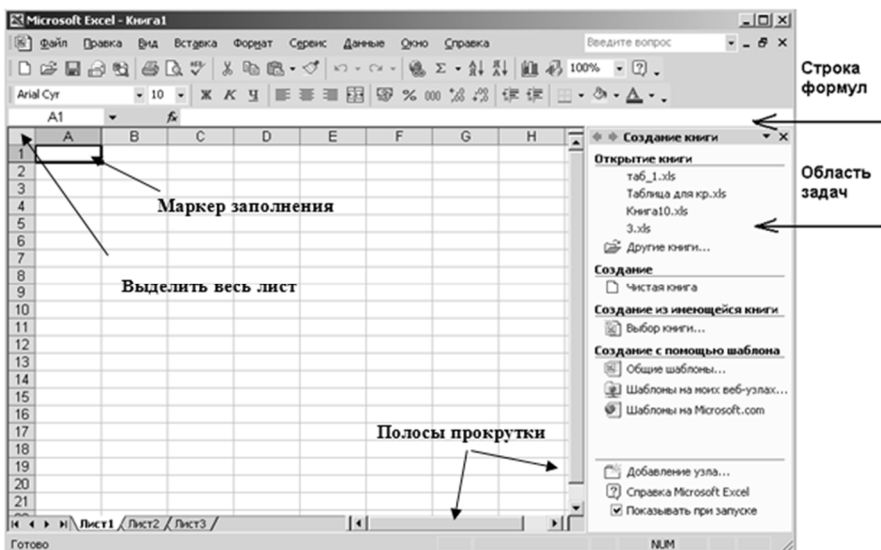


Рис. 1. Главное окно MS Excel

Ячейка является минимальным элементом для хранения данных. Имя ячейки состоит из имени столбца и номера строки, на пересечении которых она находится. Таким образом, самая первая ячейка имеет адрес A1. Затем по строке B1, C1, D1 и т. д. В следующей (второй) строке – A2, B2, C2, D2 и т. д.

Если надо задать адрес ячейки, которая находится на другом рабочем листе или даже в другой рабочей книге, перед именем ячейки задают имя рабочего листа, а во втором случае и имя рабочей книги. Например, **Лист1!A1** или **[Книга1.xls].Лист1!A1**.

Кроме понятия **ячейки**, используется понятие **диапазона**. Чаще всего в качестве диапазона понимается прямоугольная область, состоящая из нескольких (или одного) столбцов и нескольких (или одной) строк. В качестве адреса диапазона указываются адреса левой верхней и правой нижней ячеек диапазона, разделенные знаком : (двоеточие). Например, диапазон **A1 : C8** содержит 24 ячейки (3 ячейки в строках и 8 ячеек в столбцах).

1.2. ГЛАВНОЕ ОКНО MS EXCEL

Главное окно, появляющееся непосредственно после запуска программы (см. рис. 1), имеет следующие интерфейсные элементы:

- стандартную для Windows строку заголовка;
- строку меню;
- панели инструментов;
- строку ввода данных (поле имени и строка формул);
- окно книги, которое занимает большую часть экрана;
- строку состояния;
- полосы прокрутки;
- область задач.

Имеются достаточно широкие возможности настройки интерфейса.

СТРОКА МЕНЮ

Организация меню в MS Excel стандартна для Windows. Пункты меню, как правило, иерархические, раскрывающиеся списками команд, которые можно выбрать. Выбор команды приводит либо к непосредственному выполнению некоторого действия, либо к раскрытию еще одного меню, или к раскрытию диалогового окна или **Формы**.

Выбирать команду можно с помощью мыши, но иногда удобнее пользоваться клавиатурой. Например, чтобы активизировать строку меню, достаточно нажать клавиши **ALT** или **/**, а затем пользоваться клавишами со стрелками для перемещения по пунктам меню. Выбор производится выделением нужного пункта и нажатием клавиши **Enter**.

При работе с MS Excel строка меню может изменяться (могут появляться другие команды), например, если в рабочей области находится диаграмма в режиме редактирования.

MS Excel имеет много контекстных меню, каждое из которых предлагает наиболее полезные команды для текущей позиции указателя курсора.

ПАНЕЛИ ИНСТРУМЕНТОВ

Панели инструментов – это наборы кнопок для быстрого вызова наиболее часто используемых функций. Панели инструментов **Стандартная** и **Форматирование** встроены в MS Excel и появляются после установки пакета в одной строке (для удобства их можно расположить друг под другом).

Вводить данные или формулы можно либо непосредственно в **активную ячейку**, либо в **строку формул**, предварительно выделив нужную ячейку. Редактирование уже введенной формулы производится в **строке формул** после активизации ячейки с формулой и установки курсора в нужную позицию формулы.

Слева в строке ввода находится **окно имени**, где выводится имя активной ячейки или первой ячейки диапазона.

Кнопки панели инструментов **Стандартная** предоставляют быстрый доступ к наиболее распространенным операциям, таким как создать, открыть или сохранить книгу, отменить или повторить последнюю операцию и др.

Кнопки панели инструментов **Форматирование** выполняют функции обработки данных в ячейках: устанавливают шрифт, размер, цвет ячеек, стиль текста и рамок и прочее.

Можно поместить на экран и другие панели, а также изменить состав кнопок на панели.

СТРОКА ФОРМУЛ (ВВОДА ДАННЫХ)

Под панелями инструментов находится строка формул и редактирования значений или формул в ячейках или диаграммах. MS Excel выводит в этой строке значение или формулу активной ячейки.

При активизации и наборе формулы левее строки формул появляются три кнопки:



Ввод. Фиксирует введенную информацию и переносит ее на лист. Использование этой кнопки оставляет ячейку активной, а клавиша Enter активизирует ячейку ниже;



Отмена. Отменяет ввод формулы (аналогично клавише ESC);



Вызывает Мастер функций – специальную подпрограмму для более удобного ввода различных функций: математических, тригонометрических, статистических и т. д.

Окно рабочей книги составляет основную часть рабочей области (рис. 2).

Каждая рабочая книга имеет собственное имя и хранится в отдельном файле на диске. По умолчанию новым рабочим книгам Excel присваивает имена **Книга1**, **Книга2** и т. д. (при записи на диск к этому имени добавляется расширение **.xls**). В Excel допускается работа с несколькими рабочими книгами одновременно, при этом каждая открывается в собственном окне. Активной может быть только одна книга.

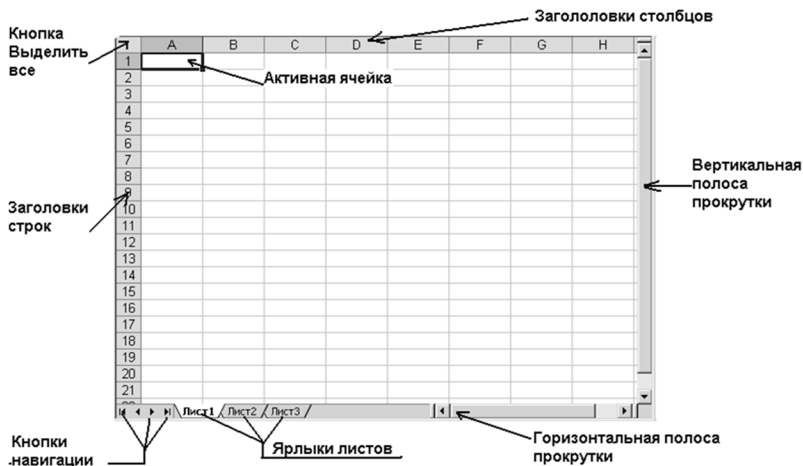


Рис. 2. Окно рабочей книги MS Excel

Книга в MS Excel – файл (с расширением **.xls**). Книга может состоять из нескольких листов (по умолчанию 3 листа), поэтому в одном файле можно сохранить разнообразные данные и установить между ними связи.

Лист – это место для хранения и обработки данных. Листы могут быть следующих типов:

- рабочие листы;
- листы диаграмм;
- листы диалога;
- модули Visual Basic;
- листы макросов.

Можно вводить и изменять данные одновременно на нескольких листах, а также выполнять вычисления на основе данных из нескольких листов.

Имена листов отображаются на ярлычках в нижней части окна книги. Ярлычок активного окна выделен. Кнопки прокрутки ярлычков позволяют просмотреть содержание книги.

Над рабочими листами можно выполнять следующие операции: выбор, удаление, копирование, переименование, вставка (добавление), скрытие и отображение, масштабирование. Большую часть этих операций можно выполнить с помощью контекстного меню ярлычка листа (щелкнуть правой кнопкой мыши на ярлычке).

Активизировать (выделить) любую ячейку можно щелчком по ячейке левой кнопкой мыши или перемещая рамку выделения с помощью клавиш →, ↓, ←, ↑. Для изменения видимой части листа можно пользоваться полосами прокрутки.

ОБЛАСТЬ ЗАДАЧ


Окно документа делится на две части, в одной из которых отображается документ, а другая называется **областью задач** и содержит набор элементов управления, обеспечивающих быстрый доступ к тем средствам MS Excel, которые применимы к документу на данном этапе работы с ним. Область задач можно всегда открыть (команда **Вид/Область задач**) или закрыть. В некоторых случаях область задач отображается автоматически (в частности, при создании новой **Книги**), поскольку некоторые средства (например, **мастера**) доступны только через область задач.

СТРОКА СОСТОЯНИЯ

Отражает сведения о состоянии рабочей области. В левой части строки состояния кратко описывается выделенная или выполняемая команда. Правая часть строки состояния содержит калькулятор и индикаторы различных режимов клавиатуры.

ПОЛУЧЕНИЕ СПРАВОЧНОЙ ИНФОРМАЦИИ

Подобно большинству приложений MS Excel имеет контекстную справочную систему. Для получения справочной информации можно воспользоваться одним из следующих способов.

1. Выбрать пункт основного меню **Справка**.
2. Использовать кнопку  на панели инструментов **Стандартная** (или в окне диалога).
3. Нажать клавишу **F1** в любой момент работы с MS Excel. В ответ будет выдана справочная информация о текущей операции.

1.3. ВВОД, РЕДАКТИРОВАНИЕ, ФОРМАТИРОВАНИЕ ДАННЫХ В ЯЧЕЙКАХ.

ВЫДЕЛЕНИЕ НЕСКОЛЬКИХ ЯЧЕЕК

В MS Excel, по крайней мере, одна ячейка выделена всегда. Группа ячеек называется **диапазоном**. Чтобы произвести с ячейкой (или ячейками) какую-либо операцию, ее необходимо выделить. Только выде-

лив ячейку, можно ее удалить, переместить, скопировать, очистить ее содержимое или отформатировать.

Многие операции в Excel (например, копирование, перемещение, форматирование) могут выполняться не только над отдельными ячейками, но и над диапазоном. Чтобы указать диапазон, во многих случаях его нужно выделить. Выделить диапазон можно с помощью мыши, а также с помощью клавиатуры. Ниже приведены основные способы выделения диапазона с помощью мыши:

- для выделения всех ячеек столбца или строки щелкните на соответствующем заголовке (столбца или строки);
- для выделения прямоугольной области ячеек переместите указатель мыши от первой к последней ячейке при нажатой левой кнопке мыши;
- для выделения смежных строк (столбцов) щелкните по заголовку первой строки (столбца), нажмите клавишу **Shift** и щелкните по заголовку последней строки (столбца) диапазона;
- для выделения несмежных строк (столбцов) щелкните по заголовку первой строки (столбца), нажмите клавишу **Ctrl** и, удерживая ее, выделите остальные ячейки или диапазоны;
- для выделения всех ячеек листа щелкните по ячейке, расположенной на пересечении заголовков строк и столбцов. Для выделения нужной ячейки можно использовать клавиши перемещения («быстрые клавиши», табл. 1).

Т а б л и ц а 1

Клавиши	Активной становится ячейка
←, →, ↑, ↓	Соответственно левее, правее, выше, ниже
Tab	Правее текущей
Shift+Tab	Левее текущей
Home	Первая ячейка текущей строки
End	Последняя используемая ячейка текущей строки
Ctrl+Home	Ячейка A1
Ctrl+End	Последняя используемая ячейка
PgUp/PgDn	Переход вверх/вниз на одно окно

РЕГУЛИРОВАНИЕ ШИРИНЫ СТОЛБЦОВ И ВЫСОТЫ СТРОК

На новом листе **рабочей книги** все столбцы имеют стандартную ширину **8.43** символа. Строки имеют стандартную высоту **12.75** пунктов. Высота строки автоматически изменяется так, чтобы разместились

данные, имеющие наибольший размер шрифта. Ширина столбца не подстраивается под данные и ее надо регулировать вручную или использовать команду **Формат/Столбец/Автоподбор ширины**. Регулировать ширину столбца можно следующими способами.

1. Выделите столбец. Подведите указатель мыши к правой границе заголовка столбца. Когда указатель изменит вид на \leftrightarrow , нажмите левую кнопку мыши и протяните границу до установления нужной ширины.

2. Выделите столбец, выберите команду **Формат/Столбец/Ширина** и задайте нужную ширину столбца.

3. Сделайте двойной щелчок на правой границе выделенного столбца. Ширина оптимально настроится под размер данных.

4. Выделите столбец и выполните команду **Формат/Столбец/Автоподбор ширины**.

Можно то же самое сделать для нескольких столбцов, если их предварительно выделить.

Изменить ширину строки можно следующими способами:

- 1) протащите с помощью мыши нижнюю границу строки до нужной высоты;

- 2) выделите строку, выберите команду **Формат/Строка/Высота строки** и задайте нужную высоту строки (строк).

ТИПЫ ДАННЫХ

В ячейки рабочего листа могут быть введены три вида данных: числа, формулы и текст.

Числа, включая даты, представляют собой значения, которые можно использовать в вычислениях.

Формулы – это инструкции к вычислениям. Формула может содержать числовые константы, ссылки на ячейки (имена ячеек), арифметические и логические операции, функции MS Excel, скобки для изменения последовательности выполнения операций. Формула всегда начинается со знака =.

Текст – это информация, которая не является ни числом, ни формулой.

Тип данных, вводимых в ячейку, определяется автоматически. Если данные могут быть числом, то MS Excel их так и интерпретирует. В противном случае данные рассматриваются как текст.

1.4. ВВОД, РЕДАКТИРОВАНИЕ И ФОРМАТИРОВАНИЕ ТЕКСТА

Для ввода информации в ячейку рабочего листа сначала необходимо сделать эту ячейку активной. Данные можно вводить с клавиатуры непосредственно в ячейку (при этом текстовый курсор находится внутри ячейки – такой режим называется *редактированием в ячейке*) либо в строку формул, предварительно щелкнув мышью в строке формул (при этом текстовый курсор находится в строке формул над рабочим листом – такой режим называется *редактированием в строке формул*).

При вводе символов необходимо помнить:

- чтобы сохранить данные в ячейке, следует завершить ввод нажатием клавиши **Enter** на клавиатуре или щелчком на кнопке **Enter** в строке формул;
- если начать ввод данных с алфавитно-цифровых символов, то данные из текущей ячейки заменяются на вводимые данные;
- если щелкнуть в строке формул или дважды в текущей ячейке, то старое содержание не удаляется и появляется возможность его редактировать;
- чтобы отменить изменения в ячейке, можно нажать клавишу **Esc** или щелкнуть на кнопке **Отмена** слева от строки формул;
- чтобы отобразить текст ячейки в нескольких строках, можно при переходе на новую строку пользоваться комбинацией клавиш **Alt+Enter**.

Когда текст не помещается в ячейке полностью, он отображается поверх следующей ячейки, если она пустая. Если в соседней ячейке уже находятся данные (или вводятся), отображение текста обрезается на границе между ячейками. Чтобы увидеть всю информацию, находящуюся в ячейке, необходимо увеличить ширину соответствующего столбца или, выделив ячейку, посмотреть на строку формул.

ФОРМАТИРОВАНИЕ ТЕКСТА

Внешний вид введенного текста (в ячейках, диапазонах) можно изменить. Для форматирования данных надо выделить диапазон (ячейку) и использовать команду **Форма/Ячейки** основного меню или команду **Формат ячеек** контекстного меню. Диалоговое окно **Формат ячеек** содержит несколько вкладок, предоставляющих пользователю различные возможности для форматирования.

- На вкладке **Число** выбирается вид и форма представления числовых данных (подробнее смотрите ниже);
 - На вкладке **Выравнивание** задается способ выравнивания содержимого ячеек. Можно задать выравнивание как по горизонтали, так и по вертикали. Здесь же можно задать один из способов вертикального расположения текста, а также можно переносить текст строки внутри ячейки по словам (установить флажок).
 - Кроме известных видов выравнивания существует новый тип – с объединением ячеек. При этом типе выравнивания происходит объединение выделенных ячеек в одну и центрирование в ней содержимого левой верхней ячейки диапазона. Такой тип выравнивания используется для заголовков таблиц, расположенных на рабочем листе.
 - На вкладке **Шрифт** задаются параметры форматирования символов. По умолчанию Excel использует шрифт **Arial** размером **10** пт.
 - На вкладке **Граница** выбираются линии границ для выделения определенных ячеек и диапазонов рабочего листа.
 - На вкладке **Вид** можно выбрать средства для оформления определенных частей листа: заголовков, итоговых сумм и т. п.
 - На вкладке **Защита** можно включить блокировку, которая предотвращает случайное изменение содержимого ячейки.
- Для применения цвета заливки ячеек, цвета шрифта и установки границ выделенных ячеек можно использовать кнопки **Цвет заливки**, **Цвет шрифта** и **Границы** панели инструментов **Форматирование**.

ВВОД, РЕДАКТИРОВАНИЕ И ФОРМАТИРОВАНИЕ ЧИСЕЛ

При вводе чисел можно использовать следующие символы: цифры от **0** до **9**, знаки **+** и **–**, а также символ **E (e)**. Символ **E** используется для ввода (отображения) в экспоненциальном формате. Например, число **0,00000005** ($5 \cdot 10^{-8}$) в таком формате будет выглядеть **5E-08**.

Если вводимое значение воспринимается как число, то содержимое ячейки выравнивается по правому краю (прижимается к правой границе ячейки). Символьные (текстовые) данные выравниваются по левому краю.

Числовые значения, которые вводятся в ячейки таблицы, обычно никак не отформатированы. Удобнее числа отформатировать так, чтобы их было легче читать. Некоторые операции форматирования можно выполнить с помощью кнопок панели инструментов **Форматирование** (например, **денежный формат**, **процентный** и т. д.). Другие форматы можно выбрать командой **Формат/Ячейки**. На вкладке **Число** имеется 12 числовых форматов, используемых в MS Excel. У категории

Числовой (формат) имеется три опции: **Число десятичных знаков**, **Разделитель групп разрядов**, **Отрицательные числа**.

При вводе даты можно использовать один из форматов, приведенных в списке, открывающемся по команде **Формат/Ячейки**, на вкладке **Число/Даты**. Примеры дат: **14.03.2008**, **14 марта 2008 г.**, **14.3**, **14.3.08**, **14.03.08** и. т. д.

ВВОД ПРОСТЕЙШИХ ФОРМУЛ

Формула рабочего листа является тем инструментом MS Excel, что делает электронные таблицы весьма полезными. Ввод формулы в ячейку начинается со знака **=**. В формулах используются:

- арифметические операции (+, -, *, /, ^);
- операции сравнения (<, <=, =, >=, >);
- операции конкатенации строк (&);
- числа;
- строковые константы;
- встроенные и пользовательские функции;
- ссылки на ячейки;
- скобки (круглые).

Формулы записываются в одну строку, без надстрочных и подстрочных символов. Порядок выполнения операций в формуле общепринятый. Для изменения порядка выполнения операций используются круглые скобки.

После ввода формулы в ячейке получается результат вычисления.

Текст в формулу вводится при помощи кавычек. Допустим, в ячейке **C1** записано: **= «Итого:» & B1**. Тогда, если в **B1** записано **1000**, в ячейке **C1** получится – **Итого: 1000**.

В табл. 2 приведены примеры записи формул в MS Excel.

Т а б л и ц а 2

Запись в MS Excel	Объяснение
=A12+3	К содержимому ячейки A12 добавлено 3
=(F23-E40)*0.05	Из содержимого ячейки F23 вычитается содержимое ячейки E40 , разность умножается на 0,05 .
=МАКС(A12:A23)-МИН(G1:G10)	Из максимального содержимого диапазона ячеек A12:A23 вычитается минимальное содержимое диапазона G1:G10
=(A1^3+B2/2)*15	Содержимое ячейки A1 возводится в куб и складывается с половиной содержимого ячейки B2 , сумма умножается на 15

С формулами, введенными в ячейки, можно работать так же, как и с другими данными. Их можно копировать, удалять, перемещать, редактировать.

1.5. АВТОМАТИЧЕСКИЙ ВВОД ЗНАЧЕНИЙ

В MS Excel имеются средства автоматизации ввода, позволяющие упростить процесс ввода данных. К этим средствам относятся **автозаполнение, автоввод и автозамена**.

Автозаполнение позволяет выполнить много полезных операций: вводить повторяющиеся данные, продолжать начатые ряды, расширять списки. Эта функция работает с числами, днями недели, месяцами, датами. В памяти MS Excel хранится несколько списков, допустимых для автозаполнения. Просматривать имеющиеся списки можно командой **Сервис/Параметры/Списки**. Имеются следующие режимы автозаполнения.

- Для получения списков месяцев (дней недели, дат) достаточно ввести в первую ячейку диапазона название месяца (дня, даты) в полном или сокращенном виде и перетащить **маркер заполнения** до конца диапазона.

Маркер заполнения – это небольшой черный квадратик в нижнем правом углу выделенной ячейки. При наведении указателя мыши на маркер заполнения указатель меняет свой вид на черный крестик.

- Чтобы создать простую последовательность чисел (где каждое следующее число на единицу больше предыдущего), достаточно ввести первое число этой последовательности в первую ячейку диапазона. Далее, захватив маркер заполнения и удерживая клавишу **Ctrl**, протащить маркер до конца диапазона. Если не зажимать клавишу **Ctrl**, введенное значение будет просто продублировано.

- Для получения последовательности, где каждое последующее число отличается от предыдущего на единицу и больше, необходимо ввести два первых числа последовательности, выделить две ячейки введенными числами и протащить маркер заполнения до конца диапазона.

Автоввод – эффективное средство для ввода повторяющихся данных. Содержимое заполненной ячейки запоминается MS Excel и предлагается для заполнения другой ячейки в этом же столбце. Например, если введено слово «**Иванов**» в ячейку **A1**, а ячейке **A2** напечатать

«Ив», то программа автоматически предложит ввести окончание. Если нужно ввести слово, отличающееся от предлагаемого (например, «Ив-кина»), надо просто продолжать набор.

Автозамена – работает точно так же, как аналогичное средство текстового процессора MS Word.

1.6. ПЕРЕМЕЩЕНИЕ И КОПИРОВАНИЕ СОДЕРЖИМОГО ЯЧЕЕК И ДИАПАЗОНОВ

Содержимое выделенной ячейки или диапазона ячеек можно:

- удалять (нажав клавишу **Delete** или комбинацией клавиш **Ctrl+X**, если надо удалить с записью в буфер обмена). Тот же результат можно получить командой **Вырезать**, например, из контекстного меню;
- копировать в буфер обмена (нажав комбинацию клавиш **Ctrl+C** или командой **Копировать**);
- вставлять данные из буфера обмена (комбинацией клавиш **Ctrl+V** или командой **Вставить**).

Кроме того, MS Excel позволяет копировать содержимое ячейки или диапазона путем буксировки. Для этого надо установить указатель мыши у одной из границ выделенного диапазона так, чтобы он превратился в крестик со стрелками. Причем если буксировать выделенный диапазон с нажатой клавишей **Ctrl**, то диапазон копируется на новое место, если производить эту операцию без клавиши **Ctrl**, то диапазон перемещается на новое место.

Те же результаты можно получить с помощью *специального перетаскивания* выделенного диапазона.

Работе с электронными таблицами Excel посвящены первые четыре лабораторные работы, описание которых приведено в настоящем пособии.

2. ЯЗЫК VBA ПОД УПРАВЛЕНИЕМ ЭЛЕКТРОННЫХ ТАБЛИЦ EXCEL

Начиная с версии 5.0, в программу MS Excel включен специальный язык программирования, который получил название **Visual Basic for Applications (VBA)**. Введение достаточно мощного языка программирования в MS Excel делает эту программную платформу весьма привлекательной для профессионалов, которые занимаются разработкой специализированных прикладных систем.

VBA позволяет работать с Microsoft Office, как с некоторым конструктором: в распоряжении разработчика VBA-приложения имеется большое количество объектов и коллекций, а также возможность настроек, позволяющая до такой степени программно настроить любое приложение, что пользователь такого приложения может и не понять, с каким приложением происходит «общение».

Важнейшее достоинство VBA – возможность объединять любые приложения Microsoft Office для решения практически любых задач по обработке информации. В этом смысле Microsoft Office можно считать системой программирования, подобной C++, Delphi, но с более мощными и разнообразными функциями, поскольку здесь имеется неизмеримо большее количество управляемых системой объектов и готовых решений для конечных пользователей.

Привлекательная особенность VBA в том, что он очень удобен для первого знакомства с программированием в среде Windows. Этому способствует широкое распространение приложений Microsoft Office, бесконечное разнообразие возможных практических задач, интуитивно понятная интегрированная среда редактора Visual Basic, возможность обучения программированию посредством анализа кода, записанного при помощи макрорекордера, наличие огромного количества объектов, которыми можно управлять из VB-кода. Даже неподготовленный пользователь может, пользуясь простыми формулами, выполнять бухгалтерские, статистические операции. Используя особенности продук-

тов MS Office, при помощи VBA можно автоматизировать решение практически любой задачи, связанной с этим пакетом программ. Вот небольшой перечень таких задач:

- оформление результатов наблюдений в таблицах и обработка этих результатов;
- формирование графиков; оформление бухгалтерской информации;
- запись различных отчетов, заполняемых в разных местах;
- выделение нужных информационных фрагментов.

VBA позволяет максимально использовать готовые интерфейсные возможности Windows с максимально доступным интерпретатором. Идеология этого языка заключается в том, что основой при реализации процедур автоматизации является не программа, а документ, в котором она хранится, хотя программы могут храниться и отдельно от документа. Язык VBA является стандартом де-факто и используется не только в программных продуктах компании Microsoft, но и в свободном программном обеспечении. Например, разработка поддержки VBA в пакете офисных программ OpenOffice (LibreOffice) ведется с 2006 года.

2.1. НАПИСАНИЕ ПЕРВОЙ ПРОГРАММЫ

Язык VBA позволяет самообучаться при включении автоматической записи макроса **Разработчик/Макросы**.

Для программирования макросов переходят в окно редактора VBA. Проще всего это сделать, нажав сочетание клавиш **Alt-F11**. Подавляющее большинство процедур и функций должны быть размещены в *Модулях*, *Классах*, *Формах*. Подпрограмма задается ключевым словом *Sub* (от слова *Subroutine* – Подпрограмма). Процедура имеет следующий синтаксис:

```
Sub имя(параметры, ...)
```

```
.....
```

```
End sub
```

Функция имеет следующий синтаксис:

```
Function имя(параметры, ...) as тип
```

```
.....
```

```
имя = out
```

```
End function
```

Для того чтобы функция возвращала искомое значение, в теле функции необходимо записать оператор присваивания переменной,

совпадающей с именем функции. Вызов процедуры оформляется следующим образом:

Call имя_процедуры

Пример вызова функции с передачей параметров приводится ниже:

c=имя_функции (парам_1, парам_2)

Если процедура или функция находится в другом модуле, то для их вызова необходимо сначала записать имя модуля, а затем через точку выбрать имя функции или процедуры:

Call Module2.пример1

Функцию также можно вызвать, поместив ее в ячейку рабочего листа и записав:

= имя_функции (A1;B5)

Для данного примера в ячейке, в которой размещена приведенная выше запись, появится результат, возвращаемый функцией, а параметры, передаваемые в функцию, будут взяты из ячеек **A1** и **B5**.

Процедура от функции отличается тем, что функция возвращает значение, а процедура не возвращает значение, но может изменять значения переменных, переданных ей в качестве параметров.

Комментарии выполняются при помощи одиночного верхнего апострофа. Модули, классы, формы хранятся в документе *xls* или *xla* (файле надстройки).

2.2. ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ

Переменная – это имя, которое программист дает области компьютерной памяти, используемой для хранения данных определенного размера (типа).

Самым простым способом создания переменной считается использование ее в операторе VBA. При этом VBA создает переменную и резервирует память для переменной. Сохранение значения данных в переменной называется присваиванием переменной. Делается это с помощью оператора присваивания (=).

Например, *Sum=120*

Этот оператор сохраняет числовое значение 120 в ячейке памяти, заданной переменной *Sum*. Такое объявление переменной называется «*неявным объявлением переменной*». Все переменные, которые VBA создает подобным образом, имеют тип данных *Variant*.

Для явного объявления переменной используется оператор *Dim* (*Dimension*). Ранее в языке *Q-BASIC* этот оператор использовался для объявления размерности массива, отсюда и название *Dimension* (размерность).

Синтаксис данного оператора: *Dim name1 [, name2...]*

Все переменные, созданные подобным образом, являются также переменными типа *Variant*. Кроме типа *Variant* существуют типизированные переменные (переменные, для которых объявляют тип). Они ускоряют выполнение кода программы. Использование типизированных переменных может значительно сократить объем памяти, необходимой для выполнения программы. Программный код с типизированными переменными легче читается и понимается. В программах, использующих типизированные переменные, легче обнаружить некоторые ошибки программистов.

Существует функция *VarType(Имя)*, которая возвращает число, соответствующее типу переменной. В табл. 3 приведены используемые типы переменных, размер выделяемой под них памяти и числа, возвращаемые функцией *VarType*, соответствующие каждому типу переменной.

Т а б л и ц а 3

Код VarType	Тип переменной	Описание типа
0	<i>пустая</i>	Пустая инициализированная переменная
1	<i>NULL</i>	Ошибочное значение
2	<i>Integer</i>	Целое число, 2 байта
3	<i>Long</i>	Целое число, 4 байта
4	<i>Single</i>	Число с плавающей точкой, 4 байта
5	<i>Double</i>	Число с плавающей точкой, 8 байт
6	<i>Currency</i>	Число с фиксированной точкой, 8 байт, 4 знака после точки
7	<i>Date</i>	Дата/время, 8 байт, дробное число – целая часть количество дней с 31.12.1899, дробная часть – часы, минуты, секунды * 24
8	<i>String</i>	Строка длиной от 0 до 65 535
9	<i>Object</i>	Объект
11	<i>Boolean</i>	Булев тип 1 или 0
17	<i>Byte</i>	Целое число, 1 байт

Ниже приведен пример использования функции *VarType()* :

Sub пример1()

```
temp = "Hello"
MsgBox VarType(temp)
temp = 4
MsgBox VarType(temp)
End sub
```

Когда VBA создает новую переменную, эта переменная инициализируется (переменной присваивается некоторое значение «по умолчанию»):

строки инициализируются пустыми строками;
числа – значением 0;
переменные типа Boolean – False;
даты – 30 декабря 1899

Создаваемые строковые переменные «по умолчанию» являются строками переменной длины, т. е. изменяют свою длину в зависимости от длины строки, сохраняемой переменной. Но иногда может понадобиться использовать строку фиксированной длины. Такие строки всегда имеют одну и ту же длину. Они полезны, если необходимо чтобы текст, сохраненный в строковой переменной, всегда содержал одно и то же число символов.

Синтаксис:

```
Dim varname As String*N
Dim имя1 as mun1, имя2 as mun2
```

2.3. ПРЕОБРАЗОВАНИЕ ТИПОВ

В VBA существует *явное и неявное* преобразование типов. Неявное преобразование типов наиболее удобно, поэтому часто употребляется. Например, если переменная *a* объявлена как целочисленная:

```
Dim a as Integer
```

то при выполнении присваивания этой переменной результата, возвращаемого функцией *InputBox*, строка, например «52», преобразуется в целое число 52. Если такое преобразование невозможно, то выдается сообщение об ошибке.

```
a=InputBox("","","5")
```

Если же переменная никак не объявлена (имеет тип *Variant*), то преобразования типов не происходит, тип самой переменной преобразуется в тип *String*.

Часто встречается распространенная ошибка при необъявленных типах переменных:

```
Sub пример2()  
a=Inputbox("", "", "3")  
b=Inputbox("", "", "7")  
c=a+b  
MsgBox c  
End sub
```

В этом случае в окне вместо ожидаемого результата *10* появляется результат конкатенации (склеивания) строк – *37*.

Неявное преобразование типов происходит также при конкатенации строк с использованием оператора *&*.

Существуют также функции, которые используются для преобразования строковых переменных в другие типы данных. В качестве примера можно привести функции:

```
CInt(), CDbl(), CDate() ....  
Dim aDateStr as string, aTimeStr as String  
Dim aDate as Date, aTime as Date  
aDateStr="2/13/2014"  
aTimeStr="5:11:23 AM"  
aDate=CDate(aDateStr)  
aTime=CDate(aTimeStr)
```

2.4. ОБЛАСТЬ ДЕЙСТВИЯ ПЕРЕМЕННЫХ

Любая переменная имеет свою *область действия*.

Термин *область действия* относится к области процедуры или модуля VBA, где данная переменная, процедура или другой идентификатор доступны.

Переменные, процедуры и идентификаторы, которые доступны только в процедуре, имеют область действия процедурного уровня, а те, которые доступны для всех процедур в модуле, имеют область действия модульного уровня.

Область в начале модуля перед любыми объявлениями процедур называют областью объявлений модуля, поэтому туда следует помещать объявления переменных модульного уровня и другие директивы, касающиеся всего модуля.

После выполнения процедуры переменная становится невоображаемой, поэтому Excel освобождает соответствующую область памяти. Если требуется сохранить значение переменной, объявите ее как *Static*.

Чтобы сделать переменную доступной во всех процедурах всех модулей VBA в проекте, необходимо объявить переменную на уровне модуля с помощью ключевого слова *Public* перед первой процедурой модуля, например, так:

Public CurrentRate as Long.

Модификатор *Public* должен вводиться в стандартном модуле VBA, а не в коде модуля листа или формы. Если переменная должна быть доступна во всех функциях или процедурах класса или формы, то ее необходимо объявлять перед всеми функциями и процедурами, используя ключевое слово *Dim*.

Константа – именованное значение или строка, которая не меняется при выполнении программы. Константы объявляются с помощью оператора *Const*.

Переменная, определенная как *Date*, является дробным числом, занимает 8 байт памяти и может содержать даты в диапазоне от 1 января 100 года до 31 декабря 9999 года. Целая часть – количество дней с 31.12.1899, дробная часть, умноженная на 24, – часы, минуты, секунды текущего времени. В VBA дата и время определяются как значения, заключенные между знаками #.

Если есть желание всегда использовать явное объявление переменных, то в начале каждого модуля надо записать команду *Option Explicit*. При описании данной директивы VBA требует объявления всех переменных перед их использованием.

Чтобы данная команда автоматически включалась в новый модуль, необходимо поставить «галочку» в следующих настройках:

Tools/Options/Editor/Require Variable declaration.

Программируя в VBA, можно объявлять структуры или пользовательские типы данных при помощи ключевого слова *Type*. Пользовательский тип должен быть объявлен в самом начале модуля:

*Type MyType
name as String
age as Integer
sex as Byte
End Type*

Затем в любой функции или процедуре объявляется переменная типа нового пользовательского типа и выполняются какие-то действия с его элементами:

```
Dim a1 as MyType  
MyType.name = "Masha"  
MyType.age = 13  
MyType.sex = 0
```

Предполагается, что 0 соответствует мужскому полу, 1 – женскому.

2.5. ДИАЛоговые ОКНА ВВОДА/ВЫВОДА

Для реализации простейшего пользовательского интерфейса применяют окно вывода *MsgBox*. Ниже будет показано, как данное окно можно использовать и для ввода в программу информации о сделанном пользователем выборе.

Рассмотрим подробнее параметры *MsgBox*. Отметим, что в квадратных скобках записаны необязательные параметры.

[с=] *MsgBox* (*сообщение* [*кнопки и значения*] [*заголовок окна*] [*справка, раздел*])

с – переменная, которой присваивается возвращаемое *MsgBox* значение.

Круглые скобки используются для описания параметров в том случае, если возвращаемое значение присваивается переменной, в остальных случаях аргументы могут быть описаны через пробел:

Сообщение – текст в окне;

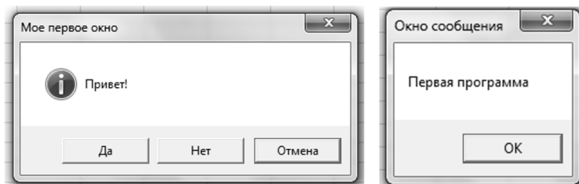
Кнопки и значок – *vbYesNoCancel* + *vbInformation* + *vbDefaultButton3*,
или 3 + 54 + 512

Справка – имя файла справки

MsgBox "Привет!", *vbYesNoCancel* + *vbInformation* + *vbDefaultButton3*,
"Мое первое окно"

MsgBox "Первая программа", , "Окно сообщения"

В этом случае окна сообщений будут выглядеть следующим образом:



Функция *InputBox*, реализующая окно ввода, обычно используется для ввода информации и при нажатии на кнопку ОК возвращает значение переменной строкового типа, которая может быть преобразована к нужному типу явным либо неявным способом.

[c=] InputBox (сообщение [,заголовок] [,строка по умолчанию] [,X][,Y] [,справка])

X, Y – координаты левого верхнего угла в единицах, которые называются *twips* (*1440 twips = 1 дюйм, 567 twips = 1 см*)

В табл. 4 приведены некоторые параметры интерактивных окон ввода/вывода.

Т а б л и ц а 4

Кнопки/значки	Код	Значение
<i>vbOKOnly</i>	0	Только ОК
<i>vbOKCancel</i>	1	ОК, отмена
<i>vbAbortRetryIgnore</i>	2	Стоп, повтор, пропустить
<i>vbYesNoCancel</i>	3	Да, нет, отмена
<i>vbYesNo</i>	4	Да, Нет
<i>vbCritical</i>	16	Критическое
<i>vbQuestion</i>	32	Вопрос
<i>vbExclamation</i>	48	Оповещение
<i>vbInformation</i>	64	Информация

В состав проверочных служебных слов входят следующие слова: *vbOK, vbCancel, vbAbort, vbRetry, vbIgnore, vbYes, vbNo*.

2.6. ОПЕРАЦИИ

Арифметические. Совершенно очевидно, что процесс обработки и преобразования числовой информации не может быть реализован без арифметических и логических операций, которые представлены в табл. 5.

Примеры использования операций:

```
Dim s as string
s="test" + 3 'ошибка
s="test" + "3" 'корректно
s="test" & 3 'корректно
17 / 5 = 3.4
17 \ 5 = 3
17 mod 5 = 2
```

Таблица 5

Операция	Описание
\wedge	Возведение в степень
$-$	Отрицание
$*$	Умножение
$/$	Деление
\backslash	Целочисленное деление
<i>mod</i>	Остаток от деления (операция по модулю)
$+$, $-$	Арифметические операции
$\&$	Объединение строк с преобразованием типов
$+$	Объединение строк (без преобразования типов)
	Перенос строки

Логические. 1. Логическое отрицание (*Not*) меняет значение логического выражения на противоположное значение.

2. Логическое И (*And*) имеет формат записи:

Условие_1 And Условие_2 [And Условие_3]...

Результат истинен тогда и только тогда, когда истинно каждое условие.

3. Логическое ИЛИ (*Or*) имеет формат записи:

Условие_1 Or Условие_2 [Or Условие_3]...

Результат истинен тогда, когда истинно хотя бы одно условие.

4. Операция исключающего ИЛИ *Xor* дает истинный результат тогда и только тогда, когда истинно только одно из условий.

5. Операция логической эквивалентности *Eqv* дает истинный результат, когда оба условия истинны или ложны.

6. Операция логической импликации *Imp* дает ложный результат, когда первое условие истинно, а второе – ложно, в остальных случаях результат истинен.

Сравнения (отношения). В VBA поддерживаются следующие виды операций сравнения:

$>$ – больше; $<$ – меньше; $>=$ – больше или равно; $<=$ – меньше или равно; $=$ – равно; $<>$ – не равно.

Операция **конкатенации** (склейки) текстовых фрагментов с образованием одной длинной строки. Обозначается символами $\&$ или $+$. Ниже приведен пример использования:

Dim A as string, B as string, C as string

A="куст"

B="зеленый"

C=A&B

В строковой переменной *C* значение «куст зеленый».

2.7. МАССИВЫ

Массив – это набор (множество) однотипных данных. Обычно в компьютерной памяти элементы массива расположены рядом для удобства и скорости доступа.

Ниже показан пример объявления целочисленного одномерного массива *Mas1* размерности 6 и целочисленного двумерного массива *Mas2* размерности 5×6. Нумерация «по умолчанию» начинается с нулевого элемента, цифра в скобках указывает на значение верхнего граничного элемента. Если необходимо изменить значение номера нижнего элемента «по умолчанию», то используют директиву *Option Base 1* (в этом случае элементы будут нумероваться с 1):

Dim Mas1(5) As Integer

Dim Mas2(4,5) As Integer

Если необходимо определенно указать, с какого номера элемента и до какого номера элементы массива будут использоваться, то допускается следующая запись:

Dim Mas3(2 to 12) As Integer

Если необходимо изменить размерность массива, можно воспользоваться оператором *ReDim*.

ReDim MyArr(10)

При таком изменении размерности массив очищается (практически под него выделяется новое место в памяти без переноса данных). Если необходимо, чтобы в массиве сохранились записанные ранее элементы, то применяют ключевое слово *Preserve*. Однако в этом случае можно изменить только верхнюю границу массива.

Если заранее неизвестен размер массива, то используют так называемые динамические массивы (объявляют массив пустой размерности):

Dim MyArr2() as Integer

После того как размерность становится известной, используем оператор *ReDim*. Необходимо отметить, что с этим оператором можно использовать только целочисленные переменные в качестве параметров объявления размерности массива:

```
Dim MyArr2()  
a=3  
b=4  
ReDim MyArr2(a,b)
```

В VBA существуют функции, которые возвращают номера соответственно минимального и максимального элемента массива: *LBound()* и *UBound()* соответственно.

При необходимости можно инициализировать (одним оператором задать) элементы массива:

```
Dim MyArr1 as Variant  
MyArr1 =Array (1,2,3,4,5)  
Dim MyArr() as Variant  
MyArr = [{1, 2, 3, 4, 5}]
```

В данном случае инициализируется одномерный массив из пяти элементов. Нумерация элементов при такой инициализации начинается с единицы. Если выполнить команду: *ReDim Preserve MyArr(4)*, то нумерация по умолчанию начнется с нуля.

В следующем примере показана инициализация двумерного массива

```
Dim MyArr as Variant  
MyArr = [{1, 2, 3; 4, 5, 6; 7, 8, 9}]
```

При использовании функций или процедур для работы с внешними массивами можно применять массивы, объявленные как глобальные переменные, однако лучше передавать их в процедуру или функцию в качестве параметра по ссылке (*ByRef*).

При передаче массива в функцию или процедуру по ссылке объявление параметра выглядит следующим образом:

```
Sub пример3 (ByRef a()) as Integer).
```

2.8. УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ

Управляющие структуры позволяют управлять последовательностью выполнения программы.

Без этих структур программа будет выполняться слева направо и сверху вниз. Однако иногда требуется многократно выполнять некото-

рый набор инструкций автоматически либо решить задачу по-другому в зависимости от значения переменных или параметров, заданных пользователем. Для этого служат конструкции управления и циклы.

VBA поддерживает следующие конструкции принятия решений:

If . . . Then
If . . . Then . . . Else
Select Case

Конструкция If . . . Then

Применяется, когда необходимо выполнить один или группу операторов в зависимости от некоторого условия. Синтаксис этой конструкции позволяет задавать ее в одной строке или в нескольких строках программы:

1) *If условие Then выражение*
2) *If условие Then*
 выражение
 End If

Следующие два оператора эквивалентны:

1) *If anyDate < Now Then anyDate = Now*
2) *If anyDate < Now Then*
 anyDate = Now
 End If

Конструкция If . . . Then . . . Else

Определяет несколько блоков операторов, один из которых будет выполняться в зависимости от условия:

If условие1 Then
 Выражение 1
ElseIf условие2 Then
 Выражение 2
 .
 .
 .
Else
 Выражение - n
End If

Рассмотрим пример вычисления функции:

$$f(x) = \begin{cases} \sin(x), & \text{если } x = -\pi / 4, \\ \cos(x), & \text{если } x = 0, \\ \operatorname{tg}(x), & \text{если } x = -\pi / 4. \end{cases}$$

```

Sub пример4()
Dim a As Single, b As Single, x As Single
Dim f As Double
a = Cells(1, 2)
b = Cells(2, 2)
Let x = CSng(InputBox("введи x", "Ввод данных", 0))
If x <= a Then
f = Sin(x)
ElseIf x >= b Then
f = Tan(x)
Else: f = Cos(x)
End If
Cells(3, 2) = f
End Sub

```

Конструкция Select Case

Эта конструкция является альтернативой конструкции If . . . Then . . . Else в случае выполнения блока, состоящего из большого набора операторов.

Конструкция *Select Case* предоставляет возможность управления, похожую на возможность конструкции *If . . . Then . . . Else*, но в отличие от нее она делает код более читаемым при наличии нескольких вариантов выбора.

Конструкция *Select Case* работает с единственным проверяемым выражением, которое вычисляется один раз при входе в эту конструкцию. Затем VBA сравнивает полученный результат со значениями, задаваемыми в операторах *Case* конструкции. Если найдено совпадение, выполняется блок операторов, ассоциированный с оператором *Case*.

```

Select Case проверяемое выражение
[Case список выражений 1
[блок операторов 1]]
[Case список выражений 2
[блок_операторов2]]
. . .
[Case Else
[блок операторов - n]]
End Select

```


Рассмотрим пример вычисления функции:

$$f(x) = \begin{cases} \sin(x), & \text{если } x \leq a, \\ \cos(x), & \text{если } a < x < b, \\ \operatorname{tg}(x), & \text{если } x \geq b. \end{cases}$$

```
Sub пример5()  
Const pi4 as Single = 0.7854  
Dim x As Single, f As Double  
Let x = CSng(InputBox("введи x", "Ввод данных", 0))  
Select Case x  
Case -pi4  
f = Sin(x)  
Case 0  
f = Cos(x)  
Case pi4  
f = Tan(x)  
Case Else  
MsgBox "Неверные исходные данные!"  
Exit Sub  
End Select  
Cells(3,2)=f  
End Sub
```

2.9. ЦИКЛЫ

Обычно циклы используются для многократного выполнения последовательности действий. Количество повторений может быть задано заранее либо может определяться одним или несколькими условиями.

Цикл For ... Next

Используется для выполнения заданного количества повторений. При этом параметр цикла (переменная цикла) может увеличиваться или уменьшаться на заданную величину, определяемую параметром *Step* (*шаг*). По умолчанию шаг равен 1. Примеры цикла *For ... Next*:

```
For i = 1 To 5  
MsgBox i  
Next i  
For n=3 to 12 Step 3  
MsgBox n
```

```

Next n
For n=10 to 1 Step -1
MsgBox n
Next n

```

Цикл For Each ... Next

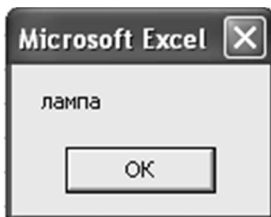
Цикл *For Each . . . Next* похож на цикл *For . . . Next*, но он повторяет группу операторов для каждого элемента из набора объектов или из массива вместо повторения операторов заданное число раз. Пример:

```

Sub пример6()
Dim TestArray(1 To 3) As String, i As Variant, sum As String
TestArray(1) = "ла"
TestArray(2) = "м"
TestArray(3) = "на"
For Each i In TestArray
sum = sum + i
Next i
MsgBox sum
End Sub

```

Результат работы программы отображен в диалоговом окне MsgBox:



Цикл Do ... Loop

Данный цикл имеет несколько модификаций: с предусловием и с постусловием (табл. 6).

Ниже приводится пример использования одной из этих конструкций.

```

Sub doTest()
x=80
Do Until x=100
x=x+1
Loop
MsgBox x
End Sub

```

Таблица 6

Конструкция цикла	Описание
<i>Do While ... Loop</i>	Цикл с предусловием выполняется до тех пор, пока условие «истинно»
<i>Do Until ... Loop</i>	Цикл с предусловием выполняется до тех пор, пока условие «ложно»
<i>Do ... Loop While</i>	Цикл с постусловием выполняется до тех пор, пока условие «истинно»
<i>Do ... Loop Until</i>	Цикл с постусловием выполняется до тех пор, пока условие «ложно»

Циклы, процедуры и функции можно прерывать. Прерывание циклов может понадобиться, если ведется поиск какого-то элемента в массиве и этот элемент найден. Если циклы вложенные, то выход происходит только из одного цикла.

For ... Next и *For Each ... Next - Exit For*

Do ... Loop - Exit Do

Конструкции *Exit Sub* и *Exit Function* позволяют досрочно прервать выполнение процедуры и функции.

2.10. СИМВОЛЬНЫЕ ДАННЫЕ. ОПЕРАЦИИ И ФУНКЦИИ ДЛЯ РАБОТЫ С СИМВОЛЬНЫМИ ДАННЫМИ

По умолчанию данные строкового типа имеют переменную длину и могут удлиняться (до длины в 2^{31} символов) или укорачиваться. Однако такие строки занимают на 10 байт памяти больше, поэтому можно объявить строки фиксированной длины, явно указав количество символов.

*Dim s as String*20*

Если количество символов будет меньше объявленного, то свободные места заполняются пробелами, при попытке занесения большего количества символов лишние отбрасываются.

Единственная операция, применяемая при работе со строками – конкатенация (склейка), зато встроенных функций для работы достаточно много. Основные из них приведены в табл. 7.

Пример. Заменить в строке буквы А, Б, В на 1, 2, 3 соответственно.

Sub пример7()

Dim s As String, sn As String, t As String

```

Dim l As Integer, i As Integer
s= Cells(1, 7)
For i = 1 To Len(s)
t = Mid(s, i, 1)
Select Case t
Case "A": sn = sn + "1"
Case "Б": sn = sn + "2"
Case "В": sn = sn + "3"
Case Else: sn = sn + t
End Select
Next i
Cells(1, 8)= sn
End Sub

```

Т а б л и ц а 7

Функция	Описание	Пример
<i>Len(строка)</i>	Определяет длину <i>строки</i>	a=len("Персонажи") a=9
<i>Left(строка, длина)</i>	Выделяет из аргумента <i>строка</i> указанное количество символов слева	Left(" 1234string", 4) ="1234"
<i>Right(строка, длина)</i>	Выделяет из аргумента <i>строка</i> указанное количество символов справа	Right(" 1234string", 6) ="string"
<i>Mid(строка, старт, длина)</i>	Выделяет из аргумента <i>строка</i> подстроку с указанным числом символов, начиная с позиции <i>старт</i>	Mid ("12345678", 4, 3) ="456"
<i>LTrim(строка)</i>	Удаляет пробелы в начале <i>строки</i>	LTrim(" печать") ="печать"
<i>RTrim(строка)</i>	Удаляет пробелы в конце <i>строки</i>	RTrim("печать ") ="печать"
<i>Trim(строка)</i>	Удаляет пробелы в начале и в конце <i>строки</i>	Trim(" печать ") ="печать"
<i>InStr([старт,] строка 1, строка 2 [,сравнение])</i>	Производит поиск подстроки в строке. Возвращает позицию первого вхождения <i>строка 2</i> в <i>строку 1</i> , <i>старт</i> – позиция, с которой начинается поиск	Instr("C:Temp test.mdb", "Test")=9 Если искомая строка не находится, функция возвращает 0
<i>Replace(строка, строка Поиск, строка Замена)</i>	Позволяет заменить в строке одну подстроку другой	Mystring = Replace ("мама", "а", "м") возвращает "мммм"

Функция	Описание	Пример
<i>StrComp(строка 1, строка 2 [, сравнение])</i>	Позволяет сравнивать две строки. Способы <i>сравнения</i> : 0 – двоичное, 1 – посимвольное сравнение без учета регистра	MyComp = StrComp("ABCD", "abcd", 1) возвращает значение 0
<i>StrConv(строка, сравнение)</i>	Позволяет преобразовать строку <i>vbUpperCase</i> 1 – к верхнему регистру. <i>vbLowerCase</i> 2 – к нижнему регистру. <i>vbProperCase</i> 3 – первую букву каждого слова в строке в прописную	MyConv = StrConv("abcd", 1) возвращает значение "ABCD"
<i>Asc(строка)</i>	Возвращает код символа по заданному символу <i>строки</i>	Asc("a") возвращает код строчной буквы <i>a</i>
<i>Chr(кодСимвола)</i>	Возвращает символ по заданному коду от 0 до 255	Chr(10) возвращает символ перевода строки

2.11. РАБОТА С ТЕКСТОВЫМИ ФАЙЛАМИ

Теперь несколько слов о работе с текстовым файлом – наиболее универсальным хранилищем информации. Может быть, текстовые файлы не всегда удобны и надежны для хранения информации, но так как они менее всего подвержены изменению «формата», то могут быть прочитаны и через десятилетия.

Для работы с текстовым файлом требуется знать имя файла, путь к нему, задать файловую переменную или дескриптор файла, объявленную, как длинное целое число (в некоторых случаях для замены переменной-дескриптора используют конструкцию из решетки и целого числа, например #1). Перед открытием файла дескриптор инициализируют, используя функцию *FreeFile*. Обычно программисты VBA работают с текстовыми файлами, используя для чтения строк конструкцию *Line Input*, а для записи – оператор *Print*.

Пример чтения/записи из одного текстового файла в другой текстовый файл:

```
Sub пример8()
Dim path As String
Dim fNameIn As String
```

```

Dim fNameOut As String
Dim hFileOut As Long
Dim hFileIn As Long
Dim strBuf As String
fNameIn = "Input.txt"
path = ActiveWorkbook.path & "\"
dim ss as string
ss=""
hFileIn = FreeFile
Open path + fNameIn For Input Access Read As hFileIn
Do Until EOF(hFileIn)
Line Input #hFileIn, strBuf
ss= ss+strBuf + vbLF
Loop
Close hFileIn
End sub

```

Пример записи в файл:

```

fNameOut="Output.txt"
hFileOut = FreeFile
Open path + fNameOut For Output Access Write As hFileOut
Print #hFileOut, Trim(strBuf)
Close hFileOut

```

В предыдущем примере показано, что при открытии файла на запись, если файл с данным именем существовал, то он должен был перезаписаться, а запись начаться заново. Если необходимо дописывать информацию в конец файла, то можно воспользоваться следующей конструкцией:

```

Sub Add_str_to_txt()
Dim iText As String
Dim iFullName As String
iFullName = "C:\testfile.txt"
iText="Текст добавляемой строки"
Open iFullName For Append As #1
Print #1,iText
Close #1
End Sub

```

3. ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Практическое задание № 1

ВВОД ДАННЫХ И КОРРЕКТИРОВКА ИНФОРМАЦИИ В ЭЛЕКТРОННОЙ ТАБЛИЦЕ

1. Задание

Ввести в электронную таблицу следующий документ в том виде, как он представлен на рис. 3 (порядок ввода документа рассматривается ниже).

	A	B	C	D	E	F	G
1							
2		Сведения о заработной плате сотрудников					
3	Табельный	Ф.И.О.	январь	февраль	март	апрель	Итого
4	1	Сидоров А.Г.	320	350	400	400	
5	2	Иванов П.П.	300	450	400	420	
6	6	Емельянов Е.О.	340	430	420	480	
7	4	Семенова А.А.	360	370	410	430	
8	3	Белоконь П.Н.	430	390	390	410	
9	7	Иванов И.И.	420	420	400	400	
10	9	Рыбин С.В.	350	380	370	400	
11	5	Петров В.Н.	300	400	390	410	
12	8	Кетков В.Н.	370	410	400	450	
13		Итого					
14							

Рис. 3. Вводимый документ

2. Ввод данных

Данные вводят в активную ячейку. Активной называется ячейка, в которой в настоящий момент расположен табличный (или оконечный) курсор. Курсор перемещают с помощью клавиатуры или мыши.

Введите данные в пустую электронную таблицу в следующем порядке:


- сделайте активной ячейку *B2* и введите название таблицы «Сведения о заработной плате сотрудников». Обратите внимание, что вводимая информация дублируется в строке формул. Для окончания ввода текста в ячейку необходимо или нажать клавишу *Enter* или перевести табличный курсор в другую ячейку;
- сделайте активной ячейку *A3* и введите название первой колонки «Табельный номер». Введенная информация размещается в нескольких ячейках;
- переведите курсор мыши в ячейку *B3* и введите название второй колонки «Ф.И.О.». Не беспокойтесь о сохранности информации в ячейке *A3*. Сделайте ячейку *A3* активной, и вы увидите полностью название колонки. На экран терминала выводится столько символов текста, сколько их можно разместить в ячейке, но в электронной таблице сохраняется весь текст;
- аналогично введите названия остальных столбцов документа. Ниже мы улучшим внешний вид шапки документа;
- введите собственно данные документа за исключением столбца и строки «Итого».

3. Операция суммирования данных


При вводе данных их тип распознается автоматически, но для формул первым вводимым символом должен быть знак $=$.

Определим содержимое подстолбца *G4:G12*. Подстолбец — это множество ячеек, расположенных рядом в одном столбце, начиная с ячейки *G4* в нашем примере и заканчивая ячейкой *G12*.

Аналогично определяется подстрока в электронной таблице. Блок ячеек, например *A1:C4*, — это множество ячеек, расположенных в прямоугольнике, левый верхний угол которого *A1*, а правый нижний — *C4*. Рассмотрим возможные варианты занесения данных в *G4:G12*:

- сделайте активной ячейку «Итого» — «Сидоров». Нажмите кнопку в пиктографическом меню  («Автосуммирование»), в ячейке появится формула $=\text{СУММ}(C4:F4)$. Нажмите *Enter*. В ячейку *G4* будет внесено число 1470, которое получено при сложении $C4 + D4 + E4 + F4 = 320 + 350 + 400 + 400 = 1470$, а в строке формул будет записана

соответствующая расчетная формула. Знаки пробела в формулах не важны;

- сделайте активной ячейку «Итого» – «Иванов». Нажмите кнопку в пиктографическом меню  («Вставка/Функция»). В списке «Категория» выберите класс функций «Математические». В списке «Функция» с помощью кнопок линейки прокрутки найдите и выделите функцию СУММ, нажмите кнопку *OK*. В строку аргумента введите с помощью мыши диапазон суммируемых ячеек, нажмите *OK*.

4. Копирование формул

Скопируйте содержимое ячейки *G5* в ячейки *G6:G12*, для чего сделайте активной ячейку *G5*. Нажав правую кнопку мыши, вызовите контекстно-зависимое меню, в котором выделите команду «Копировать». Ячейка *G5* будет помещена в рамку с движущимися элементами. Маркируйте (т.е. сделайте активной) область *G6:G12* и нажмите *Enter*.

Любым из способов заполните ячейки, в каждой из которых содержится сумма заработной платы сотрудников отдела за соответствующий месяц.

5. Копирование с помощью мыши


Выполним копирование формул еще раз, но с помощью мыши. Очистите подстолбец «Итого» (*G5:G14*), для чего маркируйте указанную область и нажмите клавишу *Del* (или выполните команду контекстно-зависимого меню «Очистить содержимое»).

Вначале выполните операцию копирования данных из ячейки в ячейку. Маркируйте ячейку «Итого» – «Сидоров» (*G4*). Переместите курсор мыши на нижний край рамки маркировки так, чтобы указатель превратился в стрелку. Нажмите клавишу *Ctrl* (рядом с курсором появится знак плюс) и, удерживая левую кнопку мыши, выполните транспортировку мыши до тех пор, пока не охватите рамкой остальные ячейки. Отпустите кнопку мыши и клавишу *Ctrl*.

Далее переместите данные с помощью мыши. При перемещении данных в отличие от копирования исходная ячейка очищается. Маркируйте ячейку *B14* и переместите курсор мыши на границу ячейки так, чтобы он превратился в стрелку. Зажав левую кнопку мыши, перенесите рамку в ячейку *A14*. Отпустите кнопку мыши. Перемещение данных закончено. Отмените выполненную операцию, нажав на кнопку пиктографического меню «Отменить».

6. Улучшение внешнего вида документа

- Выделите рамкой названия столбцов. Маркируйте область названий столбцов таблицы (A3:G3). Вызовите команду через контекстно-зависимое меню «Формат ячеек» и перейдите к листу «Рамка». Определите положение рамки и тип линии. Нажмите кнопку *ОК*.

- Расположите табельные номера по центру, для чего маркируйте содержащую их область (A4:A12). Нажмите кнопку в пиктографическом меню  («По центру»).


- Расположите название первого столбца в две строчки, для чего сделайте активной ячейку A3. По команде «Формат ячеек» после открытия диалогового окна «Формат ячеек» на листе «Выравнивание» включите переключатель «Переносить по словам» и нажмите кнопку *ОК*.

- Измените ширину второго столбца. Подведите курсор мыши к координатной ячейке столбца В справа. Курсор трансформируется в двунаправленную стрелку. Отбуксируйте вправо на нужное расстояние разделитель столбцов и отпустите кнопку мыши.

- Расположите в центре ячейки название столбцов документа.

- Заключите известным способом в рамку содержимое (A13:G13) подстроки «Итого».

- Измените шрифт для заголовка документа, используя команду контекстно-зависимого меню «Формат ячеек» и лист «Шрифт». Поэкспериментируйте с различными характеристиками оформления текста, такими как шрифт, стиль, размер, эффект. Выберите наиболее удачный, с вашей точки зрения, вариант шрифта для заголовка.

- Измените также шрифт текста в последней строке таблицы, для чего используйте кнопку в пиктографическом меню  («Повторить»). Команда «Повторить» ускоряет работу в ситуации, когда одна и та же операция выполняется над различными объектами.

7. Результат

Если вы успешно справились с каждой операцией по вводу данных и оформлению внешнего вида документа, то в конечном итоге ваш документ должен приобрести вид, показанный на рис. 4.

8. Сохранение документа на диске


Сохраните полученный документ в вашей рабочей области под именем `primer.xls`. По умолчанию электронные таблицы хранятся в файлах с расширением `.xls`. Выполните команду «Файл/Сохранить как». В открывшемся окне в поле ввода «Имя файла» введите название

файла *primer.xls*. Обратите внимание на название открытой в данный момент папки. При необходимости откройте папку, где сохраняется файл. Наконец, нажмите на командную кнопку «Сохранить». Закройте окно документа.

Сведения о заработной плате сотрудников						
табельный номер	Ф.И.О.	Январь	Февраль	Март	Апрель	Итого
1	Сидоров А.Г.	320	360	400	400	1470
2	Иванов П.П.	300	450	400	420	1570
3	Емельянов Е.О.	340	430	420	480	1670
4	Семенова А.А.	360	370	410	430	1570
5	Белоконь П.Н.	430	390	390	410	1620
6	Иванов И.И.	420	420	400	400	1640
7	Рыбин С.В.	350	380	370	400	1500
8	Петров В.Н.	300	400	390	410	1500
9	Кетков В.Н.	370	410	400	450	1630
	Итого	3190	3600	3580	3800	14170

Рис. 4. Итоговый документ

9. Размещение файла в электронной таблице

Откройте пустую электронную таблицу, для чего нажмите кнопку в пиктографическом меню с именем «Создать рабочую книгу» (по указанному имени найдите самостоятельно нужную кнопку, используя систему помощи Excel). Новая электронная таблица будет иметь имя «Книга2». Загрузите в пустую электронную таблицу документ из созданного Вами файла *primer.xls*. Используйте команду «Файл/Открыть» или кнопку  («Открыть»). В диалоговом окне «Открыть» откройте нужную папку и укажите имя файла *primer.xls*. Выполните команду «Открыть». Файл загружен.

10. Изменение данных в электронной таблице

Рассмотрим порядок внесения изменений в данные электронной таблицы. Пусть необходимо внести в документ сведения о заработной плате еще одного сотрудника отдела С.В. Попова. Для этого нужно вставить чистую строку в таблицу и заполнить ее уже известным способом. Для вставки строки маркируйте координатную ячейку строки, перед которой нужно вставить новую строку. В нашем примере это строка № 13. Вызовите через контекстно-зависимое меню команду «Добавить». После вставки новой строки нумерация строк автоматически изменится. Введите данные в 14-ю строку.

Откорректируйте формулы в подстроке «Итого». В этом случае исправления можно вносить в следующем порядке:

- маркируйте ячейку «Итого» – «Январь». Нажмите клавишу *F2*, после чего вы получите возможность изменять данные (текст, число или формулу) активной ячейки. Адресная часть формулы должна содержать адрес ячейки из добавленной строки. Корректировка заканчивается нажатием клавиши *Enter*;
- примените копирование мышью для изменения остальных столбцов.

Добавим в таблицу три новых столбца, в которых вычислим минимальную, среднюю и максимальную зарплату каждого сотрудника за весь период. Для этого, используя мастер функций, выберем из категории «Статистические» функции МИН, СРЗНАЧ, МАКС.

Аналогичным образом вычислим значения этих же функций для каждого месяца, добавив в таблицу три соответствующие строки.

11. Шаблоны электронных таблиц

Удалите данные из документа, очистив область, в которой находятся фамилии и значения заработной платы (B4:F13). Теперь ваш документ приобрел вид шаблона. Сохраните его в файле с именем *primer*, но выбрав тип файла «Шаблон». В дальнейшем этот файл (с расширением *.xlt*) можно использовать для внесения данных в новый документ с такой же структурой. Именно для этого предназначены шаблоны. В файле шаблона сохраняются не только заголовок и шапка документа, но и формулы (пока в соответствующих ячейках записаны нули), а также форматы ячеек (размеры ячеек, шрифты, выравнивание, рамки, представление числовых данных и т. д.). Для открытия шаблона используйте команду меню «Файл/Создать».

12. Построение диаграмм

Диаграммы дают возможность наглядно представить динамику изменения данных электронной таблицы. Рассмотрим работу конструктора диаграмм Excel на примере построения диаграммы изменения итоговой заработной платы сотрудников отдела по месяцам.

Маркируйте области таблицы, содержащие итоги по месяцам и названия месяцев. Маркировать вторую часть области данных необходимо при нажатой клавише *Ctrl*. Используйте команду построения диаграммы «Вставка/Диаграммы» или соответствующую пиктограмму. Далее выполняем следующие действия:

- 1) выбираем тип диаграммы «Гистограмма» и ее вид;
 - 2) проверяем диапазон адресов маркированных областей данных;
 - 3) вводим названия диаграммы и осей;
 - 4) выбираем режим построения диаграммы «На новом листе».
- Окончательный вид гистограммы показан на рис. 5.

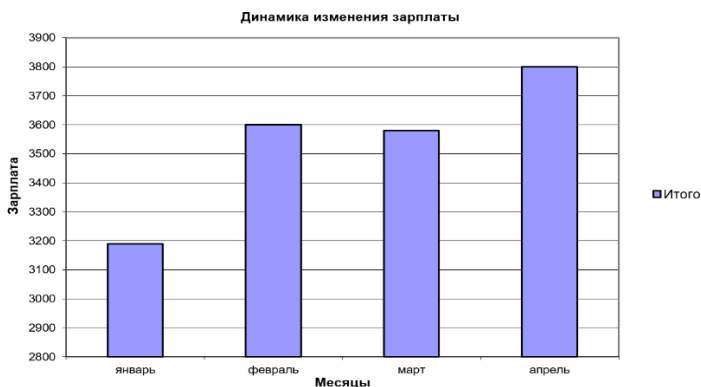


Рис. 5. Гистограмма

Затем необходимо построить диаграммы.

- Построить смешанную (комбинированную) диаграмму, в которой в виде гистограммы представлена зарплата за январь, а в виде графика — доля январской зарплаты в процентах от общего итога.

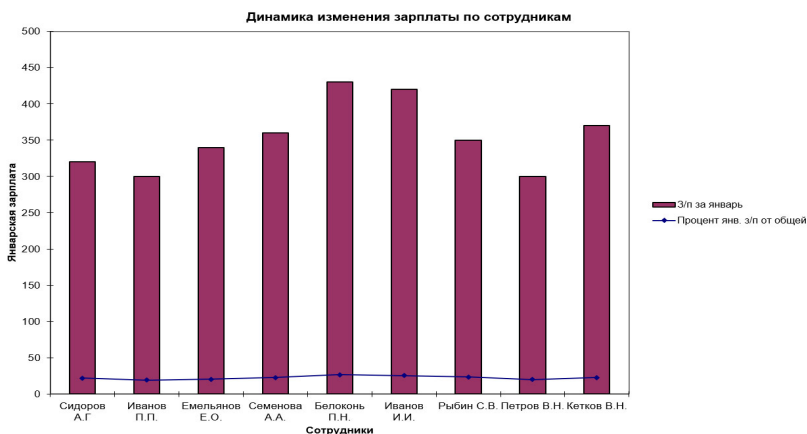


Рис. 6. Смешанная (комбинированная) диаграмма

- Построить график с дополнительной осью Y, в котором по основной оси отложена январская зарплата, по дополнительной – итоговая.



Рис. 7. График с дополнительной осью Y

СОДЕРЖАНИЕ ОТЧЕТА

1. Задания.
2. Полученные таблицы и шаблоны документа.
3. Построенные диаграммы.
4. Выводы по работе.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какого типа данные можно ввести в ячейку Excel?
2. Какие операции можно применить к ячейке (группе ячеек)?
3. Какие операции можно применить к листам рабочей книги?
4. Какие есть варианты очистки ячеек, и чем они отличаются?
5. Что такое шаблон и зачем он нужен?
6. Какие мастера есть в Excel?
7. Какие действия выполняются на каждом шаге мастера диаграмм?
8. Как построить диаграмму с дополнительной осью ординат?
9. Что (и как) можно изменить в уже построенной диаграмме?

Практическое задание № 2

ЛИНЕЙНЫЕ, РАЗВЕТВЛЯЮЩИЕСЯ, ЦИКЛИЧЕСКИЕ И ИТЕРАЦИОННЫЕ АЛГОРИТМЫ.

КРАТКОЕ ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

Алгоритмы можно представлять как некоторые структуры, состоящие из отдельных **базовых** (т. е. основных) **элементов**. Естественно, что при таком подходе к алгоритмам изучение основных принципов их конструирования должно начинаться с изучения этих базовых элементов.



Рис. 8. Базовая структура следования

Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур: *следование, ветвление, цикл*.

Характерной особенностью базовых структур является наличие в них одного входа и выхода.

Базовая структура «следование» (рис. 6.) образуется из последовательности действий, следующих одно за другим. Алгоритмы, содержащие только базовые структуры «следование», называются линейными.

Базовая структура «ветвление». Обеспечивает в зависимости от результата проверки условия (да или нет) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к **общему выходу**, так что работа алгоритма будет продолжаться независимо от того, какой путь выбран.

Структура «ветвление» существует в четырех вариантах (рис. 9).

- 1) **если условие**
 то действия
 конец если
- 2) **если условие**
 то действие 1
 иначе действие 2
 конец если
- 3) **выбор-иначе**
 при условии 1: действие 1
 при условии 2: действие 2
 при условии N: действие N
 иначе: действие N+1
 конец выбора

4) **выбор**

при условии 1: действие 1

при условии 2: действие 2

.....

при условии N: действие N

конец выбора

Базовая структура «цикл». Обеспечивает **многократное выполнение** некоторой совокупности действий, которая называется **телом цикла** и существует в трех основных вариантах.

- **Цикл для.** Предписывает выполнять тело цикла для всех значений некоторой переменной (параметра цикла) в заданном диапазоне.

- **Цикл пока.** Предписывает выполнять тело цикла пока условие.

- **Цикл делать – пока.** Предписывает выполнять тело цикла до тех пор, пока выполняется условие. Условие проверяется **после** выполнения тела цикла.

Заметим, что циклы **для** и **пока** называют также циклами с *предпроверкой условия*, а циклы **делать–пока** – циклами с *постпроверкой условия*. Иными словами, тела циклов **для** и **пока** могут не выполниться ни разу, если условие окончания цикла изначально неверно. Тело цикла **делать–пока** выполнится минимум раз, даже если условие окончания цикла изначально неверно.

Варианты базовой структуры «цикл» показаны на рис. 10.

цикл для i от i_1 до i_2 шаг i_3

тело цикла (последовательность действий)

конец цикла

цикл пока условие

тело цикла (последовательность действий)

конец цикла

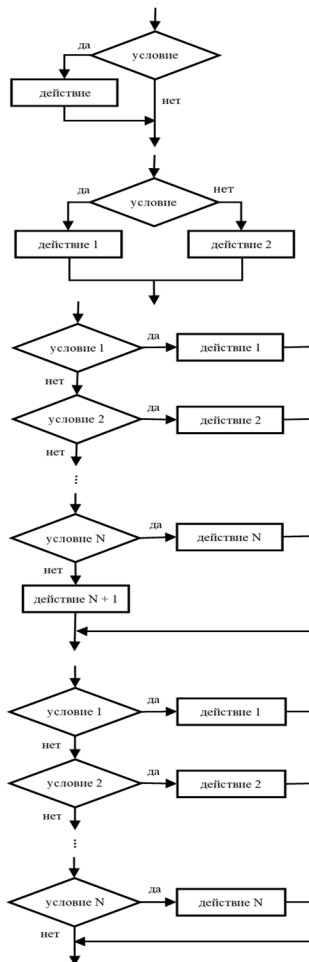


Рис. 9. Варианты базовой структуры «ветвление»

цикл делать

тело цикла (последовательность действий)

пока условие

конец цикла

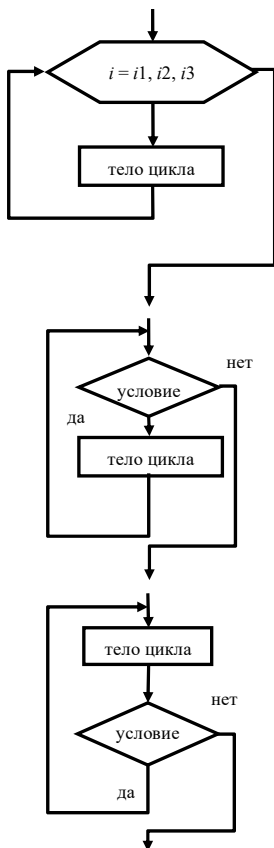


Рис. 10. Варианты базовой структуры «цикл»

Итерационные циклы. Особенность итерационного цикла в том, что число повторений операторов тела цикла заранее неизвестно. Для его организации используется цикл **пока**. Выход из итерационного цикла осуществляется в случае выполнения заданного условия. На каждом шаге вычислений происходит **последовательное приближение и проверка условия достижения искомого результата**.

Алгоритм, в состав которого входит итерационный цикл, называется *итерационным алгоритмом*. Итерационные алгоритмы используются при реализации итерационных численных методов. В таких алгоритмах необходимо непременно достижение условия выхода из цикла (*сходимость итерационного процесса*). В противном случае произойдет *заикливание* алгоритма, т. е. не будет выполняться основное его свойство – результативность.

ЗАДАНИЯ

1. Научиться вычислениям по сложным формулам с использованием математических функций, используя линейный алгоритм. Вычислить значения функции $y = f(a, b)$ для заданных значений a и b по вариантам, приведенным в табл. 8.
2. Научиться реализовывать разветвляющиеся алгоритмы. Вычислить значение функции $f(t)$ при заданных значениях a, b, n , если значение аргумента t изменяется от $t_{\min} = a$ до $t_{\max} = b$ с шагом $\Delta t = (b - a)/(n - 1)$ по вариантам в табл. 9.

Таблица 8

Номер варианта	$y = f(a, b)$	Значения	
		a	b
1	$\left(0,75\sqrt{b} - \frac{1}{2}\sqrt[3]{a}\right)\sin\left(b\frac{\pi}{3}\right)$	4	0,5
2	$\frac{8,15\sqrt[3]{b} \cdot \ln(a)}{24,38\cos(b)(e^a - a^a)}$	2	14,36
3	$\frac{0,314e^a - 0,512e^b}{\sin\left(\frac{b}{3}\pi\right)}\ln(a)$	10	0,5
4	$\sqrt[3]{\frac{0,1\ln(e^3)}{\sin^2\left(\frac{a}{b}\pi\right) + \cos^2\left(\frac{a}{b}\pi\right)}}$	2	11,05
5	$\frac{1 - \cos^2\left(\frac{b}{a}\pi\right)}{0,5e^b + 0,312e^a}$	3	0,151
6	$\frac{\sqrt[3]{a} \cdot \cos(e^b - 1)\frac{\pi}{4}}{0,5121\left(1 - \ln\left(\frac{b}{a}\right)\right)}$	4	1,310
7	$\left(\frac{11,21\left(1 + \ln\left(\frac{b}{a}\right)\right)}{1 - \sin\left(\frac{a}{b}\pi\right)\cos\left(\frac{a}{b}\pi\right)}\right)^{\frac{2}{3}}$	2	200,0
8	$\left(0,81\sqrt[3]{a} - \frac{1}{2,125}\sqrt[3]{b}\right)e^a$	1	12,21
9	$\sqrt[3]{\left(\frac{0,127e^{\left(\frac{b}{a}\right)}}{1 - \sqrt[3]{\cos\left(\frac{b}{a}\pi\right)}}\right)^2}$	3	0,521

Номер варианта	$y = f(a, b)$	Значения	
		a	b
10	$\frac{0,5(\ln(a) + \ln(b))}{\sqrt[3]{\cos\left[(a+b)^2 + e^a\right] \frac{\pi}{6}}}$	2	12,11
11	$\sqrt[3]{\frac{0,719}{b} \frac{b^2 - a^2}{a^2 + b^2} + \cos\left((\ln(b)) \frac{\pi}{6}\right)}$	1	20,01
12	$\left[0,5 \sin\left((\ln(e^{(b+a)})) \frac{\pi}{8}\right) + 1,308 \cos\left((\ln(e^{(b-a)})) \frac{\pi}{8}\right)\right]^{\frac{2}{3}}$	3	0,707
13	$\left(\frac{a^2}{b^3}\right)^{\frac{2}{3}} e^{\frac{0,807\left[1 - \sin^2\left((a-b) \frac{\pi}{4}\right)\right]}{0,312\left[1 + \cos^2\left((b+a) \frac{\pi}{4}\right)\right]}}$	3	2,712
14	$\left(\frac{b}{a}\right) \frac{4,3 \sin\left[\left(\frac{a}{b} + 1\right) \pi\right]}{1 - \cos\left[\left(\frac{a}{b} - 1\right) \pi\right]} + \ln(b)$	2	19,03
15	$7,2(a+b) \left[\left(1 + \cos^2(a)\right) \left(1 - \cos(b)\right) + 0,017 \ln(a+b)^2 \right]$	4	50,1
16	$\frac{\sqrt[3]{(a-b)^2 \sin^2\left(1 - \frac{a}{b}\right) \frac{\pi}{3} \cos^2\left(1 - \frac{b}{a}\right) \frac{\pi}{3}}}{0,701 \ln((a-b)^2)}$	3	0,501
17	$\sqrt[3]{\frac{1 - e^a}{b} \cos\left(\frac{b}{a} \pi\right) + \ln(0,708b)}$	2	13,13
18	$\sqrt[3]{\frac{a-b}{b+a} e^{\frac{\ln\left[\cos(a-b) \frac{\pi}{8}\right]}{0,137}}}$	1	0,001
19	$e^{\frac{0,5(1 - \cos(b\pi))}{1 - \sin(a\pi)}} + e^{\frac{0,3(1 + \cos(a\pi))}{1 + \sin(b\pi)}}$	4	0,707

Номер варианта	$y = f(a, b)$	Значения	
		a	b
20	$\left[0,7 \ln \left(\frac{a}{b}\right) \sin ^2\left((a+b) \frac{\pi}{10}\right)+0,8 \ln \left(\frac{b}{a}\right) \cos ^2\left((a+b) \frac{\pi}{10}\right)\right]^{\frac{3}{2}}$	1	1,201
21	$\sqrt[0,5 \ln \left(\frac{1-\cos \left[\left(\frac{a}{b}\right) \pi\right]}{1+\cos \left[\left(\frac{a}{b}\right) \pi\right]}\right)}{e}+e^{\sqrt[0,3 \ln \left(\frac{1-\sin \left[\left(\frac{b}{a}\right) \pi\right]}{1+\sin \left[\left(\frac{b}{a}\right) \pi\right]}\right)}{e}}$	2	13,17
22	$\left(e^{(1-b)}+\frac{1}{9} \sqrt[3]{(a-b)^2}\right) \cos \left(b \frac{\pi}{7}\right)$	2	3,141
23	$\frac{0,75 e^{(1-b)}+0,31 e^{(1-a)}}{0,731+\sin ^2\left(\frac{b}{a} \pi\right)}$	4	3,141
24	$\frac{1,3 \sin (a)+\cos (b)}{0,81 \sin (b)+1,1 \cos (a)} e^{\ln \left(\frac{b}{a}\right)}$	2	15,15
25	$\sqrt[4]{\frac{1,56 \sqrt{\sin (b)}}{0,8942 \ln (a)}}$	3	0,523
26	$\left(e^{(5-b)}+33,5 a \sqrt{b}-\frac{1}{7} \sqrt[5]{a b}\right) \cos \left(a \frac{\pi}{4}\right)$	2	0,7
27	$\left(10,1 \sqrt[3]{a-b}+\frac{1}{22,25} \sqrt[4]{a b}\right) e^{2 a}$	5	12,21
28	$\sqrt[4]{\frac{0,9}{a} \frac{a^3-b^3}{a^2+b^2}+\sin \left((\ln (b-a)) \frac{\pi}{3}\right)}$	3	9,301
29	$\left[2,5 \cos \left((\ln \left(e^{(b+2 a)}\right)) \frac{\pi}{4}\right)+15,8 \sin \left((\ln \left(e^{(2 b-a)}\right)) \frac{\pi}{16}\right)\right]^{\frac{2}{5}}$	5	1,105
30	$\frac{6 \sin (a b)+\cos (b)}{8,1 \sin (b)+13,1 \cos (a-b)} e^{\ln \left(\frac{b}{3 a}\right)}$	7	1,15

Таблица 9

Номер варианта	$f(t)$	Значения		
		a	b	n
1	$f(t) = \begin{cases} \sin(t) & a \leq t \leq \pi/4, \\ \cos(t) & \pi/4 < t \leq b \end{cases}$	$-\pi$	π	51
2	$f(t) = \begin{cases} -1, & a \leq t \leq -1, \\ t, & -1 < t \leq 1, \\ 1, & 1 < t \leq b \end{cases}$	-3	3	51
3	$f(t) = \begin{cases} -\sin(t) & a \leq t \leq 0, \\ \sin(t) & 0 < t \leq b \end{cases}$	$-\pi$	$\pi/2$	41
4	$f(t) = \begin{cases} 1, & a \leq t \leq -\pi/2, \\ 0, & -\pi/2 < t \leq \pi/4, \\ -1, & \pi/4 < t \leq b \end{cases}$	$-\pi$	π	41
5	$f(t) = \begin{cases} \sin(t), & a \leq t \leq 0, \\ -\sin^2(t), & 0 < t \leq b \end{cases}$	$-\pi/2$	$\pi/2$	51
6	$f(t) = \begin{cases} 1-t, & a \leq t \leq 2, \\ 1, & 2 < t \leq 3, \\ 1+t, & 3 < t \leq b \end{cases}$	0	5	51
7	$f(t) = \begin{cases} 1-2\sin(t), & a \leq t \leq 0, \\ 1+\sin(2t), & 0 < t \leq b \end{cases}$	$-\pi$	π	61
8	$f(t) = \begin{cases} t, & a \leq t \leq 5, \\ t^2+1, & 5 < t \leq b \end{cases}$	0	10	61
9	$f(t) = \begin{cases} (\sin(t)-1)^2, & a \leq t \leq \pi/4, \\ -(\sin(t)+1)^2, & \pi/4 < t \leq b \end{cases}$	$-\pi/4$	π	66
10	$f(t) = \begin{cases} t^2, & a \leq t \leq 5, \\ (1-t)^2, & 5 < t \leq b \end{cases}$	0	10	66

Номер варианта	$f(t)$	Значения		
		a	b	n
11	$f(t) = \begin{cases} \sin(2t), & a \leq t \leq -3\pi/4, \\ -(\sin(t)), & -3\pi/4 < t \leq b \end{cases}$	$-\pi$	0	41
12	$f(t) = \begin{cases} t, & a \leq t \leq 2, \\ t^3, & 2 < t \leq b \end{cases}$	0	3	41
13	$f(t) = \begin{cases} 2 - 2\sin(t), & a \leq t \leq \pi, \\ -(2 - \sin(t)), & \pi < t \leq b \end{cases}$	0	2π	61
14	$f(t) = \begin{cases} t^2, & a \leq t \leq 3, \\ 1, & 3 < t \leq 4, \\ t^3, & 4 < t \leq b \end{cases}$	0	5	61
15	$f(t) = \begin{cases} 1 - \sin^2(t), & a \leq t \leq 0, \\ \sin^2(t), & 0 < t \leq b \end{cases}$	$-\pi$	π	51
16	$f(t) = \begin{cases} t, & a \leq t \leq 10, \\ 2t + t^2, & 10 < t \leq b \end{cases}$	0	20	51
17	$f(t) = \begin{cases} \sin(t), & a \leq t \leq 0, \\ 1 + \sin(2t), & 0 < t \leq b \end{cases}$	$-\pi$	π	41
18	$f(t) = \begin{cases} (t-1)^2, & a \leq t \leq 5, \\ (t+1)^3, & 5 < t \leq b \end{cases}$	0	10	41
19	$f(t) = \begin{cases} \sin^2(t), & a \leq t \leq 0, \\ -2\sin(2t), & 0 < t \leq b \end{cases}$	$-\pi/8$	$\pi/8$	51
20	$f(t) = \begin{cases} 2t+1, & a \leq t \leq 1, \\ t^2, & 1 < t \leq b \end{cases}$	0	2	51
21	$f(t) = \begin{cases} \sqrt{t^2-1}, & a \leq t \leq 5, \\ \sqrt{t^3+1}, & 5 < t \leq b \end{cases}$	2	10	41

Номер варианта	$f(t)$	Значения		
		a	b	n
22	$f(t) = \begin{cases} -\sin^2(t), & a \leq t \leq 0, \\ \cos(2t), & 0 < t \leq b \end{cases}$	$-\pi/4$	$\pi/4$	55
23	$f(t) = \begin{cases} (t-1)^{3/2}, & a \leq t \leq 10, \\ (t+1)^{5/2}, & 10 < t \leq b \end{cases}$	5	15	56
24	$f(t) = \begin{cases} 1.5t^2 - 1, & a \leq t \leq 5, \\ \sin(\pi - 1), & 5 < t \leq b \end{cases}$	3	11	61
25	$f(t) = \begin{cases} t, & a \leq t \leq 0, \\ (t-1)(t+1), & 0 < t \leq b \end{cases}$	-1	1	61
26	$f(t) = \begin{cases} 1-t^2, & a \leq t \leq 3, \\ 5, & 3 < t \leq 5, \\ 1+t^2, & 5 < t \leq b \end{cases}$	-1	10	51
27	$f(t) = \begin{cases} 1-2\cos(t), & a \leq t \leq 0, \\ 2+\sin(2t), & 0 < t \leq b \end{cases}$	-2π	2π	61
28	$f(t) = \begin{cases} t, & a \leq t \leq 4, \\ t^3 - 3, & 4 < t \leq b \end{cases}$	0	7	61
29	$f(t) = \begin{cases} (\cos(t)-2)^2, & a \leq t \leq \pi/4, \\ -(\sin(t)+1)^3, & \pi/4 < t \leq b \end{cases}$	$-\pi/2$	π	56
30	$f(t) = \begin{cases} 1.5t^2, & a \leq t \leq 3, \\ (2-t)^3, & 3 < t \leq b \end{cases}$	0	8	66

3. Научиться использовать итерационные циклические структуры. Вычислить значение функции $f(x)$ по вариантам табл. 10, используя для этого приближенные выражения (табл. 11) с точностью $\varepsilon = 0,01$. Суммирование членов ряда прекратить, если очередной член ряда, прибавляемый к сумме, будет меньше ε .

Таблица 10

Номер варианта	$F(x)$	x	Номер варианта	$F(x)$	x
1	$\left \frac{2^x + 2^{-x}}{2^x} \right $	6	12	$\left \frac{e^x}{e^x - e^{-x}} \right $	4
2	$\sqrt{\frac{2^x + 2^{-x}}{2^x}}$	6	13	$\left \frac{e^x}{e^x - e^{-3x}} \right $	2
3	$\left \frac{2^x - 2^{-x}}{2^x} \right $	7	14	$\left \frac{e^x}{e^x + e^{-x}} \right $	4
4	$\sqrt{\frac{2^x - 2^{-x}}{2^x}}$	7	15	$\left \frac{\sin x - 0,5}{\sin x} \right $	$\pi/2$
5	$\left \frac{2^x}{2^x - 2^{-x}} \right $	5	16	$\sqrt{\left \frac{\sin x - 0,75}{\sin x} \right }$	$\pi/2$
6	$\sqrt{\frac{2^x}{2^x - 2^{-x}}}$	5	17	$\left \frac{\sin x}{\sin x - 3} \right $	$\pi/2$
7	$\left \frac{2^x}{2^x + 2^{-2x}} \right $	4	18	$\sqrt{\left \frac{\sin x}{\sin x - 5} \right }$	$\pi/2$
8	$\left \frac{e^x + e^{-x}}{e^x} \right $	4	19	$\left \frac{\sin x + 1}{4 \sin x} \right $	$\pi/2$
9	$\sqrt{\frac{e^x + e^{-x}}{e^x}}$	4	20	$\sqrt{\left \frac{\sin x}{\sin x + 3} \right }$	$\pi/2$
10	$\left \frac{e^x - e^{-x}}{e^x} \right $	4	21	$\left \frac{\cos x - 0,5}{\cos x} \right $	π
11	$\sqrt{\frac{e^x - e^{-x}}{e^x}}$	4	22	$\sqrt{\left \frac{\cos x - 0,75}{\cos x} \right }$	π

Окончание табл. 10

Номер варианта	$F(x)$	x	Номер варианта	$F(x)$	x
23	$\sqrt{\frac{\cos x}{\cos x - 3}}$	π	27	$\left \frac{2^x}{2^x + 2^{-x}} \right $	6
24	$\sqrt{\frac{\cos x}{\cos x + 5}}$	π	28	$\left \frac{2e^x}{2e^x + e^{-x}} \right $	3
25	$\left \frac{\cos x + 1}{4 \cos x} \right $	π	29	$\left \frac{3e^x + e^{-x}}{3e^x} \right $	5
26	$\sqrt{\frac{\cos x}{\cos x + 8}}$	0	30	$\left \frac{0,5 - \sin x}{\sin x} \right $	$\pi/2$

Таблица 11

Функция	Приближенное выражение
$\sin z$	$\sum_{i=1}^{n+1} (-1)^{i-1} \frac{z^{2i-1}}{(2i-1)!}$
$\cos z$	$\sum_{i=1}^{n+1} (-1)^{i-1} \frac{z^{2i-2}}{(2i-2)!}$
Q^z	$\sum_{i=1}^{n+1} \frac{(z \ln Q)^{i-1}}{(i-1)!}$
e^z	$\sum_{i=1}^{n+1} \frac{z^{i-1}}{(i-1)!}$

4. Научиться вычислять значение определенного интеграла с использованием приближенного метода трапеций. В рабочем листе вычислить значение определенного интеграла по формуле:

$$\int_p^q f(t)dt = \Delta t \{0,5 f(p) + f(p + \Delta t) + f(p + 2\Delta t) + \dots + f(p + (n - 2)\Delta t) + 0,5 f(p + (n - 1)\Delta t)\},$$

где $\Delta t = (q - p)/(n - 1)$.

Аналитические выражения функций $f(t)$ приведены в табл. 12. Номер функции для каждого варианта N определяется по формуле $N - 6 * (\text{int}((N - 1)/6))$, p принять равным $-N/2$, q – равным N .

Таблица 12

Функция	1	2	3	4	5	6
$f(t)$	$\frac{1}{\sqrt{1+t^2}}$	$\frac{1}{t^2+1}$	$\sqrt{1+t+t^2}$	$\frac{1}{\sqrt{1+t+t^2}}$	$\sqrt{ 1+t }$	$\frac{1}{t^2+t+1}$

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

К заданию 1

а) Значения a и b для тестирования подобрать таким образом, чтобы значение y и все промежуточные легко проверялись.

б) Вычисление y производить посредством не менее чем трех операторов с получением промежуточных значений.

К заданию 2

а) Для реализации ветвления использовать логическую функцию **Если**.

б) При реализации вычислений в формуле использовать для хранения значений a и b именованные ячейки.

К заданию 3

а) В одном из столбцов поместить значения a .

б) Для реализации ветвления использовать логическую функцию **Если**.

К заданию 4

а) Рассчитать значения подынтегральной функции $f(t)$.

б) Построить график $f(t)$ на интервале интегрирования.

СОДЕРЖАНИЕ ОТЧЕТА

1. Задания (полные формулировки со всеми исходными данными по варианту и методическими указаниями).
2. Полученные алгоритмы решения задач.
3. Результаты решения задач, графики.
4. Выводы по работе.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие категории функций есть в Excel?
2. Чем отличается заполнение ячеек, содержащих данные различного типа?
3. Какие параметры есть у функций **Если**, **Корень**, **Степень**?
4. Как и какие имена можно присвоить ячейке?
5. Что такое итерационный коэффициент, как он вычисляется?
6. Какие основные блоки используются для построения блок-схем?

Практическое задание № 3

ФУНКЦИИ КОМПЛЕКСНОГО ПЕРЕМЕННОГО, МАТРИЦЫ, СЛАУ

КРАТКОЕ ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

Функция комплексной переменной – это функция, которую можно записать в виде

$$w = u(x, y) + v(x, y)i,$$

где $u(x, y)$ и $v(x, y)$ – две функции двух действительных переменных.

Функция $u(x, y)$ называется **действительной частью** функции, функция $v(x, y)$ называется **мнимой частью** функции.

Матрица – математический объект, записываемый в виде прямоугольной таблицы чисел (или элементов кольца) и допускающий алгебраические операции (сложение, вычитание, умножение) между ним и другими подобными объектами.

Система t линейных алгебраических уравнений с n неизвестными (или линейная система) в линейной алгебре – это система уравнений вида

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m. \end{cases}$$

Здесь x_1, x_2, \dots, x_n – неизвестные, которые надо определить; $a_{11}, a_{12}, \dots, a_{mn}$ – коэффициенты системы; b_1, b_2, \dots, b_m – свободные члены, которые предполагаются известными. Индексы коэффициентов системы обозначают номера уравнения (i) и неизвестного (j), при котором стоит этот коэффициент, соответственно.

ЗАДАНИЯ

1. Научиться вычислять функции комплексной переменной. По передаточной функции $W(s)$, заданной по вариантам в табл. 13 и 14, построить амплитудно-частотную характеристику $AЧХ(\omega)$ и ее график.

Таблица 13

Номер варианта	$W(s)$	Значения			
		K	T_1	T_2	T_3
1	1	4	0,1	0,01	0,8
2	2	5	0,2	0,02	0,7
3	3	6	0,3	0,03	0,6
4	4	7	0,4	0,04	0,5
5	5	8	0,5	0,05	0,9
6	6	9	0,6	0,06	0,3
7	1	10	0,7	0,07	0,2
8	2	4	0,8	0,08	0,1
9	3	5	0,9	0,09	0,2
10	4	6	0,1	0,09	0,3
11	5	7	0,2	0,08	0,4
12	6	8	0,3	0,07	0,5
13	1	9	0,4	0,06	0,6
14	2	10	0,5	0,05	0,7
15	3	4	0,6	0,04	0,8
16	4	5	0,7	0,03	0,9
17	5	6	0,8	0,02	0,9
18	6	7	0,9	0,01	0,8
19	1	8	0,1	0,09	0,7
20	2	9	0,2	0,08	0,6
21	3	10	0,3	0,07	0,5
22	4	4	0,4	0,06	0,4
23	5	5	0,5	0,05	0,3
24	6	6	0,6	0,04	0,2
25	1	7	0,7	0,03	0,2
26	2	4	0,7	0,04	0,8
27	3	5	0,8	0,03	0,8
28	4	6	0,9	0,02	0,7
29	5	7	0,1	0,08	0,6
30	6	8	0,2	0,07	0,5

Таблица 14

Номер варианта	$W(s)$	Номер варианта	$W(s)$
1	$\frac{K(T_1s+1)}{T_2s^2+T_3s+1}$	4	$\frac{K(T_1s+1)}{T_2s^2-T_3s-1}$
2	$\frac{K(T_1s+1)}{T_2s^2-T_3s+1}$	5	$\frac{K(T_1s+1)}{(T_2s+1)(T_3s+1)}$
3	$\frac{K(T_1s+1)}{T_2s^2+T_3s-1}$	6	$\frac{K(T_1s+1)}{(T_2s-1)(T_3s-1)}$

2. Научиться выполнять операции над матрицами и векторами. Для произвольной матрицы размером 5×5 выполнить операции из табл. 15.

Таблица 15

Номер варианта	Задание
1	Найти определитель матрицы и сложить его со всеми элементами матрицы
2	Транспонировать матрицу
3	Найти наибольшую сумму элементов столбца и сложить ее с матрицей
4	Найти наименьшее произведение элементов строк и умножить на него матрицу
5	Умножить матрицу на число и найти определитель новой матрицы
6	Найти наибольшую сумму элементов строки и сложить ее с матрицей
7	Найти наименьшую сумму элементов строки и сложить ее с матрицей
8	Умножить матрицу на произвольный вектор размерностью 5
9	Поменять местами 2 строку и 2 столбец и подсчитать определитель новой матрицы
10	Найти наибольшее произведение элементов строк и умножить на него матрицу
11	Найти наибольшее произведение элементов столбцов и сложить его с матрицей
12	Сложить матрицу с произвольным числом, не равным 0
13	Сложить последнюю строку с первым столбцом, умноженным на 2

Номер варианта	Задание
14	Найти наибольшую сумму элементов строки и поделить на нее матрицу
15	Найти наименьшее произведение элементов строки и поделить на него матрицу
16	Разделить матрицу на число и найти определитель новой матрицы
17	Вычесть из первого столбца последнюю строку
18	Поменять местами первый столбец и последнюю строку, умноженную на 2
19	Сложить каждый столбец с каждой строкой
20	Разделить матрицу на ее определитель
21	Найти сумму максимальных элементов строк и вычесть ее из матрицы
22	Найти произведение минимальных элементов столбцов и сложить его с матрицей
23	Найти произведение максимальных элементов строк и поделить на него матрицу
24	Найти сумму минимальных элементов столбцов и умножить на нее матрицу
25	Умножить матрицу на ее определитель
26	Поменять местами последний столбец и первую строку, умноженную на 5
27	Вычесть из последнего столбца первую строку
28	Сложить матрицу с числом и найти определители обеих матриц
29	Поменять местами 3 строку и 3 столбец и подсчитать определитель новой матрицы
30	Найти наименьшую сумму элементов строки и умножить на нее матрицу

3. Научиться решать квадратные уравнения. Для произвольного квадратного уравнения с действительными некрратными корнями определить его корни.

4. Научиться решать системы линейных алгебраических уравнений. Для произвольной системы из трех уравнений с тремя неизвестными определить решение.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

К заданию 1

1. Записать $W(s)$ как функцию комплексного аргумента, заменив s на $(0 + j\omega)$.

2. В диапазоне частот $\omega \in [0 \text{ p/c}, 100 \text{ p/c}]$ с шагом дискретизации по частоте $\Delta\omega = 2$ определить $\text{Re}\{W(j\omega)\}$ и $\text{Im}\{W(j\omega)\}$, где Re – реальная, Im – мнимая части комплексного числа.

3. Вычислить $ACH(\omega) = |W(j\omega)| = \sqrt{[\text{Re}\{W(j\omega)\}]^2 + [\text{Im}\{W(j\omega)\}]^2}$.

4. Для вычислений квадрата, суммы, произведения и деления использовать комплексные функции, которые находятся в «Мастере функций» в категории «Инженерные».

К заданию 2

а) Использовать матричные операции из категории «Математические».

б) Для завершения матричных операций использовать <Ctrl + Shift + Enter>.

К заданию 3

а) Решение производить посредством команды «Поиск решения».

б) В отдельных ячейках задать начальные условия (значения) решения.

К заданию 4

а) Решение производить посредством команды «Поиск решения».

б) В отдельных ячейках задать начальные условия (значения) решения.

в) Формулы для каждого уравнения поместить в отдельные ячейки.

СОДЕРЖАНИЕ ОТЧЕТА

1. Задания (полные формулировки со всеми исходными данными по варианту и методическими указаниями).

2. Полученные алгоритмы решения задач.

3. Результаты решения задач, график.

4. Выводы по работе.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие функции комплексной переменной есть в Excel?

2. Какие матричные функции есть в Excel?

3. Каковы особенности применения матричных функций?
4. Какие параметры есть в модуле «Поиск решения»?
5. Какие условия можно использовать в ограничениях модуля «Поиск решения»?
6. Как решить квадратное уравнение с помощью модуля «Поиск решения»?
7. Как решить систему уравнений с помощью модуля «Поиск решения»?

Практическое задание № 4

ЛИНЕЙНЫЕ, РАЗВЕТВЛЯЮЩИЕСЯ, ЦИКЛИЧЕСКИЕ АЛГОРИТМЫ В VBA

ЗАДАНИЯ


1. Научиться реализовывать линейный алгоритм на языке программирования VBA. Выполнить задание 1 из лабораторной работы № 2.

2. Научиться реализовывать разветвляющиеся алгоритмические структуры на языке программирования VBA. Выполнить задание 2 из лабораторной работы № 2.

3. Научиться реализовывать итерационные циклические структуры на языке программирования VBA. Выполнить задание 3 из лабораторной работы № 2.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

К заданию 1

а) Нажав кнопку  в пиктографическом меню, перейти в редактор Visual Basic;

б) Создать новый модуль при помощи команды «Insert/Module»;

в) Описать типы всех используемых переменных;

г) Следовать методическим указаниям задания 1 из работы № 2;

д) Выполнения программы контролировать с помощью пошагового отладчика (клавиша *F8*), наблюдая значения переменных в окне отладки.

К заданию 2

а) Следовать методическим указаниям задания 2 из работы № 2;

б) Следовать методическим указаниям задания 1 из данной работы.

К заданию 3

а) Следовать методическим указаниям задания 3 из работы № 2;

б) Следовать методическим указаниям задания 1 из данной работы.

СОДЕРЖАНИЕ ОТЧЕТА

1. Задания (полные формулировки со всеми данными исходными и методическими указаниями из работы № 2).

2. Листинги программ.

3. Результаты решения задач, графики.

4. Выводы по работе.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие типы данных есть в языке VBA?
2. Каковы правила именования переменных в языке VBA?
3. Какие есть способы ввода/вывода в языке VBA?
4. Каковы варианты записи условного оператора в языке VBA?
5. Как обозначаются операции отношения в языке VBA?
6. Какие есть аддитивные и мультипликативные операции в языке VBA?
7. Какие типы циклов есть в языке VBA?
8. Какие есть способы досрочного выхода из циклов?
9. Синтаксис вложенных циклов в языке VBA.

Практическое задание № 5

ФУНКЦИИ КОМПЛЕКСНОГО ПЕРЕМЕННОГО И МАТРИЦЫ В VBA

ЗАДАНИЯ

1. Научиться вычислять функции комплексного переменного на языке программирования VBA. Выполнить задание 1 из лабораторной работы № 3.

2. Научиться выполнять операции над матрицами и векторами на языке программирования VBA. Выполнить задание 2 из лабораторной работы № 3.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

К заданию 1

а) Следовать методическим указаниям задания 1 из работ № 3, 4;

К заданию 2

а) Следовать методическим указаниям задания 2 из работы № 3;

б) Следовать методическим указаниям задания 1 из работы № 4.

СОДЕРЖАНИЕ ОТЧЕТА

1. Задания (полные формулировки со всеми исходными данными из работы № 3 и методическими указаниями из работ № 3, 4).

2. Листинги программ.

3. Результаты решения задач, график.

4. Выводы по работе.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Описание массивов различных типов.

2. Задание нижнего и верхнего пределов массива.

3. Динамические массивы и изменение их параметров.

4. Способы организации циклов с неизвестным числом повторений.

5. Циклы с пред- и постпроверкой условия в языке VBA.

6. Массивы и вложенные циклы.

7. Что такое бесконечный цикл и как его избежать?

8. Значение переменной цикла при нормальном и досрочном его завершении.

Практическое задание № 6

СТРОКОВЫЕ ДАННЫЕ. ПОДПРОГРАММЫ-ФУНКЦИИ

ЗАДАНИЯ

1. Научиться операциям со строковыми данными и использованию подпрограммы-функции. Решить задачи по вариантам табл. 16.

Таблица 16

Номер варианта	Задание
1	Заменить все вхождения буквы «М» на «Л»
2	Заменить последнюю букву на первую
3	Поменять местами первую и последнюю буквы
4	Удалить все знаки «+» и «-»
5	Заменить последовательность символов «oil» на «gas»
6	Два последних символа поменять местами
7	Заменить все нечетные символы на буквы «А»
8	Удалить все пробелы, заменив их нулями
9	Поменять местами соседние буквы (попарно)
10	Удалить все символы «!» и «?»
11	Удалить все четные символы
12	Удалить все буквы «а»
13	После каждой буквы добавить «да»
14	Заменить последовательность символов «America» на «Africa»
15	Добавить в конец первую букву столько раз, сколько букв в слове
16	Заменить буквы <i>A, B, C, D</i> на цифры 1, 2, 3, 4 соответственно
17	Два соседних одинаковых символа заменить на пробелы
18	Заменить каждый третий символ на последний
19	Заменить цифры 1, 2 на буквы <i>A, B</i>
20	Удалить все вхождения последней буквы
21	Каждую вторую букву продублировать
22	Три первые буквы поменять местами с тремя последними
23	Заменить три последние буквы на буквы <i>abc</i>
24	Все нули заменить на единицы
25	Удалить все нечетные символы
26	Исправить все многократные пробелы на одиночные
27	Удалить все пробелы, стоящие перед запятой или точкой
28	Добавить перед символами «!» символ «?»
29	Каждую третью букву продублировать
30	Удалить каждый третий символ

2. Научиться использовать текстовые функции VBA. Повторить задание 1, используя программу на VBA.

3. Научиться создавать пользовательские функции. Повторить задание 2, используя вызов подпрограммы-функции на VBA из мастера функций (категория «Определенные пользователем»).

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

К заданию 1

а) Для решения задачи использовать встроенные функции из категории «Текстовые».

б) Результат решения разместить в ячейки рабочего листа.

К заданию 2

а) Вставив модуль VBA, ввести текст подпрограммы с обязательным описанием типов данных;

б) Для решения задачи использовать функции VBA, аналогичные функциям мастера функций из категории «Текстовые»;

в) Для вывода результата решения в последовательности ячеек рабочего листа использовать подпрограмму-процедуру.

К заданию 3

а) Переписать текст подпрограммы-процедуры *Sub* задания 2 в виде подпрограммы-функции;

б) Решить задание на рабочем листе, используя вызов подпрограммы-функции на VBA из мастера функций (категория «Определенные пользователем»).

СОДЕРЖАНИЕ ОТЧЕТА

1. Задания (полные формулировки со всеми исходными данными по варианту и методическими указаниями).

2. Листинги программ.

3. Тестовые примеры, результаты.

4. Выводы по работе.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие текстовые функции имеются в Excel?

2. Какие есть способы описания строковых данных?

3. Классификация функций для работы со строковыми данными.

4. Функции сравнения строк.
5. Функции преобразования строк.
6. Функции работы с пробелами.
7. Функции обработки строк.
8. Функции работы с кодами ASCII.
9. Синтаксис подпрограммы-функции.
10. Вызов функции VBA в рабочий лист Excel.

Практическое задание № 7

СТРОКОВЫЕ МАССИВЫ. ПОДПРОГРАММЫ-ПРОЦЕДУРЫ

ЗАДАНИЯ

1. Научиться оперировать со строковыми массивами. Записать в последовательность ячеек рабочего листа набор строк (слов). Получить новый набор строк в соответствии с вариантом табл. 17.

Таблица 17

Номер варианта	Задание
1	Вывести все строки наименьшей длины
2	Вывести строки, в которых нет повторения символов
3	Вывести строки, в которых буква «а» встречается дважды
4	Вывести строки, которые начинаются и заканчиваются одинаковым символом
5	Вывести строки, которые не содержат цифр
6	Вывести строки, которые не содержат буквы «м»
7	Вывести строки, которые начинаются с того же символа, что и следующая строка
8	Вывести строки, перед которыми находятся строки с меньшей по алфавиту буквы
9	Вывести строки, у которых одинаковые первые символы
10	Вывести строки, у которых есть хотя бы одна буква «а», стоящая рядом с «м»
11	Вывести все строки в обратном порядке
12	Вывести все строки, поменяв крайние символы местами
13	Удалить во всех строках букву «а», позиции справа заполнить запятыми
14	Заменить во всех строках каждое вхождение буквы «х» на «ks»
15	Вывести все строки максимальной длины
16	Вывести строки без повторения символов
17	Оставить во всех строках только первые вхождения каждого символа
18	Вывести строки, в которые каждый символ входит более одного раза
19	Вывести строки, в которых первый символ встречается более одного раза

Номер варианта	Задание
20	Во всех строках удалить предыдущие вхождения последнего символа
21	Вывести в обратном порядке строки длиной не менее 4 символов
22	В каждой строке, где есть буква «а», добавить после нее «да»
23	Все вхождения наборов букв «abc» заменить на «def» и вывести слова в обратном порядке
24	Вывести строки, в которых буквы упорядочены по алфавиту
25	Вывести строки нечетной длины, удалив в них средний символ
26	Вывести строки, в которых есть и прописные, и строчные буквы
27	Вывести строки, в которых есть и русские, и латинские буквы
28	Вывести строки, в которых не менее двух цифр, и они упорядочены по возрастанию
29	Вывести строки, в которых есть только русские и латинские буквы
30	Вывести строки, которые не содержат ничего кроме букв и цифр

2. Модифицировать программу, написанную для задания 1 в соответствии с методическими указаниями.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

К заданию 1

- а) Для решения задачи использовать встроенные функции VBA для работы с текстом;
- б) Результат решения разместить в соседней последовательности ячеек рабочего листа.

К заданию 2

- а) Вводить данные с рабочего листа в головной программе;
- б) Преобразование строки производить в подпрограмме-процедуре;
- в) Результат решения выводить в соседнюю последовательность ячеек рабочего листа из головной программы.

СОДЕРЖАНИЕ ОТЧЕТА

1. Задания (полные формулировки со всеми исходными данными по варианту и методическими указаниями).
2. Листинги программ.
3. Тестовые примеры, результаты.

4. Выводы по работе.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Правила описания строковых массивов.
2. Особенности обработки строковых массивов.
3. Синтаксис подпрограммы-процедуры.
4. Сравнение подпрограмм функций и процедур.
5. Досрочный выход из подпрограмм.
6. Рекурсивный вызов подпрограмм.
7. Способы передачи и возврата параметров в функцию.
8. Способы передачи и возврата параметров в процедуру.
9. Подпрограммы ввода/вывода в рабочий лист Excel.

Практическое задание № 8




ВВОД/ВЫВОД В ФАЙЛЫ. ЭЛЕМЕНТЫ УПРАВЛЕНИЯ

ЗАДАНИЕ

Научиться оперировать с текстовыми файлами и использовать элементы управления. Прочитать из текстового файла набор слов, обработать в подпрограмме-процедуре в соответствии с вариантом табл. 17 и вывести в другой текстовый файл.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

а) Записать в текстовый файл с помощью программы «Блокнот» набор строк;

б) на панели инструментов VBA выбрать кнопку пиктографического меню  («Элементы управления»). Выбрать элемент  – «Кнопка» и поместить его в рабочий лист Excel. Перейти в режим конструктора, нажав кнопку . В режиме конструктора два раза щелкнуть по элементу «Кнопка», после чего откроется редактор Visual Basic;

в) написать процедуру Sub, считывающую данные из созданного текстового файла;

г) создать еще три кнопки и написать три соответствующие им процедуры, каждая из которых соответственно обрабатывает данные из текстового файла по вариантам табл. 17, записывает результат решения в другой текстовый файл, очищает содержимое рабочего листа;

д) продемонстрировать результат работы программ с помощью программы «Блокнот».

СОДЕРЖАНИЕ ОТЧЕТА

1. Задания (полные формулировки со всеми исходными данными по варианту и методическими указаниями).
2. Листинги программ.
3. Содержимое исходного файла и файла-результата.
4. Выводы по работе.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Понятие файла. Типы файлов.
2. Инструкции открытия и закрытия файлов.
3. Режимы открытия файлов и способы доступа к ним.

4. Функции чтения/записи в файл.
5. Копирование, переименование, удаление файлов.
6. Элементы управления. Кнопки управления.

4. РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ

Определить значение функционала F по вариантам табл. 18.

$$F = \left(\int_a^b f(t) dt - \int_a^d f(t) dt \right) / \left(\int_b^c f(t) dt + \int_b^d f(t) dt \right).$$

Таблица 18

Номер варианта	Порядковый номер варианта		
	для $f(t)$	для a и b	для c и d
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	1	6	6
8	2	5	5
9	3	4	4
10	4	3	3
11	5	2	2
12	6	1	1
13	6	2	1
14	5	3	2
15	4	4	3
16	3	5	4
17	2	6	5
18	1	2	6
19	1	3	1
20	2	1	2
21	3	6	3
22	4	6	4
23	5	6	5
24	6	5	6
25	6	4	3
26	5	3	4
27	4	2	5
28	3	1	6
29	2	1	5
30	1	5	4

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Пределы интегрирования a , b , c , d вычислять в подпрограмме *Sub* по вариантам табл. 19 и 20.

Для вычисления определенного интеграла по квадратурной формуле (см. работу № 2) использовать подпрограмму *Sub*. Подынтегральные функции $f(t)$ приведены в табл. 12.

СОДЕРЖАНИЕ ОТЧЕТА

1. Задание (полная формулировка со всеми исходными данными по варианту и методическими указаниями).
2. Листинги программ.
3. Результаты вычисления пределов интегрирования a , b , c , d , четырех интегралов, функционала F ; графики подынтегральной функции в пределах интегрирования.
4. Выводы по работе.

Таблица 19

Номер варианта	Параметр		Функция $\varphi(p, q)$	x_1	x_2	y_1	y_2
	a	b					
1	$\frac{\sqrt[3]{x_1^2 + y_1^2}}{\sqrt[3]{x_2^2 + y_2^2} + \sqrt[3]{x_1^2 + y_1^2}}$	$\frac{10}{\sqrt[3]{x_1^2 + x_2^2} + \sqrt[3]{y_1^2 + y_2^2}}$	$\sqrt[3]{p^2 + q^2}$	1	1	2	2
2	$\frac{\log_2(x_1) + \log_4(y_1)}{\log_6(x_1 + y_1)}$	$\frac{16}{2 \log_{10}(x_2 + y_2)}$	$\log_q(p) = \frac{\ln(p)}{\ln(q)}$	2	4	4	6
3	$\frac{(e^{x_1})^2}{2e^{x_1 - y_1} + \sqrt{e^{x_2 - y_2}}}$	$\frac{18}{e^{y_1 - y_2} + \sqrt{e^{x_1 - x_2}}}$	$e^{p - q}$	2	2	1	1
4	$\frac{(\ln(x_1) + \ln(y_1))^2}{\sqrt{\ln(x_2) + \ln(y_2)}}$	$\frac{14}{\sqrt{\ln(x_1) + \ln(x_2)} + \sqrt{\ln(y_1) + \ln(y_2)}}$	$\ln(p) + \ln(q)$	e	e	e	e
5	$\frac{\sin(x_1) + \cos(y_1)}{\sqrt{\sin(x_2) + \cos(y_2)}}$	$\frac{10}{2(\sin(x_1) + \cos(y_2))}$	$\sin(p) + \cos(q)$	0	0	0	0
6	$\frac{2(x_1^2 + \sqrt{y_1})}{\sqrt[3]{x_2^2 + \sqrt{y_2}}}$	$\frac{12}{\sqrt[3]{x_1^2 + \sqrt{y_2}}}$	$p^2 + \sqrt{q}$	1	1	1	1

Таблица 20

Номер варианта	Параметр		Приближенное значение	x	y
	c	d			
1	$\frac{3}{\sin(x)}$	$8 \sin(y)$	$\sin(z) \approx \sum_{i=1}^{n+1} (-1)^{i-1} \frac{z^{2i-1}}{(2i-1)!}, \varepsilon = 10^{-2}$	$\pi/6$	$\pi/2$
2	$\frac{10}{\sqrt{\cos(x)}}$	$6[1 - \cos(\pi + y)]$	$\cos(z) \approx \sum_{i=1}^{n+1} (-1)^{i-1} \frac{z^{2i-2}}{(2i-2)!}, \varepsilon = 10^{-2}$	$\pi/3$	$\pi/3$
3	$\frac{20}{\sqrt{2^x}}$	$\frac{36}{\sqrt{3^y}}$	$Q^z \approx \sum_{i=1}^{n+1} \frac{(z \ln Q)^{i-1}}{(i-1)!}, \varepsilon = 10^{-2}$	2	2
4	$2\sqrt{e^x}$	$\frac{20}{e^y}$	$e^z \approx \sum_{i=1}^{n+1} \frac{z^{i-1}}{(i-1)!}, \varepsilon = 10^{-2}$	2	1
5	$\frac{7}{1 - \sin(x)}$	$10 - \sin(y)$	$\sin(z) \approx \sum_{i=1}^{n+1} (-1)^{i-1} \frac{z^{2i-1}}{(2i-1)!}, \varepsilon = 10^{-2}$	$\pi/3$	$\pi/2$
6	$\frac{14}{\sqrt{\cos(x)}}$	$16\sqrt{\cos(y)}$	$\cos(z) \approx \sum_{i=1}^{n+1} (-1)^{i-1} \frac{z^{2i-2}}{(2i-2)!}, \varepsilon = 10^{-2}$	$\pi/6$	$\pi/3$

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Волченков Н.Г. Учимся программировать: VB-5: учеб. пособие. – М.: Диалог МИФИ, 1998.
2. Сока Джон, Рассмел Дэн, Комл Дебра. VB 5.0. – Минск, 1998.
3. Гуревич Н., Гуревич О. Освой самостоятельно VB 5. – М.: Бином, 1998.
4. Санна Пол. Visual Basic для приложений (VBA) 5 в подлиннике. Серия «...В ПОДЛИННИКЕ» / пер. с англ. – СПб.: «ВНУ-Санкт-Петербург», 1997.
5. Король В.И. Visual Basic 6.0, Visual Basic for Applications 6.0. Язык программирования: справочник с примерами. – М.: КУДИЦ, 2000.
6. Васильев А., Андреев А. VBA в Office 2000: учебный курс. – СПб.: Питер, 2001.
7. Биллиг В.А. Средства разработки VBA-программиста. Офисное программирование. Том 1. – М.: Издательско-торговый дом «Русская Редакция», 2001.
8. Слепцова Л.Д. Программирование на VBA в MS Office 2010. – М.; СПб.; Киев: Диалектика, 2010. – 433 с.
9. Джон Уокенбах. Microsoft Excel 2010. Профессиональное программирование на VBA. – М.; СПб.; Киев: Диалектика, 2010. – 920 с.

**Саблина Галина Владимировна
Худяков Дмитрий Сергеевич**

ИНФОРМАТИКА

Учебно-методическое пособие

Редактор *И.Л. Кескевич*
Выпускающий редактор *И.П. Брованова*
Корректор *И.Е. Семенова*
Дизайн обложки *А.В. Ладыжская*
Компьютерная верстка *Н.В. Гаврилова*

Налоговая льгота – Общероссийский классификатор продукции
Издание соответствует коду 95 3000 ОК 005-93 (ОКП)

Подписано в печать 15.09.2020. Формат 60 × 84 1/16. Бумага офсетная
Тираж 150 экз. Уч.-изд. л. 4,41. Печ. л. 4,75. Изд. 374/16. Заказ № 438
Цена договорная

Отпечатано в типографии
Новосибирского государственного технического университета
630073, г. Новосибирск, пр. К. Маркса, 20