Министерство образования, науки и молодежной политики Краснодарского края
государственное автономное профессиональное образовательное учреждение Краснодарского края
«Краснодарский информационно-технологический техникум»

Информационные технологии

Лабораторные роботы

для студентов второго курса направления подготовки 09.03.02
«ИФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»
всех форм обучения

ОГЛАВЛЕНИЕ

Л/р №1 «Стандартные программы Windows» …… … .4

Л/р №2 «Текстовый редактор Microsoft Word и табличный процессор Microsoft Excel» ……. 15

Л/р №3 «Графический редактор Adobe Photoshop 4.0» ……. 33

Л/р №4 «Математический пакет Mathcad 8 Pro» …… 46

Л/р №5 «Программа сканирования и распознования текстов FineReader» …59

Л/р №6 «Автоматизированная система лингвистического перевода PROMT 99» ……..80

Л/р №7 «Символьные вычисления с помощью программы Maple V»……94

ВВЕДЕНИЕ

Изучение дисциплины «Информационные технологии» является составной частью фундаментальной подготовки студентов специальности 210100. Она преподается в течение пятого и шестого семестров и, кроме изучения алгоритмических языков, предполагает освоение функциональных возможностей различных подсистем операционной системы Windows, а именно: текстового редактора Microsoft Word и табличного Microsoft Excel, графического редактора Adobe Photoshop 4.0, математического пакета Mathcad 8 Pro и других.

Выполнив лабораторную работу, студент должен знать структуру соответствующего программного обеспечения, его основные функции, правила использования его по назначению.

Технология выполнения лабораторных работ единообразна и включает в себя следующие этапы:

- теоретическое усвоение материала в объеме данных методических указаний и соответствующих разделов курса лекций;
- получение задания на работу, осмысление его и проведение необходимых подготовительных работ;
- выполнение задания с использованием вычислительной системы;
- оформление отчета;
- защиту выполненной лабораторной работы в виде теста.

Каждая лабораторная работа рассчитана на два или четыре аудиторных часа.

ЛАБОРАТОРНАЯ РАБОТА №1

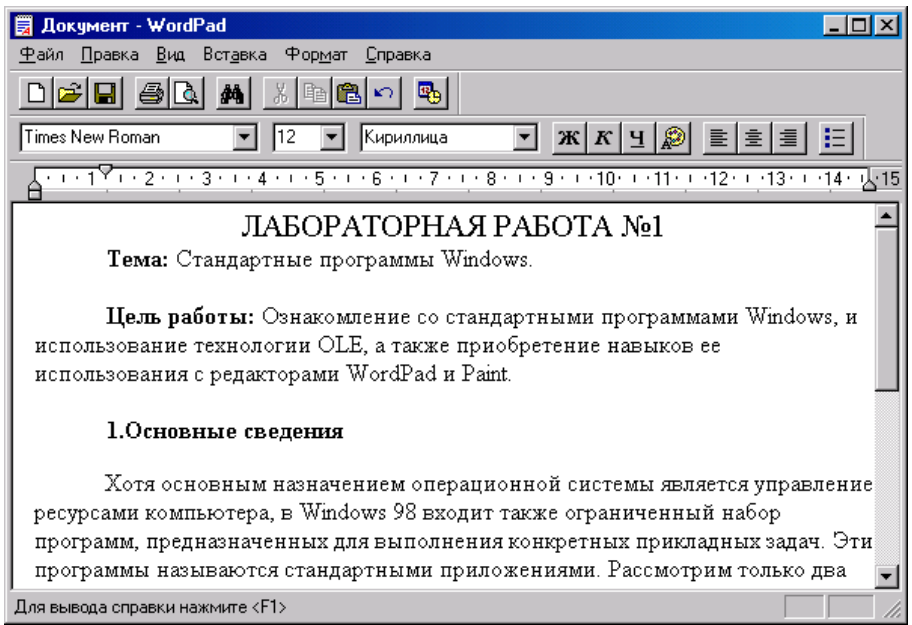
Тема: Стандартные программы Windows.

Цель работы: Ознакомление со стандартными программами Windows, и использование технологии OLE, а также приобретение навыков ее использования с редакторами WordPad и Paint.

1. Основные сведения о программе WordPad

Хотя основным назначением операционной системы является управление ресурсами компьютера, в Windows 98 входит также ограниченный набор программ, предназначенных для выполнения конкретных прикладных задач. Эти программы называются стандартными приложениями. Рассмотрим только два из них текстовый процессор WordPad и графический редактор Paint.

Текстовый процессор служит для создания, форматирования, редактирования и просмотра текстовых документов. Он вызывается командой **Пуск/Программы/Стандартные/ WordPad**. Основная область окна WordPad предназначена для отображения документа. Мигающая вертикальная черта в этой области отмечает точку ввода, в которую будут помещаться набираемые на клавиатуре символы. Следует обратить внимание, что клавиша Enter при наборе текста используется только для обозначения конца абзаца, а строки пе-



реносятся автоматически в соответствии с установленным на линейке размером либо заданным в диалоговом окне Абзац•, вызываемом командой **Формат/Абзац**. Удаление ошибочно набранного символа выполняется клавишей Backspace, если ошибка замечена сразу после набора символа, и он находится слева от точки ввода.

Рис.1.1 - Окно WordPad с набранным текстом документа

При редактировании документа изменения можно вносить в любую его часть. Для этого необходимо перемещаться по документу, что можно выполнить с помощью полос прокрутки (вертикальной и горизонтальной) окна WordPad либо клавиш управления клавиатуры. Для перевода точки ввода в начало текущей строки используют клавишу Home, а в конец строки клавишу End. Для перемещения точки ввода на размер области окна документа вверх или вниз используют клавиши PageUp и PageDown. Для произвольного позиционирования точки ввода следует щелкнуть мышью в нужном месте документа. Клавиатурные комбинации Ctrl + Home и Ctrl + End переводят точку ввода соответственно в начало всего документа или в его конец.

Двойной щелчок по слову выделяет его, а тройной щелчок выделяет весь абзац. Для выделения нескольких последовательных символов следует протащить по ним курсор мыши при нажатой левой кнопке. С выделением можно выполнять различные действия: удаление (клавиша Delete), копирование и перемещение (с помощью мыши или команд меню **Правка**).

Команда **Формат/Табуляция** позволяет задать позицию, в которую перемещается точка ввода после нажатия клавиши Tab. На линейке она отображается специальным символом. Это дает возможность создавать колонки текста или таблицы. Команда **Формат/Маркер** либо одноименная кнопка панели форматирования преобразует несколько выделенных абзацев в маркированный список.

Размещение документа на бумажной странице при выводе его на печать устанавливается в диалоговом окне, вызываемом командой **Файл/Макет страницы**. В нем можно выбрать размер используемых листов бумаги и задать величину полей.

Сохранение документа текстового процессора WordPad выполняется с расширением DOC, которое относится к программе Word из пакета Microsoft Office. При первом сохранении открывается окно Сохранить как•, позволяющее выбрать имя файла и папку для его хранения. При сохранении документа, файл которого уже существует, команда **Файл/Сохранить** заместит его копией с новым содержимым в той папке, где он находится.

Открывать ранее сохраненные документы WordPad следует командой **Файл/Открыть**, вызывающей окно Открыть• для выбора имени и места нахождения открываемого файла.

2. Основные сведения о программе WordPad

Программа Paint простейший графический редактор, предназначенный для создания и редактирования рисунков. Вызов программы осуществляется командой **Пуск/Программы/ Стандартные/Paint**. Основная область окна Paint предназначена для отображения рисунка, представленного в виде растра, то есть множества точек (пикселов).

Окно программы Paint содержит также строку меню, панель инструментов с областью настройки инструмента и палитру цветов. Кнопки панели инструментов служат для выбора инструментов рисования, а в области настройки инструмента можно задать его параметры (толщину линии, метод заполнения и т.п.). На палитре цветов щелчок левой кнопкой выбирает основной цвет, а правой кнопкой фоновый цвет. Все инструменты, кроме Ластика, выполняют рисование основным цветом. Ластик стирает изображение, заменяя его фоновым цветом. Инструмент Карандаш предназначен для рисования произвольных ли-

ний, толщину которых выбирают в области настройки. Инструмент Линия предназначен для вычерчивания прямых методом протягивания указателя мыши по рабочей области. Удерживая при этом нажатой клавишу Shift, можно получить точную ориентацию линии (строго горизонтальную, вертикальную или под углом 45°). Инструменты Прямоугольник, Эллипс, Многоугольник и Скругленный прямоугольник применяют для рисования соответствующих фигур, которые могут быть показаны только рамкой (без заполнения) либо с заполнением основным или фоновым цветом. Инструменты Кисть и Распылитель служат для свободного рисования пятнами выбранного размера.

Две верхние кнопки служат для выделения области рисунка: произвольной и прямоугольной. С выделенной областью можно производить все действия, предусмотренные в операционной системе Windows. Прием копирования и вставки выделенной области используют для создания повторяющихся фрагментов рисунка. При этом можно выбрать один из двух режимов: с непрозрачным или прозрачным фоном фрагмента.

В меню **Рисунок** представлены команды для трансформации графических объектов. Команда **Отразить/повернуть** вызывает диалоговое окно, позволяющее создать объект с зеркальной или центральной симметрией, фиксированно расположенный по отношению к исходному. Команда **Растянуть/наклонить** открывает окно, позволяющее выбрать параметры растяжения и наклона объекта относительно горизонтальной или вертикальной оси. Команда **Обратить цвета** заменяет цвет каждой точки изображения на дополнительный (то есть тот, который в сумме с первоначальным цветом дает белый цвет).

Так как в рисунках могут содержаться надписи, то инструмент Надпись позволяет ввести ее текст, который после выбора атрибутов символов преобразуется в растровый рисунок. Это накладывает ограничения на размер и количество используемых символов, а также определяет невозможность их последующего редактирования.

Сохранение рисунка выполняется аналогично текстовому документу, но расширение файла BMP указывает на то, что файл содержит растровый рисунок.

3. Технология OLE

Для создания документов, содержащий и текст и рисунки в Windows используется технология OLE (Object Linking and Embedding) выполнять внедрение и связывание объектов. Эта технология обеспечивает совместную работу нескольких приложений при создании составного документа, позволяет перемещать и копировать объекты между приложениями, а также хранить объекты в отдельных файлах, связанных с документом, где этот объект используется.

Обязательным условием для создания составного документа является поддержка технологии OLE-приложением. При этом различают приложения, их называют OLE-серверы, которые позволяют только создавать объекты для использования другими приложениями. Последние называют OLE-клиентами, так как они позволяют внедрять и связывать чужие объекты в своих документах. Так графический редактор Paint является OLE-сервером, а текстовый редактор WordPad OLE-клиентом. Поэтому составные документы создаются с его помощью и сохраняются в файлах WordPad (имеющих расширение DOC).

Под **внедрением объекта** подразумевается сохранение в файле составного документа не только всей информации об объекте, но и о приложении, в котором он был создан. Это позволяет в дальнейшем редактировать объект с помощью этого приложения. Для перехода в режим редактирования на объекте в составном документе выполняют двойной щелчок. Это действие открывает создавшее объект приложение, причем возможны два варианта работы с ним, в зависимости от организации взаимодействия приложений:

- приложение-сервер открывается в отдельном окне, и после редактирования объекта в меню **Файл** этого окна выбирают команду **Выход и возврат** для возврата к составному документу в приложении-клиенте;
- в окне приложения-клиента меню команд и панели инструментов заменяются меню и панелями приложения-сервера, и после редактирования объекта следует щелкнуть в любом другом месте документа для активизации приложения-клиента.

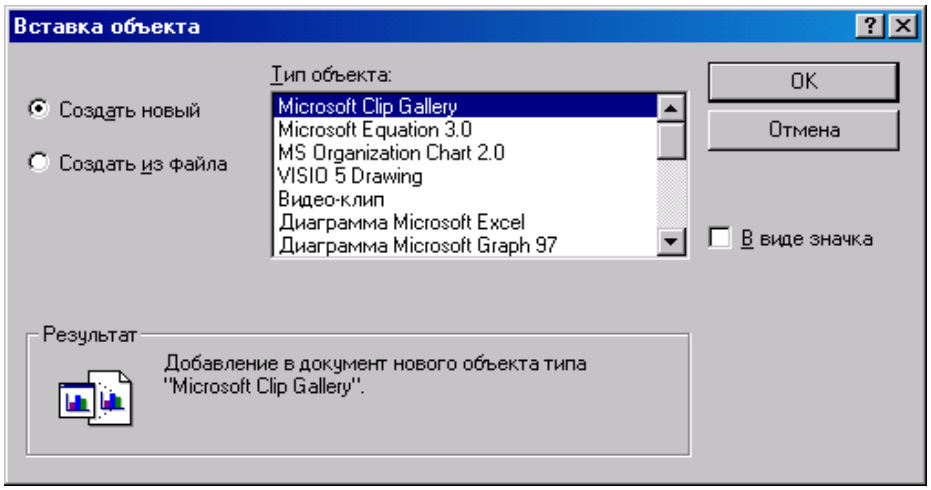
Эта технология позволяет внедрять в составной документ объекты самого различного типа, например, звуковые или видео. Конечно, они не могут быть воспроизведены средствами WordPad, поэтому в документе отображаются значком, двойной щелчок по которому активизирует просмотр объекта с помощью создавшего его приложения. Следует иметь в виду, что размеры файлов составных документов в этом случае оказываются очень большими, так как каждый из них содержит отдельную копию объекта. Редактирование одной из копий не окажет никакого воздействия на остальные, поэтому составные документы могут содержать разные варианты одного и того же исходного объекта.

Связывание объектов требует, чтобы они хранились в отдельных файлах. В составной документ в этом случае вставляется либо формула связи (имя файла и путь доступа к нему), которая лишь незначительно увеличит размер файла документа, либо такая же формула вместе с полной информацией об объекте. В первом случае при переносе файла составного документа на другой компьютер следует вместе с ним перенести и все файлы связанных объектов, во втором файл составного документа содержит всю информацию об объектах, поэтому при его переносе окажутся разорваны только связи с файлами объектов.

Связывание объектов позволяет редактировать объект независимо от составного документа. При загрузке составного документа со связями приложение-клиент запрашивает пользователя о необходимости их обновления. Если пользователь отвечает на запрос, то объект в составном документе загружается из файла заново, в противном случае составной документ отображает преды-

дущее состояние объекта. Отсюда становится ясно, что если в составном документе хранится только адрес файла объекта, то его загрузка каждый раз выполняется заново, а в случае отсутствия файла по адресу, хранящемуся в формуле связи, отображение объекта будет невозможно. В такой ситуации программа-клиент предлагает пользователю самому указать новое место нахождения файла. Это действие называется восстановлением связи.

Редактирование связанного объекта выполняется в приложении-сервере всегда в отдельном окне. После изменения объекта он должен быть сохранен в том же самом файле, что позволяет получить его новую версию во всех связанных с ним документах. Поэтому связывание чаще всего применяется при сов-



местном использовании объектов (логотип фирмы, бланки стандартных документов и т.п.).

Рис.1.2 - Окно "Вставка объекта" со списком приложений

Внедрение или связывание выполняется по команде **Вставка/Объект (Insert Object)**, открывающей одноименное окно. В нем перечислены все создающие различные объекты приложения, которые установлены на данном компьютере. Выбор одного из них и щелчок по кнопке ОК окна запустит это приложение для создания объекта с нуля•. Если же объект уже существует и хранится в файле, то в окне Вставка объекта• следует выбрать значение переключателя Создать из файла. В этом случае содержимое окна видоизменяется,

предлагая строку ввода для описания полного пути доступа к файлу. Кнопка Обзор (Browse) позволяет вставить этот путь с помощью окна навигации по папкам. Установка флажка Связь (Link to File) связывает объект с исходным файлом, а установка флажка В виде значка (Display as Icon) замещает объект значком.

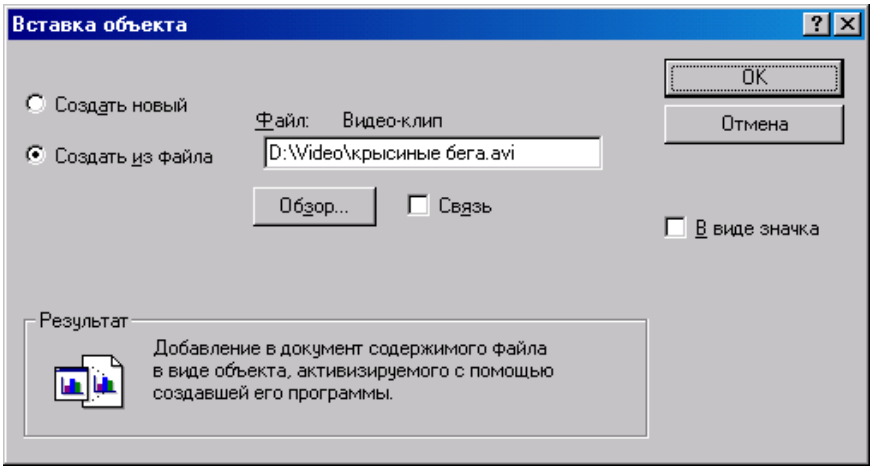


Рис.1.3 - Окно "Вставка объекта" для выбора файла объекта

С использованием команды **Вставка/Объект** он вставляется весь целиком, однако иногда необходимо вставить только часть объекта. Поэтому чаще используют встраивание объектов с помощью Буфера обмена (Clipboard). В этом случае открывают объект в приложении-сервере и выделяют ту часть, которую необходимо встроить в составной документ. Скопировав ее в Буфер обмена, переходят в приложение-клиент и используют команду **Правка/Вставить (Edit Paste)**. Выделенная часть объекта появится в точке вставки. Различая разнообразные по характеру объекты, Буфер обмена может изменять их формат в пределах возможных видов их представления. Обычно в качестве формата по умолчанию он использует формат представления исходного приложения-сервера. Для управления форматом встраиваемого объекта следует использовать команду **Правка/Специальная вставка (Edit Paste Special)**, которая выводит окно со списком доступных форматов. Выбирая нужное представление объекта можно также изменить его приложение-сервер.

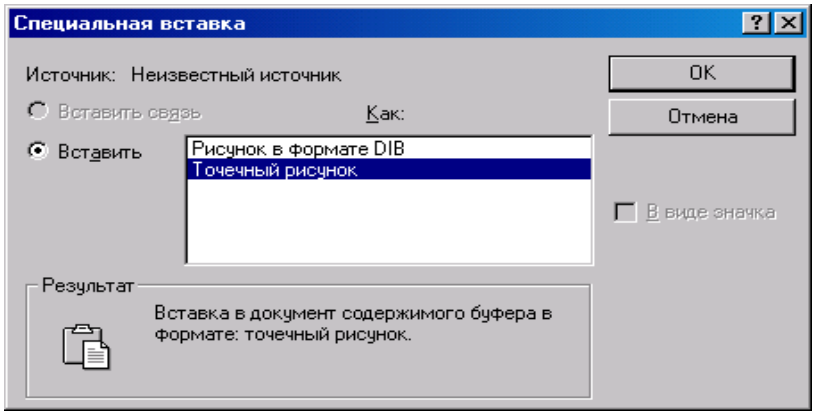


Рис.1.6 - Окно "Специальная вставка" для выбора формата объекта

Порядок выполнения работы:

1. Откройте окно текстового процессора WordPad.
2. Наберите на клавиатуре текст двух первых абзацев данного урока без соблюдения форматирования. Для изменения языка раскладки клавиатуры используйте индикатор раскладки на Панели задач.
3. ыделите текст: **Формат/Абзац•** и задайте полужирное начертание.
4. Сохраните документ под именем Пример.doc.
5. Откройте окно графического редактора Paint.
6. Задайте на палитре основной цвет черный, а фоновый белый. Нарисуйте правильный круг. Нарисуйте второй круг без заливки, перекрывающий первый.
7. Задайте на палитре основной цвет красный. Выберите в качестве основного цвета красный и выполните заливку замкнутого.контура рисунка. Выберите в качестве основного цвета желтый и выполните заливку второго круга. Аналогично заполните зеленым цветом фон всего рисунка.
8. Сохраните рисунок в файле с именем Круги.bmp.

9. Закройте окна программ WordPad и Paint.
10. Запустите текстовый процессор WordPad.
11. Откройте файл Пример.doc, и установите точку ввода в начало второго абзаца.
12. Командой **Вставка/Объект** вызовите одноименное окно, в котором выберите значение переключателя Создать из файла. В строке ввода укажите путь доступа к файлу Круги.bmp либо найдите его с помощью кнопки Обзор.
13. Вставьте рисунок в текст. Измените размер рисунка в документе, перетаскивая размерные маркеры.
14. Перейдите в конец документа и повторите вставку рисунка из файла, установив в окне Вставка объекта• флажок Связь. Также подберите подходящий размер рисунка.
15. Загрузите файл Круги.bmp в графический редактор Paint и измените цвет заливки областей. Сохраните изменения в том же файле.
16. Вернитесь в окно WordPad и убедитесь, что рисунки не изменились. Сохраните документ и закройте WordPad.
17. Заново откройте WordPad, загрузите документ Пример.doc, подтвердите необходимость обновления связи и проверьте, изменились ли рисунки.
18. Перейдите в окно графического редактора Paint и выделите часть рисунка. Скопируйте выделение в Буфер обмена.
19. Возвратитесь в окно WordPad и вставьте содержимое Буфера обмена обычным способом, а затем командой **Правка/Специальная вставка**, выбрав другой формат для рисунка.
20. Сохраните составной документ и закройте все окна.

Вопросы к защите:

1. Перечислите основные команды, используемые в редакторах WordPad и Paint.
2. Объясните назначение меню Формат WordPad и Рисунок Paint.
3. В чем различие между связью и внедрением объектов?
4. Назовите типы объектов, которые можно вставить на данном компьютере.
5. Объясните, как можно изменить OLE – сервер для внедренного объекта.

ЛАБОРАТОРНАЯ РАБОТА №2

Тема: Текстовый редактор Microsoft Word и табличный процессор Microsoft Excel.

Цель: Научится создавать простые и комплексные документы с помощью Microsoft Word, и Excel.

1. Основные сведения о программе Microsoft Word

Microsoft Word - мощный текстовой редактор, предназначенный для выполнения различных видов обработки текста. В одно из многих полезных свойств Word входит автоматическая коррекция текста по границам, автоматический перенос слов и проверка правописания слов, сохранение текста в определенный устанавливаемый промежуток времени, наличие мастеров текстов и шаблонов. Word позволяет открывать много окон для одновременной работы с несколькими текстами, а так же разбить одно активное окно по горизонтали на два и выровнять их.

2. Краткое описание основных операций работы с текстом

1. Для создания текстового файла, документа или документов Word, нужно выполнить одно из следующих действий:

№	Возможный вариант выполнения операции
1	Выйти в верхнее меню редактора, выбрать пункт меню <Файл>, выбрать пункт меню <Создать>, нажать клавишу<Enter>
2	Нажать на комбинацию клавиш <Ctrl>+<N>, выбрать нужный файл и нажать кнопку <Ok>.

3	Нажать на кнопку <Создать> в верхнем меню, выбрать нужный файл и нажать кнопку <Ok>.

2. Для открытия нового текстового файла, документа, шаблона документов или просто документов Word, нужно выполнить одно из следующих действий:

№	Возможный вариант выполнения операции
1	Выйти в верхнее меню редактора , выбрать пункт меню <Файл>, выбрать пункт меню <Открыть>, выбрать нужный файл и нажать кнопку <Ok>.
2	Нажать на комбинацию клавиш <Ctrl>+<O>, выбрать нужный файл и нажать кнопку <Ok>.
3	Нажать на кнопку <Открыть> в верхнем меню, выбрать нужный файл и нажать кнопку <Ok>.

3. Для сохранения текстового файла, документа, шаблона документов или документов Word, нужно выполнить одно из следующих действий:

№	Возможный вариант выполнения операции
1	Выйти в верхнее меню редактора , выбрать пункт меню <Файл>, выбрать пункт меню <Сохранить>, выбрать нужный файл и нажать кнопку <Ok>.
2	Нажать на комбинацию клавиш <Ctrl>+<S>, выбрать нужный файл и нажать кнопку <Ok>.
3	Нажать на кнопку <Сохранить> в верхнем меню, выбрать нужный файл и нажать кнопку <Ok>.

4. Для проверки орфографии текстового файла, документа, шаблона документов или документов Word, нужно выполнить одно из следующих действий:

№	Возможный вариант выполнения операции
1	Выйти в верхнее меню редактора, выбрать пункт меню <Сервис>, выбрать пункт меню <Орфография>, нажать клавишу <Enter> .
2	Нажать на клавишу <F7>.
3	Нажать на кнопку <Орфография> в верхнем меню.

3. Краткое описание работы с фрагментами текста

1. Для выделения фрагмента в текстовом файле, документе или шаблоне документов Word, нужно выполнить следующие действия:

№	Возможный вариант выполнения операции
1	Нажать клавишу <Shift> , удерживая ее, с помощью клавиш управления курсором, выделить нужный фрагмент в тексте.
2	Установить курсор манипулятора мышь на начало фрагмента в тексте, который надо выделить, затем нажать на левую клавишу манипулятора и отметить нужный фрагмент в тексте, после чего отпустить левую клавишу манипулятора.

2. Для того, чтобы скопировать фрагмент текста, нужно выполнить один из следующих вариантов следующих действий:

Первый вариант выполнения операции	Второй вариант выполнения операции	Третий вариант выполнения операции
Отметить нужный фрагмент текста.	Отметить нужный фрагмент текста.	Отметить нужный фрагмент текста.

Поместить выделенный фрагмент в буфер обмена, выбрав пункт меню <Копировать> в пункте верхнего меню <Правка>	Поместить выделенный фрагмент в буфер обмена нажав на комбинацию клавиш <Ctrl>+<C>.	Поместить выделенный фрагмент в буфер обмена нажав кнопку <Копировать>в верхнем меню.
Установить курсор на то место, куда будет перенесен фрагмент.	Установить курсор на то место, куда будет перенесен фрагмент.	Установить курсор на то место, куда будет перенесен фрагмент.
Нажать на клавишу <Insert> или на <Вставить> в меню	Нажать на клавишу <Insert> или кнопку <Вставить> в верхнем меню.	Нажать на клавишу <Insert> или на <Вставить> в меню.

3. Для того, чтобы удалить фрагмент текста, нужно выполнить один из следующих вариантов следующих действий:

Первый вариант выполнения операции	**Второй вариант выполнения операции**
Отметить нужный фрагмент текста.	Отметить нужный фрагмент текста.
Выбрать пункт меню <Очистить> в пункте верхнего меню <Правка>.	Нажать на клавишу <Del>.

4. Для того, чтобы вырезать фрагмент текста и поместить его в буфер обмена, нужно выполнить один из следующих вариантов следующих действий:

Первый вариант выполнения операции	**Второй вариант выполнения опера-**	**Третий вариант выполнения операции**

	ции	
Отметить нужный фрагмент текста.	Отметить нужный фрагмент текста.	Отметить нужный фрагмент текста.
Выбрать пункт меню <Вырезать> в пункте верхнего меню <Правка>.	Нажать на комбинацию клавиш <Ctrl>+<X>.	Нажать на кнопку <Вырезать> в верхнем меню.

5. Для того, чтобы изменить шрифт фрагмент текста, нужно выполнить один из следующих вариантов следующих действий:

Сделать шрифт полужирным	*Сделать шрифт курсивным*	Сделать шрифт подчеркнутым
Отметить нужный фрагмент текста.	Отметить нужный фрагмент текста.	Отметить нужный фрагмент текста.
Нажать на кнопку <Полужирный>, на верхней панели меню.	Нажать на кнопку <Курсив>, на верхней панели меню.	Нажать на кнопку <Подчеркивание>, на верхней панели меню.

4. Структура документов в Microsoft Excel

Трехмерные документы (блокноты, папки) - одно из важнейших новшеств программы Excel, позволяющее значительно упростить и ускорить работу с таблицами и управление рабочими листами. В каждом рабочем блокноте можно поместить всю деловую информацию, относящуюся к одной теме, и хранить

ее в одном файле, что позволяет значительно повысить наглядность рабочих документов.

4.1. Добавление рабочих листов

Директивы добавления расположены в меню **Insert**. Пользователь может добавить в документ элементы следующих типов:

- рабочие листы для создания таблиц;
- диаграммы (в качестве элемента таблицы или на отдельном листе);
- рабочий лист для записи макрокоманды в виде программного модуля;
- рабочий лист для создания диалогового окна.

Новый лист всегда вставляется перед активным рабочим листом. Если лист предназначен для создания таблицы, то независимо от занимаемой позиции он будет иметь название Sheet17 с последующим увеличением номера при добавлении новых таблиц. Новые диаграммы, расположенные на отдельных рабочих листах, нумеруются, начиная с Chart1 и т.д.

5. Построение таблиц

Все данные таблицы записываются в так называемые ячейки, которые находятся на пересечении строк и столбцов таблицы. По умолчанию содержимое ячейки представляется программой Excel в стандартном формате, который устанавливается при запуске программы. Например, для чисел и текстов задается определенный вид и размер шрифта.

5.1. Маркирование ячеек

Отдельные ячейки таблицы маркируются (выделяются) автоматически с помощью указателя ячеек. Одна строка или столбец таблицы маркируются

щелчком по номеру (адресу), который расположен в заголовке строки или столбца. Для маркирования нескольких строк или столбцов нужно щелкнуть по номеру первой строки (столбца), а затем, удерживая кнопку мыши нажатой, переместить манипулятор в нужную позицию.

5.2. Форматирование чисел

Прежде всего, нужно промаркировать ячейки, в которых надо изменить формат представления чисел. После этого можно либо открыть правой кнопкой мыши контекстное меню и вызвать в нем директиву **Format Cells**, либо вызвать директиву **Cell** из меню **Format**. В любом случае на экране появится диалоговое окно **Format Cells**. По умолчанию в верхнем поле этого окна будет выбрана опция **Number**, которая предназначена для форматирования чисел. В поле **Category** приведены различные типы форматов, которые выбираются щелчком мыши или с помощью клавиш управления курсором. В поле **Format Codes**: показаны коды форматов выбранного типа. В поле ввода **Code**: пользователь может задать код своего формата и запомнить его в списке форматов. В нижней части окна **Format Cells** находится поле просмотра **Sample**, в котором показано, как будет выглядеть содержимое ячейки в выбранном формате.

6. Табличные вычисления

Возможность использования формул и функций является одним из важнейших свойств программы обработки электронных таблиц. Это, в частности, позволяет проводить статистический анализ числовых значений в таблице.

Текст формулы, которая вводится в ячейку таблицы, должен начинаться со знака равенства, чтобы программа Excel могла отличить формулу от текста. После знака равенства в ячейку записывается математическое выражение, содержащее аргументы, арифметические операции и функции.

В качества аргументов в формуле обычно используются числа и адреса ячеек. Для обозначения арифметических операций могут использоваться следующие символы: + (сложение); - (вычитание); * (умножение); / (деление).

Формула может содержать ссылки на ячейки, которые расположены на другом рабочем листе или даже в таблице другого файла. Однажды введенная формула может быть в любое время модифицирована. Встроенный Менеджер формул помогает пользователю найти ошибку или неправильную ссылку в большой таблице.

6.1. Ввод формул

Программа Excel интерпретирует вводимые данные либо как текст (выравнивается по левому краю), либо как числовое значение (выравнивается по правому краю). Для ввода формулы необходимо ввести алгебраическое выражение, которому должен предшествовать знак равенства.

Ввод формул можно существенно упростить, используя маленький трюк. После ввода знака равенства следует просто щелкнуть мышью по первой ячейке, затем ввести операцию деления и щелкнуть по второй ячейке.

6.2. Сложные формулы

Применение сложных формул продемонстрируем на примере.

	B	**C**	**D**	**E**
10	РАБОТА	ВРЕМЯ	ЦЕНА	СУММА
11	Настройка клавиатуры	2	$ 17.80	$ 35.60
12	Обслуживание машин	4	$ 3.80	$ 15.20
13	Подготовка формуляров	8	$ 1.56	$ 12.48
14				

15		
16	Сумма	$ 63.28
17	НДС 15%	$ 9.49

В столбце С приведенной таблицы указано время (в часах), затраченное на выполнение работы, в столбце D - стоимость одного часа работы, а в столбце Е - промежуточная сумма, которую надо заплатить за эту работу. В ячейке Е16 нужно вычислить общую стоимость всех работ. Для этого туда надо записать следующую формулу: =E12+E13+E14. Полученную сумму умножьте на 0,15, чтобы учесть налог на добавленную стоимость, и результат поместите в ячейке Е17: =E16*0,15.

Для вычисления конечной суммы, подлежащей оплате (например, в ячейке Е19) надо сначала сложить промежуточные суммы, а затем результат умножить на 0,15. Формула должна иметь вид: =(E12+E13+E14)*0,15. Конечно, можно было бы и просто просуммировать содержимое ячеек Е16 и Е17. Для сложения чисел можно также использовать функцию суммы SUM(), тогда формула будет выглядеть следующим образом: =Sum(E12:E14)*0,15.

6.4. Групповые имена

Предположим, что необходимо вычислить сумму целой группы ячеек. Вместо того, чтобы перечислять в формуле отдельные ячейки, промаркируйте всю группу и присвойте ей имя. В дальнейшем это имя можно будет использовать в формулах.

Не допускается использование пробелов. Имя группы не должно совпадать с адресами ячеек (А1 и т.п.). В этом окне приводится также список уже присвоенных групповых имен, относящихся к данному блокноту.

Если в Вашей таблице есть заголовки строк и столбцов, то их также можно использовать в качестве имен этих областей. Для этого нужно промаркировать

соседние строки (столбцы), включая первые ячейки, где расположены имена, и вызвать директиву **Create** из меню **Insert Name**. В открывшемся диалоговом окне нужно указать местонахождение имен (в первой или последней ячейке строки или столбца) и щелкнуть по командной кнопке **OK**.

Если теперь вызвать диалоговое окно **Define Name**, то в списке имен можно увидеть, что программа Excel присвоила указанным строкам (столбцам) эти имена.

В нижнем поле **Refers to** диалогового окна **Define Name** даются табличные адреса именуемой группы. Теперь при вводе формулы вместо перечисления адресов ячеек, образующих эту группу, можно указать ее имя.

Полный список всех групповых имен расположен слева от наборной строки. Во время ввода формулы можно открыть этот список, щелкнув по стрелке списка, и выбрать в нем интересующее Вас имя. Выбранное имя группы будет сразу вставлено в формулу, независимо от местонахождения активной ячейки.

Например, если столбцу E (см. рис. 1) присвоено имя "Сумма" и нужно вычислить сумму ячеек E11, E12 и E13, то ввод формулы надо начать со знака равенства, за которым следует имя функции суммы с одной круглой скобкой: =Sum(.

Затем нужно открыть список групповых имен и щелкнуть мышью по имени "Сумма". Программа Excel вставит это имя в формулу. Остается только ввести правую скобку и нажать клавишу [Enter], чтобы завершить ввод формулы.

7. Построение и оформление диаграмм

Графические диаграммы оживляют сухие колонки цифр в таблице, поэтому уже в ранних версиях программы Excel была предусмотрена возможность построения диаграмм.

Конструктор диаграмм является одним из наиболее мощных средств в программе Excel. Построение диаграммы с его помощью выполняется за несколько

шагов. Конструктору указывается исходная область таблицы, тип диаграммы, используемые надписи и цвета. На основной панели имеется пиктограмма для вызова Конструктора диаграмм.

7.1. Построение диаграмм

Прежде чем строить диаграмму, необходимо закончить все работы в таблице, включая ее форматирование. Вызовите Конструктор диаграмм (ChartWizard).

Конструктор диаграмм будет ожидать информации о местонахождении диаграммы, поэтому указатель мыши приобретает форму пиктограммы Конструктора. Вы должны сделать с помощью указателя мыши прямоугольник на свободном месте рабочего листа, где будет размещена диаграмма. Для этого щелкните левой кнопкой мыши, а затем, не отпуская ее, отбуксируйте щелчка пунктирную рамке из верхнего левого угла прямоугольной образовавшуюся после области в правый нижний угол. Если диаграмму нужно построить на отдельном листе, то откройте в меню **Insert** подменю **Chart** и выберите в нем директиву **As New Sheet**.

После этого Конструктор открывает диалоговое окно, в котором за пять шагов нужно построить диаграмму. Но сначала нужно указать позицию, где находится таблица данных, на основе которой строится диаграмма. Для этого на первом шаге промаркируйте мышью свою таблицу, включая заголовки строк и столбцов.

Чтобы выполнить два следующих шага, щелкните по командной кнопке **Next** и выберите из предложенных образцов тип и вид диаграммы соответственно. На четвертом шаге Конструктор предлагает предварительный просмотр получившейся диаграммы. На последнем (пятом) шаге остается только задать имя диаграммы, ее легенду (т.е. соответствие между столбцами диаграммы и табличными данными), а также указать надписи на осях диаграммы.

В заключении щелкните по командной кнопке **Finish**, и диаграмма появится в указанной области рабочего листа.

8. Обработка списков

Списки позволяют эффективно работать с большими однородными наборами данных.

Списком будем называть упорядоченный набор данных, имеющих одинаковую структуру. Например, списком является телефонный справочник, в котором приведены фамилии абонентов и номера их телефонов. Каждый элемент списка занимает одну строку, в которой данные распределяются по нескольким полям (столбцам).

8.1. Создание списков

С технической точки зрения списки представляют собой обычные таблицы, для которых в программе Excel имеются дополнительные операции обработки. Сначала нужно хорошо продумать названия отдельных полей в списке. Названия полей используются программой Excel для поиска элементов в списке и его обработки. Эти названия соответствуют названиям столбцов в таблице.

В первой строке рабочего листа введите названия отдельных полей списка. Следите за тем, чтобы между ними не оставалось свободных столбцов. Эту строку программа Excel использует в качестве строки заголовков списка. Начиная со следующей строки, введите свои данные.

8.2. Ввод списка данных

После вызова директивы **Form** из меню **Data** открывается диалоговое окно, в котором программа спрашивает, можно ли рассматривать первую строку в

качестве заголовка списка. Здесь нужно ответить утвердительно, щелкнув по кнопке **OK**.

После этого откроется диалоговое окно для ввода данных, в котором каждому полю списка соответствует поле ввода. Как только все поля ввода будут заполнены данными, щелкните по командной кнопке **New**, чтобы внести введенный элемент данных в список.

Каждый новый список следует создавать на отдельном рабочем листе. Программа Excel может обрабатывать только один список на одном рабочем листе.

9. Анализ данных

Для анализа больших списков данных в программе Excel предусмотрены специальные средства.

Под анализом данных здесь понимаются методы, позволяющие лучше понять тенденции и закономерности, которым подчиняются табличные данные. Опорные таблицы (**Pivot Table**), включенные в пятую версию программы Excel, предназначены для анализа соотношений между данными в списке. Опорные таблицы строятся с помощью Конструктора таблиц за четыре шага. Структуру опорной таблицы можно легко изменить в интерактивном режиме путем перемещения названий полей данных из одной части таблицы в другую.

9.1. Опорные таблицы

Директивой **PivotTable** из меню **Data** откройте диалоговое окно Конструктора опорных таблиц. В этом окне за четыре шага нужно построить опорную таблицу.

На первом шаге следует указать источник поступления данных. Если исходные данные находятся в списке, то щелкните по селекторной кнопке

Microsoft Excel List or Database и перейдите к следующему шагу с помощью командной кнопки **Next**.

На втором шаге нужно указать область рабочего листа, где находится список. По умолчанию программа предлагает список, который расположен на активном рабочем листе. Здесь можно указать только часть списка, если для построения опорной таблицы остальная часть не нужна. Щелкните по командной кнопке **Next**, чтобы перейти к следующему шагу.

На третьем шаге задается структура (**Layout**) опорной таблицы. В правом поле расположены названия полей списка, а в среднем - представление структуры опорной таблицы. В это поле включается поле названий строк, поле названий столбцов и поле данных. Чтобы задать структуру таблицы, следует отбуксировать мышью названия полей списка в соответствующие поля таблицы. При построении опорной таблицы придерживайтесь следующих правил. Названия полей, по значениям которых будет проводится анализ данных, отбуксируйте в поле данных **Data**. В поле названий строк **Row** и в поле названий столбцов **Column** переместите названия полей, по значениям которых классифицируются (разбиваются) данные. Затем щелкните по командной кнопке **Next**, чтобы перейти к следующему шагу.

На четвертом шаге можно установить некоторые опции для построения таблицы. В поле ввода **Pivot Table** Starting **Cell** указывается адрес, с которого начинается построение таблицы (левый верхний угол). В поле ввода **Pivot Table Name** можно указать название таблицы. По умолчанию программа использует название **PivotTable**1 с последующим увеличением номера таблицы. Если включить контрольные индикаторы **Grand Totals for Columns** и **Grand Totals for Rows**, то в каждой строке и в каждом столбце опорной таблицы будут вычисляться общие суммы. Контрольный индикатор **Save Data With Table Layout** служит для сохранения опорной таблицы с заданной структурой. Включение контрольного индикатора **AutoFormat Table** позволяет автоматически сформатировать опорную таблицу. После закрытия диалогового окна Кон-

структора командной кнопкой **Finish** программа Excel размещает опорную таблицу в указанной позиции.

Между опорной таблицей и исходным списком возникает статическая связь, т.е. связь, которая существует только в момент построения таблицы. При изменении данных в списке опорная таблица не будет изменяться. Чтобы привести в соответствие содержимое таблицы и содержимое списка, нужно вызвать директиву **Refresh Data** из меню **Data**.

9.2. Нахождение значений

Иногда бывает заранее известно, каким должен быть результат вычисления некоторой функции, но неизвестно, при каких значениях аргумента он достигается.

Предположим, что нам надо найти максимальную сумму кредита, которую можно дать на три года по фиксированной процентной ставке, если клиент готов возвращать по 2000$ ежемесячно.

В программе Excel имеется функция PMT, которая по значениям суммы кредита, периода и процентной ставки вычисляет величину ежемесячного взноса для погашения кредита. Нам нужно решить обратную задачу, когда заранее известна величина ежемесячно возвращаемой суммы и нужно найти сумму кредита. Для решения этой задачи вызовите директиву Goal Seek из меню **Tools**.

Откроется диалоговое окно этой директивы. В поле ввода Set **Cell**: этого окна укажите адрес ячейки, в которой расположено заданное значение функции (в нашем случае это ячейка, где находится функция PMT). В поле ввода **To Value** укажите значение функции, которое в нашем случае равно 2000$. В поле ввода **By changing Cell** укажите адрес ячейки, в которой программа сможет подбирать нужное значение, удовлетворяющее заданным условиям. В нашем

случае здесь надо указать адрес ячейки, где находится значение суммы кредита. Закончив установки полей, щелкните по командной кнопке **OK**.

Путем подбора значений в указанной ячейке программа Excel пытается найти максимальное приближение к заданному целевому значению функции. Результат появляется в диалоговом окне. Если Вас удовлетворяет расхождение между найденным и целевым значением, то закройте диалоговое окно щелчком по кнопке **OK**, после чего найденное значение появится в таблице. В противном случае щелкните по командной кнопке **Cancel**, чтобы не передавать значение в таблицу.

10. Базы данных

В комплект поставки Excel входит программа обработки внешних баз данных MS Query.

Программа Query (Запрос) может работать с текстовыми файлами и с данными, которые были подготовлены следующими программами управления базами данных: Ms Access, dBASE, FoxPro, Oracle, Paradox, SQL Server. Набор данных, хранящийся на диске в формате, одной из перечисленных выше баз данных, будем называть внешней базой данных.

10.1. Запуск программы обработки баз данных

Вызовите Менеджер расширений директивой **Add-Ins** из меню **Tools**. Откроется диалоговое окно Менеджера. В нем можно выбрать из списка опции расширения, которые включаются и выключаются щелчком по соответствующей опциональной кнопке, расположенной перед именем расширения.

По списку опций можно перемещаться с помощью линейки прокрутки. Найдите в списке строку MS Query **Add-In** и щелкните по кнопке в этой строке,

чтобы в ней появился крестик. Затем закройте диалоговое окно щелчком по командной кнопке **OK.**

Теперь загляните в меню **Data**. В нижней части этого меню должна появиться директива **Get External Data**, с помощью которой можно запускать программу Query.

10.2. Оболочка программы Query

Окно программы Query содержит многие элементы, которые являются общими для всех программ в среде Windows. В верхней части экрана находится строка заголовка с кнопками увеличения и уменьшения размеров окна. Под ней расположена строка меню с элементами меню, которые содержат директивы работы с внешними базами данных.

Под строкой меню находится строка пиктографического меню (пиктографическая панель), в которой расположены пиктограммы для вызова наиболее часто используемых директив. Если установить указатель мыши на пиктограмму в статусной строке, появится описание действия, которое связано с этой пиктограммой.

После загрузки файла из базы данных под пиктографической панелью появляется имя этого файла вместе со списком названий полей содержащихся в нем данных. Большое поле в нижней части окна отводится для размещения данных из базы. Линейки прокрутки, расположенные в правой части окна, позволяют перемещаться внутри набора данных.

В нижней части окна слева находится поле с номером записи данных, которая в данный момент обрабатывается. Стрелки рядом с этим полем позволяют переходить к следующей или предыдущей записи. С помощью стрелок с вертикальными штрихами можно перейти в начало или в конец набора данных. В статусную строку, расположенную в нижней части окна, выводится информация о ходе выполнения команд и состояния программы.

Порядок выполнения работы

1. Добавить в документ рабочие листы всех типов.
2. Создать таблицу произвольной тематики.
3. Скорректировать ширину строк и столбцов по тексту.
4. Промаркировать столбцы и строки, установить формат процентов в одной из ячеек.
5. Ввести в ячейку таблицы формулу (используя редактор формул), использующую все возможные операции и включающую в себя ссылки на ячейки, находящиеся в таблице другого файла.
6. Произвести несколько математических операций с ячейками, объединенными одним групповым именем.
7. Для таблицы данных построить диаграмму, используя графические вставки (дополнительные объекты)
8. Создать в Excel базу данных произвольной тематики, не менее пяти столбцов и десяти строк.

Вопросы:

1. Показать, где расположены директивы добавления в программе Excel, перечислить их.
2. Что позволяет делать функция формирования чисел?
3. Как ввести ссылку на ячейку, а затем обратиться к ней?
4. Для чего используются групповые имена?
5. Что позволяют делать списки?

ЛАБОРАТОРНАЯ РАБОТА №3

Тема: Графический редактор Adobe Photoshop 4.0.

Цель работы: Ознакомление с графическим редактором Adobe Photoshop 4.0 и работе в нем.

Введение

Что же находится за кулисами Photoshop? Пиксели. Каждое изображение в Photoshop покрыто сплошной сеткой мельчайших ячеек в форме квадратиков — так называемых пикселей. Когда вы рисуете, ретушируете, вырезаете, вставляете фрагмент или каким-либо иным образом изменяете изображение в Photoshop, вы просто-напросто изменяете пиксели изображения. Пиксель представляет собой наименьший элемент изображения — по сути, слово пиксель было создано из комбинации двух слов picture (изображение) и element (элемент). В каждом квадратном дюйме изображения, созданного с разрешением 72 пикселя на дюйм, насчитывается 5184 пикселя (72 пикселя на дюйм по горизонтали x 72 пикселя на дюйм по вертикали = 5184). В общем случае, чем больше пикселей помещается на одном квадратном дюйме, тем четче выглядит изображение и тем плавнее будут переходы между цветами при выводе изображения на печать.

Но в Photoshop все далеко не так просто. Работая в Photoshop, вы должны мыслить как художник. Когда вы чертите линию, рисуете фигуру или вводите текст, вы заполняете пиксели — наносите краску на электронный холст. Как только созданный объект или текст появляется на экране, он сразу же "высыхает" на холсте, как будто приклеивается к этому месту экрана. Если вы захотите переместить этот объект, то уже придется воспользоваться весьма точным инструментом, который как бы вырежет его из картинного полотна. Только после этого вы сможете переместить объект в другое место. Чтобы удалить этот объект, вам придется закрасить его пиксели — обычно белым цветом фона.

1. Обзор Photoshop

После загрузки Photoshop на экране появятся: строка меню, палитра инструментов, три группы палитр (Navigator, Info, Options — Навигатор, Инфо, Параметры; Color, Swatches, Brushes — Цвет, Каталог, Кисти;

Layers, Channels, Paths — Слои, Каналы, Контуры), а также палитра Actions (Действия). В отличие от многих программ, которые автоматически открывают новый документ, Photoshop этого не делает. О том, как создать новый файл, вы узнаете немного позже в разделе "Создание нового файла" данной главы.

В строке меню находятся имена девяти раскрывающихся меню (с учетом меню Help). Чем лучше вы узнаете, в каком месте спрятаны те или иные возможности Photoshop, тем меньше вероятность заблудиться в дебрях этой программы.

1.1. Меню Photoshop

Меню Edit (Редактирование) в основном используется для копирования или перемещения фрагментов изображения в другие области рисунка или другие файлы. В этом меню также присутствует команда Undo (Отменить), позволяющая отменить результат последней выполненной команды. Пользователи Windows, конечно же, узнают стандартные команды из меню Edit: Undo (Отменить), Cut (Вырезать), Copy (Скопировать) и Paste (Вклеить). Это меню программы Photoshop позволит вставить изображение в выбранную область или определить свой собственный шаблон, с помощью которого можно "закрашивать" изображения. Команда Take Snapshot (Сделать снимок) позволяет сохранить копию файла в памяти компьютера, чтобы к ней можно вернуться, если вас не устраивают внесенные изменения.

Меню Image (Изображение) позволяет преобразовать цветное изображение в полутоновое, а затем полутоновое — в черно-белое. Кроме того, оно содержит команды, преобразующие файл из RGB- в CMYK-режим, а также позволяет преобразовать изображение в режим Index Color (Индексированные цвета) — промежуточный этап на пути к размещению вашей продукции в Web или программах мультимедиа.

Меню Layer (Слой) содержит команды, позволяющие создавать слои и оперировать ими. (Слой — это нечто, весьма напоминающее прозрачный лист из пластика, находящийся над плоскостью вашего изображения.)

С помощью команд из **меню Select** (Выделение) вы сможете изменять выделенную область или полностью выделить все изображение. (В Photoshop, прежде чем выполнить какое-либо действие с определенным фрагментом изображения, чаще всего вам придется изолировать или "выделить" его.) Команда Select/Grow (Выделение/Соседние пиксели) расширяет выделенную область. Select/ Inverse (Выделение/Инверсия) — инвертирует выделение выбранной области, т.е. выделяет все, кроме выделенной в данный момент области. Команды подменю Select/ Modify (Выделение / Модификация) позволяют так изменить выделенную область, чтобы она плотно соприкасалась с предыдущей выделенной областью, или, например, скруглить контур выделенной области, а также позволяет сжать или растянуть выделенную область. Команда Select/Feather (Выделение/Растушевка) сделает границы выбранной области размытыми.

Меню Filter (Фильтр) содержит команды, действие которых весьма напоминает действие фотофильтров, которые располагаются перед линзами объектива и создают специальные эффекты. Применяя фильтры из разных подменю данного меню к изображениям или их определенным областям, вы можете добиться увеличения резкости, размытия, искривления, стилизации и многих других эффектов. Photoshop содержит более 50 различных фильтров.

Благодаря командам **меню View** (Вид) вы сможете изменить вид вашего документа (увеличить или уменьшить масштаб отображения, увеличить мас-

штаб изображения до размеров всего экрана). С помощью команд этого меню вам будет предоставлена возможность создать новое окно, чтобы просмотреть одно и то же изображение в различных масштабах; причем, когда над одним из изображений выполняется определенная операция, одновременно обновляются оба окна.

Меню Window (Окно) позволяет перемещаться от одного открытого документа к другому. Кроме того, оно позволит вам открывать и закрывать различные палитры Photoshop. О меню Window мы поговорим немного позже в этой же главе.

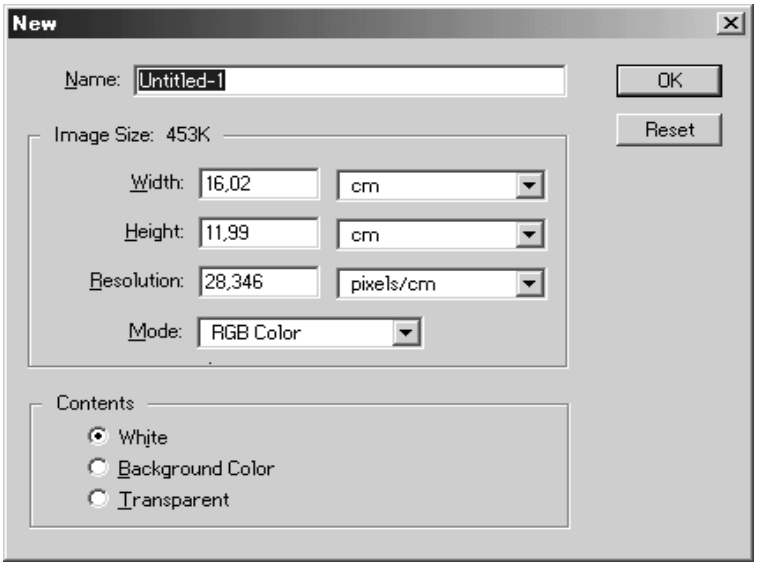
Используя **меню Help** (Справка), вы сможете быстро получить информацию о возможностях и командах Photoshop. Во многих отношениях страницы справочной системы очень напоминают руководство пользователя по Photoshop.

1.2. Создание нового файла

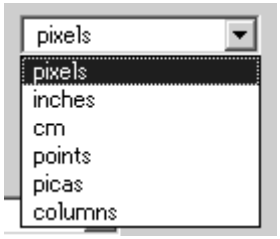
В этом разделе вы самостоятельно создадите файл, что поможет вам ближе познакомиться и научиться пользоваться окном документа Photoshop, а также даст возможность приступить к использованию инструментов из палитры инструментов. Если на вашем экране нет открытого документа, вы не сможете воспользоваться палитрами Photoshop или ее инструментами. Давайте создадим документ шириной 18 см и высотой 12 см, рабочая область которого весьма удобна для работы на многих мониторах.

1. Чтобы создать новый документ, выберите команду New (Новый) из меню File (Файл). Появится диалоговое окно New (Новый). Параметры этого диалогового окна могут быть установлены программой по умолчанию или совпадать с установленными при последнем обращении к этому окну. Работая с Photoshop, вы заметите, что программа часто запоминает параметры диалоговых окон, которые использовались при последнем обращении к ним.

2. В поле Name (Имя) можно ввести заголовок окна вашего документа еще до того, как вы сохраните его на диске. Присваивание имени документу до его сохранения на диске очень полезно, если вы работаете с несколькими несохраненными документами. В данном случае оставьте документ без имени (Unfilled-1).



3. В полях Width (Ширина) и Height (Высота) установите единицы измерения в сантиметры, выбрав cm (см) в раскрывающихся списках, расположенных справа от этих полей. Раскрывающийся список — слегка оттененное по контуру окно, содержащее кнопку со стрелкой, направленной вниз. Раскройте такой список, щелкнув мышью на кнопке со стрелкой вниз и затем выберите cm из списка.



4. Если в поле Width (Ширина) не установлено значение 18, а в поле Height (Высота) не установлено 12, введите эти значения. Щелкните мышью в поле Width и введите 18. Нажмите клавишу <Tab>, чтобы переместиться в поле Height, и введите 12. (Заметьте, что перемещаться от одного поля к другому внутри диалогового окна вы можете, щелкая мышью в соответствующем поле, а

также с помощью клавиши <TaЬ>.) Если вы допустили ошибку, нажмите клавишу <Backspace>, а затем опять введите требуемое значение.

Обратите внимание, что в поле Resolution (Разрешение) установлено значение 72 пикселя на дюйм (ppi). Photoshop подставляет 72 ppi как значение по умолчанию, поскольку многие мониторы отображают по 72 пикселя на каждый дюйм экрана. Другими словами, вы установили разрешение вашего документа, равное разрешению вашего монитора.

5. Если это необходимо, измените значение параметра Resolution на 72. Для этого нажмите клавишу <Tab>, чтобы переместиться на поле Resolution, а затем наберите на клавиатуре 72. Если в полях Resolution, Width и Height вы установите большие значения, соответственно возрастет и размер вашего файла. При выполнении практических упражнений постарайтесь избегать больших размеров изображений; они весьма громоздки для обработки и замедляют работу вашего компьютера.

Значение параметра Mode (Режим) должно быть установлено в RGB Color (Red, Green, Blue — красный, зеленый, синий). Photoshop использует цветовой режим RGB по умолчанию, как стандартный режим видеомониторов для отображения цветов. В режиме RGB все возможные цвета создаются из различных комбинаций красного, зеленого и синего цветов. Когда в поле Mode установлено значение RGB Color, используются все возможности Photoshop для рисования и редактирования изображений. Остальными значениями параметра Mode являются Bitmap (Битовый), Grayscale (Градации серого), CMYK и Lab.

6. Если это необходимо, измените значение в поле Mode на значение RGB Color, щелкнув на раскрывающемся списке Mode и выбрав из списка значение RGB Color.

7. Чтобы сделать фон вашего нового документа белым, установите переключатель Contents (Содержимое) в положение White (Белый цвет). Если вы вместо этого выберете положение Background Color (Цвет заднего плана), то цвет фона нового документа будет соответствовать цвету фона, установленному

в Photoshop непосредственно перед выполнением данной операции. Если вы выберете положение Transparent (Прозрачная основа), вы будете работать в слое с прозрачным фоном, без какого-либо цвета. Шаблон разметочной сетки представляет собой прозрачную область. Обратите внимание: вы можете изменить внешний вид прозрачного фона, если измените параметры Transparency (Прозрачность) в диалоговом окне Transparency & Gamut (Прозрачность & Гамма). Чтобы попасть в это диалоговое окно, выберите команду File->Preferences->Transparency & Gamma (Файл->Установки->Прозрачность & Гамма).

8. Чтобы закрыть диалоговое окно New (Новый), щелкните на кнопке ОК.

На экране появится окно для нового неименованного (Untitled) документа. Если вам понадобится увеличить масштаб отображения, выберите из меню Window команду Zoom In (Увеличить) или нажмите на клавиатуре комбинацию клавиш <Ctrl++(runoc)>. Чтобы уменьшить масштаб, выберите команду Zoom Out (Уменьшить) или нажмите комбинацию клавиш <Ctrl+— (минус) >. Вы можете нажимать клавиши <+> и <—> как на обычной клавиатуре, так и на дополнительной цифровой клавиатуре (включив ее нажатием клавиши <Num Lock>).

2. Палитра инструментов Photoshop

Вдоль левой стороны окна Photoshop располагается палитра инструментов (toolbox). Эта палитра содержит инструменты, необходимые для рисования, выбора и редактирования графических изображений. Каждый инструмент в палитре представлен в виде соответствующей пиктограммы. Знание назначения и возможностей инструментов дает ключ к изучению Photoshop. Как плотник знает назначение каждого своего инструмента, так и пользователь Photoshop должен знать, где и как применяется данный инструмент.

В палитре инструментов Photoshop содержатся 22 инструмента, каждый из которых, выполняя свою специфическую функцию, поможет вам при создании,

редактировании и корректировке цветов изображений. Когда вы загрузите программу Photoshop в первый раз после ее установки, в палитре по умолчанию выбран инструмент Marquee (Область). Впоследствии, всякий раз при загрузке Photoshop или создании нового файла, будет выбираться тот инструмент, с которым вы закончили работать в предыдущем сеансе. При выборе какого-либо инструмента его кнопка в палитре становится белой. Когда вы выбираете инструмент, в палитре Options (Параметры) появляется список всех возможностей, присущих этому инструменту

2.1. Инструменты выделения

Первый инструмент в первом ряду и оба инструмента второго ряда палитры инструментов называются инструментами выделения (selection tools). Если вы желаете изменить свойства какого-либо объекта, то его прежде всего выделяют среди остальных объектов с помощью одного из этих инструментов. В верхнем левом углу палитры находится инструмент Marquee (Область), который позволит вам выделять области различной формы.

Если вы выберете пункт **Elliptical** (Эллипс), на кнопке инструмента Marquee появится небольшая окружность. Если вы выберете **Single Row** (Горизонтальная строка) или **Single Column** (Вертикальная строка), пиктограмма инструмента Marquee превратится в тонкую вертикальную или горизонтальную линию.

Под инструментом Marquee находится инструмент **Lasso** (Лассо), который может использоваться как "ручной" инструмент; им пользуются для выделения области неправильной формы. Инструмент Lasso можно активизировать, нажав на клавиатуре клавишу <L>. Под кнопкой Lasso "спрятаны" два варианта этого инструмента: инструмент Lasso и инструмент **Polygon Lasso** (Многоугольное лассо). С помощью варианта Polygon Lasso вы, щелкая в разных точках экрана, сможете выделить область в форме многоугольника. С каждым щелчком Pho-

toshop увеличивает выбранную область, проводя мигающую пунктирную линию от одной до следующей указанной вами точки. Чтобы замкнуть многоугольную область, щелкните еще раз на самой первой точке или дважды щелкните в любом месте окна изображения.

Инструмент **Magic Wand** (Волшебная палочка) выделяет области на основе подобия цветов. Этот инструмент будет полезен, когда вы захотите выделить область, отличающуюся цветом от других областей вашего изображения. Например, с помощью инструмента Magic Wand можно выделить из всего пейзажа только реку.

2.2. Инструменты рисования

Инструменты **Paint Bucket** (Заливка), **Line** (Линия), **Paintbrush** (Кисть) и **Airbrush** (Аэрограф) используются для рисования основным цветом. Инструмент **Eraser** (Ластик) рисует цветом фона и служит для стирания частей изображения, в этом случае фон может просвечивать сквозь ваше изображение. Инструмент **Pencil** (Карандаш) имитирует рисование карандашом как основным цветом, так и цветом фона. Инструмент **Gradient** (Градиент) позволяет создавать цветовые переходы между основными и фоновыми цветами, а также переходы от прозрачного фонового цвета к основному и наоборот. С помощью инструмента **Eyedropper** (Пипетка) вы сможете выбрать определенный цвет непосредственно из вашего изображения, а также изменить цвета переднего плана и фона (заднего плана) в соответствии с цветом, выбранным с помощью инструмента Eyedropper. Ниже приведены клавиши для активизации различных инструментов рисования.

Инструмент	*Клавиша*
Paint Bucket (Заливка)	<K>
Gradient (Градиент)	<G>
Line (Линия)	<N>
Eraser (Ластик)	<E>
Pencil (Карандаш)	<Y>
Airbrush (Аэрограф)	<A>
Paintbrush (Кисть)	<B>
Eyedropper (Пипетка)	<I>

К инструментам редактирования принадлежат Rubber Stamp (Штамп), Smudge (Палец), Blur (Размытие), Sharpen (Резкость), Dodge (Осветлитель), Burn (Затемнитель) и Sponge (Губка).

Rubber Stamp (Штамп) — это инструмент для клонирования. Вы можете им воспользоваться для выбора некоторой области и клонирования (копирования) ее в другое место, пиксель за пикселем, и всего за несколько щелчков и перемещений мыши. Этот инструмент очень часто используется для ретуширования изображений или создания специальных эффектов. Чтобы выбрать инструмент Rubber Stamp, нажмите клавишу <S>.

Инструмент **Smudge** (Палец) позволит вам создать эффект использования акварельных красок. Он "размазывает" цвета, создавая иллюзию изображения, нарисованного акварелью. Чтобы активизировать инструмент Smudge, нажмите клавишу <U>.

Blur (Размытие) и **Sharpen** (Резкость) — эти два инструмента помещаются в одном и том же месте палитры инструментов; рисунок на кнопке палитры инструментов изменяется в соответствии с тем, какой из двух инструментов выбран.

Инструменты **Dodge** (Осветлитель), **Burn** (Затемнитель) и **Sponge** (Губка) могут изменять в изображении цвета и/или оттенки серого. Точно так же как инструменты Blur (Размытие) и Sharpen (Резкость), инструменты Dodge, Burn и Sponge расположены в одном и том же месте палитры инструментов.

2.3. Инструменты создания и редактирования контуров

Инструмент **Pen** (Перо) позволит вам создавать контуры (paths). Контуры обычно используются для создания масок. Маска может применяться для образования электронного трафарета на экране, чтобы предохранить участки изображения от случайных изменений, когда вы редактируете незащищенные области.

С помощью инструмента Pen можно активизировать все инструменты, предназначенные для работы с контурами. К ним относятся Direct Selection (Прямой выбор). Add Anchor Point (Добавить точку привязки), Delete Anchor Point (Удалить точку привязки) и Convert Anchor Point (Преобразовать точку привязки). Инструмент Direct Selection (на кнопке нарисована стрелка) позволит вам выделять контуры для их дальнейшего перемещения и редактирования. С помощью инструментов Add Anchor Point и Delete Anchor Point вы сможете добавлять или удалять точки привязки из контуров. Инструмент с рисунком уголка — **Convert Anchor Point** — предназначен для того, чтобы преобразовать криволинейные контуры в прямолинейные и наоборот. Если вы хотите переключиться с инструмента Pen на инструменты редактирования контуров, нажмите клавишу <Alt> и щелкните на кнопке Pen в палитре инструментов. Кроме того, для переключения между инструментами этой группы вы можете воспользоваться клавишей <P> на клавиатуре или просто щелкнуть в палитре инструментов на активном инструменте данной группы и перетащил указатель мыши с одного инструмента на другой.

2.4. Значки управления цветами

Значки **Foreground Color** (Цвет переднего плана) и **Background Color** (Цвет заднего плана) отображают текущие значения основного и фонового цветов. Если вы щелкнете на каком-либо из них, появится диалоговое окно Color

Picker (Палитра цветов), в котором вы сможете изменить значения для основного или фонового цветов.

Если вы щелкнете на значке **Switching Colors** (Переключение цветов), цвет значка Foreground Color изменится на цвет значка Background Color, и наоборот. Вы также можете нажать клавишу <X>, и тогда основной и фоновый цвета поменяются местами.

Значок **Default Color** (Цвет по умолчанию) восстанавливает значения, принятые по умолчанию лля (Фонового (белый) и основного (черный) цветов

2.5. Кнопки управления структурой окна

Три кнопки управления структурой окна, находящиеся в самом низу палитры инструментов, предназначены для изменения режима отображения окна на экране. Кнопка слева представляет обычный режим окна. Щелчок на средней кнопке приведет к тому, что окно займет весь экран. Правая кнопка также развернет окно на весь экран, кроме этого исчезнут абсолютно все меню. Чтобы быстро перейти из одного режима к другому, нажмите <F> на клавиатуре.

Пример использования:

"Кровавый полдень в кровавом ущелье". Техника следующая - создаем красно-черные облака, доделываем их повторением Difference Clouds до получения достаточно мелкой фактуры, далее применяем Filters/Distorb/Twirl градусов на 50 и фильтр Pinch порядк а 70. Следующий - Filters/Distorb/Polar Coordinates и Polar to Rectangural. Немного притемняем изображение путем уменьшения Transparency слоя (Layer), предварительно подложив под слой с облаками чисто черный слой. Дальше надо придать лишь

завершаюший шарм - солнце. Для этого сначала Filters/Render/Lighting Effects создаем Omni источник на месте предполагаемого светила, так чтобы это место слегка осветилось, а другие районы слегка затемнились. Потом Lense Flare, сначала 105 мм интенсивностью 50, и поверх 35 мм на 60 единицах.

Контрольные вопросы:

1. Что такое слои?
2. Назовите основные инструменты для ретуширования изображений?
3. Можно ли в Photoshop редактировать векторные изображения?
4. Для чего необходим режим «быстрой маски»?

Задание:

1. Создайте новый файл, в нем нарисуйте некий рисунок с применением многослойной структуры.
2. Примените несколько графических фильтров для обеспечения реалистичности придуманного вами изображения.
3. Откройте уже готовый рисунок и испробуйте средства по изменению и ретушеровке изображения.

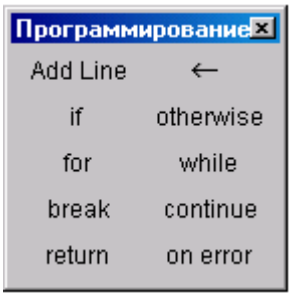
ЛАБОРАТОРНАЯ РАБОТА № 4

Тема: Математический пакет Mathcad 8 Pro.

Цель: Показать возможности программирования с помощью математического пакета Mathcad 8 Pro.

1. Панель программирования

В Mathcad, по сути, не встроен язык программирования, а просто снято ограничение на использование составных операторов в теле алгоритмических управляющих конструкций выбора (if) и повторения (until). Кроме того, введено понятие локальной переменной, добавлен цикл с параметром for, операторы досрочного выхода из цикла break и continue, а также оператор досрочного выхода из программы return. Алгоритмические конструкции в среде Mathcad вводятся



не традиционным набором через клавиатуру ключевых слов if, then, else, while и т.д., а нажатием одной из кнопок панели программирования (Programming):

Рис. 4.1 - Панель программирования Mathcad

Щелчок по одной из этих кнопок создает на дисплее заготовку соответствующей программной конструкции.

Опишем их.

Кнопка

– эта команда добавления строки в программу, в тело

цикла, в плечо альтернативы и т.д. Этим действием снимается вышеупомянутое ограничение на число операторов во вложенных конструкциях языка:

Было $a := \blacksquare$

стало $a := \left|\begin{matrix}\blacksquare\\ \blacksquare\end{matrix}\right.$

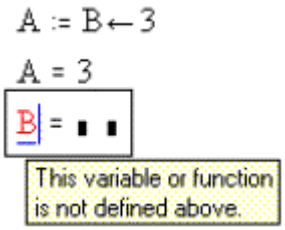
Вертикальная линия объединяет отдельные операторы в операторный блок с одним входом и одним выходом, который выполняется как единый оператор (один из трех атрибутов структурного программирования).

Кнопка [←] – это оператор присвоения значения локальной переменной. На языке Pascal мы пишем A:= B + C, на языке BASIC – A = B + C, а на языке Mathcad – A ← B + C. Почему?

A:= A:= B + C (Pascal),

В Mathcad-выражении: A:= A ← B + C все более-менее ясно: локальной переменной A (она в середине между символами «:=» и «←») присваивается значение суммы двух переменных B и C, значение которых уже задано выше в Mathcad-документе (глобальные переменные). Затем эта сумма передается глобальной переменной A (она слева от знака «=»).

Простейший пример, показывающий разницу между глобальной и локаль-



ной переменной:

Негативное изображение переменной B будет свидетельствовать о том, что ее значение вне программы (B ← 3) неопределенно. Благодаря локальным пе-

ременным можно создавать объемные Mathcad-документы, поручая разработку отдельных функций и операторов разным программистам и не заботясь о разделении переменных: в разных программах переменные могут совпадать по имени, но при этом они не будут перебегать дорогу друг другу (технология программирования «сверху вниз»).

Итак, локальная переменная распространяет свое действие только на программу, а глобальная – на весь документ (на низ документа). Но в среде Mathcad есть инструментарий, позволяющий переменным, пользовательским функциям и операторам проникать и в другие документы, но с их, так сказать, согласия.

Нажав на кнопку while мы получим на экране заготовку цикла с пред-

while ▪

▪

проверкой – слово while с двумя пустыми квадратиками:

В первый квадратик (правее while) нужно будет записать булево выражение (переменную), управляющее циклом, а во второй (ниже while) – тело цикла, операторы которого будут выполняться, пока булево выражение возвращает значение «Да» (в среде Mathcad – это числовое значение, отличное от нуля). Если в теле цикла более одного оператора (а это основное отличие оператора while от вышеупомянутой функции until), то нужно воспользоваться кнопкой Add Line (см. выше).

Кнопка if позволяет вводить в программу альтернативу с одним плечом. Так, Pascal-конструкция: if A > B then C:= D в среде Mathcad будет выглядеть несколько по-арабски (записана справа налево): C ← D if A > B.

Но если плечо альтернативы – составной оператор, то все встанет на свои места, вернее, будет записано сверху вниз:

Pascal: if A>B then begin E:=F; F:=G end[11];

Mathcad: if A>B

E←F

F←G

Кнопка otherwise - превращает неполную альтернативу в полную.

Pascal: if A > B then C := D else E:= F;

Mathcad: C ←D if A > B

E ←F otherwise

Но если в плечах полной альтернативы по одному оператору, то можно воспользоваться не оператором (кнопкой) if, а функцией if:

C←if(A > B, D, F) или if(A > B, C←D, E←F)

Понять, почему в Mathcad не было использовано традиционное слово else, можно, если принять во внимание то, что операторы if и otherwise позволяют записать в программах алгоритмическую конструкцию множественное ветвление. Разберем ее на примере задачи о расчете налогов (федеральный налог США с недельного заработка).

Функции Tax1 и Tax2 (пункт 1) возвращают налог с холостых и женатых по прогрессивной шкале налогообложения (см. график в пункте 3). В данном примере (и во всех других) без оператора otherwise можно обойтись (сравните окончания функций Tax1 и Tax2). Он необходим в тех случаях, когда булево выражение, объединяющее оставшиеся случаи ветвления, трудно сформировать. Оператор otherwise – это гибрид ключевых слов ELSE, ELSEIF и CASE ELSE языка BASIC.

Когда заранее известно, сколько раз нужно выполнить какую-то часть программы (тело цикла), то используют не цикл while, а цикл for, в заголовке которого пишут не булево выражение, а параметр цикла и указывают, какие дискретные значения он должен принимать в цикле. Эти значения можно перечислить через запятую (1, 2, 3.7) или указать диапазоном (2.. 100) или вектором (V).

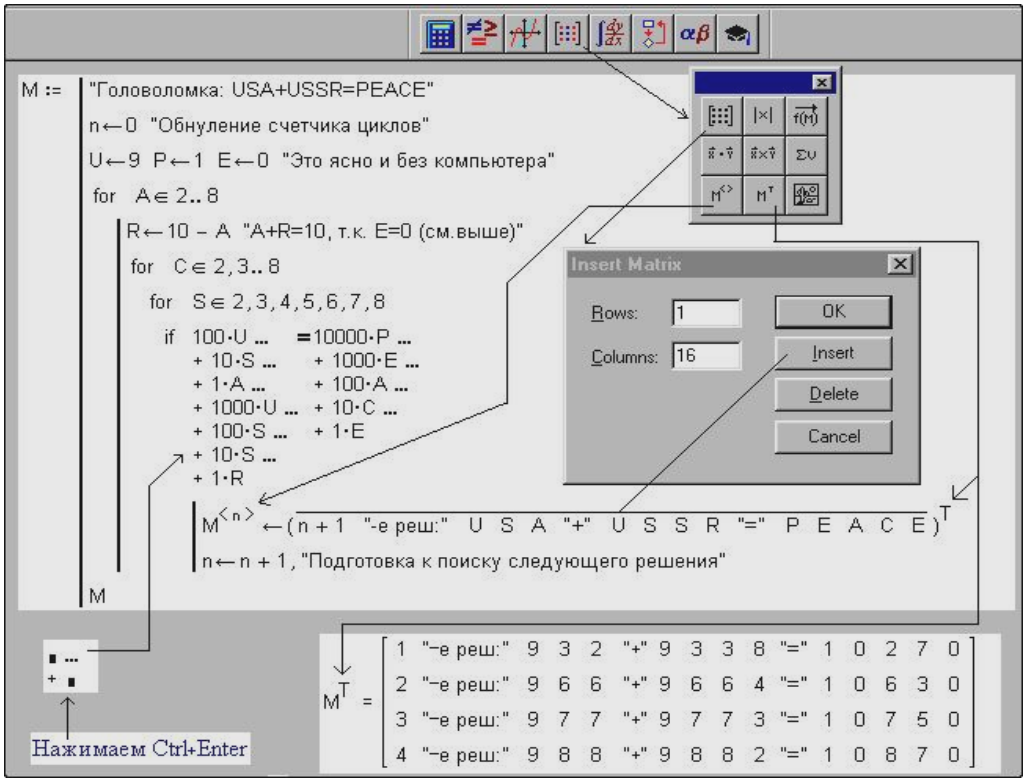


Рис. 4.2 - Решение буквенной головоломки USA+USSR=PEACE (иллюстрация цикла с параметром)

Программа на рис. 4.2 решает буквенную головоломку USA+USSR=PEACE, где требуется указать, какие цифры скрываются за буквами. В программе три цикла с параметром (A, C и S), которые вложены друг в друга. В программе, не мудрствуя лукаво, можно было записать все семь циклов – по числу неизвестных задачи U, S, A, R, P, E и C. Но тогда перебор длился бы нестерпимо долго. Несложный предварительный анализ условий задачи (U ← 9, P ← 1, E ← 0 и R ← 10 + A) сокращает число циклов до трех и делает время счета приемлемым. Один из основных недостатков языка Mathcad – это невозможность вывода на дисплей промежуточных результатов расчета. А они не только помогают отлаживать программы, но и в ряде случаев просто необходимы при поиске единственно правильного решения из множества возможных. В Mathcad-программах допустима запись вариантов ответов (промежуточных результатов) в матрицу (в вектор), которую после выполнения программы можно

просмотреть, что и сделано на рис. 4.2. При этом в матрицу M записываются не только значения числовых переменных, но и текстовые константы (“+”, “=” и др.), делающие ответ более читабельным. Правильный ответ хранится в первом столбце матрицы M. Вернее, в первой строке – матрица у нас транспонируется для большей компактности. Остальные ответы неверны – там разным буквам соответствуют одинаковые числа.

Цикл с параметром в среде Mathcad более гибок, чем его аналоги в языках BASIC или Pascal. Вот еще варианты заголовков циклов с параметром в среде Mathcad, кроме тех, которые показаны на рис. 4.2 (там специально приведены разные варианты заголовков циклов):

for A $\in$ V (V – вектор)

for A $\in$ 5, 4.7, 8.9, 7.3×10-5

for i $\in$i1.. i2.

Последний вариант примечателен тем, что переменные i1 и i2 могут принимать любые значения, и необязательно, чтобы i1 было меньше i2. В языках BASIC и Pascal разное соотношение между i1 и i2 требует различного синтаксиса заголовка цикла с параметром:

For i=i1 To i2 или For i=i2 To i1 Step -1 (BASIC)

for i:=i1 to i2 do или for i:=i2 downto i1do (Pascal).

Кнопки continue и позволя- break ют досрочно выходить из циклов while и for, а кнопка return. – совсем из программы.

Ниже приведены другие примеры программ в среде Mathcad.

2. Рекурсия

Рекурсивная функция – это такая функция, которая вызывает сама себя в момент ее создания. Рекурсия – очень мощный и удобный инструмент решения задач. Так, например, один из самых быстрых алгоритмов сортировки массива использует рекурсию. Вычислить детерминант квадратной матрицы также помогает рекурсия и т.д.

Числа Фибоначчи, которые связаны с условными кроликами:

Поколение кроликов	...	4	5	6	7	8	9	10	11	...
Число кроликов	...	3	5	8	13	21	34	55	89	

Приведенный ряд специально начат не с традиционного места (первое поколение), а с четвертого поколения (три кролика), для того чтобы задать читателю вопрос, подобный тому, который стоял в задаче о факториале: «Чему равно минимальное число кроликов в популяции – каково наименьшее число Фибоначчи?» Нормальный ответ, приводимый во всех учебниках, – ноль. Но не будем спешить и напишем программу с двусторонней рекурсией, взяв за базовые числа Фибоначчи не традиционную пару 0 и 1, а 13 и 21.

```
fibo(n) := | "Числа Фибоначчи"
           | return fibo(n + 2) - fibo(n + 1)  if  n < 7
           | return 13  if  n = 7
           | return 21  if  n = 8
           | return fibo(n - 1) + fibo(n - 2)  if  n > 8
```

n := -7.. 7

n =	fibo(n) =
-7	13
-6	-8
-5	5
-4	-3
-3	2
-2	-1
-1	1
0	0
1	1
2	1
3	2
4	3
5	5
6	8
7	13

Рис. 4.3 - Расчет чисел Фибоначчи (двусторонняя рекурсия)

Ряд кроликов Фибоначчи в «отрицательных поколениях» зеркально отображает значения в «положительных поколениях», но с переменным знаком.

Числа Фибоначчи в наше время широко применяются в вычислительной

```
fine_Fibo(n) := | "Прекрасные числа Фибоначчи"
                | f1 ← 3 f2 ← 7 f3 ← 11
                | return f1  if  n = 1
                | return f2  if  n = 2
                | return f3  if  n = 3
                |          return          if  n < 1
                |  | for  i ∈ n.. 0
                |  |  | f0 ← f3 - f2 - f1
                |  |  | f3 ← f2 f2 ← f1 f1 ← f0
                |  | f0
                |          return          if  n > 3
                |  | for  i ∈ 4.. n
                |  |  | f4 ← f1 + f2 + f3
                |  |  | f1 ← f2 f2 ← f3 f3 ← f4
                |  | f4
```

Так автор пишет программу для ее компактности. Читателю нужно писать так.

```
| f1 ← 3
| f2 ← 7
| f3 ← 11
```

В последней строке блока нужно ставить только один оператор!

n := -77.. 33

n =	fine_Fibo(n) =	n =	fine_Fibo(n) =	n =	fine_Fibo(n) =
-77	-10119510357	-7	11	15	17155
-76	16625830165	-6	-1	16	31553
-75	-8451972457	-5	-5	17	58035
-74	-1945652649	-4	5	18	106743
-73	6228205059	-3	-1	19	196331
-72	-4169420047	-2	-1	20	361109
-71	113132363	-1	3	21	664183
-70	2171917375	0	1	22	1221623
-69	-1884370309	1	3	23	2246915
-68	400679429	2	7	24	4132721
-67	688226495	3	11	25	7601259
-66	-795464385	4	21	26	13980895
-65	293441539	5	39	27	25714875
-64	186203649	6	71	28	47297029
-63	-315819197	7	131	29	86992799
-62	163825991	8	241	30	160004703
-61	34210443	9	443	31	294294531
-60	-117782763	10	815	32	541292033
-59	80253671	11	1499	33	995591267

математике, в том числе и для иллюстрации рекурсии.

Рис. 4.4 - Расчет изящных чисел Фибоначчи (двусторонняя рекуррентность)

Использование рекурсии для поиска чисел Фибоначчи – это стрельба из пушки по воробьям. Намного эффективнее рассчитывать подобные числа в цикле, рекуррентно. На рис. 4.4 представлена программа, по которой ищутся, если так можно выразиться, изящные (fine) числа Фибоначчи.

3. Об использовании операторов break continue return

Попробуем на простых примерах показать, что ввод в язык Mathcad операторов break, continue и return (см. выше) – только полшага в сторону повышения

```
Def Fny(x) = x ^ 2 - 3
Def Dny(x) = 2 * x

Input "xbeg"; x
TOL = .001
Do
        x1 = x + Fny(x)/ Dny(x)
  If Abs(x1 – x) <= TOL Then Exit Do
        x = x1
Loop
Print "y = 0 at x ="; x
```

гибкости управляющих конструкций этого языка программирования.

Рис. 4.5 - Метод Ньютона I (BASIC): цикл с выходом из середины

На рис. 4.5 приведена BASIC-программа поиска корня алгебраического уравнения методом Ньютона (касательных). Почему мы начали с языка BASIC, ведь этюд посвящен языку программирования Mathcad? Дело в том, что язык BASIC кроме традиционной тройки циклов (цикл с предпроверкой, цикл с постпроверкой, цикл с параметром) имеет и универсальный цикл с выходом из середины: Do [...] If ... Then [...] Exit Do [...] Loop. Эта конструкция наряду с дру-

гими преимуществами, о которых будет сказано ниже, позволяет реализовывать алгоритмы в их естественной последовательности. Так, в программе на рис. 6.8 объявляются функции пользователя (анализируемое уравнение y и его производная dy), запрашивается значение начального приближения к корню x и задается значение погрешности TOL. После этого организуется цикл, но не традиционный, а с выходом из середины. В цикле, следуя естественному порядку алгоритма Ньютона, рассчитывается новое приближение к корню (x1), и если оно отстает от предыдущего не более чем на величину заданной погрешности, то (Then) задача считается решенной (Exit Do). Если нет, то ведется подготовка к новому приближению (x = x1), а цикл повторяется (Loop).

При реализации на языке Mathcad этот несложный алгоритм обрастает «архитектурными излишествами», так как его приходится реализовывать функцией и «запихивать» в прокрустово ложе цикла while – см. рис. 4.6.

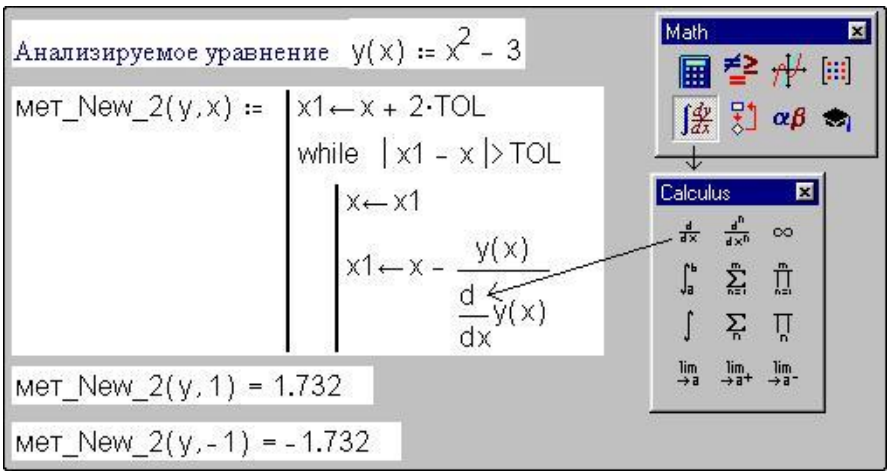


Рис. 4.6 - Метод Ньютона II: цикл с предпроверкой

Цикл с предпроверкой (цикл while) требует, чтобы булево выражение заголовка было определено еще до входа в цикл. А этого нет при поиске корня методом Ньютона. Приходится до входа (и для входа) в цикл писать x1 ←x + 2 × TOL. Подобным образом лгут детям (а машина тоже в каком-то смысле дитя), заставляя их что-то делать. Строку x1 ←x + 2 × TOL можно уподобить стартеру

двигателя внутреннего сгорания, работающего, кстати, как и программа на рис. 4.6, циклически. Вот какими аллегориями (дитя, двигатель) обросла наша простенькая программа из-за того, что в языке Mathcad нет цикла с выходом из середины. На рис. 4.6 можно отметить и другую ненатуральность программы – постановку телеги впереди лошади: в цикле сначала приходится готовиться к новому приближению (x1 ←x), хотя еще не ясно, понадобится оно или нет, а только потом проводить его.

Операторы break, continue и return, введенные в Mathcad, призваны вернуть программе на рис. 4.6 ее естественность, но...

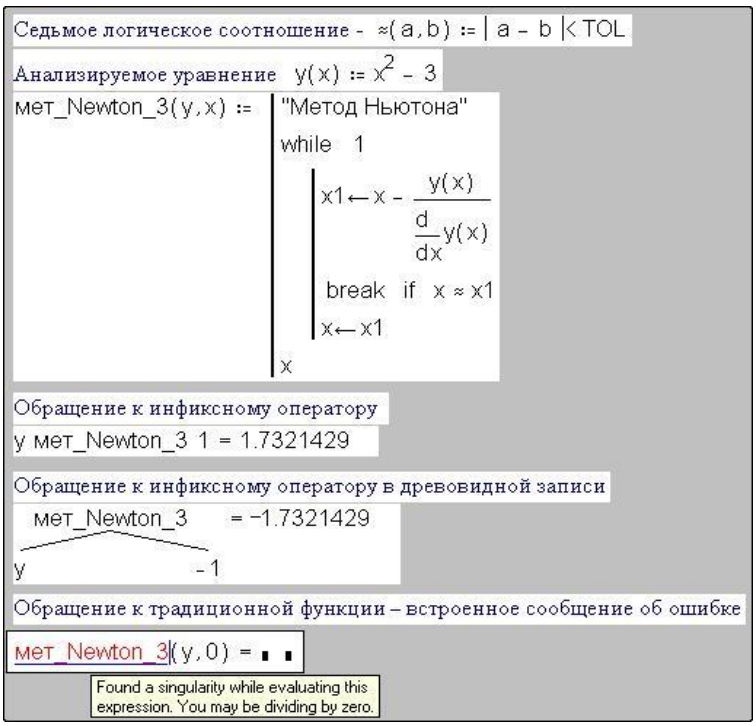


Рис. 4.7 - Метод Ньютона III: имитация цикла с выходом из середины

Оператор continue отличается от оператора break тем, что передает управление не в хвост, а в гриву (начало) цикла.

Остается рассказать о последней кнопке

на панели программирования Mathcad 8 Pro. Нажатие на нее приводит к появлению на дисплее заготовки инфиксного оператора обработки ошибок с двумя операндами:

Суть оператора проста: если при выполнении правого операнда возникнет ошибка, то выполняется левый операнд. Этот оператор позволяет реализовывать в Mathcad-программах метод проб и ошибок.

$$y(x) := 1 \quad \text{on error} \quad \frac{\sin(x)}{x}$$

Испробуем on error, на простом примере:

Но оказалось, что объявленная таким образом функция y(x) возвращает нуль, а не единицу при x = 0. Дело в том, что система Mathcad, оптимизируя произведение, выдает нуль, если первый сомножитель равен нулю. Функцию y(x) нужно переписать по другому:

$$y(x) := 1 \quad \text{on error} \quad \frac{1}{x} \cdot \sin(x)$$

но и эта запись не совсем правомочна. Проще записать:

$$y(x) := \text{if}\left(x = 0, 1, \frac{\sin(x)}{x}\right)$$

Здесь графически решается неравенство: функция f(x) возвращает нуль не только тогда, когда логарифм меньше единицы, но и тогда, когда делается попытка взятия логарифма от неположительного числа. Можно, конечно, решить квадратное уравнение и обойти ошибку функцией или оператором if.

Контрольные вопросы:

1. Знать основные компоненты Математического пакета Mathcad
2. Дать описание всем инструментам, находящимся на панели «Программирование», уметь их практически использовать.

Задание:

1. Задание на закрепление пройденного: сделать программу на рис. 4.3 рекуррентной, а на рис. 4.4 – рекурсивной.

2. Задания 1 выполнить в среде математического пакета Mathcad.

ЛАБОРАТОРНАЯ РАБОТА №5

Тема: Программа сканирования и распознования текстов FineReader.

Цель работы: Ознакомление с принципом работы програм сканирования текста, а также приобретение навыков работы с программой FineReader.

Введение

Все мы вводим тексты в компьютер. Это могут быть газетные вырезки, журнальные статьи, прошлогодние договоры, деловые письма, факсы, прайс - листы, биржевые сводки, анкеты... Подсчитано, что более 90% современных документов поступает к нам на бумаге. Десятилетия люди знали только один способ ввода печатных документов в компьютер - с клавиатуры. Помните часы, которые Вы просиживали за компьютером, перепечатывая текст длиной в вечность? А как бы нам всем хотелось, чтобы компьютер научился читать сам прямо с бумажного листа!

Программа оптического распознавания текстов (по-английски OCR – Optical Character Recognition) FineReader позволит Вашему компьютеру и сканеру самостоятельно читать печатные тексты.

1. Возможности программы ABBYY FineReader

FineReader позволяет ввести документ одним нажатием на кнопку Scan&Read, не вдаваясь в подробности работы программы. Распознанный текст можно передать в текстовый редактор или электронную таблицу, сохранить в форматах PDF и HTML с полным сохранением оформления документа или сохранить в базе данных

1.1. Что такое OCR-системы?

Системы оптического распознавания символов (Optical Character Recognition - OCR) предназначены для автоматического ввода печатных документов в компьютер.

FineReader — омнифонтовая система оптического распознавания текстов. Это означает, что она позволяет распознавать тексты, набранные практически любыми шрифтами, без предварительного обучения. Особенностью программы FineReader является высокая точность распознавания и малая чувствительность к дефектам печати, что достигается благодаря применению технологии "целостного целенаправленного адаптивного распознавания".

Процесс ввода документа в компьютер можно подразделить на два этапа:

1. Сканирование. На первом этапе сканер играет роль «глаза» Вашего компьютера: «просматривает» изображение и передает его компьютеру. При этом полученное изображение является не чем иным, как набором черных, белых или цветных точек, картинкой, которую невозможно отредактировать ни в одном текстовом редакторе.
2. Распознавание. Обработка изображения OCR-системой.

1.2. Новые возможности ABBYY FineReader 5.0

Работа с изображениями

•Новый режим отображения страниц пакета Пиктограммы, в котором страницы пакета представляются своими сильно уменьшенными изображениями.

Сканирование

•Возможность сканирования разворота книги: обе страницы раскрытой книги сканируются одновременно, но обрабатываются по отдельности. Изобра-

жение, содержащее сдвоенные страницы, записывается в две различные страницы пакета.

Анализ макета страницы

•В программу добавлена объединенная процедура анализа макета страницы и распознавания. Теперь алгоритм поиска блоков использует информацию о распознанном тексте для более точного анализа макета страницы. Новые алгоритмы анализа используются автоматически при запуске распознавания на странице без блоков, т.е. анализ макета и распознавание делаются за один шаг. Старый алгоритм анализа страницы, который не использует распознавание, по-прежнему доступен (меню Процесс>Анализ макета документа).

•Поддержка распознавания "встроенных" картинок: возможно отметить блок "картинка" внутри текстового или табличного блока.

Распознавание

•176 языков распознавания. Поддержка армянского алфавита. Полный список языков приведен в разделе Поддерживаемые языки распознавания.

•Распознавание языков программирования (Basic, Cobol, Fortran, Java, C++, Pascal).

•Распознавание подстрочных символов (например, H2O).

•Распознавание вертикального текста.

•Корректная передача цвета текста.

2. Быстрое знакомство

В этой главе Вы узнаете, как можно ввести нужный документ в компьютер, практически ничего не зная о системе FineReader. Также в этой главе приводится обзор основных элементов системы FineReader, таких как окна системы, панели инструментов, строка состояния.

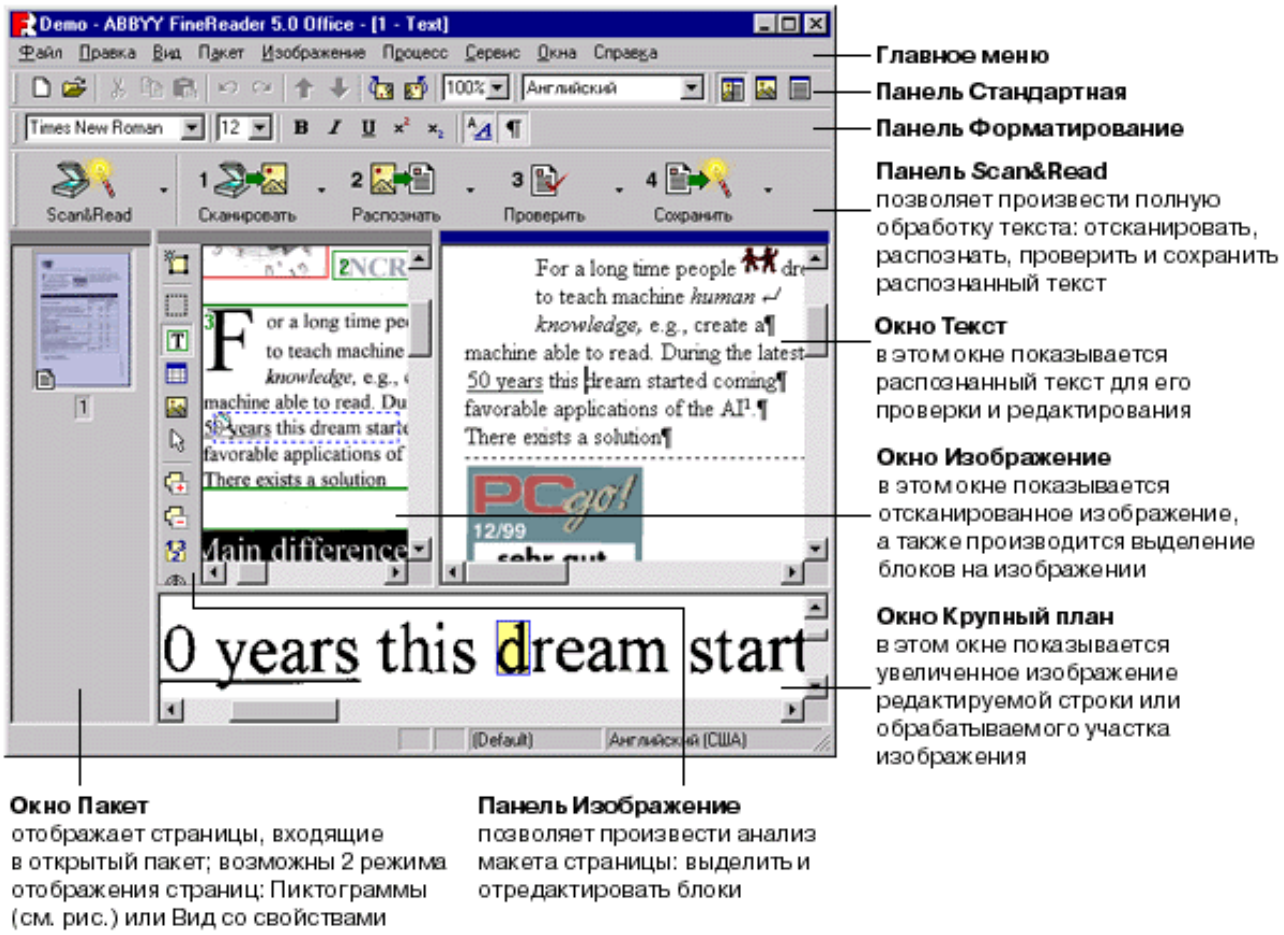
2.1. Как ввести документ за минуту

1. Включите сканер (если он имеет отдельный от компьютера источник питания). Внимание! Многие модели сканера необходимо включать до включения компьютера.
2. Включите компьютер и запустите FineReader (Пуск/Программы/ABBYY FineReader 5.0/ABBYY FineReader 5.0 Pro или Office).Перед Вами откроется Главное окно программы FineReader.
3. Вставьте в сканер страницу, которую Вы хотите распознать.
4. Нажмите на стрелку справа от кнопки Scan&Read, в открывшемся локальном меню выберите пункт Мастер Scan&Read.
5. Далее следуйте указаниям Мастера Scan&Read.

2.2. Главное окно программы FineReader

Вся работа в программе FineReader ведется в пакетном режиме. Пакет - это папка, в которой хранятся изображения и рабочие файлы программы. Каждое отсканированное изображение записывается как отдельная страница пакета.

При первом запуске FineReader на его экране появляется пакет, созданный программой по умолчанию, этот пакет содержит файл с изображением demo.tif. Вы можете продолжить работу с пакетом по умолчанию или создать новый пакет.



Вверху Главного окна FineReader находится меню системы, под ним - инструментальные панели. В программе их четыре: Стандартная, Форматирование, Изображение и Scan&Read.

Внизу окна расположена информационная панель (еще ее называют строкой состояния), где Вы можете получить сведения о состоянии системы и производимых ею операциях, а также краткую справку о выбираемых Вами пунктах меню и кнопках.

Остальное пространство Главного окна занимают по мере своего появления рабочие окна системы: Пакет, Изображение, Крупный план и Текст.

В окне Крупный план по умолчанию показывается черно-белое изображение независимо от того, какое именно (цветное, серое или черно-белое) изображение Вы открывали. Если Ваше изображение цветное и Вы хотите, чтобы в окне Крупный план показываемое изображение также было цветным, снимите

отметку с пункта черно-белая палитра в окне Крупный план в группе Окно Изображение на закладке Вид (меню Сервис>Опции).

2.3. Инструментальные панели программы FineReader

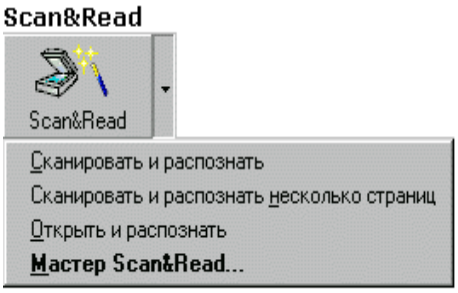
В системе FineReader инструментальных панелей всего 4: Стандартная, Изображение, Форматирование и главная панель программы Scan&Read. Кнопки на инструментальных панелях - самое удобное средство доступа к операциям системы. Те же операции можно выполнять из меню программы или с помощью горячих клавиш.

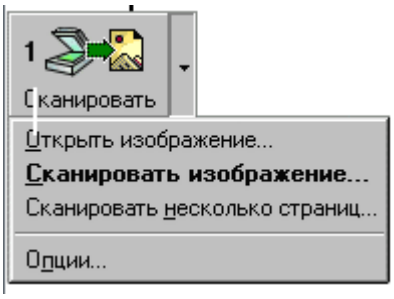
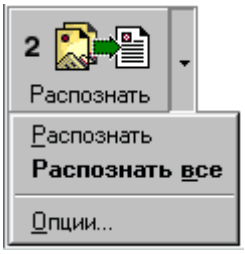
Если Вы хотите узнать о назначении той или иной кнопки на инструментальной панели, установите на ней курсор мыши. Под кнопкой появится подпись (tooltip), а на информационной панели будет выведено более подробное сообщение о назначении этой кнопки.

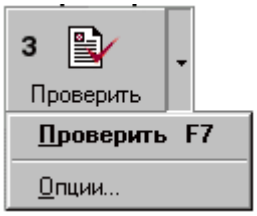
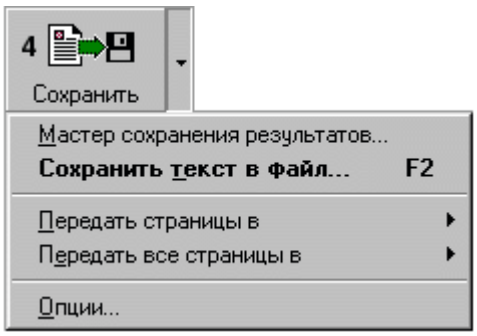
Главная панель программы - Scan&Read

Кнопки на панели Scan&Read связаны с базовыми операциями системы: Сканирование, Распознавание, Проверка и Сохранение результатов распознавания. Цифры на кнопках указывают, в каком порядке нужно выполнить действия, чтобы получить электронную версию бумажного документа. Каждое из этих действий можно провести по отдельности или объединить в одно, нажав на кнопку Мастер Scan&Read. Она позволяет провести полный цикл обработки текста автоматически. Каждая из кнопок имеет несколько режимов работы. Нажав на стрелку справа от кнопки, в открывшемся локальном меню Вы можете выбрать один из них, при этом "информация" об этом отразится на иконке

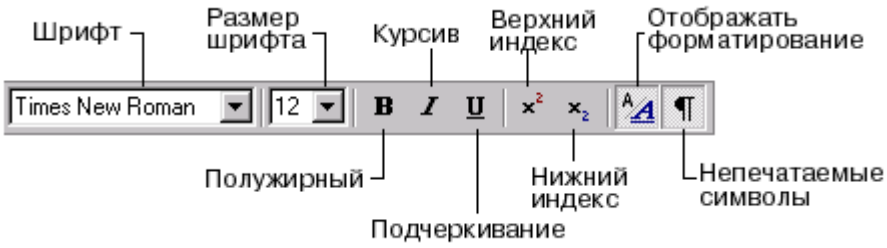
кнопки. Для того, чтобы повторить ту же операцию для другого изображения, Вам достаточно повторно нажать на кнопку.

Название кнопки	Режим работы
	Мастер Scan&Read - запускает специальный режим сканирования и распознавания, во время которого система контролирует действия пользователя и подсказывает ему, что надо делать, чтобы получить тот или иной результат. Сканировать и распознать - запускает сканирование и распознавание документа. Сканировать и распознать несколько страниц - сканирует и распознает несколько страниц в цикле. Открыть и распознать - позволяет открыть и распознать изображения, выбранные в диалоге Открыть (Open).

1-Сканировать  	Открыть изображение - добавляет изображение в пакет, при этом копия изображения сохраняется в папке пакета. Сканировать изображение - сканирует изображение. Сканировать несколько страниц - сканирует изображения в цикле. Чтобы остановить сканирование, в меню Файл выберите пункт Остановить сканирование. Опции - открывает закладку Сканирование/Открытие диалога Опции, на которой Вы может установить опции сканирования и предварительной обработки документа.
2-Распознать  	Распознать - распознает открытую страницу пакета. Распознать все - распознает все нераспознанные страницы пакета. Опции - открывает закладку Распознавание диалога Опции, на которой Вы может установить опции распознавания документа.

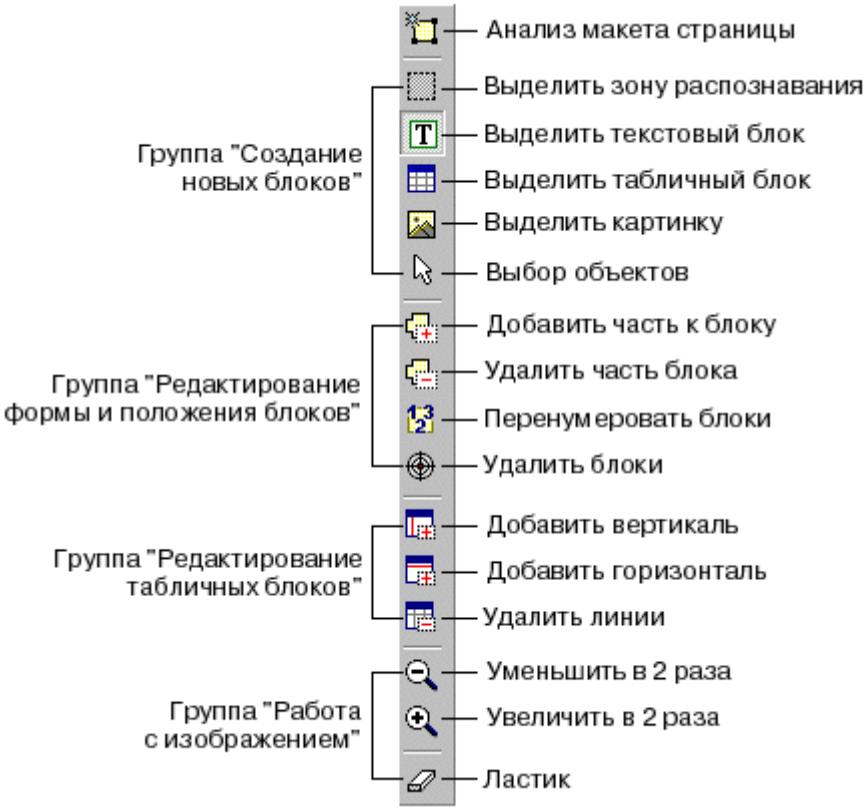
3-Проверить  	Проверить - позволяет найти в тексте слова, содержащие неуверенно распознанные символы, и неправильно написанные слова. Опции - открывает закладку Проверка диалога Опции, на которой Вы можете установить опции проверки документа.
4-Сохранить  	Мастер сохранения результатов - открывает диалог Мастер сохранения результатов, в котором Вы можете выбрать приложение для сохранения и установить опции сохранения. Сохранить текст в файл - сохраняет распознанный текст в файл на диск. Передать страницы в - напрямую передает распознанный текст в выбранное приложение без сохранения его на диск. При передаче распознанного текста с нескольких страниц пакета сначала выделите их в окне Пакет. Передать все страницы в - передает все распознанные страницы в выбранное приложение без сохранения их на диск. Опции - открывает закладку Форматирование диалога Опции, на которой Вы можете установить опции сохранения документа.

Панель Форматирование



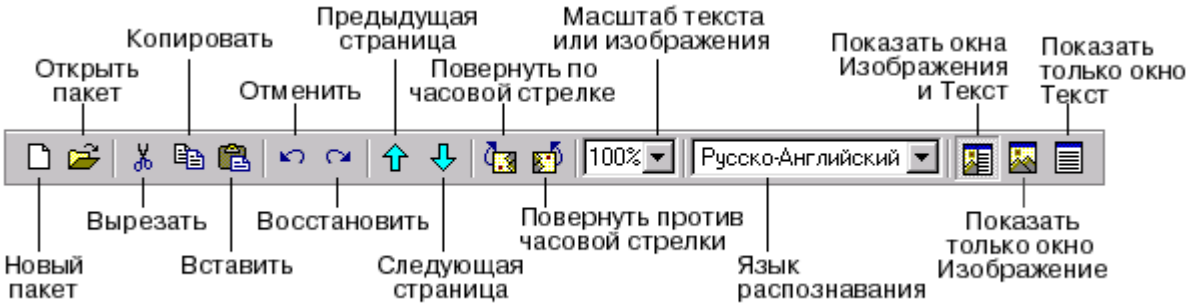
На панели Форматирование находятся кнопки, позволяющие изменить оформление текста.

Панель Изображение



Панель Изображение содержит кнопки, позволяющие производить анализ макета страницы (например, создать и отредактировать блоки), а также кнопки, позволяющие увеличить/уменьшить масштаб изображения, отредактировать изображение (например, стереть ненужные участки изображения, такие, как подписи или большие участки мусора).

Панель Стандартная



На панели Стандартная находятся кнопки, управляющие работой с файлами и изображением (отмена и повтор действия, перемещение по страницам пакета, очистка и поворот изображения), а также список языков распознавания.

3. Получение и обработка изображения программой FineReader

Качество распознавания во многом зависит от качества исходного изображения. В этой главе Вы узнаете о том, как правильно отсканировать документ, как открыть и распознать уже имеющиеся на компьютере изображения, как можно обработать изображение и устранить некоторые его дефекты (например, мусор, возникший при сканировании) и т.д

3.1. Сканирование

Чтобы запустить сканирование:

- Нажмите кнопку 1-Сканировать или в меню Файл выберите пункт Сканировать. Спустя некоторое время в Главном окне программы FineReader появится окно Изображение с "фотографией" вставленного листа.

- Если Вы хотите отсканировать несколько страниц, то нажмите стрелку справа от кнопки 1-Сканировать и в локальном меню выберите пункт Сканировать несколько страниц.

В случае если сканирование не началось сразу:

- откроется встроенный TWAIN-интерфейс сканера.

•откроется диалог Настройки сканера.

Совет:

Если Вы хотите сразу запустить распознавание отсканированных страниц, воспользуйтесь опцией Сканировать и распознать или Сканировать и распознать несколько страниц:

•Нажмите стрелку справа от кнопки Scan&Read и в локальном меню выберите один из пунктов: Сканировать и распознать или Сканировать и распознать несколько страниц.

FineReader отсканирует и распознает изображения. В Главном окне программы появятся окно Изображение с "фотографией" вставленного листа и окно Текст с результатом распознавания. Распознанный текст Вы можете сохранить во внешние редакторы и форматы.

3.2. Открытие файлов с изображениями

Если у Вас нет сканера, Вы можете распознавать графические файлы.

Чтобы открыть изображение:

•Нажмите стрелку справа от кнопки 1-Сканировать и в локальном меню выберите пункт Открыть изображение. Внешний вид иконки изменится; подпись Сканировать поменяется на Открыть.

•В меню Файл выберите пункт Открыть изображение.

В диалоге Открыть (Open) выберите одно или несколько изображений. Выбранные изображения появятся в окне Пакет, и последнее из выбранных изображений откроется в окне Изображение и в окне Крупный план на экране FineReader, при этом копия изображения помещается в папку пакета.

3.3. Проверка и корректирование полученного изображения

1. Очистить от мусора

Распознаваемое изображение может быть сильно "замусорено", т.е. содержать много лишних точек, возникших в результате сканирования документов среднего или плохого качества. Точки, близко расположенные к контурам букв, могут отрицательно сказаться на качестве распознанного текста. Чтобы уменьшить количество лишних точек, можно воспользоваться опцией Очистить от мусора. Для этого:

- В меню Изображение выберите пункт Очистить изображение от мусора.

Если Вы хотите очистить от "мусора" отдельный блок, то:

- В меню Изображение выберите пункт Очистить блок от мусора.

2. Инвертировать изображение

Некоторые сканеры инвертируют изображения при сканировании (черный цвет переводят в белый, а белый в черный). Чтобы получить стандартное представление документа (черный шрифт на белом фоне):

- В меню Изображение выберите пункт Инвертировать.

3. Повернуть или зеркально отразить изображение

При распознавании изображение должно иметь стандартную ориентацию: текст должен читаться сверху вниз, и строки должны быть горизонтальными. По умолчанию программа при распознавании определяет и корректирует ориентацию изображения автоматически. Если ориентация изображения была определена ошибочно, то на закладке Сканирование/Открытие снимите отметку с пункта Определять ориентацию страницы (в процессе распознавания) и поверните изображение вручную.

Чтобы повернуть изображение:

- на 90 градусов вправо - нажмите кнопку или выберите в меню Изображение пункт Повернуть по часовой стрелке.

- на 90 градусов влево - нажмите кнопку или выберите в меню Изображение пункт Повернуть против часовой стрелки.

- на 180 градусов - выберите в меню Изображение пункт Повернуть на 180 градусов.

Чтобы зеркально отразить изображение относительно:

•горизонтальной прямой - выберите в меню Изображение пункт Зеркальное отражение относительно горизонтали.

•вертикальной прямой - выберите в меню Изображение пункт Зеркальное отражение относительно вертикали.

4. Увеличить/Уменьшить масштаб изображения

•На панели Изображение (в окне Изображение) выберите инструмент и щелкните мышью на изображении. Изображение увеличится/уменьшится в два раза.

•Нажмите правой кнопкой мыши на изображение и в локальном меню выберите пункт Масштаб и нужный Вам масштаб.

4. Анализ макета страницы

Прежде чем приступить к распознаванию, программа должна знать, какие участки изображения надо распознавать. Для этого проводится анализ макета страницы, во время которого выделяются блоки с текстом, картинки, таблицы и штрих-коды (для версии Office).

В этой главе Вы узнаете, когда может потребоваться провести ручной анализ макета страницы, какие типы блоков бывают, как можно отредактировать полученные в результате автоматического анализа блоки, а так же, как можно упростить процесс анализа, используя шаблоны блоков.

4.1. Общая информация по анализу макета страницы

Анализ макета страницы может проводиться как автоматически, так и вручную. В большинстве случаев FineReader сам успешно справляется с анали-

зом сложных страниц. Автоматический анализ производится по кнопке 2-Распознать одновременно с распознаванием текста.

Замечание. Отдельная процедура анализа макета страницы тоже доступна (меню Процесс>Анализ макета страницы). Правда, при этом качество сегментации может быть ниже, т.к. при совместной процедуре распознавания и сегментации для анализа страницы используется дополнительная информация, полученная в процессе распознавания.

4.2. Типы блоков

Блоки - это заключенные в рамку участки изображения. Блоки выделяют для того, чтобы указать системе, какие участки отсканированной страницы надо распознавать и в каком порядке. Также по ним воспроизводится исходное оформление страницы. Блоки разных типов имеют различные цвета рамок. Вы можете изменить цвета рамок блоков на закладке Вид диалога Опции (меню Сервис>Опции) в группе Объекты. В поле Объект выберите нужный тип блока, а в поле Цвет - требуемый цвет.

При обработке изображений выделяют блоки следующих типов:

Зона Распознавания - блок используется для распознавания и автоматического анализа части изображения. После нажатия на кнопку 2-Распознать выделенный блок автоматически анализируется и распознается.

Текст - блок используется для обозначения текста. Он должен содержать только одноколоночный текст. Если внутри текста содержатся картинки, выделите их в отдельные блоки.

Таблица - этот блок используется для обозначения таблиц или текста, имеющего табличную структуру. При распознавании программа разбивает данный блок на строки и столбцы и формирует табличную структуру. В выходном тексте данный блок передается таблицей.

Вы можете выделить и отредактировать таблицу вручную.

Картинка - этот блок используется для обозначения картинок. Он может содержать картинку или любую другую часть текста, которую Вы хотите передать в распознанный текст в качестве картинки.

Штрих-код (только в версии Office) - этот блок используется для распознавания штрих-кодов. Т.е., если Ваш документ содержит штрих-код и Вы хотите передать его не картинкой, а перевести его в последовательность букв и цифр, то выделите штрих-код в отдельный блок и присвойте ему тип Штрих-код.

Замечание. По умолчанию опция, позволяющая искать и распознавать штрих-коды отключена. Чтобы подключить ее, отметьте пункт Искать штрих-коды на закладке Распознавание (меню Сервис>Опции).

4.3. Редактирование формы и положения блоков

Чтобы передвинуть границу блока:

1. Установите курсор мыши на границу блока.
2. Нажмите левую кнопку мыши и потяните в нужную сторону.
3. Отпустите кнопку мыши.

Замечание. Если Вы установите курсор мыши на угол блока, то при движении мыши будут одновременно изменяться вертикальная и горизонтальная границы блока.

Чтобы добавить прямоугольную часть блока:

1. Выберите инструмент.
2. Установите курсор мыши внутри блока, к которому Вы хотите добавить часть. Нажмите левую кнопку мыши и, не отпуская кнопки, потяните по диагонали. Выделив нужную часть изображения, отпустите кнопку мыши. Выделенный прямоугольник будет добавлен к блоку.
3. Если нужно, передвиньте границу блока.

Чтобы удалить прямоугольную часть блока:

1. Выберите инструмент .

2. Установите курсор мыши внутри блока, там, где Вы хотите вырезать часть. Нажмите левую кнопку мыши и, не отпуская кнопки, потяните по диагонали. Выделив нужную часть изображения, отпустите кнопку мыши. Выделенный прямоугольник будет удален из блока.

3. Если нужно, передвиньте границу блока.

Замечание. При удалении внутренней части блока снизу или сверху дополнительно удаляется часть блока справа до границы блока. Такая особенность программы обусловлена необходимостью передавать на распознавание неразрывные текстовые строки. В отношении боковых сторон блоков такого ограничения нет.

Чтобы выделить один или несколько блоков:

•Выберите инструмент и нажмите мышкой на нужном блоке или проведите указателем по выделяемым блокам при нажатой кнопке мыши.

Замечание. Вы можете выделить один или несколько блоков, используя стандартные инструменты создания блоков. Чтобы выбрать несколько блоков подряд, нажмите клавишу SHIFT (при этом должен быть выбран один из инструментов: , , или) и мышью нажмите на требуемых блоках. Чтобы отменить выделение уже выбранного блока или добавить невыделенные блоки, нажмите клавишу CTRL (при этом должен быть выбран один из инструментов: , , или) и мышью нажмите на требуемых блоках.

5. Распознавание

Задача распознавания состоит в том, чтобы преобразовать отсканированное изображение в текст, сохранив при этом оформление страницы. Прежде чем приступить к распознаванию текста, необходимо установить основные параметры распознавания: язык распознавания, тип печати распознанного текста и тип страницы. В этой главе описываются эти и другие параметры распознавания и приводятся ситуации, в которых они используются.

5.1. Общая информация по распознаванию

Внимание! Перед запуском распознавания проверьте установленные опции: язык распознавания, тип печати распознаваемого текста и тип страницы.

Вы можете:

- Распознать блок или несколько блоков, выделенных на изображении.
- Распознать открытую страницу или все страницы, выделенные в окне Пакет.
- Распознать все нераспознанные страницы пакета.
- Распознать все страницы в фоновом режиме. В этом режиме возможно распознавание с одновременным редактированием уже распознанных страниц.
- Распознать страницы в режиме распознавание с обучением. Данный режим используется в основном для распознавания текстов, использующих декоративные шрифты, или распознавания большого объема (более 100 страниц) документов плохого качества печати.
- Распознать страницы одного пакета на нескольких компьютерах одновременно.

Чтобы запустить распознавание:

- Нажмите кнопку 2-Распознать на панели Scan&Read.
- В меню Процесс выберите нужный Вам пункт:

Распознать - чтобы распознать открытую страницу или все страницы, выделенные в окне Пакет;

Распознать все - чтобы распознать все нераспознанные страницы пакета;

Распознать Блок - чтобы распознать блок или несколько блоков, выделенных на изображении;

Запустить фоновое распознавание - чтобы запустить распознавание в фоновом режиме.

Замечание. При распознавании уже распознанной страницы перераспознаются только отредактированные и добавленные блоки.

6. Проверка и редактирование текста

После завершения распознавания результат появляется в окне Текст. Окно Текст - это встроенный редактор программы FineReader; в нем Вы можете проверить результаты распознавания и отредактировать распознанный текст.

Одна из возможностей текстового редактора FineReader – это встроенная проверка орфографии (список языков, для которых поддерживается проверка орфографии, приведен в Поддерживаемых языках распознавания).

Система встроенной проверки орфографии позволяет:

- Находить неуверенно распознанные слова (слова, в которых есть неуверенно распознанные символы).
- Находить орфографические ошибки (неправильно написанные слова).
- Добавлять неизвестные системе FineReader слова в словарь для того, чтобы они распознавались уверенно.

6.1. Проверка распознанного текста

Неуверенно распознанные символы и слова, которых нет в словаре, выделяются различными цветами. По умолчанию для выделения неуверенно распо-

знанных символов используется голубой, для несловарных слов - розовый. Чтобы изменить цвета:

•На закладке Вид (меню Сервис>Опции) в поле Объект выберите пункт Неуверенно распознанный символ (Несловарное слово) и в поле Цвет - цвет подсветки.

Чтобы проверить результаты распознавания:

- Нажмите кнопку Проверить на панели Scan&Read (или выберите пункт Проверка в меню Сервис).
- Откроется диалог Проверка.
- В диалоге Проверка три окна. Верхнее окно - аналог окна «Крупный план» программы FineReader, в нем показано изображение слова с возможной ошибкой. Среднее окно показывает само слово с возможной ошибкой, в строке над этим окном выводится название типа ошибки. В нижнем окне, Варианты, предлагаются варианты замены данного слова (если таковые имеются).

Замечание. Для удобства проверки и редактирования текста диалог Проверка может быть увеличен. Для этого достаточно поместить указатель мыши на границу диалога (указатель превратится в двустороннюю стрелку). Перетаскивая границу, увеличьте или уменьшите размеры диалога.

Вопросы:

1. Что такое OCR-системы?
2. Инструментальные панели программы FineReader.
3. Анализ макета страницы, блоки.
4. Распознование текста.
5. Какие существуют возможности сохранения распознанного текста?

Порядок выполнения работы:

1. Создать пакет.
2. Добавить изображение со сканированным текстом (нескольких типов).
3. Выделить блоки текста автоматически и в ручную, распознать.
4. Провести анализ качества распознанного текста с выделенными блоками текста автоматически и в ручную. Попробовать обучить программу распознавать не распознанные символы.

ЛАБОРАТОРНАЯ РАБОТА №6

Тема: Автоматизированная система лингвистического перевода PROMT 99.

Цель работы: Ознакомление с автоматизированной системой перевода на базе программного продукта PROMT 99.

Введение

PROMT 99 (в дальнейшем изложении - PROMT), принадлежит к системам автоматического перевода разработанным фирмой ПРОект МТ и обеспечивающим перевод с основных европейских языков на русский и обратно. В настоящий момент система позволяет работать с текстами на английском, немецком и французском языках в зависимости от приобретенной комплектации. Это 32-разрядтюе приложение, специально предназначенное для использования среде Windows 9X. Однако, оно может полноценно функционировать и среде Windows NT, 2000, XP.

1. Быстрое начало

В этой главе описаны наиболее простые приемы работы с системой PROMT. Это может помочь вам не только быстро получить результат еще при начальном знакомстве с PROMT, но и в дальнейшей работе (например, при переводе текстов в сжатые сроки и невысоком требовании к качеству перевода).

1.1 Перевод в среде PROMT

1. Запустите PROMT

Для этого нажмите кнопку Пуск в Windows, установите указатель на пункт Программы, затем на пункт PROMT 99 и выберите команду PROMT 99 (или дважды щелкните по значку PROMT если он находится на рабочем столе). Откроется окно Добро пожаловать!

Это окно позволит вам запустить учебник, обучающий работе с программой, а также легко создать новый документ, или открыть документ или словарь, с которым вы работали в прошлый раз, открыть документ с диска.

2. Выберите документ. который хотите перевести, и нажмите кнопку Открыть. Появится окно Конвертировать файл.

3. Проверьте, правильно ли система определила формат файла и направление его перевода, и нажмите ОК. Появится окно документа с исходным текстом.

4. Поместите указатель мыши на любое слово в тексте.

Появится всплывающее окно с переводом.

Выделите несколько слов в пределах одной строки и поместите на них указатель мыши. Появится связный перевод этих слов.

Далее вы можете перевести весь текст или любой его фрагмент. Для этого выберите в меню Перевод любую из команд:

- Очередной абзац;
- Текущий абзац;
- Весь текст;
- Вверх;
- Вниз;
- Выделенные абзацы.

5. Для сохранения полученного перевода выберите в меню Файл пункт Сохранить и затем команду Перевод. Появится окно просмотра папок.

6. Задайте имя файла, его тип, а также папку, в которой его следует сохранить, и нажмите кнопку Сохранить.

7. Завершите работу в PROMT, выбрав к меню Файл команду Выход (либо нажмите Alt-F4, или Alt + X, или кнопку закрытия окна PROMT).

Документ с переводом будет сохранен, и вы можете использовать любой редактор текстов для его просмотра и редактирования.

1.2 Перевод внутри других приложений

1. В документе любого приложения или в окне справочной системы выделите текст, который нужно перевести, и скопируй его в буфер обмена. Для этого выберите в меню Правка команду Копировать.

2. Щелкните правой кнопкой мыши по значку перевод находящемуся на панели задач и в появившемся меню выберите команду перевод Clipboard. Появится окно с запросом о направлении предполагаемого перевода.

3. Выберите направление перевода и нажмите OK. Появится окно с выполненным переводом.

4. Отредактируйте перевод и нажмите кнопку Заменить. Перевод будет сохранен в буфере обмена Clipboard, откуда его можно вставить в документ любого приложения, имеющего средства редактирования.

2. Объекты системы PROMT

Основными объектами, с которыми вам придется иметь дело при работе в системе PROMT, являются:

- словари;
- PROMT-документы;
- списки зарезервированных слов.

В данной главе описано их назначение, правила создания, сохранения и подключения.

2.1 Словари

Высокое качество перевода при работе с PROMT достигается благодаря возможности настройки системы на перевод текстов конкретной тематики. Помогут в этом различные словари. Вы можете не только подключать словари к системе, но и менять их приоритет, пополнять, корректировать.

Словари системы PROMT делятся на три типа:

- генеральные,
- специальные,
- пользовательские.

2.1.1 Генеральные словари

Генеральные словари содержат общеупотребительную лексику, включая многофункциональные и служебные слова и словосочетания, которые в значительной мере формируют структуру предложения. Без подключения специализированных словарей они могут использоваться для перевода бытовых писем, газетных и журнальных статей на популярные темы, а также простых художественных текстов.

Генеральные словари закрыты для пользователя, т.е. их нельзя просматривать, корректировать и пополнять. Однако можно открыть словарную статью какого-либо слова из генерального словаря, изменить ее, а затем сохранить в своем пользовательском словаре.

2.1.2 Специализированные словари

Специализированные словари обеспечивают не только правильный перевод отдельных слов и словосочетаний из предметных областей, но и высокое качество перевода в целом. Это объясняется тем, что специализированные словари содержат не только перевод слов (словосочетаний), но и обширную информацию о структуре предложений, характерных для текстов по заданной те-

матике. Так, например, словарь по информатике позволяет правильно переводить предложения с глаголами в повелительном наклонении, т.к. именно они составляют основу документации по использованию программ.

2.1.3 Пользовательские словари

Для пополнения словарного запаса системы вы можете создавать свои собственные пользовательские словари, которые можно пополнять и корректировать. В случае если вас не устраивает перевод какого-либо слова из специализированного или генерального словаря, и вы хотите его изменить, откорректированная словарная статья может быть сохранена только в подключенном пользовательском словаре.

2.2. PROMT – документ

Для того чтобы эффективнее использовать многие возможности PROMT, необходимо хорошо освоиться с еще одним основным понятием системы — понятием PROMT - документа.

PROMT-документ это совокупность:

- текста оригинала;
- текста перевода;
- структуры, поддерживающей связь абзацев оригинала и перевода;
- списка зарезервированных слов и абзацев (тех, которые были отмечены как не требующие перевода);
- списка незнакомых слов (не найденных в словаре).

PROMT-документ автоматически образуется при загрузке существующего файла. При этом создается сложная структура информации, отличная от структуры исходного файла. Таким образом, исходный документ и соответствующий ему PROMT-документ — это существенно разные объекты.

Если вы выполнили перевод и редактирование текста не до конца и хотите в дальнейшем продолжить работу, PROMT предоставляет вам возможность сохранить PROMT документ. При этом сохраняется и оригинал, и перевод, и вся структура связи между ними, а также вышеупомянутые списки слов.

2.2.1.Открытие документа, созданного в другом приложении

Работая в PROMT, вы можете открывать и переводить документы других приложений. Ниже приведены типы файлом, которые могут быть открыты для перевода:

- Документы PROMT (STYLUS)
- Форматированный текст (RTF)
- Файлы Microsoft Write
- Файлы гипертекста HTML
- Текстовые файлы
- MS-DOS текстовые файлы.

Открывая файл, программа преобразует его в PROMT-документ. Открыть документ, созданный в другом приложении, можно как из среды PROMT, так и из среды Windows. Чтобы открыть документ из среды PROMT:

1. Запустите PROMT.

2. Выберите в меню Файл или в контекстном меню команду Открыть. Появится окно просмотра папок.

3. Выберите нужный файл и дважды щелкните мышью по его значку или нажмите кнопку Open (Открыть).

Появится окно Конвертировать файл.

Формат файла, автоматически определенный системой, будет выделен в списке. Кроме того, система сама определит предполагаемое направление перевода. Это направление определяется и зависимости от того, каких букв в тексте

документа больше - русских или латинских. Если больше русских, то система считает, что направление перевода русско-английское.

Если исходный текст английский, немецкий или французский, PROMT также попытается определить их принадлежность, но при этом возможен ошибочный выбор направления перевода. Если система ошиблась в направлении перевода или просто его необходимо сменить, то вы можете это сделать сами, выбрав нужное направление из раскрывающегося списка Направление перевода.

4. Проверьте правильность определения формата.

Если он определен неверно, рекомендуется выйти из PROMT и подготовить документ в виде форматированного текста (RTF). Если файл текстовый, нужно вручную выделить в списке его формат, т.к. система не различает форматы текстовых файлов:

Только текст Текст с разметкой	Текстовые файлы, созданные в среде Windows
MS-DOS текстовый файл	Текстовые файлы, созданные в среде DOS (в коде ASCII)

Если вы выберете формат Только текст или MS-DOS текстовый файл, каждая строка будет рассматриваться как отдельный абзац, что может привести к нарушению связности перевода. При выборе форматов Текст с разметкой или MS-DOS текст с разметкой система удалит разделители строк и попытается разбить текст на правильные абзацы.

5. Нажмите ОК.

Система преобразует исходный файл в PROMT-документ, и на экране появится окно этого документа с исходным текстом и мигающим курсором на первой позиции.

Можно одновременно открыть несколько документов. Для этого выделите имена сразу нескольких файлов, удерживая Ctrl. После нажатия кнопки Открыть

все выбранные файлы будут по очереди конвертированы и появятся в открытых окнах документов. Активным станет окно последнего из них.

Открыть документ из среды Windows можно одним из следующих способов:

- Перетащите выделенный значок документа (или его ярлык) на значок программы PROMT (или ее ярлык), удерживая правую кнопку мыши. Отпустив кнопку мыши, выберите из появившегося меню команду Открыть с помощью. Появится окно PROMT, и начнется процесс открытия документа, описанный выше.
- Перетащите выделенный значок документа (или его ярлык) на значок программы PROMT (или на ее ярлык), удерживая левую кнопку мыши. Появится окно PROMT, и начнется процесс открытия документа.

2.2.2. Открытие существующего PROMT – документа

PROMT-документ можно открыть:

- из среды PROMT,
- из среды Windows.

Чтобы открыть PROMT - документ из среды PROMT:

1. Запустите PROMT.
2. Выберите в меню Файл или в контекстном меню команду Открыть (либо нажмите Ctrl-O). Появится окно просмотра папок.
3. Выберите нужный документ и дважды щелкните мышью по его значку или нажмите кнопку Открыть. Появится окно PROMT-документа.

В среде Windows двойной щелчок мышью по значку PROMT – документа открывает окно PROMT с эти документом, Если вы открыли документ из окна папки, то его имя автоматически появится в списке команд пункта Документы данного меню Windows. После этого вам больше не придется пробираться к

данному документу через иерархию папок: его можно будет открывать при помощи меню Документы.

3. Функции системы PROMT

3.1. Последовательность работы в PROMT

Работа в PROMT включает следующие этапы:

1. Не все из приведенных на схеме этапов работы должны выполняться всегда. Обязательными являются только открытие (или создание) документа и его перевод. Выполнение остальных этапов зависит от цели перевода и требований, предъявляемых к его качеству.

2. При работе в приложениях Word и Excel, в которые встроены функции перевода, доступно большинство из перечисленных этапов, кроме чтения текста сканером, использования электронного словаря и сохранения промежуточных результатов перевода в виде PROMT - документа (вместе с соответствующими списками незнакомых и зарезервированных слов). Однако предусмотрен переход в среду PROMT для выполнения и этих функций.

3. Предусмотрен упрощенный вариант перевода текста, скопированного в буфер обмена. Такой перевод возможен при работе в любом приложении при помощи значка Интегратор PROMT, расположенного па напели задач. Подготовка исходного текста при этом производится в соответствующем приложении. Полученный перевод выводится в специальном окне, где и может быть отредактирован. Остальные этапы работы при этом недоступны. Для их выполнения, в частности, для подключения специальных словарей, необходимо перейти в среду PROMT.

3.2. Перевод

PROMT позволяет:

- Переводить текст целиком или фрагментами (абзацами, группами абзацев, в заданном направлении относительно курсора);
- Получать мгновенный перевод одного или нескольких слов в пределах одной строки;
- Переводить текст, скопированный в буфер обмена (эта возможность может быть использована при работе в других приложениях).

В ходе перевода можно редактировать текст и заново переводить измененные фрагменты, пополнять словари, резервировать слова, не требующие перевода. Все это позволяет добиваться высокого качества перевода.

3.2.1. Способы перевода

Перевод очередного абзаца.

Если вы хотите перевести первый абзац, который еще не переведен (или тот, в который после перевода внесены изменения), выберите в меню Перевод или в контекстном меню - команду Очередной абзац (либо нажмите F9 или кнопку инструментальной панели серого цвета). По этой команде система просмотрит исходный текст с самого начала, дойдет до первого измененного или непереведенного абзаца и переведет его.

Этот абзац станет текущим. Он и его перевод будут выведены вверху, каждый - в своей части окна документа.

Перевод текущего абзаца.

Чтобы перевести любой абзац независимо от того, был ли он переведен ранее, установите в него курсор и выберите в меню Перевод или в контекстном меню команду Текущий абзац (либо нажмите Alt-F9 или кнопку инструментальной панели голубого цвета). Текущий абзац в исходном тексте и его перевод будут выделены цветом и выведены вверху, каждый в своей части окна документа.

Перевод всего текста.

Чтобы перевести весь текст, выберите в меню Перевод команду Весь текст (либо нажмите Ctrl + F9 или кнопку инструментальной панели). В процессе перевода появится окно с индикатором хода работы.

Перевод вверх (вниз).

Чтобы перевести часть текста выше или ниже курсора, выберите в меню Перевод команду Вверх или Вниз соответственно. Если в указанной части текста уже есть переведенные фрагменты, система предложит подтвердить необходимость выполнения команды. В процессе перевода появится окно с индикатором хода работы.

Мгновенный перевод.

PROMT позволяет автоматически, без всякой команды высветить перевод слова, на которое помещен указатель мыши. Точно также сразу появится перевод нескольких слов, выделенных в пределах одной строки, если поместить на выделенный участок указатель. Чтобы включить или выключить режим мгновенного перевода, выберите в меню Сервис или в контекстном меню окна PROMT команду параметры и в окне настройки параметров на вкладке Разное установите или снимите флажок Использовать мгновенный перевод.

3.2.2. Быстрый перевод текста, набранного на клавиатуре

Если вы собираетесь переводить текст, набранный на клавиатуре, и вам не требуется печатать или сохранять полученный перевод, то воспользуйтесь программой QTrans. Она входит в систему PROMT и обеспечивает быстрое получение результата. При этом программа позволяет менять направление перевода, подключать словари, добавлять в них новые слова и словосочетания, и также указывать слова, оставляемые без перевода.

Запустить программу QTrans можно двойном щелчком по ее значку, или при помощи команды QTrans, из группы PROMT 99 меню Программы, а также при помощи панели или меню Интегратора.

Окно программы разбито на две части: в верхней вводится текст, а в нижней появляется его перевод.

Назначение большинства кнопок инструментальной панели ясно из их названия. Пользуясь командой меню, вызываемого кнопкой, вы сможете:

- добавлять в словарь слово или словосочетание;
- указать слово, не требующее перевода;
- выполнить простейшие операции редактирования.

4. Проверка орфографии

4.1. Программы проверки орфографии

Если на вашем компьютере установлена программа для проверки орфографии, ее можно использовать во время работы с PROMT. Вы можете проверить правописание в выделенном фрагменте русского или иностранного текста, а также проверить весь текст, являющийся в данный момент текущим (исходный текст или его перевод).

ПРИМЕЧАНИЕ:

PROMT может использовать только 32-разрядные программы проверки орфографии, предназначенные для MS Office.

4.1.1. Подключение программы проверки орфографии

1. В меню Сервис выберите команду Подключение внешних приложений.

2. В появившемся окне на вкладке Программы проверки орфографии выделите язык, для которого нужно проверять правописание.

3. Нажмите кнопку Изменить.

4. Выберите нужную программу проверки орфографии и, при необходимости, укажите пользовательский словарь, который будет использоваться. Если выбранная программа не установлена на вашем компьютере, появится соответствующее сообщение.

5. Нажмите ОК. Выбранная вами программа появится в списке рядом с названием выделенного языка.

2. При необходимости выберите программы проверки орфографии и для других языков. Причем эти программы могут быть разными.

3. Нажмите ОК (или кнопку Применить, если не хотите закрывать текущее окно).

4.1.2. Проверка орфографии

Для выполнения проверки:

1. Поместите курсор в окне документа в позицию, которой нужно начать проверку (или выделите текст).

2. Выберите в меню Сервис команду Орфография (либо нажмите F7 или кнопку инструментальной панели).

Программа проверки будет последовательно высвечивать каждое найденное неизвестное слово в поле Нет в словаре окна Проверка орфографии.

Проверка всего документа происходит от текущего положения курсора до конца документа и затем продолжается от его начала до курсора. После ее окончания появится сообщение о том, что проверка завершена. Если проверялся выделенный текст, появится сообщение о том, что проверка окончена, и предложение проверить остальную часть текста.

3. Для выхода из режима проверки нажмите ОК, если проверялся весь документ, или Нет – в ответ на предложение проверить остальную часть текста, если проверялся выделенный фрагмент.

Контрольные вопросы:

1. Определите назначение системы PROMT 99.
2. Перечислите объекты системы PROMT 99 и дайте им краткую характеристику.
3. Попрактикуйтесь в подключении, отключении и изменении приоритета словарей. Убедитесь (на примере текста специфической направленности, например, технической), что приоритет словарей существенно влияет на результат перевода.
4. Создайте свой пользовательский словарь на основе какого-либо специального словаря. В созданный словарь внесите пять новых слов или словоформ, выбранных на ваше усмотрение. Убедитесь, что система использует новые словоформы при переводе.

Задания:

1. Создайте новый PROMT – документ, введите с клавиатуры текст (определенной направленности, не менее пяти предложений), и добейтесь корректного перевода этого текста.
2. Откройте документ, созданный в каком-либо другом приложении, осуществите корректный перевод. Сохраните результат.
3. Возьмите результат предыдущего задания и произведите обратный перевод, сравните его с оригиналом текста, сделайте вывод.
4. Попрактикуйтесь в различных способах перевода текста (по абзацам, текст целиком, из буфера обмена и т.д.).
5. Внесите в панель Интегратора PROMT 99 два приложения на ваш выбор.

6. Подключите установленные на компьютере программы проверки орфографии (одну или несколько). Продемонстрируйте проверку орфографии оригинального текста и его перевода.

ЛАБОРАТОРНАЯ РАБОТА №7

Тема: Символьные вычисления с помощью программы Maple V.

Цель работы: Получить представление о возможностях предоставляемых программой Maple V. Приобрести навыки работы с программой символьных вычислений Maple V.

Введение в компьютерную алгебру

Алгебраические вычисления не потеряли своей актуальности. Однако, чаще всего они выполняются вручную, хотя первые логические устройства для их автоматизации появились достаточно давно (в 1953 году). Очень скоро стало ясно, что программное обеспечение для использования в алгебраических вычислениях должно представлять собой полную систему, включающую метод представления нечисловых данных весьма специальной структуры, язык, позволяющий манипулировать с ними, и библиотеку эффективных функций для выполнения необходимых базисных алгебраических операций.

В настоящее время появились хорошо работающие системы, такие как Maple, Mathematica, Macsyma, Derive, Axiom и некоторые другие. Все упомянутые выше системы, так же как и большинство неупомянутых, являются весьма дружественными по отношению к пользователю.

1. Аналитические преобразования

Математический пакет Maple предназначен для символьных математических вычислений. Хотя между различными пакетами имеются различия, синтаксис - не главная проблема. В общем случае нескольких часов и небольшой практики вполне достаточно. Можно утверждать, что синтаксис всех систем символьных вычислений в значительной степени аналогичен синтаксису Паскаля или Си. Обязательно имеются операторы присваивания, понятие вызывающей функции (команды), более или менее богатый выбор управляющих структур (if,do,for,while и т.д.), возможности определения процедур... - одним словом, весь арсенал классических языков программирования, необходимый для записи алгоритмов. Для языков Maple, Mathematica и Derive встроенные справочники являются, по-видимому, наиболее доступными учебниками, как по синтаксису, так и по использованию команд. В силу этого мы не будем далее стремиться к полному и детальному описанию команд и функций. Всю необходимую информацию можно получить в Maple используя команды:

- ? (help) - помощь,
- ?? (usage) - проверка синтаксиса,
- ??? (example) - пример для данной команды, или меню в оболочке Maple для Windows.

Пример 1 (помощь).

```
> ?plot;
> ???factor;
> help(linalg);
```

В Maple выполняемые математические выражения вводятся всегда после символа >, а заканчиваются точкой с запятой или двоеточием, если результат не надо выводить на экран. Чтобы продолжить запись предложения на следующей строке используют комбинацию "Shift+Enter". При нажатии клавиши "Enter" предложение выполняется. Обнаружив ошибку, Maple выводит сообщение о ней в следующей строке.

Кроме обычных знаков математических операций используют :

** или ^ - возведение в степень,

! - факториал,

:= - знак присвоения,

<, >, >= , <= , = - логические операции,

" , "", """ - результаты трех предыдущих выполненых операций,

@ - сложная функция, т.е. (f @ g)(x)= f(g(x)),

D - дифференциал и т.д.

Назначение двойных кавычек - экономить усилия при последовательной работе с документом, но приводит к неприятностям при свободном перемещении по тексту. Обратные кавычки указывают на строку символов, прямые кавычки (апостроф) используются для того, чтобы освободить переменную, которая уже использовалась. Заметим, что такие "парные" символы как кавычки, возведение в степень и некоторые другие вводятся с задержкой (советуем использовать пробел после введения такого "парного" символа).

Пример 2 (выражения).

```
> 5+4+9;
> f:=8+9+14:
> f;
> e1:=x*(x+1)*(x-1);
> factor( " );
> e:=x^3+x**2+1;
> x(y-1);
```

Всегда проверяйте написанное - в последней строке пропущен знак умножения x*(y-1) - тем не менее для Maple это не ошибка, данное выражение воспринимается как функция x от аргумента (y-1).

```
> i:=3;
> sum(i^2,i=1..6);
```

Последнюю строку Maple воспринимает как 3=1..6, т.е. как ошибку. Правильно будет использовать апострофы

> sum('i^2','i'=1..6);

1.1. Важнейшие математические константы

Обратите внимание, что имена констант начинаются с больших букв

> Pi; evalf(");

> E; evalf(");

> I;

> infinity;

> gamma; evalf (");

Более подробно смотри в ininames .

Авторы всех систем символьных вычислений следуют принципу Кронекера - "Бог создал целые числа - все остальное дело рук человеческих".

Во всех языках представление целых чисел первично, так как в этом случае алгоритмы сложения и умножения становятся, по-крайней мере в принципе, довольно простыми.

Деление представляет значительно более сложную задачу, поскольку метод, изучаемый в школе, не является алгоритмом, так как он требует угадывания цифры частного. Knuth (1981) детально описал эту проблему и предложил алгоритм такого угадывания, который почти всегда является правильным.

В силу этого "ленивого" принципа Maple выдает только целые числа или имена без численных ответов до тех пор пока вы об этом не попросите. Основные команды "принудительного" вычисления

evalb - для булевых выражений,

evalc - для комплексных выражений,

evalf - для операций с плавающей запятой,

evalm - для вычисления матричных выражений.

Пример 3 (вычисление выражений).

> 25/15+23^3;

```
> evalf(");
> sin(36);
> evalf(");
> sqrt(2+3); evalf(");
```

Как видим, на экран выводятся только целые числа, пока не попросишь об обратном. Для операции с плавающей точкой можно задать число цифр после запятой, присвоив необходимое число переменной Digits (начальное значение 10).

Совет: не используйте числа с плавающей точкой в промежуточных вычислениях.

Это приводит к потере времени, точности и иногда приводит к совершенно неверным результатам.

Можно объявлять, что переменные зависят друг от друга, т.е. строить функции и отображения. Следует отметить, что зависимость является глобальным свойством, не подчиняющимся блочной структуре. Зависят ли переменные друг от друга можно узнать командой depends

Пример 4 (зависимость).

```
> depends(sin(x)+cos(z),{x,y});
> depends(int(f(x),x=a..b),x);
> depends(int(f(x),x=a..b),a);
```

Все наиболее распространенные математические функции уже определены в виде встроенных функций. Часть из них заданы изначально, другие собраны в библиотеки программ (пакеты), которые необходимо загружать по мере необходимости.

1.2. Встроенные функции

Список встроенных функций достаточно велик. Более того, используя дополнительные пакеты данный список можно расширить. Имена функций в па-

кетах символьных вычислений максимально приближены к стандартным математическим именам. Например, sin, cos, tan, exp, ln, GAMMA, Zeta, или binomial.

Список всех основных функций смотри в inifcns.

Самые важные функции в пакетах символьных вычислений связаны не столько с математикой, сколько с проблемой общения пользователя и компьютера.

В пакете Maple это функции:

- simplify - упростить выражение,
- factor - факторизовать,
- expand - разложить (раскрыть все скобки),
- normal - привести выражение к "нормальному" виду,
- convert - переписать в заданном виде,
- coeff - выделить коэффициенты полинома,
- collect - собрать вместе части выражения.

Пока мы достаточно грубо описали эти функции. Используя их в дальнейшей работе, мы более детально изучим их свойства.

Слово "упрощение" допускает множество интерпретаций. Очевидно, что запись (x-1) более "простая", чем (x^2-1)/(x+1), но является ли выражение x^999 - x^998 + x^997 - ... -1 более простым, чем (x^1000-1)/(x+1) .

На практике каждому пользователю хочется большего, и именно здесь включается в игру субъективность.

Пример 5 (что такое "просто").

Рассмотрим выражение

> x^100-1; simplify(");

Вид этого выражениея "прост" и для компьютера и для пользователя.

Разложим этот полином на множители

> factor(x^100-1);

и попробуем упростить

> simplify(");

Как видно, это выражение так же "просто" для компьютера как и прежнее.

Но если вы получите это выражение в результате своих вычислений, то вряд ли

вид этого выражения удовлетворит вас. Чтобы упростить данное выражение раскроем все скобки

> expand("");

Данный пример показывает, что даже получив правильный ответ с помощью компьютера, вы должны еще уметь представить его в удобном для пользователя виде.

Пример 6 (упростить).

> simplify((x^a)^b+4^(1/2), power);

> simplify(exp(a+ln(b*exp(c))));

> simplify(sin(x)^2+cos(x)^2, trig);

Подстановка с помощью данной команды

> f := -1/3*x^5*y + x^4*y^2 + 1/3*x*y^3 + 1:

> simplify(f, {x^3 = x*y, y^2 = x+1});

Работа с многозначными функциями требует большего труда

> g:=sqrt(x^2);

> simplify(g);

> simplify(g,assume=real);

> simplify(g,assume=positive);

Можно всегда максимально упрощать используемые выражения, это не только приятно для глаза, но и позволяет сократить время работы и размеры необходимой для работы памяти.

Однако и довольно часто, для пользователя различные представления формул могут ассоциироваться с другими формулами. Это позволяет продвинуться в работе значительно дальше и быстрее чем простой счет на компьютере.

Факторизация означает разложение выражения на множители. В лучших алгоритмах факторизации время счета растет медленнее, чем экспоненциально, но быстрее, чем полиномиально. На практике Maple трактует факторизацию достаточно широко.

Пример 7 (факторизовать).

```
> factor(x^3+5);
> factor(x^3+5, 5^(1/3));
> factor(x^3+5, {5^(1/3),(-3)^(1/2)});
> factor(x^3+5.0);
> factor(sin(x)^2-1);
```

Для полиномов и рациональных функций это просто раскрытие всех скобок, для других функций это более сложная процедура. Причем, как показывает опыт, это одна из наиболее непредсказуемых процедур, особенно когда вы работаете с радикалами.

Пример 8 (разложить).

```
> expand((x+1)*(x+2));
> sin(x+y); expand(");
> exp(x+y); expand(");
> expand(BesselJ(2,t));
```

В последнем выражении используется функция Бесселя.

Команда normal обычно используется для полиномов и рациональных функций, хотя иногда применима и для более общих выражений.

Пример 9 (нормальный вид выражения).

```
> normal( 1/x+x/(x+1) );
> normal( 1/x+x/(x+1), expanded );
> sin(x)+1/sin(x)^2; normal(");
```

Команда convert имеет следующий ситаксис: convert(x,type) или convert(x,type,option). Как всегда, типы и опции можно найти в справочнике.

Пример 10 (преобразовать).

```
> convert( 1.23456, fraction );
> (x+a)/(x*(x-b)*(x^2+c)); convert(",parfrac,x);
> sin(x); convert(");
> exp(I*x); convert(",trig);
> convert(evalf(Pi),rational,exact);
```

Команды collect и coeff приводят выражение к нормальному виду и позволяют явно выделять коэффициенты полинома.

Пример 11 (работа с полиномами).

```
> f := x*(x+1)+y*(x+1);
> g := collect(f,x);
> coeff(g,x,1);
```

Определение нормальной формы полинома от многих переменных более сложное. Кроме лексикографического представления выделяют так же рекурсивное и распределенное представления по различным наборам переменных

```
> p:= x*y+a*x*y+y*x^2-a*y*x^2+x+a*x:
> collect(p, [x,y], recursive );
> collect(p, [x,y], distributed);
> collect(p, [y,x],recursive);
> collect(p, [y,x], distributed );
```

Кроме этого, данная команда позволяет выделять часть из целого и не только в полиномах, например

```
> f := a*ln(x)-ln(x)*x-x;
> collect(f,ln(x));
```

Команда combine является аналогом команды collect для функций общего положения, (математический алгоритм команды был разработан только в 1986 году!).

Пример 12 (комбинирование).

```
> f:=exp(sin(a)*cos(b)) * exp(cos(a)*sin(b));
> combine(f,exp);
```

> combine(f,[trig,exp]);

Последний вид, по-моему, наиболее приятен для пользователя. Кроме этого данное упрощение, проведенное в промежуточных расчетах, позволит сократить и время работы и размер необходимой памяти.

> combine(Psi(-x)+Psi(x),Psi);

Psi(x) - дигамма функция, определяемая как производная от логарифма гамма функции Psi(x) = diff(ln(GAMMA(x)) , x).

2. Представления математических объектов

Довольно часто при проведении вычислений мы называем какое-нибудь громоздкое выражение одной буквой и далее используем это обозначение. При этом мы экономим время и бумагу, но все время помним про эту замену (т.е. активно используем память). В Maple для этого используют команду alias (newname = oldname).

Пример 1 (замены).

> f:=sin(x)+sin(x)^3+sin(x)^5;

> alias(A=sin(x));

> f;

В области вывода вы видите список всех таких замен. Как видно, мнимая единица I (выделенный радикал!) определена в Maple именно таким образом.

В конце вычислений нам надо вместо A подставить первоначальное выражение. Для подстановки достаточно просто отменить замену.

> alias(A=A);

и получим результат

> f;

> alias(alpha=RootOf(x^2-2)):

> factor(x^4-4,alpha);

Представления трансцендентных функций весьма полно изучены в теории и достаточно хорошо реализованы на практике. Лежащие в основе теории принципы известны уже со времен Лиувилля, но только компьютерной алгебре потребовались точные формулировки таких теорем. Подробное изложение теории может быть найдено в [Davenport 1987]. На практике, для работы с трансцендентными функциями используют специальные опции, как и для радикалов.

Пример 2 (упрощение при дополнительных предположениях).

```
> simplify(ln(x^3), ln);
> simplify(ln(x*y));
> simplify(ln(exp(x)),ln);
```

Введем дополнительные сведения о переменных

```
> assume(x,real);
> assume(y<0);
> simplify(ln(x^3), ln);
> simplify(ln(x*y));
> simplify(ln(exp(x)));
> simplify(ln(y^3),ln);
> simplify(ln(y^4),ln);
```

Компьютерная алгебра не ограничивается конечными объектами типа полиномов. Она позволяет работать с некоторыми типами бесконечных рядов. Разумеется, компьютер реально может иметь дело только с конечным числом объектов, т.е. только с первыми членами ряда, но в принципе это количество не ограничено. В силу этого с рядами можно обращаться так же, как и с полиномами, - фактически большинство систем символьных вычислений не проводят между ними различия. Обычно, ряды строятся по системе рекурсивных вычислений (например по алгоритму Нормана [Norman 1975], который позволяет избежать потери точности).

2.1. Некоторые простые математические операции

Чтобы немного поработать с Maple рассмотрим теперь некотрые мтематические операции, основанные на алгоритмах сложения, умножения и деления.

Именно, обозначения пределов, сумм и произведений наиболее близки к обычным математическим обозначениям, а именно:

Пределы: limit или Limit

Сумма: sum или Sum

Произведение: product или Product

Различие в этих двух обозначениях можно понимать так: в первом случае вычисление проводиться немедленно, во втором случае выполнение откладывается. Более формально, первое представление отвечает значению, присвоеному этому имени (в Maple это имя с маленькой буквы). Второе представление отвечает символу без присвоения конкретного значения (в Maple это имя с большой буквы).

Разницу между ними легко понять на примерах.

Пример 3 (пределы).

```
> g:=Limit(sin(x)/x, x=0); # Это символ
> value(g);
```

Будем начинать всегда с символа, так как при этом Maple выведет математическое обозначение данного выражения.

```
> limit(sin(x)/x, x=0); # Это значение
```

Таким образом различают символ и значение!

```
> limit(exp(x), x=infinity);
> limit(exp(x), x=-infinity);
> f:=Heaviside(x);
> plot(f,x=-2..2,y=-2..2,thickness=2);
```

```
> limit(f,x=0);
> limit(f,x=0,right);
> limit(f,x=0,left);
```

Пример 4 (суммы).

```
> Sum('k^2', 'k'=0..n);
> sum('k^2', 'k'=0..n);
> sum('1/k!', 'k'=0..infinity);
> s:=Sum('1/k^2', 'k'=1..infinity);
> value(s);
> sum('1/k^2', 'k'=1..infinity);
```

Далее более сложные выражения

```
> sum('a[k]*x^k','k'=0..4);
> sum('a[k]*x^k','k'=0..n);
> Sum('k/(k+1)','k'=0..n) = sum('k/(k+1)', 'k'=0..n);
```

Эта команда достаточно интересна - посмотрите ее свойства в справочнике.

Пример 5 (произведения).

Математическая запись

```
> Product(k^2, k=1..n);
> product(k^2, k=1..n);
> product(k^2, k);
> product(a[k], k=0..4);
> product(a[k], k=0..n);
> Product(n+k, k=0..m) = product(n+k, k=0..m);
```

2.2. Составные математические объекты

Элементами составных математических объектов являются числа, имена выражения и любые другие объекты. В качестве основных составных объектов

в Maple используют: последовательности, множества, списки, таблицы и некоторые другие. Например:

1. последовательности (exprseq)

```
> s:= 1,4,9,25;
> t:= sin, cos, tan;
> whattype(s);
```

Пустая последовательность обозначается NULL. Последовательности ассоциативны, т.е. складываются

```
> s:=1,(4,9,16),25;
> t:=s,s;
```

Наиболее эффективный путь создания последовательностей - это применение специальной функции seq (не применяйте для этого циклы, данная команда более эффективна, так как не создает промежуточных последовательностей). Например:

```
> s:=seq( i^2, i=1..5 );
```

С другой стороны

```
> s:=NULL; for i from 1 to 5 do s:=s,i^2 od;
```

Как видим, цикл работает медленее и занимает больше памяти. Другие примеры:

```
> seq( x[i], i=1..5 );
> a:=38*x^2+y*x-11;
> seq(coeff(a,x,i),i=0..degree(a,x));
> seq(D(f),f=[sin,cos,tan,exp,ln]);
```

2. множества (set)

```
> s:={x,1,1-z,x};
> whattype(s);
```

Как мы видим, в множестве не может быть одинаковых элементов. Пустое множество обозначается {}. Для работы с множествами используют команды union, intersect, minus:

```
> {a,b} union {b,c};
> {a,b} intersect {b,c};
> {a,b} minus {b,c};
> a union b union a;
> {3,4} union a union {3,7};
> `union`({3,4},a,{3,7});
```

3. списки (list)

Списки отличаются от множеств тем, что в них могут присутствовать одинаковые элементы

```
> List:=[x,1,1-z,x];
> whattype(List);
> [x,y,y];
> [y,x,y];
```

Заметим, что это разные списки! Для создания списков и множеств можно использовать команду seq, это эффективнее, чем применение циклов.

```
> L := [seq(x[i],i=1..4)];
> L[2];
```

Для добавления элемента в список

```
> L := [op(L),x[5]];
```

Для выделения части списка

```
> L[-3..-2];
```

Для сокращения списка (выбрасываем второй элемент)

```
> L := subsop(2=NULL,L);
```

Список также может состоять из списков

```
> L:=[[1,2],[3,4,5]];
```

Элементами множеств и списков могут быть так же сами множества или списки.

Для работы с множествами и списками используют следующие логические операторы:

- member
- has
- convert

```
> member(y, {x, y, z});
> member(y, {x*y, y*z});
> member(x*y, [x*y, w+u, y]);
> member(w, [x, y, w, u], 'k'); k;
```

В последнем примере мы присвоили переменной k номер элемента w в данном списке. Аналог данной команды для выражений:

```
> f := (a+b^3+c)^(4/3);
> has( f, a );
> has( f, b^3 );
> has( f, b^2 );
> has( f, a+b^3+c );
> has( f, a+c );
> has( f, {c,d} );
> has( f, {d,e} );
```

Оператор преобразования

```
> convert( [1,2,3,4], `+` );
> f := seq( x[i]^i, i=1..4 );
> convert([f], `*`);
```

3. Команды ввода и вывода

Намного отвлечемся от математических операций и поговорим о программировании. Сохранить весь файл в одном из форматов Maple или в текстовом (txt, tex) форматах достаточно просто, используя меню. Достаточно часто необходимо сохранить только результат вычислений. Для этих целей существует несколько команд.

Пример 7 (команда save и некоторые другие).

Переменные x,y,z записываются в файл name.ext командой

> save x, y, z, `name.ext`:

Соответственно для ввода данных используют команду

> read `name.ext`

Если Вы хотите записать все результаты в файл, используйте команды writeto(filename), appendto(filename) или writedata, writestat, writebytes, writeline. Можно печатать результаты в файл или на экран. Для этого служат команды print, lprint и printf - печать по формату. Как обычно, при этом надо файл открыть и закрыть open, close (и это не единственные команды данного типа).

Перед тем, как записывать данные, Вы их можете конвертировать convert к необходимому Вам виду.

Для удобства, можно непосредственно переписать все в формате следующих языков: Си, Фортран и LaTeX (например, смотри fortran[procedure]).

Если Вы хотите использовать Maple для анализа и графичекой интерпретации числовой информации, то Вам потребуется считывать данные из других файлов. Для этих целей служит набор команд read, readline, readdata, readstat, readbytes, fscanf, parse.

4. Дифференцирование

Численное дифференцирование и интегрирование были одними из первых приложений для вычислительных машин. Формальное дифференцирование было реализовано на ранних этапах развития вычислительной техники в 1953 году. Дифференцирование является алгоритмической процедурой, так как знание производных элементарных функций и следующих четырех правил:

1. (a+b)'=a'+b' - дифференцирование суммы,
2. (a b)'=a' b+a b' - дифференцирование произведения,

3. (a / b)'=(a' b-a b') / b^2 - дифференцирование отношения,

4. f(g(x))'=f ' (g(x)) g'(x) - дифференцирование сложной функции,

позволяет нам продифференцировать произвольную функцию.

Фактически, настоящей проблемой при дифференцировании является упрощение результата, так как если его не упрощать, то производная от 2x+1 формально выдается в виде 0x+2*1+0. В силу этого, в современных универсальных программах алгоритм дифференцирования стал значительно сложнее, так как в данных пакетах используется достаточно широкий класс кусочно-непрерывных функций.

Алгоритм дифференцирования достаточно прост и, поэтому, попробуем реализовать этот алгоритм для дифференцирования полиномов от одной переменной

P = a[n]*x^n + a[n-1]*x^(n-1) + ... + a[0]

Замечание: Работа компьютера напоминает работу мельницы: засыпаешь зерно - получаешь муку, а засыпаешь камни - получаешь песок. Это особенно важно при работе с системами символьных вычислений. Перед началом работы с Maple необходимо понять цель работы, продумать различные подходы к решению проблемы и, как это не парадоксально, скорее всего, уже знать или предполагать конечный ответ.

Итак, для того чтобы продифференцировать полином нам надо:

1. отличать в выражении P переменную x от коэффициентов a[k] и показателей степеней n;

2. различать математические операции - сумму, произведение и степень;

3. задать правила дифференцирования.

Начнем с выделения различных частей выражения. В Maple для этих целей служит команда op(i,e) которая выделяет имя (операнд) под номером i из выражения e. Общее число операндов в выражении определяется командой nop(e).

Пример 1 (команды op - nop).

```
> v := f(x,y,z);
```

```
> nops(v);
> op(0,v);
> op(2,v);
```

Более сложный пример

```
> w := f(g(a,b),h(c,d));
> op(1,op(2,w));
> op([2,1],w);
> op([-1,-1],w);
```

Таким образом можно выделять более сложные части выражений.

```
> s := series(sin(x),x=2,6);
> op(0,s);
```

В последнем примере мы разложили функцию sin(x) в окрестности точки x=2 с точностью O(x^6).

Далее нам надо научиться определять тип переменных и выражений. Для этого используется логическая команда type(e,t) где e - выражение, а t - имя типа выражения. В ответе получаем либо "правда", либо "ложь". Типов выражений очень много и из простых типов можно составлять составные типы. Заметим, что команда whattype(e), которая должна выдавать тип выражения значительно уступает команде type по своим возможностям. Рассмотрим несколько примеров.

Пример 2 (типы выражений и переменных).

```
> type( 2, integer );
> type( a + b, `+` );
> type( a * b, `+` );
> type( a and b, `and` );
```

Это простые типы, рассмотрим составные

```
> type(x^(-2),name^integer);
> type(x^(-2),name^posint);
> type(x^(-2),algebraic^integer);
```

```
> type(x+y^(1/2)+1,`&+`(name,radical,integer));
> type(a*b*c,`&*`(algebraic,algebraic,algebraic));
> type(exp(x),exp(name));
> type([x,1],[name,integer]);
```

Так как type булевская команда, то изучим сразу же условный оператор if вида

if условие a then выполняемое выражение A

elif условие b then выполняемое выражение B

else выполняемое выражение C

Эта конструкция дает возможность в зависимости от условий a и b выполнять выражения A, B или C, в качестве которых может выступать любая последовательность Maple-команд.

Пример 3 (условный оператор).

```
> a := 3; b := 5;
> if (a > b) then a else b fi;
```

Более сложная конструкция

```
> 5*(Pi + `if`(a > b,a,b));
```

Теперь можно написать программу (процедуру) дифференцирования полиномов. Процедуры в Maple начинаются со слова proc и заканчиваются словом end. Текст процедуры может находиться в любом месте программы. После загрузки процедуры в рабочую память ее вызов осуществляется по имени. По умолчанию возвращаемым значением является значение последнего оператора из тела процедуры.

В заголовке процедуры мы присваеваем ей имя df. Входными параметрами будут имя дифференцируемого выражения p и имя независимой переменной x по которой мы его дифференцируем. После заголовка следует описание прцедуры, отделенное пробелом.

Пример 4 (процедура дифференцирования полиномов).

```
> df:=proc(p:algebraic,x:name) local u,v;
```

```
> if type(p,numeric) then 0
> elif type(p,name) then
> if p=x then 1 else 0 fi
> elif type(p,`+`) then map (df,p,x)
> elif type(p,`*`) then
> u:=op(1,p); v:=p/u;
> df(u,x)*v+df(v,x)*u
> elif type(p,anything^integer) then
> u:=op(1,p); v:=op(2,p);
> v*df(u,x)*u^(v-1)
> else ERROR(`Ошибка при дифференцировании полинома`,p)
> fi
> end:
```

Разберем структуру процедуры.

В первой строке мы определяем имя процедуры, типы входных данных p и x и вводим две внутренних переменных u,v (это можно сделать и по умолчанию).

Во второй строке задано правило: производная от числа (тип numeric) есть ноль.

В третьей строке задано правило: производная от переменной (тип name) есть dx / dx =1 или dy / dx=0.

В четвертой строке задано правило 1 о дифференцировании суммы (тип `+`). Для дифференцирования слагаемых применяется таже процедура df, к которой мы обращаемся реккурсивно, используя команду map (данная команда позволяет применить процедуру df ко всем слагаемым сразу).

В пятой строке задано правило 2 о дифференцировании произведения (тип `*`). Для дифференцирования множителей, которым присваиваются имена u и v, применяется процедура df . (в отличие от предыдущей строки мы вынуждены делать это явно)

В шестой строке задано правило дифференцирования степени переменной (тип anything^integer), при этом мы так же явно выделяем части выражения.

В седьмой строке для аварийного выхода из процедуры в случае возникновения ошибки используется команда ERROR (именно из больших букв).

Теперь можно продифференцировать что нибудь

> g:=y^5+x*y^4+z*y^3;

- df(g,y);

4.1. Встроенные процедуры дифференцирования

Для дифференцирования выражения e по переменным x, y и т.д. используют следующие две команды в среде Maple

Diff, diff, - как обычно, команды с большой и малой буквы отвечают символу и значению т.е. процедуре отложенного и немедленного выполнения;

D - позволяет найти дифференциал функции и дифференцировать процедуры;

implicitdiff - позволяет дифференцировать функции, заданные уравнением.

Пример 5 (дифференцирование).

> Diff(sin(x),x);

> diff(sin(x),x);

> Diff(sin(x),x$3);

> value('');

В последнем примере мы задали порядок дифференцирования.

Если функция не задана явно, то

> diff(f(x),x);

> diff(f(x,y),x,y);

Аналогично вычисляются и дифференциалы

> D(sin);

> D(ln);

> D(f);

> D(f)(0); # Производная от f вычисляется в точке 0

> D(D(f));

> (D@@2)(f); # Дифференциал старшего порядка как композиция простых дифференциалов

> D(f@g); # Дифференциал от композиции двух функций

Эта комманда позволяет дифференцировать и составные выражения. Например, предположим, что a -это число

> assume(a,constant);

> D(2*sin(a)*f);

Дифференциал можно преобразовать в производную и обратно

> f := D(y)(x) - a*D(z)(x);

> convert(f, diff, x);

> f := diff(y(x), x$2);

> convert(f, D);

Рассмотрим уравнение, которое определяет x(y) или y(x)

> f := x^2+y^3=1

Продифференцируем y по x или x по y

> implicitdiff(f,y,x);

> implicitdiff(f,x,y);

> implicitdiff(f,y,z);

> implicitdiff(f,y(x),x);

> implicitdiff(f,y,x,x);

4.2. Определение собственных функций

Необходимо отличать выражение вида f:=x^n+y с четко определенным значением от отображения (функционала) вида переменные -> результат, которое

является символом. Внутреннее представление данного отображения эквивалентно написанию процедуры вида proc(переменные) option operator, arrow; результат end

Рассмотрим, на примерах, различия между этими понятиями

Пример 7 (выражения).

```
> restart:
> f:=x^3+y^4+z;
```

с подстановкой и дифференцированием вида

```
> subs(x=1,y=5,z=-1,f);
> diff(f,x);
> Diff(f,y); value('');
```

Пример 8 (отображения).

```
> restart:
> F := (x -> sin(x));
> F(t);
> g:=(x)->x^2+y^4+z;
```

с подстановкой и дифференцированием вида

```
> g(1);
> D(g);
> D(g)(3);
```

Для случая многих переменных это так же несложно

```
> f := (x,y) -> exp(x*y);
> f(1,1);
> D[1](f);
> D[2](f);
```

Данные определения выражения и отображения отличаются не только способом вычисления в заданной точке и дифференцированием, но и рядом других свойств. Например, с помощью команд map и map2 можно применять заданное отображение к матрицам, спискам или другим более сложным объектам

Пример 9 (команда map).

```
> restart:
> map(f, x + y*z);
> map(f, y*z);
> map(f, {a,b,c});
```

Зададим какое-нибудь отображение

```
> f:=x-> x^5+1;
> L:=[1,-2,x,y,9];
> map(f,L);
```

Более сложное использование данных команд

```
> restart:
> map2( map, f, g(a+b, h(c)+d) );
```

Выражение можно всегда переопределить как функционал командой unapply. Фактически, в Maple даже дифференциал определён таким образом D(f) = unapply(diff(f(x), x), x). Напомним, что дифференциал преобразуется к производной и обратно командой convert.

Пример 10 (выражения и отображения).

```
> restart:
> q := x^2 + y^3 + 1;
> f := unapply(q,x);
> f(2);
> g := unapply(q,x,y);
```

Определим процедуру по правилу

```
> f := proc(x) local t1,t2;
> t1 := x^2;
> t2 := sin(x);
> 3*t1*t2+2*x*t1-x*t2
> end:
```

Вычислим производную от f по аргументу x

> D(f);

Проверим вычисление производной

> D(f)(x) - diff(f(x),x);

Следующим отличием отображений от выражений является то, что оператор композиции @, можно применять только для отображений. Данный оператор позволяет не только создавать сложные функции, но и является наиболее эффективным для создания обратных функций.

Пример 11 (композиция функций).

> sin@cos)(x);

> (sin@arcsin)(x);

> sin@arcsin;

> simplify(");

> sin@@0;

> sin@@1;

> (sin@@2)(x);

> (D@@2)(ln);

Обратные функции строяться аналогично

> sin@@(-1);

> ln@@(-3);

> LambertW@@(-1);

Для всех встроенных функций можно загрузить таблицу обратных функций (сразу всю!), но это занимает больше памяти, чем использование композиции.

> readlib(invfunc);

> invfunc[sin](1);

> Invfunc[sin](1);

> Invfunc[exp](x);

5. Интегрирование

5.1. Встроенные команды интегрирования

В стандартной (загруженной по умолчанию) библиотеке находяться процедуры Int (expr, options) и int (expr,options), т.е. символ и значение. Синтаксис и опции можно найти в справочнике int.

Пример 1 (определенные и неопределенные интегралы).

```
> f:=sin(x):
> g:=Int(f,x);
> value(g);
> int(f,x);
```

Таким образом, мы видим,. что обозначения максимально приближены к обычной математической записи. Для вычисления определенного интеграла надо дополнительно указать пределы интегрирования

```
> f:=x*exp(x):
> g:=Int(f,x=0..1);
> value(g);
> int(f,x=0..1);
```

Рассмотрим более сложные интегралы

```
> f:=sin(x^3):
> g:=int(f,x=0..Pi);
> h:=evalf(g);
```

или

```
> g:=Int(x/(x^3-x^2+1),x);
> value(g);
```

здесь сумма берется по %1 - корням полинома в скобках.

Иногда, при интегрировании получаются многозначные функции, при этом, по умолчанию, Maple использует только одно значение. Список многозначных функций и их принципиальные ветви можно посмотреть следующей командой

> restart: readlib(branches):

> branches(ln);

Рассмотрим интегралы с параметрами

> f:=exp(-a*x)*ln(x)*sqrt(x);

> int(f,x=0..infinity);

Как видим, этот интеграл неопределен при всех значениях параметра. Сделаем дополнительные предположения

> assume(a<0): int(f,x=0..infinity);

Сравним с

> assume(a>0): int(f,x=0..infinity);

Полученные при интегрировании эллиптические функции, зависящие от радикалов можно упростить

> answer := int(1/sqrt(sin(x)), x=0..Pi/2);

> radnormal(answer);

Интегралы, которые не выражаются через алгебраические функции, можно представить как ряд

> int(exp(x^3), x);

> series(", x=0);

В общем случае синтаксис для численного интегрирования имеет вид

evalf (Int (f, x = a..b , digits , flag)),

где digits- точность вычисления, и flag- код численного метода, см. int[numerical]

> int(exp(v-v^2/2)/(1+1/2*exp(v)), v = -infinity..infinity);

> evalf(", 20);

Здесь digits равно 20. Другой пример - интегрирование гамма функции с использованием алгоритма Ньютона-Котеса:

> evalf(Int(1/GAMMA(x), x = 0..2, 10, _NCrule));

Более сложные опции позволяют вычислять главное значение интеграла и реализуют некоторые дополнительные возможности

```
> int(1/x^3, x=-1..2, 'CauchyPrincipalValue');
> int( 1/(x+a)^2, x=0..2, 'continuous' );
```

В принципе, различные эвристические методы, такие как интегрирование по частям и замены переменных, не используются явно, например.

Рассмотрим несколько интегралов, которые берутся по частям или с использованием различных замен переменных

Пример 2 (интегрирование по частям и другие методы) .

```
> int(x*ln(x),x);
> int(x*sin(x^2),x);
> Int(x^2/(sqrt(1-x^2))^(3/2),x):
> value('');
```

Тем не менее, для обучения, мы можем использовать подобные команды явно. Эти команды содержатся в пакете student

Пример 3 (интегрирование с пакетом student).

Для начала мы очистим рабочую память и загрузим данный пакет.

```
> restart:
> with(student);
```

При загрузке любого пакета Maple выдает список всех загружаемых команд.

Замены переменных:

```
> f:=Int(sqrt(1-x^2),x=a..b);
> g:=changevar(x=sin(u),f,u);
```

Здесь мы вводим новую независимую переменную интегрирования u, связанную со старой переменной уравнением x=sin(u) (пределы интегрирования изменились автоматически!) Посмотрим на результат

```
> value(g);
```

В двойных интегралах интегралах это позволяет, например, перейти к полярным координатам:

```
> changevar({x=r*cos(t),y=r*sin(t)}, Doubleint(1,x,y),[t,r] );
```

Интегрирование по частям:

напомним, что правило интегрирования по частям записывается следующим образом

> f:=Int (u,v)=u*v-Int(v,u);

Синтаксис данной команды intparts(f,u)

> a:=Int(x^2*exp(x),x);

Проинтегрируем по частям

> b:=intparts(a,x^2);

проинтегрируем по частям второй раз

> c:=intparts(b,2*x);

и получим результат

> d:=value(c);

Другие полезные команды для интегрирования из данного пакета

> integrand(Int(h(x),x));

и

> Doubleint(h(x,y),x,y);

> Tripleint(h,x=1..n,y=2..4,z=w..u);

Кроме этого, в пакете student реализовано несколько численных методов:

leftbox, leftsum , rightbox, rightsum, middlebox, middlesum, simpson, trapezoid,

Синтаксис для всех команд одинаков, рассмотрим только один пример

Пример 4 (численное интегрирование).

> restart: with(student):

> f:=x^5*ln(x);

Явное интегрирование данной функции можно посмотреть в справочнике, мы же сравним численные значения

> ex:=evalf(int(f,x=1..3));

> s:=evalf(simpson(f,x=1..3,16));

> t:=evalf(trapezoid(f,x=1..3,16));

```
> ls:=evalf(leftsum(f,x=1..3,16));
```

Покажем это на рисунке

```
> leftbox(f,x=1..3,16,color=red,shading=BLUE);
```

Аналогично и для других команд

```
> rs:=evalf(rightsum(f,x=1..3,16));
> rightbox(f,x=1..3,16,color=red,shading=BLUE);
```

и

```
> ms:=evalf(middletsum(f,x=1..3,16)));
> middlebox(f,x=1..3,16,color=red,shading=BLUE);
```

5.2. Двойные и тройные интегралы в Maple

Задача 1. Найти площадь ограниченную параболой x=2y^2 и линией x=y+3.

Решение 1.

Нарисуем эту область

```
> restart:
> x1:=2*y^2: x2:=y+3:
> plot({[x1,y,y=-2..2],[x2,y,y=-2..2]});
```

Найдем точки пересечения кривых (пределы интегрирования)

```
> solve({x=2*y^2,x=y+3});
> s:=Int(Int(1,x=2*y^2..y+3),y=-1..3/2);
> value(s);
```

Задача 2. Найти среднее значение функции f=y ch (x) над областью, ограниченной кривыми x=0 , y=2 , y=sqrt(x).

Решение 2.

Посмотрим на эту область

```
> restart:
> plot({[0,t,t=-1..2.5],
```

>[t,2,t= 0..5],

>[t,sqrt(t),t=0..5]},thickness=2);

Область описывается, как 0<y<2, 0<x<y^2. Вычислим среднее

> s:=Int(Int(y*cosh(x),x=0..y^2),y=0..2)/

>Int(Int(1,x=0..y^2),y=0..2);

> value(s); evalf(");

Задача 3. Найти объем тела ограниченного поверхностями f=x^2+2y^2 - сверху, и g=6-2x^2-y^2 - снизу, над кругом единичного радиуса x^2+y^2 <1

Решение 3.

Нам необходимо найти объем следующей фигуры

> restart: with(plots):

> f:=x^2+2*y^2: g:=6-2*x^2-y^2:

Нарисуем поверхности

> surf:=plot3d({f, g },x=-1.2..1.2,y=-1.2..1.2):

и цилиндр единичного радиуса

> cyl:=plot3d([cos(t),sin(t),z],t=0..2*Pi,z=0..6):

Выведем их на экран

> display3d({surf,cyl});

Чтобы найти объем данной фигуры необходимо вычислить интеграл

> Int(Int(g-f,y=-sqrt(1-x^2)..sqrt(1-x^2)),x=-1..1);

> value(");

Задача 4. Используя сферические координаты найти объем тела, ограниченного сферой с x^2+y^2+z^2=4вырезанным в ней цилиндром x^2+y^2=1.

Решение 4.

В сферических координатах сфера задается как r=2, цилиндр как r sin(f)=1 . Нам надо выразить декартовы координаты x,y,z через сферические, подставив данные значения r для сферы и для цилиндра

> restart: with(plots):

> rs:=2; rc:=1/sin(phi);

```
> x:=sin(phi)*cos(theta);
> y:=sin(phi)*sin(theta);
> z:=cos(phi);
```

Зададим точку на сфере и цилиндре

```
> sf:=[rs*x,rs*y,rs*z]:
> cyl:=[rc*x,rc*y,2*phi-Pi]:
```

Нарисуем сферу и цилиндр

```
> plot3d({sf,cyl},theta=0..2*Pi,phi=0..Pi);
```

Вычислим пределы интегрирования, решив систему уравнений

```
> _EnvAllSolutions := true:
> solve({r=2,r*sin(phi)=1});
```

здесь B1 и Z1 целые числа. Таким образом, наша фигура может быть получена вращением следующего сегмента вокруг оси Z

```
> plot({[rs*sin(v),rs*cos(v),v=Pi/6..5*Pi/6],
> [1,v,v=-sqrt(3)..sqrt(3)]},-3..3,-2..2);
```

Вычислим ее объем

```
> Int(Int(Int((rho^2)*sin(phi),
> rho=1/sin(phi)..2),phi=Pi/6..5*Pi/6), theta=0..2*Pi);
> value(");
```

Задача 5. Используя цилиндрические координаты найти объем тела, ограниченного поверхностью (x^2+y^2)^5/2=(x^2-y^2)^2и плоскостями z=0, z=2.

Решение 5.

В цилиндрических координатах поверхность параметризуется как

r = (cos^2 (q) -- sin^2 (q)) ^ 2. Зададим параметрически ее

```
> restart:
> xs:=cos(2*theta)^2*cos(theta):
> ys:=cos(2*theta)^2*sin(theta):
> surf:=plot3d([xs,ys,z],theta=0..2*Pi,z=-1..3,grid=[60,5]):
```

и две плоскости

```
> planes:=plot3d({0,2},x=-1.5..1.5,y=-1.5..1.5):
```

Нарисуем их

```
> with(plots):
> display3d({surf,planes});
```

Посмотрим сечение

```
> polarplot((cos(theta)^2-sin(theta)^2)^2,theta=0..2*Pi);
```

Область интегрирования определить легко

0 < z < 2 , 0 < q < 2 p , 0 < r < (cos^2 (q) -- sin^2 (q)) ^ 2

Объем тела равен

```
> v:=Int(Int(Int(rho,rho=0..(cos(theta)^2-sin(theta)^2)^2),
> theta=0..2*Pi),z=0..2);
> value(v);
```

Как мы видим, главная проблема в определении пределов интегрирования. Существует несколько библиотек программ, которые позволяют находить эти пределы. Эти же пакеты предназначены для изучения векторных полей (можно посмотреть доказательство теорем Стокса, Грина и др.). Смотри www-сервер Maple Software.

6. Линейная алгебра

В Maple реализованы специальные возможности для работы с таким классическим математическим курсом как линейная алгебра. Основные команды собраны в пакетах linalg и student.

6.1. Основные операции над матрицами

Пример 1 (создание матриц и векторов).

Заметим сразу, что матрицы и вектора по сути своей это переменные типа массив и список. Основное отличие в том, что добавлены операции сложения и

умножения. В силу этого, при загрузке пакета linalg система печатает предупреждение о переопределении ряда операций. Итак, разные определения матриц

```
> restart: with(linalg):
> A:=matrix([[5,w],[f(x),6]]);
> B:=matrix(2,2,[1,2,3,4]);
> f:=(i,j)->x^(a*i+b*j);
> C:=matrix(2,2,f);
> dC:=map(diff,C,x);
```

Частично заданные матрицы

```
> A:=matrix(2,2,[x,y]);
```

В этом примере A[2,1] означает неопределенность данного элемента матрицы в данный момент

```
> B:=matrix([col(A,1..2)]);
```

Некоторые матрицы специального вида

```
> A:=diag(a,b,c,d);
> B:=array(1..4,1..4,identity): eval(B);
> B:= array(symmetric, 1..3,1..3, [ [0,x,y], [x,0,z] ] ):
> eval(B);
```

В этом примере знак вопроса означает неопределенность элемента матрицы. Тем не менее можно работать с этой матрицей. Например

```
> B[1,2];
> B[3,3];
```

Различные определения векторов

```
> a:=vector([5,4,3,2,1]);
> b:=vector(4,0);
> c:=vector(4);
> d:=
```

Перечислим некоторые команды, позволяющие получать информацию о матричных и векторных переменных, изменять параметры матриц и векторов.

Пример 2 (структура матриц и векторов).

Размеры матриц определяются таким образом

```
> restart: with(linalg):
> A:= matrix(3,4);
> rowdim(A);
> coldim(A);
```

Строки и столбцы матриц можно

```
> A:= matrix(3,3,[1,2,3,4,5,6,7,8,9]);
```

выделять

```
> row(A,2);
> col(A,2..3);
```

удалять

```
> B:=delrows(A, 2..3);
> C:=delcols(A, 1..1);
```

добавлять (команда addrow(A, r1, r2, m) возвращает матрицу B в которой строка r2 заменена на строку m*row(A,r1) + row(A,r2)):

```
> B:=addrow(A,1,2,10);
> C:=addcol(A,1,2,-x);
```

умножать

```
> B:=mulrow(A, 2, 2);
> C:=mulcol(A, 2, x);
```

переставлять

```
> A:= array( [[1,2,x],[3,4,y]] );
> swaprow(A, 1, 2);
> swapcol(A, 2, 3);
```

Матрицы можно соединять, выделять из них подматрицы и т.д.

Перейдем к командам, которые реализуют основные векторные и матричные операции. Начнем с самых простых и часто используемых операций линейной алгебры.

Пример 3 (операции сложения и умножения).

Для сложения или умножения матриц и векторов используют многоцелевую команду

evalm. Например, зададим матрицы и вектор

```
> restart: with(linalg):
> A:= array( [[1,2],[3,4]] );
> B:= array( [[0,1],[1,0]] );
> c:= array( [4,5] );
```

и посмотрим на синтаксис этих операций

```
> evalm(A+B);
> evalm(A&*B);
> evalm(A&*c);
```

Так как опреция умножения над полем метриц некомутативна, то для этой операции предусмотрены и другие команды

```
> multiply(B,A);
> multiply(B,c);
```

а для векторов еще одна операция

```
> innerprod(c,B);
```

Умножить матрицу (вектор) на скаляр позволяет операция

```
> A:= array( [[1,2],[3,4]] );
> B:=scalarmul(A, x);
```

Для векторов добавлены еще две операции обычного и векторного произведения

```
> u:= vector( [1,x,y] );
> v:= vector( [z,1,1] );
> dotprod(u, v);
> crossprod(u, v);
```

Существует много встроенных команд для работы только с матрицами. Рассмотрим часть из них.

Пример 4 (всторенные операции с матрицами).

Определим матрицу

> restart: with(linalg):

> A := matrix(3,3, [1,2,3,x,y,z,a,b,c]);

Ее определитель, ранг и след равны

> det(A);

> rank(A);

> trace(A);

Сопряженная, транспонированная и обратная к ней равны

> B:=adj(A); # или adjoint

> C:=transpose(A);

> F:=inverse(A); # или evalm(1/A)

Можно найти эрмитову транспонированную, и некоторые другие матрицы, мы не будем на этом останавливаться.

6.2. Собственные числа и вектора

Рассмотрим теперь проблему нахождения собственных значений и собственных векторов матриц. В Maple данная проблема может быть решена несколькими способами.

Пример 6 (собственные числа и вектора).

Обсудим проблему нахождения собственных чисел и векторов. Для начала загрузим пакет линейной алгебры и определим произвольную матрицу

> restart:with(linalg):

> A:= matrix([[1,2,3],[2,3,4],[4,5,6]]);

По определению, собственные числа матрицы A являются корнями характеристического полинома det(A-lambda*I) по переменной lambda. Найдем этот полином

> cp:= det(A-lambda);

Корни кубического полинома легко найти аналитически

> solve(cp,lambda);

Можно найти эти значения непосредственно из матрицы

> e:= eigenvals(A);

Однако, попробуем заставить систему вычислить эти собственные значения в форме чисел с плавающей запятой

> B:= map(evalf,A);

> eigenvals(B);

Как видим, из-за погрешностей округления нулевое собственное значение перестало быть нулем. Мораль: система Maple (Matemetica) не предназначена для численных расчетов.

Найдем теперь собственные вектора матрицы. По определению это решения уравнения A x = lambda x, т.е. данные вектора определяют пространство вырождения матрицы A - lambda I.

> nullspace(e[1]-A);

> nullspace(e[2]-A);

> nullspace(e[3]-A);

Естественно, что существует возможность определить собственные числа и вектора одновременно

> e := eigenvects(A,radical);

Ответ представляется списком, состоящим зи списков. Для выделения вложенных элементов используют общие правила

> e1:= e[1][1]; v1 := e[1][3][1];

> evalm((e1 - A) &* v1);

Аналогично можно работать с матрицами, определенными аналитически, т.е. не числовыми. Например

> T:= toeplitz([a,b,c]);

> eigenvals(T,radical);

> e:= eigenvects(T,radical);

Однако, для корней полиномов старших степеней невозможно найти аналитическое представление. В силу этого, результаты могут быть представлены достаточно неожиданным способом. Рассмотри матрицу

```
> A:= matrix(
> [[-a1/a2, 1/a2, 0, 1/(R*a2)],
> [1, 0, 0, 0],
> [-q1*a1/a2, q1/a2, 0, a2 + q1/(a2*R)],
> [q2, 0, 1, a1]]);
>
```

Характеристический полином четвертого порядка невырожден

```
> cp:= det(lambda-A);
```

т.е. не факторизуется на множители

```
> factor(cp);
```

Все корни подобных полиномов Maple представляет как единое выражение.

```
> solve(cp,lambda);
```

Для собственных векторов мы также получим одно выражение

```
> eigenvects(A);
```

Второй аргумент показывает, что все четыре собственных значения невырождены. Таким образом, можно найти корни полинома численно и подставить их в данную формулу.

Рассмотрим теперь пример, в котором характеристический полином факторизуется

```
> B:= matrix(6,6,
> [0,0,0,0,0,a,
> 0,1,0,0,0,-b,
> 0,0,0,-a,b,0,
> 0,0,-c,0,0,0,
> 0,0,d,0,0,0,
```

> c,-d,0,2,0,0]);

>

> cp:= charpoly(B,lambda);

> f:= factor(cp);

Как видно lambda=0 является собственным значением. Построим характеристическую матрицу C=lambda I -B при lambda=0

> C:= charmat(B,0);

и найдем соответствующий собственный вектор.

> nullspace(C);

Аналогично поступим и со следующими двумя собственными значениями. Например, для + sqrt(ac+bd) получим

> C:= charmat(B, sqrt(a*c+b*d));

> nullspace(C);

Теперь перейдем к кубическому полиному. Присвоим корням этого полинома имя beta

> alias(beta=RootOf(op(3,f),lambda));

и найдем соответствующие вектора

> C := charmat(B,beta);

> nullspace(C);

Для сравнения получим все эти ответы сразу, используя многоцелевую команду

> eigenvects(B);

Во-первых, следует заметить, что вектора, отвечающие кубической части полинома отличаются от полученных ранее. Это связано с тем, что собственные вектора определяются с точностью до скалярного множителя. Во вторых, когда выдается так много информации сразу, то в ней бывает достаточно сложно разобраться. Так что, это уже дело вкуса какой из путей выбрать для данных вычислений.

Рассмотрим теперь матрицы с вырожденными собственными значениями

```
> A:= matrix(4,4,[0,1,0,0,-u^2,2*u,0,0,-u*v,v,u,0,-u*t,t,0,u]);
>
> e:= eigenvects(A);
```

В ответе мы получили четыре совпадающих собственных значения и базис из трех собственных векторов. Это означает, что два вектора совпадают.

Проверим наш ответ. Соберем из векторов матрицу

```
> X:= augment(e[3][1], e[3][2], e[3][3] );
```

и подставим ее в наше уравнение

```
> evalm((u-A) &* X)
```

6.3. Приведение матриц

Практически все алгоритмы приведения матриц к различным специальным формам можно найти в Maple. Ограничимся несколькими замечаниями.

Пример 7 (приведение матриц).

Зададим матрицу

```
> restart: with(linalg):
> A:=matrix(3,3,[x,y,z,0,9,8,7,6,5]);
```

Применим алгоритм гауссова исключения

```
> gausselim(A);
```

алгоритм гауссова исключения без деления

```
> ffgausselim(A);
```

Для работы с символьными матрицами последняя команда предпочтительнее, поскольку исключает деление на ноль.

Приведение матрицы к треугольному виду при помощи алгоритмя Гаусса-Жордана

```
> A:= array( [[4,-6,1,0],[-6,12,0,1],[-2,6,1,1]] );
> gaussjord(A, 'r');
```

Эрмитова часть матрицы, зависящей от x, выделяется командой

```
> H:= inverse(hilbert(2,x));
> hermite(H,x);
```

Для определения размерности векторного пространства, порожденного столбцами (строками) матрицы и векторов, заметающих это пространство, используют команды

```
> B:=array( [[0,0,0,1,0],[0,x,0,0,0],[0,0,y,y,0],[0,x,0,1,0]] );
> rowspace(B, 'd');
> d;
> colspace(B, 'd');
> d;
> rowspan(B);
> colspan(B);
```

Алгоритм разложения Холецкого

```
> S:= matrix(3,3, [1,2,3,0,1,1,0,0,4]);
> A:= evalm(transpose(S)&*S);
> R := cholesky(A);
```

LU разложение матриц

```
> A:= matrix([[1,-2,3,1],[2,k,6,6],[-1,3,k-3,0]]);
> x:= LUdecomp(A,L='l',U='u',U1='u1',R='r',P='p',det='d',rank='ran');
```

с побочными результатами

```
> ran;
> evalm(l);
> evalm(x);
```

и QR разложение

```
> A:= matrix(3, 3, [1, 2, 3, 0, 0, 1, 2, 3, 4]):
> linalg[QRdecomp](A, Q='q', rank='r');
```

6.4. Системы линейных уравнений

Если системы уравнений заданы в матричной форме, удобнее использовать пакет linalg.

Пример 8 (решение систем уравнений).

Команда linsolve(A, B) находит вектор (матрицу) X который удовлетворяет уравнению A X = B.

Отметим так же команду leastsqrs(A, b), которая позволяет получить приближение к решению данного уравнения по методу наименьших квадратов, т.е. минимизируя норму вектора (A x - b, 2).

```
> restart: with(linalg):
> A:= matrix([[1,2],[1,3]]);
> b:= vector([1,-2]);
> linsolve(A, b);
> B:= matrix( [[1,1],[-2,1]] ):
> linsolve(A, B);
> A:= matrix([[5,7],[0,0]]):
> b:= vector([3,0]):
> linsolve(A, b, 'r');
> r;
```

В последнем примере мы вычислили так же ранг матрицы A.

```
> A:= array([[1,-1,1],[1,1,-2],[2,0,-1]]);
> b:= vector([1,2,4]);
> leastsqrs(A,b);
> leastsqrs(A,b,'optimize');
```

Дифференциальные операторы векторного анализа так же реализованы в данном пакете линейной алгебры. Мы можем вычислять ротор, лапласиан, градиент, дивиргенцию и многое другое.

Контрольные вопросы:

1. Знать и уметь пользоваться основными командами Maple приведенными выше
2. Самостоятельно рассмотреть такие темы как: «Решение уравнений» и «Решение дифференциальных уравнений».
3. Выделить список основных команд с которыми вы познакомились выполняя задание 2. Уметь ими пользоваться.

Задания:

1. Получить у преподавателя сложную функцию F(G(x,y),z))
2. Продиффиренцировать ее (найти производные всех порядков)
3. Проверить правильность полученных производных путем интегрирования
4. Получить матрицу A , (i,j=1..4) и столбец свободных членов B (i=1..4)
5. Из полученных матриц составить систему линейных уравнений вида A*X=B, и решить ее.
6. Для выполнения заданий 1..5 использовать приложение Maple.

СПИСОК ЛИТЕРАТУРЫ

1. Фигурнов В. Э. – IBM PC для пользователя. Краткий курс. – М.: Инфра - М,, 1998. – 480с.:ил.
2. Борланд Р. Знакомство с Windows 98. – М.: Microsoft Press Русская редакция, 1997, 376с.
3. Леоньтьев Ю. – Самоучитель Word 2000. – СПб: Издательство «Питер», 2000. – 313с.:ил.
4. Леоньтьев Ю. – Самоучитель Excel 2000. - СПб: Издательство «Питер», 2000. – 265с.:ил.
5. Давенпорт Дж. – Компьютерная алгебра. Символьные и алгебраические вычисления. – М.: Мир, 1996. – 354с.:ил.
6. Очков В. Ф. – Mathcad 8 Pro для студентов и инженеров. – М.: Аст-пресс, 1998, 270с.:ил.
7. Сворт Б. – Современные системы оптического распознавания текстов в использовании. – СПб: Издательство «Питер», 2000. – 420с.: ил.

8. Шикин Е.В. – Электронный офис – средство обработки документов. – М.: Инфорком-Пресс, 1999. – 520с.