

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
"БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА"

Кафедра "Информационные технологии"

ИНФОРМАТИКА

**Практикум по выполнению лабораторных
и самостоятельных работ студентов
технических специальностей I курса ФБО**

Гомель 2010

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
"БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА"

Кафедра "Информационные технологии"

ИНФОРМАТИКА

Практикум по выполнению лабораторных
и самостоятельных работ студентов
технических специальностей I курса ФБО

*Одобен методической комиссией
факультета безотрывного обучения*

Гомель 2010

УДК 002.3(075.8)
ББК 32.81
И74

Авторы: Т. А. Голдобина, С. В. Дрючкова,
А. П. Кейзер, Ю. А. Пшеничнов.

Рецензент – канд. техн. наук, доцент кафедры "Прикладная математика"
С. И. Жогаль (УО "БелГУТ").

И74 **Информатика** : практикум по выполнению лабораторных и самостоятельных работ студентов техн. спец. I курса ФБО / Т. А. Голдобина [и др.] ; М-во образования Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель : БелГУТ, 2010. – 182 с.
ISBN 978-985-468-668-4

Приведены краткие теоретические сведения, примеры, индивидуальные задания и контрольные вопросы по темам дисциплины, которые помогут в ходе выполнения лабораторных, контрольных и курсовых работ, при подготовке к зачетам и экзаменам. Практикум включает образец оформления отчетов по лабораторным работам и содержание учебного материала по дисциплине "Информатика".

Предназначен для студентов I курса ФБО технических специальностей.

УДК 002.3(075.8)
ББК 32.81

ISBN 978-985-468-668-4

© Оформление. УО «БелГУТ», 2010

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА 1. Устройство персонального компьютера.	
Операционная система <i>Microsoft Windows. Total Commander</i>	6
1 Сведения из теории.....	6
1.1 Устройство ПК.....	6
1.1.1 Базовая аппаратная конфигурация ПК.....	6
1.1.2 Состав клавиатуры ПК	6
1.1.3 Включение и выключение ПК. Правила работы в компьютерных аудиториях ВЦ БелГУТа.....	8
1.2 Операционная система <i>Windows</i>	8
1.2.1 Основные понятия.....	8
1.2.2 Структура и настройка окна	10
1.2.3 Основные приемы работы с объектами	11
1.2.4 Некоторые настройки ОС <i>Windows</i>	12
1.3 Файловый менеджер <i>Total Commander</i>	12
2 Порядок выполнения лабораторной работы.....	15
Контрольные вопросы	18
САМОСТОЯТЕЛЬНАЯ РАБОТА 1. Текстовый процессор <i>Word</i>.....	19
1 Сведения из теории.....	19
1.1 Основные возможности <i>Word</i>	19
1.1.1 Основные команды для работы с файлами документов.....	19
1.1.2 Настройка внешнего вида окна приложения <i>Word</i> и параметров документов	20
1.1.3 Набор и редактирование текста	22
1.1.4 Форматирование шрифтов и абзацев	25
2 Порядок оформления контрольной работы.....	29
Контрольные вопросы	31
ЛАБОРАТОРНАЯ РАБОТА 2. Язык <i>Pascal</i>. Программирование линейных и разветвляющихся алгоритмов. Циклы.....	32
1 Сведения из теории.....	32
1.1 Краткое описание среды <i>Free Pascal</i>	33
1.1.1 Запуск и обзор команд главного меню.....	33
1.1.2 Создание, открытие и сохранение файла документа. Работа с окнами	36
1.1.3 Набор и редактирование текста программы.....	36
1.1.4 Компиляция программы. Некоторые ошибки.....	37
1.1.5 Выполнение программы. Просмотр результатов	38
1.1.6 Завершение работы в <i>Free Pascal</i>	38
1.2 Базовые понятия языка <i>Pascal</i>	38
1.2.1 Алфавит языка программирования <i>Pascal</i>	38
1.2.2 Словарь языка программирования <i>Pascal</i>	39
1.2.3 Структура простейшей программы.....	40
1.2.4 Константы, переменные, выражения	41
1.2.5 Арифметические и логические операции.....	42
1.2.6 Данные в программах. Типы данных	43
1.3 Программирование линейных алгоритмов	46
1.3.1 Операторы языка программирования <i>Pascal</i>	46

1.3.2 Основные стандартные процедуры и функции	50
1.3.3 Запись алгебраических выражений.....	51
1.4 Программирование разветвляющихся алгоритмов.....	54
1.4.1 Логические выражения.....	54
1.4.2 Оператор условного перехода <i>if</i> . Оператор выбора <i>case</i>	55
1.5 Организация циклических вычислений.....	58
1.5.1 Циклы с неизвестным числом повторений (итерационные циклы).....	59
1.5.2 Цикл с предусловием <i>while..do</i>	59
1.5.3 Цикл с постусловием <i>repeat..until</i>	60
1.5.4 Цикл с параметром <i>for..to..do</i>	62
2 Порядок выполнения лабораторной работы.....	63
Контрольные вопросы	72
САМОСТОЯТЕЛЬНАЯ РАБОТА 2. Обработка массивов на языке <i>Pascal</i>.	
Процедуры и функции пользователя.....	74
1 Сведения из теории.....	74
1.1 Структурированный тип данных массив	74
1.1.1 Описание массивов.....	74
1.1.2 Ввод элементов массива	76
1.1.3 Вывод элементов массива.....	77
1.1.4 Базовые алгоритмы обработки массивов	78
1.1.5 Решение задач обработки одномерных массивов.....	81
1.2 Процедуры и функции	85
Контрольные вопросы	90
ЛАБОРАТОРНАЯ РАБОТА 3. <i>Mathcad</i>. Базовые операции. Ранжированные переменные. Функции. Графики.....	91
1 Сведения из теории.....	91
1.1 Основы работы в системе <i>Mathcad</i>	91
1.1.1 Создание и особенности работы в формульной области.....	92
1.1.2 Создание и особенности работы в текстовой области.....	93
1.1.3 Переменные. Предопределенные переменные.....	94
1.1.4 Форматирование результатов	95
1.2 Ранжированные (дискретные) переменные	95
1.3 Функции.....	96
1.4 Графики функций	97
2 Порядок выполнения лабораторной работы.....	98
Контрольные вопросы	106
ЛАБОРАТОРНАЯ РАБОТА 4. Массивы в <i>Mathcad</i>. Решение уравнений и систем уравнений.....	108
1 Сведения из теории.....	108
1.1 Массивы.....	108
1.1.1 Основные определения.....	108
1.1.2 Создание массивов	109
1.1.3 Шаблоны для работы с векторами и матрицами	115
1.1.4 Функции для работы с векторами и матрицами	116
1.1.5 Вычисление сумм и произведений элементов массивов	117
1.1.6 Решение задач обработки массивов в <i>Mathcad</i>	118

1.2	Решение уравнений и систем уравнений средствами <i>Mathcad</i>	124
1.2.1	Функция <i>root</i> , блоки <i>Given...Find</i> и <i>Given...Minerr</i>	124
1.2.2	Решение полиномиальных уравнений. Функция <i>polyroots</i>	133
1.2.3	Решение систем линейных уравнений	134
2	Порядок выполнения лабораторной работы.....	139
	Контрольные вопросы	144
	ЛАБОРАТОРНАЯ РАБОТА 5. Табличный процессор <i>Excel</i>	146
1	Сведения из теории.....	146
1.1	Основные возможности <i>Excel</i>	146
1.1.1	Базовые понятия и приемы работы в <i>Excel</i>	146
1.1.2	Некоторые приемы работы в <i>Excel</i>	152
1.1.3	Защита данных в ячейках	153
1.1.4	Формулы и функции. Решение задач.....	154
1.1.5	Графические возможности <i>Excel</i> . Построение графика функции.....	161
1.2	Решение уравнений и систем линейных уравнений	164
1.2.1	Решение уравнений в <i>Excel</i>	164
1.2.2	Решение систем линейных уравнений в <i>Excel</i>	168
2	Порядок выполнения лабораторной работы.....	171
	Контрольные вопросы	177
	Список рекомендуемой литературы	178
	Приложение А Пример оформления отчета по лабораторной работе.....	180
	Приложение Б Содержание учебного материала по дисциплине «Информатика»	181
	Приложение В Некоторые процедуры-функции на языке Pascal.....	184
	Приложение Г Выполнение лабораторной работы в среде Turbo Pascal	185

ЛАБОРАТОРНАЯ РАБОТА 1
**УСТРОЙСТВО ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА.
ОПЕРАЦИОННАЯ СИСТЕМА MICROSOFT WINDOWS.
TOTAL COMMANDER**

Цель работы: получить первичные знания об устройстве персонального компьютера и базовые навыки работы в операционной системе *Windows*, ознакомиться с возможностями файлового менеджера *Total Commander*.

1 Сведения из теории

1.1 Устройство ПК

1.1.1 Базовая аппаратная конфигурация ПК

В настоящее время в *базовой аппаратной конфигурации* персонального компьютера (ПК) выделяют **четыре основные части:**

- 1) системный блок;
- 2) клавиатура – для ввода информации;
- 3) монитор (дисплей) – для вывода информации во время работы ПК;
- 4) компьютерная мышь.

Системный блок представляет собой основной блок ПК, внутри которого установлены наиболее важные компоненты:

- электронные схемы – процессор, память, контроллеры и др.;
- жесткий диск (HDD);
- видеокарта;
- оптические приводы (DVD-R/RW и т. д.);
- блок питания компьютера от электрической сети.

К системному блоку можно подключать различные *дополнительные устройства*, среди которых чаще других встречаются принтер, сканер, модем.

Монитор – устройство визуального представления информации, являющееся основным устройством вывода. Его главными параметрами являются: размер по диагонали, шаг маски экрана, максимальная частота регенерации изображения, класс защиты.

Мышь – устройство управления ПК манипуляторного типа.

1.1.2 Состав клавиатуры ПК

Клавиатура является стандартным устройством ввода числовой и текстовой информации, а также команд управления ПК.

Условно все клавиши клавиатуры можно разбить на 5 групп.

Группа 1 – *алфавитно-цифровые* и *знаковые* клавиши, расположенные в центре клавиатуры и позволяющие набирать тексты и команды.

Группа 2 – клавиши *служебного назначения* (таблица 1).

Таблица 1 – Основные служебные клавиши клавиатуры

Название	Символ	Назначение
Esc		Отмена действия
Tab	⇄	Переход к следующей позиции табуляции
Caps Lock		Фиксация смены регистра (жесткое переключение между строчными и прописными буквами)
Shift	⇧	Смена регистра. Использование в комбинации с другими клавишами
Ctrl		Использование в комбинации с другими клавишами
Alt		Использование в комбинации с другими клавишам
Enter	↵	Подтверждение ввода команды. Создание нового абзаца
Delete	Del	Удаление выделенного объекта. Удаление символа справа от курсора
Backspace	←	Удаление символа слева от курсора
Print Screen		Копирование изображения на экране монитора в буфер обмена
Num Lock		Включение (отключение) дополнительной цифровой клавиатуры

Группа 3 – клавиши *управления курсором* (таблица 2).

Таблица 2 – Основные клавиши управления курсором

Обозначение	Назначение
↑	Перемещение курсора на один символ вверх (на строку вверх)
↓	Перемещение курсора на один символ вниз (на строку вниз)
←	Перемещение курсора на один символ влево
→	Перемещение курсора на один символ вправо
Home	Перемещение курсора в начало текущей строки
End	Перемещение курсора в конец текущей строки
Page Up	Страница (видимая на экране) вверх
Page Down	Страница (видимая на экране) вниз
Ins (Insert)	Переключение режимов вставки и замены символов

Группа 4 – *функциональные* клавиши **F1–F12**. Обычно **F1** – Помощь.

Группа 5 – *дополнительная цифровая клавиатура*, которая удобна при наборе цифровой информации. Для активизации этой клавиатуры используется служебная клавиша **Num Lock**.

1.1.3 Включение и выключение ПК. Правила работы в компьютерных аудиториях ВЦ БелГУТа

В компьютерных аудиториях ВЦ БелГУТа компьютеры обычно находятся в режиме ожидания, поэтому для начала работы достаточно войти в систему под определенным именем (для студентов ФБО *имя пользователя* **zf**, *пароль* вводить **не надо!**).

Итак, чтобы начать работу на компьютере в аудитории ВЦ БелГУТа, нажмите комбинацию клавиш **Ctrl Alt Del**, удерживайте их одновременно нажатыми не менее 1–2 секунд. Отобразится диалоговое окно с приглашением ввода *имени пользователя* (**User Name**) и *пароля* (**Password**). Установите курсор в поле **Пользователь** (User Name) и введите **zf** (строчными латинскими буквами). *Пароль* вводить **не надо!** Нажмите клавишу **Enter** или щелкните мышью на кнопке **ОК**. На экране монитора через некоторое время появится рабочий стол операционной системы *Windows*.

Для завершения работы следует закрыть все приложения и выполнить команду **Start / Log Off** (*Пуск / Смена пользователя*).

1.2 Операционная система *Windows*

1.2.1 Основные понятия

После загрузки операционной системы *Microsoft Windows* на экране появляется рабочий стол (Desktop) – главная системная папка, представляющая собой видимое пространство экрана, на котором работает пользователь. На рабочем столе обычно расположены пиктограммы (ярлыки) системных папок, часто используемых пользователем программ и приложений.

Основные *системные* папки:

- **My computer** (*Мой компьютер*) – отражает содержимое компьютера целиком;

- **Recycle Bin** (*Корзина*) – для хранения удаленных объектов. Средства папки используются для восстановления удаленных объектов или их окончательного удаления;

- **Network Support** (*Сетевое окружение*) – информация о структуре сети, к которой подключен данный компьютер и др.

Папка – объект *Windows* для хранения информации (имя, дата создания, размер и т. д.) о других объектах. В папке могут храниться файлы и другие папки.

Ярлык – файл, предназначенный для быстрого открытия какой-либо программы или файла с рабочего стола. В ярлыках хранится информация о местонахождении запускаемой программы или файла в памяти компьютера.

В нижней части рабочего стола обычно располагается *панель задач* (рисунок 1), которую условно можно разбить на три части:

- кнопка **Start** (*Пуск*), щелчок мышью на которой раскрывает **главное меню**;
- **область задач**, где располагаются кнопки открытых (запущенных) приложений (программ);
- **область уведомлений**, где обычно расположены индикаторы раскладки клавиатуры, резидентных программ, сетевых подключений, системного времени и т. п.



Рисунок 1 – Панель задач ОС *Windows*

Справа от кнопки **Start** (*Пуск*) предусмотрена область **Быстрый запуск** для размещения ярлыков наиболее часто используемых приложений, загрузка которых осуществляется одинарным щелчком мыши на ярлыке.

В состав главного меню входят следующие основные компоненты, наименование и количество которых может изменяться от версии к версии:

- **All Programs** (*Все программы*) – сложное подменю, предназначенное для запуска программ, инсталлированных в *Windows*;
- **My Documents** или **My Recent Documents** (*Документы*) – подменю, из которого доступны документы, с которыми пользователь работал в последнее время;
- **Control Panel** (*Панель управления*) – программы для настройки различных параметров операционной системы;
- **Search** (*Поиск*) – программы поиска объектов (программ, файлов, папок) в компьютере, в Интернете, людей в адресной книге и т. д.;
- **Help and Support** (*Справка и поддержка*) – доступ к справочным материалам;
- **Log Off** (*Выход из системы*) – завершение работы с *Windows*;
- **Shut Down** (*Выключение компьютера*) – команда, позволяющая произвести корректное выключение компьютера.

Удобным инструментом работы в операционной системе *Windows* является **контекстное меню**, раскрываемое однократным щелчком правой кнопкой мыши на выбранном объекте или нажатием специальной клавиши. Состав команд контекстного меню различен и зависит от объекта, на котором производится щелчок.

Основным объектом, с помощью которого организована работа в ОС *Windows*, является **окно**. Различают окна:


- папок (каталогов);
- приложений;
- документов;
- диалогов;


- справочной системы;
- системных (информационных) сообщений.


Основные *действия* с окнами папок, приложений, документов и справочной системы:

изменить расположение окна на рабочем столе;

свернуть окно – кнопка **Свернуть** 

развернуть окно на весь экран – кнопка **Развернуть** 

восстановить его размер до исходного – кнопка **Восстановить** 

закреть окно – кнопка **Закреть** 




Чтобы *изменить размеры* окна, следует навести указатель мыши на одну из границ или один угол окна, и, когда указатель мыши приобретет вид двунаправленной стрелки, нажать левую клавишу мыши и, не отпуская, переместить границу или угол окна на необходимое расстояние.

Если открыто несколько окон, то *активизировать* одно из них можно щелчком мышью на его заголовке или на соответствующей кнопке панели задач.

Расположение нескольких открытых окон на рабочем столе настраивается с использованием *контекстного меню панели задач* выбором одной из команд: **Cascade Windows** (*Окна каскадом*), **Tile Windows Vertically** (*Окна сверху вниз*), **Tile Windows Horizontally** (*Окна слева направо*). Свернуть все окна позволит команда **Minimize all** или **Show the Desktop** (*Показать рабочий стол*).

1.2.2 Структура и настройка окна

Рассмотрим в качестве примера окно системной папки **My computer** (*Мой компьютер*). Оно содержит:

- *строку заголовка*, в которой указано имя папки и находятся кнопки управления окном папки (,  или , 

- *строку главного меню*, команды которого сгруппированы по категориям, например, **File** (*Файл*), **Edit** (*Правка*), **View** (*Вид*), **Help** (*Справка*). Главное меню предоставляет доступ ко всем командам, выполняемым над объектами папки;

- *панели инструментов* с кнопками часто используемых команд;

- *рабочую область*, в которой отображается содержимое папки или вложенного объекта. Возможно наличие горизонтального и вертикального скроллингов для удобства перемещения по окну;

- *строку состояния*, отображающую некоторые текущие сведения об активном объекте.

Настроить внешний вид окна можно из пункта главного меню **View** (*Вид*). Отобразить или скрыть панель инструментов поможет команда **Toolbars** (*Панель инструментов*). Вывод информации о содержимом папки возможен в рабочей области в виде:

- Thumbnails (эскизы страниц);
- Tiles (плитка);
- Icons (значки);
- List (список);
- Details (таблица).

Сортируются файлы и папки в рабочей области (Arrange Icons by):

- по имени – Name;
- типу – Type;
- размеру – Size;
- времени изменения – Date.

Окна приложений, документов и других папок отличаются друг от друга составом главного меню, панелей инструментов (которые могут отсутствовать) и содержимым рабочей области.

1.2.3 Основные приемы работы с объектами

Основными рабочими инструментами операционной системы *Windows* являются:

- главное меню;
- панели инструментов;
- контекстное меню;
- комбинации клавиш клавиатуры.

Для *выделения* (активизации) объекта достаточно однократно щелкнуть левой кнопкой мыши на нем (т. е. на его ярлыке на рабочем столе, значке в окне папки или иному его обозначению). Выделенный объект затемнен, а его подпись выделена синим цветом.

Чтобы *открыть* папку или *запустить* приложение, дважды щелкните по нему левой кнопкой мыши. Если объект активен, то двойной щелчок можно заменить нажатием клавиши **Enter**.

Для *просмотра* или *настройки свойств* объекта из контекстного меню (щелчок правой кнопкой мыши) выбирают команду **Properties** (*Свойства*). Эта же команда доступна из пункта главного меню **File** (*Файл*) рабочего окна.

Чтобы *переименовать* объект, из его контекстного меню выбирают команду **Rename** (*Переименовать*), набирают новое имя и нажимают **Enter**. Для переименования выделенного объекта можно щелкнуть по его наименованию. При переходе текста названия в режим редактирования имя объекта удаляют, вводят новое и нажимают **Enter**.

Удалить выделенный объект можно нажатием клавиши **Delete** (*Удалить*) или одноименной командой из контекстного меню.

Для *копирования* объекта его выделяют, выбирают в контекстном меню команду **Copy** (*Копировать*), перемещают курсор в новое место расположения объекта и посредством щелчка на пункте контекстного меню **Paste** (*Вставить*) вставляют объект. Для команд копирования и вставки можно использовать соответствующие комбинации клавиш **Ctrl C** и

Ctrl V. Чтобы *переместить* объект, поступают аналогично одному из способов копирования, но из контекстного меню выбирают команду **Cut** (*Вырезать*), а вместо комбинации клавиш **Ctrl C** используют комбинацию клавиш **Ctrl X**.

Создавать объекты можно из контекстного меню рабочего стола:

- для *создания папки* выполняют команду **New / Folder** (*Создать / Папка*), в текстовом поле появившегося объекта вводят имя и подтверждают создание нажатием клавиши **Enter**;

- для *создания ярлыка* выполняют команду **New / Shortcut** (*Создать / Ярлык*), в отобразившемся окне **Мастера** указывают размещение программы, файла или папки, на которую будет ссылаться ярлык (кнопка **Browse – Обзор**). На следующем шаге дают имя ярлыку и нажимают кнопку **Готово**;

- для *создания файла* одной из установленных в *Windows* программ из списка **New** (*Создать*) в контекстном меню выбирают требуемый пункт. Например, для создания простого текстового документа выполняют **New / Text document** или **Text File Type** (*Создать / Текстовый документ*).

1.2.4 Некоторые настройки ОС Windows

Настройка даты и времени осуществляется из контекстного меню индикатора времени (расположенного в правой части панели задач) выбором команды **Adjust Date/Time** (*Настройка даты / времени*).

Настройку раскладки клавиатуры можно произвести, выбрав из главного меню **Control Panel** (*Панель управления*). Далее запустить **Date, Time, Language, and Local Options** (Дата, Время, Язык и региональные стандарты) и установить необходимые параметры.

Для настройки опций мыши из главного меню **Control Panel** (*Панель управления*) выбирают **Printers and Other Hardware** (Принтеры и другое оборудование) / **Mouse** (*Мышь*) и выполняют необходимые установки.

Для добавления языка в раскладку клавиатуры вызывается контекстное меню индикатора языковой панели, где выбирается команда **Settings** (*Настройка*). Для добавления выбирается команда **Add**, для удаления – **Remove**.

1.3 Файловый менеджер Total Commander

Total Commander – файловый менеджер для ОС *Windows* – программа, дающая возможность выполнять все основные операции над файлами и папками. Основными достоинствами рассматриваемого файлового менеджера являются:

- поддержка режима *Drag & Drop* – «переместил и отпустил»;
- расширенные возможности работы с файлами, папками (копирование, перемещение, создание, удаление и др.);
- возможность работать с архивами как с обычными папками, наличие нескольких внутренних архиваторов;
- наличие встроенного текстового редактора, открываемого нажатием клавиши **F4**, и встроенного средства просмотра файлов, вызываемого нажатием клавиши **F3**;
- расширенные средства поиска и многое другое.

Окно программы *Total Commander*, представленное на рисунке 2, включает строку заголовка, конфигурируемое главное меню и панели инструментов, кнопки для быстрого переключения логических дисков, две рабочие панели. Внизу окна находится строка состояния, командная строка, кнопки для выполнения некоторых команд.

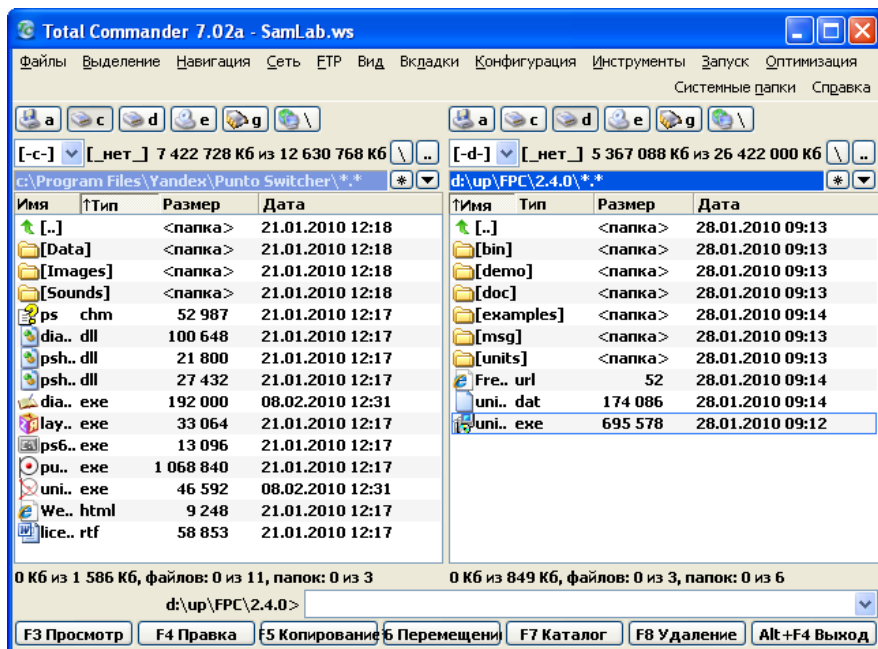


Рисунок 2 – Окно файлового менеджера *Total Commander*

При работе с файловым менеджером *Total Commander* используются как команды главного меню, так и следующие функциональные клавиши и

комбинации клавиш:

F1 – помощь;

F3 – просмотр файла;

F4 – правка, редактирование файла;

F5 – копирование объектов;

F6 – переименование / перемещение объектов;

F7 – создание каталога (папки);

F8 – удаление объекта;

Alt F1 – выбор левой панели (левого диска);

Alt F2 – выбор правой панели (правого диска);

Alt F4 – выход;

Alt F5 – архивация выбранных файлов;

Alt F7 – поиск;

Alt F9 – распаковка файлов;

Shift F4 – создание нового текстового файла и загрузка его в текстовый редактор (например, *AkelPad*);

Shift F5 – копирование объекта (с переименованием) в тот же каталог;

Shift Ctrl F5 – создание ярлыков для выделенных объектов;

Shift F10 – вызов контекстного меню;

Ctrl F1 – просмотр краткого списка папок и файлов;

Ctrl F2 – просмотр подробного списка папок и файлов;

Ctrl F3 – сортировка по имени;

Ctrl F4 – сортировка по расширению;

Ctrl F5 – сортировка по дате/времени;

Ctrl F6 – сортировка по размеру;

Ctrl F7 – без сортировки;

Ctrl F8 – отображение дерева каталогов;

Ctrl F10 – отобразить все файлы;

Ctrl F11 – отобразить только программы;

Tab – переключение курсора между левой и правой панелями;

Del – удаление файла или группы файлов;

Shift Del – удаление файлов, не помещая их в Корзину;

Insert – выделить файл или каталог;

Пробел – выделение файла или каталога. При этом на месте надписи <папка> отображается размер содержимого каталога;

Ctrl C – копирование объекта в буфер обмена;

Ctrl L – подсчет занимаемого места (для выбранных файлов);

Ctrl P – копирование текущего пути в командную строку;

- Ctrl Q** – отображение панели быстрого просмотра на месте файловой;
- Ctrl U** – поменять местами панели;
- Ctrl V** – вставить объекты из буфера обмена в текущий каталог;
- Ctrl X** – вырезать объекты в буфер обмена.

2 Порядок выполнения лабораторной работы

1 Загрузите операционную систему *Windows*, нажав одновременно три клавиши **Ctrl Alt Del** и удерживая их не менее одной-двух секунд. Задайте в диалоговом окне *имя пользователя (User Name)* **zf** (строчными латинскими буквами). Нажмите клавишу **Enter** или щелкните мышью на кнопке **OK**.

2 Исследуйте содержимое рабочего стола и главного меню ОС *Windows*. Найдите основные системные папки, отобразите главное меню **Start (Пуск)**.

3 Создайте на рабочем столе папку, используя в качестве ее названия свою фамилию.

Щелкните правой кнопкой мыши на свободном месте рабочего стола, чтобы вызвать контекстное меню. Выберите в нем команду **New / Folder (Создать / Папку)**, удалите выделенное словосочетание *New Folder (Новая папка)*, наберите имя папки и нажмите клавишу **Enter**.

4 Откройте созданную папку. Сверните, восстановите, измените размеры окна папки. Переместите папку в пределах рабочего стола. Закройте окно.

5 Создайте на рабочем столе текстовый файл.

Вызовите контекстное меню рабочего стола, выберите команду **New / Text Document (Создать / Текстовый документ)**, удалите выделенное словосочетание **Text Document.txt (Текстовый документ.txt)**, наберите имя файла (например, **task1.txt**) и нажмите клавишу **Enter**.

Откройте созданный файл, щелкнув правой кнопкой мыши на ярлыке текстового документа и выбрав **Open With / Notepad (Открыть с помощью / Блокнот)**. В текстовом поле введите фамилию, имя, отчество (кириллицей), предварительно изменив раскладку клавиатуры. Сохраните внесенные изменения – **File / Save (Файл / Сохранить)**.

6 Удалите с рабочего стола созданные папку и текстовый файл, используя для этого клавишу **Delete (Удалить)** или одноименную команду контекстного меню.

7 Откройте системную папку **My Computer (Мой компьютер)** и изучите ее содержание – содержимое компьютера.

Как обычно по умолчанию, диск **a:** служит для отображения и работы с содержимым дискеты.

Для работы студентов предназначен локальный диск **z:**.

8 Выполните задание 1, используя команды главного меню **View (Вид)**,

File / New (*Файл / Создать*) или команды контекстного меню.

Задание 1. Основные приемы работы с папками и файлами:

- выведите содержимое папки **z:** в виде таблицы с подробной информацией **Details** (*Таблица*);

- отсортируйте папки по дате создания, по размеру, по имени;

- найдите папку своей группы и зайдите в нее. В папке своей группы создайте папку, например, **Sergeev** (папку следует именовать своей фамилией), используя команду главного меню **File / New** (*Файл / Создать*);

- в своей папке создайте две папки **lab1** и **lab2**, используя команду контекстного меню **New / Folder** (*Создать / Папку*);

- в папке **lab1** создайте любым известным способом еще три папки **folder1**, **folder2**, **folder3**;

- скопируйте папки **folder1**, **folder2** в папку **lab2** с использованием команд контекстного меню;

- переместите папку **folder3** в папку **lab2** с помощью правой клавиши мыши;

- создайте копию папки **folder3** в папке **lab2** и переименуйте ее в **folder4**;

- в папке **folder4** создайте с помощью главного меню **File / New** (*Файл / Создать*) текстовый документ, откройте его, заполните информацией о своей работе и сохраните под именем **work.txt**;

- в папке **folder3** создайте с помощью контекстного меню текстовый файл **hobby.txt**, введя туда краткую информацию о своих увлечениях;

- для файла **hobby.txt** создайте ярлык на рабочем столе, выполнив из контекстного меню этого документа последовательность команд **Send To / Desktop (Create Shortcut)** – *Отправить / Рабочий стол (Создать ярлык)*;

- расположите все открытые окна каскадом, вертикально, горизонтально; сверните все открытые окна на панель задач с помощью контекстного меню панели задач;

- после проверки выполненной работы преподавателем удалите все созданные объекты и закройте окно папки **My Computer** (*Мой компьютер*).

9 Откройте файловый менеджер **Total Commander**. Это можно сделать двойным щелчком на его ярлыке на рабочем столе или выбором из главного меню последовательно **Start / All Programs / Total Commander** (*Пуск / Все программы / Total Commander*).

10 Изучите состав и содержимое окна **Total Commander**. Отобразите на левой и правой панелях корневой каталог диска **z:**.

11 Выполните задание 2 в файловом менеджере *Total Commander*.

Задание 2. Настройка внешнего вида файлового менеджера:

- отсортируйте файлы по дате создания, по размеру, по имени, используя пункт главного меню **Show** (*Вид*) или показанные на рисунке 3 кнопки табуляторов с именами полей подробного списка объектов;





Рисунок 3 – Настройка табуляторов

- выведите дерево каталогов данного диска или папки **Show / Tree** (*Вид / Дерево каталогов*);
- выведите содержимое диска **z:** в краткой (**Show / Brief** – *Вид / Краткий*), а затем в полной форме (**Show / Full** – *Вид / Подробный*);
- настройте внешний вид окна файлового менеджера, изменив используемые шрифты, цветовую гамму и язык: **Configuration / Options / Font, Color, Language** (*Конфигурация / Настройка / Шрифт, Цвет или Язык*).

12 Найдите папку своей группы и выполните задание 3 в *Total Commander*, используя все возможные приемы.

Задание 3. Основные приемы работы с папками и файлами:

- в папке своей группы создайте папку **Sergeev** (папку следует назвать своей фамилией), используя кнопку **F7 NewFolder** (*Каталог*) или клавишу **F7** клавиатуры;
- в своей папке создайте две папки **fiol** и **fio2**, а в папке **fiol** – еще три папки **dir1**, **dir2**, **dir3**;
- скопируйте папку **dir1** в папку **fio2** перетаскиванием. Скопируйте папку **dir2** в папку **fio2** с использованием кнопки **F5 Copy** (*Копировать*) или клавиши **F5** клавиатуры;
- переместите папку **dir3** в папку **fio2** с помощью кнопки **F6 Move** (*Переместить*) или клавиши **F6** клавиатуры;
- создайте копию папки **dir3** в папке **fio2** (последовательно нажимая комбинации клавиш **Ctrl C**, **Ctrl V**) и переименуйте ее в **dir4** (из контекстного меню выбрать **Rename** – *Переименовать*). Обе операции можно совместить, если сразу использовать **Shift F5**;
- в папке **dir4** создайте с помощью **Shift F4** текстовый документ, назовите его **address.txt**, напечатайте там свой домашний адрес и сохраните внесенные изменения. Закройте документ;

- просмотрите **address.txt** с помощью встроенной программы просмотра (кнопка **F3 View** – *Просмотр* или клавиша **F3** клавиатуры);
- откройте файл **address.txt** с помощью **F4** и допечатайте номер телефона;
- в папке **dir3** создайте с помощью кнопки  на панели инструментов текстовый файл **study.txt** и введите туда краткую информацию о своем вузе, факультете и специальности;
- для файла **study.txt** создайте ярлык на рабочем столе, выполнив из контекстного меню этого документа последовательность команд **Send To / Desktop (Create Shortcut)** – *Отправить / Рабочий стол (Создать ярлык)*.
- заархивируйте папку **fiol** в свою рабочую папку, используя комбинацию клавиш **Alt F5** или кнопку  панели инструментов;
- после проверки выполненной работы преподавателем удалите все созданные объекты, закрыв предварительно все открытые файлы. Завершите работу с файловым менеджером *Total Commander*.

Контрольные вопросы

- 1 Основные блоки ПК и их назначение.
- 2 Состав клавиатуры ПК. Назначение основных клавиш.
- 3 Особенности начала работы на ПК в компьютерных аудиториях ВЦ БелГУТа.
- 4 Основные понятия ОС *Windows*: рабочий стол, папка, ярлык. Основные системные папки.
- 5 Панель задач. Состав и назначение.
- 6 Состав главного меню *Windows*.
- 7 Классификация окон *Windows*.
- 8 Основные действия с рабочими окнами.
- 9 Структура и настройка внешнего вида рабочего окна.
- 10 Способы выполнения копирования, перемещения, переименования и других действий над объектами.
- 11 Создание папки, ярлыка, файла средствами *Windows*.
- 12 Назначение и основные возможности *Total Commander*.
- 13 Состав окна *Total Commander*.
- 14 Выполнение основных действий над объектами с помощью *Total Commander*.
- 15 Назначение основных функциональных клавиш.

САМОСТОЯТЕЛЬНАЯ РАБОТА 1

ТЕКСТОВЫЙ ПРОЦЕССОР *WORD*

Цель работы: углубить знания о возможностях текстового процессора *Microsoft Word*, усовершенствовать навыки редактирования и форматирования документов, научиться создавать и использовать стили, шаблоны документов, освоить автоматическое создание оглавлений.

1 Сведения из теории

1.1 Основные возможности *Word*

Текстовый процессор *Microsoft Word (Word)* предназначен для набора, редактирования, форматирования текста и оформления документов как самых простых, так и многостраничных сложной структуры, со встроеными объектами и элементами управления.

Для запуска текстового процессора *Word* из главного меню выбирают последовательно **Start / All Programs / Microsoft Word** (*Пуск / Все программы / Microsoft Word*).

1.1.1 Основные команды для работы с файлами документов

Команды для работы с файлами находятся в главном меню **Файл**:

- **Создать** – создание нового документа на основе одного из шаблонов или уже существующего файла;
- **Открыть** – открытие уже существующего файла из памяти в окно документа. После выбора данной команды в появившемся диалоговом окне указывается расположение и имя открываемого документа;
- **Закрыть** – завершение работы с данным документом и удаление его из окна документа. Если файл не был сохранен предварительно, перед его закрытием появляется запрос на сохранение;
- **Сохранить** – сохранение документа с ранее заданным именем. При сохранении файла впервые появляется диалоговое окно, в котором нужно указать имя и расположение сохраняемого файла документа;
- **Сохранить как** – сохранение документа в памяти под новым именем;
- **Параметры страницы** – установка параметров страниц документа (ширины полей, размера бумаги и др.);

• **Предварительный просмотр** – просмотр документа перед его печатью с целью внесения изменений в оформление документа (расположение на странице абзацев, рисунков, таблиц и т. п.). В режиме предварительного просмотра можно установить количество просматриваемых одновременно страниц и масштаб. Для возвращения в обычный режим нажимают кнопку **Заккрыть**;

• **Печать** – организация вывода документа на печать. Печать можно произвести как на бумажный носитель, так и в файл;

• **Свойства** – просмотр и задание некоторых свойств документа (название, автор, тема и др.). Здесь же можно узнать статистику по документу;

• **Выход** – закрытие всех открытых файлов документов и завершение работы *Word*.

1.1.2 Настройка внешнего вида окна приложения *Word* и параметров документов

Настройка окна приложения *Word* производится таким образом, чтобы оно содержало необходимые пользователю элементы. Это способствует повышению эффективности работы.

Внешний вид окна документа *Word* задается выбором в меню **Вид** одного из следующих *режимов*:

- Обычный;
- Веб-документ;
- Разметка страницы;
- Структура;
- Режим чтения;
- Во весь экран.

Горизонтальная и вертикальная *измерительные линейки* (в режиме разметки страницы) отображаются командой **Вид / Линейка**.

Подключение и настройка панелей инструментов. По умолчанию при запуске *Word* под строкой главного меню отображаются две *панели инструментов* – **Стандартная** и **Форматирование**. Эти панели содержат набор кнопок и раскрывающихся списков, позволяющих выполнять наиболее часто используемые команды (рисунок 1).

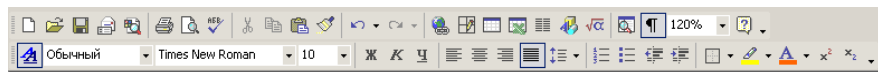


Рисунок 1 – Панели инструментов **Стандартная** и **Форматирование**

Отображение других панелей инструментов производится выбором из списка **Вид / Панели инструментов**.

Простейший способ добавить необходимую кнопку на панель

инструментов – отметить ее в списке доступных кнопок, вызываемом щелчком по скрывающейся кнопке панели инструментов (рисунок 2).

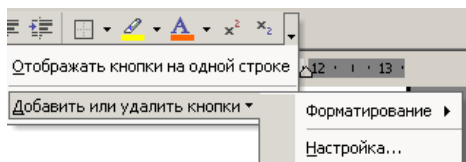


Рисунок 2 – Панели инструментов **Стандартная** и **Форматирование**

Если требуемая кнопка отсутствует в списке доступных, она может быть добавлена посредством диалогового окна **Настройка** (рисунок 3).

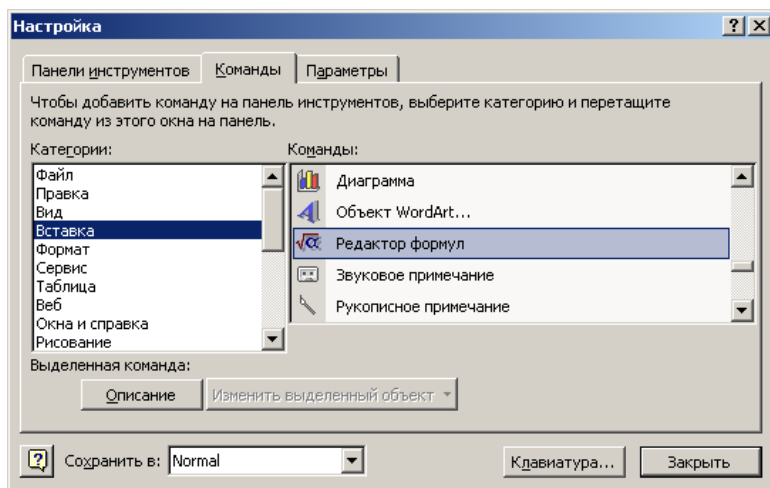




Рисунок 3 – Диалоговое окно **Настройка**

Например, для того чтобы поместить кнопку редактора формул *Microsoft Equation*  на панель инструментов, выполните команду **Вид / Панели инструментов / Настройка** (или **Сервис / Настройка**), перейдите на вкладку **Команды**, в списке **Категории** выберите **Вставка**, в списке **Команды** найдите с помощью полосы прокрутки значок , перетащите его методом *Drag & Drop* в любое удобное место на панели инструментов.

Настройка основных параметров документа. Важнейшие параметры документа настраиваются в окне **Параметры**, вызываемом последовательностью команд **Сервис / Параметры**. Это окно содержит вкладки **Вид**, **Общие**, **Правка (Edit)**, **Печать**, **Сохранение**, **Правописание**, **Безопасность** и др.

Приведем несколько примеров.

Настройка в главном меню **Файл** списка *последних файлов*, с которыми производилась работа, с целью их быстрого запуска (по умолчанию список состоит из четырех файлов) производится **Сервис / Параметры / вкладка Общие / Помнить список из ... файлов** (требуемое количество задается с помощью счетчика).

Настройка *автосохранения* документа через определенные промежутки времени осуществляется командой **Сервис / Параметры / Сохранение / Автосохранение каждые ... минут** (требуемое количество минут задается с помощью счетчика).

Устанавливают *автоматическую проверку грамматики и орфографии* на вкладке **Правописание** того же окна.

Защита документа от несанкционированного доступа выполняется командой **Сервис / Параметры / вкладка Безопасность / Пароль для открытия файла**.

1.1.3 Набор и редактирование текста

Набор текста в *Word* производится при помощи клавиатуры. При этом

- для переключения между *раскладками клавиатуры* используется комбинация клавиш **Ctrl Shift (Alt Shift)**, в зависимости от настроек пользователя). Язык можно выбрать из списка, отображаемого щелчком левой кнопкой мыши на индикаторе языковой панели **RU**;


- переключать *регистры* (верхний / нижний или прописные / строчные буквы) можно с помощью клавиши **Caps Lock** (постоянный режим) или удерживая клавишу **Shift** (смена регистра, пока нажата клавиша);

- переход *на новую строку* осуществляется **автоматически** при достижении курсором правой границы страницы. Если по какой-то причине надо перейти на новую строку, не прерывая абзац, нажимается **Shift Enter**;

- переход *на новый абзац* (красная строка) осуществляется однократным нажатием клавиши **Enter**;

- расстановка переносов производится автоматически, если выбрать эту опцию, последовательно выполняя **Сервис / Язык / Расстановка переносов**, флажок **Автоматическая расстановка переносов**;

- режим вставки / замены символов регулируется нажатием клавиши **Insert (Ins)**. Если вместо того, чтобы вставляться, раздвигая строку, символы печатаются поверх уже имеющихся, просто нажмите один раз **Insert**.

Примечание – Работать в текстовом процессоре *Word* лучше в режиме отображения непечатаемых символов, переход в который осуществляется нажатием кнопки **Непечатаемые знаки**  панели инструментов **Стандартная**. Символ абзаца имеет вид ¶, символ табуляции – →, а символ разрыва строки – ↵. Место вставки пробела обозначается точкой, расположенной в центре пробела. Любой из непечатаемых знаков можно удалить, как и обычный символ. Непечатаемые знаки, как следует из названия, на печать не выводятся.

Перемещение по тексту документа производится:

- с помощью клавиш управления курсором;
- щелчком левой кнопки мыши на позиции в тексте, куда необходимо перейти;
- с помощью клавиш **Home** и **End** – в начало и в конец текущей строки соответственно;
- с помощью клавиш **Page Up** и **Page Down** – на страницу (отображаемую на экране) вверх и вниз соответственно;
- с помощью комбинаций клавиш **Ctrl Home** и **Ctrl End** – в начало и в конец документа соответственно.

Выделить фрагмент текста можно при помощи мыши или клавиатуры:

- нажав и не отпуская левую кнопку мыши, выделить необходимый фрагмент;
- выполнив двойной щелчок левой кнопкой мыши на слове – выделить слово;
- выполнив тройной щелчок левой кнопкой мыши на любой позиции в абзаце – выделить абзац;
- выполнив тройной щелчок левой кнопкой мыши на левой границе документа – выделить весь документ;
- удерживая клавишу **Shift** и одновременно нажимая необходимые клавиши управления курсором;
- нажав комбинацию клавиш **Shift End** – выделить строку от места расположения курсора до ее конца;
- нажав комбинацию клавиш **Shift Home** – выделить строку с начала и до курсора;
- нажав комбинацию клавиш **Shift Page Up** (**Shift Page Down**) – выделить отображаемую на экране страницу вверх (вниз) от позиции курсора.
- нажав комбинацию клавиш **Ctrl A** – выделить документ целиком (выделить всё).

Для *удаления* символа, расположенного слева от текстового курсора, нажимают клавишу **Backspace** (←). Для удаления символа справа от курсора применяется клавиша **Delete** (**Del**).

Перед удалением фрагмента текста его необходимо сначала выделить, а затем нажать **Backspace** (←) или **Delete** (**Del**).

Примечание – Если фрагмент документа в виде, например, символа, слова, строки, абзаца выделен, и его необходимо заменить, то операция удаления этого фрагмента является лишней, так как ввод любого символа приводит к замене выделенного фрагмента введенным символом.

Чтобы *переместить* выделенный фрагмент текста в другое место документа:

- его перетаскивают, удерживая нажатой левую кнопку мыши, в новое

место документа;

- выбирают в главном меню **Правка (Edit)** или в контекстном меню выделенного фрагмента команду **Вырезать (Cut)**, устанавливают курсор в месте вставки фрагмента и выбирают в главном меню **Правка (Edit)** или в контекстном меню, вызванном в месте расположения курсора, команду **Вставить**;

- нажимают комбинацию клавиш **Ctrl X**, устанавливают курсор в месте вставки фрагмента и нажимают комбинацию клавиш **Ctrl V**.

Чтобы *скопировать* выделенный фрагмент в другое место документа:

- выбирают в главном меню **Правка (Edit)** или в контекстном меню выделенного фрагмента команду **Копировать (Copy)**, устанавливают курсор в месте вставки фрагмента и выбирают в главном меню **Правка (Edit)** или в контекстном меню, вызванном в месте расположения курсора, команду **Вставить (Paste)**;

- нажимают комбинацию клавиш **Ctrl X**, устанавливают курсор в месте вставки фрагмента и нажимают комбинацию клавиш **Ctrl V**.

Для *поиска и замены* в документе символов, слов и фрагментов текста используют команды главного меню **Правка (Edit)**: **Найти...(Find)**, **Заменить... (Replace)**, **Перейти... (Go To)**, вызывающие показанное на рисунке 4 диалоговое окно **Найти и заменить (Find and Replace)**.

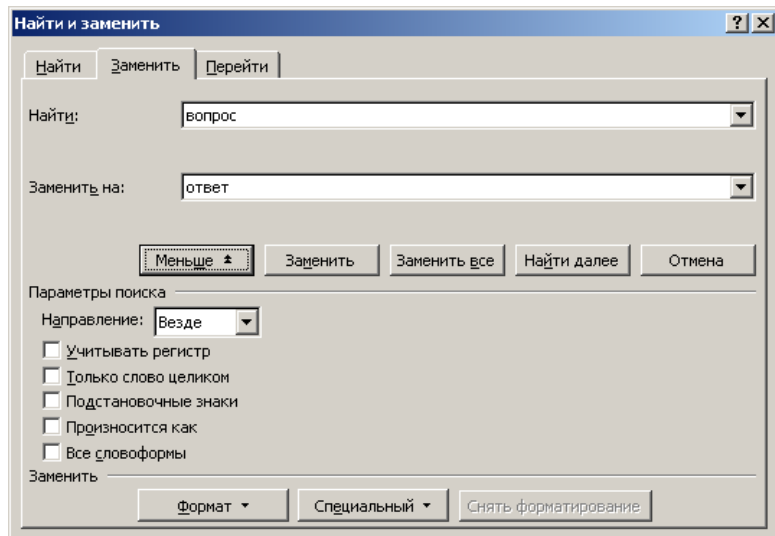


Рисунок 4 – Диалоговое окно **Найти и заменить**

Например, чтобы найти определенный фрагмент текста и заменить его другим, заполняются поля на вкладке **Заменить (Replace)**. Дополнительные

параметры поиска и замены отображаются при нажатии кнопки **Больше**. Найти и заменить словосочетания, содержащие непечатаемые знаки и другие специальные символы, поможет раскрывающийся список **Специальный**.

Вставка нумерации страниц в документе производится командой **Вставка / Номера страниц** или посредством инструментальной панели, вызываемой **Вид / Колонтитулы**.

С помощью показанной на рисунке 5 панели **Колонтитулы** в области верхнего и нижнего колонтитулов документа можно автоматически вводить дополнительную информацию (дату создания документа, название документа, фамилию, имя и отчество автора и т. п.).

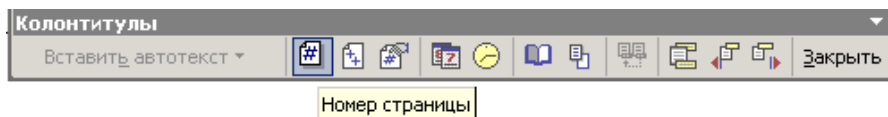


Рисунок 5 – Инструментальная панель **Колонтитулы**

Приведем еще несколько полезных советов.

1 Если необходимо изменить регистр уже напечатанного текста, например, поменять строчные буквы на прописные или наоборот, используйте команду **Формат / Регистр**.

2 Для исправления орфографической ошибки в слове можно щелкнуть по нему правой кнопкой мыши и выбрать из предложенного в контекстном меню списка слово, написанное верно.

3 Если требуется напечатать слова на расстоянии друг от друга, **не используйте многократное нажатие клавиши Пробел!** Воспользуйтесь клавишей **Tab** или настройками окна табуляции, вызываемого последовательностью команд **Формат / Табуляция**.

4 Чтобы зафиксировать интервал между словами (например, при наборе формул в строку или фамилии и инициалов), используйте «неразрывный пробел», который печатается комбинацией клавиш **Shift Ctrl Пробел**.

5 Если необходимо печатать текст с новой страницы, **не используйте многократное нажатие клавиши Enter!** Примените комбинацию клавиш **Ctrl Enter** или выберите в главном меню **Вставка / Разрыв / Начать новую страницу**.

1.1.4 Форматирование шрифтов и абзацев

Форматирование – это придание определенных параметров отображения документа. Основными в *Word* являются форматирование шрифтов (**Формат / Шрифт**) и форматирование абзацев (**Формат / Абзац**).

Если текст уже набран, то для *изменения шрифта* фрагмент текста должен быть выделен. Затем можно использовать кнопки и раскрывающиеся списки показанной на рисунке 6 панели инструментов **Форматирование**.



Рисунок 6 – Панель инструментов **Форматирование**

Другими способами являются использование комбинаций клавиш или, что предпочтительнее, диалогового окна **Шрифт (Формат / Шрифт)**, в котором сразу можно задать всевозможные параметры текста.

Диалоговое окно **Шрифт** содержит вкладки:

- **Шрифт** – для установки гарнитуры шрифта (Times New Roman, Arial и др.), начертания (полуужирный, курсив и др.), размера шрифта (кегель), цвета текста, подчеркивания (более десяти видов), видоизменения текста;

- **Интервал** – для установки масштаба, межсимвольного интервала, смещения;

- **Анимация** – для задания различных дополнительных эффектов текста.


Каждая вкладка содержит поле просмотра образца текста, полученного в результате форматирования.

Для *форматирования абзаца* (последовательности символов, заканчивающейся знаком ¶) текст выделять необязательно. Все задаваемые параметры форматирования будут автоматически назначены тому абзацу, в котором установлен курсор.

Диалоговое окно **Абзац**, вызываемое командой **Формат / Абзац**, содержит две вкладки: **Отступы и Интервалы** и **Положение на странице**.

На вкладке **Отступы и Интервалы** устанавливаются выравнивание текста (по ширине, по левому краю и т. д.), отступы текста от левого и правого поля страницы, отступ первой (красной) строки, интервалы выше и ниже абзаца (по вертикали), междустрочный интервал и др.


На вкладке **Положение на странице** устанавливаются дополнительные параметры, такие как **запрет висячих строк, не отрывать от следующего** и другие, позволяющие оптимально расположить абзац на странице.

Для перенесения всех параметров форматирования на другой фрагмент текста или абзац используется кнопка  – **Формат по образцу**.

Задание.

1 Запустите *Word: Start / All Programs / Microsoft Word (Пуск / Все программы / Microsoft Word)*.



2 Отобразите панель создания нового документа **Файл / Создать** (для Word 2003) и изучите ее содержимое.

3 Включите отображение непечатаемых знаков  и наберите абзац текста, выданного преподавателем. Разбейте текст на два абзаца (клавиша **Enter**). Выполните проверку правописания. Выполните расстановку переносов (**Сервис / Язык / Расстановка переносов**).

4 Сохраните документ в своей папке с именем **файл1.doc**, для этого

выполните команду **Файл / Сохранить как**, выберите папку, задайте имя **файл1**, щелкните на кнопке **Сохранить**. Расширение **.doc** в имени сохраненный документ получит автоматически.

5 Подключите панели инструментов WordArt, Рисование, Таблицы и границы из главного меню **Вид / Панели инструментов**. Отключите WordArt, Таблицы и границы, используя контекстное меню области панелей инструментов.

6 Поместите кнопки  – **Параметры страницы** и  – **Время** на панель инструментов и команду **Сохранить все** в меню **Файл**, пользуясь диалоговым окном **Настройка** (вкладка **Команды**), вызываемом в меню **Сервис**.

7 Настройте оптимальный режим просмотра документа посредством меню **Вид**, пробуя последовательно Обычный, Веб-документ, Разметка страницы. Выберите наиболее подходящий масштаб. Обратите внимание, что можно сразу задать отображение текста на всю ширину страницы.

8 Настройте параметры страницы **Файл / Параметры страницы**. Установите параметры всех полей документа равными **1,5**, зеркальные поля, книжную ориентацию. Измените размер бумаги: **Файл / Параметры страницы / Размер бумаги** – ширина **21 см**, высота **29,7 см**. Установите нумерацию строк.

9 Создайте в документе два раздела, вставив в него разрыв раздела. Для этого командой **Вставка / Разрыв** вызовите диалоговое окно **Разрыв** (рисунок 7), в группе **Новый раздел** установите опцию **со следующей страницы** и щелкните на кнопке **ОК**.

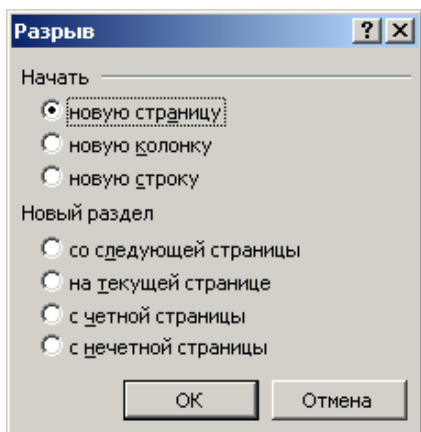


Рисунок 7 – Диалоговое окно **Разрыв**

10 Задайте название документа, авторство, учреждение, тему (**Файл / Свойства / Документ**). Просмотрите статистику по документу (**Файл / Свойства / Статистика**).

11 В каждом абзаце выполните форматирование различными шрифтами (примените как можно больше вариантов форматирования).

Выберите первый абзац (щелчком мыши). Для него установите следующие параметры:

выравнивание текста – **по правому краю**;

красная строка – **1,5 см**;

интервалы:

перед – **6 пт**;

после – **12 пт**;

междустрочный – **двойной**.

12 Переместите второй абзац на отдельный лист, для этого вставьте разрыв


страницы, нажав комбинацию клавиш **Ctrl Enter** или выполнив команду **Начать новую страницу** посредством диалогового окна **Разрыв** (см. рисунок 7), которое вызывается командой **Вставка / Разрыв**.

13 Для второго абзаца установите:
выравнивание текста – **по ширине**;
красная строка – **нет**;
отступы:
слева – **2 см**;
справа – **3 см**;
междустрочный – множитель **1,3**.

14 Замените в тексте все буквы **а** на **о** (**Правка / Заменить**). Исправьте ошибки правописания. Измените регистр символов первого предложения (**Формат / Регистр**): **Все прописные**.

15 Вставьте символы (**Вставка / Символ**):

–, —, ©, ☞, ☞, Ω, ♣, ✱, ♣

16 Наберите в редакторе формул  следующие формулы:

$$\sum_{n=1}^m \frac{2n+1}{n^2(n+1)}; \quad (1)$$

$$\lim_{x \rightarrow 0} \frac{\operatorname{tg} x - 1 + \cos 3x}{e^x - e^{-x}}; \quad (2)$$

$$A = \begin{pmatrix} x & 1 & x \\ 3x & x^2 & -x \\ 2 & x & 3 \end{pmatrix}. \quad (3)$$

17 Вставьте в документ нумерацию страниц внизу листа по правому краю с помощью команды **Вставка / Номера страниц**.

18 В верхний колонтитул (**Вид / Колонтитулы**) вставьте дату создания документа, в нижний – название документа. Используйте шрифт Courier, размер **10**, разреженный.

19 Установите следующие свойства документа (**Сервис / Параметры**): защиту документа паролем, отображение 6 последних открытых файлов в списке **Файл (помнить список из 6 файлов)**, автосохранение документа каждые 7 минут. Установите опцию печати скрытого текста.

20 Сохраните документ.

21 Создайте новый документ и сохраните его в своей папке с именем **файл2.doc**. Скопируйте первый абзац и переместите второй абзац из документа **файл1.doc** в **файл2.doc**. Сохраните **файл2.doc**.

22 Просмотрите документ **файл2.doc** в режиме **Предварительного просмотра** перед печатью (из меню **Файл**).

23 Изучите опции печати и распечатайте документ **файл2.doc**.

При выполнении п. 23 в компьютерной аудитории ВЦ БелГУТа в

диалоговом окне **Печать** необходимо правильно выбрать **Имя принтера**, которое состоит из номера аудитории и номера принтера, включить принтер, установить бумагу и только после этого отправить документ на печать.

24 Закройте сначала документ **файл2.doc (Файл / Закрывать)**, затем завершите работу с *Word*.

2 Порядок оформления контрольной работы

Прежде чем начать набор текста пояснительной записки к контрольной работе (или другого многостраничного документа), следует установить параметры компьютерного набора (верстки)*.

1 В меню **Файл** вызовите диалоговое окно **Параметры страницы** и установите:

поля, см:

верхнее – **1,7**;

нижнее – **2**;

левое – **2,5**;

правое – **1,0**;

размер бумаги – **A4** (21 × 29,7 см);

ориентация – **книжная**;

колонтитулы от края: верхний – **1,3**, нижний – **1,6 см**;

размер отступа номера страницы от текста – **4–5 мм**.

2 Создайте стиль основного текста, вызвав командой **Формат / Стиль** диалоговое окно **Стиль**:

Имя **Главный**;

Стиль **Абзаца**;

Основан на стиле (**нет**);

Стиль следующего абзаца **Главный**.

3 Щелкните на кнопке **Формат**, затем на пункте **Шрифт**. Задайте параметры абзаца:

выравнивание – **по ширине**;

отступ первой строки (красная строка) – **0,7 см**;

междустрочный интервал – **одинарный**.

Согласно технологической инструкции по набору и верстке печатных материалов страница текста не должна начинаться с неполной последней строки абзаца и не должна заканчиваться первой строкой абзаца. Данное требование будет удовлетворено, если в диалоговом окне **Абзац** на вкладке **Положение на странице** в группе **Разбивка на страницы** включить опцию **Запрет висячих строк**.

Закройте окно **Абзац**, а затем окно **Создание стиля**, щелкая каждый раз на кнопке **ОК**.

* Приведенные параметры оформления могут быть изменены в задании на контрольную работу.

4 Выполните команду **Сервис / Язык / Расстановка переносов** и установите флажок в строке **Автоматическая расстановка переносов**. Максимальное число последовательных переносов – 4.

5 Измените стили заголовков документа (т. е. стили названий глав, разделов или пунктов). Для этого выполните последовательно **Формат / Стили и форматирование**, при этом отобразится одноименная панель (в Word 2003). Найдите в списке стилей стиль **Заголовок 1**, щелкните слева на кнопку с треугольником и выберите **Изменить** (у более ранних версий Word кнопка **Изменить** находится внизу окна форматирования стиля). Установите следующие параметры заголовка, используя кнопку

Формат ▾ диалогового окна **Изменение стиля**:

- Times New Roman, **полужирный**, размер **14**, все прописные (большие, заглавные);

- выравнивание по центру, отступ первой строки – **нет**, интервалы: перед – **6 пт**, после – **3 пт**. Вкладка **Положение на странице** – установить флажки **не отрывать от следующего**, **запретить автоматический перенос слов**.

Для заголовков второго уровня (подзаголовков, подразделов, подпунктов) измените стиль **Заголовок 2**:

- Times New Roman, **полужирный**, размер **12**;

- выравнивание по левому краю, отступ первой строки – **нет**, интервалы: перед – **12 пт**, после – **6 пт**. Вкладка **Положение на странице** – установите флажки **не отрывать от следующего**, **запретить автоматический перенос слов**.

Аналогично изменяются стили заголовков третьего и других уровней.

Примечание – Не применяйте стиль **Обычный** и не основывайте на нем вновь создаваемые стили. Не включайте опцию **Обновлять автоматически** в диалоговом окне **Создание стиля**.

6 Установите автоматическую нумерацию заголовков путем создания многоуровневого списка: **Формат / Список / вкладка Многоуровневый / список на основе заголовков**. При этом заголовки первого уровня будут нумероваться одним числом, а второго уровня – двумя числами, разделенными точкой, например, 1.1 или 2.4.

Дополнительные опции нумерации устанавливаются после нажатия на кнопку **Изменить** в диалоговом окне **Список**. Избегайте вставки знака табуляции между номером и названием.

Разделы **Введение**, **Заключение**, **Список литературы** не нумеруются. Названиям этих разделов назначьте стиль заголовка первого уровня. Если номер в процессе работы отобразился, его можно просто удалить.

7 Вставьте в документ нумерацию страниц внизу листа по правому краю с помощью команды **Вставка / Номера страниц**. Номер на первой странице не указывается (следует снять соответствующую галочку).

8 Теперь можно набирать (или вставлять из ранее созданных документов) текст контрольной работы, основной текст – стиль **Главный**, заголовки

разделов – стиль **Заголовок 1**, подразделов – **Заголовок 2**, причем для форматирования абзацев достаточно установить курсор в соответствующий абзац и выбрать из списка стилей, расположенного на панели **Форматирование**, требуемый стиль.

Каждый раздел следует начинать с новой страницы (**Ctrl Enter**). Оставьте в начале документа две пустые страницы: для титульного листа и оглавления (содержания), которое создается и форматируется в последнюю очередь.

9 Рисунки и формулы выравниваются по центру. Подрисовочная подпись оформляется шрифтом на размер меньше, чем обычный текст. Можно использовать сквозную нумерацию рисунков и формул (**Вставка / Ссылка / Название**).

10 Оглавление в контрольной работе должно быть создано автоматически средствами *Word*. Перейдите на вторую страницу документа, наберите слово **Содержание** (прописные, вразрядку, размер 14, можно полужирным шрифтом), вставьте новый абзац, выполните команду **Вставка / Оглавление и указатели / вкладка Оглавление /** и щелкните на кнопке **ОК**. При необходимости на вкладке **Оглавление** внесите изменения в параметры оформления.

Оглавление составляется из заголовков документа, оформленных стилями **Заголовок 1**, **Заголовок 2** и **Заголовок 3**, причем сразу с нумерацией страниц. Поэтому не назначайте абзацу со словом **Содержание** стили заголовков.

Для изменения оглавления достаточно выбрать в его контекстном меню команду **Обновить поле**.

Контрольные вопросы

- 1 Основные параметры документа *Word*.
- 2 Подключение и настройка панелей инструментов.
- 3 Режимы просмотра документов.
- 4 Набор и редактирование текста.
- 5 Какие приемы используются для перемещения по тексту?
- 6 Как выделить фрагмент текста?
- 7 Копирование, перемещение и удаление фрагментов текста.
- 8 Поиск и замена фрагментов текста.
- 9 Параметры страницы и свойства документа.
- 10 Проверка правописания.
- 11 Расстановка переносов.
- 12 Форматирование символов.
- 13 Что такое абзац в *Word*? Форматирование абзацев.
- 14 Нумерация страниц. Колонтитулы.
- 15 Создание и изменение стилей абзацев.
- 16 Автоматическое создание оглавления.

ЛАБОРАТОРНАЯ РАБОТА 2

ЯЗЫК *PASCAL*. ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ И РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ. ЦИКЛЫ

Цель работы: получить представление об основных возможностях среды программирования *Free Pascal*, *PascalABC* или *Turbo Pascal*. Изучить алфавит и основные понятия языка программирования *Pascal*. Получить навыки набора, сохранения, отладки, просмотра и анализа результатов работы программы. Научиться программировать линейные, разветвляющиеся и циклические вычислительные алгоритмы.

1 Сведения из теории

Одним из наиболее распространенных языков программирования является *Pascal*, созданный в 1968–1971 годах профессором Высшей технической школы (*ETH*) г. Цюриха (Швейцария) Никлаусом Виртом и названный в честь французского математика и философа Блеза Паскаля (1623–1662). Благодаря небольшому количеству базовых понятий, простому синтаксису и принципам, в основе которых лежат системный подход и структурное программирование, язык *Pascal* широко используется для обучения программированию.

Пользователь ПК, начинающий программировать на языке *Pascal*, прежде всего, сталкивается со средой (оболочкой) или системой программирования, включающей в себя совокупность программных средств:

- текстовый редактор, предназначенный для набора и редактирования текста программы (исходного кода, листинга);
- компилятор – программа (совокупность программ), обрабатывающая исходный код, и в случае полного соответствия правилам языка, переводящая (транслирующая) его на машинный язык (объектный код);
- редактор связей (компоновщик), преобразующий объектный код в машинный;
- загрузчик, генерирующий готовый к исполнению **exe**-файл (исполняемый код), который можно переносить на другой компьютер;
- библиотеку функций;
- справочную систему.

Среда программирования предназначена для набора текста программы, отладки, выполнения и получения результатов работы программы.

1.1 Краткое описание среды *Free Pascal*

1.1.1 Запуск и обзор команд главного меню

Для запуска среды *Free Pascal* можно воспользоваться ярлыком с рабочего стола *Windows* или выполнить последовательность команд главного меню **Start / All Programs / Free Pascal** (*Пуск / Все программы / Free Pascal*). В состав главного меню, расположенного в верхней части окна (рисунок 1) входят команды:

- **File** (*Файл*) – для работы с файлами;
- **Edit** (*Правка*) – редактирование файла;
- **Search** (*Поиск*) – поиск или замена фрагментов текста;
- **Run** (*Прогон*) – выполнение программы;
- **Compile** (*Компиляция*) – компиляция программы;
- **Debug** (*Отладка*) – отладка;
- **Tools** (*Сервис*) – команды вспомогательного характера;
- **Options** (*Параметры*) – настройка среды программирования;
- **Window** (*Окно*) – для работы с окнами документов;
- **Help** (*Помощь, Справка*) – вызов справочной информации.

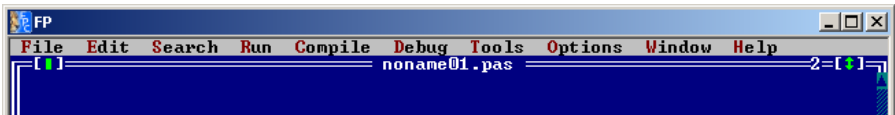


Рисунок 1 – Главное меню *Free Pascal*

В нижней части окна программы находится описание некоторых функциональных клавиш, список которых изменяется в зависимости от выполняемой команды главного меню (рисунок 2).



Рисунок 2 – Строка функциональных клавиш *Free Pascal*

Основные команды меню **File**:

- **New** – создать новый документ;
- **Save (F2) (Save as...)** – сохранить, сохранить как...;
- **Open (F3)** – открыть существующий файл;
- **Print** – печать;
- **Exit (Alt X)** – выход;
- **Change Dir** – изменение текущего директория.

В меню **Edit** входят команды:

- **Undo (Alt Backspace)** – отмена действия;
- **Redo** – возврат на действие вперед;

- **Copy (Ctrl Ins)** – копировать;
- **Cut (Shift Del)** – вырезать;
- **Clear (Ctrl Del)** – удалить;
- **Paste (Shift Ins)** – вставить;
- **Select All** – выделить все;
- **Unselect** – снять выделение;
- **Show Clipboard** – просмотреть содержимое буфера обмена;
- **Copy to Windows** – копировать в *Windows*;
- **Paste from Windows** – вставить из *Windows*.

Посредством меню **Search (Поиск)** можно выполнить команды:

- **Find** – найти фрагмент текста;
- **Replace** – заменить;
- **Go to line number** – перейти к строке с указанным номером.

В меню **Run (Выполнение)** собраны команды, предназначены для выполнения программы:

- **Run (Ctrl F9)** – компиляция, выполнение, вывод результатов работы программы;
- **Trace into (F7)** – трассировка, пошаговое выполнение программы;
- **Step over (F8)** – выполнение программы по шагам, без отслеживания работы вызываемых процедур и функций;
- **Go to cursor (F4)** – выполнение программы до строки, помеченной курсором в текущем окне редактора;
- **Program reset (Ctrl F2)** – сброс всех ранее задействованных отладочных свойств, прекращение отладки программы, удаление из памяти исполнявшейся программы.

Компиляция программы выполняется выбором пунктов меню **Compile (Компиляция)** одной из команд **Compile (Alt F9)**, **Make (F9)** или **Build**. Здесь же находится команда **Information**, показывающая статистику программы.

Посредством меню **Debug (Отладка)** чаще других выполняются команды:

- **Output** – открытие окна, в котором отображаются на экране результаты выполнения программы в текстовом режиме;

- **User screen (Alt F5)** – переход в окно результатов выполнения программы.

Меню **Tools (Сервис)** содержит команды вызова встроенного калькулятора (**Calculator**), таблицы кодов (**Ascii table**) и некоторые другие.

В меню **Options (Параметры)** доступны команды настройки, управления генерацией машинного кода программы, регулирующие размер памяти, режим работы компоновщика и т. д. В дополнительном подменю команды **Environment** собраны пункты:

- **Preferences** – параметры, определяющие условия работы в целом;
- **Editor** – параметры, определяющие условия работы редактора;
- **Keyboard&Mouse** – опции работы клавиатуры и мыши;
- **Startup** – опции, определяющие начальные параметры;
- **Colors** – цвет для каждого из элементов *Free Pascal*.

Команды для работы с окнами документов находятся в пункте главного меню **Window** (*Окно*):

- **Tile** – расположение окон всех открытых документов равномерно по всему экрану;
- **Cascade** – расположение открытых окон каскадом;
- **Close all** – закрытие всех окон;
- **Refresh display** – обновление экрана;
- **Size / Move (Ctrl F5)** – изменение размера и расположения окна на экране (см. функциональные клавиши внизу окна);
- **Zoom (F5)** – масштабирование окна;
- **Next (F6)** – активизация следующего окна;
- **Previous (Shift F6)** – активизация предыдущего окна;
- **List (Alt 0)** – вывод на экран списка всех открытых окон среды;
- **Close (Alt F3)** – закрытие окна текущего документа.

В пункте главного меню **Help** (*Помощь*), в частности, находятся:

- **Content** – вывод содержания справочной службы;
- **Index** – алфавитный перечень всех ссылок справочной службы;
- **Topic search** – поиск в окрестности курсора зарезервированного слова, стандартной функции и выдача соответствующей справки;
- **Previous topic** – предыдущее справочное сообщение;
- **About** – информация о программе.

Основные функциональные клавиши и комбинации клавиш:

- **Esc** – отмена команды;
- **F1** – обращение за справкой к встроенной справочной службе;
- **F2** – сохранение редактируемого текста программы в дисковый файл;
- **F3** – вызов окна открытия файла;
- **F9** – компиляция программы без последующего выполнения;
- **F10** – активизация пунктов главного меню;
- **Alt F9** – компиляция программы;
- **Ctrl F9** – прогон программы: компиляция программы, находящейся в редакторе, загрузка ее в оперативную память и выполнение;
- **Alt F3** – закрытие активного окна документа;
- **Alt F5** – отображение окна вывода результатов работы программы;
- **Alt F10** – вызов контекстного (локального) меню;
- **Alt X** – закрытие окна приложения (среды программирования).



Примечания

1 Если активизировать главное меню нажатием клавиши **F10**, то любую команду можно выбрать с помощью клавиш управления курсором и подтвердить выбор нажатием клавиши **Enter**.

2 Любой из пунктов главного меню раскрывается комбинацией клавиш **Alt <выделенная буква в названии пункта меню>**. Команда из раскрытого подменю вызывается **Shift <выделенная буква в названии команды>**. Например, вызвать окно изменения текущего директория можно, нажав **Alt F**, а затем – **Shift C**.

1.1.2 Создание, открытие и сохранение файла документа. Работа с окнами

Для *создания окна нового документа* выполняют последовательно **File / New** (*Файл / Создать*). Каждый новый файл получает имя и порядковый номер, отображаемые в верхней части окна документа, например, **noname01.pas**. Номер открытого документа в текущем сеансе работы отображается справа от имени файла.

Примечание – Закрывающая кнопка окна документа  находится в левой, а масштабирующая  – в правой верхней части окна документа.

Одновременно в редакторе интегрированной среды может быть открыто несколько окон с программами. *Переход между окнами* осуществляется с помощью клавиши **F6**. *Закрывается* активное окно редактора (но не приложения *Free Pascal*) комбинацией клавиш **Alt F3**. Остальные команды для работы с окнами находятся в меню **Window** (*Окно*).

Новый файл следует сразу *сохранить в личной папке*, выполнив команду **File / Save as** (*Файл / Сохранить как*). За клавишей **F2** закреплено дублирование этой команды. В процессе работы нажимайте клавишу **F2** всякий раз, когда вносите изменения в текст создаваемой программы.

Для вызова диалогового окна *открытия* уже существующего файла используется последовательность команд **File / Open** (*Файл / Открыть*) или клавиша **F3**.

Примечание – Сохранение пустого документа («чистого окна редактора») с именем ранее созданного пользователем файла программы приводит к потере этой программы.

1.1.3 Набор и редактирование текста программы

Текст программы (листинг) набирается в окне редактирования. При этом используются обычные приемы перемещения по тексту (клавиши управления курсором, в начало строки – **Home**, в конец строки – **End**, постраничный переход – **Page Up**, **Page Down**). Для набора текста с новой строки используется клавиша **Enter**.

Кроме выделения фрагментов текста с помощью мыши следует помнить о комбинациях клавиш **Shift** <клавиши управления курсором>, **Shift Home**, **Shift End** и др.

Команды редактирования текста (копировать – **Copy**, вырезать – **Cut**, вставить – **Paste**, удалить – **Clear** и др.) можно найти в главном меню **Edit** (*Правка*), вызываемом также **Alt E**, или в контекстном меню (**Alt F10**). Удалить строку, в которой находится курсор, можно **Ctrl Y**, а от курсора до конца строки – **Ctrl Q, Y**.

Среда *Free Pascal* предоставляет возможность назначения привычных комбинаций клавиш **Ctrl C**, **Ctrl X** и **Ctrl V** при выполнении операций копирования – **Copy**, вырезания – **Cut** и вставки – **Paste**. Командой **Options /**

Environment / Keyboard & mouse откройте диалоговое окно **Mouse Options** (рисунок 3) и щелкните в группе **Keys for cut, copy and paste** на скобках слева от пункта **Microsoft convention**, а затем на кнопке **OK**.

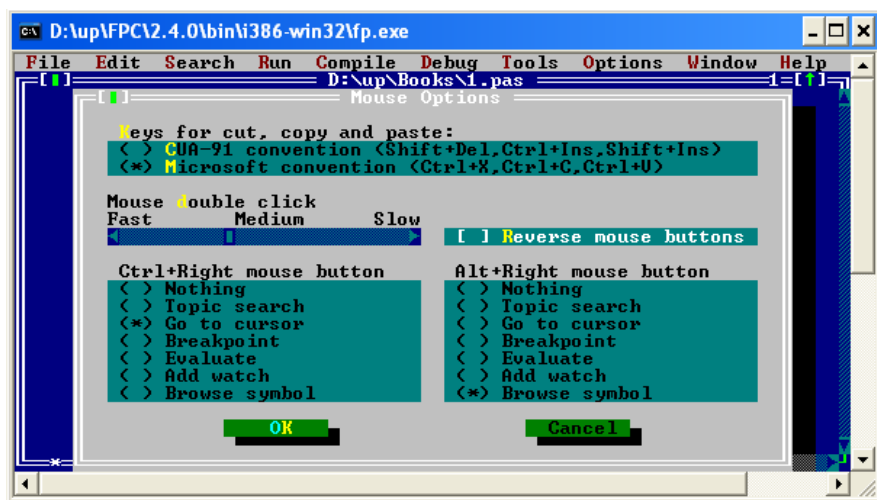


Рисунок 3 – Переход к **Microsoft convention** (Ctrl C, Ctrl X и Ctrl V) в *Free Pascal*

1.1.4 Компиляция программы. Некоторые ошибки

После набора текста программы и исправления очевидных ошибок ввода можно произвести компиляцию (перевод программы в машинные коды), в ходе которой система находит *синтаксические* ошибки (*ошибки во время компиляции* – *Compile-time error*).

Наиболее часто встречаются ошибки:

- 3 – Unknown identifier – неизвестный идентификатор (имя переменной или константы не описано в соответствующем разделе);
- 4 – Duplicate identifier – дублирование (повторение) идентификатора;
- 5 – Syntax error – синтаксическая ошибка;
- 14 – Invalid file name – недопустимое имя файла;
- 26 – Type mismatch – несоответствие типов;
- 85 – «;» expected – ожидается символ «;»;
- 215 – Arithmetic overflow error – ошибка выполнения арифметической операции.

Для выполнения компиляции лучше использовать команду **Compile / Make** (клавиша **F9**), так как в этом случае система не требует выполнения программы непосредственно за компиляцией. После исправления всех синтаксических ошибок внизу окна сообщения появится надпись, показанная на рисунке 4.

Compile successful: Press any key

Рисунок 4 – Сообщение об успешном окончании компиляции в *Free Pascal*


Для перехода в режим редактирования достаточно нажать любую клавишу (например **Пробел**).

1.1.5 Выполнение программы. Просмотр результатов

Даже после того, как исправлены все ошибки компиляции, не следует сразу осуществлять старт откомпилированной программы, так как в ней могут быть *ошибки выполнения* (Run Time Error) или же алгоритмические ошибки, приводящие к сбоям во время работы программы или влекущие за собой получение неверных результатов. Рекомендуется сначала выполнить *трассировку* – пошаговое выполнение программы (клавиша **F7** или **F8**). Результаты выполнения каждого шага можно просмотреть в окне результатов, вызываемом нажатием комбинации клавиш **Alt F5**.

После нажатия комбинации клавиш **Ctrl F9** (если ошибок нет) программа будет успешно выполнена. Результаты выполнения программы выводятся на черный экран командной строки, отображаемый комбинацией клавиш **Alt F5**. Возврат в окно редактора (к тексту программы) достигается однократным нажатием любой клавиши. Прервать выполнение программы можно комбинацией клавиш **Ctrl F2**.

1.1.6 Завершение работы в *Free Pascal*

Выход из *Free Pascal* после окончания работы со средой осуществляется командой **File / Exit** (*Файл / Выход*), щелчком мыши на закрывающей кнопке окна приложения  или нажатием комбинации клавиш **Alt X**.

Среда *Free Pascal* не сообщает пользователю о необходимости сохранения последних изменений в файле программы. Поэтому в обязанность пользователя входит сохранение его файла программы перед завершением работы в *Free Pascal*.

1.2 Базовые понятия языка *Pascal*

1.2.1 Алфавит языка программирования *Pascal*

Алфавит языка программирования – это разрешенный для использования (т. е. воспринимаемый компилятором) набор символов, с помощью которых могут быть образованы слова и другие конструкции данного языка. Алфавит языка *Pascal* включает:

- буквы латинского алфавита: строчные от *a* до *z* и прописные от *A* до *Z*,
- знак подчеркивания *_*;
- арабские цифры от 0 до 9;
- шестнадцатеричные цифры *\$0, ..., \$9, \$a, ..., \$f*;

- специальные символы

+ - * / = , ' . : ; < > [] { } () ^ @ \$ #

пары символов, воспринимающиеся компилятором как один символ,

- := <> <= >= (* *) (. .)
- пробел.

1.2.2 Словарь языка программирования *Pascal*

Под *словом* (лексемой, лексической единицей) языка программирования понимают неделимую последовательность символов, имеющую самостоятельный однозначный смысл.

Словарный состав языка *Pascal* – это специальные символы, зарезервированные слова, идентификаторы, константы и др.

Зарезервированные слова применяются для служебных целей, имеют фиксированный смысл и назначение и не могут быть использованы в другом качестве, например, имен констант или переменных. Наиболее часто при составлении программ на языке *Pascal* применяются зарезервированные слова:

and	end	of	string
array	file	or	then
begin	for	procedure	until
case	function	program	uses
const	if	record	uses
div	in	repeat	var
do	label	set	while
downto	mod	shl	with
else	not	shr	xor

Кроме того, используются директивы **absolute**, **index**, **private**, **public**, **export**, **forward**, **virtual** и другие, которые можно, но нежелательно переопределять.

Идентификаторами называют имена констант, переменных, процедур, функций и других объектов программы (меток, типов и т. д.). Различают *стандартные идентификаторы* (наименование стандартных функций, процедур, типов данных) и *идентификаторы пользователя*.

Для идентификаторов существует ряд *ограничений*: они должны начинаться с буквы или знака подчеркивания, включать только буквы, цифры и знак подчеркивания, не содержать пробелы и другие специальные символы. Длина идентификатора не должна превышать 63 символа. Например, можно использовать такие имена:

b, eps, _alpha, task1_3, program_with_massiv и т. п.

Нельзя использовать, например, такие имена:

1progr – начинается с цифры;
task1.3 – содержит специальный недопустимый символ – точку;
task 1 – содержит пробел;
program – совпадает с зарезервированным словом.

Желательно во избежание возможных ошибок не применять имена, латинское начертание которых совпадает с кириллическим. Например,

а, А, aa, с, С, е, Е, М, о, О, Р, Н, К, х, Х

и тому подобные.

1.2.3 Структура простейшей программы

Программа на языке *Pascal* представляет собой набор строк, длина каждой из которых не превышает 127 символов, и в общем виде может быть представлена следующим образом:

```
Program <имя_программы>; {заголовок программы}
Uses <модуль1>,<модуль2>,...; {подключаемые библиотечные модули}
Label <метка1>, <метка2>,...; {раздел описания меток}
Const <имя1>=<значение1>;
<имя2>=<значение2>,...; {раздел описания констант}
Type <имя_типа>=<описание>,...; {описание типов данных пользователя}
Var {раздел описания переменных}
<имя_переменной1>:<тип1>;<имя_переменной2>:<тип2>,...;
Procedure; {объявление процедур пользователя}
Function; {объявление функций пользователя}
Begin
... {раздел операторов}
End.
```

Обязательным является только наличие раздела операторов, который начинается зарезервированным словом **Begin** и заканчивается словом **End** с точкой в конце. Все операторы отделяются друг от друга символом «;» (точка с запятой). Любой из остальных разделов может отсутствовать. Однако все переменные, константы, процедуры, функции пользователя, метки и т. д., используемые в разделе операторов, должны быть указаны в разделе описаний.

В программу могут входить комментарии – фрагменты текста, заключенные в фигурные скобки: {комментарий} или (*комментарий*). Отдельную строку можно превратить в комментарий, вставив перед строкой два знака /, например,

// комментарий.

1.2.4 Константы, переменные, выражения

Основными элементами, из которых конструируется исполняемая часть программы, являются константы, переменные, обращения к функциям. Каждый из этих элементов (*операндов*) характеризуется своим значением и принадлежит к какому-либо типу данных.

Константами называют данные, значения которых не изменяются в процессе выполнения программы. Эти значения известны, например, задаются в разделе описания констант.

В языке *Pascal* константами могут быть целые, вещественные и шестнадцатеричные числа и символы. Используются также строковые, логические константы и др.

Целые числа (положительные и отрицательные) могут быть заданы в диапазоне от -2147483648 до 2147483647.

Вещественные числа могут быть положительными или отрицательными, заданными с использованием десятичной точки, в виде десятичной дроби (в формате с фиксированной точкой) или включать экспоненциальную часть (E – десятичное основание), например:

-34.234; 0.023;

6.762E2 – число $6,762 \cdot 10^2 = 676,2$;

-1.0493E-3 – число $-1,0493 \cdot 10^{-3} = -0,0010493$.

Шестнадцатеричные числа задаются с использованием шестнадцатеричных цифр в диапазоне от \$00000000 до \$ffffffff. Каждому шестнадцатеричному числу предшествует символ \$, например, \$3ac4.

Символьной константой может быть любой символ, заключенный в апострофы, например, 'a', '<' и т. д.

Символ можно задать путем обращения к его коду с предшествующим символом #, например:

#65 – код прописной латинской буквы A;

#92 – код строчной латинской буквы a;

#192 – код прописной буквы А кириллицы;

#224 – код строчной буквы а кириллицы.

Строковая константа – последовательность символов, заключенная в апострофы.

Логические константы – слова true (истина) или false (ложь).

Существуют *зарезервированные* константы, которые можно использовать без предварительного описания, например, константа Pi – ее значение 3.1415926536 (число π), константа Maxint – ее значение 32767 (максимальное двухбайтовое целое число), логические константы true и

false.

Переменная может изменять свои значения в процессе выполнения программы. Это элемент программы, предназначенный для хранения, коррекции и передачи данных внутри нее. Каждая переменная характеризуется именем, значением и типом, к которому она относится. Имя переменной в процессе выполнения программы неизменно. Все переменные, встречающиеся в программе, должны быть описаны в разделе описания переменных. Для того чтобы описать переменную, указывают ее имя и тип.

С помощью знаков операций и скобок из операндов (констант, переменных и обращений к функциям) составляют *выражения*, представляющие собой правила получения новых значений.

Выражения могут быть арифметическими (алгебраическими), отношения и логическими.

Результатом арифметического выражения является число.

Результатом логического выражения является логическое значение true (истина) или false (ложь).

Выражение отношения, которое тоже является логическим, состоит из двух выражений со знаком отношения.

1.2.5 Арифметические и логические операции

При вычислении значения выражения сначала определяются значения входящих в него переменных, констант и функций. Затем выполняются вычисления в зависимости от расстановки скобок и приоритета операций.

В языке *Pascal* допустимы операции, приведенные в таблице 1 в порядке убывания приоритета.

Таблица 1 – Операции в языке *Pascal*

Тип операции	Операция	Описание операции
Унарные	not @	Логическое отрицание Адрес
Мультипликативные	* , / div , mod shr , shl and	Умножение (пересечение множеств), деление Целочисленное деление, остаток от деления Целочисленный сдвиг вправо, влево Логическое И
Аддитивные	+ - or , xor	Сложение (объединение множеств, сцепление строк) Вычитание Логическое ИЛИ, исключающее ИЛИ

Отношения	=, <, >, <=, >=, in	Операции отношения Принадлежность к множеству
-----------	---------------------	--

1.2.6 Данные в программах. Типы данных

Данные в программах можно подразделить:

- на *входные* данные – исходная для обработки информация, вводимая с помощью процедур ввода или передаваемая как параметры;
- *промежуточные* данные – это данные, используемые для вычислений, организации циклов, временного хранения результатов;
- *выходные* данные – информация, получаемая в результате работы программы и выводимая на экран, на печать или другой носитель информации.

Любые данные, используемые в программе, должны относиться к определенному *типу*, характеризующему множество значений (диапазон), которые могут принимать данные, способ представления этих данных в памяти компьютера и множество применимых к ним операций и функций.

Типы данных языка *Pascal* подразделяются на простые (порядковые, вещественные) и структурированные (массивы, записи, строки, множества, файлы).

Кроме того, можно отметить указатели, процедурные типы и объекты.

Наиболее часто используемые типы данных:

- *integer* – целый (целые числа);
- *real* – вещественный (действительные числа);
- *boolean* – логический (*true*, *false*);
- *char* – символьный.

Переменные одного из указанных выше типов могут быть описаны в разделе описания, например, следующим образом:

Var

```
a, n : integer; {переменные a, n – целого типа}
x, y : real; {переменные x, y – вещественного типа}
logic : Boolean; {переменная logic – логического типа}
symb : char; {переменная symb – символьного типа}
```

Порядковые типы данных имеют конечное число значений, которые можно упорядочить, следовательно, с каждым из них можно сопоставить некоторое целое число – порядковый номер значения. К стандартным порядковым типам относят *целые*, *логические* и *символьный* типы данных.

Среди нестандартных различают *перечисляемый* тип и *тип-диапазон*.

Для всех порядковых типов определены функции:

ord(*x*) – порядковый номер величины *x* порядкового типа;

$\text{pred}(x)$ – предыдущее значение величины x порядкового типа;
 $\text{succ}(x)$ – следующее значение величины x порядкового типа;
 $\text{low}(x)$ – наименьшее значение величины данного типа;
 $\text{high}(x)$ – наибольшее значение величины данного типа.

Диапазон значений данных *целых типов* (таблица 2) зависит от того, сколько байтов они занимают в памяти компьютера.

Таблица 2 – Целые типы данных языка *Pascal*

Имя типа	Длина, байт	Диапазон значений
byte	1	0–255
shortint	1	–128–127
word	2	0–65535
integer	2	–32768–32767
longint	4	–2147483648–2147483647

Данные целых типов можно складывать, вычитать, выполнять с ними операции умножения, целочисленного деления **div** (частное при целочисленном делении) и **mod** (остаток при целочисленном делении). К данным целых типов применимы стандартные процедуры и функции, перечисленные в таблице 3.

Таблица 3 – Процедуры и функции, применимые к данным целых типов

Функция	Тип результата	Действие
abs(x)	x	Возвращает модуль числа
chr(x)	char	Возвращает символ по его коду
dec(x[, i])	x	Уменьшает значение x на i , а при отсутствии i – на 1
inc(x[, i])	x	Увеличивает значение x на i , а при отсутствии i – на 1
hi(i)	byte	Возвращает старший байт аргумента
lo(i)	byte	Возвращает младший байт аргумента
odd(x)	boolean	Возвращает true, если аргумент – нечетное число
random(x)	real	Число, равномерно распределенное на (0; x)
sqr(x)	x	Возвращает квадрат аргумента
swap(i)	integer	Меняет местами байты в слове
swap(x)	word	То же

Логический тип данных относится к порядковым типам. Переменные логического типа занимают в памяти один байт и могут принимать только одно из двух значений – true (истина) или false (ложь), причем $\text{true} > \text{false}$.

В языке *Pascal* используются следующие логические операции: **not** (логическое отрицание), **and** (логическое И, логическое умножение), **or** (логическое ИЛИ, логическое сложение) и **xor** (исключающее ИЛИ). В таблице 4 true обозначено как 1, а false – 0.

Таблица 4 – Значения логических операций

Операнды		Значение операции			
x	y	not x	x and y	x or y	x xor y
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

Переменная *символьного типа* занимает в памяти 1 байт, содержимым которого является код символа в диапазоне от 0 до 255 в соответствии с таблицей *ASCII* (American Standard Code For Information Interchange – американский стандартный код обмена информацией). Так как этот код семибитовый, то им кодируются символы от 0 до 127. Символы с номерами от 128 до 255 не ограничены жесткими рамками стандарта, их набор может быть разный на различных компьютерах.

Значение символьной переменной должно быть заключено в апострофы!

Над данными *символьного типа* можно производить операции сравнения в соответствии с их кодами. Кроме функций, действующих над данными всех порядковых типов, с символами используется функция *uppercase* (x), которая заменяет строчную (малую) букву латинского алфавита на прописную (большую).

Наиболее широко в программировании используются вещественные (действительные) числа, являющиеся значениями физических констант и величин.

Переменные *вещественных типов* данных (таблица 5) занимают в памяти от 4 до 10 байт.

Таблица 5 – Вещественные типы данных в языке *Pascal*

Имя типа	Название	Длина, байт	Значащих цифр	Диапазон абсолютных значений
real	Вещественный	6	11–12	2.9E–39...1.7E+38
single	Одинарной точности	4	7–8	1.5E–45...3.4E+38
double	Двойной точности	8	15–16	5.0E–324...1.7E+308
extended	Расширенный	10	19–20	3.4E–4951...1.7E+4932

comp	Целое в формате вещественного	8	19–20	-2E63+1...+2E63-1
------	-------------------------------	---	-------	-------------------

Для данных вещественного типа определены операции сложения, вычитания, умножения, деления и сравнения.

1.3 Программирование линейных алгоритмов

Линейным алгоритмом называют такую алгоритмическую конструкцию, которая реализована в виде последовательности действий, причем каждое действие выполняется один раз.

Чаще всего линейные программы создают и используют для вычисления значений выражений, заданных формулами.

Раздел операторов линейной программы формируется из операторов присваивания и вызова процедур, которые выполняются в том порядке, в котором встречаются в тексте программы.

Не забывайте, операторы отделяются друг от друга точкой с запятой!

1.3.1 Операторы языка программирования *Pascal*

Все операторы языка *Pascal* можно подразделить на простые (присваивания, вызова процедуры, безусловного перехода, пустой) и структурированные (составной, условия, выбора, повтора).

Оператор присваивания. Формат оператора:

<имя переменной> := <выражение>

Принцип действия: вычисляется выражение, находящееся в правой части оператора, и его значение присваивается переменной в его левой части.

Тип выражения в правой части оператора присваивания и переменной в левой его части должны быть совместимы по типу. Например, недопустимо присваивать вещественное число переменной целого или символьного типа.

Если в правой части – алгебраическое выражение, то оно может содержать переменные и константы как целого, так и вещественного типа. При этом тип результата будет вещественным.

Пример 1. Соответствие и несовместимость типов данных в выражениях с оператором присваивания.

```

Program types;
Var x : real;
n : integer;
ch : char;
st : string;
Begin
n:=5;

```

```

x:=n; {правильно, вещественной переменной присваивается целое
      значение}
n:=3.4; {ошибка, целой переменной присваивается вещественное
      значение}
n:=4/2; {ошибка, результат операции деления всегда
      вещественного типа}
n:=cos(x); {ошибка, результатом вычисления функции косинус
      является вещественное число}
ch:='a';
st:=ch; {правильно, строковой переменной присваивается
      символьное значение}
ch:='Pascal'; {ошибка, символьной переменной присваивается
      строковое значение}
n:=ch; {ошибка, целой переменной присваивается символьное
      значение}

```

End.

Процедуры ввода-вывода. *Ввод данных* с клавиатуры во время работы программы реализуется с помощью обращений к процедурам:

```

read(список переменных, перечисленных через запятую)
readln(список переменных, перечисленных через запятую)

```

Принцип действия: выполнение программы приостанавливается, и компьютерная система переходит в режим ожидания (отображается мигающий курсор) ввода пользователем значения переменной или значений переменных. При использовании в программе процедуры ввода `read` пользователь должен вводить значения переменных, отделяя их пробелом, а при использовании процедуры ввода `readln` после ввода каждого значения переменной необходимо нажать клавишу **Enter**.

После ввода пользователем значений всех указанных в списке процедуры ввода переменных выполнение программы возобновляется, а введенные значения присваиваются этим переменным.

Для *вывода информации* (сообщений или результатов работы программы) используют обращение к процедурам:

```

write(список переменных, констант, выражений через запятую)
writeln(список переменных, констант, выражений через запятую)

```

Например, при обращении к процедуре

```
writeln('x=', x, ' sin x =', sin(x))
```

выводятся последовательно строковая константа из двух символов `x=`

(без кавычек и апострофов), значение переменной x , строковая константа $\sin x =$ (без кавычек и апострофов, первый символ пробел) и вычисленное значение выражения $\sin(x)$.

Принцип действия: значения констант, переменных и выражений выводятся на экран (в файл, на печать), причем при использовании в программе процедуры `write` курсор остается в конце текущей строки, и следующие выходные данные будут печататься в той же строке, а при использовании процедуры `writeln` курсор переходит на новую строку.

Примечание – Вызов процедуры `writeln` без параметров используется для перехода на новую строку или для пропуска строки.

Процедуры `write` и `writeln` предоставляют возможность *форматирования выходных данных*, т.е. представления результатов в удобном для пользователя виде (F-формат, формат с фиксированной точкой). Минимальное количество позиций, которое должно занимать при выводе значение *целого* или *строкового типа* указывается форматом `:n`, например:

`writeln(k:5)` – для отображения переменной k целого или строкового типа при выводе выделяется 5 позиций;

`writeln(a, ' ':3, b)` – значения переменных a и b при выводе разделяются тремя пробелами.

Количество позиций, отводимое при выводе *для вещественного числа*, указывается форматом `:n:m`, где n – общее число позиций, m – число позиций для дробной части, например:

`writeln(y:6:2)` – для отображения вещественной переменной y при выводе отводится 6 позиций, из них 2 – после десятичной точки, причем одну позицию занимает символ точка, т.е. значение $y = 23.17445$ будет выведено как `23.17` с пробелом впереди числа.

Форматирование результатов не является обязательным. Но тогда для данных целого типа при выводе выделяется 15 позиций, а для данных вещественного типа – 18 позиций, что часто бывает чрезмерным. При этом вещественные числа представляются в E-формате (другими словами, в экспоненциальной форме, формате с плавающей точкой, в виде мантиссы и порядка), например:

`-3.456000000E+02`

Такие числа иногда трудны для восприятия и могут сливаться друг с другом, что является недочетом, иногда приводящим к неправильному толкованию результата. Поэтому рекомендуется не только форматировать результаты при выводе, но и отделять значения выводимых переменных и выражений пробелами. Например, при выводе значения переменной $p = 2,74$ и выражения y , зависящего от p (получено значение $y = 16,570824$), есть несколько способов, представленных в таблице 6.

Таблица 6 – Форматирование выходных данных в языке *Pascal*

Фрагмент кода	Вывод на экран (в файл)
<code>writeln(p,y)</code>	2.7400000000E+00 1.6570824E+01
<code>write('p=',p:5:2,'y=',y:8:4)</code>	p= 2.74y= 16.5708
<code>write('p=',p:4:2,' y=',y:6:4)</code>	p=2.74 y=16.5708

Составной оператор объединяет в одно целое группу операторов, которые после этого могут считаться одним оператором. Свое применение он находит в структурированных операторах ветвления и цикла. Составной оператор начинается зарезервированным словом **begin**, затем следуют операторы, перечисленные через точку с запятой, и оканчивается зарезервированным словом **end**.

Пара служебных (ключевых) слов **begin** и **end** называется *операторными скобками*.

Составной оператор может включать другой составной оператор, причем следует помнить о правильном вложении пар операторных скобок, например:

```
begin...
begin...
begin
...
end;
...;
end;
end;
```

Оператор безусловного перехода. Операторы в программе выполняются в том порядке, в котором они записаны. Чтобы изменить последовательность выполнения операторов, необходимо использовать оператор безусловного перехода **goto**, действие которого состоит в передаче управления оператору, снабженному меткой, который может находиться в любой строке раздела операторов программы.

Формат оператора безусловного перехода:

goto <метка>

Метку можно установить перед любым оператором основной программы (кроме операторов в составе составного), отделив её двоеточием:

<метка> : <оператор>

В качестве метки можно использовать как имя, предложенное пользователем, так и число от 0 до 9999. Все метки, встречающиеся в

программе, надо предварительно указать в разделе описаний после зарезервированного слова **Label** (см. структуру простейшей программы на языке *Pascal*):

Label <список меток>;

Оператор безусловного перехода рекомендуется использовать только для обработки исключений (аварийных ситуаций, ошибок).

1.3.2 Основные стандартные процедуры и функции

Для построения вычисляемых выражений, используемых в правой части оператора присваивания или в качестве фактических параметров при обращении к процедурам, в частности, к процедурам вывода `write` или `writeln`, можно применить ряд стандартных встроенных функций, большинство из которых приведено в таблице 7.

Используя в программном коде стандартные функции, следует помнить, что *аргумент всегда заключается в круглые скобки!*

Таблица 7 – Стандартные математические функции

<i>Pascal</i> -код	Математическая запись	Тип аргумента	Тип результата	Действие
<code>abs (x)</code>	$ x $	integer, real	integer, real	Модуль (абсолютное значение) числа x
<code>sqr (x)</code>	x^2	integer, real	integer, real	Квадрат числа x
<code>sqrt (x)</code>	$\sqrt{x}, x \geq 0$	integer, real	real	Квадратный корень из числа x
<code>sin (x)</code>	$\sin x$	integer, real	real	Синус числа x
<code>cos (x)</code>	$\cos x$	integer, real	real	Косинус числа x
<code>arctan (x)</code>	$\arctg x$	integer, real	real	Арктангенс числа x
<code>ln (x)</code>	$\ln x, x > 0$	integer, real	real	Натуральный логарифм числа x
<code>exp (x)</code>	e^x	integer, real	real	Экспонента числа x
<code>trunc (x)</code>	–	real	integer	Отбрасывает дробную часть x
<code>round (x)</code>	–	real	integer	Округляет число x
<code>int (x)</code>	–	real	real	Целая часть числа x
<code>frac (x)</code>	–	real	real	Дробная часть числа x

В языке *Pascal* традиционно не поддерживаются такие распространенные математические функции как тангенс, котангенс, арксинус, арккосинус, степенная функция, десятичный логарифм и другие функции.

Для записи математических функций, отсутствующих в перечне встроенных, используются известные математические формулы, приведенные в таблице 8.

Таблица 8 – Представление некоторых функций через встроенные функции

Математическая функция	Математическая формула	Запись на языке <i>Pascal</i>
$\operatorname{tg} x$	$\frac{\sin x}{\cos x}$	$\sin(x) / \cos(x)$
$\operatorname{ctg} x$	$\frac{\cos x}{\sin x}$	$\cos(x) / \sin(x)$
$\arcsin x$	$\operatorname{arctg} \frac{x}{\sqrt{1-x^2}}, -1 < x < 1$	$\arctan(x / \operatorname{sqrt}(1 - \operatorname{sqr}(x)))$
$\arccos x$	$\operatorname{arctg} \frac{\sqrt{1-x^2}}{x}, 0 < x \leq 1$	$\arctan(\operatorname{sqrt}(1 - \operatorname{sqr}(x)) / x)$
$\operatorname{arcctg} x$	$\operatorname{arctg} \frac{1}{x}, x > 0$	$\arctan(1/x)$
a^x	$e^{x \ln a}, a > 0$	$\exp(x * \ln(a))$
$\lg x$	$\frac{\ln x}{\ln 10}$	$\ln(x) / \ln(10)$
$\log_a x$	$\frac{\ln x}{\ln a}$	$\ln(x) / \ln(a)$
$\operatorname{sh} x$	$\frac{e^x - e^{-x}}{2}$	$(\exp(x) - \exp(-x)) / 2$
$\operatorname{ch} x$	$\frac{e^x + e^{-x}}{2}$	$(\exp(x) + \exp(-x)) / 2$

1.3.3 Запись алгебраических выражений

Как упоминалось ранее, алгебраические выражения составляются из

констант, переменных, обращений к функциям с помощью знаков арифметических операций и скобок. При этом следует соблюдать ряд правил и рекомендаций.

1 Необходимо учитывать приоритет операций (см. таблицу 1): сначала выполняется умножение (знак $*$) и деление (знак $/$), затем сложение и вычитание.

Все знаки арифметических операций, даже те, которые при корректной математической записи ставить не принято (например, как при записи произведения $ab = a \cdot b$), необходимо указывать при составлении выражения на *Pascal*. Изменить порядок выполнения операций можно с помощью скобок, как показано в таблице 9.

Таблица 9 – Учет приоритета операций посредством скобок

Математическое выражение	Запись на языке <i>Pascal</i>	Возможные ошибки
$\frac{a-b}{c+d}$	$(a-b) / (c+d)$	Отсутствие скобок изменит смысл выражения: $a-b/c+d \sim a - \frac{b}{c} + d$
$\frac{\sin x + y}{xy}$	$(\sin(x)+y) / (x*y)$	1 Пропущенный знак умножения означает указание на новую (возможно, несуществующую) переменную xy : $(\sin(x)+y) / xy$ 2 Неправильно расставленные или пропущенные скобки изменяют смысл выражения: $(\sin x + y) / (x * y) \sim$ введено новое имя $\sin x$; $(\sin(x) + y) / x * y \sim \frac{\sin x + y}{x} \cdot y$ – изменен смысл

2 Аргументы сложных функций (функция функции) необходимо заключать в скобки (таблица 10).

Таблица 10 – *Pascal*-код сложных функций

Математическое выражение	Запись на языке <i>Pascal</i>	Возможные ошибки
$\sqrt{ \sin x }$	$\text{sqrt}(\text{abs}(\sin(x)))$;	Потеря скобок или неправильный порядок расположения функций
$\ln^2 2x$	$\text{sqr}(\ln(2*x))$;	Потеря скобок, неправильный порядок следования функций, потеря знака умножения
$e^{2\cos^2 x}$	$\text{exp}(2*\text{sqr}(\cos(x)))$;	Неправильный порядок следования функций, потеря знака умножения или скобок

3 При построении на языке *Pascal* сложных арифметических выражений целесообразно использовать вспомогательные переменные. Например, при кодировании формулы

$$f = \frac{1 + \operatorname{tg}^2 x}{1 - 2 \operatorname{tg}^2 x} \quad (1)$$

можно ввести промежуточную переменную *f1* и только после этого записать оператор присваивания для вычисления *f*:

```
f1:=sin(x)/cos(x);
f:=(1+sqr(f1))/(1-2*sqr(f1));
```

4 Для записи на языке *Pascal* экспоненциальной функции e^x , где $e \approx 2,718281828$ – основание натурального логарифма, используется ее представление в виде $\exp(x)$. Ряд примеров приведен в таблице 11:

Таблица 11 – *Pascal*-код экспоненциальной функции

Математическое выражение	Запись на языке <i>Pascal</i>
e^{a+b}	$\exp(a+b)$
e	$\exp(1)$
$\frac{1}{e}$	$\exp(-1)$

5 Определенные трудности возникают при записи степенных и показательных выражений, например, x^7 , $\sqrt[3]{x}$ или a^x . *Pascal*-код в таких случаях составляется с учетом формул элементарной математики:

$$a^x = e^{x \ln a}, \quad (2)$$

$$\sqrt[x]{a} = a^{\frac{1}{x}}. \quad (3)$$

Примеры приведены в таблице 12.

Таблица 12 – *Pascal*-код степенных и показательных выражений

Математическое выражение	Альтернативная математическая запись	Запись на языке <i>Pascal</i>
$(a+b)^{\frac{x}{2}}$	$e^{\frac{x}{2} \ln(a+b)}$	$\exp(x/2 * \ln(a+b))$
$\sqrt[3]{\frac{a}{b}}$	$e^{\frac{1}{3} \ln \frac{a}{b}}$	$\exp(1/3 * \ln(a/b))$

6 Необходимо учитывать область определения функций, к которым в выражении происходит обращение. Такие ошибки легче предотвратить,

чем исправлять, так как они не обнаруживаются компилятором. Например, для выражения x^{17} , записанного на языке *Pascal* в виде

$$\exp(17 * \ln(x)),$$

при выполнении программы возникнет ошибка при отрицательных значениях переменной x . Поэтому его лучше записать в форме

$$\text{sqr}(\text{sqr}(\text{sqr}(\text{sqr}(x)))) * x,$$

исключающей подобную ошибку.

Два других примера представлены в таблице 13.

Таблица 13 – Учет области определения функций

Математическое выражение	Запись на языке <i>Pascal</i>	Альтернативная запись на языке <i>Pascal</i>
$\sqrt[8]{x} = x^{\frac{1}{8}}$	$\exp(1/8 * \ln(x));$	$\text{sqrt}(\text{sqrt}(\text{sqrt}(x)))$
$\sqrt[5]{x}$	$\exp(1/5 * \ln(x));$	<pre> if x>0 then y:= exp(1/5*ln(x)) else if x<0 then y:= - exp(1/5*ln(abs(x))) else y:=0; </pre>

7 Не следует забывать о скорости выполнения операций и рациональной записи выражений. Арифметические операции в порядке возрастания их времени выполнения располагаются так: сложение и вычитание, умножение и деление. Поэтому в некоторых случаях целесообразно заменять умножение сложением или использовать функции или процедуры вместо операций сложения и умножения. Например, выражение $x*x$ лучше записать как $\text{sqr}(x)$, а вместо оператора присваивания $x:=x+1$ использовать обращение к процедуре $\text{inc}(x)$.

Пример 2. Запишите на языке *Pascal* выражение $2x^4 - 3x^3 + 5x^2 - x + 7$.

Решение.

$$2x^4 - 3x^3 + 5x^2 - x + 7 \sim x * (x * (x * (2 * x - 3) + 5) - 1) + 7.$$

1.4 Программирование разветвляющихся алгоритмов

Разветвляющимся алгоритмом называется такая алгоритмическая конструкция, которая обеспечивает выбор между несколькими альтернативами в зависимости от условия или значений входных данных.

1.4.1 Логические выражения

Выражение называется *логическим*, если результатом его вычисления

является логическое значение `true` (истина) или `false` (ложь). Логические выражения в языке *Pascal* служат для записи условий выполнения определенных операторов или поиска необходимых данных и чаще всего встречаются в операторах ветвления и цикла.

Простые логические выражения могут содержать константы, переменные и выражения сравнимого типа, соединенные между собой знаками операций отношения (`<`, `>`, `<=`, `>=`, `<>`). Из простых логических выражений составляются более сложные с использованием логических операций **not**, **and**, **or**, **xor** (см. таблицу 1). Приоритет выполнения логических операций следующий:

- 1) **not**;
- 2) **and**;
- 3) **or**; **xor**.

Изменение приоритета логических операций достигается применением скобок. В таблице 14 приведено несколько примеров записи логических выражений на *Pascal*.

Таблица 14 – Логические выражения

Логическое выражение	Запись на языке <i>Pascal</i>
Оба числа a и b положительны	$(a > 0)$ and $(b > 0)$
Хотя бы одно из чисел a и b положительно	$(a > 0)$ or $(b > 0)$
Только одно из чисел a и b положительно	$(a > 0)$ xor $(b > 0)$
$x \in [a; b]$	$(x >= a)$ and $(x <= b)$
$x \notin [a; b]$	$(x < a)$ or $(x > b)$ или not $((x >= a)$ and $(x <= b))$
$x \notin [a; b]$ и $x \neq c$	$((x < a)$ or $(x > b))$ and $(x <> c)$
Число x принадлежит одному из интервалов $[a; b]$ или $[c; d]$	$(x >= a)$ and $(x <= b)$ or $(x >= c)$ and $(x <= d)$

1.4.2 Оператор условного перехода **if**. Оператор выбора **case**

Для реализации ветвления на *Pascal* используются условный оператор **if** и оператор множественного выбора **case**.

Условный оператор (оператор альтернативы) – это структурированный оператор, предназначенный для выделения из составляющих его операторов одного, который и выполняется в дальнейшем.

Оператор **if**, как и остальные структурированные операторы, часто включает в себя составные операторы.

Формат условного оператора (*полная форма*):

if <логическое выражение> **then** <оператор 1> **else** <оператор 2>;

где **if** означает *если*, **then** – *то* или *тогда*, **else** – *иначе*.

Принцип действия: вычисляется значение логического выражения (другими словами, проверяется условие). Если значением логического выражения является **true** (истина), то выполняется оператор, следующий после слова **then**, т. е. оператор 1. Если значением логического выражения является **false** (ложь), то выполняется оператор, следующий после слова **else**, которым является оператор 2.

Можно использовать *краткую* форму условного оператора:

if <логическое выражение> **then** <оператор>;

Принцип действия: вычисляется значение логического выражения, т. е. проверяется условие. Если значением логического выражения является **true** (истина), то выполняется <оператор>, следующий после слова **then**. Если значением логического выражения является **false** (ложь), то в этом случае <оператор> пропускается, а управление передается тому оператору, который следует за условным.

Примечания

1 Если в зависимости от условия предполагается выполнить несколько операторов, то соответствующие операторы заключаются в операторные скобки **begin** и **end**, т. е. организуется составной оператор.

2 Перед зарезервированными словами **then** и **else** знак **;** никогда не ставится, т. к. конструкция **if..then..else..** является единым структурированным оператором.

Пример 3. Составьте программу для вычисления значения выражения при различных значениях переменной **a1**:

$$z(a1) = \begin{cases} \lg^2 a1 & \text{if } a1 > 0; \\ \frac{\text{ctg } a1}{\text{ctg}^3 a1 + 5} & \text{if } a1 < 0; \\ 0 & \text{if } a1 = 0. \end{cases} \quad (4)$$

Решение.

```
Program task_if;
```

```
Var
```

```
a1 : real; {описание переменной}
```

```
z, z1 : real; {описание вспомогательной и  
вычисляемой переменной}
```

Begin

```
write('Введите значение a1'); {Вывод подсказки}
read(a1); {ввод значения переменной a1}
if a1>0
then z:=sqr(ln(a1)/ln(10))
else
if a1<0
then begin z1:=cos(a1)/sin(a1); {расчет котангенса}
z:=z1/(sqr(z1)*z1+5)
end
else z:=0;
writeln('a1=',a1:3:2,' z=',z:6:3)
End.
```

Оператор множественного выбора case позволяет выбрать одно из нескольких возможных продолжений программы. Параметром, по которому осуществляется выбор, служит *ключ выбора* – выражение любого порядкового типа (кроме longint).

Формат оператора выбора:

```
case <ключ_выбора> of
<список_выбора 1> : <оператор 1>;
<список_выбора 2> : <оператор 2>;
...
<список_выбора N> : <оператор N>;
[else <операторы>]
end;
```

где **case**, **of**, **else**, **end** – зарезервированные слова (в переводе означающие *выбор, из, иначе, окончание* соответственно);

<ключ_выбора> – имя переменной или выражение порядкового типа (кроме longint);

<список_выбора 1>, ..., <список_выбора N> – константа или список констант того же типа, что и выражение <ключ_выбора>;

<оператор 1>, ..., <оператор N> – операторы языка *Pascal*, каждый из которых может быть простым, составным и структурированным оператором.

Принцип действия оператора **case**: вычисляется значение выражения

<ключ_выбора>, затем в последовательности <список_выбора> отыскивается тот, которому принадлежит константа, равная вычисленному значению ключа выбора. Оператор, соответствующий выбранному списку, выполняется, остальные – игнорируются. После этого оператор выбора **case** завершает свою работу. Если в списке выбора не будет найдена константа, соответствующая вычисленному значению ключа выбора, управление передается оператору, стоящему за словом **else**. Часть **else** <операторы> может и отсутствовать.

Пример 4. Составить программу, организующую ввод значений x и y , и, в зависимости от выбора пользователя, отображающую на экране значение суммы, разности, произведения или частного от деления x и y .

Решение.

```
Program Calculus;  
Label m1;  
Var  
operation : char; {знак операции}  
x, y, z : real; {операнды и результат}  
Begin  
writeln('Введите переменные x,y= '); readln(x,y);  
write('Введите знак операции: '); readln(operation);  
case operation of  
  '+' : z:=x+y;  
  '-' : z:=x-y;  
  '*' : z:=x*y;  
  '/' : z:=x/y  
else begin  
    writeln('Неверный ввод');  
    goto m1;  
  end;  
end;  
writeln ('Результат=', z:8:3);  
m1:End.
```

1.5 Организация циклических вычислений

Большинство вычислительных процессов имеют следующую особенность: отдельные этапы вычислений повторяются многократно, при этом всякий раз используются новые значения при вычислении. Повторяющийся этап вычислений называют телом *цикла*, а вычислительный процесс – циклическим.

Виды задач, программируемых с помощью циклических конструкций:

- табулирование функции (построение таблиц значений функции при

различных значениях аргумента);

- вычисление суммы ряда;
- нахождение предела последовательности;
- отделение и уточнение корней уравнения;
- вычисление значения интеграла и некоторые другие задачи.

Различают циклы с известным и неизвестным числом повторений.

Циклы с известным числом повторений (арифметические циклы) характеризуются заданием начального и конечного параметров цикла. Закон изменения параметров цикла: условием окончания такого цикла является достижение параметром цикла значения, большего конечного.

1.5.1 Циклы с неизвестным числом повторений (итерационные циклы)

Итерационным называется вычислительный процесс, в котором для определения последующего значения переменной используется ее предыдущее значение. В итерационных циклах реализуется метод последовательных приближений. К итерационным циклам относят *цикл с предусловием* и *цикл с постусловием*.

1.5.2 Цикл с предусловием `while..do`

Цикл с предусловием реализуется на языке *Pascal* с помощью оператора `while..do`. Количество повторений такого цикла заранее не определено и зависит от входных данных задачи. В данной циклической конструкции перед выполнением следующего повторения цикла проверяется значение логического выражения.

Формат оператора `while..do`:

```
while <условие> do <тело_цикла>;
```

где **while**, **do** – служебные слова (*пока, выполнять*);

<условие> – выражение логического типа (например, $b \leq 5$);

<тело_цикла> – тело цикла, состоящее из оператора языка *Pascal*, который может быть простым, составным или структурированным.

Принцип действия оператора `while..do`.

1 Вычисляется значение логического выражения (проверяется условие).

2 Если результатом вычисления значения логического выражения (результатом проверки условия) является значение `true` (истина), то выполняется оператор, составляющий *тело цикла* и стоящий после служебного слова **do**. После этого происходит возврат к п. 1.

3 Если результатом вычисления значения логического выражения является значение `false` (ложь), то оператор `while..do` заканчивает свою работу и управление передается следующему в коде программы оператору.

Согласно описанному алгоритму повторение п. 1 и 2 продолжается до тех

пор, пока результатом вычисления значения логического выражения (результатом проверки условия) не станет значение `false` (ложь). Заметим, что если значением логического выражения сразу является `false`, то тело цикла оператора `while..do` не выполняется ни разу.

Примечание – Тело цикла обязательно должно содержать оператор, действие которого влияет на значение логического выражения (изменяет значения входящих в него операндов). Иначе неизбежна ситуация «зацикливания», проявляющаяся как бесконечное повторение выполнения тела цикла.

Пример 5. Составить, используя оператор цикла с предусловием, программу для вычисления и вывода на экран значений функции

$$y = \cos\left(\frac{5^x}{|x+2|}\right) \quad (5)$$

при значениях переменной x , изменяющихся в диапазоне от -3 до 6 с шагом $0,3$.

Решение.

```
Program prim_w;  
Var x, y: real;  
Begin  
x:=-3; {левая граница диапазона изменения переменной}  
while x<=6 do {условием выхода из цикла ставится  
достижение переменной x правой границы диапазона}  
begin  
y:=cos(exp(x*ln(5))/abs(x+2));  
writeln('x=',x:5:2,' y=',y:7:3);  
x:=x+0.3 {увеличение значения переменной x на величину шага}  
end;  
End.
```

1.5.3 Цикл с постусловием `repeat..until`

Цикл с постусловием, в котором сначала выполняется оператор или группа операторов, а затем проверяется условие, реализуется на языке *Pascal* с помощью структурированного оператора `repeat..until`.

Формат оператора:

```
repeat  
<операторы>  
until <условие>;
```

где **repeat**, **until** – служебные слова (*повторять*, до тех пор, пока);

<операторы> – последовательность операторов (*тело цикла*).

Использовать в качестве тела цикла операторные скобки **begin** и **end** нецелесообразно, т. к. роль скобок в данном случае играют зарезервированные слова **repeat** и **until**;

<условие> – выражение логического типа.

Принцип действия оператора **repeat..until**.

1 Выполняются операторы, расположенные между операторами **repeat..until**, составляющие тело цикла.

2 Вычисляется значение логического выражения – проверяется условие. Если результатом вычисления значения логического выражения является значение **false** (ложь), то операторы, составляющие тело цикла, выполняются вновь.

3 Если результатом вычисления значения логического выражения (результатом проверки условия) является значение **true** (истина), то оператор **repeat..until** заканчивает свою работу, и управление передается следующему в коде программы оператору.

Как следует из описанного алгоритма, повторение действий 1 и 2 продолжается до тех пор, пока результатом вычисления логического выражения не станет значение **true** (истина), т. е. пока не выполнится условие.

Примечания

1 Тело цикла **repeat..until** обязательно должно содержать оператор, действие которого влияет на значение логического выражения (изменяет значения входящих в него операндов). Иначе неизбежна ситуация «зацикливания», при которой происходит бесконечное повторение выполнения тела цикла.

2 Тело цикла оператора **repeat..until** выполняется хотя бы один раз независимо от начального значения логического выражения.

Пример 6. Составить, используя оператор цикла с постусловием, программу для вычисления и вывода на экран значений функции

$$y = \cos\left(\frac{5^x}{|x+2|}\right) \quad (6)$$

при значениях x , изменяющихся в диапазоне от -3 до 6 с шагом $0,3$.

Решение.

```
Program prim_r;  
Var x, y: real;  
Begin  
x:=-3; {левая граница диапазона изменения переменной}
```

repeat

```
y:=cos(exp(x*ln(5))/abs(x+2));  
writeln('x=',x:5:2,' y=',y:7:3);  
x:=x+0.3 {увеличение значения переменной x на шаг}  
until x>6 {условием выхода из цикла является превышение  
переменной x значения правой границы диапазона}  
End.
```

1.5.4 Цикл с параметром for..to..do

Цикл с заранее известным числом повторений может быть реализован на языке *Pascal* с помощью оператора цикла с параметром **for..to..do**.

Формат оператора:

```
for <параметр цикла> := <нач_знач> to <кон_знач>  
do <оператор>;
```

где **for**, **to**, **do** – служебные слова (*для*, *до*, *выполнить*);

<параметр цикла> – параметр цикла, в качестве которого выступает переменная порядкового типа;

<нач_знач> – начальное значение параметра цикла: константа, переменная или выражение порядкового типа;

<кон_знач> – конечное значение параметра цикла: константа, переменная или выражение порядкового типа;

<оператор> – простой, составной или структурированный оператор (*тело цикла*).

Принцип действия оператора **for..to..do**.

1 Параметру цикла присваивается начальное значение, и выполняются операторы тела цикла.

2 Параметр цикла увеличивается на единицу, т. е. <параметр_цикла> := <нач_знач> + 1. Операторы тела цикла выполняются во второй раз.

3 После каждого увеличения параметра цикла на единицу происходит выполнение операторов тела цикла.

4 В последний раз тело цикла выполняется при достижении параметром цикла его конечного значения, т. е. когда <параметр_цикла> = <кон_знач>.

Таким образом, операторы тела цикла **for..to..do** выполняются <кон_знач> – <нач_знач> + 1 раз.

Для случаев, когда конечное значение параметра цикла меньше начального, в языке *Pascal* предусмотрен оператор **for..downto..do**.

Формат оператора:

```
for <параметр цикла> := <нач_знач> downto <кон_знач>  
do <оператор>;
```

В этом случае шаг изменения параметра цикла отрицательный, равный минус 1.

Пример 7. Составить, используя оператор цикла с известным числом

повторений, программу для вычисления и вывода на экран значений функции

$$y = \cos\left(\frac{5^x}{|x+2|}\right) \quad (7)$$

при x , изменяющемся в диапазоне от -3 до 6 с шагом $0,3$.

Решение.

```
Program prim_f;  
Var x,y: real; i,n: integer;  
Begin  
x:=-3; {левая граница диапазона изменения переменной}  
n:=round((6-(-3))/0.3)+1; {количество повторений}  
for i:=1 to n do  
  begin  
    y:=cos(exp(x*ln(5))/abs(x+2));  
    writeln('x=',x:5:2, ' y=',y:7:3);  
    x:=x+0.3  
  end  
End.
```

2 Порядок выполнения лабораторной работы

1 Создайте с помощью *Total Commander* посредством нажатия клавиши **F7** на диске **z:** папку, предназначенную для хранения файлов программ. В имени папки (не более 8 символов) используйте латинские буквы и цифры, но не пробелы и точки. Данную папку принято называть личной.

2 Загрузите интегрированную среду (IDE) *Free Pascal*, *PascalABC* или *Turbo Pascal*. Для этого выполните:

для *Free Pascal* **Start / All Programs / Free Pascal** (*Пуск / Все программы / Free Pascal*);

для *PascalABC* **Start / All Programs / PascalABC / PascalABC** (*Пуск / Все программы / PascalABC / PascalABC*).

Можно также дважды щелкнуть на соответствующем ярлыке, который находится на рабочем столе *Windows*.

3 Создайте новый файл, выполнив команду **File / New** (*Файл / Новый*). Откроется окно нового документа. При этом файлу назначается автоматически имя `noname01.pas` (`Program1.pas`), отображаемое в верхней части окна документа как показано на рисунках 5 и 6.

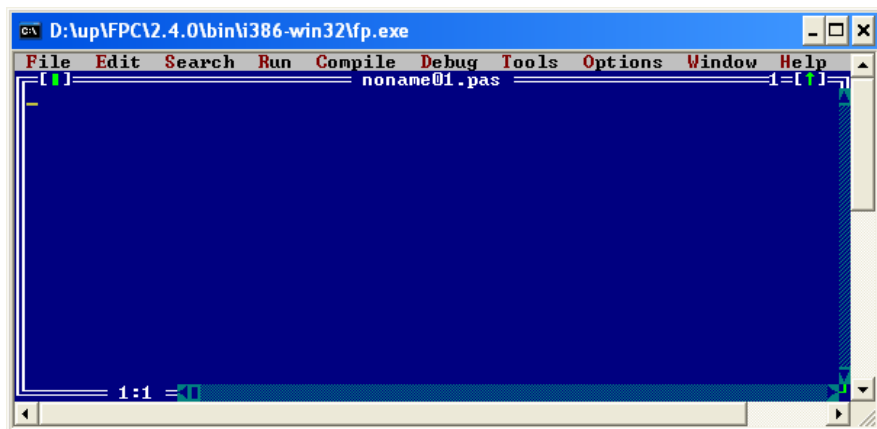


Рисунок 5 – Окно нового документа в среде *Free Pascal*

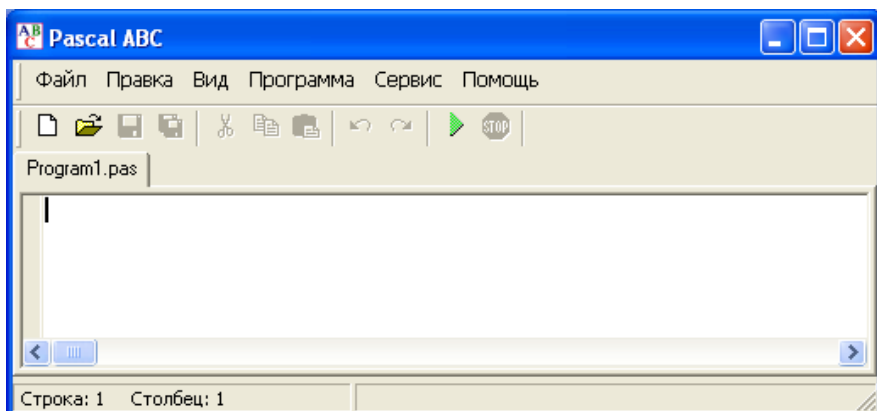


Рисунок 6 – Окно нового документа в *PascalABC*

4 Измените имя файла и одновременно сохраните новый файл в личной папке на диске **z:**. Для этого выполните команду **File / Save as** (*Файл / Сохранить как*) и в появившемся окне наберите имя диска, вложенные каталоги, имя файла, например, при работе в *Free Pascal* путь к файлу вводится в поле **Name** так, как показано на рисунке 7.



Рисунок 7 – Диалоговое окно **Save File as** в *Free Pascal*

После этого щелкните на кнопке **ОК**. Периодически сохраняйте файл в процессе работы (клавиша **F2**).

5 Наберите текст линейной программы, вычисляющей расстояние между двумя точками $(x_1; y_1)$ и $(x_2; y_2)$ по формуле

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} . \quad (8)$$

```

Program prim_1; {имя программы}
Var x1, x2, y1, y2, d: real;
    {описание переменных вещественного типа}
begin {начало раздела операторов}
write('Введите координаты первой точки x1 и y1');
readln(x1, y1); {ввод значений переменных x1, y1
                через нажатие клавиши Enter}
write('Введите координаты второй точки x2 и y2');
readln(x2, y2); {ввод значений переменных x2, y2
                через нажатие клавиши Enter}
d:=sqrt(sqr(x1-x2)+sqr(y1-y2)); {расчет расстояния}
writeln('Расстояние между точками равно ',d:5:2);
    {вывод значения расстояния}
End.

```

6 Произведите компиляцию программы, нажимая клавишу **F9**.

7 Если появляется сообщение об ошибках, исправьте их и запустите программу на выполнение посредством нажатия комбинации клавиш **Ctrl F9**.

8 Чтобы посмотреть результат, нажмите комбинацию клавиш **Alt F5**.

Выполните задание 1.

Задание 1. Составьте линейную программу, реализующую решение задачи (таблица 15). Сохраните файл в своей папке под именем **task21.pas**. Откомпилируйте программу. При наличии ошибок исправьте их, запустите программу на выполнение и просмотрите результаты. Листинг программы скопируйте в окно документа текстового процессора *Word* или *WordPad* и распечатайте.

Таблица 15 – Варианты задания 1

Вариант	Задача
1	Определить координаты середины отрезка АВ по формулам $x = \frac{x_1 + x_2}{2}$, $y = \frac{y_1 + y_2}{2}$, где значения координат точек А($x_1; y_1$) и В($x_2; y_2$) вводятся с клавиатуры
2	Расчитать площадь треугольника по формуле Герона

	$S = \sqrt{p(p-a)(p-b)(p-c)}$, где длины трех сторон a, b, c вводятся с клавиатуры, а p – полупериметр
3	Рассчитать площадь треугольника $S = \frac{ah_a}{2}$ по стороне a и проведенной к ней высоте h_a , значения которых вводятся с клавиатуры
4	Найти биссектрису $w_a = \frac{2}{b+c} \sqrt{bcp(p-a)}$, проведенную к стороне a треугольника, по трем сторонам a, b, c , длины которых вводятся с клавиатуры, и полупериметру p
5	Найти биссектрису $w_b = \frac{2}{a+c} \sqrt{acp(p-b)}$, проведенную к стороне b треугольника, по трем сторонам a, b, c , длины которых вводятся с клавиатуры, и полупериметру p

Окончание таблицы 15

Вариант	Задача
6	Найти биссектрису $w_c = \frac{2}{a+b} \sqrt{abp(p-c)}$, проведенную к стороне c треугольника, по трем сторонам a, b, c , длины которых вводятся с клавиатуры, и полупериметру p
7	Найти радиус $R = \frac{abc}{4S}$ описанной около треугольника окружности, где длины сторон a, b и c вводятся с клавиатуры, а площадь S вычисляется по формуле Герона (см. вариант 2)
8	Найти радиус $r = \frac{S}{p}$ окружности, вписанной в треугольник со сторонами a, b, c , длины которых вводятся с клавиатуры, площадь S вычисляется по формуле Герона (см. вариант 2), а p – полупериметр
9	Найти высоту $h_a = \frac{2S}{a}$, опущенную на сторону a треугольника со сторонами a, b, c , длины которых вводятся с клавиатуры, а площадь S вычисляется по формуле Герона (см. вариант 2)
10	Найти высоту $h_b = \frac{2S}{b}$, опущенную на сторону b треугольника со сторонами a, b, c , длины которых вводятся с клавиатуры, а площадь S вычисляется по формуле Герона (см. вариант 2)

11	Найти высоту $h_c = \frac{2S}{c}$, опущенную на сторону c треугольника со сторонами a, b, c , длины которых вводятся с клавиатуры, площадь S вычисляется по формуле Герона (см. вариант 2)
12	Вычислить площадь $S = \frac{a+b}{2}h$ равнобокой трапеции с основаниями a и b , длины которых вводятся с клавиатуры, острым углом $\alpha = \frac{\pi}{3}$ и высотой $h = \frac{ b-a }{2} \operatorname{tg} \alpha$

9 Создайте новый файл: **File / New** (*Файл / Новый*). Сохраните его в личной папке на диске **z:** под именем **fi022.pas**. Рекомендуется периодически сохранять файл в процессе работы посредством нажатия клавиши **F2**.

10 Наберите текст программы для вычисления значений составной функции

$$f = \begin{cases} x^3 - \sin x, & x \in [-1; 3]; \\ \sqrt[3]{\cos x}, & x \notin [-1; 3]. \end{cases} \quad (9)$$

```

Program prim_2; {имя программы}
Var {раздел описания переменных}
x : real; {описание переменной-аргумента}
f : real; {описание вычисляемой переменной, т. е. функции}
Begin {раздел операторов}
write('Введите значение x');
    {вывод вспомогательной фразы}
read(x); {ввод значения переменной x}
if (x>=-1) and (x<=3) {условие}
    then f:=sqr(x)*x-sin(x) {если условие истинно}
    else {если условие ложно}
        begin
            if cos(x)>0 {расчет кубического корня из косинуса x}
                then f:=exp(1/3*ln(cos(x)))
            else
                begin
                    if cos(x)<0
                        then f:=-exp(1/3*ln(abs(cos(x))))
                    else f:=0;
                end;
        end;

```

```

end;
writeln('x=', x:5:2, ' f=', f:6:3) {вывод результатов}
End.

```

11 Произведите компиляцию программы (клавиша **F9**). При наличии ошибок исправьте их. Запустите программу на выполнение (**Ctrl F9**) и просмотрите результаты (**Alt F5**).

12 Выполните задание 2.

Задание 2. Составьте программу, вычисляющую значение функции в зависимости от значения вводимой с клавиатуры переменной (таблица 16). Откомпилируйте программу. При наличии ошибок исправьте их, запустите программу на выполнение и просмотрите результаты. Листинг программы скопируйте в окно документа текстового процессора *Word* и распечатайте или запишите в отчет о выполнении лабораторной работы.

13 Создайте новый файл: **File / New (Файл / Создать)**. Сохраните его в личной папке на диске **z:** под именем **fi023.pas**. Периодически сохраняйте файл в процессе работы (клавиша **F2**).

Таблица 16 – Варианты задания 2

Вариант	Функция	Вариант	Функция
1	$f_{un1}(a) = \begin{cases} a^2 - \ln(\cos a), & \cos a > 0; \\ \cos^3 a, & \cos a \leq 0 \end{cases}$	7	$f_{un2}(b) = \begin{cases} b^3 - \ln b^2, & b \neq 0; \\ 0, & b = 0 \end{cases}$
2	$f_2(x_2) = \begin{cases} \ln(\sin x_2), & \sin x_2 > 0,5; \\ \cos^2 x_2, & \sin x_2 \leq 0,5 \end{cases}$	8	$z_1(c) = \begin{cases} \frac{e^c}{2c^2 + 1}, & e^c > 50; \\ 2ce^c, & e^c \leq 50 \end{cases}$
3	$y_1(x) = \begin{cases} \arctg^2 x - \sqrt[3]{x}, & x > 0; \\ \ln\left(\frac{e^x}{ x+0,1 }\right), & x \leq 0 \end{cases}$	9	$z_2(x) = \begin{cases} \frac{\cos^2 x}{x^3 - x^2}, & x > 0; \\ 3xe^x + \sin 2x, & x \leq 0 \end{cases}$
4	$y_2(a) = \begin{cases} \arcsin^2 a - 1, & a \in [-1; 1]; \\ \sin^4 a, & a \notin [-1; 1] \end{cases}$	10	$z(x) = \begin{cases} e^{2\cos x}, & \cos x \leq 0; \\ \ln(\cos x), & \cos x > 0 \end{cases}$
5	$z_3(c) = \begin{cases} \arctg(\cos c^2), & c^2 > 5; \\ \sin^2 c + \cos^4 3c, & c^2 \leq 5 \end{cases}$	11	$f(y) = \begin{cases} \arccos y, & y \in [-1; 1]; \\ 2\sin^3 3y, & y \notin [-1; 1] \end{cases}$

6	$fun3(z) = \begin{cases} \lg(\operatorname{tg} z) + \operatorname{tg}^2 2z, \operatorname{tg} 2z > 0; \\ \cos^4 z + e^{-z}, \operatorname{tg} 2z \leq 0 \end{cases}$	12	$fun4(x) = \begin{cases} \cos^2 5x, x \geq 0; \\ \operatorname{ctg}^2 2x, x < 0 \end{cases}$
---	--	----	--

14 Наберите в окне редактора **fiо23.pas** текст программы табулирования функции, т. е. вычисления значения функции

$$fn(vr) = \frac{\operatorname{tg}^3 vr + c}{\sqrt{vr^2 + c^2}} \quad (10)$$

при изменении аргумента vr в диапазоне от -1 до 4 с шагом $0,4$, где $c = 12$. Используйте для реализации повторений вычисления функции цикл **while..do** с предусловием.

```

Program prim_3;
Var a, b, c : integer; {описание целых переменных}
vr, hvr, fn1, fn: real; {описание вещественных переменных}
Begin
a:=-1;b:=4; {границы диапазона изменения переменной}
hvr:=0.4; {шаг изменения переменной vr}
c:=12; {значение константы c}
vr:=a; {переменной vr присваивается начальное значение}
while vr<=b do {условием выхода из цикла требуется достижение
                переменной vr правой границы диапазона}
begin {составной оператор}
fn1:=sin(vr)/cos(vr); {расчет тангенса vr}
fn:=(sqr(fn1)*fn1+c)/sqr(sqr(vr)+sqr(c));
    {расчет значения функции fn}
writeln('vr=',vr:5:2,' fn=',fn:7:3);
    {вывод текущего значения переменной vr и функции fn}
vr:=vr+hvr {увеличение значения переменной vr на величину шага}
end;
End.

```

15 Произведите компиляцию программы (клавиша **F9**). При наличии ошибок исправьте их. Запустите программу на выполнение (**Ctrl F9**) и просмотрите результаты (**Alt F5**).

16 Создайте новый файл: **File / New (Файл / Создать)**. Сохраните его в личной папке на диске **z:** под именем **fiо24.pas**. Периодически сохраняйте файл в процессе работы, нажимая клавишу **F2**.

17 Наберите в окне редактора **fio24.pas** текст программы табулирования функции, заданной формулой (10), при изменении аргумента vr в диапазоне от -1 до 4 с шагом $0,4$, где $c = 12$. Используйте для организации повторений при вычислении функции цикл **repeat..until** с постусловием.

```
Program prim_4;  
Var a, b, c : integer; {описание целых переменных}  
vr, hvr, fn1, fn: real; {описание вещественных переменных}  
Begin  
a:=-1;b:=4; {границы диапазона изменения переменной}  
hvr:=0.4; {шаг изменения переменной vr}  
c:=12; {значение переменной c}  
vr:=a; {переменной vr присваивается начальное значение}  
repeat {начало цикла с постусловием}  
fn1:=sin(vr)/cos(vr); {расчет тангенса vr}  
fn:=(sqr(fn1)*fn1+c)/sqrt(sqr(vr)+sqr(c));  
{расчет значения функции fn}  
writeln('vr=',vr:5:2,' fn=',fn:7:3);  
    {вывод текущего значения переменной vr и функции fn}  
vr:=vr+hvr {увеличение значения переменной vr на величину шага}  
until vr>b {условием выхода из цикла является превышение  
    переменной vr значения правой границы диапазона}  
End.
```

18 Откомпилируйте программу (клавиша **F9**). При наличии ошибок исправьте их. Запустите программу на выполнение (**Ctrl F9**) и просмотрите результаты (**Alt F5**).

19 Создайте новый файл: **File / New** (*Файл / Создать*). Сохраните его в личной папке на диске **z:** под именем **fio25.pas**.

20 Наберите в окне редактора **fio25.pas** текст программы табулирования функции, заданной формулой (10), при изменении аргумента vr в диапазоне от -1 до 4 с шагом $0,4$ ($c = 12$). Используйте для организации повторений при вычислении функции цикл **for..to..do** с известным числом повторений.

```
Program prim_5;  
Var a, b, c : integer; {описание целых переменных}  
vr, hvr, fn1, fn: real; {описание вещественных переменных}  
Begin  
a:=-1;b:=4; {границы диапазона изменения переменной}
```

```

hvr:=0.4; {шаг изменения переменной vr}
c:=12; {значение переменной c}
vr:=a; {переменной vr присваивается начальное значение}
n:=round((b-a)/hvr)+1; {количество повторений}
for i:=1 to n do
Begin
fn1:=sin(vr)/cos(vr); {расчет тангенса vr}
fn:=(sqr(fn1)*fn1+c)/sqrt(sqr(vr)+sqr(c));
    {расчет значения функции fn}
writeln('vr=',vr:5:2,'      fn=',fn:7:3);
    {вывод текущего значения переменной vr и функции fn}
vr:=vr+hvr {увеличение значения переменной vr на величину шага}
end
End.

```

21 Откомпилируйте программу (клавиша **F9**). При наличии ошибок исправьте их. Запустите программу на выполнение (**Ctrl F9**) и просмотрите результаты (**Alt F5**).

22 Выполните задание 3.

Задание 3. Составить три варианта программы табулирования функции – вычисления значений функции при изменении аргумента в заданном диапазоне с постоянным шагом (таблица 17) – с использованием каждого из трех операторов цикла. Произвести компиляцию, получить и просмотреть результаты. Текст каждой из программ скопировать в *Word* и распечатать или записать в отчет о выполнении работы.

Таблица 17 – Варианты задания 3

Вариант	Диапазон, шаг	Значение параметра	Функция
1	$x \in [-3; 4], \Delta x = 0,3$	$a = 4$	$f1(x) = x^3 - \sin^2 \frac{5x}{a}$
2	$y \in [-2; 5], \Delta y = 0,45$	$b = -2$	$f2(y) = y^2 + \text{ctg}^2 by + b$
3	$z \in [-3; 3], \Delta z = 0,35$	$c = 6$	$f3(z) = \frac{1}{c} \sin^2 \frac{1-z^3}{1+z}$
4	$x \in [1; 6], \Delta x = 0,25$	$b = 0,4$	$a1(x) = bx - \text{ctg}^3 \frac{x}{x+b}$
5	$y \in [-2; 2], \Delta y = 0,3$	$b = -8$	$a2(y) = y^3 + b^2 \ln \left \frac{1}{y+b} \right $

6	$z \in [-3; 5], \Delta z = 0,5$	$b = 3,2$	$a3(z) = \frac{\cos^2 bz}{z+b} - z^3$
7	$x \in [-5; 2], \Delta x = 0,6$	$w = -2$	$p1(x) = \ln \left \frac{x^6 + w}{\cos^3 x} \right $
8	$y \in [2; 9], \Delta y = 0,3$	$v = 6$	$p2(y) = -\frac{2v}{\sqrt[4]{y^3 - v}}$
9	$t \in [1; 7], \Delta t = 0,3$	$a1 = 0,5$	$w1(t) = \ln^2 \frac{t - a1}{\sqrt{5\pi + t}}$
10	$w \in [-4; 5], \Delta w = 0,7$	$p = 3$	$z1(w) = -\frac{p}{4w^2} + \frac{\sin^2 w}{w - p}$
11	$t \in [2; 6], \Delta t = 0,3$	$q = 14$	$w2(t) = 3 \sqrt[3]{\frac{q - \ln 2t}{q + \ln 3t}}$
12	$q \in [-3; 4], \Delta q = 0,35$	$a = 5$	$fun(q) = \frac{1}{a} \log_3 a + q^3 $

Контрольные вопросы

- 1 Назначение языка программирования *Pascal*.
- 2 Состав среды программирования.
- 3 Алфавит языка программирования *Pascal*.
- 4 Резервированные слова, стандартные идентификаторы и идентификаторы пользователя. Правила создания имен и ограничения.
- 5 Структура простейшей программы. Из каких разделов состоит программа на языке *Pascal*? Какие из разделов необязательны? Для чего служит раздел описания переменных, раздел операторов?
- 6 Переменные и константы в языке *Pascal*. Имя и значение переменной. Данные в программах. Типы данных. Наиболее часто используемые типы данных.
- 7 Порядковые типы данных: целые, символьный, логический.
- 8 Вещественные типы данных.
- 9 Операции, процедуры и функции, применимые к порядковым типам данных.
- 10 Стандартные процедуры и функции языка *Pascal*. Основные математические функции.
- 11 Как записать математические функции, которым нет соответствия на *Pascal*?
- 12 Операции языка *Pascal*. Приоритет операций.
- 13 Правила записи арифметических выражений.
- 14 Операторы языка *Pascal*. Оператор присваивания. Формат и принцип действия. Согласование типов. Составной операторы.
- 15 Процедуры ввода-вывода. Формат и принцип действия. Форматирование результатов.
- 16 Логические выражения. Логические операции. Приоритет логических операций.

17 Оператор условного перехода. Полная и краткая формы. Использование составного оператора. Оператор выбора. Тип ключа выбора.

18 Основные циклические конструкции. Формат и принцип действия оператора с предусловием **while..do**. Особенности применения.

19 Формат и принцип действия оператора с постусловием **repeat..until**. Особенности применения.

20 Формат и принцип действия оператора **for..to..do**. Особенности применения.

САМОСТОЯТЕЛЬНАЯ РАБОТА 2

ОБРАБОТКА МАССИВОВ НА ЯЗЫКЕ *PASCAL*. ПРОЦЕДУРЫ И ФУНКЦИИ ПОЛЬЗОВАТЕЛЯ

Цель работы: углубить знания о возможностях интегрированной среды программирования *Free Pascal*, *PascalABC* или *Turbo Pascal*, развить навыки отладки программы и анализа результатов ее работы, научиться программировать задачи с использованием одномерных массивов.

1 Сведения из теории

1.1 Структурированный тип данных массив

Структурированные данные представляют собой элементы, которые в свою очередь являются данными простых или структурированных типов. В языке *Pascal* различают следующие структурированные типы данных: массивы, строки, записи, множества, файлы.

Массив в *Pascal* представляет собой конечный набор однотипных объектов (чисел, символов, строк и т. п.), имеющих общее имя и хранящихся в последовательно размещенных ячейках памяти. Использование массивов позволяет хранить элементы векторов и матриц в памяти компьютера с теми же обозначениями, что и в математических записях; организовать циклические вычисления с различными исходными данными; уменьшить объем программ, сделать их более наглядными и надежными.

Размер массива – это количество элементов, содержащихся в нем. Каждый элемент массива занимает определенное местоположение, определяемое порядковым номером, который называется *индексом*. Количество индексов, необходимое для точного определения положения элемента, называется *размерностью* массива. В языке *Pascal* различают одно-, двух- и многомерные массивы. Наиболее широко используются *одномерные* массивы, соответствующие математическому представлению *векторов*, и *двумерные*, соответствующие прямоугольным таблицам – *матрицам*.

1.1.1 Описание массивов

При описании массива указывают его имя, размерность и диапазон изменения каждого индекса, а также тип входящих в массив элементов.

Формат описания массива:

<имя массива> : **array**[диапазон индексов] **of** <тип элементов>

Можно декларировать массив-константу, тип данных массив или переменную-массив, например:

```
Const mas_const = (2.3, 4, 6.4, -2, -3.7);
```

```
{массив из пяти вещественных элементов-констант}
```

```
Var
```

```
D : array[1..10] of integer; {одномерный массив-переменная  
из десяти целочисленных элементов}
```

```
В : array[0..50,0..20] of char; {двумерный массив-переменная,  
размером 51 × 21, из символьных элементов}
```

```
С : array[-3..4] of Boolean; {одномерный массив-переменная  
из восьми элементов логического типа}
```

```
Type norm_mas = array[1..10,1..10] of integer;
```

```
{описание типа данных пользователя – массив 10 × 10 из целых чисел}
```

```
Var D1, D2 : norm_mas; {двумерные переменные-массивы типа  
norm_mas}
```

Над массивами целиком определена операция присваивания, например, оператор вида

```
D1:=D2
```

означает, что всем элементам массива D1 присвоены значения элементов массива D2.

Могут быть также использованы логические выражения вида

```
D1=D2
```

и

```
D1<>D2.
```

Все остальные операции с массивами осуществляются поэлементно.

Формат обращения в программе к конкретному элементу массива:

```
<имя массива> [<номер элемента>]
```

Например,

D[1] – обращение к первому элементу одномерного массива D,

D[i] – обращение к *i*-му элементу одномерного массива D;

D[2*i+1] – обращение к элементу одномерного массива D, номер которого является значением выражения 2*i+1.

W[i, j] – обращение к элементу, расположенному на пересечении *i*-й строки и *j*-го столбца в двумерном массиве W,

W[1, 5] – обращение к элементу, являющемуся пятым в первой строке двумерного массива W,

W[3, j] – обращение к элементу третьей строки и *j*-го столбца.

1.1.2 Ввод элементов массива

Для работы с массивами широко используют циклы и вложенные циклы. Ввод элементов одномерного массива D размером n можно кодировать следующим образом:

```
for i:=1 to n do read(D[i]);
```

Выполняется указанный фрагмент кода в случае ввода значений-чисел следующим образом:

- 1) параметру цикла (целочисленной переменной i) присваивается начальное значение (единица);
- 2) выполнение программы приостанавливается в ожидании ввода пользователем значений элементов массива;
- 3) после ввода пользователем строки из n чисел, разделенных пробелом, и нажатия клавиши **Enter** выполнение программы возобновляется, и введенные числа заносятся в качестве значений в массив D .

Вариантом кода, реализующего ввод значений элементов массива, является конструкция

```
for i:=1 to n do readln(D[i]);
```

Она выполняется так:

- 1) параметру цикла (целочисленной переменной i) присваивается начальное значение (единица);
- 2) выполнение программы приостанавливается в ожидании ввода пользователем первого элемента массива (обращение к процедуре ввода `readln(D[i]);`);
- 3) после ввода пользователем значения первого элемента массива, который может быть целым или вещественным числом, символом или строкой (в зависимости от того, как описан массив D в разделе описаний программы), необходимо нажать клавишу **Enter**, что возобновит выполнение программы – введенное пользователем значение станет значением первого элемента массива D ;
- 4) параметру цикла i присваивается следующее значение (два). Далее повторяются шаги 2–4 алгоритма.

Когда значение параметра цикла достигнет конечного (станет равным n), и пользователь введет значение последнего элемента массива и нажмет **Enter**, цикл `for..to..do` завершит свою работу и управление перейдет следующему оператору.

Ввод элементов массива – многошаговая процедура. Поэтому следует позаботиться о том, чтобы в процессе выполнения программы были введены корректные данные. Один из способов повышения надежности ввода – вывод на экран подсказки в виде строки текста, дающего представление о

количестве и типе элементов массива, например:

```
n:=7; {количество элементов массива}
writeln('Введите ',n,' целочисленных элементов
        массива D ');
for i:=1 to n do
begin
write('D[' ,i, ']=');
readln(D[i])
end;
```

Первый раз приостановка выполнения программы произойдет тогда, когда на экране монитора отобразится подсказка

```
D[1]=
```

Далее необходимо набрать значение *первого* элемента массива и нажать клавишу **Enter**. Подобная подсказка будет появляться каждый раз перед тем, как программа требует ввода очередных данных.

В языке *Pascal*, реализованном в *Free Pascal*, присвоение элементам массива заданных списком значений допустимо также и при описании массива. Например,

```
Var
D : array[1..10] of integer =
(3,5,8,12,7,4,6,34,3,43);
```

Для двумерного массива W , состоящего из $n \times m$ элементов, простейший фрагмент кода, реализующего ввод элементов массива, имеет вид:

```
for i:=1 to n do
for j:=1 to m do read(W[i,j]);
```

При выполнении этого кода сначала требуется ввести с клавиатуры значения элементов первой строки, затем второй и т. д. Завершается ввод нажатием клавиши **Enter**.

1.1.3 Вывод элементов массива

Вывод одномерного массива тоже производится поэлементно с помощью, например, цикла **for..to..do**:

```
for i:=1 to n do write(D[i], ' ');
```

или

```
for i:=1 to n do writeln('D[' ,i, ']=' ,D[i]);
```

Посредством первого оператора цикла значения элементов массива D , разделенные пробелами, выводятся в строку.

Например,

3 5 8 12 7 4 6 34 3 43

Второй оператор цикла выводит значения элементов массива D в виде 10 строк:

```
D[1]=3
D[2]=5
D[3]=8
D[4]=12
D[5]=7
D[6]=4
D[7]=6
D[8]=34
D[9]=3
D[10]=43
```

Вывести двумерный массив можно в строку и в столбец. Но наиболее информативным является представление в форме прямоугольной таблицы. Код, реализующий такой способ вывода, имеет вид:

```
for i:=1 to n do
begin
for j:=1 to m do write(W[i,j]);
  {вывод элементов  $i$ -й строки через пробел}
  writeln {переход на новую строку}
end;
```

1.1.4 Базовые алгоритмы обработки массивов

Кроме рассмотренных выше операций ввода (заполнения) и вывода (визуализации) элементов массива часто выполняют суммирование, нахождение произведения, поиск элементов массива, удовлетворяющих некоторым условиям. Каждое из перечисленных действий предусматривает использование циклических конструкций.

Вычисление суммы и произведения элементов массива. При нахождении *суммы*:

- 1) переменной суммирования S присваивается начальное значение, равное *нулю*;
- 2) при первом выполнении тела цикла к текущему значению переменной суммирования S прибавляется первое слагаемое суммы, и результат присваивается переменной суммирования S , т. е.

$S := S + \langle \text{слагаемое} \rangle;$

3) при каждом следующем выполнении тела цикла производится аналогичное действие.

Данный алгоритм при вычислении суммы элементов одномерного массива D размера n реализуется в виде следующего кода:

```
S:=0; {начальное значение суммы}
for i:=1 to n do S:=S+D[i]; {вычисление суммы}
writeln('S=', S); {вывод суммы}
```

Для программирования нахождения *произведения*:

1) переменной произведения P присваивается начальное значение, равное *единице*;

2) при первом выполнении тела цикла текущее значение переменной произведения P умножается на первый сомножитель произведения, и результат присваивается переменной произведения P , т. е.

$P := P * \langle \text{множитель} \rangle;$

3) при каждом следующем выполнении тела цикла производится аналогичное действие.

Данный алгоритм при вычислении произведения значений элементов одномерного массива D размера n реализуется в виде следующего кода:

```
P:=1; {начальное значение произведения}
for i:=1 to n do P:=P*D[i]; {вычисление произведения}
writeln('P=', P); {вывод произведения}
```

Более сложный код применяется для вычисления произведения элементов двумерного массива W размером $n \times m$:

```
P:=1; {начальное значение произведения}
for i:=1 to n do
for j:=1 to m do P:=P*W[i, j]; { вычисление произведения}
writeln('P=', P); {вывод произведения}
```

Поиск максимального и минимального элементов массива. При нахождении минимального элемента массива:

1) скалярной переменной минимума, предназначенной для хранения максимального значения, присваивается значение первого элемента массива, выступающего в качестве начального приближения или гипотезы;

2) в цикле текущее значение переменной сравнивается с каждым

элементом массива и переопределяется, если рассматриваемый элемент массива меньше текущего значения переменной минимума.

Код, реализующий данный алгоритм для одномерного массива D , состоящего из n элементов, имеет вид:

```
min:=D[1]; {первый элемент массива назначается минимальным}  
for i:=2 to n do  
if D[i]<min then min:=D[i];  
{текущий минимальный элемент сравнивается с каждым элементом  
массива и переопределяется при выполнении условия}  
writeln('min=',min); {вывод минимального элемента}
```

При нахождении максимального элемента массива:

1) скалярной переменной максимума, предназначенной для хранения максимального значения, присваивается значение первого элемента массива;

2) в цикле текущее значение переменной сравнивается с каждым элементом массива и переопределяется, если текущее значение элемента массива больше текущего значения переменной максимума.

Программный код, реализующий данный алгоритм для одномерного массива D , состоящего из n элементов, имеет вид:

```
max:=D[1]; {первый элемент массива назначается минимальным}  
for i:=2 to n do  
if D[i]>max then max:=D[i]; {текущий максимальный элемент  
сравнивается с каждым элементом массива и переопределяется при  
выполнении условия}  
writeln('max=',max); {вывод максимального элемента}
```

Максимальный элемент двумерного массива W , состоящего из $n \times m$ элементов, определяется с использованием вложенного цикла:

```
max:=W[1, 1]; {первый элемент массива назначается максимальным}  
for i:=1 to n do  
for j:=1 to m do  
if W[i, j]>max then max:=W[i, j]; {текущий максимальный  
элемент сравнивается с каждым элементом массива и переопределяется при  
выполнении условия}  
writeln('max=',max); {вывод максимального элемента}
```

Подсчет количества элементов массива, удовлетворяющих заданным условиям. Чтобы определить количество элементов, имеющих заданные свойства, необходимо:

1) переменной-счетчику k присвоить начальное значение, равное нулю

(k:=0);

2) в цикле вычислить значение логического выражения. Если это значение равно True, увеличить значение счетчика на единицу, т. е. выполнить оператор k:=k+1 или inc(k).

Данный алгоритм реализуется кодом

```
k:=0;  
if <условие> then k:=k+1;
```

где <условие> – условие, которому должны соответствовать элементы массива.

Программа, определяющая количество элементов одномерного массива, модуль которых не превосходит числа 3, представлена ниже.

```
Program count1;  
Var  
D: array[1..10] of integer;  
{описание одномерного массива из десяти целых чисел}  
i, k: integer;  
begin  
for i:=1 to 10 do readln(D[i]);  
{ввод десяти элементов одномерного массива}  
k:=0; {начальное значение счетчика}  
for i:=1 to 10 do  
if abs(D[i])<=3 then k:=k+1;  
{значение счетчика увеличивается на единицу при выполнении условия}  
writeln('k=', k);  
{количество элементов, соответствующих условию}  
End.
```

1.1.5 Решение задач обработки одномерных массивов

Пример 1. Ввести одномерный массив M из пятнадцати элементов и число v . Для каждого элемента массива вычислить

$$MN_i = \sqrt{|\operatorname{tg} M_i|} - \frac{v^2 - \log_2 i}{2 + 2 \sin M_i}. \quad (1)$$

Вывести полученный массив. Поменять местами второй и максимальный элементы массива MN . Найти среднее арифметическое элементов модифицированного массива MN , индексы которых при делении на три дают остаток 2. Заменить полученным значением отрицательные элементы массива MN . Если отрицательных элементов нет, вывести соответствующее


```

if (i mod 3)=2 then
begin k:=k+1; {значение счетчика увеличивается на единицу,
                если остаток от деления номера i на 3 равен 2}
S:=S+MN[i]
{суммирование элементов, соответствующих условию}
end;
Average:=S/k; {расчет среднего арифметического}
writeln('Average=',Average:5:2);
{вывод среднего арифметического}
k:=0; {начальное значение счетчика}
for i:=1 to 15 do
if MN[i]<0 then
begin
k:=k+1;
MN[i]:=Average;
end;
if k>0 then
for i:=1 to 15 do writeln('MN[' ,i, ']=' ,MN[i]:5:2)
else writeln('Отрицательных элементов нет');
End.

```

Задание 1. Составить программу решения задачи из таблицы 1. Результаты вычислений распечатать.

Таблица 1 – Задачи по обработке одномерных массивов

Вариант	Задача
1	Ввести одномерный массив D из семи элементов и число b . Для каждого элемента массива вычислить $Z_i = \sqrt{2D_i} + b \operatorname{tg} D_i$. Вывести полученный массив. Найти сумму элементов массива Z с нечетными индексами. Подсчитать количество отрицательных элементов Z
2	Ввести одномерный массив B из восьми элементов и число x . Для каждого элемента массива вычислить $G_i = \operatorname{ctg}^2 B_i - x\sqrt{2B_i}$. Вывести полученный массив. Найти сумму элементов массива G с четными индексами. Подсчитать количество положительных элементов G
3	Ввести одномерный массив C из десяти положительных элементов и число a . Для каждого элемента массива вычислить $Y_i = a^2 \sqrt{3C_i} + \ln^2 C_i$. Вывести полученный массив. Найти максимальный элемент Y . Подсчитать количество элементов Y , больших числа 5
4	Ввести одномерный массив M из девяти элементов и число w . Для

	каждого элемента массива вычислить $X_i = \sqrt[3]{ w - M_i } + \sin^2 M_i$. Вывести полученный массив. Найти минимальный элемент X . Подсчитать количество элементов X , значение которых больше двух
5	Ввести одномерный массив D из шести элементов и число t . Для каждого элемента массива вычислить $Z_i = t^3 + \lg^2 D_i$. Вывести полученный массив и найти среднее арифметическое его положительных элементов

Окончание таблицы 1

Вариант	Задача
6	Ввести одномерный массив D из пятнадцати элементов и число b . Для каждого элемента массива вычислить $V_i = \frac{b^2 \cos \sqrt{ D_i + 1 }}{b + 0,3}$. Вывести полученный массив. Заменить значение первого элемента массива V значением минимального элемента этого массива
7	Ввести одномерный массив F из одиннадцати элементов и число m . Для каждого элемента массива вычислить $G_i = \sqrt{ \sin F_i } + \frac{m^2 F_i}{2}$. Вывести полученный массив. Найти среднее арифметическое элементов массива G , больших числа четыре. Определить количество таких элементов
8	Ввести одномерный массив F из десяти элементов и число k . Для каждого элемента массива вычислить $G_i = k^3 + \frac{k \sin F_i}{\sqrt{F_i + 1}}$. Вывести полученный массив. Найти сумму отрицательных элементов массива G . Заменить модулем полученного значения элементы с четными индексами
9	Ввести одномерный массив H из тринадцати элементов и число a . Для каждого элемента массива вычислить $K_i = a \sqrt{ \cos H_i } - \frac{H_i^3}{a + 2,5}$. Вывести полученный массив. Найти максимальный отрицательный элемент массива K и заменить второй элемент массива K найденным значением
10	Ввести одномерный массив D из десяти элементов и число b . Для каждого элемента массива вычислить $R_i = \frac{\operatorname{tg}^2 D_i - \operatorname{ctg} b D_i}{2b + 1}$. Вывести полученный массив. Поменять местами минимальный и максимальный элементы R
11	Ввести одномерный массив U из десяти элементов и число n . Для каждого элемента массива вычислить $D_i = \frac{\operatorname{ctg} U_i}{n + 1} - \ln n U_i $. Вывести полученный массив. Определить произведение отрицательных элементов массива D . Поменять местами значения последнего и минимального элементов D

12	<p>Ввести одномерный массив G из четырнадцати элементов и число h. Для каждого элемента массива вычислить $Y_i = 4\sqrt{ h + G_i^2 } + \sin 2h$. Вывести полученный массив. Найти среднее арифметическое элементов Y, значения которых по модулю не превосходят числа 2. Определить среди них минимальный элемент</p>
----	---

1.2 Процедуры и функции

Процедуры и функции пользователя (*подпрограммы*) – это самостоятельные, законченные по смыслу фрагменты программы, имеющие имя и предназначенные для однократного и многократного выполнения в ходе основной программы.

Процедура описывается между разделом описаний и разделом операторов основной программы.

Формат описания:

```

procedure <имя>(<список формальных параметров>);
<раздел описаний процедуры>
begin
<раздел операторов процедуры>
end;

```

В *списке формальных параметров* перечисляются переменные и указываются их типы.

Различают *параметры-аргументы* (входные данные для процедуры) и *параметры-результаты* (для возвращения в основную программу результатов работы процедуры), причем перед списком переменных-результатов каждого типа указывается слово **Var**.

Например, для многократных вычислений и вывода значения функции $f(x) = \sqrt{|x|} - \sin x$ можно описать процедуру, где x – формальный параметр-аргумент, a – формальный параметр-результат.

```

Procedure prim(x:real; var f:real);
begin
f:=sqrt(abs(x))-sin(x);
writeln('f(',x:4:2,')=' ,f:7:4);
end;

```

Структура *раздела описаний процедуры* аналогична разделу описаний основной программы. Перечисленные в разделе объекты носят локальный характер (например, *локальные переменные*) и могут использоваться только в той процедуре, в которой описаны.


```

Procedure Print; {процедура вывода элементов массива}
begin
  {цикл для вывода 10 элементов массива}
  for i:=1 to 10 do writeln('M[' , i, ']=' , M[i]:5:2);
end;
begin {основная программа}
  writeln('Введите 10 элементов массива');
  for i:=1 to 10 do {цикл для ввода 10 элементов M}
  begin
    write('M[' , i, ']=' );
    readln(M[i]); {ввод элемента с текущим индексом i}
  end;
  Print; {вызов процедуры вывода}
  for i:=1 to 10 do M[i]:=sin(M[i]);
  Print; {вызов процедуры вывода}
  for i:=1 to 10 do
  if M[i]<0 then M[i]:=1;
  Print; {вызов процедуры вывода}
End.

```

Задание 2. Составить программу решения варианта задачи из таблицы 1 с использованием процедур пользователя (для ввода, вывода элементов массивов, выполнения базовых операций суммирования, вычисления произведений, нахождения минимального элемента и т. д.).

Другая разновидность подпрограмм – *функция пользователя* – возвращает в точку вызова только один параметр-результат простого стандартного типа. В общем виде функцию описывают перед разделом операторов, после описания процедур пользователя (если таковые имеются):

```

Function <имя функции>
  (<список формальных параметров-аргументов>) :
  <тип функции (результата)>;
  <раздел описаний функции>
begin
  <раздел операторов функции, содержащий оператор присваивания>
end;

```

В *раздел операторов функции* должен обязательно входить оператор, который присваивает значение необходимого выражения имени функции:

```

<имя функции>:=<выражение>

```

Вызов функции из основной программы возможен *только из выражения* (например, в операторе присваивания или операторе вызова *процедуры*)

путем указания имени функции и фактических параметров-аргументов, заключенных в круглые скобки:

<имя функции>(<список фактических параметров>)

При этом вычисленное значение функции возвращается в точку вызова функции в программе так же, как возвращается значение любой стандартной (встроенной) функции, например, $\sin(x)$.

Пример 3. Вычислить значение выражения

$$r = \frac{\operatorname{tg} \frac{x}{2} + \operatorname{ctg}^2 x}{1 + 2 \operatorname{tg} 3x^2}. \quad (2)$$

Решение.

В данной задаче часто используется функция тангенс с разными аргументами, отсутствующая в языке *Pascal*. Поэтому для вычисления тангенса целесообразно описать и использовать функцию пользователя.

```
Program r1;
Var x, r: real;
{описание функции с формальным параметром z}
Function tg(z:real):real;
begin
  tg:=sin(z)/cos(z);
end;
Begin
  writeln('Введите значение x');
  read(x);
  {при вызове функции тангенс используются фактические параметры,
   т. е. z заменяется на x}
  r:=(tg(x/2)+sqr(1/tg(x)))/(1+2*tg(3*sqr(x)));
  writeln('x=',x:4:1,' r=',r:5:2);
End.
```

Пример 4. Вычислить сумму ряда

$$S(x) = \sum_{n=0}^m \frac{(2n)!}{(n!)^2 (2n+1)} \cdot \left(\frac{x}{2}\right)^{2n+1} \quad (3)$$

при положительных значениях переменной x .

Решение.

Вычисление факториала – неоднократно повторяющаяся операция при решении данной задачи. Целесообразно описать и использовать в основной

программе функцию пользователя для вычисления факториала.

```

Program sum1;
Var x, un, S: real;
i, n: integer;
{описание функции с формальным параметром m}
Function fact(m: integer):real;
Var j:integer;
begin
fact:=1;
for j:=1 to n do fact:=fact*j;
end;
Begin {основная программа}
writeln('Введите число x > 0');
read(x);
writeln('Введите число слагаемых ряда');
read(n);
un:=x/2; S:=un;
{начальные значения степенного выражения и суммы ряда}
for i:=1 to m-1 do
begin
un:=un*sqr(x/2);
S:=S+un*fact(2*i)/(sqr(fact(i))*(2*i+1))
{вычисление текущих значений степенного выражения и суммы ряда}
end;
writeln('S=', S:6:4);
End.

```

Задание 3. Составить программу для вычисления суммы или произведения ряда при произвольных значениях переменной x (таблица 2).

Таблица 2 – Суммы и произведения

Вариант	Ряд	Вариант	Ряд
1	$\sum_{n=0}^m \frac{x^{4n}}{(4n)!}$	7	$\sum_{n=0}^m \frac{x^{2n} (n!)^2}{(n)!}$
2	$\sum_{n=1}^m \frac{((n-1)!)^2}{(2n)!} (2x)^{2n}$	8	$\sum_{n=1}^m n! \left(\frac{x}{n}\right)^n$
3	$\sum_{n=0}^m \frac{n^n}{n!} x^n$	9	$\sum_{n=0}^m \frac{x^{n^2}}{n!}$

4	$\sum_{n=0}^m \frac{\sqrt{n^n}}{n!} x^n$	10	$\sum_{n=0}^m \frac{x^{3n}}{(3n)!}$
5	$\sum_{n=1}^m \frac{n!}{(x+n)^2}$	11	$\sum_{n=1}^m \frac{(-1)^{n-1}}{(2n)!} x^{2n-1}$
6	$\prod_{n=1}^m \frac{\sqrt{2n^2 + 1} \cdot x}{n!}$	12	$\prod_{n=1}^m \frac{\operatorname{tg} nx}{n}$

Контрольные вопросы

- 1 Перечислите структурированные типы данных языка *Pascal*.
- 2 Сформулируйте определение массива в языке *Pascal*.
- 3 Какие массивы используются на *Pascal*?
- 4 Как в программном коде осуществляется доступ к отдельному элементу массива.
- 5 Описание одно-, двумерного массивов. Тип элементов массивов.
- 6 Организация ввода и вывода элементов массива.
- 7 Суммирование и вычисление произведений элементов массива.
- 8 Поиск минимального и максимального элементов массива.
- 9 Подсчет количества элементов массива, удовлетворяющих заданным условиям.
- 10 Как поменять местами значения двух элементов массива?
- 11 Понятие процедуры и функции пользователя.
- 12 Как описать процедуру или функцию пользователя?
- 13 Как осуществляется в программном коде вызов подпрограммы?
- 14 Назначение формальных и фактических параметров.

ЛАБОРАТОРНАЯ РАБОТА 3

MATHCAD. БАЗОВЫЕ ОПЕРАЦИИ. РАНЖИРОВАННЫЕ ПЕРЕМЕННЫЕ. ФУНКЦИИ. ГРАФИКИ

Цель работы: получить первичные навыки работы в системе *Mathcad*. Изучить основные понятия и базовые операции, способы задания и использования ранжированной переменной, способы создания и форматирования графиков в системе *Mathcad*.

1 Сведения из теории

1.1 Основы работы в системе *Mathcad*

Запустить систему математических, научных и инженерных расчетов *Mathcad* можно из главного меню: **Start / All Programs / Mathsoft Apps / Mathcad** (Пуск / Все программы / Mathsoft Apps / Mathcad). Интерфейс *Mathcad* внешне напоминает интерфейс большинства приложений *Windows*.

При запуске системы, как правило, автоматически создается новый документ *Mathcad* с именем `Untitled1` (Безымянный), который желательно сразу сохранить, выполнив команду **File / Save as** (*Файл / Сохранить как*). В открывшемся окне необходимо выбрать папку и задать имя файла. По умолчанию файлы *Mathcad* получают расширение `.mcd`.

Для ввода математических символов, греческих букв, шаблонов матриц и графиков служат панели математических операторов и шаблонов. Они собраны на единой панели математических инструментов **Math** (*Математическая*), показанной на рисунке 1. Чтобы отобразить данную панель, необходимо выполнить команду **View / Toolbars / Math** (*Вид / Панели инструментов / Математическая*).

Документ *Mathcad* просматривается, интерпретируется и выполняется **слева направо и сверху вниз** и может включать три вида областей:

- формульные;
- текстовые;
- графические.

Каждая область документа имеет прямоугольную форму. Выделение области осуществляется щелчком мыши на любом ее месте. Чтобы переместить



Рисунок 1 – Панель инструментов Math

область, необходимо навести указатель мыши на ее границу и, когда он примет вид стилизованной кисти руки, нажать на левую клавишу мыши и, не отпуская клавишу, переместить выделенную область на новое место документа.

Для копирования и перемещения областей используются команды меню **Edit** (*Правка*) или стандартные комбинации клавиш операционных систем семейства *Windows*. Кроме того, удалить выделенную область можно, нажав комбинацию клавиш **Ctrl D**.

Выровнять несколько областей по горизонтали можно командой **Format / Align Regions / Across** (*Формат / Выравнивание областей / Горизонтально*). Вариант **Down** (*Вертикально*) предназначен для выравнивания областей по вертикали. Отделение областей друг от друга при их наложении производится командой **Format / Separate Regions** (*Формат / Отделить области*).

1.1.1 Создание и особенности работы в формульной области

Позицию в документе, с которой начинается набор формулы, указывает *визир* в виде *красного крестика*. По умолчанию ввод с клавиатуры любого символа приводит к созданию формульной области. При этом она выделяется тонкой сплошной прямоугольной рамкой. Управляющий *курсор-уголок* синего цвета является основным инструментом при наборе формул.

Чтобы назначить операцию (сложение, умножение, возведение в степень и т. п.) операнду (константе, переменной, выражению), необходимо сначала выделить операнд целиком управляющим курсором-уголком синего цвета, а затем ввести знак операции. Причем знак операции отображается справа или слева от операнда в зависимости от ориентации курсора-уголка.

Выделение (охват) курсором-уголком операнда производится клавишами управления курсором и клавишей **Пробел**, а изменение ориентации курсора-уголка – клавишей **Insert (Ins)**.

Каждое математическое выражение набирается в отдельной формульной области. **Одна формула – одна область!**

По назначению все математические выражения в формульных областях подразделяются на *определяемые* (задаваемые пользователем) и *вычисляемые*, например:

$$a := 5$$

$$f := \frac{a^2 + \sin(a)}{a - 2} \quad \text{– определяемые выражения}$$

$$f = 8.014 \quad 3f + 4 = 28.041 \quad \text{– вычисляемые выражения}$$

Пользователь может задать выражение с использованием оператора *локального присваивания*, который выбирается щелчком мыши по кнопке **:=** (Definition) на палитре **Calculator** (*Калькулятор*) или задается комбинацией клавиш **Shift :** (клавиша **двоеточие** в верхнем регистре клавиатуры).

Для вывода значения переменной или получения результата используется символ **=** (равно), являющийся оператором *численного вычисления* (Evaluate Numerically =).

В *Mathcad* реализованы два режима вычислений: ручной и автоматический. Для простых вычислений целесообразно использовать автоматический режим, установка которого осуществляется командой **Math / Automatic Calculation** (*Математика / Автоматическое вычисление*).

В *ручном* режиме для получения каждого результата необходимо нажать клавишу **F9**.

1.1.2 Создание и особенности работы в текстовой области

Чтобы вставить в документ *Mathcad* текстовую область, выполняют команду **Insert / Text Region** (*Вставка / Область текста*) или нажимают комбинацию клавиш **Shift ʹ**. Можно также создать сначала формульную область, набрав первое слово или символ, а затем нажать клавишу **Пробел**: формульная область преобразуется в текстовую.

Текстовая область имеет рамку с маркерами, позволяющими изменять ее размеры. Курсор в текстовой области имеет вид вертикальной черты красного цвета.

Форматирование данных в текстовой области производится командами, кнопки которых находятся на панели инструментов **Formatting** (*Форматирование*) или посредством диалоговых окон: **Text Format** и **Paragraph Format**. Эти диалоговые окна вызываются командами **Format / Text** (*Формат / Текст*) и **Format / Paragraph** (*Формат / Абзац*) соответственно.

При возникновении проблем, связанных с набором кириллического текста в текстовой области, можно поступить следующим образом:

- перейти на русскую раскладку клавиатуры (**RU**);
- вставить текстовую область (**Insert / Text Region**);
- установить кириллический шрифт, например, Courier, Arial Cyr или Times New Roman Cyr;
- набрать необходимый текст;
- щелкнуть мышью вне текстовой области.

Примечание – После удаления всех символов из текстовой области она автоматически удаляется, и рассмотренный алгоритм необходимо повторять заново.

1.1.3 Переменные. Предопределенные переменные

Переменные могут использоваться в выражениях путем указания их имени. Имя переменной называется *идентификатором*.

Идентификаторы в *Mathcad* начинаются с буквы и могут содержать:

- строчные и прописные буквы латиницы или кириллицы;
- строчные и прописные буквы греческого алфавита;
- арабские цифры от 0 до 9;
- знак подчеркивания `_`.

Учитывая, что в *Mathcad* документ просматривается и выполняется процессором слева направо и сверху вниз, переменная должна быть *локально определена* выше и левее выражения, в которое она входит, т. е. ей необходимо присвоить значение:

`<имя переменной> := <значение>`

Попытка использовать неопределенную ранее переменную влечет появление сообщения об ошибке.

В *Mathcad* предусмотрен также оператор глобального присваивания, действие которого распространяется на весь документ независимо от того, где располагается такой оператор.

Формат оператора глобального присваивания:

`<имя переменной> ≡ <значение>`

где `≡` – знак глобального присваивания, набираемый нажатием комбинации клавиш **Shift** ~ (клавиша ~ расположена слева от клавиши 1).

Однако применение такого оператора не всегда удобно, особенно если переменную требуется переопределять в процессе работы.

Кроме переменных, значения которых задаются пользователем, существуют *встроенные*, иначе говоря, *системные* переменные (**Build-In Variables**), которым изначально присвоены значения, указанные в таблице 1. Значения таких предопределенных переменных могут быть изменены пользователем в процессе работы как в операторе присваивания, так и в диалоговом окне **Math / Options** на вкладке **Build-In Variables** (*Встроенные переменные*).

Таблица 1 – Некоторые системные переменные

Переменная	Значение по умолчанию	Описание
π	3.14159...	Число π , 15 значащих цифр
e	2.71828...	Основание натурального логарифма, 15 значащих цифр

∞	10^{307}	Бесконечность
%	0.01	Процент
ORIGIN	0	Индекс первого элемента массива
TOL	10^{-3}	Допустимая погрешность численных алгоритмов

1.1.4 Форматирование результатов

Под *форматированием результатов* понимают изменение внешнего вида выводимой на экран информации, например, числа знаков после запятой. Различают *локальный* (для активной области или выделенной области) и *глобальный* (для всего документа) форматы результатов.

Для форматирования результатов вычислений используется вызываемое командами **Format / Result** (*Формат / Результат*) диалоговое окно **Result Format**. Оно содержит вкладки: **Number Format** (*Формат числа*), **Display Options** (*Параметры отображения*), **Unit Display** (*Отображение единиц измерения*) и **Tolerance** (*Допуск*).

1.2 Ранжированные (дискретные) переменные

В *Mathcad* существует тип переменных, принимающих не одно, а множество значений. Такие переменные носят название *ранжированных* или *дискретных*. Значением ранжированной переменной является упорядоченная последовательность чисел, каждое из которых отличается от соседнего на постоянную величину, называемую *шагом*.

Ранжированная переменная общего вида определяется выражением

$$\langle \text{имя} \rangle := x_1, x_2 \dots x_n$$

где $\langle \text{имя} \rangle$ – задаваемое пользователем имя ранжированной переменной;

x_1 – первое значение ранжированной переменной;

x_2 – второе значение ранжированной переменной ($x_2 = x_1 + \Delta x$ – первое значение плюс шаг);

x_n – последнее значение ранжированной переменной.

Значения x_1 и x_2 разделяются запятой, последовательность двух точек представляет собой единый знак операции, указывающей, что операнды являются границами диапазона значений. Знак \dots вводится с клавиатуры нажатием клавиши «» или щелчком на кнопке **m..n** (Range Variable), находящейся на панели математических инструментов **Matrix**.

Например, если ранжированная переменная задается последовательностью значений в интервале $x \in [-2; 4]$ с шагом $\Delta x = 0,4$, то определяющий ее оператор имеет вид:

$$x := -2, -1.6 \dots 4$$

Ранжированную переменную можно определить и обобщенно:

$a := -2$ $b := 4$ $\Delta x := 0.4$ $x := a, a + \Delta x.. b$
--

Шаг изменения значений ранжированной переменной в явном виде обычно не задается, он определяется как $x_2 - x_1$. Если $x_1 > x_2$, то шаг – отрицательный. Существует сокращенная форма определения ранжированной переменной:

$$\langle \text{ИМЯ} \rangle := x_1 .. x_n$$


В этом случае шаг изменения ранжированной переменной по умолчанию равен 1 или -1 (если $x_1 > x_2$).

Любое выражение с ранжированной переменной создает после знака равенства = *таблицу вывода*. Числа в таблицах можно форматировать с помощью окна **Result Format** (*Формат Результата*).

Ранжированные переменные используются при выполнении циклических (повторяющихся) вычислений, для вывода численных значений функций в виде таблиц, построения графиков функций, задания массивов.

1.3 Функции

Функции в системе *Mathcad* можно условно разделить на две группы: встроенные и функции пользователя.

Встроенные функции изначально заданы в системе разработчиками. Имя функции вводится с клавиатуры, как правило, в нижнем регистре, т.е. строчными буквами (символами). Полный перечень встроенных функций отображается командой **Insert / Function** (*Вставка / Функция*) или щелчком на кнопке  панели инструментов. При этом появляется диалоговое окно, в котором слева перечислены возможные категории функций, а справа – список функций из выделенной категории.

Наиболее часто применяются функции:

- $\sin(z)$ – синус z (z задается в радианах);
- $\cos(z)$ – косинус z ;
- $\tan(z)$ – тангенс z ;
- $\cot(z)$ – котангенс z ;
- $\exp(z)$ – экспоненциальная функция, значение e (основание натурального логарифма) в степени z ;
- $\ln(z)$ – натуральный логарифм z ;
- $\log(z)$ – десятичный логарифм числа z ;
- $\log(z, b)$ – логарифм числа z по основанию b ;
- $\operatorname{acos}(z)$ – арккосинус z ;

- $\text{acot}(z)$ – арккотангенс z ;
- $\text{asin}(z)$ – арксинус z ;
- $\text{atan}(z)$ – арктангенс z ;
- $\text{round}(z, n)$ – при $n > 0$ возвращает округленное значение z с точностью до n знаков после десятичной точки. При $n < 0$ возвращает значение z , округленное на n разрядов слева от десятичной точки. При $n = 0$ или при использовании функции без второго аргумента возвращается значение z , округленное до ближайшего целого;
- $\text{trunc}(z)$ – целая часть числа z ;
- $\text{rnd}(x)$ – функция генерации случайных чисел, равномерно распределенных на интервале от 0 до x ;
- $\text{mod}(n, k)$ – возвращает остаток от деления n на k , причем аргументы могут быть действительными, а результат имеет тот же знак, что и n ;
- $\text{if}(\text{cond}, x, y)$ – функция условия, возвращает выражение x , если условие cond истинно, и выражение y в остальных случаях.

Функция пользователя сначала должна быть определена, а затем к ней можно обращаться при вычислениях, записи алгебраических выражений, построении графиков и т. п.

Формат определения функция пользователя:

$\langle \text{имя} \rangle (\langle \text{список аргументов} \rangle) := \langle \text{выражение} \rangle$

Сначала задается имя функции, в круглых скобках указывается список аргументов функции (перечень используемых переменных), разделяемых запятыми. Затем вводится оператор присваивания. Справа от него записывается выражение, содержащее операнды, аргументы определяемой функции и функции с аргументами, указанными в списке аргументов.

Обращение к функции осуществляется по ее имени с подстановкой на место аргументов переменных и выражений, значения которых определены до обращения к функции, а также констант.


1.4 Графики функций


Основные *виды графиков* и инструменты для работы с ними представлены кнопками изображенной на рисунке 2 панели математических инструментов **Graph** (*График*) или в подменю **Graph** главного меню **Insert** (*Вставка*).




Рисунок 2 – Панель инструментов для работы с графиками

Построение графика функции одной переменной в декартовой системе координат в *Mathcad* включает:

- 1) определение функции пользователя;
- 2) задание ранжированной переменной в необходимом диапазоне значений аргумента функции;
- 3) вставку в документ *Mathcad* шаблона двумерного графика **X-Y Plot** ;

4) заполнение полей ввода  именем аргумента (ось абсцисс) и именем функции с аргументом в скобках (ось ординат);

5) завершение построения, щелчок левой кнопкой мыши вне области графика.

Примечание – Для построения на одной координатной плоскости графиков двух и более функций их имена или определяющие выражения перечисляются *через запятую* в поле ввода , расположенное слева от оси ординат.


Форматирование графиков осуществляется следующими командами из подменю **Graph** (*График*) меню **Format** (*Формат*):


- **X-Y Plot** – изменение формата двумерного графика в декартовой системе координат;

- **Polar Plot** – изменение формата графика в полярных координатах;

- **3D Plot** – изменение формата трехмерного графика.

Дополнительными инструментами более детального исследования графиков функций являются трассировка и масштабирование:

- **Trace** (пиктограмма ) – отслеживание координат точек графика;

- **Zoom** (пиктограмма ) – изменение размеров отдельных участков плоскости графика.

Их вызов осуществляется командами **Format / Graph / Trace** (*Формат / График / Трассировка*) и **Format / Graph / Zoom** (*Формат / График / Масштаб*) соответственно. Пиктограммы этих инструментов также находятся на панели **Graph** (*График*).

2 Порядок выполнения лабораторной работы

1 Загрузите *Mathcad*: **Start / All Programs / Mathsoft Apps / Mathcad** (*Пуск / Все программы / Mathsoft Apps / Mathcad*).

2 Сохраните в личной папке на диске **z:** новый документ с именем **fi03** – лучше использовать латинские буквы. Производите сохранение регулярно в процессе работы (**Ctrl S**).

3 Вставьте текстовую область **Insert / Text Region** (*Вставка / Область*

текста) и введите в нее текст:

Лабораторная работа 3
Базовые операции системы *Mathcad*.
Ранжированные переменные. Функции. Графики.

4 В новой текстовой области введите фамилию, имя, отчество, учебный шифр и номер варианта.

5 Повторите описанный ниже расчет.

Пример 1. Задать переменные $x = 2,34$; $y = -0,83$; $z = 12$. Вычислить значения выражений


$$\frac{x^2 - \sin 3y}{\sqrt{x}} - 2xz \quad (1)$$


и

$$\ln \sqrt{\frac{1 - \cos^2 x}{1 + \cos^2 z}} + \operatorname{tg}^3 \sqrt[4]{1 - 0,01y^2} \quad (2)$$

при указанных значениях переменных. Установите глобально для всех формульных областей формат вывода результатов 4 знаков после точки и локально для формульной области выражения (1) формат результата 5 знаков после запятой.


Решение.

Отобразите панель математических инструментов командой **View / Toolbars / Math** (*Вид / Панели инструментов / Математика*). Щелчком мыши на кнопке  выведите на экран представленную на рисунке 3 панель **Calculator** (*Калькулятор*).

Укажите посредством визира место расположения создаваемой формульной области и наберите $x :=$, выбрав знак присваивания $:=$ на панели **Calculator** (*Калькулятор*). В появившемся поле ввода  напечатайте число 2,34, используя в качестве разделителя целой и дробной части числа точку:

$x := 2.34$

Аналогично задайте значения переменным y и z , создавая для каждой из них отдельную формульную область.

В четвертой формульной области введите посредством клавиатуры знак деления или щелкните на кнопке  панели **Calculator** (*Калькулятор*). В документе *Mathcad* отобразится шаблон дроби.

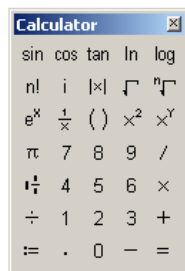
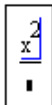


Рисунок 3 – Панель инструментов **Calculator** (*Калькулятор*)

Щелкните мышью на поле ввода \blacksquare в числителе шаблона и введите имя переменной x . Далее щелкните мышью на кнопке \times^2 панели **Calculator** (*Калькулятор*), чтобы возвести в квадрат переменную x . Нажмите клавишу **Пробел**. При этом управляющий курсор-уголок переместится таким образом, что выделенным окажется выражение x^2 :



Введите знак минус, а затем щелкните на кнопке \sin панели **Calculator** (*Калькулятор*). При этом в формульной области отобразится $\sin(\blacksquare)$, т. е. функция с круглыми скобками, между которыми располагается поле ввода \blacksquare , охваченное курсором-уголком и предназначенное для вставки аргумента.

Наберите аргумент $3y$. Обратите внимание, что между 3 и y автоматически ввелся знак умножения.

Щелкните мышью на поле ввода \blacksquare в знаменателе шаблона дроби:

$$\frac{x^2 - \sin(3 \cdot y)}{\blacksquare}$$

Вставьте \sqrt{x} с помощью панели **Calculator** (*Калькулятор*) и нажмите клавишу **Пробел**, чтобы охватить всю дробь целиком управляющим курсором-уголком:

$$\frac{x^2 - \sin(3 \cdot y)}{\sqrt{x}}$$

При наборе выражения $2xz$ не забудьте напечатать знак умножения между переменными x и z , иначе последовательность символов xz будет восприниматься *Mathcad* как имя новой переменной xz , но никак не выражение $x \cdot z$:

$$\frac{x^2 - \sin(3 \cdot y)}{\sqrt{x}} - 2 \cdot x \cdot z$$

Чтобы вычислить и отобразить в документе значение выражения (1), нажмите на клавишу $=$, т. е. введите в формульной области после выражения знак равно $=$.

Окончательно формульная область примет вид

$$\frac{x^2 - 3 \sin(3y)}{\sqrt{x}} - 2x \cdot z = -51.391$$

Для изменения формата результата глобально по всему документу *Mathcad* щелкните мышью в любом месте между областями документа и выполните команду **Format / Result** (*Формат / Результат*). В появившемся окне форматирования на вкладке **Number Format** (*Формат числа*) выберите количество знаков после точки **4** (разделителя целой и дробной части числа).

Для изменения формата результата локально только в формульной области с выражением (1) щелкните мышью на этой области, выполните команду **Format / Result** (*Формат / Результат*) и выберите количество знаков 5, как показано на рисунке 4.

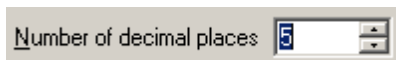
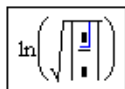


Рисунок 4 – Выбор количества знаков после точки в результате

При вводе выражения (2) в новом формульном блоке используйте последовательно кнопки **ln**, **√**, **|x|** и **/** панели **Calculator** (*Калькулятор*). В документе отобразится формульная область вида



Наберите единицу и знак минус. Щелкните на кнопке **cos** на панели **Calculator** (*Калькулятор*), в скобках введите аргумент функции косинус, нажмите клавишу **Пробел**, чтобы охватить управляющим курсором-уголком косинус вместе с аргументом, только после этого щелкните на кнопке **x²**. Аналогично наберите знаменатель.

Примечание – В математике принято обозначать квадрат функции косинус в виде $\cos^2(x)$. Попытка набрать в документе *Mathcad* один к одному данное выражение приведет к ошибке, так как операция возведения в квадрат не может быть применена только к имени функции. В данном случае необходимо набирать $(\cos(x))^2$ или $\cos(x)^2$.

Выделите введенное ранее выражение управляющим курсором-уголком и добавьте знак **+**. Обратите внимание, что функция тангенс в системе *Mathcad* именуется как **tan**, но не **tg**. Щелкните последовательно на кнопке **tan** и на кнопке **√ⁿ** панели **Calculator** (*Калькулятор*), а затем в верхнее поле ввода вставьте степень корня 4. Далее наберите аргумент функции тангенс

$$1 - 0.01y^2.$$

Для возведения тангенса в куб охватите его вместе с аргументом курсором-уголком, щелкните на кнопке **x^y** панели **Calculator** (*Калькулятор*)

и в верхнем поле ввода наберите цифру 3.

Созданный программный код будет иметь вид:

$$\begin{aligned}
 &x := 2.34 \\
 &y := -0.83 \\
 &z := 12 \\
 &\frac{x^2 - \sin(3y)}{\sqrt{x}} - 2x \cdot z = -52.18404 \\
 &\ln\left(\sqrt{\frac{1 - \cos(x)^2}{1 + \cos(z)^2}}\right) + \tan\left(\sqrt[4]{1 - 0.01y^2}\right)^3 = 3.1353
 \end{aligned}$$

6 Выполните задание 1.

Задание 1. Задать переменные и вычислить значения выражений 1 и 2, указанные в таблице 2.

Таблица 2 – Варианты задания 1

Вариант	Значения переменных	Выражение 1	Выражение 2
1	$x = 2,35;$ $y = -3,83;$ $z = 13,8$	$\sin 2x + \frac{3y + z}{4 - x^2}$	$\frac{\pi}{(1 + x^2)\sqrt{1 + y^2}} + \ln^2 \frac{\sqrt{e^y + 1}}{\sqrt{e^z + 1}}$
2	$a = -0,48;$ $b = 3,98;$ $c = 3,08$	$\frac{a - 2b^2}{\cos 4c} - \sqrt{2c - 1}$	$\ln^2(1 + \sin a) - \sqrt[3]{\frac{b^2 + c^3}{a}}$
3	$d = -6,5;$ $f = 0,59;$ $g = 8$	$\operatorname{tg} 2d + \frac{5\sqrt{f} + 3g}{4d^2}$	$\sin^2\left(\frac{d^2 + d\sqrt{2} + 1}{f^2 - f\sqrt{2} + 1}\right) + \frac{g!}{2\sqrt[3]{\sqrt{2}}}$
4	$k = 6,12;$ $n = -0,54;$ $m = 10,5$	$\ln \frac{3k + n}{\sqrt{m}} + n - 3m^2 $	$\frac{2}{3}\sqrt[5]{k} + \frac{1}{3}\left \operatorname{tg}^2 \frac{m}{1 - n^2}\right $
5	$p = 1,95;$ $q = 0,81;$ $r = -1,1$	$\frac{\sin 7p + 3}{ r } + 2q^2$	$\sqrt{p - \sqrt[3]{\sqrt{q}}} + \frac{e^p}{p} + \cos^2\left(\frac{r^3}{1 + q}\right)$
6	$s = 34,1;$ $t = -1,02;$ $x = 1,8$	$\frac{s^2 + 3t}{\ln x + 2} - \cos 2x$	$\operatorname{tg}^2(s^3 + t^2) + \sqrt[3]{\frac{\pi}{t}} + \frac{ t - 1 + 1}{x}$
7	$u = 6,04;$ $v = 8,1;$ $w = 0,61$	$\sqrt{\frac{2u + w}{v^3}} - \ln 3v$	$\sqrt[3]{(2u + 1)^2} + \operatorname{tg}^2 v + \frac{ w + w^{-3} }{u}$

8	$x = 2,87;$ $y = -3,3;$ $z = 5,25$	$\operatorname{tg} 3x^2 + \frac{\sqrt[3]{y+4z}}{4z^2-1}$	$\ln^2\left(e^x + \sqrt{x^2 + y^2}\right) + \frac{ x^2 - \sqrt{2}y - z }{2z}$
9	$a = 1,4;$ $b = -0,3;$ $c = 3,09$	$\frac{ a-4 + \sin b}{\sqrt[3]{c-c^2}}$	$\pi \cos^2 \frac{2-a}{ b \sqrt{2}} + \frac{3a+2}{4b^2} \sqrt{c-1}$
10	$d = -3,84;$ $f = 1,654;$ $g = 15$	$\frac{d^2 - 3f}{\ln g} + \sqrt{f+3}$	$\ln^3 \frac{\sqrt{1-d}}{\sqrt{1+f}} + \left \frac{g^2 - \sqrt[3]{g}}{d} \right $
11	$u = 1,349;$ $v = -3,93;$ $w = 12,5$	$3\pi + \frac{e^u \sqrt{-v}}{\cos 3w}$	$\operatorname{tg}^2\left(\frac{u}{2} + \frac{\pi}{4}\right) + \sin \left \frac{1}{v} \right + \frac{2w-1}{\sqrt[3]{\sqrt{3}}}$
12	$x = -0,345;$ $p = 3,883;$ $q = 0,8$	$\frac{5x^2 \sqrt{p}}{\sin q} + 2-3x $	$\sqrt{\frac{1 - \sin^2 p}{1 + \sin^2 q}} + \cos \sqrt[3]{1-x^2}$

7 Повторите описанный ниже расчет и построение графика.

Пример 2. Вычислить значения функции

$$\operatorname{fun1} = \frac{\sqrt{m}}{\operatorname{ctg}^2 m + b1} \quad (3)$$

при $b1 = 1,3; -2 \leq m \leq 5; \Delta m = 0,6$.

Решение.

Присвойте переменной $b1$ значение 1.3.


Наберите имя функции $\operatorname{fun1}$, открывающую скобку, имя переменной x , закрывающую скобку и знак присваивания $:=$. В документе отобразится формульная область

$$\operatorname{fun1}(x) := \blacksquare$$

Далее нажмите клавишу $\frac{1}{x}$ на клавиатуре: появится шаблон дроби. Убедитесь, что курсор установлен в числителе дроби, щелкните на кнопке $\frac{\square}{\square}$ панели **Calculator** (*Калькулятор*) и наберите x . Переместите курсор-уголок в знаменатель дроби и напечатайте на клавиатуре имя функции котангенс \cot , открывающую скобку, имя переменной x и закрывающую скобку:

$$\operatorname{fun1}(x) := \frac{\sqrt{x}}{\cot(x)} \quad (4)$$

Функцию котангенс можно вызвать щелчком мыши на ее имени в списке окна **Insert Function** (*Вставка функции*), отображаемого нажатием комбинации клавиш **Ctrl E**.

Охватите управляющим курсором-уголком имя функции $\cot(x)$ вместе с аргументом, используя, например, нажатия клавиши **Пробел**, и щелкните мышью на кнопке  панели **Calculator** (*Калькулятор*):


$$\text{fun1}(x) := \frac{\sqrt{x}}{\cot(x)^2}$$

Нажмите клавишу **Пробел** и напечатайте $+b1$ на клавиатуре. Окончательно формульная область примет вид


$$\text{fun1}(x) := \frac{\sqrt{x}}{\cot(x)^2 + b1} \quad (5)$$

Создайте ранжированную переменную m , набрав последовательно букву m , двоеточие, минус 2, минус 1.4, точку с запятой, цифру 5. Щелкните мышью за пределами формульной области. Получим

$$m := -2, -1.4..5$$

Примечание – Отобразившиеся после цифры 1.4 две точки $..$ являются единым математическим знаком, обозначающим диапазон значений. Этот знак *нельзя* набрать двукратным нажатием на клавишу клавиатуры **Точка**. Он вводится с использованием клавиши **Точка с запятой** или щелчком мыши на кнопке  панели **Matrix** (*Матрицы*).

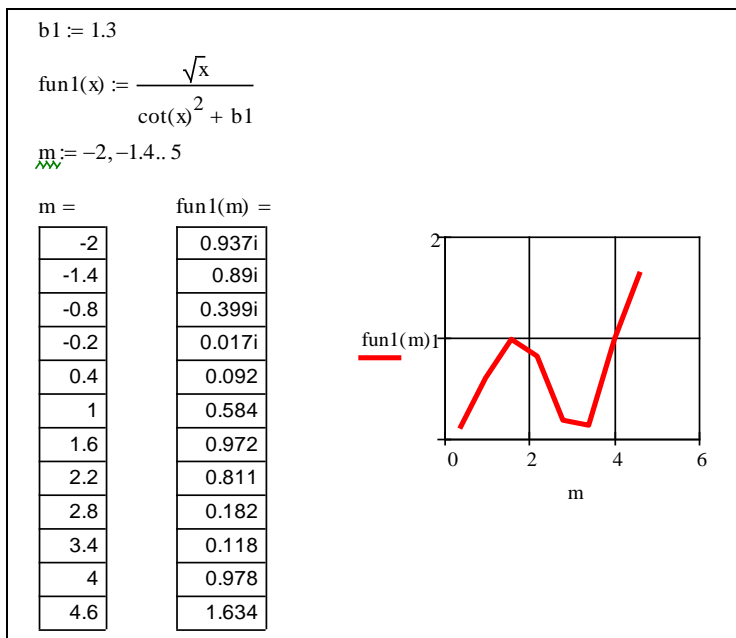
Наберите на клавиатуре $m =$, чтобы в документе *Mathcad* вывести таблицу значений переменной m . Введите с клавиатуры $\text{fun1}(m) =$. В документе *Mathcad* отобразится еще одна таблица – таблица значений функции fun1 на заданном промежутке ее аргумента.

Щелкните мышью на пиктограмме  панели **Graph** (*График*), чтобы вставить в документ *Mathcad* шаблон двумерного декартова графика **X-Y Plot**. В нижнем поле ввода наберите имя переменной m , а в поле слева от оси ординат – имя функции с аргументом в скобках $\text{fun1}(m)$. Щелкните мышью вне графической области. Отобразится график заданной функции.

Форматирование графика осуществляется в окне, вызываемом щелчком правой кнопкой мыши на графической области и выбором из контекстного меню команды **Format**.

Установите пересечение осей выбором переключателя **Crossed** (*Только оси*). Задайте отображение вспомогательных линий установкой флажков по осям X и Y в строке **Grid Lines** (*Вспомогательные линии*).

Фрагмент документа *Mathcad*, реализующий задание, имеет вид:



Примечание – При выполнении задания использовалась функция *fun1* с формальным параметром *x*. При обращении к этой функции формальный параметр заменяется фактическим параметром *m*. Следует помнить, что документ *Mathcad* выполняется сверху вниз. Поэтому сначала следует задать переменную *m*, а затем ниже использовать выражение для искомой функции с этой переменной.

8 Выполните задание 2.

Задание 2. Задать ранжированную переменную и вычислить значения функции при изменении аргумента с постоянным шагом (таблица 3). Построить и отформатировать график функции.

Таблица 3 – Варианты задания 2

Вариант	Переменная	Функция	Вариант	Переменная	Функция
1	$x \in [1; 7],$ $\Delta x = 0,4$	$f(x) = \frac{\sqrt[3]{\cos^2 3x}}{3x \operatorname{ctg} x }$	7	$z \in [0; 2],$ $\Delta z = 0,25$	$y(z) = \ln \frac{\sin^3 2z}{\sqrt{1/3 + e^z}}$
2	$y \in [1; 7],$ $\Delta y = 0,5$	$g(y) = 5y^3 - y^2 \operatorname{arctg} \frac{y}{2}$	8	$t \in [2; 6],$ $\Delta t = 0,5$	$f(t) = \frac{\sqrt[5]{3t - e^t}}{2 \operatorname{ctg}^3(1+t)}$

3	$z \in [-1; 6],$ $\Delta z = 0,6$	$y(z) = \frac{\sqrt{ \ln z }}{z \arccos 0.1z}$	9	$y \in [-5; 5],$ $\Delta y = 1,25$	$g(y) = \arccos^3 \frac{y}{25}$
4	$t \in [3; 8],$ $\Delta t = 0,25$	$g(t) = 2t^3 - \arcsin \frac{\sqrt{t}}{30}$	10	$a \in [-1; 6],$ $\Delta a = 0,8$	$z(a) = \ln \left \operatorname{ctg} a^2 - 1 \right $
5	$x \in [1; 6],$ $\Delta x = 0,3$	$z(x) = \frac{1}{e^x} - \log_{\frac{3}{2}} x - 1$	11	$b \in [0; 6],$ $\Delta b = 0,5$	$y(b) = \lg \left \cos b^3 - 1 \right $
6	$x \in [-1; 7],$ $\Delta x = 0,4$	$f(x) = e^{2 x } + \arcsin \frac{x}{15}$	12	$c \in [-3; 5],$ $\Delta c = 0,2$	$f(c) = \frac{\sqrt[5]{3c - c}}{2 \operatorname{tg}^3(c + 2)}$

9 Подготовьте документ для печати: разделите и выровняйте формульные, графические и текстовые области.

10 Просмотрите документ перед печатью. Распечатайте документ.

11 Завершите работу с *Mathcad*.

Контрольные вопросы

- 1 Назначение системы *Mathcad*.
- 2 Структура окна приложения *Mathcad* и его элементы.
- 3 Отображение и состав панели математических инструментов.
- 4 Документ *Mathcad*. Виды областей документа и их назначение.
- 5 Создание и особенности работы в формульной области.
- 6 Вставка, редактирование и форматирование текстовой области.
- 7 Переменная. Идентификатор. Правила записи идентификаторов.
- 8 Как присвоить значение переменной? Локальное и глобальное определения переменной.
- 9 Системные (предопределенные) переменные. Как переопределить значение системной переменной?
- 10 Определяемые и вычислительные формульные области *Mathcad*. Назначение и отличие.
- 11 Режимы вычислений в *Mathcad*.
- 12 Форматирование результатов.
- 13 Что такое ранжированная переменная и как она задается? Правила записи ранжированной переменной.
- 14 Что такое шаг изменения ранжированной переменной? С каким шагом изменяется ранжированная переменная, если задать только начальное и конечное ее значения?
- 15 С каким шагом изменяется ранжированная переменная, если начальное значение больше конечного?
- 16 Что такое таблица вывода? Перечислить свойства таблиц вывода.
- 17 Что понимают под встроенной функцией? Как вставить в формульную область встроенную функцию? Перечислить основные категории встроенных функций.

18 Что такое функция пользователя и как она задается? Как обратиться к функции пользователя?

19 Типы графиков в *Mathcad*. Способы создания графиков в декартовой системе координат.

20 Форматирование графиков. Содержание вкладок окна форматирования графика в декартовой системе координат.

ЛАБОРАТОРНАЯ РАБОТА 4

МАССИВЫ В *MATHCAD*. РЕШЕНИЕ УРАВНЕНИЙ И СИСТЕМ УРАВНЕНИЙ

Цель работы: изучить способы создания и использования массивов, основные векторные и матричные операции и функции. Получить навыки решения алгебраических уравнений, систем линейных и нелинейных уравнений средствами системы *Mathcad*.

1 Сведения из теории

1.1 Массивы

1.1.1 Основные определения

Массив в пакете *Mathcad* – совокупность конечного числа упорядоченных пронумерованных элементов, которая имеет уникальное имя.

Обычно используют одномерные (*векторы*) и двумерные (*матрицы*) массивы, содержащие числовые, символьные или строковые данные.

Различают векторы-строки (матрицы $1 \times n$) и векторы-столбцы (матрицы $m \times 1$), например:

$$\begin{pmatrix} 3 \\ -5 \\ 1 \end{pmatrix} \text{ – вектор-столбец;}$$

$$(3 \quad -5 \quad 1) \text{ – вектор-строка.}$$

Примечание – Вектор-строка и вектор-столбец при векторных и матричных операциях даже при одинаковом наборе элементов ведут себя по-разному.

Порядковый номер элемента массива называется *индексом*. Другими словами, местоположение элемента в массиве задается индексом. Нижняя граница индексации, т. е. номер первого элемента массива, определяется значением системной переменной *ORIGIN*. По умолчанию

$$\boxed{\text{ORIGIN} = 0}$$

и может принимать только целые значения, положительные и отрицательные.

Чтобы переопределить значение системной переменной *ORIGIN*,

необходимо командой **Math / Options** открыть диалоговое окно математических свойств документа **Options**, перейти на вкладку **Build-In Variables** (*Встроенные переменные*) и в поле-счетчике **Array Origin (ORIGIN)** установить новое значение **ORIGIN**, например, 1. Альтернативным способом переопределения системной переменной **ORIGIN** является присваивание ей нового значения непосредственно в документе *Mathcad*, например, посредством создания формульной области

$$\text{ORIGIN} := 1$$

Влияние значения системной переменной **ORIGIN** на индексацию элементов массива демонстрирует следующий код:

$$\begin{array}{cccc} \text{ORIGIN} = 0 & U := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} & U_1 = 2 & U_0 = 1 \\ \text{ORIGIN} := 1 & U_1 = 1 & \text{ORIGIN} := 0 & U_1 = 2 \end{array}$$

Матрицей называется математический объект, элементы которого упорядочены в виде прямоугольной таблицы, состоящей из строк и столбцов. Местоположение элемента матрицы задается двумя индексами, первый из которых определяет номер строки, а второй – номер столбца. По умолчанию верхняя строка и левый столбец имеют номера, равные 0. Чтобы в *Mathcad* перейти к привычной, начинающейся с единицы, нумерации строк и столбцов матрицы, достаточно переопределить системную переменную **ORIGIN**, задав ее, как и в случае векторов, равной 1.

1.1.2 Создание массивов

Существует несколько способов создания массивов.

1 Использование диалогового окна **Insert Matrix** (*Вставка матрицы*).

Наберите имя массива и оператор присваивания. Например,

$$A := \begin{bmatrix} \\ \\ \end{bmatrix}$$

Выполните одно из перечисленных ниже действий:

- откройте меню **Insert** (*Вставка*) и щелкните на его пункте **Matrix** (*Матрица*);
- нажмите комбинацию клавиш **Ctrl M**;
- щелкните на пиктограмме с изображением шаблона матрицы на панели **Math**, показанной на рисунке 1, а затем на пиктограмме

отобразившейся панели **Matrix** (*Матрица*), приведенной на рисунке 2.



Рисунок 1 – Панель математических инструментов **Math**



Рисунок 2 – Панель для работы с векторами и матрицами

Любое из этих действий вызывает появление диалогового окна (рисунок 3), в котором необходимо указать количество строк (**Rows**) и столбцов (**Columns**) матрицы.

Для векторов один из этих параметров должен быть равен 1. Ввод единицы в поле **Rows** приводит к созданию шаблона вектора-строки, ввод единицы в поле **Columns** – шаблона вектора-столбца.

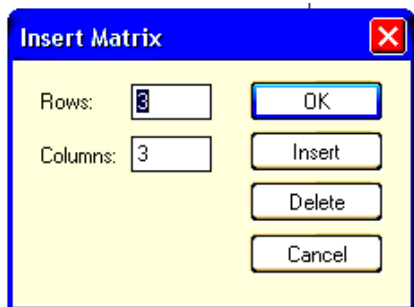
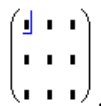


Рисунок 3 – Диалоговое окно вставки шаблона матрицы

После щелчка на кнопке **Insert** в месте расположения курсора появится шаблон



в поля которого необходимо ввести значения элементов массива.

Переход к следующему полю производится нажатием клавиши **Tab**, с помощью клавиш управления курсором или посредством щелчка мышью.

2 Использование ранжированной переменной.

Целочисленные ранжированные переменные удобно использовать для задания индексов массива. Для нумерации элементов вектора формируется ранжированная переменная, например, вида

$$\text{ORIGIN} := 1 \quad i := 1..4$$

Далее следует набрать имя вектора, квадратную скобку [, имя ранжированной переменной и знак оператора присваивания. Например,

$$V_i := \blacksquare$$

Затем в поле ввода необходимо последовательно печатать значения элементов вектора, отделяя их друг от друга запятой. После набора первого значения 3 формульная область будет иметь, например, вид

$$V_i := 3$$

Ввод запятой добавит в формульную область столбец из двух ячеек, а набор каждого нового значения с запятой продляет таблицу в виде столбца на одну ячейку (строку). Результаты этих последовательных действий показаны ниже:

$V_i :=$	$V_i :=$	$v = \begin{pmatrix} 3 \\ 7 \\ 0 \\ 0 \end{pmatrix}$	$V_i :=$	$V_i :=$	$v = \begin{pmatrix} 3 \\ 7 \\ 4 \\ 9 \end{pmatrix}$											
<table border="1" style="margin: auto;"><tr><td style="text-align: center;">3</td></tr><tr><td style="text-align: center;">■</td></tr></table>	3	■	<table border="1" style="margin: auto;"><tr><td style="text-align: center;">3</td></tr><tr><td style="text-align: center;">7</td></tr></table>	3	7		<table border="1" style="margin: auto;"><tr><td style="text-align: center;">3</td></tr><tr><td style="text-align: center;">7</td></tr><tr><td style="text-align: center;">■</td></tr></table>	3	7	■	<table border="1" style="margin: auto;"><tr><td style="text-align: center;">3</td></tr><tr><td style="text-align: center;">7</td></tr><tr><td style="text-align: center;">4</td></tr><tr><td style="text-align: center;">9</td></tr></table>	3	7	4	9	
3																
■																
3																
7																
3																
7																
■																
3																
7																
4																
9																

Для нумерации значений индексов матрицы необходимо задать две ранжированные переменные, например, посредством двух формульных областей, например,

$$\text{ORIGIN} := 1 \quad i := 1..4 \quad j := 1..4$$

Последовательность ввода элементов матрицы и промежуточные результаты демонстрирует следующий фрагмент кода *Mathcad*.

$W_{i,j} := \blacksquare$	$W_{i,j} := 3.$	$W_{i,j} :=$	$W_{i,j} :=$	$W = \begin{pmatrix} 3 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$W_{i,j} :=$	$W = \begin{pmatrix} 3 & 7 & 4 & 8 \\ 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$									
		<table border="1" style="margin: auto;"><tr><td style="text-align: center;">3</td></tr><tr><td style="text-align: center;">■</td></tr></table>	3	■	<table border="1" style="margin: auto;"><tr><td style="text-align: center;">3</td></tr><tr><td style="text-align: center;">7</td></tr></table>	3	7		<table border="1" style="margin: auto;"><tr><td style="text-align: center;">3</td></tr><tr><td style="text-align: center;">7</td></tr><tr><td style="text-align: center;">4</td></tr><tr><td style="text-align: center;">8</td></tr><tr><td style="text-align: center;">5</td></tr></table>	3	7	4	8	5	
3															
■															
3															
7															
3															
7															
4															
8															
5															

Mathcad представляет возможность набрать сразу весь список значений в отдельной формульной области вида

3, 7, 4, 8, 5, 1, 7, 9, 2, 5, 7, 2, 1, 9, 6, 4.

и, скопировав ее содержимое, вставить в поле ввода в новой формульной области

$W_{i,j} :=$ ■

В документе *Mathcad* отобразится таблица из одного столбца, заполненного шестнадцатью числами из списка.

Вывод созданной матрицы

$$W = \begin{pmatrix} 3 & 7 & 4 & 8 \\ 5 & 1 & 7 & 9 \\ 2 & 5 & 7 & 2 \\ 1 & 9 & 6 & 4 \end{pmatrix}$$

показывает, что первые 4 числа формируют первую строку, следующие 4 числа – вторую строку и т. д.

Если значения вектора или матрицы задаются *явной формулой*, то их создание упрощается – достаточно в поле ввода новой формульной области

$W_{i,j} :=$ ■

набрать выражение из правой части формулы.

Обращение к отдельным элементам вектора или матрицы осуществляется по индексу, который оформляется как нижний индекс после имени массива. При обращении к элементу матрицы после ее имени вводятся через запятую два нижних индекса, обозначающих номер строки и номер столбца соответственно:

$$W_{3,2} = 5$$

Примечание – Наряду с индексированными переменными вида x_i , являющимися элементами вектора, в *Mathcad* применяются скалярные переменные, часть имени которых оформляется в виде подстрочного символа или строки символов. Такие переменные часто используются для обозначения скалярных физических величин. Например, масса одного тела m_1 , масса второго тела m_2 , максимальное давление p_{max} и т. д. В отличие от нижнего индекса элемента вектора, вставка подстрочного символа или строки символов должна начинаться с точки. В *Mathcad 14* эта точка отображается при выделении формульной области. К сожалению, в старых версиях *Mathcad* эта точка скрыта (невидима), что затрудняет корректную идентификацию переменных, так как итоговое изображение, например, элемента вектора x_1 внешне похоже на изображение скалярной величины x_1 .

Пример 1. Задать вектор-столбец v , состоящий из 5 элементов путем создания соответствующего шаблона. Сформировать вектор w из пяти элементов с использованием ранжированной переменной. Построить вектор, каждый элемент которого вычисляется по формуле $x_i = \sin^3 i$, $i = 1..5$.

Решение.

Задайте значение системной переменной **ORIGIN** равным единице:

ORIGIN := 1

Наберите имя вектора и знак присваивания $v :=$. Не перемещая курсор-уголок, вызовите диалоговое окно **Insert Matrix** (*Вставка матрицы*). Задайте число строк (**Rows**) равным 5, а число столбцов (**Columns**) – 1. Щелкните на кнопке **OK**. Введите в отобразившийся шаблон вектора-столбца произвольные числа, перемещаясь между полями ■ в шаблоне вектора с помощью клавиши **Tab**. Например,

$$v := \begin{pmatrix} 4 \\ -8 \\ 7 \\ 5 \\ 1 \end{pmatrix}$$

Убедитесь, что первый элемент вектора v равен 4. Для этого наберите $v[1]=$. В документе *Mathcad* должна отобразиться формульная область

$v_1 = 4$

Задайте ранжированную переменную для нумерации элементов создаваемых векторов:

$i := 1..5$

Наберите имя вектора w , нажмите клавишу $\boxed{[}$ или щелкните на кнопке \times_n панели инструментов **Matrix** (*Матрица*), введите нижний индекс i , знак присваивания $:=$ и значения элементов вектора w через запятую.

Чтобы создать вектор, элементы которого определяются заданной формулой, наберите имя вектора x , квадратную скобку $[$, нижний индекс i , знак присваивания и арифметическое выражение. Например,

$x_i := \sin(i)^3$

Отобразите значения созданного вектора w , набрав его имя, знак $=$ и нажав клавишу **Enter**. Аналогично отображаются значения вектора x :

$$w = \begin{pmatrix} -3 \\ 2 \\ 6 \\ 12 \\ 4 \end{pmatrix} \quad x = \begin{pmatrix} 0.596 \\ 0.752 \\ 0.003 \\ -0.433 \\ -0.882 \end{pmatrix}$$

Пример 2. Создать матрицу A размером 3×2 , используя соответствующий шаблон, и матрицу B посредством ввода элементов списком. Сформировать матрицу C размером 3×2 , каждый элемент которой вычисляется по формуле $C_{i,j} = \operatorname{tg} i + \sqrt{j}$, $i = 1..3$, $j = 1..2$.

Решение.

Введите имя матрицы и знак присваивания $A :=$. Не перемещая курсор, вызовите диалоговое окно **Insert Matrix** (*Вставка матрицы*). Задайте число строк (**Rows**) равным 3, а число столбцов (**Columns**) равным 2. Щелкните на кнопке **OK**. Введите произвольные значения элементов матрицы, перемещаясь между полями в шаблоне вектора с помощью клавиши **Tab**. Например,

$$A := \begin{pmatrix} 7 & -8 \\ 2 & 4 \\ -5 & 0 \end{pmatrix}$$

Задайте значение системной переменной **ORIGIN** равным 1 и две ранжированные переменные для нумерации элементов создаваемых массивов (каждую в отдельной формульной области):

$$\begin{array}{l} \text{ORIGIN} := 1 \\ i := 1..3 \\ j := 1..2 \end{array}$$

Наберите последовательность символов $B[i, j]$: и далее произвольные значения элементов матрицы B через запятую, т. е. списком.

В новой формульной области составьте оператор присваивания на основе формулы

$$C_{i,j} = \operatorname{tg} i + \sqrt{j}, \quad (1)$$

учитывая, что обращение к функции тангенс с аргументом i реализовано в *Mathcad* как $\tan(i)$.

Итоговый фрагмент документа *Mathcad* должен иметь следующий вид:

$$B_{i,j} := \begin{matrix} 5 \\ -1 \\ 4 \end{matrix} \quad C_{i,j} := \tan(i) + \sqrt{j}$$

$$B = \begin{pmatrix} 5 & -1 \\ 4 & 0 \\ 0 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 2.557 & 2.972 \\ -1.185 & -0.771 \\ 0.857 & 1.272 \end{pmatrix}$$

1.1.3 Шаблоны для работы с векторами и матрицами

Использование представленных в таблице 1 шаблонов требует соблюдения правил матричного исчисления.

Таблица 1 – Шаблоны панели инструментов Matrix (*Матрица*)

Кнопка	Назначение	Кнопка	Назначение
	Создание массива		Транспонирование массива
	Нижний индекс		Диапазон дискретной переменной
	Инверсия (обратная матрица)		Скалярное произведение
	Определитель матрицы, модуль вектора		Векторное произведение
	Операция векторизации		Суммирование элементов вектора
	Выделение столбца матрицы		Изображение

Кроме того, к векторам и матрицам одинаковой размерности применимы операции сложения и вычитания. Если число столбцов первой матрицы совпадает с числом строк второй, то по отношению к матрицам разрешена операция умножения. Квадратную матрицу можно возвести в заданную степень.

Поэлементное умножение элементов массивов одинаковой размерности или вычисление значений скалярной функции от каждого элемента матрицы возможно с использованием оператора векторизации (см. таблицу 1).

1.1.4 Функции для работы с векторами и матрицами

Функции, аргументами которых являются векторы, принято называть *векторными функциями*. В их число входят функции:

- $\text{last}(v)$ – возвращает индекс последнего элемента вектора v ;
- $\text{length}(v)$ – возвращает длину вектора v .
- $\text{diag}(v)$ – создает диагональную матрицу, элементы главной диагонали которой формируются из элементов вектора v .

Из уже существующих массивов можно создавать новые, используя функции *формирования массивов*:

- $\text{augment}(A, B, C, \dots)$ – объединяет в один массивы A, B, C и т. д., имеющие одинаковое число строк (слияние идет бок о бок);
- $\text{stack}(A, B, C, \dots)$ – объединяет массивы A, B, C и т. д., имеющие одинаковое число столбцов (слияние массивов идет сверху вниз);
- $\text{submatrix}(A, ir, jr, ic, jc)$ – возвращает часть массива A , состоящую из элементов, содержащихся в строках с ir по jr и в столбцах с ic по jc ;
- $\text{matrix}(m, n, f)$ – позволяет создать матрицу размерности $m \times n$, каждый i -й, j -й элемент которой есть функция двух переменных $f(i, j)$, где $i = 0, 1, \dots, m$ и $j = 0, 1, \dots, n$.

Для создания матриц специального вида предназначены функции:

- $\text{geninv}(A)$ – возвращает левую обратную матрицу для A ;
- $\text{identity}(n)$ – создает единичную квадратную матрицу размером $n \times n$;
- $\text{gref}(A)$ – ступенчатый вид массива A .

Следующие встроены функции возвращают значения элементов и специальные характеристики массивов:

- $\text{max}(A, B, C, \dots)$ – возвращает максимальный по значению элемент;
 - $\text{min}(A, B, C, \dots)$ – возвращает минимальный по значению элемент.
- Аргументами функций min и max могут быть не только массивы, но и переменные, числа и строки;
- $\text{IsArray}(x)$ – возвращает значение 1, если x – матрица или вектор, иначе возвращает 0;
 - $\text{lookup}(z, A, B)$ – функция ищет значение z в массиве A и возвращает элементы массива B , стоящие на тех же местах, что и z в массиве A (A и B имеют одинаковую размерность);
 - $\text{match}(z, A)$ – возвращает индекс (индексы) элемента z в массиве A ;
 - $\text{cols}(A)$ – возвращает число столбцов массива A ;
 - $\text{rows}(A)$ – возвращает число строк массива A ;
 - $\text{rank}(A)$ – возвращает ранг массива A ;
 - $\text{tr}(M)$ – возвращает след (сумму диагональных элементов) квадратной матрицы M ;
 - $\text{mean}(A)$ – среднее арифметическое значение элементов массива A ;

- $\text{gmean}(A)$ – для массива, элементы которого положительны, возвращает среднее геометрическое значение элементов массива A ;
- $\text{median}(A)$ – возвращает медиану элементов массива A ;
- $\text{mode}(A)$ – возвращает наиболее часто встречающееся значение элементов массива A .

Для сортировки элементов массивов предназначены функции:

- $\text{sort}(v)$ – сортировка элементов вектора в порядке возрастания;
- $\text{reverse}(v)$ – перестановка элементов вектора в обратном порядке;
- $\text{csort}(M, n)$ – перестановка строк матрицы M таким образом, чтобы отсортированным оказался n -й столбец;
- $\text{rsort}(M, m)$ – перестановка столбцов матрицы M таким образом, чтобы отсортированной оказалась m -я строка.

1.1.5 Вычисление сумм и произведений элементов массивов

Для вычисления суммы всех элементов вектора-столбца предназначен шаблон оператора суммирования $\sum v$, кнопка которого находится на панели инструментов **Matrix** (*Матрица*). Достаточно задать вектор, отобразить шаблон оператора суммирования в документе *Mathcad* и вписать в поле ввода шаблона имя вектора:

$$v := \begin{pmatrix} 6 \\ -3 \\ 0 \\ 7 \end{pmatrix} \quad \sum v = 10$$

Суммирование и вычисление произведений элементов массива (вектора или матрицы) выполняется также с использованием соответствующих шаблонов показанной на рисунке 4 панели **Calculus** (*Исчисление*).

Например, после вставки в документ шаблона

$$\sum_{i=1}^n v_i$$

необходимо задать четыре величины: переменную i , значениями которой являются номера элементов вектора, константу n , являющуюся номером (индексом) последнего учитываемого при суммировании элемента вектора, константу m , обозначающую номер первого учитываемого элемента вектора. Эти величины заносятся в поля ввода \blacksquare шаблона. Вычисление суммы производится

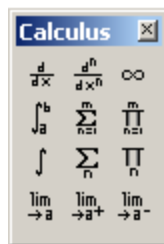


Рисунок 4 – Панель инструментов **Calculus**

нажатием клавиши $\boxed{=}$. Результат представляется формульными областями

$$\boxed{m := 1} \quad \boxed{n := 4} \quad \boxed{\sum_{i=m}^n v_i = 10}$$

или формульной областью

$$\boxed{\sum_{i=1}^4 v_i = 10}$$

Фрагмент документа *Mathcad*, в котором вычисляются сумма элементов второй строки и сумма всех элементов матрицы A имеет вид:

$$A := \begin{pmatrix} 7 & 9 & -7 \\ 0 & 12 & 6 \\ -3 & 5 & 2 \end{pmatrix} \quad \sum_{i=1}^3 A_{2,i} = 18 \quad \sum_{i=1}^3 \sum_{j=1}^3 A_{i,j} = 31$$

Фрагмент документа *Mathcad*, в котором вычисляется произведение элементов третьего столбца и произведение всех элементов матрицы A , включает следующие формульные области:

$$A := \begin{pmatrix} 7 & 9 & -7 \\ 0 & 12 & 6 \\ -3 & 5 & 2 \end{pmatrix} \quad \prod_i A_{i,3} = -84 \quad \prod_{i=1}^3 \prod_{j=1}^3 A_{i,j} = 0$$

1.1.6 Решение задач обработки массивов в *Mathcad*

Пример 3. Ввести массив $MS [1..5]$, состоящий из произвольных чисел, и значение $var = 0,95$. Для каждого элемента массива MS вычислить значение RES по формуле $RES_i = \log_2^3 |var + 2\sqrt[3]{MS_i}|$. Подсчитать сумму элементов массива RES . Найти минимальное значение RES . Отсортировать элементы массива RES по убыванию. Подсчитать произведение элементов массива RES , стоящих на нечетных местах. Найти косинус каждого элемента RES .

Решение.

Задайте порядок нумерации элементов в массивах, начиная с единицы, и ранжированную переменную для нумерации элементов создаваемых векторов MS и RES :

$$\boxed{ORIGIN := 1} \quad \boxed{i := 1..5}$$

Введите имя исходного массива и знак присваивания. Не изменяя положение курсора-уголка, щелкните на кнопке с изображением матрицы дважды и в отобразившемся диалоговом окне укажите размерность вектора-столбца: **Rows** – количество строк 5, **Columns** – количество столбцов 1.

Введите произвольные значения элементов массива MS , используя для перехода от одного поля ввода к другому клавишу **Tab** или клавиши управления курсором:

$$MS := \begin{pmatrix} 9 \\ -3 \\ 5 \\ 1 \\ 12 \end{pmatrix}$$

Присвойте значение переменной var :

$$var := 0.95$$

Наберите формулу для вычисления i -го элемента массива RES :

$$RES_i := \log\left(\left|var + 2 \cdot \sqrt[3]{MS_i}\right|, 2\right)^3$$

Для просмотра результатов в виде таблицы наберите $MS_i =$ и $RES_i =$, а для вывода результатов в виде векторов-столбцов – $MS =$ и $RES =$:

$MS_i =$	$RES_i =$		
9	13.034	$MS =$	$\begin{pmatrix} 9 \\ -3 \\ 5 \\ 1 \\ 12 \end{pmatrix}$
-3	0.863	$RES =$	$\begin{pmatrix} 13.034 \\ 0.863 \\ 9.631 \\ 3.802 \\ 15.014 \end{pmatrix}$
5	9.631		
1	3.802		
12	15.014		

Вычислите сумму элементов массива RES . Для этого щелкните на кнопке Σ панели инструментов **Matrix** (*Матрица*), наберите в поле ввода \blacksquare имя массива RES и нажмите клавишу $=$. Формульная область примет вид:

$$\sum RES = 42.343$$

Для определения минимального элемента массива используйте функцию `min`. Ее применение по отношению к массиву `RES` приведет к результату

$$\min(\text{RES}) = 0.863$$

Выполните сортировку массива `RES` по возрастанию значений элементов, а затем измените порядок расположения элементов в массиве на обратный (от большего по величине значения к меньшему). Это достигается применением кода

$$\text{sort}(\text{RES}) = \begin{pmatrix} 0.863 \\ 3.802 \\ 9.631 \\ 13.034 \\ 15.014 \end{pmatrix} \quad \text{reverse}(\text{sort}(\text{RES})) = \begin{pmatrix} 15.014 \\ 13.034 \\ 9.631 \\ 3.802 \\ 0.863 \end{pmatrix}$$

Вычислите произведение элементов, расположенных на нечетных местах, применив функцию `if`, как показано в формульной области

$$\prod_i \text{if}(\text{mod}(i, 2) = 1, \text{RES}_i, 1) = 1884.73$$

Примечание – Обращение к функции `if` имеет вид

$$\text{if}(\langle \text{условие} \rangle, \langle \text{выражение1} \rangle, \langle \text{выражение2} \rangle)$$

где `<условие>` – логическое выражение, сформированное с использованием логических операций и операций отношения, набираемых посредством элементов панели инструментов **Boolean** (*Логические*);

`<выражение1>` – выражение, которое выполняется, если `<условие>` имеет значение «истина»;

`<выражение2>` – выражение, которое выполняется, если `<условие>` имеет значение «ложь».

В *Mathcad* допускается использование массива в качестве аргумента.

Вычисление, например, значения функции косинус каждого элемента массива `RES` осуществляется следующим образом:

$$\cos(\text{RES}) = \begin{pmatrix} 0.893 \\ 0.65 \\ -0.979 \\ -0.79 \\ -0.769 \end{pmatrix}$$

Пример 4. Задать матрицу M размером 10×10 согласно формуле

$$M_{i,j} = \frac{(10 \sin i)^3}{j+2} \operatorname{tg} j. \quad (2)$$

Создать подматрицу $M1$ размером 5×5 из элементов M , расположенных на пересечении первых пяти строк и пяти последних столбцов.

Найти сумму элементов $M1$.

Подсчитать произведение тех элементов матрицы $M1$, сумма значений индексов которых является четным числом.

Определить минимальный элемент второй строки $M1$.

Создать матрицу $M2$, каждый элемент которой – десятичный логарифм соответствующего элемента матрицы $M1$.

Комплексные элементы матрицы $M2$ заменить их действительной частью.

Найти максимальные элементы в нечетных столбцах $M2$.

Определить количество отрицательных элементов массива $M2$.

Сформировать вектор z из диагональных элементов матрицы $M2$ и найти его модуль.

Сформировать вектор w из элементов третьего столбца $M2$.

Создать матрицу $M3$ из четных строк матрицы $M2$ и векторов z и w .

Решение.

Определите переменную **ORIGIN** и ранжированные переменные для нумерации элементов массивов (каждая формула в отдельном блоке):

$$\text{ORIGIN} := 1 \quad i := 1..10 \quad j := 1..10$$

Создайте матрицу M и ее подматрицу $M1$, используя встроенные функции:

$$M_{i,j} := \frac{(10 \sin(i))^3}{j+2} \cdot \tan(j)$$

$$M1 := \text{submatrix}(M, 1, 5, 6, 10) \quad M1 = \begin{pmatrix} -21.674 & 57.692 & -405.143 & -24.5 & 32.192 \\ -27.348 & 72.798 & -511.221 & -30.915 & 40.621 \\ -0.102 & 0.272 & -1.911 & -0.116 & 0.152 \\ 15.767 & -41.971 & 294.739 & 17.824 & -23.42 \\ 32.075 & -85.379 & 599.575 & 36.258 & -47.642 \end{pmatrix}$$

Вычислите сумму элементов $M1$, вызывая дважды шаблон суммирования панели **Calculus** (*Исчисление*):


$$\sum_{i=1}^5 \sum_{j=1}^5 M1_{i,j} = -21.375$$

Вычислите произведение тех элементов матрицы $M1$, у которых сумма значений индекса i и индекса j является четным числом. Для этого с помощью функции **if** при нахождении произведения замените элементы, не удовлетворяющие данному условию, на единицу. Соответствующая формульная область имеет вид:

$$\prod_{i=1}^5 \prod_{j=1}^5 \text{if}(\text{mod}(i+j, 2) = 0, M1_{i,j}, 1) = -1.293 \times 10^{16}$$

Найдите минимальный элемент второй строки $M1$, выполнив транспонирование матрицы, а затем применив функцию **min** по отношению ко второму столбцу транспонированной матрицы:

$$\min[(M1^T)^{(2)}] = -511.221$$

Создайте матрицу $M2$ посредством операции векторизация, набрав выражение $\log(M1)$ и щелкнув мышью на кнопке  панели инструментов **Matrix** (*Матрица*):

$$M2 := \overrightarrow{\log(M1)}$$

$$M2 = \begin{pmatrix} 1.336 + 1.364i & 1.761 & 2.608 + 1.364i & 1.389 + 1.364i & 1.508 \\ 1.437 + 1.364i & 1.862 & 2.709 + 1.364i & 1.49 + 1.364i & 1.609 \\ -0.99 + 1.364i & -0.565 & 0.281 + 1.364i & -0.937 + 1.364i & -0.819 \\ 1.198 & 1.623 + 1.364i & 2.469 & 1.251 & 1.37 + 1.364i \\ 1.506 & 1.931 + 1.364i & 2.778 & 1.559 & 1.678 + 1.364i \end{pmatrix}$$

Выделите действительные части элементов комплексной матрицы:

$$M2 := \overrightarrow{\text{Re}(M2)} \quad M2 = \begin{pmatrix} 1.336 & 1.761 & 2.608 & 1.389 & 1.508 \\ 1.437 & 1.862 & 2.709 & 1.49 & 1.609 \\ -0.99 & -0.565 & 0.281 & -0.937 & -0.819 \\ 1.198 & 1.623 & 2.469 & 1.251 & 1.37 \\ 1.506 & 1.931 & 2.778 & 1.559 & 1.678 \end{pmatrix}$$

Найдите максимальные элементы в каждом нечетном столбце матрицы $M2$, задав последовательность из нечетных индексов матрицы посредством ранжированной переменной k , выделив (сформировав) необходимые векторы-столбцы $M2^{<k>}$ и применив функцию \max , возвращающую максимальный элемент каждого нечетного столбца матрицы. Фрагмент документа *Mathcad*, выполняющий эти действия, имеет вид:

$$k := 1, 3..5 \quad \max(M2^{<k>}) = \begin{matrix} 1.506 \\ 2.778 \\ 1.678 \end{matrix}$$

Определите количество отрицательных элементов матрицы $M2$, воспользовавшись функцией

$$\text{if}(M2_{i,j} < 0, 1, 0),$$

которая возвращает значение 1 для отрицательного элемента матрицы и 0 для положительного элемента матрицы. Сумма всех единиц в этом случае совпадает с количеством отрицательных элементов матрицы $M2$:

$$\sum_{i=1}^5 \sum_{j=1}^5 \text{if}(M2_{i,j} < 0, 1, 0) = 4$$

Создайте вектор z из элементов матрицы $M2$ и вычислите его модуль. Сформируйте вектор на основе третьего столбца матрицы $M2$. Данные действия реализует следующий фрагмент документа *Mathcad*:

$$z_i := M2_{i,3} \quad z = \begin{pmatrix} 1.336 \\ 1.862 \\ 0.281 \\ 1.251 \\ 1.678 \end{pmatrix} \quad |z| = 3.116 \quad w := M2^{<3>} \quad w = \begin{pmatrix} 2.608 \\ 2.709 \\ 0.281 \\ 2.469 \\ 2.778 \end{pmatrix}$$

Создайте матрицу $M3$ из четных строк матрицы $M2$ и векторов z и w . Для этого транспонируйте матрицу $M2$, сформируйте из нее необходимые вектор-столбцы и используйте функцию augment , чтобы объединить массивы с одинаковым количеством строк. Соответствующий фрагмент документа *Mathcad* имеет вид:

$$M3 := \text{augment}\left[\left(M2^T\right)^{\langle 2 \rangle}, \left(M2^T\right)^{\langle 4 \rangle}, z, w\right] \quad M3 = \begin{pmatrix} 1.437 & 1.198 & 1.336 & 2.608 \\ 1.862 & 1.623 & 1.862 & 2.709 \\ 2.709 & 2.469 & 0.281 & 0.281 \\ 1.49 & 1.251 & 1.251 & 2.469 \\ 1.609 & 1.37 & 1.678 & 2.778 \end{pmatrix}$$

1.2 Решение уравнений и систем уравнений средствами *Mathcad*

Пакет *Mathcad* обладает широкими возможностями численного решения уравнений и систем уравнений.

1.2.1 Функция *root*, блоки *Given...Find* и *Given...Minerrg*

В ходе нахождения корней уравнения обычно выделяют два этапа:

1) *отделение* корней – определение интервала нахождения каждого корня или определение приблизительного значения корня. В *Mathcad* наиболее наглядным является отделение корней уравнения *графическим способом*;

2) *уточнение* корней – нахождение численного значения корня с заданной точностью.

Точность нахождения корня устанавливается с помощью системной переменной *TOL* (Convergence Tolerance – Допустимое отклонение), значение которой по умолчанию равно 10^{-3} . Чем меньше значение *TOL*, тем с большей точностью, вообще говоря, находится корень уравнения.

Чтобы переопределить значение системной переменной *TOL*, необходимо посредством команды **Math / Options** открыть диалоговое окно математических свойств документа **Options**, перейти на вкладку **Build-In Variables** (*Встроенные переменные*) и в поле *TOL* (Convergence Tolerance) ввести новое значение, например, 0.0001. Это значение распространяется на весь документ *Mathcad*. Присваивание системной переменной *TOL* непосредственно в документе *Mathcad* нового значения, например,

$$\boxed{TOL := 0.0001}, \quad (3)$$

изменяет *TOL* для всех формульных и графических областей, расположенных правее и ниже оператора присваивания (3).

Для решения *одного уравнения с одной неизвестной* предназначена встроенная функция *root*, формат обращения к которой имеет вид:

$$\text{root}(f(x), x, [a, b]).$$

Данная функция возвращает значение переменной *x*, при котором функция *f(x)* обращается в ноль. Аргументы функции *root*:

f(x) – функция в левой части уравнения $f(x) = 0$;

x – переменная, относительно которой требуется решить уравнение;

a, b – необязательные действительные числа, такие что $a < b$, причем на

интервале $[a, b]$ находится только один корень.

Если функция *root* не может найти корни уравнения, то рекомендуется уточнить начальное приближение, изменить границы интервала $[a, b]$ нахождения корня или увеличить значение системной переменной *TOL*.

Для решения уравнений или систем нескольких уравнений и неравенств используются функции *Find* или *Minerr*, каждая из которых завершает формирование в документе *Mathcad* *вычислительного блока*, начинающегося ключевым словом *Given* и включающего в себя набор уравнений и неравенств.

Функция

$$\text{Find}(x_1, x_2, \dots, x_n)$$

возвращает значения n неизвестных x_1, x_2, \dots, x_n , обращающих уравнения в тождества, т. е. возвращает решение системы уравнений. В случае решения системы из n уравнений с n неизвестными эта функция возвращает вектор, состоящий из n элементов.

Функция *Find* позволяет находить и решение *одного уравнения с одной неизвестной*, хотя для этой цели обычно используется функция *root*.

Если в результате поиска не может быть получено решение с заданной точностью, то функция *Find* выдает сообщение об ошибке. В этом случае можно использовать функцию

$$\text{Minerr}(x_1, x_2, \dots, x_n),$$

возвращающую решение системы уравнений или одного уравнения (число уравнений должно быть равно числу неизвестных).

Порядок применения вычислительных блоков *Given..Find* и *Given..Minerr* для численного решения уравнений и систем уравнений:

- всем неизвестным, входящим в систему, присваивается начальное приближение;

- при необходимости задаются требуемые значения системным переменным *TOL* и *CTOL* (Constraint Tolerance – Допустимое отклонение ограничений);

- вводится ключевое слово *Given*, начинающее вычислительный блок;

- набираются уравнения и ограничения в виде неравенств (если необходимо) в произвольном порядке, каждое в отдельной формульной области, причем для записи используются операции отношения панели **Boolean** (*Логические*). Допускается использование двусторонних неравенств вида $a \leq x \leq b$;

- применяется функция *Find* или *Minerr*.

Внутри блока решения недопустимы следующие операции и выражения:

- ранжированные переменные;

- выражения, содержащие знак \neq ;

- локальное ($:=$) или глобальное (\equiv) определение переменных и функций,

за исключением выражения, в которое входит функция *Find* или *Minerr*;

- вложенный вычислительный блок.

В случае появления сообщения об ошибке вида

No solution was found. Try changing the guess value or the value of TOL or CTOL.

означающей, что решение не было найдено, рекомендуется изменить начальное приближение или значения системных переменных *TOL* и *CTOL*.

Пример 5. Решить уравнение

$$\left(\frac{x}{2}\right)^3 + \sin^2(2x) = 3\ln|x|. \quad (4)$$

Решение.

Приведите уравнение к стандартной форме $f(x) = 0$, перенося все слагаемые в его левую часть:

$$\left(\frac{x}{2}\right)^3 + \sin^2(2x) - 3\ln|x| = 0 \quad (5)$$

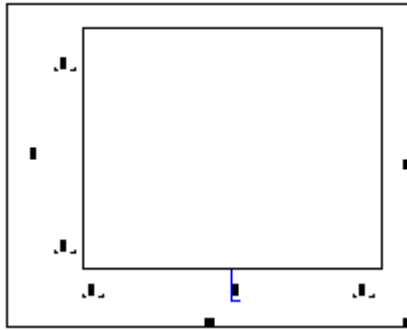
Определите функцию пользователя

$$f(x) := \left(\frac{x}{2}\right)^3 + \sin(2x)^2 - 3 \cdot \ln(|x|)$$

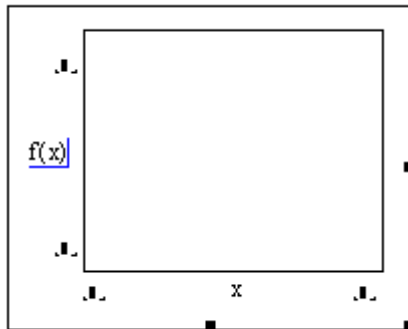
Вставьте в документ шаблон двумерного графика в декартовой системе координат. Для этого щелкните на пиктограмме с изображением графика



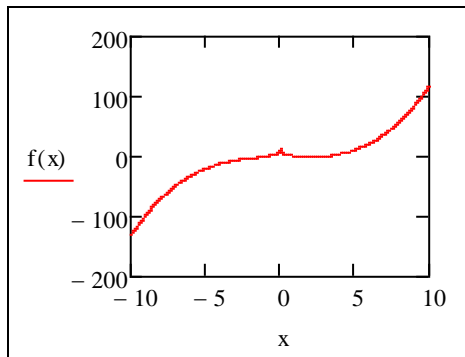
сначала на панели **Math** (*Математика*), затем на панели **Graph** (*График*) или выполните команду **Insert / Graph / X-Y Plot** (*Вставка / График / X-Y график*). В документе *Mathcad* отобразится шаблон двумерного графика



В поле ввода ■ в центре под осью абсцисс наберите имя аргумента x , а слева от оси ординат – имя функции с аргументом $f(x)$:



Щелкните левой кнопкой мыши вне шаблона. В документе *Mathcad* отобразится графическая область



По умолчанию *Mathcad* при построении графика использует

стандартный диапазон изменения x от -10 до 10 .

Уменьшите значения левой и правой границы x , заменив число -10 на -5 , а число 10 на 5 соответственно. Введите максимальное значение функции 15 , а минимальное -10 .

Отформатируйте график функции $f(x)$. Для этого щелкните правой кнопкой мыши на графической области и выполните команду **Format** (*Формат*) контекстного меню. В окне приложения *Mathcad* отобразится диалоговое окно, показанное на рисунке 5.

Добавьте координатную сетку, установив флажки слева от строк **Grid Lines** (*Вспомогательные линии*) для оси X и оси Y .

Отмените режим **Autogrid** и установите значения 10 и 5 в полях **Number of grids** (*Количество линий*) по оси X и Y соответственно.

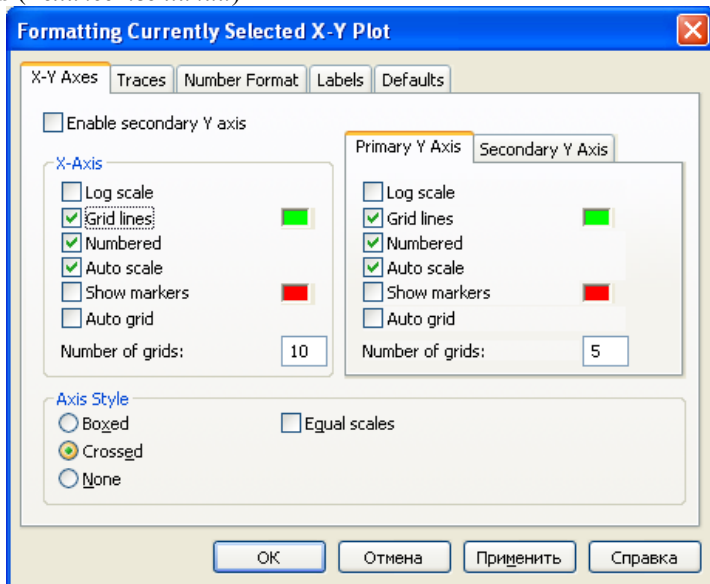
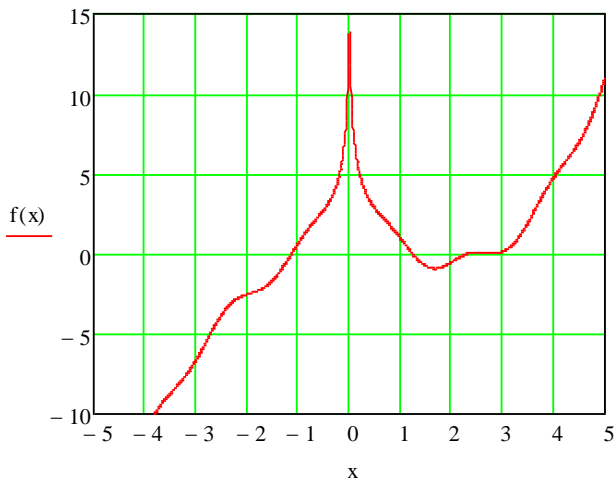


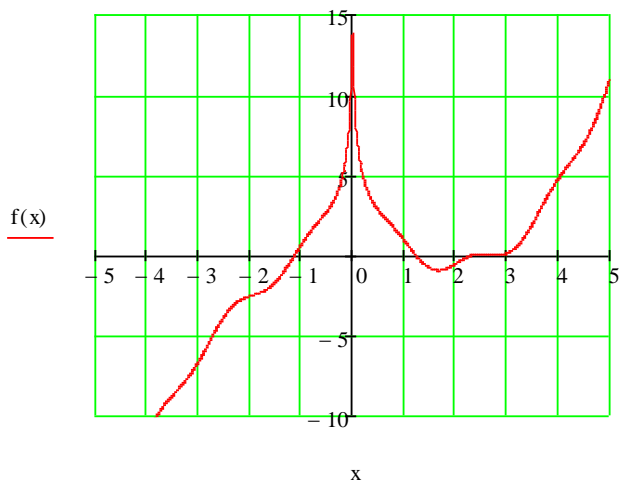
Рисунок 5 – Установка параметров форматирования декартова графика

Подтвердите внесенные изменения, щелкнув последовательно на кнопках **Apply** (*Применить*) и **OK**.

Графическая область примет вид:



Назначьте стиль осей графика **Crossed** (*Пересечение, Только оси*).
 После указанных преобразований график функции $f(x)$ примет вид:



Из графика функции $f(x)$ видно, что уравнение

$$\left(\frac{x}{2}\right)^3 + \sin^2(2x) = 3\ln|x| \quad (6)$$

имеет три корня, которые приблизительно равны: $x_1 \approx -1$; $x_2 \approx 1$; $x_3 \approx 2,5$.
 Этап отделения корней завершен.

Уточните теперь корни уравнения различными способами.

1 Присвойте начальное приближение переменной x и укажите точность поиска корня:

$$x := -1 \quad \text{TOL} := 0.0001$$

Уточните заданное приближение к значению корня с помощью функции *root*:

$$\begin{aligned} x1 &:= \text{root}(f(x), x) \\ x1 &= -1.1395 \end{aligned}$$

Выполните проверку, подтверждающую, что первый корень найден с заявленной точностью:

$$f(x1) = -2.9 \times 10^{-14}$$

Начальное приближение можно не задавать при использовании в качестве аргументов *root* границ отрезка нахождения корня. Уточнение второго корня произведите следующим образом:

$$\text{root}(f(x), x, 1, 2) = 1.232$$

2 Присвойте начальное приближение переменной x для уточнения третьего корня:

$$x := 2.5$$

Наберите служебное слово *Given*, а затем уравнение $f(x) = 0$, разделяя его левую и правую части знаком логического равенства (полу жирное равно **=**), который вставляется нажатием комбинация клавиш **Ctrl** **=**, или щелчком мыши на кнопке **=** панели **Boolean** (*Логические*). Завершите вычислительный блок оператором присваивания $x3 := \text{Find}(x)$ и выведите значение найденного третьего корня.

Фрагмент документа *Mathcad*, в котором уточняется корень уравнения с помощью вычислительного блока *Given...Find*, имеет вид:

$$\begin{aligned} x &:= 2.5 \\ \text{Given} \\ f(x) &= 0 \\ x3 &:= \text{Find}(x) \\ x3 &= 2.2917 \end{aligned}$$

3 Применение вычислительного блока *Given...Minerr* для нахождения первого корня осуществляется следующим образом:


```

x := -1
Given
f(x) = 0
Minerr(x) = -1.1395

```

Примечание – Для уточнения корня в данном примере установлена точность 0,0001. Поэтому целесообразно изменить формат вывода результатов (4 знака после десятичного разделителя) в диалоговом окне форматирования результатов на вкладке **Number Format**.

Пример 6. Решить систему уравнений

$$\begin{cases} 2y - x^2 = 1; \\ 3x - 4\cos(2y) = 0,5. \end{cases} \tag{7}$$

Решение.

Отделите решения системы графически. Определите две функции аргументов x и y соответственно, выразив для этого из первого уравнения системы y , а из второго – x :

$$f1(x) := \frac{1 + x^2}{2} \quad f2(y) := \frac{0.5 + 4\cos(2y)}{3}$$

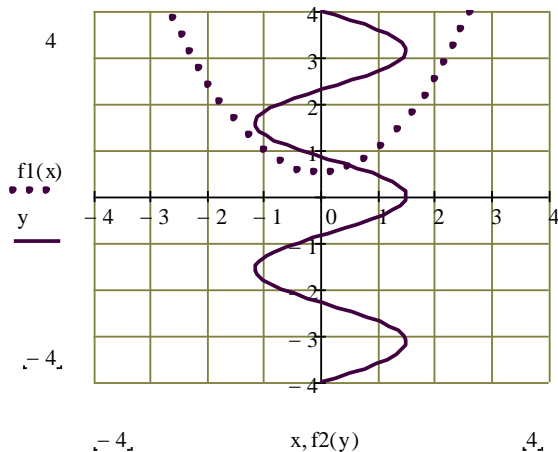
Для более детального построения графика создайте ранжированные переменные:

```

x := -4, -3.9..4   y := -4, -3.9..4

```

Вставьте в документ *Mathcad* шаблон графика **X-Y Plot**. Аргументы по оси абсцисс и ординат введите *через запятую*. После форматирования график примет вид:



Значения координат точки, в которой пересекаются линии графиков функций, являются решением системы уравнений. Присвойте приблизительные значения координат точки пересечения в качестве начального приближения для переменных x и y :

$$x := 0.6 \quad y := 0.6$$

Наберите ключевое слово *Given*, а затем уравнения системы, каждое в отдельной формульной области, отделяя левую и правую части знаком логического равенства **=** (**Ctrl** =). Решение уточните с помощью функции *Minerr*:

$$\begin{array}{l} \text{Given} \\ 2y - x^2 = 1 \\ 3x - 4 \cdot \cos(2y) = 0.5 \\ X := \text{Minerr}(x, y) \quad X = \begin{pmatrix} 0.538 \\ 0.644 \end{pmatrix} \end{array}$$

Выполните проверку найденного решения системы уравнений. Для этого задайте значение системной переменной **ORIGIN:=1**. Первый элемент вектора X – это значение переменной x , а второй – значение переменной y . Подставьте их в левые части уравнений системы и произведите вычисления:

$\begin{aligned} \text{ORIGIN} &:= 1 \\ 2 X_2 - (X_1)^2 &= 1 \\ 3 X_1 - 4 \cos(2 X_2) &= 0.5 \end{aligned}$

Сравните полученные значения 1 и 0,5 со значениями правых частей уравнений (7). Видно, что они совпадают. Следовательно, система уравнений решена правильно.

1.2.2 Решение полиномиальных уравнений. Функция `polyroots`

Для решения полиномиальных уравнений вида

$$v_n x^n + v_{n-1} x^{n-1} + \dots + v_1 x + v_0 = 0$$

или нахождения всех корней полинома степени n используют функцию

`polyroots(v)`,

возвращающую вектор-столбец длины n , состоящий из корней полинома, как действительных, так и комплексных. Аргументом функции `polyroots` является вектор v длины $n + 1$ ($1 \leq n \leq 100$), содержащий коэффициенты полинома.

При решении полиномиального уравнения в *Mathcad* необходимо:

- набрать в формульной области полином;
- выделить управляющим курсором переменную, которую надо найти;
- создать вектор из коэффициентов полинома командой **Symbolics / Polynomial Coefficients** (*Символика / Коэффициенты полинома*);
- скопировать вектор коэффициентов полинома в буфер обмена;
- набрать имя функции `polyroots`, вставить из буфера обмена вектор коэффициентов полинома в качестве аргумента функции и отобразить результаты нахождения корней, нажав клавишу равно $\boxed{=}$.

Второй способ включает:

- создание вектора v ;
- выполнение оператора присваивания

$$X := \text{polyroots}(v);$$

- визуализацию результатов решения X .

Доступ к каждому отдельному корню – элементу вектора X – осуществляется с помощью индекса, например, $X_i =$.

Пример 7. Решить уравнение

$$x^4 - 2x^2 - 7x + 9 = 0. \tag{8}$$

Решение.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix};$$

b – столбец свободных слагаемых,

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}.$$

Известно, что система линейных алгебраических уравнений имеет решение, если ее определитель отличен от 0, т. е. если

$$\det(A) = |A| \neq 0. \quad (10)$$

Умножим обе части матричного уравнения $Ax = b$ на обратную матрицу коэффициентов при неизвестных системы A^{-1} слева:

$$A^{-1}Ax = A^{-1}b. \quad (11)$$

С учетом того, что

$$A^{-1}A = E, \quad (12)$$

вектор-столбец решений системы принимает вид:

$$x = A^{-1}b. \quad (13)$$

Реализация данного алгоритма в *Mathcad* включает:

- формирование матрицы A коэффициентов при неизвестных и вектора-столбца b свободных слагаемых системы уравнений;
- нахождение решения системы по формуле $x := A^{-1}b$;
- вывод (отображение в документе) значений вектора решений системы уравнений $x =$.

Кроме того, *Mathcad* имеет встроенную функцию

$$\text{lsolve}(A, b),$$

возвращающую вектор-столбец решений системы линейных алгебраических уравнений. Аргументами функции *lsolve* являются матрица коэффициентов при неизвестных системы уравнений и столбец свободных слагаемых. Порядок решения аналогичен рассмотренному, но вместо вычисления по формуле $x := A^{-1}b$ обращаются к функции *lsolve* в операторе присваивания

$x := \text{lsolve}(A, b)$.

Реализовать широко известный метод Гаусса решения систем линейных уравнений позволяет встроенная функция $rref(M)$, возвращающая ступенчатый вид матрицы M . Если в качестве аргумента взять расширенную матрицу системы, то в результате применения $rref$ получится матрица, на диагонали которой – единицы, а последний столбец представляет собой столбец решений системы.

Решение системы линейных уравнений можно найти с помощью блоков *Given...Find*, *Given...Minerr*.

Пример 8. Решить систему линейных уравнений

$$\begin{cases} 3x + 4y - 5z = 1; \\ 7x - 4y + z = -2; \\ 2y - 9z = 5. \end{cases} \quad (14)$$

Сделать проверку полученного решения.

Решение.

1 Использование блока *Given...Find*.

Задайте всем неизвестным, входящим в систему уравнений, начальные приближения, например,

$$\boxed{x:=0 \quad y:=0 \quad z:=0}$$

Наберите ключевое слово *Given* и уравнения системы, разделяя левую и правую часть каждого из них логическим знаком равенства $=$, который вводится нажатием комбинации клавиш **Ctrl** =. Завершите вычислительный блок вводом оператора присваивания

$$X := \text{Find}(x, y, z) \quad (15)$$

Mathcad-код и результаты решения системы уравнений в виде вектора, состоящего из трех элементов, представлены в следующем фрагменте документа *Mathcad*:

Given	
$3x + 4y - 5z = 1$	
$7x - 4y + z = -2$	
$2y - 9z = 5$	
$X := \text{Find}(x, y, z)$	$X = \begin{pmatrix} -0.345 \\ -0.257 \\ -0.613 \end{pmatrix}$

Сделайте проверку результатов решения системы уравнений, подставив полученные значения неизвестных в выражения для левых частей уравнений системы и завершив набор каждого такого выражения нажатием клавиши =:

$$\begin{array}{l} 3 X_1 + 4 X_2 - 5 X_3 = 1 \\ 7 X_1 - 4 X_2 + X_3 = -2 \\ 2 X_2 - 9 X_3 = 5 \end{array}$$

Видно, что полученные значения выражений левых частей уравнений системы совпадают с константами их правых частей, т. е. найденные значения неизвестных действительно являются решением системы уравнений.

В данном примере системная переменная `ORIGIN = 1`.

2 Использование блока *Given...Minerr*.

Порядок решения системы этим способом аналогичен порядку использования блока *Given...Find*. Результаты решения вместе с результатами проверки представлены в следующем фрагменте документа:

$$\begin{array}{l} x := 0 \quad y := 0 \quad z := 0 \\ \text{Given} \\ 3x + 4y - 5z = 1 \\ 7x - 4y + z = -2 \\ 2y - 9z = 5 \\ X := \text{Minerr}(x, y, z) \quad X = \begin{pmatrix} -0.345 \\ -0.257 \\ -0.613 \end{pmatrix} \\ 3 X_1 + 4 X_2 - 5 X_3 = 1 \\ 7 X_1 - 4 X_2 + X_3 = -2 \\ 2 X_2 - 9 X_3 = 5 \end{array}$$

3 Решение системы линейных уравнений матричным способом.

Создайте матрицу A , состоящую из коэффициентов при неизвестных системы. Для этого наберите $A :=$ и вызовите диалоговое окно создания массивов (**Ctrl M**). Число строк (**Rows**) и столбцов (**Columns**) матрицы данной системы равно 3. Заполните шаблон матрицы как показано ниже:

$$A := \begin{pmatrix} 3 & 4 & -5 \\ 7 & -4 & 1 \\ 0 & 2 & -9 \end{pmatrix}$$

Задайте вектор b правых частей системы уравнений, введя сначала $b :=$, затем вставив шаблон матрицы (**Ctrl M**) с количеством строк (**Rows**) и столбцов (**Columns**) равным 3 и 1 соответственно и заполнив его поля ввода необходимыми числами:

$$b := \begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix}$$

Представьте расчетную формулу матричного метода оператором присваивания и выведите найденные значения вектора-решения:

$$X := A^{-1}b \quad X = \begin{pmatrix} -0.345 \\ -0.257 \\ -0.613 \end{pmatrix}$$

Найдите решение системы уравнений с помощью функции *lsolve*:

$$X := \text{lsolve}(A, b) \quad X = \begin{pmatrix} -0.345 \\ -0.257 \\ -0.613 \end{pmatrix}$$

Получите решение системы уравнений посредством функции *rref*:

$$AR := \text{augment}(A, b) \quad X := \text{rref}(AR) \langle 4 \rangle \quad X = \begin{pmatrix} -0.345 \\ -0.257 \\ -0.613 \end{pmatrix}$$

Полученная путем объединения матрицы при неизвестных системы и столбца правых частей уравнений матрица AR называется *расширенной матрицей* системы уравнений.

Проверка правильности решения системы уравнений X , полученного матричным способом, заключается в вычислении произведения $A \cdot X$ и сравнении найденного вектора

$$A \cdot X = \begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix}$$

с вектором-столбцом b .

2 Порядок выполнения лабораторной работы

1 Загрузите *Mathcad*.

2 Сохраните в личной папке на диске **z:** новый документ *Mathcad* с именем **fi04**.

Примечание – В процессе работы целесообразно периодически сохранять документ, нажимая комбинацию клавиш **Ctrl S**.

3 Вставьте в документ *Mathcad* текстовую область и введите в нее текст:


Лабораторная работа 4

Массивы в *Mathcad*.


Решение уравнений и систем уравнений


4 В новой текстовой области введите фамилию, имя, отчество, учебный шифр и номер варианта.


5 Задайте значение системной переменной **ORIGIN** равным 1.

6 Создайте квадратные матрицы *A* и *B* размером 4×4 , используя шаблон  панели **Matrix** (*Матрица*).

7 Вычислите определитель матрицы *A*, применив шаблон оператора .

8 Найдите матрицу, обратную *B*, с помощью шаблона оператора .

9 Получите транспонированные матрицы A^T и B^T , используя шаблон оператора  панели **Matrix** (*Матрица*).

10 Извлеките из матрицы *A* ее третий столбец, используя шаблон оператора .

11 Извлеките из матрицы *B* ее вторую строку, используя последовательно операции транспонирования и выделения столбца.

12 Найдите для элементов главной диагонали матрицы *A* их сумму и произведение, а также минимальный и максимальный элементы.

13 Выполните задание согласно варианту (таблица 2).

Таблица 2 – Задачи по обработке массивов

Вариант	Задача
1	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = \cos i + 2$. Найти сумму элементов и минимальный элемент v. Создать матрицу <i>C</i> размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = \sqrt{2i} + \text{tg } j$. Подсчитать сумму элементов первого столбца, произведение элементов второй строки и среднее арифметическое элементов матрицы <i>C</i>. На основе третьей строки матрицы <i>C</i> создать диагональную матрицу и найти ее определитель</p>

Продолжение таблицы 2

Вариант	Задача
2	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = \sin i - 3$. Найти произведение элементов и максимальный элемент v. Создать матрицу C размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = \operatorname{ctg} i - \sqrt{2j}$. Подсчитать сумму элементов второго столбца, минимальный элемент третьей строки и произведение отрицательных элементов матрицы C. На основе четвертого столбца матрицы C создать диагональную матрицу и найти ее обратную</p>
3	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = 2\sin i + 6$. Найти косинус каждого элемента v и сумму элементов v. Создать матрицу C размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = \sqrt{3j} + \ln i$. Подсчитать минимальный элемент второго столбца, произведение элементов первой строки и произведение положительных элементов матрицы C. На основе второго столбца C создать диагональную матрицу и получить ее подматрицу, состоящую из первых трех строк</p>
4	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = \ln i + 3$. Найти среднее арифметическое и произведение элементов v. Создать матрицу C размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = \sqrt{5j} + \sin j$. Подсчитать сумму элементов третьего столбца, максимальный элемент второй строки и количество положительных элементов матрицы C. На основе первой строки C создать диагональную матрицу и получить ее подматрицу, состоящую из первых трех столбцов</p>
5	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = 0,2e^i$. Найти среднее геометрическое и сумму элементов v. Создать матрицу C размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = 2\sin \sqrt{i+2j}$. Подсчитать произведение элементов четвертого столбца, сумму элементов второй строки и количество отрицательных элементов матрицы C. Получить матрицу, состоящую из нечетных строк матрицы C</p>
6	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = \lg^2 3i$. Найти среднее арифметическое и произведение элементов v. Создать матрицу C размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = 3\cos \sqrt{2i-j}$. Подсчитать максимальный элемент первого столбца, произведение элементов четвертой строки и минимальный положительный элемент матрицы C. Получить матрицу, состоящую из</p>

	нечетных столбцов матрицы C
--	-------------------------------

Продолжение таблицы 2

Вариант	Задача
7	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = \sqrt[3]{i+2} - 1$. Найти минимальный элемент v и сумму элементов v. Создать на основе v диагональную матрицу и транспонировать ее. Создать матрицу C размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = 4 \ln^2(i+j+1)$. Найти максимальный элемент второго столбца, произведение элементов третьей строки и максимальный отрицательный элемент матрицы C. Получить массив из элементов, стоящих на пересечении первых двух строк и столбцов C</p>
8	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = \cos(\ln i)$. Найти максимальный элемент v и произведение элементов v. Создать на основе v диагональную матрицу и обратную ей. Создать матрицу C размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = 2i - \sin 3j$. Подсчитать сумму элементов третьего столбца, минимальный элемент первой строки и количество неотрицательных элементов матрицы C. Получить матрицу из элементов, стоящих на пересечении последних двух строк и столбцов C</p>
9	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = \sin(\lg i)$. Найти сумму и среднее арифметическое элементов v. Создать матрицу C размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = 2 \cos i - 3j$. Подсчитать произведение элементов четвертого столбца, максимальный элемент второй строки и количество неположительных элементов матрицы C. Получить матрицу из элементов четных строк и столбцов C</p>
10	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = \sqrt[3]{\ln i + i^2}$. Найти произведение и среднее геометрическое элементов v. Создать матрицу C размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = \operatorname{ctg} i - 3j^2$. Подсчитать сумму элементов первого столбца, минимальный элемент третьей строки и заменить положительные элементы матрицы C единицей. Получить матрицу из элементов нечетных строк и столбцов C</p>
11	<p>Задать вектор-столбец v, состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = \sqrt[3]{\sin 2i}$. Найти минимальный элемент и произведение элементов v. Создать матрицу C размером 4×4, каждый элемент которой вычисляется по формуле $C_{i,j} = \operatorname{tg}^2 i - \sqrt{j}$. Подсчитать сумму элементов второго столбца, максимальный элемент второй строки и заменить отрицательные элементы матрицы C числом 2.</p>

	Получить матрицу из элементов четных столбцов и главной диагонали C
--	---

Окончание таблицы 2

Вариант	Задача
12	Задать вектор-столбец v , состоящий из четырех элементов, каждый из которых вычисляется по формуле $v_i = \sqrt{2 - \sin 2i}$. Найти произведение и максимальный из элементов v . Создать матрицу C размером 4×4 , каждый элемент которой вычисляется по формуле $C_{i,j} = \cos^3 i - \sqrt{2+j}$. Подсчитать минимальный элемент третьего столбца, сумму элементов первой строки матрицы C . Заменить отрицательные элементы матрицы C их модулями. Получить матрицу из элементов нечетных столбцов матрицы C

14 Решите уравнение (таблица 3) с точностью 10^{-5} : отделите корень графически и уточните его с помощью функции *root*; блока *Given...Find*; блока *Given...Minerr*.

15 Решите систему уравнений (см. таблицу 3), отделив решение графически и уточнив с помощью блока *Given..Minerr*. Выполните проверку решения.

Таблица 3 – Нелинейные уравнения и системы уравнений

Вариант	Уравнение	Система уравнений
1	$\sin^2 x = 1,5 \ln(x + 2,5) - 1,7$	$\begin{cases} x^2 = 3 - y; \\ \sin(y + 2) = 1,6 + 2x \end{cases}$
2	$x + \cos^2 x = -1,5x^2$	$\begin{cases} \sin(2x - 1,7) = 1,4 - y; \\ x - y^2 = 2,8 \end{cases}$
3	$\sqrt[3]{\sin x} = 0,7 \cdot (1 + x^2) - 3,5$	$\begin{cases} \cos(y + 0,5) = 0,9 - x; \\ \cos(x - 0,3) = 1,8 + 2y \end{cases}$
4	$(x + 2)^2 \ln(3x + 2) = 2,5$	$\begin{cases} \cos(2,2y - 1) = 1,2 - x; \\ \sin(1,8x + 1) = y - 0,7 \end{cases}$
5	$e^{2-x} + \sin^2(x) = 1,8$	$\begin{cases} \cos(y + 0,5) = 0,9 - x; \\ 0,4x^2 = 1,8 + 2y \end{cases}$
6	$\sin^{2x+1}(x + 2) = 4,4x$	$\begin{cases} \sin(y + 1) - x = 1,9; \\ \cos(x + 0,05) = 2 - 2,1y \end{cases}$
7	$3^{x-2} - 2x^3 = \arctg(x)$	$\begin{cases} 0,5x^2 - y = 1; \\ \cos(y - 2,3) + x = 0,3 \end{cases}$

8	$0,3x + \sin^3(3,5x+1) = 2$	$\begin{cases} \cos(x+0,5) + y = 1,5; \\ 2,2x - y^2 = 1,2 \end{cases}$
---	-----------------------------	--

Окончание таблицы 3

Вариант	Уравнение	Система уравнений
9	$e^x = 85(\cos^2 2x + 1)$	$\begin{cases} \sin(x+3) - 0,9y = 1,8; \\ \cos(y-2,5) = 2,5 - x \end{cases}$
10	$\log_5^2(x+1,7) + 2x = 0$	$\begin{cases} \cos(x^3 - 1,2) + y = 0,4; \\ x - \cos(0,9y + 0,2) = 3,3 \end{cases}$
11	$1 = 0,25x(\cos^3 x + 2)$	$\begin{cases} \sin(x+1) - 0,9y = 1,3; \\ 2x + \cos y = 2 \end{cases}$
12	$\operatorname{tg}^2(x+2) - x = 2,5$	$\begin{cases} \cos(y+0,5) = 2 + x; \\ \sin x = 1 + 3y \end{cases}$

16 Решите полиномиальное уравнение (таблица 4).

Таблица 4 – Полиномиальные уравнения

Вариант	Уравнение	Вариант	Уравнение
1	$x^4 - 6x^2 + 1,2x - 8,5 = 0$	7	$x^4 + 5x^3 + x^2 + 5x - 2 = 0$
2	$x^4 + 6x^3 + 1,4x^2 - 2x - 2,8 = 0$	8	$x^4 + 6x^3 + 1,3x^2 + 2x + 2 = 0$
3	$x^4 + 2,5x^3 - 4x^2 + x - 5 = 0$	9	$x^4 + 2x^3 + x^2 + 2x - 6,4 = 0$
4	$x^4 - 8x^3 + 1,6x + 15 = 0$	10	$x^4 - x^3 - 1,7x^2 - 8,2x - 6 = 0$
5	$x^4 + 2x^3 - 4x^2 + 1,6x - 8 = 0$	11	$x^4 + 2x^3 + 4,5x^2 + x - 3 = 0$
6	$x^4 + x^3 + 2x^2 - 4,5 = 0$	12	$2x^4 - 7x^3 - 2,3x^2 + 4,3x - 1 = 0$

17 Решите систему линейных алгебраических уравнений (таблица 5) с помощью блока *Given...Find* и матричным способом.

Таблица 5 – Системы линейных уравнений

Вариант	Система линейных уравнений	Вариант	Система линейных уравнений
1	$\begin{cases} 10,6x_1 - 1,4x_2 + 1,1x_3 = 1,8; \\ 2,6x_1 + 14,4x_2 - 1,5x_3 = 2; \\ 1,5x_1 + 1,3x_2 + 1,2x_3 = 1,6 \end{cases}$	3	$\begin{cases} 1,3x_1 - 11,7x_2 + 1,8x_3 = 1,3; \\ 1,1x_1 - 10,5x_2 - 1,7x_3 = 1,1; \\ 1,5x_1 - 0,5x_2 + 1,8x_3 = 10 \end{cases}$

2	$\begin{cases} 2x_1 + 0,5x_2 - 32,1x_3 = 2,5; \\ 12x_1 - 23x_2 + 30,2x_3 = 1,3; \\ 3,4x_1 - 0,7x_2 + 3,3x_3 = -1 \end{cases}$	4	$\begin{cases} 15x_1 - 1,2x_2 - 1,8x_3 = 1,4; \\ 7,5x_1 + 2,3x_2 + 7,2x_3 = 4,2; \\ 1,3x_1 + 1,6x_2 - 1,3x_3 = 10 \end{cases}$
---	---	---	--

Окончание таблицы 5

Вариант	Система линейных уравнений	Вариант	Система линейных уравнений
5	$\begin{cases} 1,9x_1 + 23,1x_2 + 1,3x_3 = 2; \\ -14x_1 + 1,7x_2 + 20,4x_3 = 1; \\ 17x_1 - 1,6x_2 - 2,2x_3 = 2,8 \end{cases}$	9	$\begin{cases} 1,6x_1 - 1,8x_2 + 5,2x_3 = 2,7; \\ 2,7x_1 - 6,8x_2 + 10,3x_3 = 2,1; \\ 2,3x_1 + 6,7x_2 - 9,9x_3 = 27,3 \end{cases}$
6	$\begin{cases} 8,5x_1 + 3,2x_2 - 4,6x_3 = 12,8; \\ 4,8x_1 - 2,7x_2 + 12,4x_3 = 8,9; \\ 2,5x_1 + 6,6x_2 - 1,8x_3 = -10,6 \end{cases}$	10	$\begin{cases} 0,7x_1 - 7,9x_2 + 16,5x_3 = -6,6; \\ 15,6x_1 + 6,6x_2 + 14,4x_3 = 12,4; \\ 7,5x_1 + 12,2x_2 - 8,3x_3 = 9,2 \end{cases}$
7	$\begin{cases} 5,4x_1 - 2,3x_2 + 16,7x_3 = -4,5; \\ 7,4x_1 - 12,3x_2 - 4,2x_3 = 5,4; \\ 2,7x_1 + 17,7x_2 + 6,4x_3 = 8,2 \end{cases}$	11	$\begin{cases} 6,8x_1 - 8,3x_2 + 14,3x_3 = 17,2; \\ 7,6x_1 + 7,7x_2 - 18,3x_3 = -5,4; \\ 13,2x_1 - 5,2x_2 - 6,5x_3 = 6,5 \end{cases}$
8	$\begin{cases} 32x_1 - 31,9x_2 + 13,4x_3 = 2,6; \\ 1,7x_1 + 21,5x_2 - 4,9x_3 = 15,9; \\ 7,6x_1 - 14,8x_2 + 6,2x_3 = -1,2 \end{cases}$	12	$\begin{cases} 18,5x_1 + 5,2x_2 - 8,3x_3 = 21,5; \\ 8,3x_1 - 3,4x_2 + 4,3x_3 = 6,5; \\ 7,4x_1 + 11,5x_2 + 2,3x_3 = -6,2 \end{cases}$

18 Подготовьте документ для печати и распечатайте его.

19 Завершите работу в *Mathcad*.

Контрольные вопросы

- 1 Сформулируйте определение массива, вектора, матрицы.
- 2 Назначение переменной ORIGIN. Как задается ее значение?
- 3 Перечислите способы создания массивов.
- 4 Перечислите операции для работы с массивами.
- 5 Как обратиться к отдельному элементу вектора и матрицы?
- 6 Перечислите и определите векторные и матричные функции.
- 7 Как найти начальное приближение корней уравнений?
- 8 Какие функции для решения одного уравнения известны в *Mathcad*?
- 9 Как системная переменная TOL влияет на решение уравнения?
- 10 Назовите функции для решения систем уравнений в *Mathcad* и особенности их применения. Дайте их сравнительную характеристику.
- 11 Что такое вычислительный блок и какова его структура?
- 12 Какой знак равенства используется при составлении уравнений в вычислительном блоке решения? Нажатием какой комбинации клавиш вставляется этот знак в документ?
- 13 Какие выражения недопустимы внутри вычислительного блока решения?
- 14 Какие способы решения систем линейных алгебраических уравнений

реализуются в *Mathcad*?

15 Назовите способы решения матричных уравнений.

ЛАБОРАТОРНАЯ РАБОТА 5

ТАБЛИЧНЫЙ ПРОЦЕССОР *EXCEL*

Цель работы: получить и упрочить навыки работы с табличным процессором *Microsoft Excel*. Научиться выполнять расчеты, вводить и обрабатывать массивы данных, строить и форматировать графики функций и диаграммы, решать уравнения и системы уравнений.

1 Сведения из теории

1.1 Основные возможности *Excel*

Электронная таблица – это компьютерный эквивалент таблицы, состоящей из строк и столбцов, на пересечении которых располагаются ячейки, содержащие число, формулу или текст.

Электронные таблицы создаются и обрабатываются программами, называемыми *табличными процессорами*. Часто термин «*Электронные таблицы*» используют для именованя табличного процессора.

Одним из наиболее распространенных табличных процессоров является *Microsoft Excel*, входящий в *Microsoft Office*.

Сферой применения *Excel* является выполнение экономических расчетов, статистическая обработка данных, решение различных инженерных задач. *Excel* предоставляет возможность построения графических зависимостей и диаграмм, создания простых баз данных и многое другое.

Запуск *Excel* осуществляется командой **Start / All Programs / Microsoft Excel** (**Пуск / Все программы / Microsoft Excel**)¹ или двойным щелчком на соответствующем ярлыке на рабочем столе *Windows*.

1.1.1 Базовые понятия и приемы работы в *Excel*

Окно приложения *Excel* и его основные настройки. Интерфейс *Excel* содержит все стандартные элементы приложений *Microsoft Office* (рисунки 1):

- строку заголовка с кнопкой системного меню и кнопками **Свернуть**, **Развернуть**, **Заккрыть**;
- главное меню программы и панели инструментов;

¹ Версии *Microsoft Excel 97–2003*

- строку имени и формулы, которые характерны только для *Excel*;
- рабочую область (окно документа);
- курсорную рамку;
- указатель мыши;
- строку состояния;
- вертикальную и горизонтальную линейки прокрутки.

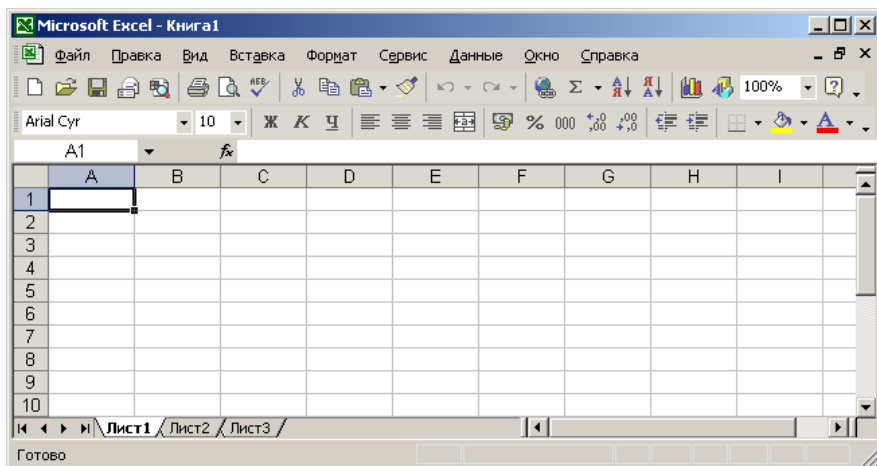


Рисунок 1 – Окно табличного процессора *Excel*

Настройка интерфейса *Excel* осуществляется:

- командами меню **Вид**;
- посредством диалоговых окон, вызываемых командами **Сервис / Параметры** и **Сервис / Настройка**.

Командами меню **Вид** можно переключать режимы отображения рабочего листа (электронной таблицы).

По умолчанию для вновь создаваемых документов установлен режим отображения *Обычный*, который используется для выполнения большинства задач *Excel*, таких, как ввод и обработка данных, форматирование данных и ячеек, вычисления, построение диаграмм и т. д. Режим *разметки страницы* предназначен для оформления страниц документа и подготовки их к печати.

Подключение дополнительных панелей инструментов производится с помощью контекстного меню панелей инструментов или щелчком на необходимом пункте списка, вызываемого командой **Вид / Панели инструментов**.

Рабочая книга. Основным документом *Excel* является *рабочая книга* – файл с произвольным именем и расширением *.xls*. Чтобы создать новую книгу, необходимо выполнить команду **Файл / Создать** или нажать

комбинацию клавиш **Ctrl N**.

Сохранение рабочей книги осуществляется командами:

Файл / Сохранить – для сохранения уже существующего файла под тем же именем;

Файл / Сохранить как – для сохранения с новым именем или в другом каталоге.

При сохранении вновь созданной книги необходимо в диалоговом окне **Сохранить как** указать ее имя и расположение в файловой системе.

Свойства рабочей книги собраны в диалоговом окне **Свойства**, вызываемом командой **Файл / Свойства**. *Основные параметры* настраиваются в диалоговом окне **Параметры**, отображаемом при выполнении команды **Сервис / Параметры**.

Работа с листами. По умолчанию рабочая книга *Excel* содержит три *листа*, именуемые по умолчанию Лист1, Лист2, Лист3. Ярлыки листов располагаются в нижней левой части окна приложения *Excel*, как показано на рисунке 2. Активизация необходимого листа достигается щелчком на его ярлыке.

Чтобы *переименовать* лист *Excel*, дважды щелкните на его ярлыке левой кнопкой мыши, введите имя листа и нажмите клавишу **Enter**.

Чтобы *вставить* (добавить) в рабочую книгу новый рабочий лист *Excel*, выполните команду **Вставка / Лист**, а чтобы *удалить* лист – команду **Правка / Удалить лист**.

Основные *операции с листами* доступны также в *контекстном меню* листа (см. рисунок 2).

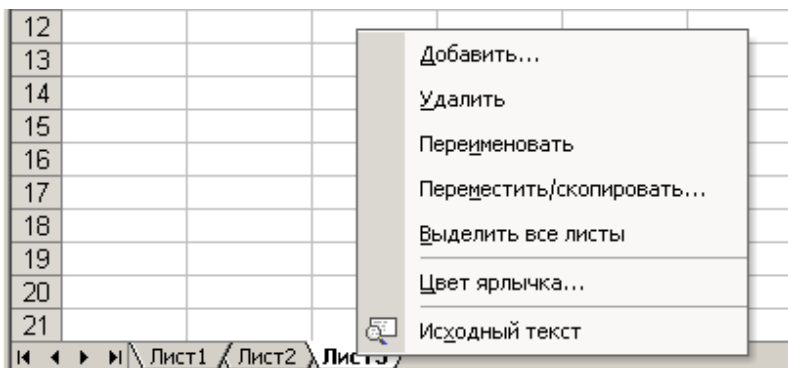



Рисунок 2 – Кнопки прокрутки, ярлыки и контекстное меню листов

Ячейка. Основными объектами электронной таблицы *Excel* являются ячейка, строка и столбец.

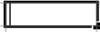
Ячейка – первичный элемент таблицы, содержащий данные. Каждая

ячейка таблицы имеет уникальный адрес, который состоит из имени столбца (А, В, ..., Z, AA, AB, ..., AZ, ..., IV) и номера строки (нумеруются арабскими цифрами), например, В5 – ячейка, находящаяся на пересечении столбца В и строки с номером 5. Адрес *текущей (активной) ячейки* отображается в строке имени ячейки, расположенной обычно под панелями инструментов. Кроме адреса ячейка может иметь имя, которое задается пользователем и может быть использовано в расчетах.

Адрес ячейки при использовании в формулах иногда называют *ссылкой*.

При работе с электронными таблицами следует обращать внимание на оформление ячейки. Полу жирная курсорная рамка , называемая *указателем ячейки*, с маркером-квадратиком в нижнем правом углу, определяет текущую ячейку. Ввод, удаление данных и некоторые другие действия относятся по умолчанию к текущей ячейке. Указатель ячейки (курсорная рамка) перемещается по рабочему листу щелчком левой кнопкой мыши или с помощью клавиш управления курсором. Перемещение вдоль строки слева направо осуществляется нажатием клавиши **Tab** клавиатуры, а справа налево – нажатием комбинации клавиш **Tab Shift**.

Чтобы ввести данные в текущую ячейку, необходимо набрать данные на клавиатуре и нажать клавишу **Enter**. При этом текущей станет ячейка, находящаяся под ячейкой, в которую введены данные. Нажатие комбинации клавиш **Shift Enter** приводит к перемещению на ячейку *вверх*.

Тонкая курсорная рамка  с мигающим текстовым курсором обозначает ячейку, находящуюся в *режиме ввода (редактирования) данных*. Чтобы войти в режим редактирования ячейки достаточно дважды щелкнуть левой кнопкой мыши на ячейке.

Внимание! Ввод данных на рабочем листе в активную ячейку (выделенную полу жирной курсорной рамкой) удаляет все хранящиеся в ней данные.

Удаление данных из активной ячейки (очистка ячейки) достигается:

- нажатием клавиши **Delete** клавиатуры;
- выполнением команды **Правка / Очистить / Содержимое**.

Данные в ячейках. В ячейку *Excel* можно поместить:

- число;
- текст (строку символов или символ);
- формулу.

Примечание – Ввод формулы всегда начинается со знака = (равно).

По умолчанию число выравнивается в ячейке по правому краю, а текст – по левому.

Если ширина ячейки недостаточна для отображения содержащегося в ней *текста*, то текст отображается поверх соседних ячеек, расположенных

справа (если они свободны).

Строка текста в ячейке разбивается на две нажатием комбинации клавиш **Alt Enter**.

Числовые данные допускают представление в одном из следующих форматов: общий, числовой, дата, время, финансовый, процентный и др. Изменение формата данных осуществляется в диалоговом окне **Формат ячеек**, вызываемом командой **Формат / Ячейки**.

Если ширина столбца недостаточна для вывода числового значения, то в ячейке отображается строка символов решетки #####. Данная ошибка преодолевается командой **Формат / Столбец / Автоподбор ширины**.

Формула в *Excel* – начинающееся со знака = (равно) математическое выражение, составленное из адресов или имен ячеек, обращений к функциям, арифметических и логических операций и скобок. Адреса ячеек исполняют роль операндов выражения.

Например,

$$=2 * \text{SIN} (A4) + \$B\$5.$$

Примечание – Операнд в виде обращения к функции, например, $\text{SIN}(A4)$, всегда включает имя функции, открывающуюся скобку, аргумент функции (адрес ячейки или выражение) и закрывающуюся скобку.

Элемент формулы, который используется для указания на данные какой-либо ячейки электронной таблицы, называется *ссылкой* или *адресом*.

Относительная ссылка, например, A4, изменяется при копировании формулы. При этом правило вхождения адреса ячейки в формулу относительно нового местоположения формулы сохраняется.

Абсолютная ссылка при копировании формулы остается неизменной и записывается, например, так: \$B\$5. В формулах допускается использование *частично абсолютных ссылок*, фиксирующих имя столбца или номер строки.

Например,

D\$3 – фиксируется номер строки,

\$F2 – фиксируется номер столбца.

Вводить адреса ячеек (ссылки) в формулу можно щелчком мыши по ячейке, значение которой используется в формуле.

Если формула набрана верно, то после нажатия клавиши **Enter** в ячейке отобразится вычисленное по формуле значение. Если при вводе формулы допущены ошибки, не позволяющие выполнить вычисления, то в ячейке появится сообщение об ошибке. Наиболее часто встречаются следующие ошибки:

- **#ЗНАЧ!** – при использовании недопустимого типа (формата) аргумента или операнда, например, при сложении числовых и текстовых данных;

- *#ИМЯ?* – Excel не может распознать имя, используемое в формуле;
- *#ДЕЛ/0!* – при попытке деления на 0;
- *#ССЫЛКА!* – при использовании в формуле недопустимой ссылки на ячейку и др.

Форматирование ячеек. Изменение внешнего вида данных в активной ячейке или в выделенном диапазоне ячеек производится в диалоговом окне **Формат ячейки**, вызываемом командой **Формат / Формат ячейки** или из контекстного меню ячейки.

К операциям форматирования относятся:

- изменение числовых форматов или формы представления чисел (вкладка **Число**);
 - выравнивание текста и чисел в ячейках по горизонтали и по вертикали, изменение направления текста и др. (вкладка **Выравнивание**);
 - изменение шрифта, начертания, размера и цвета символов (вкладка **Шрифт**);
 - выбор типа обрамления ячеек и цвета границы (вкладка **Граница**);
 - заливка ячеек, цвет фона или фоновый рисунок ячеек (вкладка **Вид**);
 - защита ячеек от несанкционированного доступа (вкладка **Защита**).
- Вкладка **Выравнивание** показана на рисунке 3.

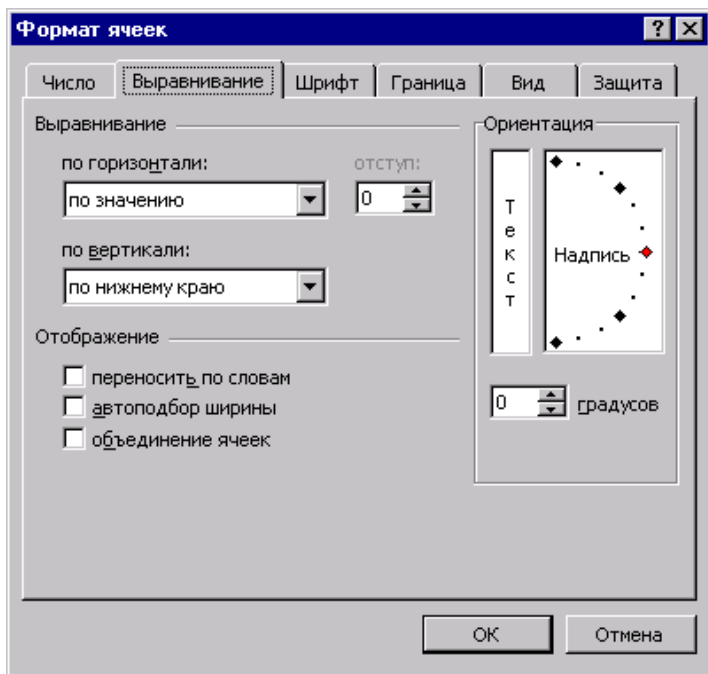


Рисунок 3 – Вкладка **Выравнивание** диалогового окна **Формат ячеек**

1.1.2 Некоторые приемы работы в Excel

Перемещение по рабочему листу всегда сопровождается активизацией ячейки. При этом используются клавиши управления курсором, клавиши **Tab** (↹ – в ячейку справа), **Home** (в начало строки), **Page Up** (на экранную страницу вверх), **Page Down** (на экранную страницу вниз) и **Enter** (на ячейку вниз с подтверждением введенных данных), а также ряд комбинаций клавиш.

Чтобы *перейти* в любую видимую ячейку, необходимо навести указатель мыши в виде белого толстого крестика на заданную ячейку, и щелкнуть левой кнопкой мыши.

Изменение ширины столбца достигается перемещением границы заголовка столбца с помощью указателя мыши в виде двусторонней стрелки на требуемое расстояние.

Чтобы *изменить высоту строки*, следует переместить мышью нижнюю границу заголовка строки до нужной высоты.

При выполнении этих действий отображается всплывающая подсказка со значением текущей ширины (высоты).

Если требуется изменить ширину нескольких столбцов (строк) так, чтобы все они имели одинаковую ширину (высоту), их следует предварительно


выделить, а затем изменить размеры только одного столбца (одной строки).

Изменение размеров столбцов и строк электронной таблицы производится также посредством меню **Формат** выбором соответствующего пункта.

Выделение диапазонов ячеек производится при необходимости выполнить одни и те же действия (например, форматирование) для нескольких ячеек, столбцов или строк. Чтобы выделить *диапазон ячеек*, щелкните на ячейке в любом углу и, удерживая нажатой левую кнопку мыши, расширьте рамку выделения до нужных размеров. Указатель мыши при этом должен оставаться в виде белого толстого крестика. Для выделения диапазона ячеек можно использовать клавишу **Shift** в сочетании с клавишами управления курсором или щелчком левой клавишей мыши в противоположном углу диапазона. При *выделении диапазона несмежных ячеек* удерживают нажатой клавишу **Ctrl**.

Чтобы *выделить строку (столбец)*, достаточно щелкнуть левой кнопкой мыши на номере строки (имени столбца). Щелчок правой кнопкой мыши на объекте сопровождается его выделением и появлением контекстного меню объекта.

Для *выделения всего листа* целиком достаточно щелкнуть на кнопке в виде серого прямоугольника, расположенной на пересечении столбца номеров строк и строки имен столбцов.

Объединение смежного диапазона ячеек производится щелчком мыши на кнопке  (**Объединить и поместить в центре**) панели инструментов **Стандартная**.

Добавление ячеек, строк и столбцов на лист производится из главного меню **Вставка** или из контекстного меню ячейки выбором одной из команд **Добавить ячейки...**

Excel предоставляет инструмент *автозаполнение ячеек*, значительно ускоряющий процесс ввода данных. В простейшем случае автозаполнение ячеек осуществляется следующим образом.

Щелкните мышью на какой-нибудь ячейке с данными. Она становится активной, о чем свидетельствует окружающая ее более жирная рамка. В нижнем правом углу рамки находится небольшой квадратик, именуемый *маркером заполнения*. Подведите указатель мыши к маркеру заполнения. Указатель превратится в черный крестик **+**. Нажмите кнопку мыши и, не отпуская ее, перемещайте указатель **+** вдоль столбца или вдоль строки, расширяя область выделения на необходимое количество ячеек. Отпустите кнопку мыши. Все ячейки выделенного диапазона заполнились данными.

1.1.3 Защита данных в ячейках

Excel предоставляет возможность полностью или выборочно запретить пользователю изменять содержимое ячеек любого заданного листа.

Чтобы защитить данные отдельных ячеек, выделите ячейки, которые не

надо защищать (если таковые есть), вызовите командой **Формат / Ячейки** диалоговое окно **Формат ячеек** и на вкладке **Защита** снимите флажок **Защищаемая ячейка** (Locked). Все ячейки, для которых этот флажок останется установленным, будут защищены при включении защиты листа.

Чтобы защитить текущий лист, выполните команду **Сервис / Защита / Защитить лист**, введите пароль и щелкните на кнопке **ОК**.

1.1.4 Формулы и функции. Решение задач


В *Excel* имеется большое количество встроенных функций, которые можно использовать в формулах. Для этого достаточно указать их имя и, как правило, аргумент или список аргументов, заключенных в круглые скобки. Аргументы функций отделяются друг от друга знаком ; (точка с запятой). В качестве аргументов могут использоваться числа, адреса или имена ячеек, диапазоны, арифметические выражения и функции.

Если в качестве аргумента функции используется *диапазон смежных ячеек*, то он задается указанием адреса верхней левой и нижней правой ячеек диапазона, разделенных двоеточием, например, А3:С5. Диапазоны несмежных ячеек перечисляются через точку с запятой, например:

=СУММ(А3:С5;А10:С10)

В формуле можно сослаться на адрес ячейки, находящейся на другом листе. При этом указывается имя листа с восклицательным знаком, а затем – ссылка на ячейку или диапазон ячеек (без пробелов), например:

=SIN(Лист2!А3)

Для вставки функции в формулу служит **Мастер функций**, вызываемый щелчком мыши на кнопке  панели инструментов или командой **Вставка / Функция**. Мастер функций дает возможность найти и вставить функцию одной из представленных категорий, узнать назначение или вызвать справку по выбранной функции.

Обычно диалоговое окно **Мастер функций** разделено на две части. Слева (сверху) расположен список категорий встроенных функций, справа (снизу) – перечень функций из выбранной категории. Наиболее часто используемые функции кратко описываются ниже:

Математические функции:

ABS(число) – возвращает модуль (абсолютную величину) числа;

COS(число) – возвращает косинус угла;

EXP(число) – экспонента заданного числа;

LN(число) – натуральный логарифм;

LOG(число; основание) – логарифм числа по указанному основанию;

SIN(число) – возвращает синус угла;

TAN(число) – возвращает тангенс угла;

КОРЕНЬ(число) – возвращает значение квадратного корня;

ОКРУГЛ(число; число_разрядов) – округляет число до указанного количества десятичных разрядов;

ПИ() – число π , округленное до 15 знаков после запятой;

ПРОИЗВЕД(число1; число2; ...) – произведение аргументов;

СЛЧИС() – случайное число, равномерно распределенное на диапазоне [0; 1); функция используется без аргументов;

СТЕПЕНЬ(число; степень) – результат возведения в степень;

СУММ(число1; число2; ...).

Статистические функции:

МАКС(число1; число2; ...) – возвращает максимальное значение из списка аргументов;

МИН(число1; число2; ...) – минимальное значение из списка аргументов;

СРЗНАЧ(число1; число2; ...) – среднее арифметическое аргументов;

СЧЁТЗ(значение1; значение2; ...) – количество значений в списке аргументов и непустых ячеек;

Логические функции:

ЕСЛИ(условие; значение_если_истина; значение_если_ложь) – проверяет условие. Если условие верно, то функция возвращает одно значение, если условие не выполняется – другое;

И(условие1; условие2; ...) – возвращает значение ИСТИНА, если истинны все аргументы;

ИЛИ(условие1; условие2; ...) – возвращает ИСТИНА, если значение ИСТИНА имеет хотя бы один аргумент. В противном случае возвращает значение ЛОЖЬ.

Дата и время:

СЕГОДНЯ() – возвращает текущую дату в формате даты;

ДЕНЬНЕД(дата_в_числовом_формате; тип) – число от 1 до 7, соответствующее номеру дня недели для текущей даты, для России тип равен 2.

Текстовые:

НАЙТИ(искомый_текст; просматриваемый_текст; нач_позиция) – возвращает позицию начала текущей строки в содержащей ее строке;

ПРОПИСН(текст) – делает все буквы в текстовой строке прописными;

СТРОЧН(текст) – делает все буквы в строке текста строчными;

СЦЕПИТЬ(текст1; текст2; ...) – объединяет несколько текстовых строк в одну;

ТЕКСТ(значение; формат) – форматирует число и преобразует его в текст.

В категории **Полный алфавитный перечень** указаны все встроенные функции *Excel*.

Пример 1. Вычислить значения функции

$$\text{fun1}(m) = \frac{\sqrt{m}}{\text{tg}^2 m + b1} \quad (1)$$

при $b1 = 1,3$, если m изменяется в промежутке $[5; 11]$ с шагом $\Delta m = 0,6$.

Решение.

1 В ячейку A1 введите текст $b1=$, в ячейку B1 – текст $m=$; в ячейку C1 – текст $\text{fun1} =$.

Примечание – В активную ячейку *Excel* можно ввести либо текст, либо число, либо формулу. Если ввести в ячейку последовательность символов $b1=1,3$, то эта запись будет восприниматься табличным процессором как текст, который, несмотря на знак $=$, не предназначен для выполнения математических операций, а имеет только информационное назначение.

2 В ячейке A2 наберите число 1,3. В *Excel* в качестве десятичного разделителя по умолчанию используется *запятая*.

3 В ячейку B2 введите начальное значение переменной m , равное 5.

4 В ячейку B3 введите значение 5,6, которое подсчитывается как начальное значение 5 плюс шаг 0,6.

5 Выполните автозаполнение ячеек в столбце B значениями переменной m .

Для этого выделите две ячейки B2 и B3, содержащие первое и второе значения этой переменной, как показано на рисунке 4.

	A	B	C
1	b1=	m=	fun1=
2	1,3	5	
3		5,6	

Рисунок 4 – Выделение двух ячеек


Наведите указатель мыши на *маркер заполнения* (правый нижний угол рамки, обрамляющей ячейки B2 и B3) так, чтобы указатель мыши принял вид крестика $+$. Нажав в этом положении левую кнопку мыши, протяните маркер заполнения вниз, не отпуская, пока на всплывающей подсказке не отобразится последнее значение переменной m , равное 11, как показано на рисунке 5. После этого отпустите левую кнопку мыши. В ячейках столбца отобразятся все значения переменной m от 5 до 11 с шагом 0,6 (рисунок 6).

	A	B	C
1	b1=	m=	fun1=
2	1,3	5	
3		5,6	
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			11
14			

Рисунок 5 – Автозаполнение ячеек в столбце

	A	B	C
1	b1=	m	fun1=
2	1,3	5	
3		5,6	
4		6,2	
5		6,8	
6		7,4	
7		8	
8		8,6	
9		9,2	
10		9,8	
11		10,4	
12		11	

Рисунок 6 – Результаты автозаполнения ячеек

В ячейке C2 наберите знак = и выражение для вычисления значения функции $fun1(m)$. Вызовите мастер вставки функции щелчком мыши на кнопке  панели инструментов или командой **Вставка / Функция**. Выберите категорию функций **Математические**, а в ней – функцию с именем **КОРЕНЬ** и щелкните на кнопке **ОК**. Появится диалоговое окно **Аргументы функции**, показанное на рисунке 7.

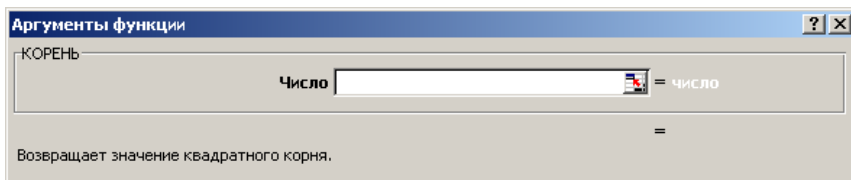
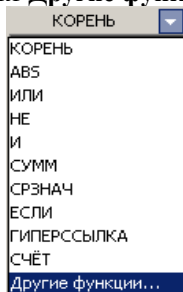


Рисунок 7 – Диалоговое окно **Аргументы функции**

По условию задачи операция извлечения квадратного корня применяется к значению переменной m . Но в качестве аргумента функции в *Excel*, как правило, используется *не имя переменной, а адрес ячейки*, в нашем примере это B2, в которой хранится ее значение. Причем для ввода необходимого адреса достаточно щелкнуть левой кнопкой мыши на соответствующей ячейке. При этом в строке формул отобразится $=КОРЕНЬ(B2)$, а ячейка B2 выделится мерцающей рамкой.

Примечание – Не следует закрывать диалоговое окно **Мастера функции**, пока формула не набрана целиком. Это позволит откорректировать любую функцию в составе формулы, перейдя на нее щелчком левой кнопкой мыши, осуществляемым на имени функции в строке формул.

Перейдите в строку формул и наберите знак деления / (можно использовать клавишу дополнительной клавиатуры), а затем открывающуюся скобку (**Shift 9**). Так как окно **Мастера функций** открыто, для возведения тангенса в квадрат обратитесь к списку в начале строки формул, откуда выберите пункт **Другие функции...**:



В категории функций **Математические** найдите функцию **СТЕПЕНЬ** и щелкните на кнопке **ОК**. В *Excel* отобразится диалоговое окно **Аргументы функции**, показанное на рисунке 8.

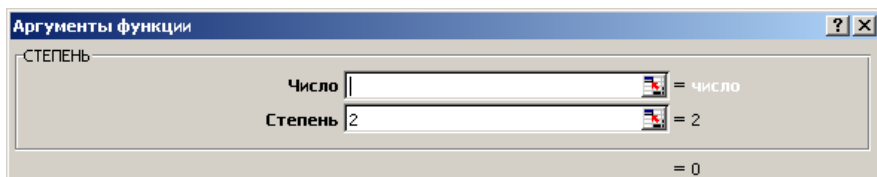


Рисунок 8 – Вставка обращения к степенной функции

В поле **Степень** напечатайте число 2 и перейдите в строку **Число**, чтобы ввести тангенс переменной *m*. В категории **Математические** окна **Мастер функций** выберите **TAN**. В строке **Число** укажите адрес B2 (рисунок 9), щелкнув для этого на ячейке B2, содержащей число 5.

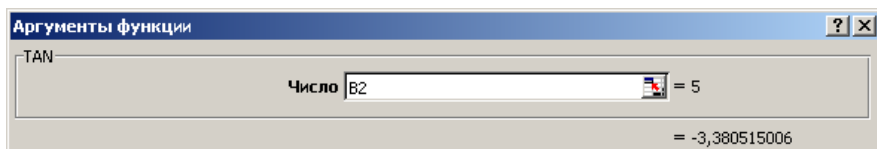


Рисунок 9 – Вставка обращения к функции тангенс

В строке формул отобразится:

$$=\text{КОРЕНЬ}(\text{B2}) / (\text{СТЕПЕНЬ}(\text{TAN}(\text{B2}) ; 2)$$

Наберите далее знак + и адрес ячейки A2, где находится значение

константы $b1$. Так как оно единственное, то его следует зафиксировать, создав абсолютную ссылку. Для отображения знаков \$ установите курсор на A2 и нажмите клавишу **F4**.

Введите закрывающую скобку (**Shift 0**) и нажмите клавишу **Enter**.
Итоговая формула должна иметь вид:

$$=КОРЕНЬ(B2) / (СТЕПЕНЬ(TAN(B2); 2) + A2)$$

7 Выполните автозаполнение столбца C значениями функции $fun1$ для соответствующих значений переменной m из столбца B. Для этого выделите ячейку C2 и наведите указатель мыши на правый нижний угол ее рамки так, чтобы указатель мыши принял вид тонкого черного крестика **+**. Протяните указатель мыши **+** вниз до последнего значения переменной m .

Фрагмент итоговой электронной таблицы в режиме отображения численных результатов расчета приведен на рисунке 10, а в режиме отображения формул – на рисунке 11.

	A	B	C
1	b1=	m=	fun1=
2	1,3	5	0,17568265
3		5,6	1,20582292
4		6,2	1,90518108
5		6,8	1,606694
6		7,4	0,49463799
7		8	0,05950064
8		8,6	1,18689598
9		9,2	2,24299467
10		9,8	2,15133038
11		10,4	0,92733696
12		11	6,4962E-05
13			

Рисунок 10 – Режим отображения чисел

	A	B	C
1	b1=	m=	fun1=
2	1,3	5	=КОРЕНЬ(B2)/(СТЕПЕНЬ(TAN(B2);2)+\$A\$2)
3		5,6	=КОРЕНЬ(B3)/(СТЕПЕНЬ(TAN(B3);2)+\$A\$2)
4		6,2	=КОРЕНЬ(B4)/(СТЕПЕНЬ(TAN(B4);2)+\$A\$2)
5		6,8	=КОРЕНЬ(B5)/(СТЕПЕНЬ(TAN(B5);2)+\$A\$2)
6		7,4	=КОРЕНЬ(B6)/(СТЕПЕНЬ(TAN(B6);2)+\$A\$2)
7		8	=КОРЕНЬ(B7)/(СТЕПЕНЬ(TAN(B7);2)+\$A\$2)
8		8,6	=КОРЕНЬ(B8)/(СТЕПЕНЬ(TAN(B8);2)+\$A\$2)
9		9,2	=КОРЕНЬ(B9)/(СТЕПЕНЬ(TAN(B9);2)+\$A\$2)
10		9,8	=КОРЕНЬ(B10)/(СТЕПЕНЬ(TAN(B10);2)+\$A\$2)
11		10,4	=КОРЕНЬ(B11)/(СТЕПЕНЬ(TAN(B11);2)+\$A\$2)
12		11	=КОРЕНЬ(B12)/(СТЕПЕНЬ(TAN(B12);2)+\$A\$2)
13			

Рисунок 11 – Режим отображения формул

Чтобы перейти в режим отображения формул в электронной таблице, вызовите командой **Сервис / Параметры** диалоговое окно **Параметры** и на вкладке **Вид** в поле **Параметры окна** установите флажок **Формулы**.

Обратите внимание, как изменяется относительная ссылка при переходе от одного значения аргумента m к другому.

Пример 2. Создать одномерный массив MS , состоящий из десяти произвольных чисел. Для каждого элемента MS вычислить

$$RES = \sqrt[3]{MS + p9} + |MS|, \quad (2)$$

где $p9 = 5,7$.

Найти сумму, произведение элементов массива RES , а также его минимальный элемент. Подсчитать количество значений RES , больших числа 5.

Решение.

1 Введите в ячейку A1 поясняющий текст $p9=$, в ячейку B1 – текст $MS=$, а в ячейку C1 – текст $RES=$.

2 В ячейку A2 введите числовое значение $p9$, а именно, 5,7.

3 Заполните ячейки B2 – B11 произвольными значениями массива MS .

4 В ячейке C2 наберите формулу
 $=СТЕПЕНЬ(B2+\$A\$2;1/3)+ABS(B2)$

Адреса ячеек лучше вводить щелчком левой кнопкой мыши на соответствующей ячейке.

5 Примените инструмент автозаполнения для диапазона ячеек C2:C11, используя формулу в ячейке C2.

6 Введите в ячейку B13 поясняющий текст $СУММА(RES)=$.

7 Перейдите в ячейку C13 и щелкните на кнопке Σ (**Автосумма**) панели инструментов **Стандартная**, чтобы вычислить сумму элементов массива RES . В строке формул и в ячейке C13 отобразится формула
 $=СУММ(C2:C12)$

Откорректируйте ее, изменив ссылку (адрес) C12 на C11. Для этого выделите диапазон ячеек C2:C11, используя указатель мыши, либо просто установите текстовый курсор на ссылке C12 в формуле, а затем щелкните на ячейке C11. Нажмите клавишу **Enter**.

8 В ячейку B14 введите вспомогательный текст $MIN(RES)=$, а в ячейку C14 знак = (равно).

Щелчком мыши на кнопке f_x панели инструментов **Стандартная** запустите **Мастер функций**, выберите категорию **Статистические**, в перечне функций выделите **МИН** и щелкните на кнопке **ОК**. В открывшемся диалоговом окне укажите диапазон поиска минимального элемента C2:C11 и щелкните на кнопке **ОК**. В ячейке C14 отобразится значение минимального элемента массива RES .

9 Вычисление произведения элементов массива RES производится аналогичным образом с помощью функции **ПРОИЗВЕД(C2:C11)** из категории **Математические**.


10 Количество элементов RES , больших числа 5, найдите посредством функции **СЧЁТЕСЛИ(диапазон; критерий)** из категории **Статистические**.

Итоговая электронная таблица в режиме отображения формул приведена на рисунке 12.

	A	B	C
1	p9=	MS=	RES=
2	5	6	=СТЕПЕНЬ(B2+\$A\$2;1/3)+ABS(B2)
3		1	=СТЕПЕНЬ(B3+\$A\$2;1/3)+ABS(B3)
4		5	=СТЕПЕНЬ(B4+\$A\$2;1/3)+ABS(B4)
5		3	=СТЕПЕНЬ(B5+\$A\$2;1/3)+ABS(B5)
6		6	=СТЕПЕНЬ(B6+\$A\$2;1/3)+ABS(B6)
7		2	=СТЕПЕНЬ(B7+\$A\$2;1/3)+ABS(B7)
8		8	=СТЕПЕНЬ(B8+\$A\$2;1/3)+ABS(B8)
9		-3	=СТЕПЕНЬ(B9+\$A\$2;1/3)+ABS(B9)
10		3,6	=СТЕПЕНЬ(B10+\$A\$2;1/3)+ABS(B10)
11		5	=СТЕПЕНЬ(B11+\$A\$2;1/3)+ABS(B11)
12			
13		СУММА(RES)	=СУММ(C2:C11)
14		МИН(RES)	=МИН(C2:C11)
15		ПРОИЗВ(RES)	=ПРОИЗВЕД(C2:C11)
16		КОЛИЧ(RES)<5	=СЧЁТЕСЛИ(C2:C11;">5")

Рисунок 12 – Фрагмент электронной таблицы в режиме отображения формул

1.1.5 Графические возможности Excel. Построение графика функции

Для построения графиков и диаграмм в Excel используется **Мастер диаграмм**, вызываемый командой **Вставка / Диаграмма** или щелчком мыши на кнопке  панели инструментов **Стандартная**.

В диалоговом окне **Мастер диаграмм** работа проводится в четыре этапа (шага).

1 На *первом шаге* выбирается **тип диаграммы**. Для построения графика функциональной зависимости следует выбрать тип **График** или **Точечная**.

2 На *втором шаге* определяется **источник данных диаграммы** (диапазон данных), добавляются или удаляются ряды данных.

3 Диалоговое окно **Мастер диаграмм: параметры диаграммы** *третьего шага* (рисунок 13) содержит шесть вкладок, на каждой из которых выбираются опции, соответствующие имени вкладки.

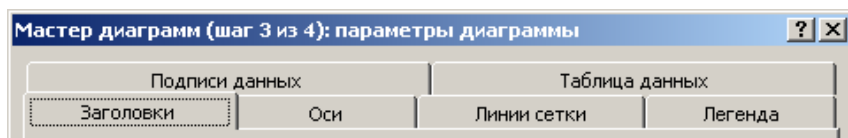


Рисунок 13 – Мастер диаграмм

4 Место расположения диаграммы – на имеющемся или на отдельном листе – определяется на последнем, *четвертом шаге* применения инструмента **Мастер диаграмм**.


Пример 3. Построить график функции

$$f(m) = \frac{\sqrt{m}}{\operatorname{tg}^2 m + b1} \quad (3)$$

при $b1 = 1,3$.

Решение.

1 Для построения графика выделите диапазон ячеек C1:C12 со значениями функции $fun1(m)$.

2 Вызовите **Мастер диаграмм** командой **Вставка / Диаграмма** или щелчком на кнопке  панели инструментов **Стандартная**.

3 Выберите *тип* диаграммы **Точечная** и *вид* диаграммы со значениями, соединенными сглаживающими линиями (первая во втором ряду) на вкладке **Стандартные**. Щелкните на кнопке **Далее** для перехода ко второму шагу **Мастера диаграмм**.

4 На вкладке **Диапазон данных** в поле **Диапазон** видна автоматически сформированная ссылка на предварительно выделенный диапазон данных для построения графика в виде =Лист1!\$C\$1:\$C\$12 и выбрана опция **Ряды** в столбцах.


5 Откройте вкладку **Ряд**, поместите текстовый курсор в пустое поле **Значения X**, показанное на рисунке 14, и щелкните на кнопке .



Рисунок 14 – Поле **Значения X** в диалоговом окне **Мастер диаграмм**

Окно **Мастер диаграмм** свернется и примет вид, изображенный на рисунке 15, предоставив возможность пользователю выделить на рабочем листе диапазон ячеек B1:B12, хранящих последовательность значений переменной m .

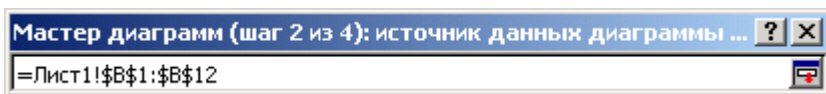


Рисунок 15 – Ссылка на значения m в свернутом окне **Мастер диаграмм**

Восстановите диалоговое окно **Мастер диаграмм**, щелкнув мышью на кнопке .

В поле **Значения X** отобразится ссылка

=Лист1!\$B\$1:\$B\$12

Щелкните на кнопке **Далее** для перехода к шагу 3.

6 В поле **Название диаграммы** наберите текст **Значения функции $fun1$** ; в поле **Ось X (категорий)** – аргумент m ; в поле **Ось Y (значений)** – $fun1$.

7 Перейдите на вкладку **Оси** и выберите параметры графика, показанные

на рисунке 16.

8 На вкладке **Линии сетки** выберите вариант **основные линии** по оси X и по оси Y.

9 Перейдите на вкладку **Легенда** и уберите флажок **Добавить легенду**.

10 На вкладке **Подписи данных** дополните подписи данных значениями аргумента и функции.

11 Щелкните на кнопке **Далее** для перехода к последнему четвертому шагу построения диаграммы. Выберите вариант **Поместить диаграмму на листе**, чтобы диаграмма отобразилась на текущем листе и щелкните на кнопке **Готово**.

Диаграмма должна принять вид, показанный на рисунке 17.

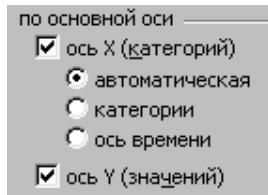


Рисунок 16 – Выбор параметров графика



Рисунок 17 – Созданный график функции $fun1(m)$

Примечания

1 Для изменения местоположения диаграммы следует установить указатель мыши в любом ее месте и, не отпуская левую кнопку мыши, тянуть диаграмму до нужного положения на листе.

2 Чтобы изменить размеры диаграммы, сначала щелкают левой клавишей мыши в области диаграммы. Затем наводят указатель мыши на маркер границы так, чтобы он принял вид двунаправленной стрелки, и тянут, не отпуская левую кнопку, пока не будут достигнуты требуемые размеры.

3 Для изменения параметров диаграммы дважды щелкают левой кнопкой мыши в том месте, которое нуждается в изменении. Можно также щелкнуть правой кнопкой мыши в области диаграммы для вызова контекстного меню и выбрать нужный пункт из списка. В результате появляется окно, в котором настраиваются желаемые опции.

1.2 Решение уравнений и систем линейных уравнений

1.2.1 Решение уравнений в Excel

Решение уравнений в Excel производится в два этапа:

- локализация корней, т. е. определение приближенного значения корня или интервала его нахождения (аналитически или графически);
- уточнение корней, которое проще выполнять, пользуясь инструментом

Подбор параметра.

Пример 4. Решить уравнение

$$\left(\frac{x}{2}\right)^3 - \ln |x| = 2. \quad (4)$$

Решение.

1 Преобразуйте уравнение к стандартной форме $f(x) = 0$:

$$\left(\frac{x}{2}\right)^3 - \ln |x| - 2 = 0. \quad (5)$$

2 С целью отделения корней создайте столбец значений переменной x , изменяющейся в диапазоне $[-5; 5]$ с шагом 0,5, и столбец соответствующих значений функции

$$f(x) = \left(\frac{x}{2}\right)^3 - \ln |x| - 2. \quad (6)$$

Для этого сначала в ячейке A1 наберите $x =$, а в ячейке B1 – $f(x) =$.


3 В ячейку A2 введите отрицательное число -5 , являющееся левой границей интервала изменения переменной x .

4 В ячейку A3 занесите число $-4,5$, которое на величину шага 0,5 больше числа из ячейки A2.

5 Выделите ячейки A2 и A3:



6 Примените инструмент автозаполнения ячеек в диапазоне от A4 до A22.

7 В ячейку B2 введите формулу, начиная ее знаком $=$. Для ввода степени вызовите мастер функций, щелкнув на кнопке  панели инструментов **Стандартная**. В категории **Математические** выберите функцию СТЕПЕНЬ. Появится окно, отображенное на рисунке 18.

8 Чтобы окно не мешало набору формулы, переместите его на свободное место рабочего листа посредством технологии *Drag-and-Drop*.

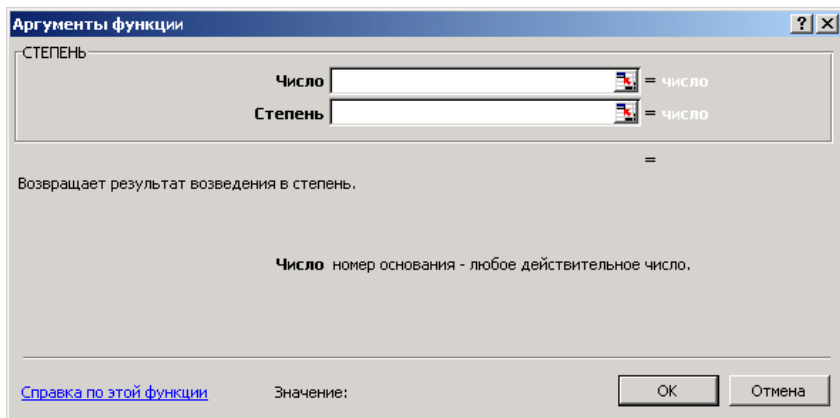


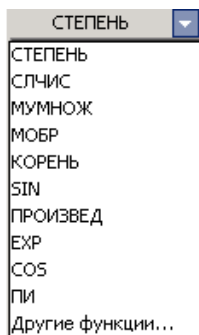
Рисунок 18 – Диалоговое окно **Аргументы функции**

В поле **Степень** введите число 3.

9 Установите курсор в поле **Число**. Для ввода аргумента $\frac{x}{2}$ щелкните на ячейке A2, наберите знак деления /, а затем число 2. Должно получиться A2/2.

10 Не закрывая окно **Аргументы функции**, щелкните в окне *Excel* в строке формул в конце выражения и наберите знак минус (-).

11 В списке функций в левом верхнем углу окна *Excel*



выберите **Другие функции...** Далее в категории **Математические** найдите и вставьте в формулу обращение к функции LN (), а затем в качестве аргумента функции LN – обращение к функции ABS (A2) .

12 Перейдите в конец формируемого выражения в строке формул, наберите знак минус, число 2 и нажмите клавишу **Enter**.

В ячейке B2 должен отобразиться результат в виде отрицательного числа -19,23444.

13 Выделите ячейку В2 и примените инструмент автозаполнения ячеек в диапазоне от В3 до В22.

Фрагменты итоговой электронной таблицы в режиме отображения значений и в режиме формул соответственно показаны на рисунке 19.

	А	В		А	В
1	x=	f(x)=	1	x=	f(x)=
2	-5	-19,23444	2	-5	=СТЕПЕНЬ(A2/2;3)-LN(ABS(A2))-2
3	-4,5	-14,8947	3	-4,5	=СТЕПЕНЬ(A3/2;3)-LN(ABS(A3))-2
4	-4	-11,38629	4	-4	=СТЕПЕНЬ(A4/2;3)-LN(ABS(A4))-2
5	-3,5	-8,612138	5	-3,5	=СТЕПЕНЬ(A5/2;3)-LN(ABS(A5))-2
6	-3	-6,473612	6	-3	=СТЕПЕНЬ(A6/2;3)-LN(ABS(A6))-2
7	-2,5	-4,869416	7	-2,5	=СТЕПЕНЬ(A7/2;3)-LN(ABS(A7))-2
8	-2	-3,693147	8	-2	=СТЕПЕНЬ(A8/2;3)-LN(ABS(A8))-2
9	-1,5	-2,82734	9	-1,5	=СТЕПЕНЬ(A9/2;3)-LN(ABS(A9))-2
10	-1	-2,125	10	-1	=СТЕПЕНЬ(A10/2;3)-LN(ABS(A10))-2
11	-0,5	-1,322478	11	-0,5	=СТЕПЕНЬ(A11/2;3)-LN(ABS(A11))-2
12	0	#ЧИСЛО!	12	0	=СТЕПЕНЬ(A12/2;3)-LN(ABS(A12))-2
13	0,5	-1,291228	13	0,5	=СТЕПЕНЬ(A13/2;3)-LN(ABS(A13))-2
14	1	-1,875	14	1	=СТЕПЕНЬ(A14/2;3)-LN(ABS(A14))-2
15	1,5	-1,98359	15	1,5	=СТЕПЕНЬ(A15/2;3)-LN(ABS(A15))-2
16	2	-1,693147	16	2	=СТЕПЕНЬ(A16/2;3)-LN(ABS(A16))-2
17	2,5	-0,963166	17	2,5	=СТЕПЕНЬ(A17/2;3)-LN(ABS(A17))-2
18	3	0,276388	18	3	=СТЕПЕНЬ(A18/2;3)-LN(ABS(A18))-2
19	3,5	2,106612	19	3,5	=СТЕПЕНЬ(A19/2;3)-LN(ABS(A19))-2
20	4	4,613706	20	4	=СТЕПЕНЬ(A20/2;3)-LN(ABS(A20))-2
21	4,5	7,886548	21	4,5	=СТЕПЕНЬ(A21/2;3)-LN(ABS(A21))-2
22	5	12,01556	22	5	=СТЕПЕНЬ(A22/2;3)-LN(ABS(A22))-2

Рисунок 19 – Результаты вычислений

Обратите внимание, что при $x \in [2,5;3]$ функция $f(x)$ меняет знак с минуса на плюс. Следовательно, корень уравнения $f(x) = 0$ находится в интервале x от 2,5 до 3.

Этап отделения корней завершен.

14 Чтобы зафиксировать найденное начальное приближение, скопируйте содержимое ячеек А18 и В18 и вставьте, например, в ячейки А25 и В25 соответственно. Можно также поместить в ячейку А25 среднее арифметическое чисел, содержащихся в ячейках А18 и А19.

15 Установите точность дальнейших вычислений равной 0,0001. Для этого командой **Сервис / Параметры** вызовите диалоговое окно **Параметры**, на его вкладке **Вычисления** в поле **Относительная погрешность** введите число 0,0001 и щелкните на кнопке **Enter**.

16 Активизируйте ячейку В25, в которой находится значение функции $f(x)$ при $x = 3$, близком к корню уравнения.

17 Выполните команду **Сервис / Подбор параметра**. В отобразившемся диалоговом окне заполните поля, как показано на рисунке 20.

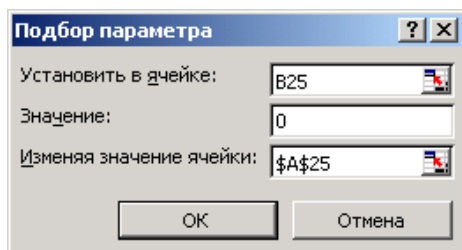


Рисунок 20 – Результаты вычислений

Значение переменной x , которое находится в ячейке $\$A\25 , будет варьироваться таким образом, чтобы соответствующее ему значение функции (в ячейке B25) стало равным 0, т. к. правая часть уравнения (5), корень которого определяется, равна нулю.

18 После щелчка мышью на кнопке **ОК** появится информационное окно, показанное на рисунке 21.

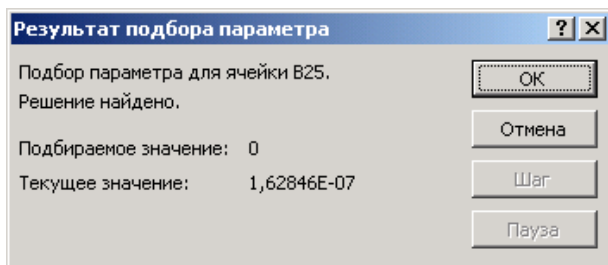


Рисунок 21 – Результаты подбора параметра

Видно, что решение найдено с заданной точностью (погрешность – менее 10^{-6}). Значение корня уравнения $x = 2,9057$ отобразилось в ячейке A25, а соответствующее ему значение функции – в ячейке B25 (рисунок 22), т. е. в тех ячейках, в которые ранее были введены начальное приближение для корня и значение функции.

24			
25	2,905713	1,62846E-07	
26			

Рисунок 22 – Результаты определения корня уравнения

1.2.2 Решение систем линейных уравнений в Excel

В Excel имеется ряд функций для обработки двумерных массивов. В частности, категория **Математические** содержит функции:

- МОБР(массив) – возвращает обратную матрицу для выбранного диапазона ячеек, хранящего значения элементов квадратной матрицы;
- МОПРЕД(массив) – возвращает определитель матрицы (матрица хранится в массиве);
- МУМНОЖ(массив1; массив2) – возвращает произведение матриц;
- ТРАНСП(массив) – преобразует вертикальный диапазон ячеек в горизонтальный, или наоборот;
- ЧИСЛСТОЛБ(массив) – возвращает число столбцов в массиве;
- ЧСТРОК(массив) – возвращает число строк в массиве.

Перечисленные функции позволяют реализовать решение системы линейных алгебраических уравнений методом обратной матрицы.

Пример 5. Решить систему уравнений

$$\begin{cases} 3,4x - 5,5y + 1,6z = -4,5; \\ 7,9x + 2,1y - 5z = 0,3; \\ 12,3x - 4,6y - 7,8z = -2. \end{cases} \quad (7)$$

Решение.

1 В ячейке A2 наберите текст A =, а в ячейки диапазона B1:D3 поместите значения коэффициентов при неизвестных системы уравнений (7).

2 В ячейку F2 введите текст b =, а ячейки диапазона G1:G3 заполните значениями правых частей уравнений системы.

Результаты ввода исходных данных приведены на рисунке 23.

	A	B	C	D	E	F	G
1		3,4	-5,5	1,6			-4,5
2	A=	7,9	2,1	-5		b=	0,3
3		12,3	-4,6	-7,8			-2
4							

Рисунок 23 – Исходные данные системы уравнений

3 В ячейке A5 наберите текст |A| = и перейдите в ячейку B5. Чтобы вычислить определитель матрицы A, вызовите **Мастер функций** и в категории **Математические** щелкните на имени функции МОПРЕД, которая возвращает величину определителя матрицы. Откроется диалоговое окно **Аргументы функции** для функции МОПРЕД. Выделите на рабочем листе диапазон ячеек B1:D3. При этом в диалоговом окне **Аргументы**

функции, показанном на рисунке 24, в поле **Массив** появится ссылка на диапазон ячеек B1:D3, содержимое первых ячеек данного диапазона и равное -234,024 значение определителя матрицы.

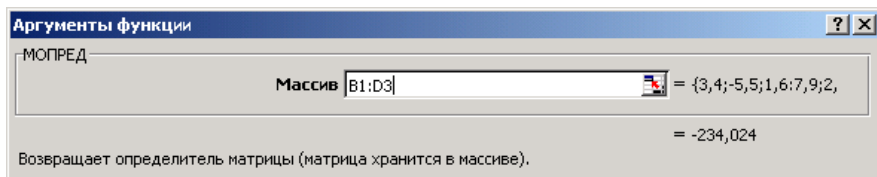


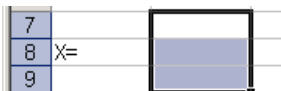
Рисунок 24 – Результаты вычисления определителя матрицы

Щелкните на кнопке **ОК**. На рабочем листе в ячейке B5 отобразится величина определителя матрицы A: $5 \quad |A| = -234,024$.

4 Так как определитель матрицы отличен от нуля, то система линейных уравнений имеет единственное решение, определяемое по формуле

$$X = A^{-1}b. \quad (8)$$

5 В ячейке A8 наберите текст $X =$ и выделите диапазон ячеек B7:B9, предназначенный для отображения найденного решения:



6 Поместите курсор в строку формул, введите знак = и вызовите **Мастер функций**. Щелкните на имени функции **МУМНОЖ**, которая возвращает результат умножения матриц. Откроется диалоговое окно **Аргументы функции** для функции **МУМНОЖ**. В формулу (8) входит обратная матрица. Для ее нахождения установите курсор в поле **Массив1** и наберите строку **МОБР (B1:D3)**, а в поле **Массив2** введите ссылку на диапазон ячеек **G1:G3**, определяющий значения столбца свободных слагаемых b . Диалоговое окно **Аргументы функции** для функции **МУМНОЖ** примет вид, показанный на рисунке 25.

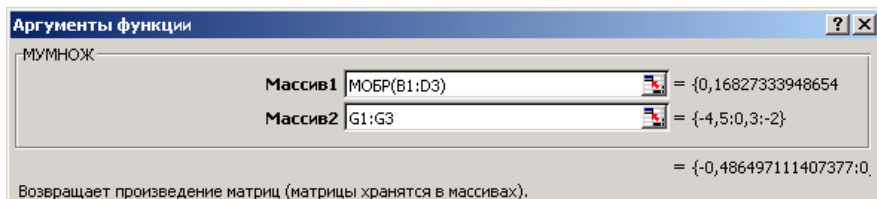


Рисунок 25 – Результаты вычислений

7 Завершите ввод формулы не традиционным щелчком на кнопке **OK**, а комбинацией клавиш **Ctrl Shift Enter**. Нажимать их следует последовательно и не отпускать 1–2 секунды, пока не зафиксируется одновременное нажатие всех трех клавиш. При этом формула в строке формул будет заключена в фигурные скобки, а диапазон B7:B9 заполнится числами, являющимися решением системы уравнений. Соответствующий фрагмент электронной таблицы приведен на рисунке 26.

	A	B	C	D	E	F	G
1		3,4	-5,5	1,6			-4,5
2	A=	7,9	2,1	-5		b=	0,3
3		12,3	-4,6	-7,8			-2
4							
5	A =	-234,024					
6							
7		-0,4865					
8	X=	0,31484					
9		-0,69643					

Рисунок 26 – Результаты вычислений

8 Выполните умножение матрицы коэффициентов при неизвестных системы A на столбец со значениями найденного решения X . Если система уравнений решена правильно, то в результате умножения должен получиться столбец ячеек, числа в которых отличаются от значений вектора b на величину погрешности расчета или совпадают с этими значениями.

Выделите диапазон ячеек D7:D9, предназначенный для отображения результатов умножения A на X , введите в строку формул последовательность символов `=МУМНОЖ(B1:D3;B7:B9)` и нажмите комбинацию клавиш **Ctrl Shift Enter**.

Введенная формула преобразуется к виду

$$\{=\text{МУМНОЖ}(B1:D3;B7:B9)\},$$

а на рабочем листе появится результат проверки решения системы уравнений: числа в ячейках диапазона D7:D9, как видно из рисунка 27, совпадают с элементами вектора b .

7		-0,4865		-4,5
8	X=	0,31484	Проверка	0,3
9		-0,69643		-2

Рисунок 27 – Результат проверки

Следовательно, система уравнений решена правильно.


2 Порядок выполнения лабораторной работы

1 Запустите *Microsoft Excel*: **Start / All Programs / Microsoft Excel** (*Пуск / Все программы / Microsoft Excel*).

2 Сохраните рабочую книгу в личной папке на диске **z:** под именем **fi05**.

3 Добавьте в рабочую книгу три новых листа: один лист командой **Вставка / Лист**, а два других командой **Добавить** из контекстного меню ярлыка листа.

4 Переименуйте листы рабочей книги следующим образом: Задание 1, Формулы 1, Задание 2, Формулы 2, Задание 3, Задание 4. Для переименования дважды щелкните левой кнопкой мыши на ярлыке листа, введите новое название и нажмите клавишу **Enter**.

5 Перейдите на лист **Задание 1** и выполните объединение пяти первых ячеек первой строки (выделите их и щелкните на кнопке  – **Объединить и поместить в центре** панели инструментов).

6 Введите в объединенную ячейку текст

Лабораторная работа 5
Табличный процессор *Excel*,

используя для перехода на новую строку в ячейке комбинацию клавиш **Alt Enter**, а для выравнивания – кнопки панели инструментов или окно форматирования ячеек (**Формат / Ячейки**), вкладка **Выравнивание**.

7 Во второй строке электронной таблицы объедините пять первых ячеек, а затем введите фамилию, имя, отчество, учебный шифр и номер варианта. Отформатируйте введенный текст в диалоговом окне **Формат / Ячейки** на вкладках **Выравнивание** и **Шрифт**.

8 В ячейку **C3** введите текст **Задание 1**.

9 В первые четыре ячейки четвертой строки введите последовательно: k_1 , k_2 , **Переменная**, **Функция**, создавая тем самым шапку таблицы.

10 Выполните задание 1 в соответствии с вариантом таблицы 1.

Задание 1. Ввести значения констант k_1 , k_2 в ячейки **A5** и **B5** соответственно. Заполнить значениями переменной ячейки столбца **C** (начиная с ячейки **C5**) так, чтобы ее изменение происходило с заданным шагом в указанном интервале.

В ячейку **D5** ввести заданную функцию, обращаясь к значениям констант посредством абсолютных ссылок, а к значению переменной с помощью относительной ссылки. Заполнить столбец **D** значениями функции, соответствующими значениям переменной.

Скопировать решение задачи на лист Формулы 1 и представить его в режиме отображения формул.

Вернуться на лист Задание 1, построить график функции с помощью инструмента **Мастер диаграмм** и поместить график на текущий лист. Произвести форматирование графика в соответствии с вариантом таблицы 2.

Таблица 1 – Варианты задания 1

Вариант	k1	k2	Переменная	Шаг изменения	Функция
1	0,1	3	$z \in [-1; 6]$	$\Delta z = 0,5$	$y(z) = \frac{\sin^3 2z - k1}{\sqrt{1/k2 + e^z}}$
2	0,2	0,5	$t \in [2; 6]$	$\Delta t = 0,25$	$f(t) = \frac{\sqrt[5]{3t} - e^{k2}}{2 \operatorname{ctg}(1+k1)}$
3	0,8	2	$y \in [-5; 5]$	$\Delta y = 0,25$	$g(y) = k1 \cdot \arccos^3 \frac{y}{25} - k2$
4	1	3,5	$a \in [-1; 6]$	$\Delta a = 0,3$	$z(a) = \frac{\ln \operatorname{ctg} a^2 - k1 }{k2 + a}$
5	0,2	1	$b \in [-3; 3]$	$\Delta b = 0,5$	$y(b) = k1 + \lg \cos b^3 - k2 $
6	4,5	2	$c \in [-3; 5]$	$\Delta c = 0,4$	$f(c) = \frac{\sqrt[5]{3k1 - c}}{2 \operatorname{tg}^3(c+k2)}$
7	1,3	5	$x \in [-1; 3]$	$\Delta x = 0,4$	$f(x) = \frac{\sqrt[3]{\cos^2 3x \cdot k1}}{3k2 \operatorname{ctg} x }$
8	2,2	7	$y \in [1; 7]$	$\Delta y = 0,5$	$g(y) = \frac{5y^3}{k1} - k2 \cdot \operatorname{arctg} \frac{y}{2}$
9	3	6,3	$z \in [-1; 6]$	$\Delta z = 0,5$	$y(z) = \frac{\sqrt{k1+z} - k2}{k2 \cdot \arccos 0,1z}$
10	12	0,1	$t \in [3; 8]$	$\Delta t = 0,25$	$g(t) = t^3 - k1 \cdot \arcsin \frac{\sqrt{t+k2}}{30}$
11	1,5	-7	$x \in [1,5; 6]$	$\Delta x = 0,3$	$z(x) = \frac{k1}{e^x} - \log_2^3 x - k2$
12	0,3	-5	$x \in [-1; 7]$	$\Delta x = 0,4$	$f(x) = k1 \cdot e^{2 x } + k2 \cdot \arcsin \frac{x}{15}$



Таблица 2 – Форматирование графика

Вариант	Тип линии	Цвет линии	Тип маркера	Цвет маркера	Цвет области построения	Размер шрифта
1		Синий	✱	Черный	Голубой	8
2		Красный	•	Коричневый	Желтый	9
3		Зеленый	◆	Белый	Оливковый	10
4		Фиолетовый	■	Синий	Белый	11
5		Черный	+	Красный	Розовый	8
6		Белый	▲	Желтый	Черный	9
7		Желтый	✱	Красный	Коричневый	10
8		Черный	•	Синий	Желтый	11
9		Зеленый	◆	Черный	Серый	8
10		Коричневый	■	Красный	Желтый	9
11		Оранжевый	+	Зеленый	Кремовый	10
12		Синий	▲	Фиолетовый	Голубой	11

11 Перейдите на лист Задание 2. Объедините пять ячеек первой строки и введите туда текст `Обработка массивов`.

12 В ячейке `A2` наберите текст `A =`, в ячейке `B2` – `k =`, в ячейке `C2` – `B =`.

13 Выполните задание 2 в соответствии с вариантом таблицы 3.

Задание 2. Ввести массив A , состоящий из десяти произвольных чисел (ячейки `A3:A12`) и значение k (ячейка `B3`). Для каждого элемента A вычислить значение элемента массива M (ячейки `C3:C12`). Вычислить сумму и произведение элементов массива M . Найти максимальное и минимальное значения массива M . Подсчитать количество отрицательных элементов массива M .

Скопировать решение задачи на лист `Формулы 2` и представить его в режиме отображения формул.

Таблица 3 – Варианты задания 2

Вариант	k	Массив M	Вариант	k	Массив M
1	$k = 5$	$M = 2\sin(A+k) - \sqrt[5]{A}$	7	$k = 4$	$M = 2\operatorname{tg}(A+k) + A $
2	$k = 2$	$M = A^3 - \ln(A+k)$	8	$k = 6$	$M = \cos^2(A) - \frac{A+k}{5}$
3	$k = 2$	$M = \cos(A) - \frac{A}{k}$	9	$k = 4$	$M = \ln^2 A-k - \frac{k}{A}$
4	$k = 5$	$M = e^A - \frac{kA}{\cos(A)}$	10	$k = 2$	$M = \sin^3\left(\frac{A}{k}\right) - \cos(A)$
5	$k = 4$	$M = \sin(kA) - \frac{\sqrt[3]{A}}{k}$	11	$k = 8$	$M = 2\operatorname{tg}(A) + A-k $
6	$k = 3$	$M = \cos A^3 - \sin^2(A-k)$	12	$k = 6$	$M = A^2 - \operatorname{tg}(A+k)$

14 Перейдите на лист Задание 3. Объедините пять ячеек первой строки и введите в объединенную ячейку текст Решение уравнений.

15 В ячейке A2 наберите $x =$, а в ячейке B2 – $f(x)$.

16 Выполните задание 3 в соответствии с вариантом таблицы 4.

Задание 3. Решить уравнение, отделив его корни аналитически и графически и уточнив один из них с точностью 0,0001, используя инструмент **Подбор параметра**. Проверить правильность решения уравнения.

Таблица 4 – Уравнения

Вариант	Уравнение	Вариант	Уравнение
1	$\ln^2(x+1) + 3x = 1$	7	$\cos^3 x + 10 = x^2$
2	$2x^3 - \frac{9}{2}x^2 - x + 2 = 0$	8	$\frac{x^4}{3} + 3x^3 - 4x^2 - 2x + 1 = 0$
3	$\cos^2 x + 5 = 7 x $	9	$x^2 + \frac{5}{\sin x + x} = \frac{x}{3} + 5$
4	$x^4 - 5x^2 - 8x + 1 = 0$	10	$\frac{e^x}{3} + 5 = 8x^2$
5	$3x^4 = \sin^3 x + 5$	11	$\sin^3 2x - 3x = 4$

6	$1 - 2x - 3x^2 - 3x^3 - 4x^4 = 0$	12	$\ln(x + 5,2) + 3x = 2$
---	-----------------------------------	----	-------------------------

17 Перейдите на лист Задание 4. Объедините пять первых ячеек первой строки и введите в объединенную ячейку текст

Решение системы линейных уравнений

18 Выполните задание 4 в соответствии с вариантом таблицы 5.

Задание 4. Найти решение системы линейных уравнений: задать матрицу коэффициентов при неизвестных системы и столбец свободных членов. Вычислить определитель матрицы коэффициентов при неизвестных и решить систему уравнений методом обратной матрицы. Сделать проверку.

Таблица 5 – Системы линейных уравнений

Вариант	Система линейных уравнений	Вариант	Система линейных уравнений
1	$\begin{cases} 35,9x - 9,2y + 3z = 2,7; \\ 4,2x - 4,1y - 5,5z = 3,4; \\ 5,2x + 4,4y + 1,3z = 16,2 \end{cases}$	7	$\begin{cases} 2,8x_1 + 3,6x_2 - 5,3x_3 = -4,7; \\ -6,9x_1 + 0,8x_2 - 6,3x_3 = 10,8; \\ 7,3x_1 + 5,8x_2 - 4,4x_3 = -8,9 \end{cases}$
2	$\begin{cases} 8,9a - 4,7b + 3,2c = -7,3; \\ 0,3a - 2,2b - 8,7c = 3,9; \\ 8,4a + 37,1b + 2,4c = -3,5 \end{cases}$	8	$\begin{cases} 1,2y_1 + 5,9y_2 - 3,3y_3 = 6,7; \\ 9,4y_1 - 5,2y_2 - 2,2y_3 = -0,3; \\ 13,5y_1 - 3,8y_2 - 12,4y_3 = 3,9 \end{cases}$
3	$\begin{cases} 3,5a - 4,7b + 15,2c = -6,6; \\ -0,5a + 5,2b - 5,8c = 5,9; \\ -6,4a + 17,2b - 4,4c = -23,7 \end{cases}$	9	$\begin{cases} 9,4x_1 - 2,9x_2 + 0,3x_3 = -4,2; \\ -3,2x_1 + 7,5x_2 - 5,3x_3 = -2,7; \\ 5,2x_1 + 5,8x_2 - 6,4x_3 = -18,9 \end{cases}$
4	$\begin{cases} 5,8a - 14,2b + 7,3c = -7,9; \\ 7,6a - 5,7b + c = 5,3; \\ 8,7a + 12,8b - 4,5c = -3,7 \end{cases}$	10	$\begin{cases} 12,2x + 34,4y + 32z = 72,7; \\ 5,2x - 44,1y - 12,45z = 3,9; \\ 76,1x + 2,3y + 54,4z = -32,3 \end{cases}$
5	$\begin{cases} 12,3x + 14,8y - 11,3z = -0,9; \\ 4,9x - 9,4z = 13,3; \\ 1,2x - 7,9y + 6,9z = -25,4 \end{cases}$	11	$\begin{cases} 2,2x - 4,4y + 3z = -2,7; \\ 6,2x - 4,1y + 12,5z = 13,2; \\ -6,1x + 5,3y - 4,9z = -3,3 \end{cases}$
6	$\begin{cases} 5,7x + 3,7y - 8z = -6,1 \\ 1,7x - 6,6y + 5,5z = 6,9 \\ 9,2x - 8,4y + 7,3z = 0,4 \end{cases}$	12	$\begin{cases} 8,5x_1 - 4,2x_2 + 10,3x_3 = 11,6; \\ 7,3x_1 + 13,6x_2 - 2,3x_3 = 9,5; \\ -7,4x_1 + 9,9x_2 - 2,8x_3 = -4,7 \end{cases}$

- 19 Сохраните рабочую книгу **fiо5** в личной папке на диске **z:**.
- 20 Завершите работу с *Excel*.

Контрольные вопросы

- 1 Понятие и сфера применения электронных таблиц.
- 2 Как настроить внешний вид окна и панели инструментов *Excel*?
- 3 Какие элементы интерфейса приложения *Excel* отличают его от других приложений, входящих в *Microsoft Office*?
- 4 Рабочая книга *Excel*. Настройка основных свойств.
- 5 Операции с листами рабочей книги.
- 6 Ячейка. Редактирование данных в ячейке электронной таблицы.
- 7 Данные в ячейках.
- 8 Автозаполнение ячеек.
- 9 Ввод формулы в ячейку. Копирование формул.
- 10 Ссылка на ячейку. Относительные, абсолютные, частично абсолютные ссылки.
- 11 Наиболее часто встречающиеся ошибки в формулах.
- 12 Форматирование ячеек. Форматирование электронной таблицы.
- 13 Использование мастера функций.
- 14 Основные категории встроенных функций в *Excel*.
- 15 Основные математические функции.
- 16 Основные статистические и логические функции.
- 17 Графические возможности *Excel*. Мастер диаграмм.
- 18 Форматирование графиков и диаграмм.
- 19 Решение уравнений в *Excel*. Инструмент **Подбор параметра**.
- 20 Встроенные функции для обработки массивов (матриц).
- 21 Решение систем линейных уравнений методом обратной матрицы.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1 **Борисенко, М. В.** Основы информационных технологий : пособие / М. В. Борисенко. – Гомель : БелГУТ, 2005. – 180 с.

2 **Голдобина, Т. А.** Mathcad для экономистов : практикум / Т. А. Голдобина. – Гомель : БелГУТ, 2007. – 107 с.

3 **Гораев, О. П.** Операционная система WINDOWS. Операционные оболочки : практикум по компьютерным технологиям / О. П. Гораев, Ю. П. Лыч. – Гомель : БелГУТ, 2003. – 68 с.

4 **Гораев, О. П.** Табличный процессор MS EXCEL : практикум по компьютерным технологиям / О. П. Гораев, Ю. П. Лыч. – Гомель : БелГУТ, 2003. – 67 с.

5 **Гораев, О. П.** Текстовый процессор MS WORD : практикум по компьютерным технологиям / О. П. Гораев, Ю. П. Лыч. – Гомель : БелГУТ, 2003. – 68 с.

6 **Дьяконов, В. П.** Энциклопедия Mathcad 2001i и Mathcad 11 / В. П. Дьяконов. – М. : СОЛОН-Пресс, 2004. – 832 с.

7 **Информатика. Базовый курс** : учеб. пособие для студ. высш. техн. учеб. заведений / С. В. Симонович [и др.] ; под общ. ред. С. В. Симоновича. – СПб. : Питер, 2003. – 640 с.

8 **Информатика. Программирование на языке Паскаль** : практикум по лабораторным работам. Ч. 1 / А. П. Кейзер [и др.] ; под общ. ред. Ю. А. Пшеничнова. – Гомель : БелГУТ, 2005. – 46 с.

9 **Кирьянов, Д. В.** Самоучитель Mathcad 13 / Д. В. Кирьянов. – СПб. : БХВ-Петербург, 2006. – 560 с.

10 **Литвинович, Т. Н.** Информатика : пособие по выполнению контрольной работы / Т. Н. Литвинович. – Гомель : БелГУТ, 2008. – 89 с.

11 **Лыч, Ю. П.** Электронные таблицы : учеб. пособие / Ю. П. Лыч. – Гомель : БелГУТ, 2000. – 123 с.

12 **Макаров, Е. Г.** Инженерные расчеты в Mathcad. Учебный курс / Е. Г. Макаров. – СПб. : Питер, 2005. – 448 с.

13 **Очков, В. Ф.** Физические и экономические величины в Mathcad и Maple / В. Ф. Очков. – М. : Финансы и статистика, 2002. – 192 с.

14 **Пшеничнов, Ю. А.** Информатика : практикум / Ю. А. Пшеничнов. – Гомель : БелГУТ, 2008. – 343 с.

15 **Рогачёва, Н. А.** Информатика. Язык программирования Паскаль: пособие для студ. дневной формы обучения техн. спец. Ч. II / Н. А. Рогачёва, М. В. Борисенко, Т. Н. Литвинович. – Гомель : БелГУТ, 2004. – 39 с.

16 **Рогачёва, Н. А.** Информатика. Язык программирования Паскаль : пособие для студ. дневной формы обучения техн. спец. Ч. III / Н. А. Рогачёва, М. В. Борисенко, Т. Н. Литвинович. – Гомель : БелГУТ, 2004. – 30 с.

17 **Салманов, О. Н.** Математическая экономика с применением Mathcad и Excel / О. Н. Салманов. – СПб. : БХВ-Петербург, 2003. – 464 с.

18 **Фаронов, В. В.** Турбо Паскаль 7.0. Начальный курс : учеб. пособие / В. В. Фаронов. – 7-е изд., перераб. – М. : Нолидж, 2001. – 576 с.

19 **Фигурнов, В. Э.** IBM PC для пользователей / В. Э. Фигурнов. – 7-е изд. – М. : ИНФРА, 2001. – 638 с.

20 **Черняк, А. А.** Высшая математика на базе Mathcad. Общий курс / А. А. Черняк, Ж. А. Черняк, Ю. А. Доманова. – СПб. : БХВ-Петербург, 2004. – 608 с.

21 **Шушкевич, Г. Ч.** Введение в Mathcad 2000 : учеб. пособие / Г. Ч. Шушкевич, С. В. Шушкевич. – Гродно : ГрГУ, 2001. – 140 с.

ПРИЛОЖЕНИЕ А

(обязательное)

Пример оформления отчета по лабораторной работе

Выполнил:
студент группы *ЗСа-11*
учебный шифр *08-345-3С*
Сергеев В. Е.
Вариант *1*

Тема

Цель: { кратко формулируются цели и задачи лабораторной работы, перечисляются получаемые в результате её выполнения умения и навыки }.

Краткие теоретические сведения.

{ Кратко, не более 1–2 страниц, излагаются основные факты, использованные при выполнении работы }

Порядок выполнения работы.

{ Излагается план выполнения лабораторной работы. При этом перечисляются задания (три и более), входящие в ее состав. Если лабораторная работа состоит из одного – двух заданий, то приводится алгоритм выполнения каждого из них }.

Индивидуальное задание.

{ Формулируется индивидуальное задание и прилагается распечатка его выполнения. }

Контрольные вопросы.

{ Даются краткие ответы на вопросы, сформулированные преподавателем по лабораторной работе }

Выводы по работе: *{ Перечисляются полученные в ходе выполнения лабораторной работы знания, умения и навыки }.*

Дата выполнения работы

Подпись студента

ПРИЛОЖЕНИЕ Б

(обязательное)

Содержание учебного материала по дисциплине «Информатика»

1 Основные понятия информатики и информационных технологий.

1.1 Основные понятия информатики.

Информационное общество. Информационная система и ее основные составляющие: пользователи (персонал, *liveware*), аппаратное обеспечение (*hardware*), программное обеспечение (*software*), данные и документация. Данные, методы обработки данных, информация. Кодирование данных двоичным кодом. Бит. Байт. Измерение количества информации. Кодирование целых и действительных чисел. Кодирование текста, кодовая таблица *ASCII*, *Unicode*. Представление аналоговой графической и звуковой информации в дискретной форме.

1.2 Основные устройства компьютера, их функции и взаимосвязь. Операционные системы.

Назначение операционных систем, их основные функции. Операционные системы с командным интерфейсом. Файловые системы. Файлы. Каталоги. Путь (маршрут). Полное имя файла. Операционные системы с графическим интерфейсом. Основные понятия графического интерфейса: рабочий стол, окно, меню, пиктограмма, понятие о буфере обмена. Управление ОС: клавиатура, манипулятор “мышь”, указатель мыши и текстовый курсор. Работа с файлами и папками (каталогами). Понятие о файловых менеджерах. Запуск приложений, работа с приложениями, завершение приложений. *Web*-браузеры.

1.3 Сети персональных компьютеров.

Технологическое обеспечение сетей. Программное обеспечение сетей. Основные понятия, определения и классификация компьютерных сетей. Университетская компьютерная сеть.

2 Алгоритмизация и программирование.

2.1 Понятие алгоритма. Алгоритмический язык *Pascal*.

Основные принципы разработки алгоритмов. Блок схемы. Базовые алгоритмы. Этапы решения задачи на компьютере. Алгоритмический язык *Pascal*. Алфавит и словарь языка. Константы и переменные. Типы данных. Стандартные скалярные типы данных. Структурированные типы данных.

2.2 Выражения, операнды и операции в языке *Pascal*. Код программы.

Арифметические выражения и операции. Логические выражения и операции. Структура программы. Пример простейшей программы. Среда программирования *Free Pascal*, ее интерфейс.

2.3 Простые операторы.

Операторы присваивания и вызова процедуры. Процедуры ввода-вывода.

2.4 Структурные операторы языка *Pascal*.

Оператор *if..then..else*. Оператор *case*. Операторы цикла. Оператор цикла *for..to..do*. Оператор цикла *while..do*. Оператор цикла *repeat..until*.

2.5 Структурированные типы данных. Массив.

Одномерные массивы, их описание. Ввод-вывод элементов массивов. Двумерные массивы. Действия над массивами и элементами массивов.

2.6 Структурированные типы данных. Строки и записи.

Строковые константы и переменные. Функции обработки строк. Записи. Оператор With.

2.7 Функции и процедуры.

Встроенные функции и процедуры. Процедуры и функции пользователя. Формальные и фактические параметры. Файлы. Описание файловой переменной. Стандартные процедуры и функции обработки файлов.

3 Аппаратное и программное обеспечение компьютера.

3.1 Персональный компьютер.

Аппаратное обеспечение компьютера (*hardware*). Процессор, его назначение и характеристики. Системная плата. Чипсет. Структура памяти. RAM. ROM. CMOS-память. Кэш-память. Жесткий диск. Флэш-память. Видеосистема компьютера. Мониторы. Устройства ввода-вывода. Классификация компьютеров и их основные характеристики.

3.2 Программное обеспечение персонального компьютера (*software*).

Системное и прикладное программное обеспечение. Классификация прикладных программ. Среды программирования.

4 Компьютерное моделирование.

4.1 Компьютерное моделирование инженерных задач.

Численные методы. Пакеты прикладных программ. Системы автоматизации проектирования научно-технических задач, их интерфейс. Базовые операции и вычислительные конструкции, форматы результатов вычислений. Ранжированные переменные. Построение графиков.

4.2 Автоматизация вычислений и аналитических преобразований.

Вычисления сумм, произведений, производных и интегралов. Матричная алгебра. Решение систем уравнений и неравенств. Редактирование, форматирование и печать документов. Символьные (аналитические) преобразования.

5 Офисные программы.

5.1 Текстовые редакторы и процессоры, назначение и основные возможности.

Текст как информационный объект. Основы редактирования документов. Структуризация документа. Понятие о стилях оформления. Шаблоны документов. Формулы, таблицы и рисунки. Проверка правописания и грамматики, словари. Применение полей и форм. Ссылки, заголовки, оглавления. Создание макросов. Печать документов. Создание презентаций.

5.2 Табличные процессоры, назначение и основные возможности.

Ввод чисел, формул и текстов. Основные элементы табличного процессора: ячейка, столбец, строка, рабочий лист. Адресация ячеек. Типы данных. Математические функции. Построение диаграмм. Средства автоматизации вычислений: макросы. Печать рабочих листов.

5.3 Системы управления базами данных.

Структура базы данных, свойства полей базы данных, типы данных. Проектирование баз данных: режимы работы с базами данных, объекты базы данных. Таблицы. Структура таблицы, данные таблицы. Режим конструктора. Выбор для поля таблицы типа данного, размер поля, поле типа "счётчик". Поддержание целостности данных, создание межтабличных связей. Формы.

Создание форм с помощью мастера, структура формы, элементы управления формы, дизайн формы. Запросы. Использование условий отбора. Отчеты.

6 Сетевые технологии.

6.1 *Internet*: основные понятия и определения.

Средства просмотра и поиска информации, служба *World Wide Web (WWW)*, электронная почта, списки рассылки, телеконференции, служба имен доменов (*DNS*), служба передачи файлов (*FTP*), *ICQ*. Подключение к *Internet*, провайдеры. Создание *web*-сайтов. Редакторы *HTML (Web-редакторы)*, средства просмотра *web*-сайтов.

7 Информационная безопасность.

7.1 Защита информации.

Основные понятия. Угроза. Источники угрозы. Атака. Угрозы доступности. Вредоносное программное обеспечение. Угрозы целостности и конфиденциальности. Компьютерные вирусы и антивирусные программные средства. Защита от спама. Брандмауэр. Защита домашнего компьютера.

8 Взгляд в прошлое и будущее.

8.1 История, этапы и перспективы развития средств информационных технологий.

ПРИЛОЖЕНИЕ В

(обязательное)

Некоторые процедуры-функции на языке *Pascal*²

Функция арксинус $\arcsin(x)$

```
Function arcsin(x: real): real;  
begin  
if abs(x)=1 then arcsin := 0  
else arcsin := arctan(x/sqrt(1-sqr(x)))  
end;
```

Функция арккосинус $\arccos(x)$

```
Function arccos(x: real): real;  
begin  
if x = 0 then arccos := pi/2  
else arccos := arctan(sqrt(1-sqr(x))/x)+pi*byte(x<0)  
end;
```

² <http://vbrus.narod.ru/PascalInfo3.htm>

ПРИЛОЖЕНИЕ Г

(обязательное)

Выполнение лабораторной работы в среде *Turbo Pascal*

Краткое описание среды *Turbo Pascal*

Общие принципы работы, составление, отладка и выполнение программ в среде *Turbo Pascal 7.0* (как и в средах *Borland Pascal* для *MS-DOS* и *Windows*, *PascalABC* и др.) в целом подобны работе в среде *Free Pascal*.

Вход в среду *Turbo Pascal* можно выполнить из файлового менеджера *Total Commander* следующим образом:

1 Нажмите комбинацию клавиш **Alt F1 (Alt F2)**, выберите локальный диск D и нажмите клавишу **Enter**.

2 На диске D выберите каталог TP и нажмите клавишу **Enter**.

3 В каталоге TP выберите подкаталог bin и нажмите клавишу **Enter**.

4 В каталоге D:\TP\bin найдите и запустите файл turbo.exe.

Для работы в среде *Turbo Pascal* используются следующие основные клавиши и комбинации клавиш:

F2 – сохранение редактируемого текста программы;

F9 – компиляция программы;

F10 – активизация пунктов главного меню;

Ctrl F9 – прогон программы: компиляция программы и ее выполнение;

Alt F5 – отображение окна вывода результатов работы программы;

Alt X – закрытие окна приложения (среды программирования).

Набор и выполнение программ в среде *Turbo Pascal*

1 Создание каталогов для работы в *Turbo Pascal* (TP).

1.1 Загрузите *Norton Commander*, нажмите комбинацию клавиш **Alt F1**, выберите локальный диск D (или C) и подтвердите выбор нажатием клавиши **Enter**.

1.2 На диске D (или C) выберите каталог TP и нажмите клавишу **Enter**.

1.3 В каталоге TP найдите подкаталог bin и нажмите клавишу **Enter**.

1.4 Найдите каталог своей группы, например, ZFMT11 – факультет безотрывного обучения ZF, группа MT11, и нажмите клавишу **Enter**. При отсутствии этого каталога, создайте его, нажав клавишу **F7** и введя имя каталога.

1.5 Создайте подкаталог LAB2 для хранения файлов лабораторной работы.

2 Форматирование дискеты и копирование основных файлов среды TP.

2.1 Вставьте чистую дискету в дисковод.

2.2 В командной строке наберите `>format a:` и нажмите клавишу **Enter**.

2.3 Нажмите комбинацию клавиш **Alt F2** и выберите диск А.

2.4 Скопируйте на дискету основные файлы среды ТР: turbo.exe, turbo.tp, turbo.tph, turbo.tpl.

2.5 Нажмите комбинацию клавиш **Alt F2** и выберите локальный диск D.

2.6 Извлеките аккуратно дискету из дисковода.

Чтобы получить возможность выполнять задания за пределами университета, достаточно скопировать с дискеты или флэш-карты указанные файлы в любой каталог жесткого диска любого компьютера.

3 Копирование основных файлов и запуск среды ТР.

3.1 Из каталога bin в подкаталог своей группы (например, ZFMT11) скопируйте файлы среды ТР: turbo.tp, turbo.tph, turbo.tpl, turbo.exe.

3.2 Запустите стартовый файл turbo.exe нажатием клавиши **Enter**. На экране монитора откроется окно интегрированной среды ТР.

4 Решение первой задачи лабораторной работы.

Пример Г.1. Подсчитать значения функции $p = \sin \frac{a}{t}$, где $a = 1,6$, переменная $1 \leq t \leq 2$ изменяется с шагом $\Delta t = 0,2$.

4.1 Наберите *Pascal*-программу решения задачи.

```
program lab2_1;  
var a, t, p : real;  
begin  
a:=1.6;t:=1;  
while t<=2 do  
begin  
p:=sin(a/t);  
writeln(t:3:1,p:9:4);  
t:=t+0.2  
end  
end.
```

4.2 Сохраните программу в подкаталоге LAB2 каталога ZFMT11 под именем lab2_1.pas, выполнив команду **File / Save as**, введя имя файла

ZFMT11\LAB2\lab2_1

и нажав клавишу **Enter**.

4.3 Выполните компиляцию программы (**F9**) и исправьте возможные ошибки.

4.4 Запустите программу на выполнение (**Ctrl F9**).

4.5 Просмотрите полученный результат (**Alt F5**).

4.6 Измените раздел операторов программы lab2_1.pas, используя оператор цикла **repeat..until** следующим образом:


```

begin
a:=1.6;t:=1;
repeat
p:=sin(a/t);
writeln(t:3:1,p:9:4);
t:=t+0.2
until t>2
end.

```

4.7 Выполните компиляцию программы, запустите программу на выполнение и просмотрите полученный результат.

4.8 Модифицируйте программу lab2_1.pas, используя оператор цикла **for..to..do** следующим образом:

```

program lab2_1;
var a, t, p : real; i, n : integer;
begin
a:=1.6;t:=1; n:=round((2-1)/0.2)+1;
for i:=1 to n do
begin
p:=sin(a/t);
writeln(t:3:1,p:9:4);
t:=t+0.2
end
end.

```

4.9 Выполните компиляцию программы, запустите ее на выполнение и просмотрите полученный результат.

Задание Г.1. Составить программу на языке *Pascal* для решения задачи: вычислить значения функции $s = \cos \frac{p}{c}$, где $c = 1,22$, переменная $2 \leq p \leq 4$ изменяется с шагом $\Delta p = 0,3$. Сохранить программу под именем lab2_2.pas и выполнить ее.

5 Решение второй задачи лабораторной работы.

Пример Г.2. Ввести произвольные числа y_1, y_2, \dots, y_6 и значение $z = 2,34$. Для каждого y_i вычислить и вывести на экран значения

$$a_i = \sqrt{y_i + \frac{z}{2 + y_i}}.$$

Найти сумму элементов y_i и минимальное среди значений a_i .

5.1 Наберите программу решения задачи на языке *Pascal*:

```
program lab2_3;
var z, sum, min : real;
i : byte;
y, a : array[1..6] of real;
begin
z:=2.34;
writeln('Введите y[i]=');
for i:=1 to 6 do readln(y[i]);
for i:=1 to 6 do
begin
a[i]:=sqrt(y[i]+z/(2+y[i]));
writeln(y[i]:3:1,a[i]:7:3)
end;
sum:=0;
for i:=1 to 6 do sum:=sum+y[i];
min:=a[1];
for i:=1 to 6 do
if a[i]<min then min:=a[i];
writeln('Сумма y[i]=' ,sum:4:1, ' min=' ,min:7:3);
end.
```

5.2 Сохраните программу в подкаталоге LAB2 каталога ZFMT11 под именем lab2_3.pas. Для этого выполните команду **File / Save as**, введите имя файла

ZFMT11\LAB2\lab2_3

и нажмите клавишу **Enter**.

5.3 Выполните компиляцию программы (**F9**).

5.4 Запустите программу на выполнение (**Ctrl F9**).

5.5 Просмотрите полученный результат (**Alt F5**).

Задание Г.2. Составить программу на языке *Pascal* для решения задачи: Ввести числа x_1, x_2, \dots, x_6 и значение $y = 3,45$. Для каждого x_i вычислить и вывести на экран значения $z_i = \cos(x_i) + y$. Найти произведение элементов z_i и максимальное среди значений z_i .