

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
Витебский государственный технологический университет

Е.Ю. Вардомацкая

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ТЕХНИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Конспект лекций

для студентов специальностей
1-25 01 04 «Финансы и кредит»,
1-25 01 07 «Экономика и управление на предприятии»,
1-25 01 09 «Товароведение и экспертиза товаров»,
1-27 01 01 «Экономика и организация производства»,
1-26 02 03 «Маркетинг»

Витебск
2022

УДК 004
ББК 32.97
В18

Рецензенты:

к.ф.-м.н., заведующий кафедрой «Прикладное и системное программирование» УО «ВГУ им. П.М. Машерова» Ермоченко С.А.;

к.т.н., заведующий кафедрой «Экономика и информационные технологии»
УО ФПБ «Витебский филиал Международного университета МИТСО»
Замостоцкий Е.Г.

Рекомендовано к опубликованию редакционно-издательским
советом УО «ВГТУ», протокол № 10 от 29.06.2022.

Вардомацкая, Е. Ю.

В18 Информационные технологии. Техническое и программное обеспечение:
конспект лекций / Е. Ю. Вардомацкая. – Витебск : УО «ВГТУ», 2022. – 116 с.
ISBN 978-985-481-721-7

Конспект лекций составлен в соответствии с действующей учебной программой дисциплины «Информационные технологии» для экономических специальностей дневной и заочной форм обучения и учитывает требования к профессиональным компетенциям специалистов экономического профиля и опыт преподавания данной дисциплины в УО «ВГТУ». В издании в определенной логической последовательности рассмотрены основные вопросы, обусловленные необходимостью практического использования информационных технологий и пакетов программ в профессиональной деятельности специалистов экономического профиля.

Адресуется студентам экономических специальностей, а также студентам технологических специальностей вузов, магистрантам и аспирантам, всем, интересующимся возможностью использования компьютерных информационных технологий в профессиональной деятельности.

УДК 004
ББК 32.97

ISBN 978-985-481-721-7

© УО «ВГТУ», 2022

Содержание

Введение	4
Лекция 1 ПРЕДМЕТ И ОСНОВНЫЕ ПОНЯТИЯ ИТ.....	5
Лекция 2 ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ.....	15
Лекция 3 КОМПЬЮТЕРНЫЕ СЕТИ	21
Лекция 4 СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИТ.....	40
Лекция 5 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ.....	46
Лекция 6 ТЕХНОЛОГИИ И СИСТЕМЫ ОБРАБОТКИ ТАБЛИЧНОЙ ИНФОРМАЦИИ.....	55
Лекция 7 КОМПЬЮТЕРНАЯ ГРАФИКА.....	75
Лекция 8 ПАКЕТЫ ДЛЯ МАТЕМАТИЧЕСКОЙ ОБРАБОТКИ ДАННЫХ.....	84
Лекция 9 ИНСТРУМЕНТАЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ.....	99
Литература	115

ВВЕДЕНИЕ

Целью изучения раздела «Техническое и программное обеспечение» учебной дисциплины «Информационные технологии» является подготовка студентов к использованию информационных технологий в качестве инструмента для решения научных и практических задач в своей предметной области на высоком профессиональном уровне.

При изучении этого курса студенты должны получить теоретические знания об основных понятиях информационных технологий (ИТ), техническом обеспечении ИТ, сетевых информационных технологиях, системном и прикладном программном обеспечении (ПО), системах программирования.

Кроме теоретических знаний студенты должны получить и практические навыки в области применения современных концепций и компьютерных информационных технологий для решения задач, связанных с будущей профессиональной деятельностью. Уметь определять конфигурацию персонального компьютера; использовать сервисы сети Internet при решении профессиональных задач; разрабатывать и публиковать Web-страницы; разрабатывать макросы и модули на языке VBA; обрабатывать текстовую информацию средствами текстового редактора; обрабатывать экономическую информацию средствами табличного процессора MSExcel.

Конспект лекций включает в себя как теоретический материал по ключевым темам дисциплины, так и практические примеры с пояснениями. Материал лекций четко структурирован, излишние подробности отсутствуют. Большое внимание уделяется возможностям использования компьютерных технологий для обработки экономической информации и решения задач экономики и управления, характеристике современных пакетов прикладных программ специального назначения.

Для удобства работы и повышения степени усвоения теоретического материала к конспекту лекций прилагается диск с решением демонстрационных примеров по каждой теме. Примеры, записанные на диске, являются интерактивными, то есть пользователь имеет возможность адаптировать их под разный набор исходных данных и, как следствие, самостоятельно провести анализ полученных результатов.

Автор выражает благодарность студенткам ФЭиБУ Асоблевой Полине, Емельяновой Надежде, Кужель Дарье, Телице Анастасии и Григорук Анне за помощь в отладке демонстрационных примеров и выражает надежду, что предлагаемый конспект лекций будет полезен не только студентам, но и магистрантам и аспирантам, изучающим эту дисциплину.

Лекция 1. ПРЕДМЕТ И ОСНОВНЫЕ ПОНЯТИЯ ИТ

План лекции

1. Предмет и задачи курса «Информационные технологии» (ИТ).
2. Понятие информационной технологии.
3. Составляющие информационных технологий.
4. Свойства информационных технологий.
5. Классификация информационных технологий.
6. Обработка информации.
7. Проблемы и перспективы развития информационных технологий.

1 Предмет и задачи курса «Информационные технологии» (ИТ)

Первая часть курса «**Информационные технологии**» (ИТ) является базовой в 2-семестровом цикле подготовки студентов экономических специальностей в области использования информационных технологий в профессиональной деятельности. В этой части дисциплины изучаются офисные технологии обработки графической, текстовой и табличной информации, технологии работы с математическими пакетами, основы языка VBA для программирования приложений MS Office, основы сетевых технологий и технологий работы с сервисами INTERNET. Эта часть курса призвана не только систематизировать получаемые студентами знания, но и привести их в соответствие с требованиями времени и действующими стандартами.

Цель изучения данной дисциплины – подготовка студентов и других категорий обучающихся к использованию современных компьютерных информационных технологий как инструмента для решения научных и практических задач в своей предметной области на высоком профессиональном уровне, а также к участию в разработке и внедрении этих технологий в рамках корпоративных информационных систем на уровне постановки задачи, и контроль над ее решением, что так необходимо для современного руководителя.

В результате изучения первой части курса ИТ студенты должны:

- иметь представление о техническом обеспечении современных информационных технологий, особенностях его организации и эксплуатации, перспективах его развития;
- ознакомиться с математическим, программным, организационным и другими видами обеспечения компьютерных информационных технологий;
- получить представление о развитии компьютерных информационных

технологий в своей предметной области;

- научиться применять средства современных компьютерных технологий при решении задач в своей предметной области;
- уметь осуществлять навигацию и поиск информации в сети INTERNET;
- уметь работать с электронными сервисами и службами сети INTERNET.

2 Понятие информационной технологии

Под *информационной технологией* понимают совокупность методов, производственных и программно-технологических средств, объединенных в технологическую цепочку, обеспечивающую сбор, хранение, обработку, вывод и распространение информации. Информационные технологии предназначены для снижения трудоемкости процессов использования информационных ресурсов.

Компьютерные информационные технологии подразумевают использование средств вычислительной техники и сетевых технологий для реализации широкого круга задач:

- подготовка текстовых документов;
- создание банков данных, их обработка и использование;
- автоматизация финансовых и научных расчетов;
- верстка книг, газет, журналов;
- создание и редактирование графических и фотоизображений;
- создание мультфильмов;
- создание электронных энциклопедий и электронных версий других книг;
- создание и компьютерная обработка звука;
- создание и компьютерная обработка телевизионных и видео произведений;
- создание и использование телекоммуникационных компьютерных сетей;
- компьютерное проектирование различных механизмов, архитектурных строений, создание географических карт и др.;
- моделирование природных, технических и других процессов.

3 Составляющие (структура) информационных технологий

Информация – это набор сведений о свойствах объекта или процесса, усваиваемый субъектом в форме знаний.

Всю информацию, которой пользуется человек, можно разделить на следующие виды:

- *математическая* – это любая информация, связанная с числами и

формулами. Это может быть, кроме собственно математической, и физическая, и статистическая информация. Математическую информацию можно обрабатывать при помощи различных счетных машин и аппаратов и хранить на бумаге в виде записей и книг;

- *текстовая* – эту информацию можно записывать на бумагу от руки или при помощи печатных машинок и типографского оборудования и хранить на бумаге (рукописи, документы, книги, газеты и т.д.);

- *графическая* – эту информацию можно обрабатывать при помощи различных изобразительных средств и способов (изобразительное искусство, фотография) и хранить в виде картин, рисунков, скульптур, фотокарточек и т.д.;

- *звуковая* – эту информацию можно обрабатывать при помощи магнитофонной записи и хранить на магнитных лентах, пластинках и звуковых компакт-дисках;

- *видеоинформация* – эту информацию можно обрабатывать при помощи кино- и видеотехники и хранить на киноплёнке и видеокассетах.

Все эти виды информации существовали еще до появления компьютера. Современный персональный компьютер позволил обрабатывать все эти виды информации и значительно облегчил их совместное использование.

Сведения – набор сигналов физических процессов, воспринимаемых субъектом через органы его чувств.

Данные – сведения, полученные путем измерения, наблюдения, логических или арифметических операций, представленные в форме, пригодной для хранения, передачи и обработки.

Информационные процессы. Понятие информации неразделимо с понятием информационных процессов. К **информационным процессам** относятся:

- передача информации;
- получение информации;
- хранение информации;
- обработка информации и ее представление для использования;
- использование информации.

Информационные технологии практически могут реализовываться как в неавтоматизированном (традиционном или, по-другому, «бумажном»), так и в автоматизированном виде (рис. 1.1).

Технические средства – это персональный компьютер, оргтехника, линии связи, оборудование сетей.

Программное обеспечение находится в прямой зависимости от технического и информационного обеспечения и реализует функции накопления, обработки, анализа, хранения, интерфейса с компьютером.

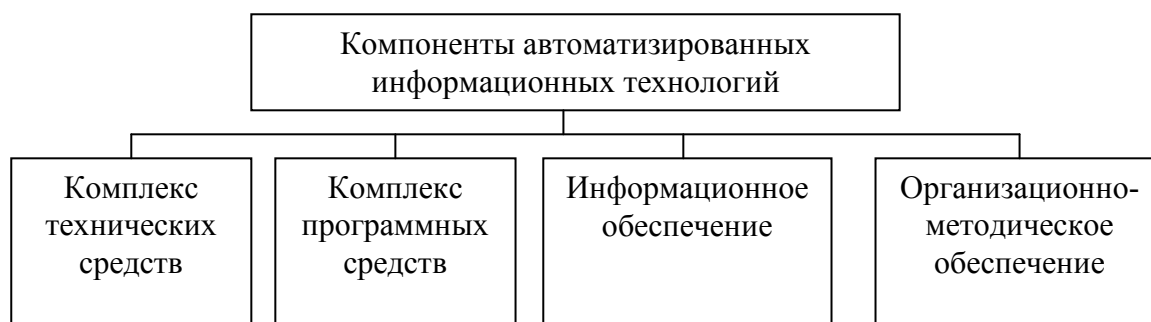


Рисунок 1.1 – Структура автоматизированных информационных технологий

Информационное обеспечение – совокупность данных, представленных в определенной форме для компьютерной обработки.

Организационно-методическое обеспечение представляет собой комплекс мероприятий, направленных на функционирование компьютера и программного обеспечения для получения искомого результата.

Существует и другой подход к рассмотрению структуры автоматизированной ИТ, согласно которому любая ИТ может быть разделена на три взаимозависимых и равнозначных компонента, составляющих ее ядро:

- аппаратное обеспечение (Hardware);
- программное обеспечение (Software);
- алгоритмическое (интеллектуальное) обеспечение (Brainware).

Кроме упомянутого выше ядра ИТ существует еще один очень важный компонент – сеть поддержки ИТ, инфраструктура (Infrastructure) – это необходимые физические, административные и организационные структуры, культурные схемы, стандарты и критерии и т.д.

Сторонники этого подхода считают, что на практике нужно рассматривать объединение или слияние информационных технологий в более крупные структуры – системы технологий (используется термин «информационные технологии и системы» – ИТ/С).

Однако простого наличия необходимых компонентов, составляющих ИТ (комплексов технических и программных средств, а также информационного и организационно-методического обеспечения), недостаточно для того, чтобы «оживить» технологию. Реализация ИТ возможна в определенной среде – информационной системе.

4 Свойства информационных технологий

Целесообразность – повышение эффективности производства на базе использования современных ЭВМ, распределенной переработки информации, распределенных баз данных, различных информационных вычислительных сетей (ИВС) путем обеспечения циркуляции и переработки информации.

Наличие компонентов (конкретное содержание процессов циркуляции и переработки информации) и *структуры* (внутренняя организация, представляющая собой взаимосвязи образующих ее компонентов,

объединенных в две большие группы: опорную технологию и базу знаний).
Взаимодействие с внешней средой, целостность, развитие во времени.

5 Классификация информационных технологий

Классификация информационных технологий зависит от критерия классификации. В качестве критерия может выступать показатель или совокупность признаков, влияющих на выбор той или иной информационной технологии.

Возможны различные схемы классификации информационных технологий. Каждая из них строится на определенных классификационных признаках.

Первый признак классификации – *отсутствие или наличие автоматизации*. В этом случае говорят о *традиционных и автоматизированных технологиях*.

Различают *обеспечивающие* и *функциональные* информационные технологии. *Обеспечивающие технологии* могут использоваться в качестве инструментария в различных предметных областях для решения различных задач. Они могут быть классифицированы по видам задач, которые решают. Примерами обеспечивающих технологий являются технологии обработки текстов, технологии систем управления базами данных. Обычно эти технологии могут выполняться на разных компьютерах и в разных программных средах. Основная задача – объединение этих технологий в единую информационную систему.

Функциональные технологии – совокупность обеспечивающих технологий для автоматизации некоторой задачи, функции.

Технология обработки информации на компьютере может заключаться в заранее определенной последовательности операций и не требовать вмешательства пользователя в процесс обработки. В данном случае диалог с пользователем отсутствует, и информация будет обрабатываться в *пакетном режиме обработки*.

Задачи, решаемые в пакетном режиме, характеризуются следующим:

- алгоритм решения задачи формализован, процесс ее решения не требует вмешательства человека;
- имеется большой объем входных и выходных данных, значительная часть которых хранится на магнитных носителях;
- расчет выполняется для большинства записей входных файлов;
- большое время решения задачи обусловлено большими объемами данных;
- задачи решаются с заданной периодичностью.

Если необходимо непосредственное взаимодействие пользователя с компьютером, при котором на каждое свое действие пользователь получает немедленные действия компьютера, используется *диалоговый режим обработки* информации. Диалоговый режим является развитием пакетного

режима обработки. Диалоговый режим предполагает отсутствие жестко закрепленной последовательности операций обработки данных, если она не обусловлена предметной технологией.

Следующий классификационный признак – *тип обрабатываемой информации*. Условная классификация современных компьютерных информационных технологий в зависимости от типа обрабатываемой информации приведена на рисунке 1.2.

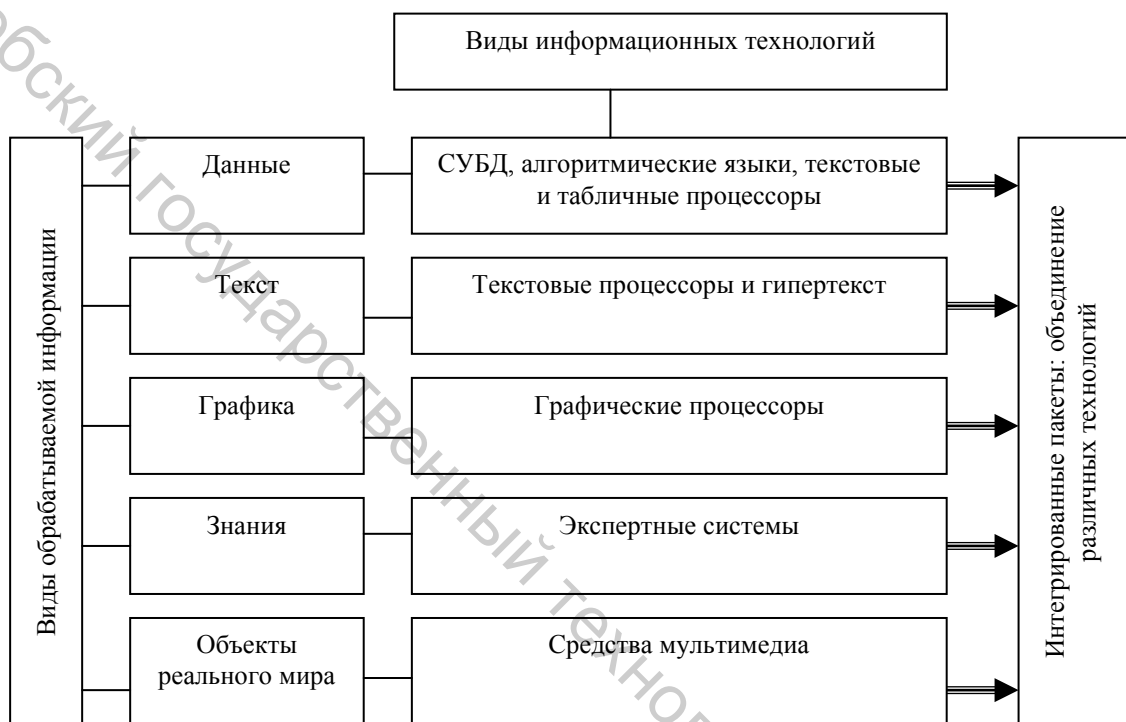


Рисунок 1.2 – Классификация информационных технологий по типу обрабатываемой информации

Классификация информационных технологий по другим классификационным признакам представлена в таблице 1.1.

Информационные технологии включают в себя *системы автоматизированного проектирования (САПР)*, где в качестве объекта может быть отдельная задача или элемент автоматизированной или компьютерной информационной системы.

Неотъемлемой частью информационной технологии является *электронная почта*, представляющая собой набор программ, позволяющий хранить и пересылать сообщения между пользователями.

Таблица 1.1 – Классификация информационных технологий

Классификационный признак	Вид информационной технологии
Степень автоматизации задач управления	– электронная обработка данных; – автоматизация функций управления; – электронный офис; – поддержка принятия решений
Тип пользовательского интерфейса	– командный – предполагает выдачу на экран приглашения для ввода команды; – WIMP-интерфейс – графический интерфейс пользователя; – SILK-интерфейс – поисковый интерфейс, в котором при воспроизведении речевой команды происходит переход от одних поисковых изображений к другим согласно семантическим связям
Способ построения сети ЭВМ	– локальные; – многоуровневые; – распределенные
Тип носителя	– бумажная (входные и выходные документы); – безбумажная (сетевая технология, современная оргтехника, электронные деньги, документы);
Степень типизации операций	– пооперационная – за каждой операцией закрепляется рабочее место с техническим средством; – попредметная – выполнение всех операций на одном рабочем месте, например, автоматизированное рабочее место (АРМ)
Вид информационного процесса	– информационная технология обработки данных (рис. 1.3); – информационная технология управления (рис. 1.4)

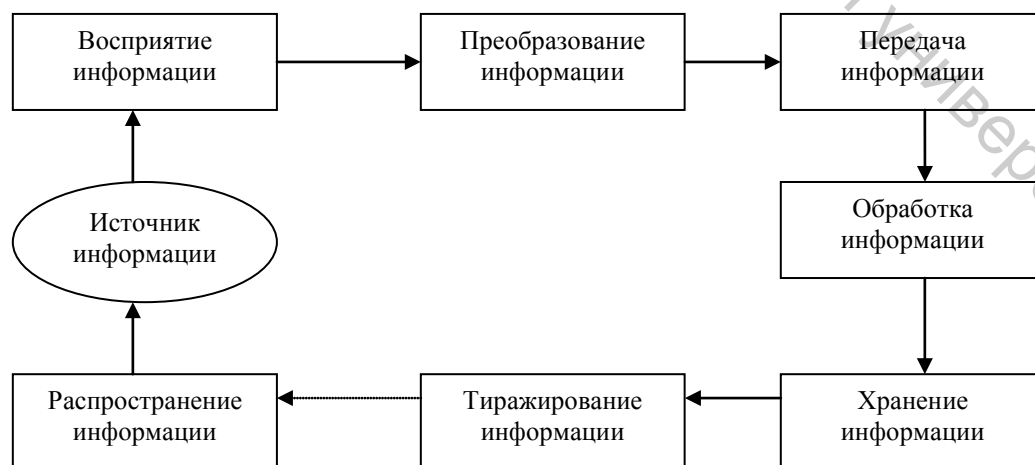


Рисунок 1.3 – Информационная технология обработки данных

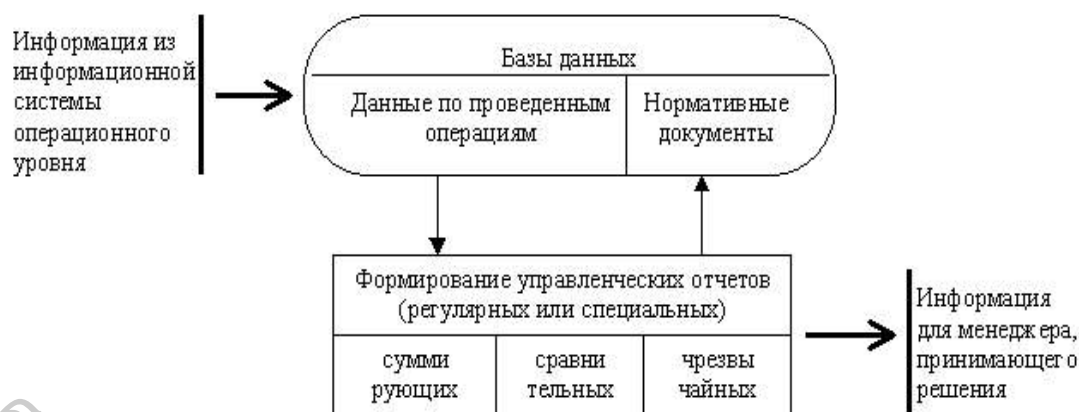


Рисунок 1.4 – Информационная технология управления

6 Обработка информации

Вся информация, поступающая в компьютер, кодируется или оцифровывается, т.е. всем характеристикам информации присваиваются числа. Таким образом, компьютер оперирует и обрабатывает не звук или видеоизображение, а ряд чисел. После обработки числа опять переводятся в звук или видеоизображение, и пользователь слышит музыку и видит мультфильм на экране компьютера. Чтобы технически упростить запись и обработку информации, используется двоичная система счисления. Если в десятичной системе счисления для записи всех чисел используется десять цифр 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, то в двоичной системе используется только две цифры 0 и 1, и все числа записываются при помощи комбинации нулей и единиц:

Десятичная система	Двоичная система
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
100	1100100
1000	1111101000

Какое-либо количество информации называется **объемом информации**.

Единица объема информации называется **бит**. В ячейке памяти компьютера объемом 1 бит может храниться 1 или 0.

8 бит составляют 1 **байт**.

У байта имеются кратные единицы:

Килобайт (Кбайт или Кб). $1 \text{ Кб} = 1024 \text{ байт}$.

Мегабайт (Мбайт или Мб). $1 \text{ Мб} = 1024 \text{ Кб}$.

Гигабайт (Гбайт или Гб). $1 \text{ Гб} = 1024 \text{ Мб}$.

Терабайт (Тбайт или Тб). $1 \text{ Тб} = 1024 \text{ Гб}$.

Для примера можно сказать, что если внести в компьютер текст одной машинописной страницы, то он будет иметь объем примерно 2500 байт.

7 Проблемы и перспективы развития информационных технологий

Для информационных технологий является вполне естественным то, что они устаревают и заменяются новыми.

На смену технологии пакетной обработки программ на большой ЭВМ в вычислительном центре пришла технология работы на персональном компьютере на рабочем месте пользователя.

- Телеграф передал все свои функции телефону.
- Телефон постепенно вытесняется службой экспресс-доставки.
- Телекс передал большинство своих функций факсу и электронной почте.

При внедрении новой информационной технологии в организации необходимо оценить риск отставания от конкурентов в результате ее неизбежного устаревания со временем, так как информационные продукты, как никакие другие виды материальных товаров, имеют чрезвычайно высокую скорость сменяемости новыми видами или версиями. Периоды сменяемости колеблются от нескольких месяцев до одного года.

Если в процессе внедрения новой информационной технологии этому фактору не уделять должного внимания, возможно, что к моменту завершения перевода фирмы на новую информационную технологию она уже устареет и придется принимать меры к ее модернизации. Такие неудачи с внедрением информационной технологии обычно связывают с несовершенством технических средств, тогда как основной причиной неудач является отсутствие или слабая проработанность методологии использования информационной технологии.

При внедрении информационной технологии в фирму необходимо выбрать одну из двух основных концепций, отражающих сложившиеся точки зрения на существующую структуру организации и роль в ней компьютерной обработки информации.

Первая концепция ориентируется на существующую структуру фирмы. Информационная технология приспособливается к организационной структуре, и происходит лишь модернизация методов работы. Коммуникации развиты

слабо, рационализируются только рабочие места. Происходит распределение функций между техническими работниками и специалистами. Степень риска от внедрения новой информационной технологии минимальна, так как затраты незначительны и организационная структура фирмы не меняется. Основным недостаток такой стратегии – необходимость непрерывных изменений формы представления информации, приспособленной к конкретным технологическим методам и техническим средствам. Любое оперативное решение «вязнет» на различных этапах информационной технологии. К достоинствам стратегии можно отнести минимальные степень риска и затраты.

Вторая концепция ориентируется на будущую структуру фирмы. Существующая структура будет модернизироваться, данная стратегия предполагает максимальное развитие коммуникаций и разработку новых организационных взаимосвязей. Продуктивность организационной структуры фирмы возрастает, так как рационально распределяются архивы данных, снижается объем циркулирующей по системным каналам информации и достигается сбалансированность между решаемыми задачами.

К основным ее недостаткам следует отнести:

- существенные затраты на первом этапе, связанном с разработкой общей концепции и обследованием всех подразделений фирмы;
- наличие психологической напряженности, вызванной предполагаемыми изменениями структуры фирмы и, как следствие, изменениями штатного расписания и должностных обязанностей.

Достоинствами данной стратегии являются:

- рационализация организационной структуры фирмы;
- максимальная занятость всех работников;
- высокий профессиональный уровень;
- интеграция профессиональных функций за счет использования компьютерных сетей.

Новая информационная технология в фирме должна быть такой, чтобы уровни информации и подсистемы, ее обрабатывающие, связывались между собой единым массивом информации. При этом предъявляются два требования. Во-первых, структура системы переработки информации должна соответствовать распределению полномочий в фирме. Во-вторых, информация внутри системы должна функционировать так, чтобы достаточно полно отражать уровни управления.

Лекция 2. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

План лекции

1. Классификация компьютеров по назначению и функциональным возможностям.
2. Классификация персональных компьютеров (ПК).
3. Структурная схема ПК.
4. Принцип работы ПК.

Компьютер – это электронное вычислительное устройство. Существует достаточно много систем классификации компьютеров. Рассмотрим некоторые из них, наиболее часто упоминающиеся в доступной технической литературе.

1 Классификация компьютеров по назначению

Это один из наиболее ранних методов классификации, который связан с тем, как компьютер применяется. По этому принципу различают:

- **Большие ЭВМ** – мэйнфреймы, на базе которых создают ИВЦ, включающие в себя несколько отделов или групп. Это большие компьютеры, с высоким быстродействием и большими вычислительными ресурсами, которые могут одновременно обрабатывать запросы нескольких тысяч пользователей.
- **Мини-ЭВМ** – для управления производственными процессами, автоматического проектирования, выполнения технологических и финансовых расчетов. Для организации работы мини-ЭВМ также требуется ВЦ.
- **Микро-ЭВМ** – как правило, для выполнения вспомогательных операций на ВЦ.
- **Персональные компьютеры (ПК)** – компьютеры, которые могут использоваться одним человеком автономно, независимо от других компьютеров.

2 Классификация ПК

ПК в свою очередь условно подразделяются на:

- **бытовые ПК;**
- **профессиональные ПК.**

Но начиная с 1999 г. в области ПК начал действовать международный сертификационный стандарт – сертификация PC99, в соответствии с которым

установлены следующие категории ПК:

- **Массовые ПК** (Consumer PC);
- **Деловые ПК** (Office PC) – минимизированы требования со средством воспроизведения графики;
- **Портативные ПК** (Mobile PC) – обязательно наличие средств для создания соединений удаленного доступа, то есть средств компьютерной связи;
- **Рабочие станции** (Workstations PC) – повышены требования к устройствам хранения данных;
- **Развлекательные ПК** (Entertainment PC) – к средствам воспроизведения графики и звука.

Кроме того, ПК классифицируют по типоразмерам:

- **настольные** (desktop) – используются в стационарных условиях комнаты или кабинета и располагаются на рабочем столе. Настольные компьютеры состоят из нескольких блоков – системного блока, монитора, клавиатуры и мышки, которые соединены между собой;
- **портативные** (notebook) – ноутбук имеет жидкокристаллический дисплей, клавиатуру, совмещенную с системным блоком, дисковод для 3,5" дискет и оптический дисковод (CD-ROM, CD-RW или комбинированный DVD+RW). Кроме того, обязательно имеется манипулятор для управления курсором. По размеру ноутбуки такие, что легко помещаются в портфель;
- **карманные** (palmtop), что означает «лежащий на ладони». В этих компьютерах программы занесены в микросхемы. Кроме палмтопов есть карманные компьютеры, которые называются PDA – personal digital assistant – личный цифровой ассистент. Эти компьютеры не имеют клавиатуры. Они оснащены сенсорным экраном, и информация вводится на экран при помощи специальной указки-стека. Общее название карманных компьютеров – handheld computers – компьютеры, которые держат в руках. Переносной и карманный компьютеры удобны для использования в поездках.

ПК классифицируют по совместимости:

- **аппаратная совместимость** (существуют две платформы) – IBM PC совместимые ПК и Apple Macintosh совместимые ПК;
- **совместимость на уровне ОС;**
- **программная совместимость;**
- **совместимость на уровне данных.**

3 Структурная схема ПК

Персональный компьютер включает в свой состав следующие основные устройства:

- системный блок, содержащий большинство электроники, ответственной за работоспособность ПК;
- монитор – устройство для отображения визуальной информации;
- клавиатура, позволяющая вводить в компьютер информацию и

управляющие команды;

- манипулятор «мышь».

К периферийным устройствам ПК относятся: принтер, джойстик, сканер, внешние оптические дисководы, модем, стример, плоттер, дигитайзер, цифровая видеокамера, световое перо, графический планшет и другие устройства, подключаемые к системному блоку через порты.

Системный блок (рис. 2.1) содержит следующие устройства:

- Блок питания – обеспечивает электроэнергией все электронные устройства ПК. Характеризуется максимальной отдаваемой мощностью.
- Системную (материнскую) плату.
- Комплекс адаптеров для подключения внешних устройств – обеспечивают возможность вывода информации из компьютера на внешнее устройство.
- Дисковод НЖМД – внешнее запоминающее устройство, предназначенное для долговременного хранения информации на компьютере.
- Дисковод CD-ROM – устройство, предназначенное для чтения компакт-дисков на компьютере.

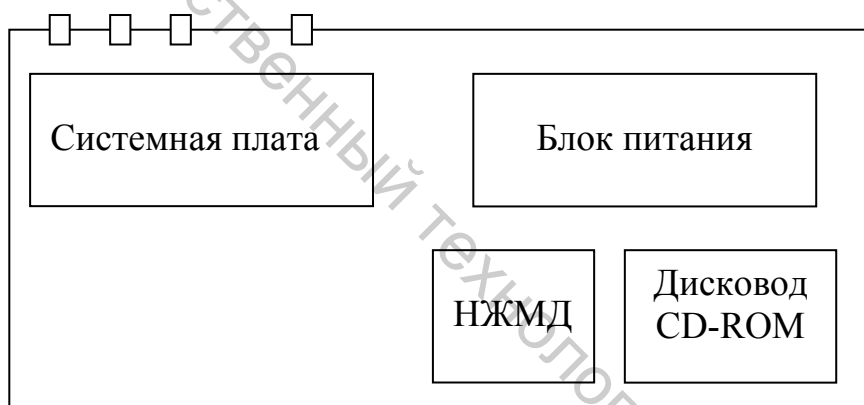


Рисунок 2.1 – Системный блок

Основным элементом системного блока является системная (материнская) плата.

На системной плате располагаются следующие элементы:

1. Микропроцессор – центральный блок ПК, предназначенный для управления работой всех остальных устройств компьютера и выполнения арифметических и логических операций над информацией. В состав процессора входят:

- **устройство управления** – отвечает за порядок выполнения команд, из которых состоит программа;
- **арифметико-логическое устройство (АЛУ)** в комплекте с математическим сопроцессором – предназначено для выполнения всех арифметических и логических операций над числовой и символьной информацией;

- **внутренняя микропроцессорная память** (регистры) – служит для кратковременного хранения, записи и выдачи информации;
- **блок декодировки** – преобразует данные в двоичную систему;
- **блок предварительной выборки** – получает команду от устройства (клавиатура и т.д.) и запрашивает инструкции в системной памяти;
- **кэш-память 1-го уровня** (или просто кэш) – хранит часто используемые инструкции и данные;
- **кэш-память 2-го уровня** – хранит часто используемые данные. Кэш 2-го уровня имеет больший объем, чем кэш 1-го уровня;
- **блок шины** – служит для ввода и вывода информации.

Основными характеристиками процессора являются:

- **тактовая частота** – это частота синхронизирующих импульсов синхронной электрической схемы, то есть количество синхронизирующих тактов, поступающих извне на вход системы за секунду. Тактовая частота процессора определяет количество элементарных операций (тактов), которое процессор может выполнить в единицу времени (секунду). То есть чем больше операций в секунду может выполнять процессор, тем быстрее он работает. Таким образом, в самом первом приближении тактовая частота характеризует производительность подсистемы;

- **внутренняя тактовая частота** – это тактовая частота, с которой происходит работа внутри процессора;

- **внешняя тактовая частота** или частота системной шины – это тактовая частота, с которой происходит обмен данными между процессором и оперативной памятью компьютера;

- **разрядность процессора** – количество информации, которое может быть обработано за 1 такт – определяется разрядностью его регистров. Если регистры могут хранить 32 единицы информации, то они 32-разрядные, и процессор 32-разрядный, если регистры 64-разрядные, то и процессор 64-разрядный и т.д. Чем большая разрядность процессора, тем большее количество информации он может обработать за один такт, а значит, тем быстрее работает процессор;

- **интерфейсная система микропроцессора**, реализующая сопряжение и связь с другими устройствами ПК.

2. Разъемы для плат оперативной памяти (ОЗУ) и постоянного запоминающего устройства (ПЗУ):

- **ПЗУ** служит для хранения программной и справочной информации, которая не изменяется в процессе работы ПК. Это тест-мониторные программы (они проверяют работоспособность компьютера в момент его включения), драйверы и др. ПЗУ – не энергозависимая память;

- **ОЗУ** используется для кратковременного хранения текущей информации и допускает изменение своего содержимого в ходе выполнения процессором вычислительных операций – программы пользователя, исходные,

промежуточные и конечные данные. ОЗУ – энергозависимая память.

3. Микросхемы быстрой памяти – внешняя кэш-память – служит для ускорения работы с оперативной памятью. В кэш-память записывается та часть информации оперативной памяти, которая изменяется в данный момент. Таким образом, уменьшается время поиска информации в памяти и увеличивается скорость выполнения программы.

4. Микросхема базовой системы ввода-вывода (BIOS). При включении питания BIOS тестирует состояние компьютера и его элементов и затем передает управление компьютером центральному процессору.

5. Разъемы для плат расширения (порты), которые служат для соединения с компьютером периферийных устройств: дисководов, монитора, мышки, клавиатуры, сканера, системы лазерного диска и т.д.

6. Шина расширения и адресная шина (на рисунке 2.1 не показаны) – обеспечивающие сопряжение и связь всех устройств компьютера между собой.

4 Принцип работы ПК

Большинство современных процессоров для ПК основаны на той или иной версии циклического процесса последовательной обработки информации, изобретенного Джоном фон Нейманом. Отличительной особенностью архитектуры фон Неймана является то, что инструкции и данные хранятся в одной и той же памяти. Упрощенно основные элементы ЭВМ (персонального компьютера) и связи между ними могут быть представлены следующей структурой (рис. 2.2).

Алгоритм работы ЭВМ (ПК) может быть сформулирован следующим образом.

1. Из процессора на шину адреса (находится внутри системной шины) выдается адрес очередной команды – число, хранящееся в регистре счетчика команд.

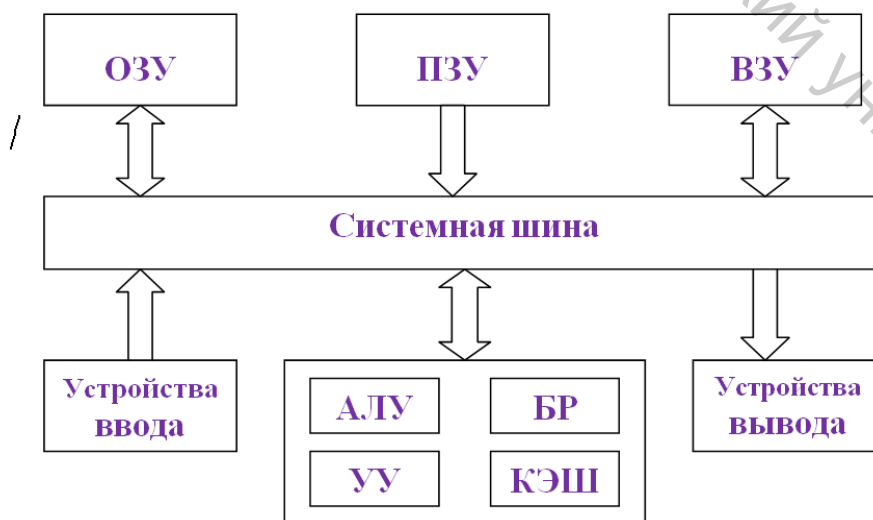


Рисунок 2.2 – Структурная схема ЭВМ

2. Считанная по этому адресу команда (например, из ПЗУ) поступает по шине данных (внутри системной шины) в процессор, где она выполняется с помощью АЛУ.

3. Если эта команда не является командой перехода, процессор увеличивает на единицу (в предположении, что длина каждой команды равна единице) число, хранящееся в счетчике команд, и в результате там образуется адрес следующей команды.

4. Устройство управления процессора определяет адрес следующей выполняемой команды (фактически номер очередной ячейки памяти, где хранится следующая команда и т.д.).

5. Снова выполняется действие 1.

Данный цикл выполняется неизменно, и именно он называется процессом (откуда и произошло название устройства – процессор).

Во время процесса процессор считывает последовательность команд, содержащихся в памяти, и исполняет их. Такая последовательность команд называется программой и представляет алгоритм работы процессора.

Сигналы, передаваемые по управляющей шине (*шине команд*), синхронизируют работу процессора, памяти, устройств ввода-вывода информации. Порядок выбора адресов из памяти (и очередности выполнения команд) определяет программа, которая может располагаться в ПЗУ, или загружаться в ОЗУ с внешнего запоминающего устройства, по вычислительной сети или вводиться с клавиатуры.

Лекция 3. КОМПЬЮТЕРНЫЕ СЕТИ

План лекции

1. Классификация компьютерных сетей.
2. Схемы компьютерных сетей.
3. Протоколы и адресация в ЛВС.
4. Глобальная сеть Интернет.
5. Стек протоколов TCP/IP.
6. Протоколы прикладного уровня сети Internet.
7. Доменные имена и IP-адреса.
8. Сервисы Internet. Виды сервисов в Internet, их назначение и особенности.
9. Виды информационных ресурсов и инструменты поиска в WWW.

1 Классификация компьютерных сетей

Компьютерной сетью (КС), или **сетью ЭВМ**, называется комплекс территориально рассредоточенных ЭВМ, связанных между собой каналами передачи данных и сетевым программным обеспечением для предоставления совместного доступа к общему ресурсу сети потенциальному пользователю сети и обмена информацией.

По территориальному признаку КС делятся на:

- **локальные (Local Area Network, LAN)** – сети, организованные в пределах существенно ограниченной территории (комната, этаж, здание, соседние здания). Размер локальной сети не превышает нескольких километров;

- **региональные (Metropolitan Area Network, MAN)** – сети, расположенные на обширном участке местности. Региональная сеть может соединять компьютеры внутри города, экономической зоны или отдельно взятой страны;

- **глобальные (Wide Area Network, WAN)** – сети, которые простираются на расстояния от десятков до десятков тысяч километров и могут объединять сотни локальных. Среди глобальных компьютерных сетей наиболее популярной является сеть Интернет.

Указанные выше сети различаются:

- **методами передачи данных.** В локальных сетях используются методы, не требующие предварительной установки соединения – данные просто передаются в канал связи без подтверждения готовности их принять.

Глобальные сети ориентированы на соединение, т.е. еще до начала передачи данных между компьютерами сети устанавливается соединение, которое подтверждается обменом компьютеров между собой специальными сигналами (кодами);

- **скоростью передачи данных.** Скорость передачи данных в локальных сетях – 10, 16 и 100 Мбит/с, в глобальных от 2,4 Кбит/с до 2 Мбит/с;

- **разнообразием услуг.** В локальных сетях существует широкий набор услуг, таких как файловые службы, услуги печати, услуги баз данных и т.д. Глобальные сети предоставляют в основном услуги, связанные с почтой и обменом файлами;

- **масштабируемостью** (возможностью расширения при сохранении качества). Локальные сети обладают плохой масштабируемостью. Глобальным сетям присуща хорошая масштабируемость, так как они изначально разрабатывались для сколь угодно большого числа пользователей.

В современном мире большую популярность приобрели *корпоративные компьютерные сети*, которые могут содержать различные сочетания всех выше перечисленных признаков и представляют собой сложный комплекс технических, системных и программных средств, функционирующих в рамках отдельных предприятий или корпораций. Территориальный признак в них не имеет никакого значения.

Для организации локальной КС необходимо наличие аппаратного и программного компонентов.

Аппаратный компонент КС составляют:

1. Компьютеры. Разделяются на компьютеры, предоставляющие ресурсы (серверы), и компьютеры, потребляющие ресурсы (рабочие станции – клиенты).

Сервер сети – это компьютер, подключенный к сети и предоставляющий пользователям сети набор некоторых услуг по использованию и распределению ресурсов сети, например, одновременный доступ пользователей к общим данным, печать заданий, прием и обработка запросов к базам данных и т.д. **Сервером** также называют программное обеспечение, хранящее соответствующую своему ресурсу информацию и отвечающее на запросы клиентского программного обеспечения.

Рабочая станция (клиент) – это подключенный к сети ПК, на котором пользователь непосредственно выполняет свою работу, использует свою операционную систему и с помощью которого имеет доступ к аппаратным, программным и информационным ресурсам сети. Так как рабочие станции в сети выступают клиентами, то *клиентом* называется и программа, устанавливаемая на компьютере пользователя для составления и отправки запросов соответствующему серверу, получения и отображения информации на компьютере пользователя. Серверы предоставляют материалы, а клиенты пользуются ими. В общем случае *клиентом* может быть и пользователь.

2. Линии связи или каналы передачи данных. При построении

локальных КС в качестве линий связи используются различные типы физических сред передачи данных:

– **проводные**, построенные на основе кабеля с электрическим сигналом передачи данных – коаксиального (жесткий медный проводник, отделенный от оплетки толстым слоем изоляционного материала), витой пары (скрученная пара проводников, заключенная в оболочку); с оптическим сигналом передачи данных – волоконно-оптического (состоит из тонкие волокон, по которым распространяются световые сигналы);

– **беспроводные** (с радиосигналом, микроволновым или инфракрасным сигналом передачи данных) – каналы наземной и спутниковой связи.

Выбор той или иной линии связи определяется требуемой скоростью передачи данных, а также расстояниями между отдельными узлами сети.

3. Коммутационное оборудование. Используется для связи сегментов сети: концентраторы, повторители, коммутаторы, маршрутизаторы, шлюзы. В настоящее время наблюдается тенденция к универсализации коммутационного оборудования при сохранении базовых названий.

Для передачи данных между двумя компьютерами необходимо организовать физическую связь. В локальных сетях традиционно используется некоммутируемая линия связи (линия связи соединяет два компьютера постоянно) и для физического подключения компьютера в сеть применяется специальное устройство – сетевой адаптер (сетевая карта), устанавливаемый обычно в компьютере и соединяющий его с линией связи. Для функционирования сетевых адаптеров необходимы специальные программы – драйверы.

Для соединения по коммутируемой линии (связь между компьютерами устанавливается только на время сеанса обмена данным между ними) используется модем. Модем (модулятор/демодулятор) – устройство, которое позволяет передавать (принимать) информацию по телефонной линии между удаленными компьютерами. Передающий модем преобразует цифровые данные в аналоговые сигналы, которые могут передаваться по телефонной линии. Принимающий модем переводит аналоговые сигналы в обратно цифровую форму.

4. Соединительное оборудование. К соединительному оборудованию относятся: коннекторы – разъемные и/или неразъемные соединители, прикрепляемые к кабелям, различные кабельные адаптеры и разветвители – для стыковки разных типов кабелей, усилители и другие устройства. Соединительное оборудование обеспечивает возможность подключения различных сетевых устройств к линиям связи.

Программный компонент КС определяют:

1. Сетевая операционная система ОС – связывает все компьютеры и периферийные устройства в сети, координирует функции всех компьютеров и периферийных устройств в сети, обеспечивает защищенный доступ к данным и периферийным устройствам в сети, предоставляет собственные ресурсы и

определенные услуги в общее пользование. В зависимости от распределения функций (клиент/сервер) между компьютерами локальные сети делят на одноранговые и сети с выделенным сервером.

В одноранговых локальных сетях каждый компьютер может выполнять как функции клиента, так и функции сервера. Одноранговые сети, как правило, строятся для небольшого количества компьютеров, иначе возникают проблемы по управлению работой сети и обеспечению защиты информации. Пользователю такой сети при соответствующих настройках могут быть доступны все ресурсы других компьютеров: жесткий диск, принтер, сканер и т.д. В одноранговых сетях на каждом компьютере устанавливается ОС, поддерживающая работу сети. В качестве ОС на рабочих станциях используют преимущественно Windows 9x, Windows ME, Windows 2000, Windows XP и Windows NT Workstation. Однако возможности такого сервера сильно ограничены.

При построении сложных сетей, как правило, один или два компьютера выделяют для выполнения сетевых функций (*сети с выделенным сервером*). К таким серверам предъявляются более высокие требования по их техническому обеспечению, производительности, объему памяти, надежности защиты информации. В сетях с выделенным сервером на сервере устанавливают серверную ОС, обладающую значительными сетевыми возможностями. К преимуществам серверных ОС можно отнести обеспечение высокой производительности сети, наличие развитых средств управления и администрирования в сети. На серверах ведутся учетные записи пользователей сети, которые содержат информацию обо всех пользователях локальной сети, подключенных к данному серверу и их правах доступа к ресурсам сети, имена компьютеров, рабочих групп и т.д. Примером сетевой ОС, устанавливаемой на выделенном сервере является **Windows 2000 Server**. При этом на рабочие станции устанавливают те же ОС, что и в одноранговых сетях.

2. Сетевые программные приложения – это прикладные программы, которые расширяют возможности сетевых ОС. Среди них можно выделить сервер баз данных, сервер Интернета, сервер информационного обмена и т.д.

2 Схемы компьютерных сетей

Физически компьютеры могут объединяться в сеть разными способами. Структура физического соединения компьютеров в сеть называется топологией.

Топология – это физически реализованная основа сети, ее непросто изменить после того, как сеть уже создана.

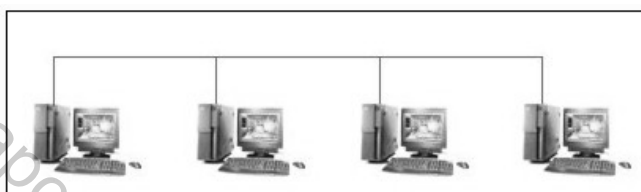
Каждая КС обладает своей уникальной топологией, однако в ней всегда можно выделить базовые:

- **простое** (последовательное) соединение. Информация передается последовательно с одного компьютера на другой и обратно. Схема работает быстро, но при разрыве одного из соединений или при неисправности одного компьютера вся сеть выходит из строя.

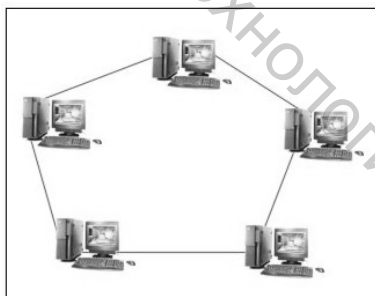
Практически эта схема почти не используется.



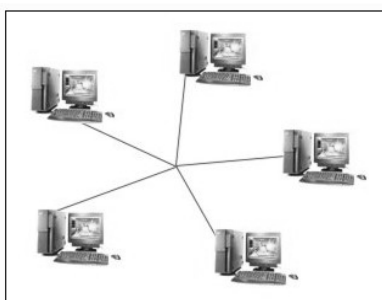
- **общая шина** (компьютеры подключаются к одному общему кабелю (шине), по которому происходит обмен информацией) При таком соединении обмен может производиться между любыми компьютерами сети, независимо от остальных. При повреждении связи одного компьютера с общей шиной, этот компьютер отключается от сети, но вся сеть работает. В этом смысле сеть достаточно устойчива, но если повреждается шина, то вся сеть выходит из строя.



- **кольцо** (сеть не имеет конечного пункта соединения, данные передаются от компьютера к компьютеру по кольцу в одном направлении). Один разрыв не выводит сеть из строя, но два разрыва делают сеть нерабочей. Кольцевая сеть достаточно широко применяется. В основном, из-за высокой скорости передачи данных. Кольцевые сети самые скоростные.



- **звезда** (каждый компьютер отдельным кабелем подключается к концентратору, находящемуся в центре сети). Сеть устойчива к повреждениям. При повреждении одного из соединений от сети отключается только один компьютер. Кроме того, эта схема соединения позволяет создавать сложные разветвленные сети.



Как правило, локальные сети ориентированы на передачу алфавитно-цифровой, текстовой, графической, мультимедийной информации. Совмещение в одной сети потоков разных видов информации является одним из основных требований, предъявляемых к локальным КС.

3 Протоколы и адресация в ЛВС

Протокол – это набор правил, в соответствии с которым компьютеры обмениваются информацией. Эти правила включают формат, время и последовательность передачи данных, способы контроля и коррекции ошибок. В соответствии с моделью OSI (Open System Interconnection) существует семь уровней протоколов:

1. Физический уровень. Побитовая передача сигналов в кабелях: типы кодирования и физические характеристики сигналов, скорость передачи сигналов и т.д.

2. Канальный уровень. Передача кадров данных между сетевыми картами компьютеров. В самом общем виде кадр данных – это группа битов, состоящая из заголовка кадра и поля данных. Канальный уровень – это аппаратное взаимодействие сетевая карта – сетевая карта.

3. Сетевой уровень. Сетевая логическая адресация сетевая карта – сетевая карта. Кроме того, сетевой уровень отвечает за маршрутизацию (доставку) пакетов данных вне зависимости от сложности топологии сети.

4. Транспортный уровень. Обеспечивает надежность доставки пакетов данных: установка виртуального канала передачи данных между сетевыми картами, контроль искажения или утери пакетов данных, повторная передача пакетов данных при необходимости.

5. Сеансовый уровень. Сеансовый уровень управляет диалогом сетевая карта – сетевая карта: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации.

6. Представительский уровень. Позволяет менять форму представления информации, не меняя ее содержания. Например, преобразования кодировки ASCII в кодировку EBCDIC, или шифрование передаваемых по сети данных при помощи протокола SSL (Secure Socket Layer).

7. Прикладной уровень. Набор разнообразных протоколов, при помощи которых взаимодействуют между собой прикладные программы. Каждая программа по желанию программиста может иметь свой собственный протокол или использовать один из широко известных прикладных протоколов.

Модель OSI является международным стандартом, однако для практических целей, чаще всего пользуются упрощенной моделью, в которой физический уровень подразумевается, но не рассматривается, а сеансовый и представительский уровни объединены с прикладным. Таким образом, упрощенная модель включает в себя четыре уровня протоколов:

1) канальный уровень;

2) сетевой уровень: протоколы IP и ICMP стека протоколов TCP/IP;

3) транспортный уровень: протоколы TCP и UDP;

4) прикладной уровень: протоколы HTTP, FTP, SMTP, IMAP, POP3, TELNET.

4 Глобальная сеть Интернет

Интернет (Internet, от лат. Inter – между и net – сеть, паутина) – глобальная компьютерная сеть, то есть совокупность соединенных между собой компьютеров и множества сетей.

Идея создания сети Интернет зародилась в военном ведомстве Соединенных Штатов Америки. Цель проекта состояла в разработке такой сети, которая сможет обеспечить бесперебойную передачу данных между децентрализованными компьютерами военного назначения.

В сентябре 1969 г. группа студентов из Калифорнийского университета в Лос-Анджелесе под руководством профессора Леонарда Клейнрока создала первую в мире компьютерную сеть, соединив два вычислительных комплекса кабелем длиной 4,5 метра.

К концу 1969 г. эти четыре узла были объединены в первую конфигурацию сети ARPAnet, ставшей прототипом сети Интернет. Название сети ARPAnet происходит от названия ее учредителя Advanced Research Project Agency (ARPA – Управление перспективных исследовательских программ США).

В 1984 г. из сети ARPAnet была выделена сеть военного назначения (MILnet), а сеть ARPAnet, в дальнейшем переименованная в Интернет, стала сетью образовательного назначения.

В середине 1980-х годов контроль за доступом к сети Интернет осуществлял Национальный научный фонд (National Science Foundation – NSF) и другие правительственные ведомства США. Сетевой трафик принадлежал правительству США и им же финансировался (трафик – информация, передаваемая по сети Интернет). В апреле 1995 г. правительство США отказалось от контроля над Интернетом в пользу независимых административных советов.

Сегодня функционирование основных коммуникаций финансируется из различных средств, а вопросы, связанные с технической поддержкой, решаются рядом общественных комитетов и советов. Одним из наиболее значимых является Инженерный комитет Интернета (Internet Engineering Task Force – IETF). IETF – общественный комитет, в котором ученые и эксперты решают технические проблемы и вопросы дальнейшего развития сети. Еще одной организацией, оказывающей значительное влияние на дальнейшее развитие сети, является Совет по архитектуре Интернета (Internet Architecture Board – IAB). IAB – добровольная организация, состоящая из группы приглашенных экспертов.

Существуют национальные и международные сегменты Интернета, которые финансируются из различных источников и управление которыми

осуществляется собственной администрацией. Решения IAB и IETF публикуются в сети Интернет в виде документов «Запросы на комментарии и предложения» (Request for Comment – RFC). Такие RFC-документы хранятся на многих узловых компьютерах Интернета.

5 Стек протоколов TCP/IP

В основу сети ARPAnet был положен способ группирования информации в пакеты. Теория пакетной технологии передачи данных была разработана Л. Клейнроком в середине 1960-х годов. Параллельно с ним разработкой пакетной коммутации данных занималась группа сотрудников компании RAND (название этой компании происходит от аббревиатуры R&D, что означает «исследования и разработки»).

Суть разработанного способа состоит в передаче пакета данных в так называемое скопление большого числа компьютеров, подключенных к сети. Помимо передаваемой информации к пакету присоединяется заголовок, содержащий адрес получателя. Получивший такой пакет компьютер проверяет принадлежность передаваемой информации любому из его клиентов и пересылает невостребованные пакеты следующему компьютеру, которому они могут принадлежать. После затребования соответствующим компьютером пакет раскрывается и из него извлекается сообщение.

Управление доставкой пакетов осуществляет протокол – набор правил, определяющих способ передачи и форматирования данных, передаваемых по компьютерной сети.

Одним из главных итогов развития ARPAnet, перешедших в Интернет, стало создание сетевых протоколов TCP/IP.

TCP/IP (Transmission Control Protocol/Internet Protocol) – семейство протоколов, определяющих, как данные разбиваются на пакеты для передачи по сети и как приложения могут пересылать пакеты. **TCP** (Transmission Control Protocol) – транспортный протокол, определяющий размер пакета передаваемых данных и проводящий тонкую настройку параметров передачи. **IP** (Internet Protocol) – основной сетевой протокол, реализующий межсетевое соединение.

Протокол TCP/IP (Transmission Control Protocol/Internet Protocol – протокол контроля передачи данных/протокол передачи данных между сетями, Internet). Является основным протоколом, применяющимся в Internet. В состав стека протоколов TCP/IP входят протоколы: IP и ICMP – сетевой уровень, TCP и UDP – транспортный уровень.

Протокол IP отвечает за адресацию в сети и доставку пакетов между компьютерами сети, без установления соединения и гарантий доставки пакета. При использовании протокола IP, каждый компьютер в рамках сети должен иметь уникальный IP-адрес, представляющий собой 32-битное число. Для удобства чтения, IP-адрес разбивают на четыре 8 битовых числа, называемых октетами, например 149.76.12.4. В локальной сети, которая не подключена к

Internet или другим сетям, можно назначать IP-адреса произвольно. Однако в Internet IP-адреса выделяются централизованно организацией InterNIC. InterNIC выдает адреса не на каждый отдельный компьютер, а в целом на локальную сеть.

В IP-адресе выделяют две части: сетевую часть (адрес локальной сети) и адрес компьютера в сети. Сетевая часть адреса может иметь переменную длину, которая зависит от класса IP-адреса и маски подсети. Выделяют следующие классы IP-адресов:

Класс А включает сети с адресами от 1.0.0.0 до 127.0.0.0. Сетевой номер содержится в первом октете (1-127), что предусматривает 126 сетей по 1.6 миллионов компьютеров в каждой.

Класс В включает сети с адресами от 128.0.0.0 до 191.255.0.0. Сетевой номер находится в первых двух октетах (128.0 – 191.255), что предусматривает 16320 сетей с 65024 компьютерами в каждой.

Класс С включает сети с адресами от 192.0.0.0 до 223.255.255.0. Сетевой номер содержится в первых трех октетах (192.0.0 – 223.255.255). Это предполагает почти 2 миллиона сетей по 254 компьютеров в каждой.

Классы D включает адреса от 224.0.0.0 до 239.255.255.0. Эти адреса являются групповыми (multicast). Если нескольким компьютерам в сети назначен один и тот же групповой адрес, то пакет, адресованный на этот адрес, получают все компьютеры. Такие адреса в локальных сетях используются редко и зарезервированы для того времени, когда технические возможности сети Internet позволят организовывать теле- и радиовещание на группы компьютеров.

Классы E и F включает адреса, попадающие в диапазон от 240.0.0.0 до 254.0.0.0, являются или экспериментальным, или сохранены для будущего использования и не определяют какую-либо сеть.

Помимо адресов из классов A, B, C, D, E, F, существует также несколько зарезервированных адресов. IP-адрес, в котором все биты октеты адреса компьютера равны 0, относится ко всей сети, а где все биты октеты адреса компьютера равны 1, назван широковещательным (broadcast) адресом. Он относится к каждому компьютеру сети. Таким образом, 149.76.255.255 – не существующий адрес компьютера, который относится ко всем компьютерам из сети 149.76.0.0.

Имеются еще два зарезервированных IP-адреса, 0.0.0.0 и 127.0.0.0. Первый назван: путь пакетов по умолчанию (default route), второй – кольцевым (loopback) адресом или ссылкой на самого себя. В несуществующей сети 127.0.0.0 адрес 127.0.0.1 будет назначен специальному интерфейсу, который действует подобно закрытому кругуобороту. Любой IP-пакет, переданный на этот адрес, будет возвращен на этот же компьютер так, как если бы пакет пришел откуда-то из сети. Это позволяет тестировать сетевое программное обеспечение без использования «реальной» сети.

Также имеется ряд «серых» IP-адресов, которые зарезервированы для использования только в локальных сетях. Пакеты с «серыми» адресами не

передаются маршрутизаторами Internet. К таким адресам относятся:

Сеть класса А 10.0.0.0

Сеть класса В от 172.16.0.0 до 172.31.0.0

Сеть класса С от 192.168.0.0 до 192.168.255.0

По соображениям безопасности рекомендуется использовать в локальных сетях только «серые» адреса. В таком случае прямой доступ из Internet к компьютерам ЛВС, в обход прокси-сервера организации, будет невозможен.

Кроме адресации компьютеров в сети, протокол IP также отвечает за маршрутизацию (выбор маршрута доставки) пакетов данных в сетях с произвольной топологией.

Протоколы транспортного уровня в стеке TCP/IP представлены двумя протоколами: **TCP** и **UDP**.

Протокол TCP позволяет устанавливать виртуальный канал передачи данных между компьютерами. После установления канала программа может направлять в него данные непрерывным потоком как на стандартное устройство ввода-вывода. Протокол TCP сам разобьет данные на пакеты, при помощи алгоритма «скользящего окна» обеспечит подтверждение факта получения пакетов принимающей стороной и повторную передачу пакетов, если в этом будет необходимость. Кроме того, в протоколе TCP реализованы достаточно сложные механизмы регулирования загрузки сети и устранения заторов в сети.

Протокол UDP более быстр, чем протокол TCP, однако менее надежен. Данные передаются без установления виртуального канала, в предположении, что принимающая сторона ждет данные. Программа должна сама позаботиться о разбиении передаваемых данных на пакеты, протокол не содержит средств подтверждения факта доставки сообщения и средств коррекции ошибок – все эти задачи должна решать программа.

6 Протоколы прикладного уровня сети Internet

В соответствии с архитектурой клиент-сервер, программа делится на две части (одна работает на сервере, вторая – на компьютере пользователя), функционирующие как единое целое. Протоколы прикладного уровня описывают взаимодействие клиентской и серверной частью программы. Выделяют следующие наиболее известные прикладные протоколы:

HTTP (Hyper Text Transfer Protocol) – протокол передачи гипертекста. Используется в WWW для передачи гипертекстовых HTML-страниц. При работе по этому протоколу, каждый элемент HTML-страницы загружается отдельно, причем соединение между загрузками прерывается и никакой информации о соединении не сохраняется. Это сделано для того, чтобы при использовании Web-страниц каждый получал «по чуть-чуть, в порядке общей очереди». В противном случае могла бы создаться ситуация, когда один человек качает страницу с большим количеством рисунков высокого разрешения, а все остальные ждут, пока он это закончит.

FTP (File Transfer Protocol) – протокол передачи файлов. Предназначен для копирования файлов между компьютерами. Полностью занимает канал, пока не будет получен файл, сохраняет информацию о соединении. При сбое возможна докачка с того места, где произошел сбой.

SMTP, IMAP-4, POP3 – почтовые протоколы (электронная почта). Отличие: SMTP – протокол, рассчитанный на доставку почты до конкретного получателя, POP3 и IMAP-4 – протоколы взаимодействия пользователя со своим почтовым ящиком на сервере.

TELNET – используется для подключения и управления удаленным компьютером. После подключения каждый символ, введенный на локальной машине, обрабатывается так, как если бы он был введен на удаленной машине. Либо может использоваться командный режим – управление удаленной машиной при помощи специальных команд. Фактически TELNET – это протокол эмуляции терминала: при помощи TELNET можно подключиться и вручную набрать все необходимые поля заголовка письма, изменив адрес отправителя (обычно эти поля заполняются автоматически специальными почтовыми программами) и отправить само письмо.

7 Доменные имена и IP-адреса

Для получения информации в сети Интернет необходимо знать адреса ресурсов Интернета: электронной почты, хост-компьютеров и т.д.

Хост-компьютеры (узловые компьютеры) – компьютеры, включенные в сеть и предоставляющие различные виды сетевого сервиса. Все хост-компьютеры Интернета идентифицируются в соответствии с уникальным доменным именем и числовым адресом, называемым IP-адресом. Адрес веб-сайта, как правило, содержит имя хост-компьютера, на котором находится веб-сайт.

Понятие «**домен**» используется для обозначения группы компьютеров, имеющих общую часть в адресах. **Доменное имя** – это адрес, состоящий из двух или более групп слов, разделенных точками, определяющих конкретный хост-компьютер, владеющую им организацию и иерархию доменов (объединение компьютеров и сетей), к которым она принадлежит.

Таким образом, в полное имя хост-компьютера в дополнение к имени конкретного компьютера последовательно включаются имена всех вышестоящих доменов.

Примеры доменных имен: www.company.com; www.pfu.edu.ru; www.imeb.ru.

По доменному имени можно получить некоторую информацию о принадлежности данного компьютера к определенной организации или сети. Это имя декодируется справа налево. Крайняя справа часть доменного имени является именем зоны (вместо имени зоны часто используют понятие домена верхнего, или первого, уровня). Далее идет доменное имя, причем наиболее смысловой является крайняя левая его часть. Выбор названий, расположенных

в доменном имени слева, является прерогативой организации, владеющей сервером. Многие организации ограничиваются единственным словом, определяющим имя конкретного компьютера в их сети. В больших организациях могут быть добавлены дополнительные имена, определяющие название подразделения. Например: `www.virginia.edu` (имя зоны – `edu`); `econ.rudn.rssi.ru` (имя подразделения – `econ`, имя зоны – `ru`).

До 2001 г. имена зон подразделялись только на двух- и трехбуквенные (табл. 3.1).

Таблица 3.1 – Выборочный список трехбуквенных имен зон

Имена зон	Расшифровка
com	Коммерческие организации
edu	Образовательные учреждения
gov	Правительственные учреждения и ведомства США
int	Международные организации
mil	Военные ведомства и учреждения
net	Поставщики сетевых услуг
org	Любые некоммерческие организации, главным образом, профессиональные общества

С 2001 г. были разрешены имена других типов, например `.info`, `.museum`, `.name` и др. Обычно такие имена зон присваиваются в зависимости от тематики веб-сайта. Но до сих пор большого распространения они не получили и по-прежнему широко распространены двухбуквенные имена зон, обозначающие коды стран, и трехбуквенные – обозначающие типы организаций.

Каждому доменному имени соответствует числовой IP-адрес, который необходим для того, чтобы программное обеспечение, поддерживающее работу сети Интернет, могло передать или принять информацию с хост-компьютера, т.е. IP-адрес – числовой аналог доменного имени.

IP-адрес – уникальный номер компьютера, состоящий из четырех чисел, разделенных точками, каждое из которых лежит в пределах от 0 до 255. Например: `192.58.107.230`.

Обычно программное обеспечение автоматически переводит доменное имя хост-компьютера в IP-адрес через специальную службу DNS. DNS (Domain Name Service — служба доменных имен) – это часть программного обеспечения, предназначенная для преобразования IP-адресов в простые для чтения и запоминания пользователем доменные имена.

8 Сервисы Internet

World Wide Web. Протокол HTTP. URL-адресация web-ресурсов

Доступ к ресурсам сети Интернет и их использование могут осуществляться различными способами с помощью специальных служб сети Интернет.

Первые приложения для Интернета, такие, как FTP или электронная почта, были разработаны исключительно для обмена данными между компьютерами. Другие приложения, такие как Telnet, были разработаны для того, чтобы пользователь получил возможность доступа не только к информации, но и к ресурсам конкретного удаленного компьютера.

По мере развития Интернета возникла необходимость разработки других способов поиска информации и доступа к ней. Встала задача создания приложений, позволяющих использовать информационные ресурсы, не обращая внимания на формат данных и место их хранения в сети.

Для решения этих задач сначала были созданы поисковая система Archie и система Gopher, а затем были созданы системы WAIS и WWW, предлагающие новые методы получения и отображения информации.

Служба FTP. На начальной стадии развития Интернета основным методом доступа к информационным ресурсам сети было использование сервиса FTP. FTP (File Transfer Protocol) – протокол передачи данных, регламентирующий процедуру передачи файлов между компьютерами. FTP позволяет установить соединение с удаленным компьютером в сети Интернет, просмотреть файлы, доступные на нем, и при необходимости скопировать их. Но надо отметить, что эта служба не предназначена для обработки файлов, например, для просмотра их содержимого.

FTP – пример системы с архитектурой «клиент – сервер». Программное обеспечение делится на сервер-программу, запущенную на удаленном компьютере, и программу-клиент, запущенную на компьютере пользователя. Программой-клиентом может быть обычный веб-браузер. Практически все современные операционные системы включают программу ftp-клиент для работы в командной строке, которая обычно так и называется – ftp.

Основным ограничением службы FTP является то, что запись и удаление файлов на FTP-серверах обычно возможны только с определенным именем пользователя. Однако распространены анонимные FTP-серверы, для работы с которыми не нужно иметь специального имени пользователя и пароля. На таких серверах в качестве имени пользователя используется *anonim*, а в качестве пароля – чаще всего адрес электронной почты.

Адреса FTP-серверов формируются по тем же правилам, что и адреса веб-страниц. Существует лишь одно отличие – вместо префикса *http://* используется префикс *ftp://*, указывающий, что работа с данным сервером будет проходить по протоколу FTP. Пример адреса: <ftp://ftp.microsoft.com>.

FTP-сервер до сих пор является одним из главных способов распространения программного обеспечения в Интернете. Для поиска файлов,

размещенных на FTP-серверах, существуют специальные поисковые системы. Одна из таких систем – FileSearch. Адрес этой системы в сети Интернет: www.filesearch.ru.

Служба Archie. Одна из первых поисковых систем *Archie* относится к поисковым системам, предназначенным для решения вопросов локализации информационных ресурсов Интернета. Принцип ее работы заключается в создании индексируемой базы данных о файлах, доступных на анонимных FTP-серверах, и в поиске в этой базе данных нужного файла. Для поиска файла с помощью службы Archie необходимо знать хотя бы часть его имени.

Основным недостатком этой системы является то, что не все хост-компьютеры, предоставляющие анонимный доступ, находят отражение в базе данных Archie.

Служба Gopher. Термин «Gopher» относится к сетевому протоколу, к типу серверов Интернета и программному обеспечению для доступа в Интернет. Gopher предоставляет возможность работать с информационными ресурсами, не зная заранее их местонахождения. Хотя для того, чтобы сделать первый шаг, необходимо знать название хотя бы одного Gopher-сервера. Далее вся работа строится через систему меню, похожую на оглавление книги. Gopher позволяет производить поиск по ключевым словам и фразам в названиях файлов и каталогов.

Совокупность информационных ресурсов, имеющихся на всех *Gopher-серверах*, принято называть *Gopher-пространством*.

Служба WAIS. WAIS (Wide Area Information Servers – распределенная система информационных серверов) – служба, предназначенная для поиска документов по заданным ключевым словам. Ключевые слова в виде запроса отправляются на один из WAIS-серверов. **WAIS-серверы** – это серверы Интернета, которые осуществляют поиск документов в специальных индексируемых базах данных и возвращают списки документов, удовлетворяющих критерию поиска.

Служба Telnet. Telnet – это название протокола, используемого для терминального доступа к удаленному компьютеру. Служба Telnet позволяет связаться с конкретным удаленным компьютером и обменяться с ним информацией. Для работы с Telnet достаточно иметь установленную ОС Windows на компьютере. Эта служба вызывается из стандартных программ главного меню Windows.

Многие крупные организации используют эту службу. Для работы с удаленной системой надо точно знать ее адрес. Например, адрес Библиотеки конгресса США выглядит так: *locis.loc.gov*.

Служба Whois. Служба Whois – база данных, содержащая информацию о пользователях сети. Недостаток этой системы состоит в том, что пользователи, не желающие предоставлять информацию о себе, в базе данных не фигурируют.

Служба Finger служит для определения, работает ли в данный момент пользователь на компьютере.

Служба WWW. World Wide Web (WWW или 3W – (Всемирная паутина) – служба Интернета, предоставляющая возможность работы с гипертекстовыми документами, размещенными на серверах в сети Интернет. Гипертекстовые документы – это документы, содержащие ссылки на другие, связанные по смыслу документы, которые могут находиться и на удаленных компьютерах. В таких документах могут быть объединены текст, графические иллюстрации, звук и т.д.

Проект WWW был начат в марте 1989 г. программистом Тимом Бернерсом-Ли, работавшим в Европейской лаборатории физики элементарных частиц. В 1990 г. Т. Бернерс-Ли написал программу под названием «редактор гипертекста», которая позволяла выделять одним щелчком мыши информацию в тексте документа для ссылки на другие документы, находившиеся в компьютерной сети лаборатории. Вскоре сотрудники лаборатории начали пользоваться этим редактором для пересылки друг другу научных статей. Впоследствии документы начали передаваться по всей сети Интернет. Такое виртуальное пространство получило название WWW (Всемирная паутина), а компьютеры, предоставлявшие гипертекстовые документы, – веб-серверами или WWW-серверами.

Программы, предназначенные для просмотра гипертекстовых документов, называются программами для просмотра WWW или веб-браузерами (WWW-браузерами). Веб-браузер получает затребованные документы, интерпретирует данные и после этого отображает информацию на экране.

Веб-серверы и веб-браузеры соединяются между собой в основном с помощью протокола HTTP. HTTP (HyperText Transfer Protocol) – протокол передачи гипертекстовых документов, обеспечивающий прием и передачу веб-страниц.

Веб-страница (Web-page) – составная часть веб-сайта, которая представляет собой файл, содержащий гипертекстовый документ. Веб-сайт (Web-site) – совокупность веб-страниц, объединенных по смыслу и размещенных на одном сервере.

В отличие от обычных документов веб-страницы содержат команды, задающие структуру документа (заголовки разного уровня, абзацы основного текста и т.д.), что дает возможность веб-браузеру отформатировать документ для его отображения на экране в соответствии с возможностями конкретного компьютера.

Для того чтобы такое форматирование стало возможным, необходимо было разработать универсальный формат для веб-документов. Решением этой задачи стал стандарт HTML.

HTML (HyperText Markup Language) – язык разметки гипертекста, использующийся при подготовке веб-документов. Основной отличительной особенностью гипертекстовых документов является наличие ссылок, называемых гиперссылками. Гиперссылка отличается от обычной текстовой ссылки тем, что при щелчке кнопкой мыши по ней указанный в тексте

документ отображается на экране.

Одной из целей проекта WWW была разработка стандартного способа указания на ресурсы в Интернете. Для решения этой задачи было введено понятие URL (Uniform Resource Locator) – универсальный адрес ресурсов. Это имя, обозначающее адрес конкретного веб-сайта: название файла и каталога, адрес компьютера и метод доступа к файлу. Таким образом, URL описывает местонахождение информационного ресурса и его содержание.

В языке HTML URL-адреса записываются стандартным образом примерно так: Протокол://Доменное имя/Путь/Параметры вызова файла.

В данном случае URL состоит из четырех составляющих:

- протокол – показывает способ обмена данными между сервером и клиентом;
- доменное имя – имя компьютера, на котором находится информационный ресурс;
- путь – указывает обычный путь к файлу или документу, содержащему информацию на компьютере;
- параметры вызова файла – обычно отделяются от остальных частей URL-адреса символом «?». Если параметров несколько, то они разделяются символом «&».

Пример такого адреса: <http://www.company.com/texts/redis?page=40>.

Для входа на сервер с ограниченным доступом в URL добавляются еще два компонента: имя пользователя и пароль, например для входа на FTP-серверы или серверы электронной почты. В таком случае они отделяются от остальной части адреса знаком «@». Пример такого адреса: <http://name:password@www.company.com/emails/redis?page=40>.

Чаще используются простые URL-адреса. Пример такого адреса:

<http://www.pfu.edu.ru>, <http://www.imeb.ru>.

Если не указан путь к файлу или документу, то открывается файл, имеющий специальное название. В большинстве случаев это index.htm(html) или default.htm(html).

9 Виды информационных ресурсов и инструменты поиска в WWW

Количество информации, размещенной в Интернете, увеличивается ежедневно. При этом перечень видов информационных ресурсов, представленных в сети, также расширяется достаточно быстро.

На данный момент информационные ресурсы можно классифицировать следующим образом.

Веб-страницы. Это основной источник информации в Интернете. По различным оценкам, число веб-сайтов в Интернете превышает 300 млн, а они, в свою очередь, содержат до 550 млрд реальных веб-страниц. Причем количество таких страниц увеличивается на тысячу практически ежедневно. Веб-страницы могут представлять интернет-версии обычных печатных изданий (журналов, газет и т.д.), рекламные сообщения, визитные карточки

предприятий. В отдельную группу веб-страниц можно вынести электронные словари, энциклопедии, самоучители. В Интернете имеется огромное хранилище фотографий и музыкальных произведений.

Веб-сайты компаний, ведущих бизнес в Интернете. Это могут быть электронные магазины, электронные супермаркеты, электронные торговые системы, электронные биржи и аукционы и т.д.

Порталы. Портал – это веб-сайт, обычно служащий начальной точкой поиска информации и содержащий дополнительные сервисы для пользователей.

Публикации в телеконференциях (группах новостей) и форумах. Эти ресурсы Интернета позволяют пользователям обмениваться мнениями по интересующей их тематике.

Для перехода к нужному информационному ресурсу необходимо знать его URL, или перемещаться по гиперссылкам со страницы на страницу в веб-браузере.

Одним из источников информации о ресурсах глобальной сети Интернет является справочник «Желтые страницы Интернета». Это периодическое издание, где перечислены мировые и российские ресурсы сети Интернет.

Однако основным источником информации об Интернете является сам Интернет. Для поиска информационных ресурсов в сети предназначены поисковые системы, которые можно разделить на три группы:

- 1) тематические каталоги;
- 2) поисковые системы, осуществляющие контекстный поиск по индексным базам данных;
- 3) специализированные поисковые системы.

Надо отметить, что деление на тематические каталоги и поисковые системы весьма условно, так как большинство тематических каталогов имеют еще и поиск по ключевым словам, а поисковые серверы имеют еще и тематические каталоги. Разница между ними состоит в том, что каталоги создаются людьми, а поисковые системы – роботами или другим специальным программным обеспечением. Каталоги меньше по объему, но лучше структурированы. Поисковые серверы имеют больший объем, но обеспечивают чисто формальный сбор и анализ материала.

Также иногда к поисковым системам относят базы данных адресов электронной почты, системы поиска в группах новостей, поиск в архивах Gopher, системы поиска FTP-файлов и т.д.

Тематический каталог – это иерархическая база данных, организованная по предметным областям.

В сети Интернет не существует единых стандартов деления информации на разделы. Все тематические каталоги создаются людьми – сотрудниками компаний. Ряд тематических каталогов представляет материалы на нескольких языках (например, каталоги Google, Altavista). Одним из самых первых и самых популярных в Интернете тематических каталогов является каталог Yahoo! (www.yahoo.com).

Поисковые системы – это специально созданные системы для поиска информации в Интернете по заданным условиям (табл. 3.2). Они используют механизмы контекстного поиска по ключевым словам в индексных базах данных. По окончании поиска такая система возвращает список сайтов, которые удовлетворяют критериям поиска. В этом списке обычно представлены ссылки.

Таблица 3.2 – Основные поисковые системы

Поисковая система	URL-адреса
Google	http://www.google.com , http://www.google.ru
Altavista	http://www.altavista.com
Lycos	http://www.lycos.com
Excite	http://www.excite.com
HotBot	http://www.hotbot.com
Yahoo!	http://www.yahoo.com
Rambler	http://www.rambler.ru
Yandex	http://www.yandex.ru
Aport	http://www.aport.ru

Специализированные поисковые системы – это поисковые системы, сосредоточенные на поиске информации по определенной теме. К достоинствам таких систем относятся более релевантные результаты поиска и меньшая затрата времени на поиск.

Надо отметить, что эти системы хранят небольшие объемы данных и имеют небольшой трафик, потому что имеют ограниченный круг пользователей.

Примеры специализированных поисковых систем:

- www.mapquest.com – поисковый сервер географических карт;
- www.maps.expedia.com – поисковик по картам местности;
- www.theater.ru – поисковая система по театрам и репертуарам театров;
- www.weather.com – поисковый сервер, где можно узнать прогноз погоды и др.

Мегапоисковые системы – это специальные системы, которые позволяют проводить одновременный поиск сразу в нескольких поисковых системах. Такие системы позволяют ускорить поиск и расширить охват информационных ресурсов. (www.dogpile.com, www.37.com, meta360.com и т.д.).

Поиск информации в WWW (Internet)

Для задания расширенного поиска во многих поисковых системах используется специальный язык поисковых машин. Использование этого языка облегчает и ускоряет процедуру поиска информации, в особенности это касается поиска по нескольким ключевым словам, где число вариантов значительно вырастает.

Поисковые системы применяют множество различных методов для поиска информации по ключевым словам, но большинство из них основано на Булевой логике. Поисковые пауки и роботы, которые для фильтрации информационных ресурсов используют логические операторы И, ИЛИ, НЕТ, символы «+», «-», «=» и т.п. Также кроме русских значений для логических операторов используются и их английские аналоги.

Булева логика предоставляет возможность комбинации ключевых слов, что позволяет сузить поиск.

Основные логические операторы, используемые в построении запросов по ключевым словам, приведены в таблице 3.3.

Таблица 3. 3 – Основные логические операторы

Логический оператор	Символ	Альтернативный символ	Результат
И (AND)	&	+	Поисковая программа возвращает только те документы, в которых содержатся все слова, объединенные этим оператором
ИЛИ (OR)	1		При использовании этого оператора выбираются те документы, которые содержат хотя бы одно из слов, перечисленных в строке для поиска
НЕ (NOT)	!	-	Этот оператор указывает, какие слова не должны встретиться в искомом документе
РЯДОМ (NEAR)			Рядом с этим оператором обычно ставится число, указывающее максимально допустимое количество промежуточных слов, разделяющее ключевые слова в строке для поиска. Если числовое значение пропущено, то в большинстве случаев по умолчанию это число 10
Grouping	0		Этот оператор указывает, что ключевые слова, заключенные в кавычки, должны содержаться в найденных документах в точно таком же виде, как и в строке для поиска

При помощи специальных поисковых серверов можно найти электронные адреса, почтовые адреса и телефоны людей по всему миру. Такой поиск, например, можно осуществить на поисковом сервере *WhoWhere?* (www.whowhere.com).

Лекция 4. СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИТ

План лекции

1. Назначение системного программного обеспечения.
2. Классификация системного программного обеспечения.
3. Системы программирования.
4. Требования к системному программному обеспечению.

1 Назначение системного программного обеспечения

Программное обеспечение любого персонального компьютера можно разделить на три основные части:

1. Системное программное обеспечение (СПО).
2. Прикладное программное обеспечение (ППО).
3. Инструментальные языки и системы программирования.

Системное программное обеспечение (System Software) – совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ.

СПО управляет ресурсами компьютерной системы и позволяет пользователям программировать в более выразительных языках, чем машинных, язык компьютера. Состав СПО мало зависит от характера решаемых задач пользователя.

Системное программное обеспечение предназначено для:

- создания операционной среды функционирования других программ (другими словами, для организации выполнения программ);
- автоматизации разработки (создания) новых программ;
- обеспечения надежной и эффективной работы самого компьютера и вычислительной сети;
- проведения диагностики и профилактики аппаратуры компьютера и вычислительных сетей;
- выполнения вспомогательных технологических процессов (копирование, архивирование, восстановление файлов программ и баз данных и т.д.).

Данный класс программных продуктов тесно связан с типом компьютера и является его неотъемлемой частью.

Программные продукты данного класса в основном ориентированы на квалифицированных пользователей – профессионалов в компьютерной области: системного программиста, администратора сети, прикладного программиста, оператора.

Однако знание базовой технологии работы с этим классом программных продуктов требуется и конечным пользователям персонального компьютера, которые самостоятельно не только работают со своими программами, но и выполняют обслуживание компьютера, программ и данных.

Программные продукты данного класса носят общий характер применения, независимо от специфики предметной области.

К системным программным продуктам предъявляются высокие требования по надежности и технологичности работы, удобству и эффективности использования.

2 Классификация системного программного обеспечения

В СПО традиционно включают системные управляющие и системные обрабатывающие программы.

Управляющие системные программы организуют корректное функционирование всех устройств системы.

Основные системные функции управляющих программ:

- управление вычислительными процессами и вычислительными комплексами;
- работа с внутренними данными ОС.

Как правило, они находятся в основной памяти. Это **резидентные** программы, составляющие ядро ОС. Управляющие программы, которые загружаются в память непосредственно перед выполнением, называют **транзитными** (transitive).

В настоящее время системные управляющие программы поставляются фирмами-разработчиками и фирмами-дистрибьюторами в виде инсталляционных пакетов операционных систем и драйверов специальных устройств.

Обрабатывающие системные программы выполняются как специальные прикладные задачи, или приложения.

Эти программы поставляются чаще в виде дистрибутивных пакетов, включающих ПО. В пакеты системных программ помимо основных программ, допускающих реконфигурацию, входят специальные настроечные программы, называемые программами инсталляции.

Часто системное ПО компьютера подразделяют на **базовое** и **сервисное** программное обеспечение.

Базовое программное обеспечение (base software) – минимальный набор программных средств, обеспечивающих работу компьютера.

К базовому программному обеспечению компьютера относятся:

- операционные системы и драйверы в составе ОС;

- интерфейсные оболочки для взаимодействия пользователя с ОС (операционные оболочки) и программные среды;
- системы управления файлами.

Операционная система – совокупность программных средств, обеспечивающая управление аппаратной частью компьютера и прикладными программами, а также их взаимодействием между собой и пользователем.

Операционная система предназначена для управления выполнением пользовательских программ, планирования и управления вычислительными ресурсами ЭВМ. Операционная система, с одной стороны, выступает как интерфейс между аппаратурой компьютера и пользователем с его задачами, с другой стороны, предназначена для эффективного использования ресурсов вычислительной системы и организации надежных вычислений.

Системы управления файлами предназначены для организации более удобного доступа к данным, организованным как файлы.

Вместо низкоуровневого доступа к данным с указанием конкретных физических адресов система управления файлами позволяет использовать логический доступ с указанием имени файла. Любая система управления файлами не существует сама по себе – она разработана для работы в конкретной ОС и с конкретной файловой системой.

Сервисное программное обеспечение – программы и программные комплексы, которые расширяют возможности базового программного обеспечения и организуют более удобную среду работы пользователя.

Это набор сервисных, дополнительно устанавливаемых программ, которые можно классифицировать по функциональному признаку следующим образом:

- драйверы специфических и специальных устройств (те, которые не поставляются в составе ОС);
- программы диагностики работоспособности компьютера;
- антивирусные программы, обеспечивающие защиту компьютера, обнаружение и восстановление зараженных файлов;
- программы обслуживания дисков, обеспечивающие проверку качества поверхности магнитного диска, контроль сохранности файловой системы на логическом и физической уровнях, сжатие дисков, создание страховых копий дисков, резервирование данных на внешних носителях и др.;
- программы архивирования данных, которые обеспечивают процесс сжатия информации в файлах с целью уменьшения объема памяти для ее хранения;
- программы обслуживания сети.

Эти программы часто называются **системными утилитами** (к антивирусным средствам этот термин обычно не применяется).

Утилиты – программы, служащие для выполнения вспомогательных операций обработки данных или обслуживания компьютеров (диагностики, тестирования аппаратных и программных средств, оптимизации использования

дискового пространства, восстановления разрушенной на магнитном диске информации и т.п.).

Наибольшее распространение сегодня имеют комплекты утилит: Norton Utilities – фирма Symantec; Checkit PRO Deluxe 2.0 – фирма Touch Stone; PC Tools for Windows 2.0; программа резервного копирования HP Colorado Backup for Windows 95.

3 Системы программирования

Это набор специализированных программных продуктов, которые являются инструментальными средствами разработчика. Программные продукты данного класса поддерживают все этапы процесса программирования, отладки и тестирования создаваемых программ.

Система программирования, как правило, включает следующие программные компоненты:

- редактор текста;
- транслятор с соответствующего языка;
- компоновщик (редактор связей);
- отладчик;
- библиотеки подпрограмм.

Заметим, что любая система программирования может работать только в соответствующей ОС, под которую она и создана, однако при этом она может позволять разрабатывать программное обеспечение и под другие ОС.

Например, одна из популярных систем программирования на языке C/C++ от фирмы Watcom для OS/2 позволяет получать программы и для самой OS/2, и для DOS, и для Windows.

Редактор текста – это программа для ввода и модификации текста.

Трансляторы предназначены для преобразования программ, написанных на языках программирования, в программы на машинном языке. Программа, подготовленная на каком-либо языке программирования, называется исходным модулем. В качестве входной информации трансляторы применяют исходные модули и формируют в результате своей работы объектные модули, являющиеся входной информацией для редактора связей. Объектный модуль содержит текст программы на машинном языке и дополнительную информацию, обеспечивающую настройку модуля по месту его загрузки и объединение этого модуля с другими независимо оттранслированными модулями в единую программу.

Трансляторы делятся на два класса: **компиляторы** и **интерпретаторы**.

Компиляторы переводят весь исходный модуль на машинный язык.

Интерпретатор последовательно переводит на машинный язык и выполняет операторы исходного модуля.

Основной недостаток интерпретатора – низкая скорость работы интерпретируемых программ.

Преимущество интерпретатора перед компилятором состоит в том, что программа пользователя имеет одно представление в виде текста. При компиляции одна и та же программа имеет несколько представлений: в виде текста и в виде выполняемого файла.

Компоновщик, или редактор связей – системная обрабатывающая программа, редактирующая и объединяющая объектные (ранее оттранслированные) модули в единые загрузочные, готовые к выполнению программные модули. Загрузочный модуль может быть помещен ОС в основную память и выполнен.

Отладчик позволяет управлять процессом исполнения программы, является инструментом для поиска и исправления ошибок в программе. Базовый набор функций отладчика включает:

- пошаговое выполнение программы (режим трассировки) с отображением результатов;
- остановка в заранее определенных точках;
- возможность остановки в некотором месте программы при выполнении некоторого условия;
- изображение и изменение значений переменных.

Загрузчик – системная обрабатывающая программа, объединяющая основные функции редактора связей и программы выборки в одном пункте задания. Загрузчик помещает находящиеся в его входном наборе данных объектные и загрузочные модули в оперативную память, объединяет их в единую программу, корректирует перемещаемые адресные константы с учетом фактического адреса загрузки и передает управление в точку входа созданной программы.

Средства сетевого доступа обеспечивают обработку, передачу и хранение данных в сети.

Сетевые операционные системы, например, предоставляют пользователям различные виды сетевых служб (управление файлами, электронная почта, процессы управления сетью и др.).

Ключом к использованию этих ресурсов является сервер, специальная программа на компьютере, подключенная к сети, которая принимает запросы (или команды) и посылает ответы автоматически.

Программы, предназначенные для подачи запросов серверу, называются программами-клиентами. Сервер предназначен для их обслуживания. Клиент посылает запросы пользователя на сервер, используя стандартизированный формат, называемый протоколом. Ответ сервера содержит информацию, представленную в виде файла, содержащего данные того или иного формата.

Постоянно ведется разработка все новых программ-клиентов, предлагающих более удобные способы взаимодействия с сервером.

Примерами программ-клиентов могут служить, например, такие приложения, как Netscape Navigator, Internet Explorer.

Таким образом, в системном ПО можно выделить пять групп системных программ:

- операционные системы;
- интерфейсные оболочки для взаимодействия пользователя с ОС (операционная оболочка) и программные среды;
- системы управления файлами;
- системы программирования;
- утилиты;
- средства сетевого доступа.

В ходе развития компьютерных систем наиболее используемые прикладные программы могут быть перенесены на уровень системных, что позволяет использовать их в различных приложениях. Например, средства управления диалоговым взаимодействием с пользователем в системных оболочках (типа Windows).

С другой стороны, наиболее распространенные и критические по времени системные функции были частично или полностью реализованы аппаратно. Например, средства управления многопрограммным защищенным режимом и средства управления мультимедиа-устройствами в процессорах фирмы Intel.

4 Требования к системному программному обеспечению

Системные программы должны удовлетворять следующим требованиям:

- прозрачность работы;
- гарантированная надежность выполнения в соответствии со спецификациями (спецификациями называются функциональные требования);
- максимальная скорость выполнения;
- минимальные затраты на хранение машинных кодов;
- поддержка стандартных средств связи с прикладными программами.

Эффективность системных программ зависит от времени их создания и надежности исполняемого кода. Требование эффективности системных программ вызывает необходимость использования специальных языков:

- машинно-ориентированных, типа языка Assembler;
- высокого уровня типа C или C++.

К типам данных этих языков отнесены указатели на данные различных типов или адреса данных и программных объектов.

Работа с большинством пакетов для разработки системного программного обеспечения предполагает знание и использование ассемблера для создания модулей и ассемблерных вставок.

Лекция 5. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ

План лекции

1. Классификация прикладных программных средств.
2. Классификация текстовых процессоров.
3. Принципы создания и редактирования документа в MS Word.
4. Google Документы в системе Google Диск.

1 Классификация прикладных программных средств

Прикладные программные средства (ППС) составляют категорию программных средств, обращенных непосредственно к пользователю. Их назначение состоит в том, чтобы пользователь с помощью компьютера мог решать свои повседневные задачи, приобретал определенные навыки, проводил свой досуг.

К основным прикладным программным средствам можно отнести:

- **системы подготовки текстов:** текстовые редакторы (процессоры), настольные издательские системы, позволяющие наиболее удобным образом создавать, корректировать текстовую и графическую информацию: Word, Page Maker, Page Plus, Corel Ventura, QuarkXPress, MS Publisher;

- **системы компьютерной графики:** графические редакторы – пакеты, предназначенные для обработки графической информации. Различают редакторы растровой графики, рисующие изображение по точкам разных цветов: Aldus Photo, Adobe PhotoShop, Photo Finish, Paintbrush, редакторы векторной графики, рисующие изображение линиями: Corel Draw, Adobe Illustrator, MS Draw, и редакторы фрактальной графики.

- **демонстрационная графика:** конструкторы графических образов деловой информации, предназначенные для представления в наглядной и динамичной форме результатов аналитических исследований: Power Point, World Perfect Presentation, Frulance Graphics;

- **пакеты программ мультимедиа** – предназначены для использования ЭВМ для отображения и обработки аудио и видеоинформации;

- **табличные процессоры** (электронные таблицы), обеспечивающие удобные средства работы с двумерными таблицами: Lotus, SuperCalc, Excel;

- **системы управления базами данных (СУБД)**, которые обеспечивают эффективные действия по обработке больших объемов информации, их поиску,

корректировке и хранению;

- **пакеты прикладных программ, содержащие стандартные подпрограммы**, для производства расчетов по известным алгоритмам;

- **прикладное программное обеспечение, используемое для автоматизации работ в конкретной области**: финансовые программы для ведения личных финансов, автоматизации бухучета малых и крупных фирм, разработки технико-экономических обоснований финансовых сделок – MS Money, Money Counts, Бемби, 1С Бухгалтерия; система управления предприятием – автоматизированный комплекс «Галактика», САПР – AutoCAD;

- **обучающие системы** в различных областях знаний (энциклопедии);

- **игры** развлекательного и тренировочного характера;

- **вирусы** – программы, мешающие работе ПЭВМ, или разрушающие информацию и оборудование компьютера; они используются для защиты запатентованных программ или систем, нанесения информационного ущерба;

- **антивирусы** – программы для обнаружения и обезвреживания вирусов.

Могут использоваться также интегрированные системы, объединяющие наиболее часто используемые прикладные программы.

2 Классификация текстовых процессоров

Текстовый процессор – это специальная программа для создания и обработки текстов, которая позволяет подготовить, отформатировать и распечатать текстовые документы различной длины и структуры.

По типу обрабатываемого объекта текстовые процессоры можно разделить на четыре группы:

- **редакторы текстов** – предназначены для создания и редактирования простых текстов или программ: MultiEdit, Norton Editor;

- **редакторы научных текстов** – обеспечивают подготовку и редактирование научных текстов, содержащих большое количество математических формул, графиков, специальных символов и т.п.: MathOr, Tex;

- **редакторы документов** – предназначены для работы со сложными документами, которые могут содержать вложенные документы, страницы, абзацы, таблицы, диаграммы, рисунки, которые могут быть созданы и в других приложениях: Лексикон, AmiPro, Microsoft Word. Современные редакторы документов имеют возможности подготовки документов с формулами;

- **издательские системы** – используются для подготовки больших, сложных документов (книги, альбомы, журналы, газеты), а также для верстки текста: Corel Ventura Publisher, Adobe PageMaker, QuarkXPress. Верстка текста заключается в размещении текста по страницам создаваемого документа, вставке рисунков, использовании различных шрифтов применительно к документам, уже созданным с помощью редакторов документов. Эти системы

предполагают применение редакторов документов для предварительной подготовки материалов, а для окончательной верстки используется уже сама издательская система. Многие функциональные возможности издательских систем имеют и современные редакторы документов.

В настоящее время отличия между процессорами различных типов перестали быть принципиальными.

Текстовый процессор MS Word является самым популярным редактором документов и используется для подготовки текстовой документации различного уровня сложности. Вводимый документ отображается на экране в том виде, в каком он будет выведен на печать. Пользователю доступны такие средства редактирования, как изменение типа и начертания шрифта, установка и выравнивание абзацев, установка межстрочных интервалов, выделение рамками, многоколоночный текст, выделение абзацев, висячие отступы, автонумерация страниц и многое другое. Созданный документ можно сохранить на диске, в том числе и с указанием пароля для его защиты, или считать с диска.

В текст можно вставлять таблицы, ячейки которых могут содержать как текст, числа, графические объекты, формулы и т.д., но не могут содержать таблицы. Встроенное средство построения диаграмм позволит легко построить графики различных типов. В таблицах можно проводить различные несложные вычисления. Текст может сопровождаться формулами, а также графикой. Графические объекты можно вставлять из уже созданных файлов различных форматов, а также создавать, используя встроенный графический редактор.

Технология IntelliSense позволяет автоматизировать многие типовые и часто используемые операции, такие как изменение строчного символа в начале строки на прописной, динамическая коррекция ошибок, вставка автотекста. При многократном использовании одной и той же фразы автотекст позволяет указать сокращение, при появлении которого в тексте будет выполняться автоматическая замена на нужную фразу.

MS Word использует следующие типы файлов:

- ***.docx** – файл основного документа;
- ***.dotx** – файл шаблона;
- ***.wmf** – файл иллюстрации.

Интерфейс MS Word, как и любого программного приложения Windows, представляет собой набор меню и окон. При запуске MS Word автоматически создается одно пустое окно с заголовком **Документ**. Такое окно называется **Окном документа**. Окно документа, с которым в данный момент работает пользователь, называется **активным**. Заголовок **Документ** с порядковым номером окна после сохранения в файл заменяется именем файла, например, **Отчет**.

Команды MS Word можно вызвать одним из трех способов:

- **с помощью командного пиктографического меню** – выбор команды осуществляется из раскрывающегося списка;

– из **динамического (контекстно-зависимого) меню**, появляющегося при нажатии правой кнопки мыши и содержащего команды, относящиеся к выделенному документу;

– **при помощи «горячих» клавиш** – комбинаций клавиш, позволяющих вызвать любую команду меню. Комбинации клавиш зависят от настроек операционной системы.

Все команды текстового редактора MS Word интуитивно понятны и не требуют подробных объяснений.

3 Обзор возможностей текстового редактора MS Word

3.1 Принципы создания и редактирования документа в MS Word

Создавать документ в текстовом редакторе MS Word можно одним из следующих способов: набрать исходный текст, а затем его отформатировать, либо набирать текст и последовательно его форматировать. Под **форматированием** текстового документа понимается установка параметров страницы, шрифта, абзацев, отступов, выравнивания текста, нумерации страниц и т.д.

Создание любого документа начинается с выбора из меню **Файл** с помощью команды **Создать** тип создаваемого документа – обычный документ, письмо, записка, резюме и т.д., после чего его следует сохранить. Для этого можно воспользоваться командами **Сохранить** или **Сохранить как** из меню **Файл**. В открывшемся окне следует выбрать диск, на котором будет сохранен документ, папку на диске и указать имя документа.

Далее следует установить параметры страницы, используя команду **Поля** из меню **Разметка страницы**. В окне **Поля** необходимо указать размеры верхнего, нижнего, левого и правого полей в сантиметрах, на вкладке **Размер** уточнить формат (по умолчанию установлен формат А4 размером 21х29,7 см), на вкладке **Ориентация** – ориентацию (книжная, альбомная) листа. Сохранить эти изменения в уже существующем файле, используя команду **Сохранить**. После этого можно приступить к набору текста.

Основной структурной единицей текста в Word является *абзац*, под которым понимают группу смежных строк текста, первая из которых может начинаться отступом или выступом, а все последующие начинаются с отличного от пробела символа. Концом абзаца считается строка, за которой следует либо конец документа, либо начало следующего абзаца. Конец абзаца обозначается символом ¶, который отображается на экране при нажатой пиктограмме ¶ (непечатаемые символы).

Различают следующие структурные элементы страницы: основной текст (строки, абзацы, таблицы, рисунки), верхний и нижний колонтитулы, сноски.

Для установки отступов можно воспользоваться меню **Главная, Абзац** или использовать горизонтальную линейку. Не допускается установка абзацных отступов с помощью пробела или клавиши **Tab**.

Для установки отступов с помощью меню **Главная** необходимо вызвать команду **Абзац**. В окне **Отступы и интервалы** установить размер отступа красной строки (по умолчанию 1,25–1,27 см), **межстрочный интервал** (по умолчанию – одинарный), **выравнивание текста** (по ширине, по центру, правому или левому краю) и при необходимости **отступ** или **выступ** справа или слева. Отступ измеряется в положительных единицах, выступ – в отрицательных, например, 0,63 см – отступ, -0,63 см – выступ.

Используя верхнюю горизонтальную линейку, можно задавать отступ красной строки, используя нижнюю горизонтальную линейку – отступ или выступ справа или слева. При работе с документом не забывать записывать изменения в файл, используя команду **Сохранить** (Ctrl+S).

Работая с текстом документа, можно изменять его фрагменты. Для этого фрагмент текста необходимо выделить, а затем произвести необходимые изменения, например, выбрать вид шрифта, размер, начертание, выровнять его по правому краю, по центру, по левому краю, по ширине. Фрагмент текста можно удалить из документа, выделив его и нажав клавишу <Delete>.

Фрагмент текста можно копировать из одного места в другое, оставляя его копию в исходном месте, либо перемещать (вырезать) фрагмент и вставлять его в другое место без сохранения его копии в исходном месте. Для этого используется буфер обмена.

При копировании (вырезании) необходимо выделить фрагмент текста, затем из меню **Главная** выбрать команду **Копировать (Вырезать)**, установить курсор в том месте, где должен находиться фрагмент и вставить фрагмент, используя команду **Вставить** меню **Главная**. Все эти действия также можно выполнять, используя динамическое или пиктографическое меню, а также «горячие» клавиши.

3.2 Объекты в MS Word

Команды меню **Вставка** позволяют вставлять в документ Microsoft Word различные объекты: таблицы, рисунки, картинки, фигуры, объекты Smart Art, диаграммы, специальные символы и пр. Так, например, для вставки иллюстрации в текст используется команда **Картинки** (ЛКМ), после выполнения которой открывается библиотека иллюстраций, входящих в MS Word. Для размещения иллюстрации в тексте после ее вставки следует задать способ размещения рисунка на странице. Удобно это делать через динамическое меню, вызываемое правой клавишей мыши.

Для написания сложных формул Microsoft Word предлагает специальную программу – редактор формул **MS Equation 3.0**, вызвать которую можно, используя команду **Объект**. В открывшемся окне из предложенного списка следует выбрать тип объекта **Microsoft Equation 3.0**. После чего откроется окно редактора формул. Пользуясь меню, можно написать любую формулу.

MS Word позволяет создавать и использовать **таблицы** для размещения в них информации различных видов – текстовой, числовой, графической и др.

Каждая таблица состоит из ячеек, сгруппированных в строки и столбцы. Столбцы именуются латинскими буквами, строки нумеруются арабскими числами. Число строк и столбцов таблицы ограничено используемой версией редактора, например, в MS Word 2000 количество столбцов – не более 63. Адрес ячейки образуется из имени столбца и номера строки, например, G5.

Выделяют такие этапы работы с таблицами, как создание таблиц, ввод и редактирование данных; форматирование таблиц; организация вычислений и сортировка.

Для создания таблицы следует воспользоваться командой **Таблица** из меню **Вставка**, указав количество столбцов и строк таблицы. Ввод и редактирование данных (текста, чисел и т.д.) осуществляется как обычно.

Для форматирования таблицы используются команды меню **Макет**, такие как **Вставить строку** (столбец), **Удалить ячейки**, **Объединить ячейки** и др., а также команда **Границы и заливка** (меню **Конструктор**), которая позволяет задавать тип, размер и цвет линий обрамления ячеек таблицы, цвет заливки и др.

В любую ячейку таблицы может быть введено не только значение, но и формула, которая может состоять из знаков операций, операндов и стандартных функций.

Формат ввода формулы: = **формула (список аргументов)**.

Основные стандартные функции, поддерживаемые MS Word:

- ABS(X) – модуль числа;
- AND – логическое И;
- AVERAGE() – вычисляет среднее арифметическое аргументов;
- COUNT() – подсчитывает число ячеек в столбце от активной ячейки до конца столбца;
- IF(X,Y,Z) – используется для проверки логического условия и организации перехода в зависимости от его выполнения или невыполнения. X – логическое условие, Y – действие, выполняемое, если условие истинно, Z – действие, выполняемое, если условие ложно;
- INT(X) – приводит значение аргумента к целочисленному типу;
- MAX(), MIN() – используется для поиска максимума или минимума среди аргументов;
- MOD(X,Y) – возвращает остаток от целочисленного деления аргумента X на Y;
- NOT(X) – отрицание (логическое НЕ);
- OR(X,Y) – выбор (логическое ИЛИ);
- PRODUCT() – вычисляет произведение аргументов;
- SIGN(X) – возвращает знак аргумента;
- SUM() – вычисляет сумму аргументов.

Функции с пустыми скобками могут принимать любое число аргументов. В них можно указывать ссылки на ячейки перечислением (A1, A2, B6), диапазоном (A1:A10), блоком (A1:C5).

Вместо диапазона ячеек можно использовать позиционные аргументы, обозначаемые следующими *ключевыми* словами:

- ABOVE – диапазон ячеек от активной до начала столбца;
- BELOW – от активной ячейки до конца столбца;
- LEFT – от активной ячейки до начала строки;
- RIGHT – от активной ячейки до конца строки.

В формулах допустимо использование таких знаков операций, как «+» – сложение, «-» – вычитание, «*» – умножение, «/» – деление, «%» – процент, «^» – возведение в степень, и операторов сравнения «=» – равно, «<>» – не равно, «>» – больше, «<» – меньше, «>=» – больше либо равно, «<=» – меньше либо равно.

Для выполнения вставки формулы (функции) в ячейку необходимо установить в нее курсор, из меню **Макет** выбрать команду **Формула**, в открывшемся диалоговом окне для выполнения простых вычислений в поле **Формула** ввести формулу. Для более сложных вычислений в поле **Вставить функцию** выбрать нужную функцию.

Например, формула **=SUM(ABOVE)** позволит просуммировать числовые значения всех ячеек, начиная от активной до начала столбца. Текст, пустые ячейки и графические объекты будут игнорироваться.

Для просмотра формул в ячейках нужно нажать <Shift> + <F9>, повторное нажатие этой комбинации позволит вернуться в нормальный режим.

Данные таблицы можно отсортировать. Для этого необходимо выделить данные, подлежащие сортировке, вызвать команду **Сортировка** из меню **Макет**. В открывшемся окне выбрать объект сортировки (столбец1, столбец2 и т.д.), указать критерий сортировки (текст, число, дата) и способ упорядочивания (по возрастанию или убыванию). Для разных столбцов допустимо указание различных критериев сортировки и способов упорядочивания. При этом необходимо учитывать, что каждому значению одной ячейки строки таблицы соответствует значение другой ячейки той же строки таблицы, поэтому при сортировке таких значений следует указывать не один столбец, а несколько. В противном случае может возникнуть несоответствие отсортированных данных исходным. Сортировать можно не только данные таблицы, но и абзацы текста.

3.3 Использование стилей оформления документа

Использование стандартных стилей оформления (меню **Главная** → **Стили**) позволяет формировать оглавление документа. Если заголовки документа были оформлены с использованием встроенных стилей Word, то можно составить оглавление, используя команду **Оглавление** из меню **Ссылка**. В появившемся диалоговом окне на вкладке **Оглавление** задать вид оглавления, выбрав соответствующий шаблон. Оглавление будет сформировано после нажатия кнопки ОК в месте, где находился курсор ввода.

Для добавления верхних и нижних колонтитулов, а также корректировки номеров страниц используется команда **Колонтитулы** из меню **Вставка**. После выполнения этой команды текст документа становится временно недоступным для внесения изменений, а в области верхнего и нижнего колонтитулов можно добавлять любую информацию. Повторный выбор команды вернет пользователя в прежний режим работы.

3.4 Ссылки на литературу и список литературы

MS Word позволяет автоматизировать формирование списка литературных источников и размещение в тексте ссылок на эти источники. Для этого служат команды вкладки **Ссылки и списки литературы** меню **Ссылки**.

4 Google Документы в системе Google Диск

Google Диск – это сервис хранения, редактирования и синхронизации файлов, разработанный компанией Google. Google Диск был представлен 24 апреля 2012 года. Согласно представителям Google с ноября 2013 года по май 2019 года количество активных пользователей выросло со 120 млн до более 1 млрд человек. Его функции включают хранение файлов в Интернете, общий доступ к ним и совместное редактирование.

В состав Google Диска входят Google Документы, Таблицы и Презентации – набор бесплатных офисных приложений для совместной работы над текстовыми документами, электронными таблицами, презентациями, чертежами, веб-формами и другими файлами. Общедоступные документы на Диске индексируются поисковыми системами. С его помощью можно создавать и редактировать документы онлайн и работать над ними вместе с другими пользователями в режиме реального времени.

С октября 2012 года эти названия официально используются для текстового процессора, редактора электронных таблиц и приложения для работы с презентациями.

Все три сервиса доступны в виде веб-приложений, приложений Chrome с поддержкой офлайн-режима, а также мобильных приложений для Android и iOS и поддерживают форматы Microsoft Office. Помимо этих редакторов в пакет входят:

- Google Формы – <https://forms.google.com>;
- Google Рисунки;
- Google Сводные таблицы.

Формы и Сводные таблицы доступны только в виде веб-приложений, а Рисунки – ещё и как приложение Chrome. Google Документы полностью интегрированы с Google Диском, на котором хранятся все создаваемые в редакторах файлы.

Google Документы созданы на основе двух продуктов: Writely и Google Таблиц. Writely – это текстовый процессор компании Upstartle, запущенный в

августе 2005 года. Принципы создания и редактирования документов в Google Документы практически аналогичны принципам работы в среде MS Word.

Google Таблицы были запущены 6 июня 2006 года, после того как Google приобрела 2Web Technologies, и сначала предлагались в виде экспериментальной версии.

Редактор презентаций был добавлен 17 сентября 2007 года после приобретения Tonic Systems (17 апреля 2007 года).

Google Документы официально поддерживают браузеры Firefox (двух последних версий), Internet Explorer (версии 9 и более поздних), Safari и Chrome, а также операционные системы Windows, Apple MacOS, Linux и Chrome OS.

Google Диск viewer позволяет просматривать файлы следующих форматов:

Обычные файлы:

- архивы (ZIP, RAR, TAR, GZIP);
- аудиофайлы (MP3, MPEG, WAV, OGG, OPUS);
- изображения (JPEG, PNG, GIF, BMP, TIFF, SVG);
- файлы разметки/кода (CSS, HTML, PHP, C, CPP, H, HPP, JS, JAVA, PY);
- текстовые файлы (TXT);
- видеофайлы (WebM, MPEG4, 3GPP, MOV, AVI, MPEGPS, WMV, FLV, OGG).

Файлы Adobe:

- Autodesk AutoCad (DXF);
- Illustrator (AI);
- Photoshop (PSD);
- документы переносимого формата (PDF);
- Postscript (EPS, PS);
- масштабируемые векторные рисунки (SVG);
- изображения в формате TIFF;
- TrueType (TTF).

Файлы Microsoft:

- Excel (XLS, XLSX);
- PowerPoint (PPT, PPTX);
- Word (DOC, DOCX);
- XML Paper Specification (XPS);
- файлы Microsoft Office, защищенные паролем.

Файлы Apple:

- файлы редактора (KEY, NUMBERS).

Лекция 6. ТЕХНОЛОГИИ И СИСТЕМЫ ОБРАБОТКИ ТАБЛИЧНОЙ ИНФОРМАЦИИ

План лекции

1. Табличные процессоры и их функциональные возможности.
2. Основные понятия табличного процессора MS Excel.
3. Технология разработки электронной таблицы.
4. Вычисления в таблицах.
5. Встроенные функции рабочего листа ТП MS Excel.
6. Средства деловой графики.

1 Табличные процессоры и их функциональные возможности

Для проведения расчетов данных, представленных в табличной форме, были разработаны пакеты прикладных программ, получивших название табличный процессор.

Табличный процессор (ТП) – это набор языковых и программных средств, предназначенный для создания электронных таблиц, заполнения их данными и последующей их обработки, то есть специальный комплекс программ для управления электронной таблицей. **Электронная таблица (ЭТ)** – компьютерный эквивалент обычной таблицы, в клетках (ячейках) которой записаны данные различных типов: текст, даты, формулы, числа. Использование табличных процессоров целесообразно в тех случаях, когда:

- числа, с которыми требуется работать при решении поставленной задачи, можно расположить в виде таблицы, т.е. в строках и графах;
- числа в одной строке или графе связаны с числами в других строках или графах и предполагается использование математических вычислений над данными таблицы;
- предполагается статистическая обработка данных; возможно частое изменение информации; отслеживается большое число показателей; предполагается изготовление нужного числа копий табличных документов.

Первыми программами, реализующими концепцию электронных таблиц, были VisiCalk (разработана Д. Бриклиным и Б. Фрестоном в 1979 г.), SuperCulc, Lotus, QutroPro. Электронная таблица представлялась экраном дисплея с сеткой, разделяющей его на столбцы и строки, обозначенные соответственно буквами латинского алфавита и цифрами. Затем были созданы интегрированные пакеты Framework, Мастер и др. Эти пакеты совмещали в себе табличные процессоры, базы данных и текстовые редакторы.

Впоследствии появляются прикладные пакеты, работающие под ОС Windows, такие как Excel, Works, Lotus 1-2-3 (5) и др. На сегодня по статистике лидером среди программного обеспечения данного типа является табличный процессор (ТП) Excel фирмы Microsoft (его используют более 80 % пользователей во всем мире). Технология работы с табличным документом аналогична процедурам подготовки текстовых документов: все внесенные изменения сразу же отображаются на экране компьютера. Если один раз отработать форму таблицы и установить характер необходимых расчетов, то в дальнейшем технологический процесс сводится только к вводу данных и, при необходимости, к их редактированию.

2 Основные понятия табличного процессора Excel

Табличный процессор Microsoft Excel (ТП MS Excel) – это наиболее популярная программа для работы с электронными таблицами. В качестве программного приложения она входит в пакет Microsoft Office. Функциональные возможности Excel позволяют широко использовать его для финансовой обработки данных, научных расчетов, инженерно-технических расчетов, автоматизации учетно-контрольной деятельности, эффективной обработки больших объемов информации, заданных в табличном виде.

К основным возможностям табличного процессора Excel относятся:

- быстрое построение, корректировка, сохранение таблиц;
- использование стандартных функций для проведения расчетов в таблицах;
- защита табличных данных;
- построение по табличным данным двух- и трехмерных графиков и диаграмм, содержит разнообразные инструменты для редактирования графиков и диаграмм, включая средства для создания смешанных двухмерных графиков;
- поддержка OLE-технологии и технологии drag-and-drop;
- работа со связанными таблицами;
- работа с таблицей как с базой данных;
- одновременная работа с несколькими книгами и др.

Кроме того, ТП MS Excel обеспечивает:

- быстрое построение таблиц любой формы одноразового и многократного пользования и сохранение на магнитном носителе в виде отдельного файла с последующим чтением;
- возможность обработки таких типов данных, как числа, даты, формулы;
- возможность корректировки уже созданной таблицы (перемещение строк и столбцов, их копирование, удаление и т.д.);
- возможность выбора цветового оформления таблицы, а также выбора различных шрифтов и стилей, включая автоформатирование;
- наличие механизма мастеров, которые позволяют автоматизировать

выполнение операций (например, мастер диаграмм или мастер функций);

- автоматизированную обработку таблиц с помощью макрокоманд, а также модулей на встроенном языке программирования Visual Basic for Application;

- поддержку работы в сети.

Основными понятиями табличного процессора Excel являются:

- **Рабочая книга** – это файл, используемый для обработки и хранения данных, состоящий из отдельных листов. Рабочая книга сохраняется в едином файле, имеющем по умолчанию расширение *.xls (файл-таблица) и может содержать до 255 листов, расположенных в произвольном порядке, следующих типов: рабочий лист; лист с диаграммой; лист макросов.

- **Лист** – разделен на строки и столбцы и служит основой для выполнения вычислений. Рабочий лист состоит из 256 (28) столбцов и 65536 (216) строк.

- **Ячейка** – область, определяемая пересечением столбца и строки электронной таблицы (ЭТ). На рабочем листе расположено $16\,777\,216$ ($28 \times 216 = 224$) ячеек.

- **Текущая (активная) ячейка** – ячейка ЭТ, в которой в данный момент находится курсор.

- **Адрес ячейки** определяется названием (номером) столбца и номером строки. Формат адресов в Excel представлен двумя видами: A1-формат и R1C1-формат.

В первом случае столбцы именуются буквами, например, A, B, C ... и так далее или сочетаниями букв, например, AQ, BU, IV. Строки нумеруются числами, например, 1, 2, 3 ... и так далее.

Во втором случае используется буквенно-цифровое обозначение: для строк (R – Row – строка) – от R1 до R256, для столбцов (C – Column – столбец) – от C1 до C65536.¹

При необходимости задать адрес некоторой ячейки, находящейся на другом листе в пределах одной и той же рабочей книги, указывается имя этого листа и адрес ячейки, например, «=Лист1!B3». Если необходимо сослаться на ячейку, находящуюся в другой книге, то указывается имя этой книги, имя листа этой книги и адрес ячейки. **Например**, «=[Книга]задача!B2». Здесь **книга** – имя книги Excel, в которой расположена нужная ячейка, **задача** – имя листа, содержащего необходимую ячейку, **B2** – адрес конкретной ячейки.

Таким образом, с помощью адреса можно найти любую ячейку в любой рабочей книге. Адреса ячеек в MS Excel могут быть заданы в виде *диапазона* или *блока*. *Диапазон ячеек* – группа смежных ячеек в строке или столбце. Для ссылки на диапазон необходимо задать адрес первой и последней ячеек диапазона через двоеточие, например, A1:A5. *Блок ячеек* – группа смежных ячеек, определяемая адресом верхней левой и нижней правой ячеек в

¹ Изменить формат адресации можно, выбрав из меню Офис (Файл) ==> Параметры Excel ==> Формулы ==> Стилль ссылок R1C (Excel 2010).

прямоугольнике, образуемом блоком. Например, D4:F13.

– **Ссылка** – указывает на ячейку или диапазон ячеек листа, которые требуется использовать в формуле (то есть адрес ячейки, помещенный в формулу).

– **Формула**² – это конструкция, начинающаяся со знака «=», состоящая из математических операторов, значений, ссылок на ячейки и имен функций, при этом результатом выполнения формулы является некоторое новое значение.

Окно приложения табличного процессора Excel имеет вид, представленный на рисунке 6.1.

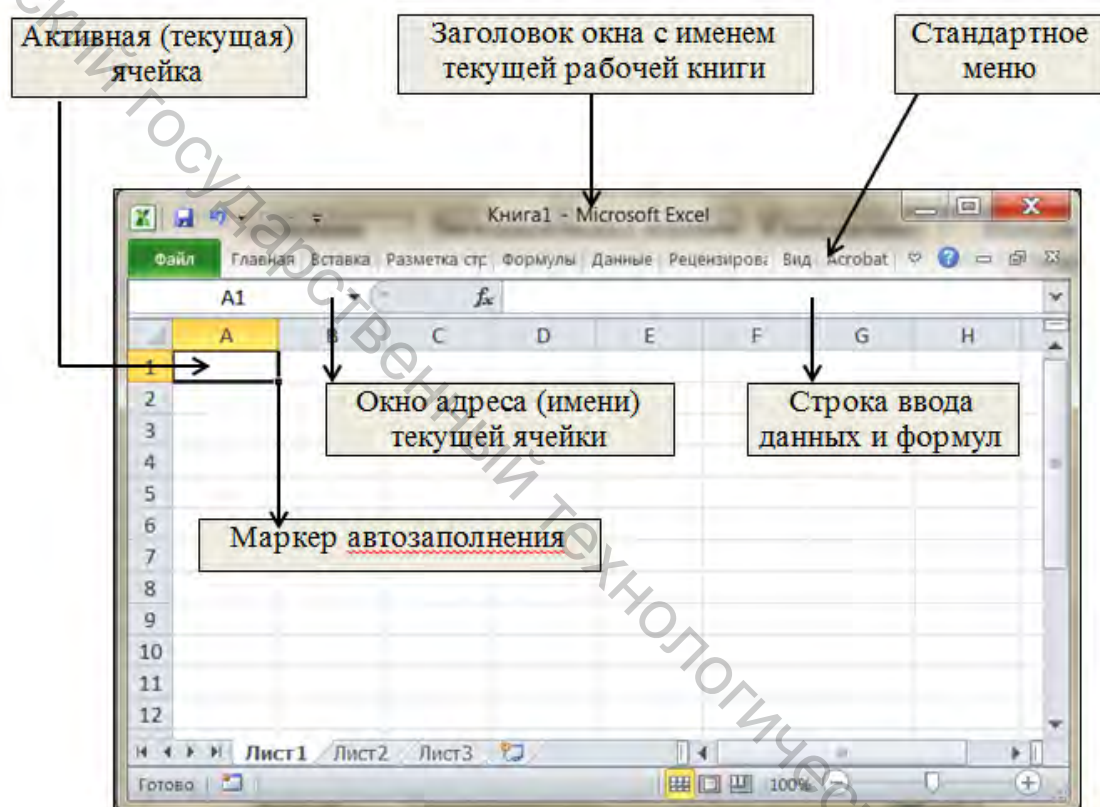


Рисунок 6.1 – Окно Microsoft Excel 2010

Основное (стандартное) меню содержит команды программы. Строка формул отображает вводимые в ячейку данные и формулы, содержащиеся в текущей ячейке. Поле имен отображает адрес активной ячейки.

Принятые в Excel расширения файлов:

XLSX – файл рабочей книги;

XLTX – файл деловой графики;

XLTX – файл шаблона;

XLAM – файл дополнительных макрок команд;

XLTX – файл описания пиктографического меню и пр.

² Подробнее о ссылках и формулах ТП MS Excel изложено в разделе 4 лекции 2.

3 Технология разработки электронной таблицы

При создании электронных таблиц средствами табличного процессора пользователь производит ряд действий, характерных для этого вида работы. Эти действия составляют технологию создания электронной таблицы:

1. Проектирование и разработка форм выходных документов (на бумаге), а также алгоритмов получения расчетных данных.
2. Разработка электронной таблицы, т.е. создание заголовка, шапки, внесение формул в расчетные колонки.
3. Ввод данных и получение расчетных значений.
4. Сохранение таблицы на внешнем носителе.
5. Вывод таблицы на печать.

4 Вычисления в таблицах

Формулы. Все вычисления в электронных таблицах производятся с помощью формул. Любая формула в MS Excel должна начинаться со знаков «равно», «плюс» или «минус». Без этих знаков формула интерпретируется как текст. Каждая формула состоит из операторов и операндов.

Операнд – это числовое значение, текст, ссылка на ячейку или группы ячеек. В качестве операндов могут выступать и встроенные функции Excel.

Оператор – это знак операции, которую следует выполнить над операндами формулы. Операторы Excel можно классифицировать следующим образом: арифметические, текстовые, операторы сравнения и операторы ссылок.

Операторы сравнения используются для сравнения двух значений. Результатом сравнения является логическое значение ИСТИНА или ЛОЖЬ. Используются следующие операторы сравнения: = (знак равенства), > (знак больше), < (знак меньше), >= (знак больше или равно), <= (знак меньше или равно), <> (знак не равно).

Арифметические операторы служат для выполнения арифметических действий над числами. В таблице 6.1 представлены арифметические операторы, которые могут использоваться в Excel.

Таблица 6.1 – Арифметические операторы

Обозначение	Название	Действие
+	Знак плюс	Сложение
–	Знак минус	Вычитание
*	Звездочка	Умножение
/	Косая черта	Деление
%	Знак процента	Вычисление процента от числа
^	Тильда	Возведение в степень

Текстовый оператор конкатенации (амперсанд) используется для объединения нескольких текстовых строк в одну строку и обозначается &.

Оператор ссылки служит для описания ссылок на диапазоны или блоки, то есть группы смежных ячеек.

Для ссылки на диапазон необходимо задать адрес первой и последней ячеек диапазона через двоеточие, *например*, A1:A5, для ссылки на блок задается адрес верхней левой и нижней правой ячеек блока через двоеточие, *например*, A1:E5.

Возможно использование следующих операторов ссылок:

: (двоеточие) – ссылка на диапазон, *например*, A2:A10, или блок, *например*, A2:C10;

; (точка с запятой) – оператор объединения, объединяет несколько ссылок в одну ссылку, *например*, =СУММ(A2:A10;C2:C10);

(пробел) – оператор пересечения множеств, служит для ссылки на общие ячейки двух диапазонов, *например*, (B2:D10 C4:C6).

При организации вычислений формулы можно копировать из ячейки в ячейку, при этом адреса ячеек, указанные в формулах, могут изменяться в соответствии с типом ссылки, используемой в формуле.

Ссылкой на ячейку (ссылкой на адрес ячейки) называется указание адреса этой ячейки в формуле или функции Excel. Различают относительную ссылку, абсолютную ссылку и смешанную ссылку на ячейку.

Относительная ссылка на ячейку записывается в виде последовательности заголовков столбца и строки, *например*, A3, D4. Ее особенность заключается в том, что при переносе формулы на некоторое число позиций эта ссылка на ячейку заменяется ссылкой на другую ячейку, смещенную относительно исходной на такое же число позиций и в том же направлении, что и формула.

Абсолютная ссылка на ячейку записывается с использованием знака \$, *например*, \$A\$4, \$A\$2, и при переносе формулы из одной ячейки в другую адрес ячейки, на которую ссылается формула, остается неизменным.

Смешанная ссылка на ячейку также использует знак \$, *например*, \$A3, B\$2, и представляет собой сочетание относительной и абсолютной ссылок. При копировании формулы каждая часть ссылки ведет себя соответственно указанному адресу: абсолютная часть адреса остается неизменной, относительная часть адреса меняется.

5 Встроенные функции³

Функция в Excel представляет собой объединение нескольких вычислительных операций над значениями, выступающими как аргументы, для решения определенной задачи.

³ С форматами и примерами использования всех встроенных функций можно подробно ознакомиться в справочной системе ТП MS Excel.

Аргументы – значения исходных данных для функции, используемые для выполнения операций или вычислений. Аргументы функции должны быть указаны в порядке, определенном для данной функции. Каждый из них должен иметь требуемый данной функцией тип. Аргументами функции могут являться числовые значения, ссылки на ячейки, диапазоны, имена, текстовые строки, выражения и вызовы других функций.

В Excel для упрощения расчетов имеется широкий набор встроенных функций, которые делятся на категории. Каждая из этих категорий включает функции, обеспечивающие определенные вычисления. Краткая характеристика категорий функций Excel приведена в таблице 6.2.

Таблица 6.2 – Краткая характеристика категорий функций Excel

Категория	Краткая характеристика
Финансовые	Функции для осуществления типичных финансовых расчетов
Дата и время	Функции для анализа и работы со значениями даты и времени в формулах
Математические	Содержит арифметические и тригонометрические функции, позволяющие производить простые и сложные математические вычисления
Статистические	Функции для выполнения статистического анализа диапазонов данных
Ссылки и массивы	Функции для осуществления поиска в списках или таблицах, нахождения ссылок к ячейке
Работа с базой данных	Функции для работы со списками
Текстовые	Функции для выполнения действий над строками текста
Логические	Функции для проверки выполнения одного или нескольких условий
Проверка свойств и значений	Функции для проверки свойств и значений данных, вводимых в ячейки
Инженерные	Функции для выполнения инженерного анализа. Их можно разделить на три группы: – функции для работы с комплексными числами; – функции для преобразования чисел из одной системы счисления в другую; – функции для преобразования величин из одной системы мер и весов в другую
Информационные	Функции для определения типа данных, хранимых в ячейке. Проверяют выполнение какого-то условия и возвращают в зависимости от результата значение ИСТИНА или ЛОЖЬ
Аналитические	Функции для выполнения статистического анализа многомерных данных
Совместимость	Функции для совместимости с Excel2007 и более ранних версий

Доступ к функциям производится с помощью **Мастера функций**, кнопка вызова которого  находится на панели пиктографического меню **Формулы**.

Следует заметить, что пользователь может разрабатывать и собственные функции с помощью языка Visual Basic For Applications (VBA). В этом случае формируется категория «Определенные пользователем».

Десять функций, с которыми наиболее часто работает пользователь, собраны в группу «10 недавно использовавшихся». Кроме того, все функции Excel представлены в полном алфавитном перечне **Мастера функций**.

5.1 Математические функции Excel

Наиболее распространенными и часто используемыми являются функции категории «Математические». Полную информацию обо всех функциях этой категории можно получить в справочной системе MS Excel.

Рассмотрим те функции, которые при расчетах используются наиболее часто.

– **СУММ(число1; число2; . . . ; число n)** – вычисляет сумму аргументов. Здесь *Число1*, *число2*, ... , *число n* – от 1 до 30 аргументов, для которых требуется определить итог или сумму.

Например, функция **=СУММ(B5:B10)** означает, что нужно сложить содержимое ячеек с B5 до B10, а функция **=СУММ(B5:B10; A11)** означает, что складывается содержимое ячеек с B5 до B10 с содержимым ячейки A11.

Поскольку функцию СУММ применяют чаще других функций, для ее вызова на системном пиктографическом меню (СПИ) **Формулы** предусмотрена

специальная кнопка **Автосуммирование** .

– **СУММКВ(число1; число2; . . . ; число n)** – вычисляет сумму квадратов аргументов.

– **ПРОИЗВЕД(число1; число2; . . . ; число n)** – возвращает произведение аргументов.

– **СТЕПЕНЬ(число; степень)** – возвращает результат возведения аргумента число в указанную степень.

– **КОРЕНЬ(число)** – возвращает значение квадратного корня из аргумента число.

– **СУММЕСЛИ(диапазон; критерий; диапазон_суммирования)** – суммирует ячейки, заданные указанным условием. Здесь:

диапазон – диапазон анализируемых ячеек;

критерий – критерий в форме числа, выражения или текста, определяющий суммируемые ячейки;

диапазон_суммирования – фактические ячейки для суммирования.

– **ОКРУГЛ(число; число разрядов)** – округляет число до указанного количества десятичных разрядов.

ОКРУГЛ(82,736; 0) вернет результат 83 (округление до целых).
ОКРУГЛ(82,736; 1) вернет результат 82,7 (округление до десятых).
ОКРУГЛ(82,736; 2) вернет результат 82,74 (округление до сотых).

Функции **ОКРУГЛВЕРХ(число; число разрядов)** и

– **ОКРУГЛВНИЗ(число; число разрядов)** подобны функции ОКРУГЛ за исключением того, что округление производится либо с избытком (ОКРУГЛВЕРХ) либо с недостатком (ОКРУГЛВНИЗ).

Например,

ОКРУГЛВЕРХ(82,736; -1) вернет результат 90 (округление до десятков с избытком).

ОКРУГЛВНИЗ(82,736; -1) вернет результат 80 (округление до десятков с недостатком).

Кроме того, можно выполнить округление с заданной точностью с помощью функций.

– **ОКРВВЕРХ(число; точность)** и **ОКРВНИЗ(число; точность)**. Эти функции выполняют округление с избытком (ОКРВВЕРХ) или недостатком (ОКРВНИЗ) до ближайшего целого, кратного заданной точности.

Например, если в значениях цен необходимо избежать рублей, а товар стоит 829 рубля, функция =ОКРВВЕРХ(829;10) округлит цену до 830 руб., а функция =ОКРВНИЗ(829;10) округлит цену до 829 руб. И в том и в другом случае точность округления равна 10 руб.

– **НЕЧЕТ(число)** и **ЧЕТН(число)** – округляют аргумент число до ближайшего нечетного или четного значения.

– **СУММПРОИЗВ(массив1; массив2; ...; массив n)**. Здесь *Массив1; массив2; ...; массив n* – от 2 до 30 массивов, чьи компоненты нужно перемножить, а затем сложить.

– **СУММКВРАЗН(массив_х; массив_у)**. Здесь *массив_х* – первый массив или интервал значений, *массив_у* – второй массив или интервал значений.

– **СУММРАЗНКВ(массив_х; массив_у)** – возвращает сумму разностей квадратов соответствующих значений в двух массивах.

– **СУММСУММКВ(массив_х; массив_у)** – возвращает сумму квадратов соответствующих элементов двух массивов.

– **МОБР(массив)** – возвращает матрицу, обратную данной.

– **МОПРЕД(массив)** – возвращает определитель матрицы.

– **МУМНОЖ(массив1; массив2)** – возвращает матричное произведение двух массивов.

Обращение к функциям **СУММПРОИЗВ**, **СУММКВРАЗН**, **СУММРАЗНКВ**, **СУММСУММКВ** принципиально аналогично. В качестве аргументов используются массивы данных, которые вводятся либо диапазоном ячеек, либо перечислением ячеек. Различны лишь формулы вычисления результата, заложенные в каждой из функций.

5.2 Логические функции EXCEL

Достаточно часто при решении производственных, экономических и вообще любых реальных задач возникает необходимость анализа данных. Любой анализ предполагает использование логических функций.

Функций категории «Логические» всего семь: ЕСЛИ, ЕСЛИОШИБКА, И, ИЛИ, НЕ, ИСТИНА, ЛОЖЬ. Они достаточно понятны и просты в использовании.

– **ЕСЛИ(логическое_выражение;значение_если_истина;значение_если_ложь)** – используется для проверки значений выражений или формул и организации переходов в зависимости от результатов этой проверки.

Возвращает аргумент *значение_если_истина*, если *логическое выражение* при вычислении дает значение ИСТИНА, и аргумент *значение_если_ложь*, если аргумент *логическое выражение* при вычислении дает значение ЛОЖЬ. В качестве возвращаемых значений могут использоваться числовые значения, ссылки на ячейки, формулы или текст.

Пример. Предположим, что специалисту присваивается категория только в том случае, если он набрал более 100 баллов при сдаче квалификационного экзамена, в противном случае категория не присваивается. Занесем значение результата квалификационного экзамена в ячейку В4 и реализуем это условие функцией ЕСЛИ: =ЕСЛИ(В4<100;"--";"I категория").

Если в ячейке В4 содержится например, число 35, что меньше 50, результатом функции ЕСЛИ будет знак "--". Если в ячейке В4 содержится например, число 105, что больше 100, результатом функции ЕСЛИ будет текст "I категория".

При организации сложных вычислений до 3 функций ЕСЛИ могут быть вложены друг в друга в качестве значений аргументов.

– **И(логическое_значение1;логическое_значение2;....;логич_значение n);**

– **ИЛИ(логическое_значение1;логическое_значение2;;логич_значение n).**

Функции И и ИЛИ могут содержать до 30 проверяемых условий. Чаще всего эти функции используются в других условных функциях, например, в функции ЕСЛИ, для организации сложных условий. Аргументы должны быть логическими значениями, массивами или ссылками, которые содержат логические значения.

Расширим шкалу присвоения категории специалистам:

=ЕСЛИ(В4<50;"--";ЕСЛИ(И(В4>=50;В4<=100); "I категория"; "Высшая категория"))).

Если ячейка В4 содержит число 75, то в качестве результата будет выведен текст «I категория», если в ячейке В4 содержится число 150, то результатом является текст «Высшая категория», если же в ячейке В4 содержится число меньше 50, то результатом будет знак "--".

– **НЕ(логическое_значение).** Эта функция меняет на противоположное

логическое значение своего аргумента и используется в тех случаях, когда необходимо быть уверенным в том, что значение не равно некоторой конкретной величине.

НЕ(8>2) возвратит значение ЛОЖЬ.

НЕ(8<2) возвратит значение ИСТИНА.

– **ИСТИНА** – возвращает логическое значение ИСТИНА.

– **ЛОЖЬ** – возвращает логическое значение ЛОЖЬ.

– **ЕСЛИОШИБКА** – возвращает указанное значение, если вычисление по формуле вызывает ошибку; в противном случае функция возвращает результат формулы. Функция ЕСЛИОШИБКА позволяет перехватывать и обрабатывать ошибки в формулах (Формула. Совокупность значений, ссылок на другие ячейки, именованных объектов, функций и операторов, позволяющая получить новое значение. Формула всегда начинается со знака равенства (=)).

5.3 Статистические функции EXCEL

К категории статистических функций отнесено огромное количество функций – более 80, позволяющих выполнять разнообразные расчеты. Многие из этих функций довольно специализированные, но некоторые из них полезны и для тех, кто мало знаком со статистикой.

– **СРЗНАЧ(зн1, зн2, ... , знN)** – возвращает среднее арифметическое значение диапазона ячеек. Если в диапазоне находятся пустые ячейки или ячейки, содержащие текст, то они игнорируются. Можно использовать любое число аргументов.

– **СРЗНАЧА(значение1, значение2, ...)** – вычисляет среднее арифметическое значений аргументов, которые, помимо чисел, могут быть текстом или логическими значениями, такими как ИСТИНА и ЛОЖЬ. Массивы и ссылки, содержащие текст, а также пустой текст, интерпретируются как 0 (ноль). Аргументы, содержащие значение ИСТИНА, интерпретируются как 1, аргументы, содержащие значение ЛОЖЬ, интерпретируются как 0 (ноль).

Пример.

В ячейках рабочего листа ТП MS Excel содержатся следующие данные:

B2:B5 → {428; 356; 280; 250} – сдельный тарифный заработок, тыс. руб.

C2:C5 → {165; 160; 170; --} – количество отработанных часов.

Найти средний тарифный заработок по бригаде и среднее количество часов, отработанных каждым членом бригады.

Среднее значение сдельного тарифного заработка:

=СРЗНАЧ(B2:B5) ⇒ 325,5.

Так как столбец «Количество отработанных часов» содержит текстовое значение «--», то для вычисления среднего значения следует воспользоваться функцией СРЗНАЧА:

=СРЗНАЧА(C2:C5) ⇒ 123,75.

– **СРГЕОМ(число1;число2; ...)** – возвращает среднее геометрическое значений массива или интервала положительных чисел.

– **СЧЕТЕСЛИ (интервал, критерий)** – количество удовлетворяющих заданному критерию ячеек внутри интервала. Критерий может задаваться в виде числа, выражения или текста.

Например:

СЧЕТЕСЛИ(A1:A5, «>100») – возвратит число ячеек, содержащее которых будет более 100.

– **СЧЕТ(зн1, зн2, ... , знN)** – возвращает количество чисел в списке аргументов. Здесь

зн. (*значение*)1, *зн.*2, ... – это от 1 до 30 аргументов, которые могут содержать или ссылаться на данные различных типов, но в подсчете участвуют только числа, даты, или тексты, изображающие числа. Аргументы, которые являются значениями ошибки или текстами, которые нельзя интерпретировать как числа, игнорируются.

Пример:

СЧЁТ (A1:A7) возвращает 3

СЧЁТ (A4:A7) возвращает 2

СЧЁТ (A1:A7, 2) возвращает 4

	A
1	Продажи
2	21.08.21
3	
4	19
5	14.04
6	ИСТИНА
7	#ДЕЛ!/0

– **СЧЕТЗ(зн1, зн2, ... , знN)** – подсчитывает количество непустых значений (с данными любого типа) в списке аргументов.

Пример:

СЧЕТЗ (A1:A7) возвращает 6

СЧЕТЗ (A4:A7) возвращает 4

СЧЕТЗ (A1:A7, 2) возвращает 7

	A
1	Продажи
2	21.08.21
3	
4	19
5	14.04
6	ИСТИНА
7	#ДЕЛ!/0

– **МАКС(зн1, зн2, ... , знN)** и **МИН(зн1, зн2, ... , знN)** – используют для определения наибольшего и наименьшего значений в диапазоне. Можно задавать аргументы, которые являются числами, пустыми ячейками, логическими значениями или текстовыми представлениями чисел. Аргументы, которые являются значениями ошибки или текстами, не преобразуемыми в числа, вызывают значения ошибок.

– **МАКСА(зн1; зн2,...)** и **МИНА(зн1;зн2,...)** – возвращает наибольшее и наименьшее значение в списке аргументов, причем, наряду с числовыми значениями, выполняется также сравнение текстовых и логических значений. Пустые ячейки, логические значения или тексты в массиве или ссылке не игнорируются.

– **НАИБОЛЬШИЙ(массив;k)** и **НАИМЕНЬШИЙ(массив;k)** – возвращает k-ое наибольшее значение из множества данных. Эта функция используется, чтобы выбрать значение по его относительному местоположению. Например, функцию **НАИБОЛЬШИЙ** можно использовать, чтобы определить наилучший, второй или третий результат в баллах, показанный при тестировании.

5.4 Функции категории «Ссылки и массивы»

– **ПРОСМОТР(искомое_значение;просматриваемый_вектор;вектор_результатов)** – просматривает диапазон, в который входят значения только одной строки или одного столбца (так называемый вектор), в поисках определенного значения и возвращает значения из той же позиции второго диапазона. Здесь

Искомое_значение – это значение, которое **ПРОСМОТР** ищет в первом векторе. *Искомое_значение* может быть числом, текстом, логическим значением, именем или ссылкой, ссылающимися на значение.

Просматриваемый_вектор – это интервал, содержащий только одну строку или один столбец. Значения в аргументе *просматриваемый_вектор* могут быть текстами, числами или логическими значениями.

Следует иметь в виду, что значения в аргументе *просматриваемый_вектор* должны быть расположены в порядке возрастания: ..., -2, -1, 0, 1, 2, ..., A-Z, ЛОЖЬ, ИСТИНА. В противном случае функция **ПРОСМОТР** может вернуть неверный результат. Тексты в нижнем и верхнем регистре считаются эквивалентными.

Вектор_результатов – это интервал, содержащий только одну строку или один столбец. Он должен быть того же размера, что и *просматриваемый_вектор*.

Если **ПРОСМОТР** не может найти *искомое_значение*, то подходящим считается наибольшее значение в аргументе *просматриваемый_вектор*, которое меньше, чем *искомое_значение*.

Если *искомое_значение* меньше, чем наименьшее значение в аргументе *просматриваемый_вектор*, то функция **ПРОСМОТР** возвращает значение ошибки #Н/Д.

– **ВПР(искомое_значение;инфо_таблица;номер_столбца;интервальный_просмотр)** – ищет значение в крайнем левом столбце таблицы и возвращает значение в той же строке из указанного столбца таблицы.

Искомое значение – это значение, которое должно быть найдено в первом столбце массива. *Искомое значение* может быть значением, ссылкой или текстовой строкой.

Инфо_таблица – это таблица с информацией, в которой ищутся данные. Можно использовать ссылку на интервал или имя интервала, например, База_Данных или Список.

Интервальный просмотр – одно из двух значений: ИСТИНА или ЛОЖЬ. Если *интервальный просмотр* имеет значение ИСТИНА, то значения в первом столбце аргумента *инфо_таблица* должны быть расположены в возрастающем порядке: ..., -2, -1, 0, 1, 2, ..., A-Z, ЛОЖЬ, ИСТИНА; в противном случае функция ВПР может выдать неправильный результат. Если *интервальный просмотр* имеет значение ЛОЖЬ, то *инфо_таблица* не обязана быть сортированной.

Если ВПР не может найти *искомое значение* и *интервальный просмотр* имеет значение ИСТИНА, то используется **наибольшее** значение, которое меньше, чем *искомое значение*. Если *искомое значение* меньше, чем наименьшее значение в первом столбце аргумента *инфо_таблица*, то функция ВПР возвращает значение ошибки #Н/Д.

Номер столбца – это **номер** столбца в массиве *инфо_таблица*, в котором должно быть найдено соответствующее значение. Если *номер столбца* равен 1, то возвращается значение из первого столбца аргумента *инфо_таблица*; если *номер столбца* равен 2, то возвращается значение из второго столбца аргумента *инфо_таблица* и так далее.

Пример. Поиск данных в таблице.

Таблица для выполнения поиска расположена в ячейках D2:F9 (рис. 6.2). Рабочая таблица A2:B6 расположена таким образом, чтобы пользователь мог ввести номер участника в ячейку B2 и в ячейках B4 и B5 получить необходимую информацию из исходной таблицы.

	A	B	C	D	E	F
2	Номер участника	205		НОМЕР	ИМЯ	ВЗНОС
3				145	Михаил	124,5
4				156	Петр	254,3
5	Имя	Федор		187	Павел	45,8
6	Взнос	589,6		205	Федор	589,6
7				225	Александр	258,3
8				319	Тимофей	25,87
9				377	Евгений	145

Рисунок 6.2 – Поиск данных в таблице

Формулы для выполнения поиска имеют следующий вид:

В ячейке B5:

=ВПР(\$B\$2;\$D\$2:\$F\$9;2) или =ПРОСМОТР(B2;\$D\$3:\$D\$9;\$E\$3:\$E\$9).

В ячейке B6:

=ВПР(\$B\$2;\$D\$2:\$F\$9;3) или =ПРОСМОТР(B2;\$D\$3:\$D\$9;\$F\$3:\$F\$9).

Формула в ячейке B5 просматривает первый столбец таблицы D2:F9 в поисках величины, введенной в ячейку B2 (это значение 205). Она выбирает соответствующее ему (205) значение из столбца, номер которого задан в формуле (это столбец 2), и возвращает полученное значение (имя Федор) в ячейку B5. Аналогично работает формула в ячейке B6.

Если введенное значение не найдено в таблице, формула возвратит #Н/Д. С помощью функции ЕНД можно изменить формулу таким образом, чтобы сообщение было более понятным:

=ЕСЛИ(ЕНД(ВПР(\$B\$2;\$D\$2:\$F\$9;2;ЛОЖЬ));"Не найдено";ВПР(\$B\$2;\$D\$2:\$F\$9;2;ЛОЖЬ)).

Функция **ГПР** работает точно так же, как и **ВПР**, с той лишь разницей, что просматривает значения ячеек первой строки таблицы (выполняет поиск по горизонтали).

– **ВЫБОР(номер_индекса; значение1; значение2;...)** – использует номер_индекса, чтобы выбрать и вернуть значение из списка аргументов-значений. Функция **ВЫБОР** используется, чтобы выбрать одно значение из списка, в котором может быть до 29 значений.

Номер_индекса – это номер выбираемого аргумента-значения. *Номер_индекса* должен быть числом от 1 до 29, формулой или ссылкой на ячейку, содержащую число в диапазоне от 1 до 29. Например, если значения от *значение1* до *значение7* – это дни недели, то функция **ВЫБОР** возвращает один из дней при использовании числа от 1 до 7 в качестве аргумента *номер_индекса*.

Если *номер_индекса* равен 1, то функция **ВЫБОР** возвращает *значение1*; если он равен 2, то функция **ВЫБОР** возвращает *значение2* и так далее.

Если *номер_индекса* меньше 1 или больше, чем номер последнего значения в списке, то функция **ВЫБОР** возвращает значение ошибки #ЗНАЧ!.

Если *номер_индекса* является дробным, то он усекается до меньшего целого.

Значение1, значение2 ... – это от 1 до 29 аргументов-значений, из которых **ВЫБОР**, используя *номер_индекса*, выбирает значение или выполняемое действие. Аргументы могут быть числами, ссылками на ячейки, именами, формулами, функциями или текстами.

Примеры.

– **ВЫБОР(2;"1-ый";"2-ой";"3-ий";"Последний") = 2-ой.**

– **СУММ(A1:ВЫБОР(3;A10;A20;A30)) = СУММ(A1:A30)**

Если ячейка A10 содержит 4, то:

– **ВЫБОР(A10;"Гвозди";"Винты";"Гайки";"Болты") = Болты.**

Если A10-3 равняется 3, то:

– **ВЫБОР(A10-3;"1-ый";"2-ой";"3-ий";"Последний") = 3-ий.**

Пример. Рассмотрим комплексный пример, представляющий собой расчет размеров комиссионных вознаграждений менеджеров по продажам. Исходные данные для расчета комиссионных приведены в таблице ниже:

Объем продаж за месяц (у.е.)	Комиссионные (%)
0–9999	8
10000–19999	10,5
20000–39999	12
40000 и более	14

Выполним два варианта расчета комиссионных:

Вариант 1. С использованием функции **ВПР** (см. рис. 6.3).

3	A	B	C	D	E	F	G	H	I
4	Объем продаж	Комиссионные, %		ФИО	Стаж	Объем продаж	Комиссионные, %	Комиссионные, ден.ед.	С учетом стажа
5	0	8		Тихонов А.	5	5000	8	400	420
6	10000	10,5		Иванов С.	12	7000	8	560	627,2
7	20000	12		Крутой И.	10	11000	10,5	1155	1270,5
8	40000	14		Увалов Н.	1	15000	10,5	1575	1590,75
9				Пронин В.	8	21500	12	2580	2786,4
10				Кошкин И.	4	25000	12	3000	3120
11				Фунтик А.	6	12000	10,5	1260	1335,6
12				Зайцев С.	7	15000	10,5	1575	1685,25
13				Воронов В.	5	39900	12	4788	5027,4
14				Баранов Н.	4	45000	14	6300	6552

Рисунок 6.3 – Расчет комиссионных с использованием функции ВПР

– Для определения величины комиссионных вознаграждений (в %) в столбце G5:G14 воспользуемся формулой, которую внесем в ячейку G5: =ВПР(F5;\$A\$4:\$B\$8;2).

– Скопируем эту формулу в ячейки G6:G14.

– Для определения суммы комиссионных (в ден. ед.) воспользуемся формулой, которую внесем в ячейку H5: =F5*G5/100.

– Скопируем эту формулу в ячейки H6:H14.

– Для определения суммы комиссионных с учетом стажа воспользуемся формулой, которую внесем в ячейку I5: =H5+H5*E5/100.

– Скопируем эту формулу в ячейки I6:I14.

Вариант 2. С использованием функции **ЕСЛИ** (рис. 6.4):

	К	Л	М	Н	О
	ФИО	Стаж	Объем продаж	Комиссионные, ден.ед.	С учетом стажа
4					
5	Тихонов А.	5	5000	400	420
6	Иванов С.	12	7000	560	627,2
7	Крутой И.	10	11000	1155	1270,5
8	Увалов Н.	1	15000	1575	1590,75
9	Пронин В.	8	21500	2580	2786,4
10	Кошкин И.	4	25000	3000	3120
11	Фунтик А.	6	12000	1260	1335,6
12	Зайцев С.	7	15000	1575	1685,25
13	Воронов В.	5	39900	4788	5027,4
14	Баранов Н.	4	45000	6300	6552

Рисунок 6.4 – Расчет комиссионных с использованием функции ЕСЛИ

- В ячейку N5 внесем формулу:
`=ЕСЛИ(И(М5>=0;М5<=9999,99);М5*0,08;
ЕСЛИ(М5<=19999,99;М5*0,105;ЕСЛИ(М5<=39999,99;М5*0,12;М5*0,14)))`;
- Скопируем эту формулу в ячейки N6:N14;
- В ячейку O5 внесем формулу: `=N5+N5*L5/100`;
- Скопируем эту формулу в ячейки O6:O14.

5.5 Функции категории «Текстовые»

- **СЦЕПИТЬ (текст1;текст2;...)** – объединяет несколько текстовых строк в одну. Здесь

Текст1, текст2, ... – это от 1 до 30 элементов текста, объединяемых в один элемент текста. Элементами текста могут быть текстовые строки, числа или ссылки, которые ссылаются на одну ячейку. Вместо функции СЦЕПИТЬ для объединения текстов можно использовать оператор "&".

- **ЗАМЕНИТЬ(старый_текст;нач_ном;число_символов;нов_текст)** – замещает указанную часть символов текстовой строки другой строкой текста. Здесь

Старый_текст – текст, в котором желательно заменить некоторые символы.

Нач_ном – это позиция символа в тексте старый_текст, начиная с которой символы заменяются текстом нов_текст.

Число_символов – это число символов в тексте старый_текст, которые заменяются текстом новый_текст.

Новый_текст – это текст, который заменяет символы в тексте старый_текст.

Пример

– ЗАМЕНИТЬ(«абвгдежзийк»;6,5; «*») = **абвгд*к** – заменяет текстом **новый_текст** пять символов начиная с шестого в тексте **старый_текст**. Символы с шестого по десятый заменены на «*».

– ЗАМЕНИТЬ(«2007»;3,2; «08») = 2003 – заменяет две последние цифры в тексте 2002 на 03.

Кроме рассмотренных функций при обработке текстовых значений интерес представляют и следующие:

– **СИМВОЛ** – возвращает знак с заданным кодом;
– **РУБЛЬ** – преобразует число в текст, используя денежный формат доллара;

– **СОВПАД** – проверяет идентичность двух текстов;
– **НАЙТИ** – ищет вхождение одного текста в другой (с учетом регистра);

– **ФИКСИРОВАННЫЙ** – форматирует число и преобразует его в текст с заданным числом десятичных знаков;

– **ДЛСТР** – возвращает количество знаков в текстовой строке;

– **СТРОЧН** – делает все буквы в тексте строчными;

– **ПРОПНАЧ** – делает прописной первую букву в каждом слове текста;

– **ПОИСК** – ищет вхождение одного текста в другой (без учета регистра);

– **ПОДСТАВИТЬ** – заменяет в текстовой строке старый текст новым.

– **Т** – преобразует аргумент в текст;

– **ТЕКСТ** – форматирует число и преобразует его в текст;

– **СЖПРОБЕЛЫ** – удаляет из текста пробелы;

– **ПРОПИСН** – делает все буквы в тексте прописными.

5.6 Функции категории «ДАТА И ВРЕМЯ»

Рассмотрим некоторые функции этой категории, которые представляют интерес для бухгалтерских и финансовых расчетов.

– **ГОД(дата_в_числовом_формате)** – возвращает год, соответствующий аргументу *дата_в_числовом_формате*. Год определяется как целое в интервале 1900-9999. Здесь *Дата_в_числовом_формате* – это дата, год которой необходимо найти. Даты должны вводиться с использованием функции ДАТА или как результат вычисления других формул и функций. Например, для 23-го мая 2008 года следует использовать ДАТА(2008;5;23);

– **ДАТА(год;месяц;день)** – возвращает целое число, представляющее определенную дату. Если до ввода этой функции форматом ячейки был Общий, результат будет отформатирован как дата;

– **ДЕНЬ(дата_в_числовом_формате)** – возвращает день, соответствующий аргументу *дата_в_числовом_формате*;

- **ДАТАЗНАЧ(дата_как_текст)** – возвращает числовой формат даты, представленной в виде текста. Функция ДАТАЗНАЧ используется для преобразования даты из текстового представления в числовой формат;
- **ДЕНЬНЕД** – преобразует дату в числовом формате в день недели;
- **ДНЕЙ360** – вычисляет количество дней между двумя датами на основе 360-дневного года;
- **МЕСЯЦ** – преобразует дату в числовом формате в месяцы;
- **РАБДЕНЬ(нач_дата;количество_дней;праздники)** – возвращает число, представляющее дату, отстоящую на заданное количество рабочих дней вперед или назад от даты *нач_дата*. Рабочими днями не считаются выходные дни и дни, определенные как праздничные. Функция РАБДЕНЬ используется, чтобы исключить выходные дни или праздники при вычислении дат платежей, ожидаемых дат доставки или количества фактически отработанных дней;
- **СЕГОДНЯ()** – возвращает текущую дату в числовом формате. Если до ввода этой функции форматом ячейки был Общий, результат будет отформатирован как дата;
- **ТДАТА** – возвращает текущую дату и время в числовом формате;
- **ЧИСТРАБДНИ(нач_дата;кон_дата;праздники)** – возвращает количество рабочих дней между *нач_дата* и *кон_дата*. Праздники и выходные в это число не включаются. Функцию ЧИСТРАБДНИ можно использовать для вычисления оплаты работника на основе количества дней, отработанных в указанный период. Даты должны вводиться с использованием функции ДАТА или как результат других формул и функций.

6 Средства деловой графики

Данные из рабочих листов Excel можно представить в виде разнообразных диаграмм, которые объединены общим понятием «деловая графика».

Диаграммы – это графическое представление взаимосвязей между рядами данных для облегчения их сравнения.

Ряд данных представляет собой группу связанных точек диаграммы, построенной на основе значений одной строки или столбца таблицы. На диаграмме в зависимости от ее типа может быть отображен один или несколько рядов данных.

ТП MS Excel предлагает стандартные и нестандартные двумерные и трехмерные диаграммы различных типов. Каждый тип стандартных диаграмм имеет несколько разновидностей.

Диаграммы можно строить как на рабочих листах с данными, так и на отдельных листах рабочей книги в любом месте. Диаграммы можно совмещать, накладывая друг на друга. При построении диаграммы следует выбирать такой ее тип, который наиболее полно и понятно отобразит полученные результаты.

Деловая графика Excel представлена следующими типами стандартных диаграмм.

– **Гистограмма** позволяет представить изменение данных на протяжении отрезка времени.

– **Линейчатая диаграмма** позволяет сравнивать отдельные значения.

– **График** отражает тенденции изменения данных за равные промежутки времени.

– **Круговая диаграмма** показывает как абсолютную величину каждого элемента ряда данных, так и его вклад в общую сумму. На ней может быть представлен только один ряд данных.

– **Кольцевая диаграмма** показывает вклад каждого элемента в общую сумму, но, в отличие от круговой диаграммы, она может содержать несколько рядов данных. Каждое кольцо в кольцевой диаграмме представляет отдельный ряд данных.

– **Биржевая диаграмма** отображает наборы данных из нескольких значений курсов акций и используется для демонстрации цен на них.

– **Точечная диаграмма** отображает взаимосвязь между числовыми значениями в нескольких рядах или представляет две группы чисел в виде одного ряда точек в координатах XY. Она обычно используется для представления данных научного характера.

– **Диаграмма с областями** подчеркивает величину изменения в течение определенного периода времени, показывая сумму введенных значений, а также отображает вклад отдельных значений в общую сумму.

– **Пузырьковая диаграмма** является разновидностью точечной диаграммы. Размер маркера данных указывает значение третьей переменной.

– **Лепестковая диаграмма** позволяет сравнивать совокупные значения нескольких рядов данных. В ней все категории имеют собственные оси координат, расходящиеся лучами из начала координат. Линиями соединяются значения, относящиеся к одному ряду.

– **Поверхность** используется для поиска наилучшего сочетания в двух наборах данных. Здесь области, относящиеся к одному диапазону значений, выделяются одинаковым цветом или узором.

Для создания и форматирования диаграммы Excel предлагает специальную программу **Мастер диаграмм**, которая за несколько шагов позволяет построить диаграмму любого типа. Для вызова мастера можно воспользоваться командой меню **Вставка**. Для построения диаграммы вначале следует выделить нужные значения, причем выделение данных, расположенных в несмежных диапазонах ячеек, производится при нажатой клавише <CTRL>. Затем выбирается тип диаграммы в соответствии с решаемой задачей (рекомендуемый тип – гистограмма). При необходимости диаграмму можно дополнительно оформить: ввести заголовок диаграммы, указать подписи данных, варианты расположения осей, вывод значений и т.д.

Как видно, интерфейс ТП MS Excel достаточно прост и дружелюбен и предоставляет пользователю широкий набор элементов управления (панели, команды, пиктограммы, простой доступ к справочной информации через специальное меню **Помощь**), значительно облегчающих работу.

Лекция 7. КОМПЬЮТЕРНАЯ ГРАФИКА

План лекции

1. Виды компьютерной графики.
 - 1.1. Растровая графика.
 - 1.2. Векторная графика.
 - 1.3. Фрактальная графика.
2. Цветовые модели.
3. Презентация и ее структура.

1 Виды компьютерной графики

Несмотря на то, что для работы с компьютерной графикой существует множество классов программного обеспечения, различают всего три вида компьютерной графики. Это растровая графика, векторная графика и фрактальная графика. Они отличаются принципами формирования изображения при отображении на экране монитора или при печати на бумаге.

1.1 Растровая компьютерная графика

Поскольку персональный компьютер – это вычислительная машина, обрабатывающая цифры, всякое изображение, прежде чем оно станет доступно на компьютере, должно быть описано математически, или «оцифровано». Существует два возможных способа математического описания: аналитическое, с помощью математических формул, и описание массивом (набором) цифр.

Растровая, или **битовая**, форма компьютерной графики своим происхождением обязана второму способу оцифровки изображений. Если коротко, суть его заключается в следующем. Представьте себе, что у вас имеется какая-либо фотография. С помощью линейки и карандаша нанесите на ее поверхность горизонтальные и вертикальные линии, расположенные друг от друга на одинаковом расстоянии. Получится сетка, состоящая из квадратных ячеек, каждый квадрат содержит какой-либо элемент исходного изображения. Цвет в пределах любой из таких ячеек будет меняться. Давайте выберем для каждой ячейки некое постоянное значение цветового оттенка, например, методом простого усреднения. Если теперь пронумеровать наши ячейки от первой до последней, мы получим набор пар цифр – первая представляет собой номер квадрата, вторая описывает усредненный оттенок цвета. Именно такой метод лежит в основе описания любого растрового изображения.

Сетка, создаваемая для реальной оцифровки, содержит огромное количество ячеек настолько малых, что глаз человека их не видит, воспринимая все изображение как целое. Сама сетка получила название растровой карты, а ее единичный элемент (квадратная ячейка) называется растром или пикселом (от английского *pixel* – *PICTure Single ELe ment*). Растровая карта представляет собой набор (массив) троек чисел: две координаты раstra на плоскости и его цвет.

В растровой графике изображение строится по точкам. Обычно в качестве исходного изображения используются отсканированные фотографии и рисунки, нарисованные художником на бумаге, или изображения, полученные цифровыми фото- и видеокамерами. Затем графическими редакторами, например: Photoshop, изображения корректируются, создаются специальные эффекты или производится монтаж нескольких изображений.

Количество точек на экране зависит от физических характеристик монитора, видеокарты и настроек операционной системы. В Windows обычно используются графические разрешения 640*480, 800*600, 1024*768 и др. Для подготовки Web-страницы не рекомендуется брать большое разрешение, так как на мониторах с небольшим разрешением изображение целиком не поместится на экране. А работать с горизонтальной и вертикальной прокруткой неудобно. При увеличении масштаба количество точек в изображении не изменяется и размер точки просто увеличивается. При большом приближении изображение становится зернистым – этот эффект называется пикселизацией.

Используется растровая графика в электронных (мультимедийных) и полиграфических изданиях. Недостатки растровой графики:

- большой размер файла, так для фотографии 10*15 с разрешением 300 dpi требуется 6 Мб;
- при увеличении изображения детализация невозможна и появляется эффект зернистости (пикселизация изображения).

1.2 Векторная компьютерная графика

Программные средства для работы с векторной графикой наоборот предназначены, в первую очередь, для создания иллюстраций и в меньшей степени для их обработки. Такие средства широко используют в рекламных агентствах, дизайнерских бюро, редакциях и издательствах. Оформительские работы, основанные на применении шрифтов и простейших геометрических элементов, решаются средствами векторной графики намного проще. Существуют примеры высокохудожественных произведений, созданных средствами векторной графики, но они скорее исключение, чем правило, поскольку художественная подготовка иллюстраций средствами векторной графики чрезвычайно сложна.

Основным логическим элементом графики в векторной форме является геометрический объект. В качестве объекта принимаются простые геометрические фигуры (так называемые примитивы – прямоугольник,

окружность, эллипс, линия), составные фигуры или фигуры, построенные из примитивов, цветовые заливки, в том числе градиенты.

В векторной графике изображение строится по линиям (прямым или кривым). Если для прямой «линии» в растровой графике хранится информация о каждой точке линии, то в векторной графике хранятся параметры уравнения линии, координаты начала x_1 и конца x_2 (координаты остальных точек вычисляются), а также свойство линии: форма линии, толщина, цвет, тип линии: сплошная, пунктирная и т.д. Замкнутые линии имеют свойство заполнения (цвет, градиент, текстура, карта). Линии имеют вершины, которые называются узлами. Узлы также имеют свойства.

В настоящее время лидерами среди программных пакетов обработки векторной графики можно назвать CorelDraw корпорации Corel; Adobe Illustrator; FreeHand. Все пакеты имеют версии для платформ PC и Macintosh, достаточно развиты по своим возможностям, чтобы считаться профессиональными инструментами, и имеют больше сходных черт, чем отличных.

Недостатком векторной графики является сложность получения изображения. Достоинство: небольшой размер файла.

1.3 Фрактальная компьютерная графика

Программные средства для работы с фрактальной графикой предназначены для автоматической генерации изображений путем математических расчетов. Создание фрактальной художественной композиции состоит не в рисовании или оформлении, а в программировании. Фрактальную графику редко применяют для создания печатных или электронных документов, но ее часто используют в развлекательных программах.

Разрешение изображения задаётся пользователем при создании файла в графическом редакторе или при сканировании. Измеряется в пикселях на дюйм и определяет физический размер изображения.

2 Цветовые модели

В природе большинство цветовых оттенков получаются смешением основных цветов. Способ разделения цветового оттенка на составляющие компоненты называется цветовой моделью.

Цветовая модель RGB. Цветовое изображение рассматривается в проходящем свете (мониторы, слайд-проекторы) и получается суммированием трёх основных цветов (Аддитивная модель):

- **R(red)** – красного,
- **G(green)** – зелёного,
- **B(blue)** – синего.

Максимальная яркость (FF) всех трёх основных цветов в сумме дают белый.

Цветовая модель СМΥΚ. Модель используется для подготовки печатных изображений, которые рассматриваются в отражённом свете. Чем больше краски положено на бумагу, тем больше света она поглощает и меньше отражает, следовательно – тем темнее цвет. В модели СМΥΚ используются цвета, которые получаются методом вычитания основного цвета из белого:

- **C(Cyan)** – голубой= белый – красный;
- **M(Magenta)** – пурпурный (фиолетовый)= белый – зелёный;
- **Y(Yellow)** – жёлтый= белый – синий;
- **K(black)** – чёрный.

Три цвета: **голубой, пурпурный и жёлтый** называют дополнительными, так как они дополняют основные цвета (красный, зелёный и синий) до белого. Так как максимальная яркость (количество краски) в сумме даёт грязно серый цвет (вместо чёрного – теоретически), а используется модель для печати, то в модель добавлен чистый чёрный цвет.

Цветовая палитра – это таблица данных с информацией о методе кодировки цвета.

Индексная палитра – соответствие конкретного оттенка номеру (индексу). В разных изображениях одному оттенку может быть присвоен разный индекс.

Безопасная палитра – фиксированные 256 цветов для всех типов компьютеров.

При всех преимуществах векторной графики необходимо заметить, что создание фотореалистичных изображений является очень трудоемким процессом и требует своеобразной техники.

3 Презентация и ее структура

Термин «презентация» имеет два значения. Рассмотрим их определения.

Презентация – документ, содержащий ряд кадров (слайдов). Презентация предназначена для иллюстрации (сопровождения) доклада, технического проекта, лекции, информационного сообщения, результатов научных исследований, бизнес-плана, дипломного проекта, диссертации и т.п. Презентация может содержать эффекты анимации, звуковые и видеоэффекты, которые усиливают эмоциональное воздействие на аудиторию (слушателей и зрителей), помогают сконцентрировать внимание на важных моментах сообщения. Первое определение можно кратко выразить так: **презентация** – это комплект иллюстративных материалов, предназначенных для сопровождения доклада. Порой презентацию (в этом значении термина) называют на английский манер – слайд-шоу (показ слайдов).

Второе значение термина «**презентация**» – это мероприятие (встреча, совещание, банкет), целью которого является представление (показ, демонстрация, ознакомление, публикация) чего-либо нового (книги, фильма, фирмы, продукции, изделия, товара, сайта и т.п.). Второе значение рассматриваемого термина можно лаконично определить так: **презентация** – это мероприятие, предназначенное для ознакомления с чем-то новым.

В дальнейшем термин «презентация» будет использоваться в обоих смыслах. Значение термина будет, как правило, понятным из контекста.

Презентация предполагает использование заранее подготовленного иллюстративного материала и технических средств для его демонстрации. К современным техническим средствам проведения презентаций относятся локальные и глобальные сети. Причем могут применяться разнообразные формы презентации: пассивная демонстрация готового материала через Интернет и активная совместная работа над презентацией в виде сетевой конференции нескольких участников.

Презентации могут использоваться для ознакомления людей (клиентов, покупателей, посетителей, зрителей, учащихся) с учебными заведениями, предприятиями, фирмами, продукцией, выпускаемой предприятиями, компьютерными программами, экспонатами музея, артистами театра, спортивными клубами, туристическими маршрутами и т.п. Презентации удобно использовать в учебных целях, например, для объяснения принципа действия какого-либо устройства или порядка работы с программой для ЭВМ. Презентации можно использовать и в быту, в частности, для создания фотоальбомов, которые посвящены проведенному отпуску или конкретному члену семьи (ребенку).

Важным элементом при рассмотрении вопросов презентации является понятие «слайд».

Слайд (термин происходит от английского слова Slide – диапозитив) – логически и художественно завершенный документ, который может содержать текст, рисунки, фотографии, таблицы, диаграммы, колонтитулы, гиперссылки, ярлыки, видеоклипы, аудиоклипы, примечания, дату и время, порядковый номер. Отдельные элементы, размещенные на слайде, называются **объектами**.

Слайд может существовать в электронном виде, тогда он демонстрируется с помощью компьютера, от которого сигналы поступают в локальную или глобальную сети, на мультимедийный проектор, плазменную или светодиодную панели, либо телевизионный приемник с большим экраном.

Электронный слайд можно скопировать на фотопленку шириной 35 мм (или 60 мм) и демонстрировать его с помощью проектора слайдов. Подобные проекторы используются в быту для просмотра диапозитивов, создаваемых с помощью фотоаппаратов.

Электронный слайд можно отпечатать с помощью принтера на прозрачной пленке. Затем пленку формата А4 (прозрачку), содержащую изображение слайда, можно проецировать на экран с помощью специальных проекционных аппаратов: видеоимиджера или оверхед-проектора.

Наконец, электронный документ может быть отпечатан на бумаге и использован как раздаточный материал для каждого участника конференции. Кроме того, существует возможность отпечатанных на бумаге слайдов демонстрировать на большом экране. Для этого служат телевизионные камеры и специальные проекторы, называемые эпидиаскопами, эпипроекторами и видеоимиджерами.

Для создания презентаций разработаны специальные программы, которые сочетают в себе элементы текстовых, графических и звуковых редакторов. В этих программах имеются средства для разработки и демонстрации совокупности логически связанных кадров (слайдов). Наиболее распространены такие программы, как MS PowerPoint, Harvard Graphics, Lotus Freelance Graphics, Charisma.

Программа **MS PowerPoint** имеет наибольшее распространение среди пользователей благодаря активной рекламной политике, проводимой фирмой Microsoft, и несомненным достоинствам этой программы. Данная программа входит в комплект программ MS Office, поэтому часто автоматически устанавливается пользователями на жесткий диск одновременно с текстовым фактором, электронными таблицами и системой управления базами данных. Совместимость всех программ MS Office позволяет легко переносить необходимые объекты по технологии OLE из других программ в MS PowerPoint.

В Microsoft PowerPoint имеется три основных режима работы: обычный режим, режим сортировщика слайдов и режим просмотра слайдов.

Обычный режим используется при создании и редактировании презентаций. В данном режиме на экране монитора по умолчанию появляются три области. Первая область располагается на экране слева и используется для редактирования структуры презентации. Эта область содержит две закладки: **Структура** и **Слайды**. С помощью закладки **Структура** легче анализировать логические связи между созданными слайдами, составлять план доклада. В этом режиме на экран выводятся только заголовки и подзаголовки слайдов. С помощью закладки **Слайды** можно видеть эскизы созданных слайдов и порядок их расположения в презентации. В этом режиме легко представить общую картину доклада.

Вторая область размещается на экране справа. В этом месте в крупном масштабе отображается текущий слайд. Третья область используется для хранения заметок к слайду. Она располагается на экране снизу. Заметки предназначены только для докладчика (лектора) и не видны во время демонстрации слайдов. Заметки – это как бы краткий конспект доклада (шпаргалка для лектора).

Режим **сортировщика** слайдов дает возможность видеть одновременно несколько десятков слайдов (число видимых на экране кадров зависит от выбранного масштаба), выполнить реорганизацию презентации (поменять местами слайды), добавить или удалить слайды, производить предварительный просмотр эффектов анимации и смены слайдов. Перемещение слайдов осуществляется по технологии Drag and Drop (буксировка). В этом режиме идет монтаж слайд-фильма.

В режиме **просмотра** слайдов презентацию можно посмотреть в окончательном виде на полном экране.

Программы для создания средств презентации (ПССП) позволяют: создавать новые презентации, редактировать существующие презентации,

демонстрировать готовые презентации.

Создать новую презентацию в MS PowerPoint можно на основе: рекомендаций мастера автосодержания, готовых шаблонов, пустых слайдов, уже существующей презентации.

Мастер **автосодержания** содержит несколько стандартных заготовок (структур). Перечислим заголовки некоторых заготовок: Общий доклад, Предлагаем стратегию, Сообщаем дурные Новости, Учебный курс, Мозговой штурм, Диплом, Бизнес-план и т.д. Для создания простейшей презентации начинающему пользователю достаточно лишь полно ответить на вопросы, поставленные мастером автосодержания.

Шаблон – документ (заготовка), позволяющий наглядно представить будущий слайд. Шаблон определяет стиль презентации, цветовую схему, размеры и координаты областей (рамки), в которых размещаются объекты.

Стиль – это параметры слайда, которые определяют максимальное число допустимых для использования шрифтов, размеры шрифтов заголовка и текста, количество пунктов в маркированном списке, внешний вид и цвет маркера в списке, количество строк в заголовке, форму написания заголовка и основного текста (например, все буквы прописные, все буквы строчные или как в обычном предложении). **Список** – это конструкция текста, содержащего перечисления каких-то элементов. Списки бывают маркированными и нумерованными.

Шаблон может быть применим ко всем слайдам презентации или лишь к выборочным из них.

Макет (рис. 7.1) – документ (заготовка), который схематично показывает расположение на слайде будущих объектов (заголовков, подзаголовков, рисунков, текста, диаграмм, таблиц, видеоклипов).

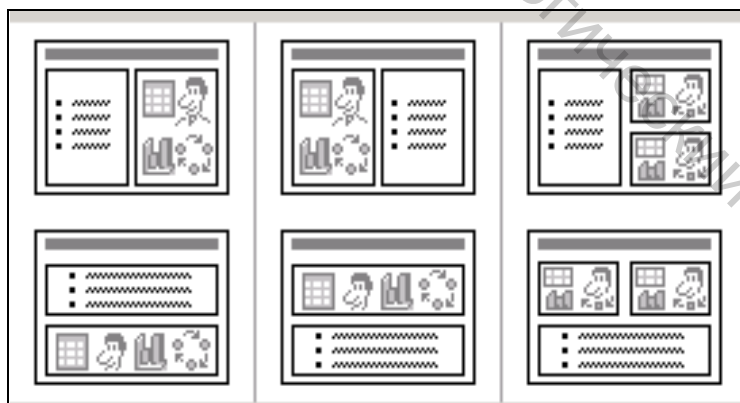


Рисунок 7.1 – Примеры макетов слайдов презентации

Заранее продуманный план презентации позволяет докладчику иметь в своем распоряжении несколько альтернативных вариантов доклада. Различные варианты проведения доклада можно выбирать с помощью гиперссылок, в зависимости от заданных вопросов и необходимости привлечения дополнительного материала.

Гиперссылки, размещенные на слайдах, помогают докладчику в нужный момент времени изменить линейный (последовательный) характер демонстрации слайдов, гибко перестроить структуру (план) доклада в зависимости от реакции аудитории и заданных вопросов. С помощью текстовых или графических объектов, исполняющих роль гиперссылок, докладчик (или лектор) может по своему усмотрению вызвать предыдущий, первый или последний слайды, либо сделать переход на заранее указанный слайд (выбор происходит по их порядковым номерам). Делается это стандартными приемами, принятыми в Web-дизайне: щелчком мышью по гиперссылке. Причем переход на другой кадр можно осуществлять либо щелчком по гиперссылке, либо наведением курсора на гиперссылку (наездом). Гиперссылкой может служить фрагмент текста или графический объект. Создавать графические гиперссылки в MS PowerPoint удобно с помощью управляющих кнопок, которые размещаются в пункте **Автофигуры** (панель **Рисование**).

При помощи гиперссылки можно вызвать файл, содержащий другую презентацию. После исполнения вызванной презентации программа вернется к демонстрации исходного слайда. Для привлечения дополнительной информации с помощью гиперссылки прямо из презентации можно соединиться с необходимым сервером в Интернет.

Получить произвольную последовательность демонстрации слайдов позволяет также опция **Произвольный показ** (находится в пункте **Показ слайдов**). С её помощью можно заранее подготовить несколько альтернативных вариантов показа презентации.

Заметим, что существует еще одна возможность изменения линейного (последовательного) характера демонстрации слайдов. Для этого во время демонстрации презентации нужно сделать щелчок правой кнопкой мыши по слайду и выбрать опцию **Переход**, а там – один из пунктов: **Навигатор слайдов** или **Выбор слайда по имени**. Указанными способами можно перейти к любому слайду презентации.

При редактировании презентаций можно устанавливать слайды в нужном порядке (сортировать), делать временно отдельные слайды невидимыми, изменять шаблоны, цветовые схемы, выбирать виды переходов между слайдами. Для облегчения процедуры разметки создаваемого слайда можно вывести на экран сетку. Созданная презентация может быть защищена паролем от несанкционированного использования другими лицами. При этом пользователь выбирает один из восьми методов шифрования и длину использованного ключа. На слайде могут быть размещены рисунки, которые входят в состав стандартной коллекции картинок, либо иллюстрации из собственных коллекций.

При демонстрации презентаций ПССП предоставляют в распоряжение докладчика разнообразные возможности: например, можно использовать указатель в виде электронной стрелки, с помощью которого удобно показывать интересные объекты на слайде. Можно использовать электронный карандаш, который позволяет выделять, подчеркивать ключевые места на слайде.

Программа MS PowerPoint способна изменять цвет электронного карандаша. Движением стрелки и карандаша управляют с помощью ручного манипулятора «мышь» или цифрового планшета.

Смена кадров во время демонстрации происходит по командам, которые подаются докладчиком (лектором), или в заранее установленном темпе (через определенные промежутки времени). В принципе презентация может демонстрироваться автоматически без участия докладчика. Это удобно делать во время проведения конференций, ярмарок-выставок или при установке компьютера в витрине магазина.

Создаваемые презентации являются мультимедийными продуктами за счет использования звуковых эффектов, видеоклипов и анимации.

Анимация – добавление специальных видеоэффектов, которые оживляют слайды. Например, можно создать эффект вращения заголовка или динамичного вытеснения одного слайда другим. Анимация придает слайдам подвижность, позволяет дополнительными средствами концентрировать внимание аудитории на важных моментах доклада, делает материал более запоминающимся.

Перечислим некоторые **эффекты анимации**, используемые в MS PowerPoint при демонстрации заголовков и текстов: подчеркивание, растворение, случайные полосы, центрифуга (вращающиеся слова), колесо (вращающиеся буквы), нейтрон (хаотичное движение букв), бумеранг (вылет заголовка из угла и последующее возвращение его обратно), смещение текста в виде титров, движение букв по траектории эллипса и др.

Переход от одного слайда к другому может сопровождаться разнообразными визуальными эффектами. При этом очередной слайд может проступать через предыдущий слайд различными способами. Опция MS PowerPoint **Смена слайдов** позволяет создать следующие эффекты: жалюзи горизонтальные и вертикальные (горизонтальные и вертикальные полосы), шашки (смена слайдов проступанием через мелкие квадраты), объединение по вертикали или горизонтали, наплывы в разные стороны, сдвиги, плавное угасание и т.п.

Здесь уместно заметить, что программы создания презентаций дают возможность создать эффектную, броскую форму подачи доклада, четче продумать структуру доклада, благодаря шаблонам и автосодержанию не забыть о ключевых моментах доклада. Однако содержание презентации определяется лишь профессиональной подготовкой автора разработки, его квалификацией. Естественно, что броская форма доклада не может заменить необходимого содержания. Причем чрезмерное увлечение мелькающими надписями и звуковыми эффектами может вызвать негативную реакцию аудитории.

Лекция 8. ПАКЕТЫ ДЛЯ МАТЕМАТИЧЕСКОЙ ОБРАБОТКИ ДАННЫХ

План лекции

1. Характеристика и функциональные возможности пакетов для математической обработки данных.
2. Технологии работы с системой компьютерной математики Maple.
 - 2.1. Общие сведения о СКМ Maple.
 - 2.2. Функции.
 - 2.3. Типовые средства графики.
 - 2.4. Решение уравнений.
 - 2.5. Решение систем линейных алгебраических уравнений.
 - 2.6. Вычисление интегралов и производных.
3. Использование СКМ Maple для решения задач экономического профиля.

1 Характеристика и функциональные возможности пакетов для математической обработки данных

Для автоматизации математических расчетов используются разнообразные вычислительные средства от программируемых микрокалькуляторов до сверхмощных супер ЭВМ. Тем не менее такие расчеты остаются сложным делом. Более того, применение компьютеров внесло новые свои трудности: прежде чем начать расчеты, пользователь должен освоить основы программирования на одном или нескольких языках программирования и численные методы расчетов. Положение стало меняться после появления специализированных программных комплексов для автоматизации математических и инженерно-технических расчетов.

Математическими системами, универсальными математическими пакетами (средами) называют пакеты прикладных программ, содержащие разнообразные инструменты для решения математических задач. Такие системы являются программными приложениями, поддерживаемыми средой Windows и ресурсами самого ПК, а также позволяют импортировать документы из других приложений в широком диапазоне их форматов.

К наиболее распространенным математическим пакетам можно отнести следующие.

MatLab – высокопроизводительная система для технических расчетов, включающая вычисления, визуализацию и программирование в удобной среде,

где задачи и решения выражаются в форме, близкой к математической. MatLab можно использовать для:

- математических вычислений;
- создания алгоритмов;
- анализа данных, исследования и визуализации;
- научной и инженерной графики;
- разработки интерфейса, включая создание графического интерфейса.

MathCad – программное средство, разработанное фирмой Mathsoft Inc. в середине 80-х годов. Это среда для выполнения на компьютере разнообразных математических и технических расчетов, снабженная простым в освоении и в работе графическим интерфейсом, которая предоставляет пользователю инструменты для работы с формулами, числами, графиками и текстом. В среде MathCad доступны более сотни операторов и логических функций, предназначенных для численного и символьного решения математических задач различной сложности. Перечень вычислительных инструментов, доступных в среде MathCad, следующий:

- решение алгебраических уравнений и систем (линейных и нелинейных);
- решение обыкновенных дифференциальных уравнений и систем (задача Коши и краевая задача);
- решение дифференциальных уравнений в частных производных;
- статистическая обработка данных (интерполяция, экстраполяция, аппроксимация и многое другое);
- работа с векторами и матрицами (линейная алгебра и др.);
- поиск минимумов и максимумов функциональных зависимостей.

Можно отметить одну очень важную особенность, выгодно отличающую MathCad от других математических пакетов: математические выражения, обрабатываемые в MathCad, почти в точности повторяют обычную математическую символику.

Mathematica – многофункциональный интегрированный пакет, созданный примерно лет десять тому назад. В своих последних версиях имеет чрезвычайно широкий набор средств, переводящих сложные математические алгоритмы в программы. В этой системе реализованы все так называемые элементарные функции и огромное количество специальных функций, алгебраические и логические операции. Из вычислительных возможностей пакета можно выделить следующие:

- высокая степень точности вычислений;
- алгебраические и численные вычисления производных и интегралов;
- решение систем алгебраических, дифференциальных и разностных уравнений;
- наличие широкого набора встроенных математических функций (общим количеством более 200), включая преобразования Фурье, статистические и др.;

- поддержка целого ряда функций матричных и векторных вычислений;
- поддержка вычислений как в области действительных чисел, так и комплексных чисел.

Технология MathML, являющаяся расширением формата HTML, позволяет отображать на интернет-странице формулы, созданные в Mathematica, используя технологию кодирования Mathematica.

National Instruments LabVIEW – представляет собой высокоэффективную среду графического программирования с возможностью распределенного интеллекта для решения задач управления, измерений и проектирования. В LabVIEW можно создавать гибкие и масштабируемые приложения с минимальными временными и денежными затратами. LabVIEW сочетает в себе гибкость традиционного языка программирования с интерактивной технологией, которая включает в себя автоматическое создание кода, использование помощников при конфигурировании измерений, шаблоны приложений и настраиваемые Экспресс Виртуальные Приборы. Благодаря этим особенностям и новички, и эксперты могут легко и быстро создавать приложения в LabVIEW. Интуитивно понятный процесс графического программирования позволяет больше уделять внимания решению проблем, связанных с измерениями и управлением, а не программированию. LabVIEW имеет:

- интуитивно понятный процесс графического создания приложений для измерений, управления и тестирования;
- полноценный графический язык программирования;
- средства для сбора данных, управления приборами, обработки результатов, генерации отчетов, передачи данных и др.;
- совместимость с широким набором измерительных приборов благодаря наличию более 2000 драйверов;
- широкий набор шаблонов приложений и тысячи реальных примеров;
- высокая скорость выполнения откомпилированных программ.

LabVIEW может работать под управлением операционных систем Windows2000/NT/XP, Mac OS X, Linux и Solaris.

Благодаря своей гибкости и масштабируемости, LabVIEW может использоваться на всех этапах технологического процесса: от моделирования и разработки прототипов продуктов до широкомасштабных производственных испытаний. Применение интегрированной среды LabVIEW для измерения сигналов, обработки результатов и обмена данными повысит производительность всего предприятия.

Maple – система компьютерной математики (СКМ) создана в фирме Waterloo Maple Inc. и рассчитана на широкий круг пользователей. Maple с равным успехом может использоваться как для простых, так и для самых сложных вычислений и выкладок. СКМ Maple (к настоящему времени выпущена версия 16) предназначена для символьных вычислений, хотя имеет ряд средств и для численного решения. Обладает развитыми графическими

средствами. Имеет собственный язык программирования. Ядро системы Maple используется в ряде других математических систем, например в MATLAB и Mathcad, для реализации в них символьных вычислений. Так же, как и пакет Mathematica, СКМ Maple поддерживает технологию MathML.

СКМ Maple – типичная интегрированная программная система. Она объединяет в себе:

- мощный язык программирования (он же язык для интерактивного общения с системой);
- редактор для подготовки и редактирования документов и программ;
- современный многооконный пользовательский интерфейс с возможностью работы в диалоговом режиме;
- мощную справочную систему со многими тысячами примеров;
- ядро алгоритмов и правил преобразования математических выражений;
- численный и символьный процессоры;
- систему диагностики;
- библиотеки встроенных и дополнительных функций;
- пакеты функций сторонних производителей и поддержку некоторых других языков программирования и программ.

Система Maple прошла долгий путь развития и апробации. Она реализована на больших ЭВМ, рабочих станциях Sun, ПК, работающих с операционной системой Unix, ПК класса IBM PC, Macintosh и др.

Языки системы Maple. Maple способна решить огромное число задач вообще без какого-либо программирования в общепринятом смысле этого понятия. Достаточно лишь описать алгоритм решения задачи и разбить его на отдельные вопросы, на которые система Maple способна дать ответы. Более того, имеются примеры тысяч задач, для которых алгоритмы решения уже реализованы в виде её функций и команд. Тем не менее это вовсе не означает, что в Maple нельзя программировать. На самом деле Maple поддерживает три собственных языка: входной, реализации и программирования.

Входной язык. Maple имеет входной язык сверхвысокого уровня, ориентированный на решение математических задач практически любой сложности. Он служит для задания системе вопросов или, говоря иначе, задания входных данных для последующей их обработки. Это язык интерпретирующего типа и по своей идеологии напоминает Бейсик.

Язык программирования. Имеет Maple и свой язык процедурного программирования – Maple-язык. Этот язык имеет вполне традиционные средства структурирования программ: операторы циклов, операторы условных и безусловных переходов, операторы сравнения, логические операторы, команды управления внешними устройствами, функции пользователя, процедуры и т.д. Он также включает в себя все команды и функции входного языка, ему доступны все специальные операторы и функции. Многие из них являются весьма серьезными программами, например символьное

дифференцирование, интегрирование, разложение в ряд Тейлора, построение сложных трехмерных графиков и т.д.

Язык реализации. Языком реализации Maple является один из самых лучших и мощных универсальных языков программирования – Си. На нем написано ядро системы, содержащее тщательно оптимизированные процедуры. Большинство же функций, которые содержатся в пакетах, написаны на Maple-языке, благодаря чему их можно модифицировать и даже писать свои собственные библиотеки. Синтаксис структурных операторов языка Maple напоминает смесь Бейсика и Паскаля. Это облегчает знакомство с ним тем, кто имеет хотя бы начальный опыт программирования на этих языках.

2 Технологии работы с системой компьютерной математики Maple

2.1 Общие сведения о СКМ Maple

Работа в СКМ Maple организована в диалоговом режиме: вопрос – ответ в отдельном блоке. Блок выделяется слева квадратной скобкой, длина которой зависит от размеров и количества исходных выражений (вопросов) и результатов вычислений (ответов). Строка ввода математических выражений имеет отличительный символ >.

Алфавит языка содержит 26 прописных и строчных латинских букв (от А до Z и от а до z), 10 арабских цифр (0–9) и 32 специальных символа. Идентификатор должен быть уникальным, начинаться с буквы и может содержать буквы, цифры и знак подчеркивания. СКМ Maple различает прописные и строчные символы.

Выражение задается с помощью функций и операторов, записываемых в командной строке. Результат вычислений (по умолчанию) возвращается в виде математических формул. Ввод выражения завершается символом фиксации конца выражения – **точкой с запятой**, если ответ выводится в ячейку вывода, или **двоеточием**, если ответ не выводится.

Выражения формируются из операторов и операндов. **Операндами** могут быть константы, переменные и значения функций.

В СКМ Maple могут использоваться следующие **операторы**:

+	– оператор сложения
-	– оператор вычитания
*	– оператор умножения
/	– оператор деления
**, ^	– возведение в степень
!	– факториал
.	– десятичная точка
:=	оператор присваивания

-->	– функциональный оператор
<	– менее чем
>	– более чем
=	– равно
<=	– менее чем и равно
>=	– более чем и равно
<>	– неравно
or	логическое ИЛИ
and	логическое И

2.2 Функции

Важным понятием СКМ Maple является понятие функции. Функция возвращает результат некоторого преобразования исходных данных – параметров функции.

Встроенные функции Maple хранятся в его ядре и пакетах расширений⁴. Дополнительные функции из пакетов расширения должны применяться после объявления загрузки пакета с помощью команды

```
with (name),
```

где name – имя применяемого пакета.

Пакеты Maple охватывают многие крупные разделы математики и существенно дополняют возможности системы, предоставляемые средствами ее ядра. Пакеты расширения пишутся на Maple-языке программирования, поэтому они могут легко модернизироваться и пополняться.

СКМ Maple имеет полный набор элементарных математических функций. Практически все функции, кроме арктангенса двух аргументов, задаются именем и аргументом, в качестве которого может задаваться целое, рациональное, дробно-рациональное, вещественное или комплексное число или математическое выражение, например $\sin(x)$. В ответ на обращение к ним элементарные функции возвращают соответствующее значение. Поэтому они могут быть включены в математические выражения. Все эти функции называются встроенными, поскольку они реализованы в ядре системы.

Математические функции (все они известны):

$\sin(x)$, $\cos(x)$, $\tan(x)$, $\cot(x)$, $\arcsin(x)$, $\arccos(x)$,
 $\arctan(x)$, $\exp(x)$, $\sqrt{x}$, $\text{abs}(x)$,
 ilog10 , ilog – целочисленные логарифмы ($\text{ilog10}(25)=1$);
 $\ln(x)$, $\log[a](x)$, $\log_{10}(x)$ – логарифмы по разному основанию.

Некоторые целочисленные функции:

$\text{factorial}(n)$ – альтернатива $!$;
 $\text{iquo}(a,b)$ – частное от деления a на b ;
 $\text{irem}(a,b)$ – остаток от деления a на b ;
 $\text{igcd}(a,b)$ – наибольший общий делитель;
 lcm – наименьшее общее кратное;

Функции с элементами сравнения:

ceil – наименьшее целое $\geq x$;
 floor – наибольшее целое $\leq x$;

⁴ С перечнем пакетов расширений можно ознакомиться в справочной системе СКМ Maple.

frac – дробная часть числа x ;
trunc – меньшее целое, округленное в направлении к нулю;
round – округленное значение числа;
signum – функция знака $(-1, 1)$;

Функции пользователя в СКМ Maple могут задаваться разными способами. Рассмотрим на примерах.

Пример. Задать функцию $m = \sqrt{x^2 + y^2}$.

Способ 1. С помощью оператора присваивания **:=**.

> m:=sqrt(x^2+y^2); #определение функции

$$m := \sqrt{x^2 + y^2}$$

> x:=3: y:=4: m; #вычисление значения функции m для x=3,y=4

$$\sqrt{25}$$

Способ 2. С помощью функционального оператора **→** в формате:

name:=(x,y,...) → expr;

Вызов функции осуществляется в виде: name(x,y),

где x,y – список формальных параметров.

> m:=(x,y)→sqrt(x^2+y^2); #определение функции

$$m := (x, y) \rightarrow \sqrt{x^2 + y^2}$$

> m(3,4); #вычисление значения функции m для x=3,y=4

$$5$$

Способ 3. С помощью оператора unapply в формате:

name:= unapply(expr, var1, var2,...);

> m:=unapply(sqrt(x^2+y^2),x,y); #определение функции

$$fm := (x, y) \rightarrow \sqrt{x^2 + y^2}$$

> rez:=m(3,4); #вычисление значения функции m для x=3,y=4

$$rez := \sqrt{25}$$

Для оценивания выражения, т.е. представления его в числовом виде существует функция evalf (из группы eval).

Ее формат: evalf(expr, n) – вычисляет значение выражения expr и возвращает вычисленное значение в форме с плавающей точкой, имеющей n

значимых цифр. Параметр n является необязательным, при его отсутствии $n=10$.

```
> evalf(rez, 3);
```

5.00

2.3 Типовые средства графики

В само ядро Maple встроено ограниченное число функций графики. Это, прежде всего, функция для построения двумерных графиков `plot` и функция для построения трехмерных графиков `plot3d`. Они позволяют строить графики наиболее распространенных типов в различных системах координат как на плоскости, так и в трехмерном пространстве. Для построения графиков более сложных типов необходимо подключать пакеты расширений Maple.

Для построения **двумерных графиков** используется команда `plot`.

Формат: `plot(function, var_x {, var_y}{, option})`,

где `function` – функция, график которой строится;

`var_x` – переменная, указывающая область изменения по горизонтали;

`var_y` – переменная, указывающая область изменения по вертикали;

`option` – набор опций⁵, задающий стиль построения графика функции.

При построении графиков функцию можно определять через переменную, тогда в команде `plot` к ней можно обращаться по имени.

Пример. Построить график функции $y = x^3 + 1 - e^x$ для x в промежутке $[-8, 8]$, для y в промежутке $[-4, 12]$:

Решение:

```
> plot(x^3+1-exp(x), x=-8..8, y=-4..12, thickness=3);
```

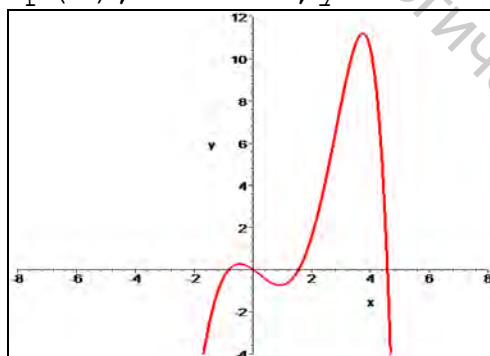


Рисунок 8.1 – График функции $y = x^3 + 1 - e^x$

Пример. Построить несколько графиков на одном рисунке.

Для вывода более одной графической зависимости на одной диаграмме, список нужных графиков следует заключить в квадратные скобки `[ ]`, оставляя прочие атрибуты команды `plot` неизменными.

⁵ Полный перечень опций для построения двумерных графиков приведен в справочной системе СКМ Maple.

Решение (рис. 8.2):

```
> plot([cos(x), x^2], x=-1..5, y=-4..4, color= [blue, black], thickness=3);
```

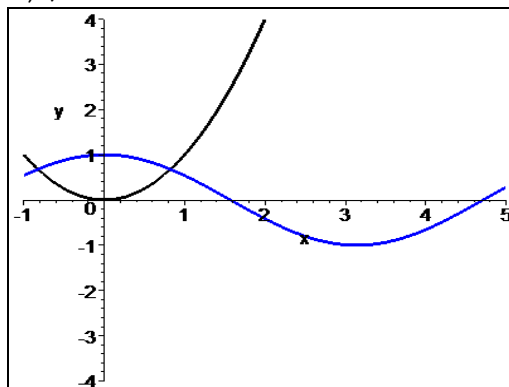


Рисунок 8.2 – Графики двух зависимостей на одной диаграмме

Здесь командой `color` определены цвета для каждой из функций. Цвета назначаются в том же порядке, в каком следуют функции. Цвета также должны быть заданы списком в квадратных скобках [].

Пример. Построить график кривой, заданной параметрическими зависимостями $x = t^2 - t$ и $y = 2t - t^3$ для интервала изменения параметра t в пределах $[-2, 2]$.

Команда `plot` может строить и графики параметрических кривых. Для прорисовки кривой, со значениями, вычисляемыми по двум параметрическим зависимостям: $x=f(t)$ и $y=g(t)$ на интервале $[a, b]$ команда `plot` используется в следующем формате:

```
plot([f(t), g(t), t=a..b], x=xmin..xmax, y=ymin..ymax);
```

В квадратных скобках перечислены функции x и y , область изменения параметра t , и отдельно указана область показа графика ($x = \text{xmin}..\text{xmax}$, $y = \text{ymin}..\text{ymax}$).

Решение (рис. 8.3):

```
> plot([t^2-t, 2*t-t^3, t=-2..2], x=-2..5, y=-5..5, thickness=3);
```

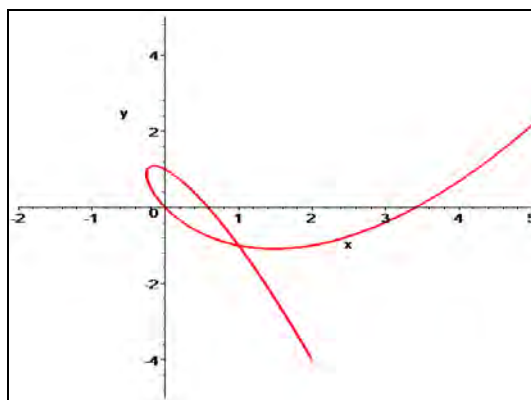


Рисунок 8.3 – График кривой, заданной параметрическими зависимостями

В СКМ Maple можно строить и трехмерные графики. Трехмерными называют графики, отображающие функции двух переменных $z(x, y)$. На деле трехмерные графики представляют собой объемные проекции в аксонометрии. Для построения **трехмерных графиков** СКМ имеет встроенную в ядро функцию `plot3d`. Она может использоваться в следующих форматах:

```
plot3d(expr1, x = a..b, y = c..d, p),
plot3d(f, a..b, c..d, p),
plot3d([exprf, exprg, exprh], s = a..b, t = c..d, p),
plot3d([f, g, h], a..b, c..d, p).
```

Здесь `p` – параметры, с помощью которых можно в широких пределах управлять видом трехмерных графиков.

Пример. Построить поверхность h^2 в цилиндрической системе координат.

Решение (рис. 8.4):

```
> plot3d(h^2, a=-Pi..Pi, h=-5..5, coords=cylindrical,
style =patch, color=sin(h));
```

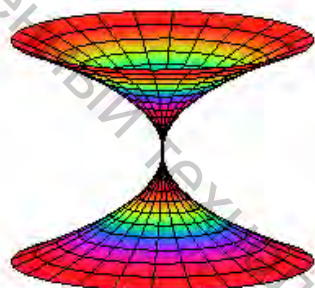


Рисунок 8.4 – Пример трехмерного графика

2.4 Решение уравнений

Для решения уравнений, неравенств и их систем в СКМ Maple используется функция `solve`, которая возвращает последовательность решений.

Формат: `solve(eqn, var)`,

где `eqn` – уравнение, неравенство или процедура;

`var` – имя переменной.

Уравнение и его решение можно представлять в виде отдельных объектов, отождествленных с определенной переменной.

Пример. Решить уравнение вида $y = x^2 + 2 \cdot x - 3$.

Решение:

```
> y:=x^2+2*x-3;          # задание уравнения через переменную eqn
```

```

> rez:=solve(y,x); # решение уравнения и присвоение корней
переменной rez.
> x1:= rez [1];      # присвоение первого корня переменной x1
> x2:= rez [2];      # присвоение второго корня переменной x2
> subs(x=x1, y);     # подстановка первого корня в уравнение
> subs(x=x2, y);     # подстановка второго корня в уравнение

```

Если решений нет или функция не может найти решение, то возвращается пустая последовательность NULL. В этом случае целесообразно использовать функцию fsolve, которая возвращает корень уравнения в форме вещественного числа.

Формат: fsolve(eqn, var),
 где eqn – уравнение, неравенство или процедура;
 var – имя переменной.

Пример. Решить уравнение вида $e^x + \ln(2 \cdot x) - 4,2 \cdot x = 0$.

Решение:

```

> solve(exp(x)+ln(2*x)-4.2*x);

```

0.2270523742 + 0.3228803688 I, 0.2270523742 - 0.3228803688 I, 1.886222494

Как видно из результата решения данного уравнения, корни представлены с использованием мнимой единицы, что не дает представления об их числовых значениях, поэтому для решения этого уравнения следует воспользоваться командой fsolve.

```

> fsolve(exp(x)+ln(2*x)-4.2*x);

```

1.886222494

2.5 Решение систем линейных алгебраических уравнений

Системы линейных алгебраических уравнений (СЛАУ) можно решать также, используя команду solve. Такое решение в силу простоты записи может быть предпочтительным. Для решения система уравнений и перечень неизвестных задаются в виде множеств, то есть с использованием фигурных скобок.

Пример. Решить систему уравнений вида:

$$s = \begin{cases} 4 \cdot x - 5 \cdot y + 5 \cdot z = 18 \\ 5 \cdot x + y - z = 8 \\ 6 \cdot x - 4 \cdot y + 3 \cdot z = 21 \end{cases}$$

Решение:

```

> y1:=4*x-5*y+5*z=18: y2:=5*x+y-z=8: y3:=6*x-
4*y+3*z=21:
> kor:=fsolve({y1,y2,y3},{x,y,z});

```

```

kor := {x = 2., y = -3., z = -1.}
> subs(kor, y1);
18. = 18
> subs(kor, y2);
8. = 8
> subs(kor, y3);
21. = 21

```

2.6 Вычисление интегралов и производных

Вычисление неопределенного интеграла обычно заключается в нахождении первообразной функции. Для вычисления неопределенных интегралов Maple предоставляет следующие функции:

`Int(f, x)` – отложенного действия, `int(f, x)` – прямого действия.

Команда отложенного действия используется для отображения интеграла в естественной математической записи в документах, команда прямого действия возвращает значение интеграла.

Для вычисления определенных интегралов используются команды:

`Int(f, x=a..b, continuous)` – отложенного действия,

`int(f, x=a..b, continuous)` – прямого действия.

Здесь f – подынтегральная функция,

x – переменная, по которой выполняются вычисления,

a и b – верхний и нижний пределы интегрирования,

`continuous` – необязательное дополнительное условие.

Для вычисления значения определенного интеграла необходимо использовать функцию `evalf`: `evalf(int(f, x=a..b))`.

Если верхним пределом интегрирования является бесконечность, то в функции `int` она обозначается словом `infinity`.

Пример. Вычислить значения определенных интегралов:

$$\int_0^1 \frac{\sin(x)}{x} dx \quad \text{и} \quad \int_0^{\infty} x e^{(-x)} dx = 1$$

Решение:

```
> Int(sin(x)/x, x=0..1.)=int(sin(x)/x, x=0..1.);
```

$$\int_0^1 \frac{\sin(x)}{x} dx = .9460830704$$

```
> Int(x*exp(-x), x=0..infinity)=int(x*exp(-x), x =
0..infinity);
```

$$\int_0^{\infty} x e^{(-x)} dx = 1$$

Вычисление производных функций $f^n(x) = df^n(x)/dx^n$ – одна из самых распространенных задач математического анализа. Для ее реализации СКМ Maple имеет следующие основные функции:

<code>diff(f, x1, x2, ..., xn)</code>	Производная первого порядка функции n переменных $\frac{d^j}{dx_j \dots dx_1} f$
<code>diff(f, [x1\$n])</code>	Производная n-го порядка функции одной переменной $\frac{d^n}{dx_1^n} f$
<code>diff(f, x1\$n, [x2\$n, x3], ..., xj, [xk\$m])</code>	Производная n-го порядка функции n переменных $\frac{d^n}{dx_k^m dx_j \dots dx_3 dx_2^n dx_1^n} f$

Здесь f – дифференцируемое алгебраическое выражение, в частности функция $f(x_1, x_2, \dots, x_n)$ ряда переменных, по которым производится дифференцирование;

$(x_1, x_2, \dots, x_n)$ – переменные, по которым осуществляется дифференцирование;

n – порядок производной.

Функция `Diff`⁶ является инертной формой вычисляемой функции `diff` и может использоваться для отображения производной в естественной математической записи в документах.

В простейшей форме `diff(f(x), x)` вычисляет первую производную функции $f(x)$ по переменной x . При n большем 1, вычисления производных выполняются рекурсивно, например `diff(diff(f(x), x), y)`. Или же для вычисления производных высокого порядка можно использовать оператор `$`. Например, выражение `diff(f(x), x$4)`, вычисляющее производную четвертого порядка по x , эквивалентно по записи `diff(f(x), x, x, x, x)`.

Пример. Вычислить значение производной первого порядка функции

$$f(x) = \frac{x^{27}}{27} + 5x^{0.2}$$

Решение :

> `f:=x^27/27+5*x^0.2;`

#Определение функции f:

⁶ Иначе функцию `diff` называют командой прямого действия, а функцию `Diff` – командой отложенного действия.

$$f := \frac{x^{27}}{27} + 5 x^{0.2}$$

> Diff(f, x) = diff(f, x);

#Вычисление производной:

$$\frac{d}{dx} \left(\frac{x^{27}}{27} + 5 x^{0.2} \right) = x^{26} + \frac{1.0}{x^{0.8}} :$$

Пример. Вычислить значение производных первого и второго порядка функции $f(x, y) = \cos(x) \cdot y^3$.

Решение :

Определяем функцию f:

> f(x, y) := cos(x) * y^3;

$$f(x, y) := \cos(x) y^3$$

Вычисляем производную 1-го порядка:

> Diff(f(x, y), x) = diff(f(x, y), x); #

$$\frac{\partial}{\partial x} \cos(x) y^3 = -\sin(x) y^3$$

Вычисляем производную 2-го порядка:

> Diff(f(x, y), x\$2, y\$2) = diff(f(x, y), x\$2, y\$2); #

$$\frac{\partial^4}{\partial y^2 \partial x^2} \cos(x) y^3 = -6 \cos(x) y$$

3 Использование СКМ Maple для решения задач экономического профиля

Пример. Объем выпуска ковровых изделий, млн m^2 , предприятиями Республики Беларусь в зависимости от года выпуска можно описать следующей зависимостью: $y = 0.011 * x^3 - 0.19 * x^2 + 0.69 * x + 8.19$,

где x – год выпуска продукции.

Построить кривую изменения объемов производства ковровых изделий предприятиями Республики Беларусь за период с 1995 по 2005 год. Определить предполагаемые значения объемов производства ковровых изделий за 2000 и 2005 годы.

Решение:

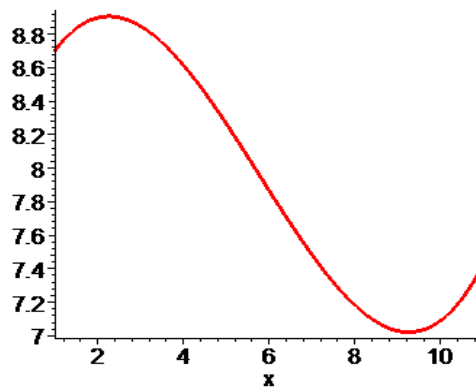
Определяем функцию y:

> y := 0.011 * x^3 - 0.19 * x^2 + 0.69 * x + 8.19;

$$y := 0.011 x^3 - 0.19 x^2 + 0.69 x + 8.19$$

Строим график функции y на интервале 1...11 (1995 г. – 1 период, 1996 г. – 2 период, ... 2005 г. – 11 период).

> plot(y, x=1..11, thickness=3);



Находим значения объемов производства в 2000 (6-й период) и 2005 (11-й период) гг.

```
> CH_2000:=subs (x=6, y) ;
                                CH_2000 := 7.866
> CH_2005:=subs (x=11, y) ;
                                CH_2005 := 7.431
```

Лекция 9. ТЕХНОЛОГИИ И ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПРОГРАММИРОВАНИЯ

План лекции

1. Основы алгоритмизации инженерно-технических задач.
 - 1.1. Алгоритмы и их свойства.
 - 1.2. Способы описания алгоритмов.
 - 1.3. Основные управляющие структуры.
2. Этапы подготовки и решения задачи на компьютере.
3. Типы данных языка Visual Basic for applications (VBA).
4. Операторы языка VBA.
5. Встроенные функции языка VBA.
6. Создание пользовательских функций в ТП MS Excel.

1 Основы алгоритмизации инженерно-технических задач

1.1 Алгоритмы и их свойства

Понятие «алгоритм» является одним из основных в информатике. Это слово происходит от имени узбекского математика аль-Хорезми, что означает «из Хорезма». В IX веке он разработал правила арифметических действий над десятичными числами. Длительное время алгоритмами пользовались только математики, понимая под алгоритмом любое описание процесса решения задачи. Описание процессов решения математических задач предназначалось для человека, поэтому не требовалось большой строгости в описании действий.

С развитием техники появились автоматические устройства, способные воспринимать команды и исполнять соответствующие действия. Для этих устройств последовательность действий должна быть четко сформулирована. Иначе говоря, исполнителю должен быть указан алгоритм решения задачи, представленный на понятном ему языке. Под исполнителем подразумевается как человек, так и компьютер.

Алгоритм решения задачи – это конечная последовательность четко сформулированных правил решения некоторого класса задач, приводящая к получению результата.

В обыденной жизни мы часто сталкиваемся с алгоритмами как с последовательностью действий, приводящих к достижению поставленной цели. Это правила перехода улицы, рецепты приготовления различных блюд, поиск нужного слова в словаре и др. Алгоритмы в математике – это правила

выполнения арифметических действий, нахождения наибольшего общего делителя и наименьшего общего кратного, разложения числа на простые множители и др.

Для адекватности восприятия и понимания исполнителем алгоритмы должны обладать целым рядом свойств: дискретностью, точностью, понятностью, результативностью, массовостью.

- **Дискретность** – это разбиение алгоритма на ряд отдельных законченных действий – шагов.

- **Точность** – это четкое указание последовательности шагов.

- **Понятность** – это однозначное понимание и исполнение каждого шага алгоритма его исполнителем.

- **Результативность** – обязательное получение результата за конечное число шагов.

- **Массовость** – применимость алгоритма к решению целого класса однотипных задач.

Если бы человек знал алгоритм решения всех задач, то их исполнение можно было бы поручить машине. Но оказалось, что не все задачи, которые нам хотелось бы решить, имеют алгоритмы решения. Задачи, в принципе не имеющие общего решения, называют алгоритмически неразрешимыми. Это, например, задача о трисекции угла, о квадратуре круга и др.

1.2 Способы описания алгоритмов

Существуют различные способы представления алгоритмов. Рассмотрим некоторые из них.

Текстовый – алгоритм решения задачи задается в виде текстового описания шагов решения. Таким способом составляются различные инструкции (например, пользование бытовыми приборами, управление автоматическими игрушками, рецепты приготовления блюд и др.). Эти алгоритмы предназначены для человека, поэтому в качестве команд могут использоваться привычные для человека предложения, фразы. Форма записи должна четко определять порядок исполнения команд.

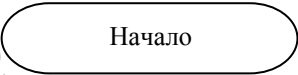
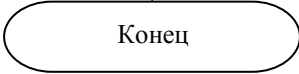
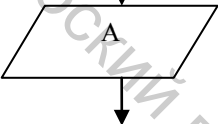
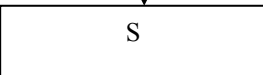
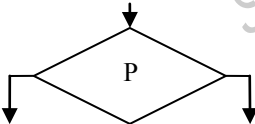
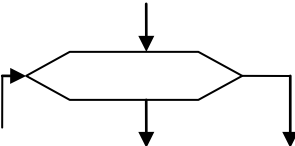
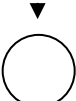
Математический – такой способ описания алгоритма также предназначен для человека и представляет собой логически связанную последовательность математических формул, реализующих решение задачи.

В виде блок-схем – схема алгоритма представляет собой графическое изображение алгоритма на формализованном языке, описывающего содержание и логические связи задачи. Под **блоком** подразумевается любой конечный этап вычислительного процесса, принимаемый в данной схеме как целое. В блок-схеме все части вычислительного процесса представлены фигурами, содержащими пояснения и соединенными между собой линиями. Применяемые графические символы, отражающие основные операции процесса обработки данных, устанавливает ГОСТ 19.003-80 (обозначение символов соответствует международному стандарту ISO 1028-73).

Правила выполнения схем алгоритмов регламентирует ГОСТ 19.002-80 (соответствует международному стандарту ISO 2636-73).

Часто используются блоки, представленные в таблице 9.1.

Таблица 9.1 – Блоки, используемые при построении блок-схем

	Начальный блок		Конечный блок
	Блок ввода – вывода. Используется для обозначения операций обмена информацией между устройствами		Блок вычислений или функциональный блок. В этом блоке осуществляется преобразование информации по заданному действию S
	Блок проверки условия используется для управления преобразованием информации. В результате анализа выполнения условия Р выбирается одно из двух возможных направлений «да» или «нет», и управление передается блоку, записанному на выбранном направлении		Блок модификации. Используется для организации цикла с параметром. Сначала управление передается к блоку по стрелке вниз, который повторяется заданное число раз, затем управление передается к блоку, следующему за блоком модификации
	Соединитель		Межстраничный соединитель

Блок-схема должна иметь начало и конец. Связь между блоками отмечается линиями потока информации, которые показывают последовательность решения задачи. Основное направление потока информации идет сверху вниз и слева направо. Здесь стрелки на линиях можно не указывать. Нумерация блоков производится от начала в порядке их расположения на блок-схеме. Направление линий потока, идущих снизу вверх и справа налево, показывается стрелками. По отношению к блоку линии потока могут быть входящими или выходящими. Количество входящих линий для блока принципиально не ограничено. Выходящая линия может быть только одна. Исключение составляют логические блоки (блоки проверки условия), имеющие не менее двух выходящих линий потока, каждая из которых соответствует одному из возможных исходов проверки логического условия. Над линиями выхода обозначается логическое условие выхода: «Да», если условие выполнено, и «Нет», если условие не выполнено), а также блоки модификации.

При значительном количестве пересекающихся линий, большой их длине

и многократных изменениях направления допускается разрывать линии потока информации, размещая на обоих концах разрыва соединитель.

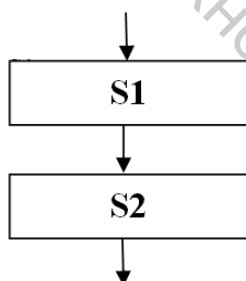
Внутри каждого поля соединителя ставится одинаковая маркировка отдельной буквой или буквенно-цифровой координатой блока, к которому подходит линия потока. Если схема располагается на нескольких листах, переход линий потока с одного листа на другой обозначается с помощью межстраничного соединителя. При этом на листе с блоком-источником соединитель содержит номер листа и координаты блока-приемника, а на листе с блоком-приемником – номер листа и координаты блока-источника. Внутри блоков и рядом с ними проставляют записи и обозначения так, чтобы их можно было читать слева направо и сверху вниз независимо от направления потока.

Представление алгоритма на языке программирования. На основании алгоритма, сформулированного одним из перечисленных способов, программистом составляется программа решения поставленной задачи на специальном языке для записи алгоритмов. Такие языки получили название алгоритмические языки или языки программирования высокого уровня.

1.3 Основные управляющие структуры

При изображении алгоритмов при помощи схем используются базовые управляющие структуры: следование, развилка, повторение.

Следование – структура, означающая, что действия S1 и S2 должны быть исполнены одно за другим. В качестве действий S1 и S2 могут выступать любые управляющие структуры или их последовательности.

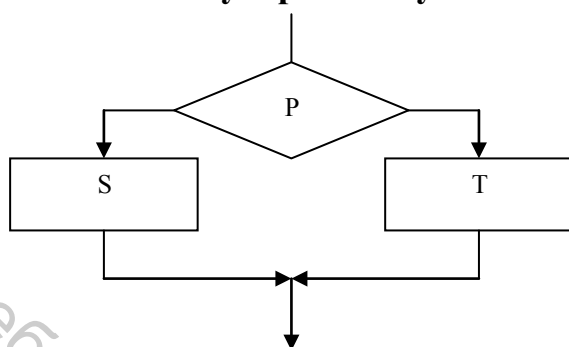


Примером использования структуры «следование» на производстве является технологическая последовательность на изготовление изделия. Здесь каждое действие представлено в виде технологически неделимой операции, а сама последовательность есть порядок выполнения этих операций. Например, Вы не сможете проложить отделочную строчку по воротнику, предварительно его не собрав.

Развилка – это действие, осуществляющее анализ условия Р (истинно или ложно), и альтернативный выбор дальнейшего направления в последовательности выполнения действий в зависимости от значения Р.

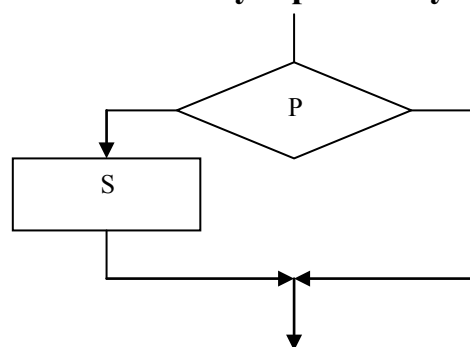
Различают:

полную развилку



и

неполную развилку



Словесно полная развилка описывается так: если условие P истинно, то исполнять S , иначе T .

Неполную развилку словесно можно описать так: если условие P истинно, то исполнять S .

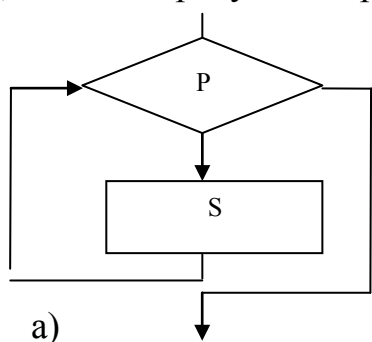
На производстве полной развилкой можно описать процесс определения качества изделия. Если изделие отвечает требуемым показателям, то оно является качественным, соответственно его можно пускать в продажу. Если изделие не отвечает предъявляемым требованиям, то оно считается бракованным, и его необходимо переделать. Неполную развилку можно представить на таком примере. Если в модели брюк есть накладной карман, то его необходимо настроить на требуемое место, иначе следует перейти к операции, следующей за настрачиванием кармана.

Повторение – это структура, описывающая циклические вычислительные процессы. Различают цикл-пока и цикл-до.

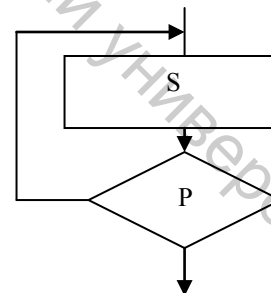
Цикл-пока словесно можно описать так: пока условие P истинно, выполнять тело цикла S .

Цикл-до: выполнять тело цикла S , до тех пор, пока условие P не станет истинным.

Все пять алгоритмических структур могут комбинироваться одна с другой, как того требует алгоритм.



а)



б)

Структурой «повторение» можно описать процесс изготовления изделия на поточной линии. Каждый рабочий выполняет определенную операцию. На потоке изготавливается несколько изделий, следовательно, рабочий должен повторить выполнение заданной операции столько раз, сколько изделий будет

изготовлено. Этот процесс продолжается до тех пор, пока не наступит перерыв (цикл-до). Если описать этот процесс циклом-пока, то условие продолжения цикла будет таким: процесс выполнения операции будет продолжаться, пока время работы рабочего не равно времени смены. В данном примере опущено время на обед или личные нужды рабочего, и процесс представлен укрупненно.

2 Этапы подготовки и решения задачи с использованием ПК

Решение любой задачи на ПК представляет собой сложный и трудный процесс, в котором на различных этапах принимают участие различные специалисты: экономисты, инженеры, математики, программисты, статистики и т.д. Этот процесс можно условно разбить на несколько этапов.

1. Постановка задачи. На этом этапе специалист формирует цели решения задачи. Подробно раскрывает ее содержание, указывает количество и характер всех величин, используемых в задаче в качестве исходных данных.

2. Математическая формулировка задачи. Здесь устанавливаются в окончательном виде те формулы и математические зависимости, которые подлежат решению.

3. Выбор численного метода. Среди всех численных методов решения данной задачи выбирается тот из них, который наилучшим образом обеспечивает выполнение всех требований этой задачи.

4. Разработка логической схемы алгоритма и ее минимизация. Разрабатывается алгоритм решения, т.е. устанавливается необходимая последовательность арифметических и логических действий, с помощью которых может быть реализован выбранный численный метод. Если возможно, то алгоритм упрощается (минимизируется) с помощью операций и формул математической логики. Полученный алгоритм изображается наглядно в виде блок-схемы.

5. Программирование. Алгоритм решения задачи на основании блок-схемы записывается на одном из языков программирования.

6. Отладка программы. Задача данного этапа состоит в том, чтобы путем опробования на компьютере вновь разработанной программы выявить ошибки, допущенные на всех предыдущих этапах. Для проверки правильности вычислений по составленной программе вручную решается один из вариантов задачи, называемый отладочным или контрольным вариантом. Затем этот же вариант по составленной программе просчитывается на ПК. При совпадении результатов счета на ПК и контрольного варианта считают, что программа составлена правильно.

7. Решение задачи на компьютере. После отладки программы производится непосредственное решение задачи на ПК с целью получения результатов для всех вариантов исходных данных.

8. Анализ полученных результатов. Производится специалистом, поставившим задачу. Если результаты по каким-либо причинам не устраивают,

принимаются решения либо об изменении алгоритма, либо об уточнении исходных данных.

Эти этапы выполняются последовательно, один за другим. Ошибка на предыдущем этапе может привести к получению некорректного результата, даже если в тексте программы нет ошибок. Поэтому при подготовке задачи к решению на ПК большое внимание уделяется именно начальным этапам: постановке задачи и ее математической формулировке, выбору метода решения и составлению алгоритма решения задачи.

3 Типы данных языка Visual Basic for applications (VBA)

VBA – язык программирования, достаточно простой для освоения и позволяющий конструировать профессиональные приложения для решения большинства задач, которые могут встретиться в среде Windows. В VBA применяется технология визуального программирования, т.е. конструирование приложения и элементов его управления непосредственно на экране, а также запись всей программы или её частей при помощи Macro Recorder.

Тип данных определяет множество допустимых значений, которые может принимать указанная переменная.

Таблица 9.2 – Основные типы данных VBA

Имя типа	Возможные значения	Требуемая память, байты
Byte (байт)	0 ... 255	1
Boolean (логический)	True, False	2
Integer (целое число)	- 32 768... +32 767	2
Long (длинное целое число)	-2 147 483 648... +2 147 483 647	4
Decimal (масштабируемое)	Примерно 30 десятичных цифр. Можно указать число цифр после десятичной точки	12
Single (число с плавающей запятой)	-3,4E38... -1,4,9E -324 для отрицательных значений 4,9E -324... 1,7E308 для положительных значений	8
Double (число с плавающей запятой двойной точности)	От -1,79769313486232E308 до -4,94065645841247E-324 для отрицательных значений; от 4, 94065645841247E-324 до 1,79769313486232E308 для положительных значений	8
Currency (денежный)	От -922 337 203 685 477,5808 до 922 337 203 685 477,5807	8

Окончание таблицы 9.2

String (строка)	Строки фиксированной длины имеют до 2^{16} символов. Строки переменной длины имеют до 2^{31} символов	10+1 на символ
Date (дата и время)	Даты изменяются в диапазоне от 1 января 100 года до 31 декабря 9999 года	8
Object (объект)	Ссылка на объект (указатель)	4
Variant (строковые подтипы)	Универсальный тип, значениями которого могут быть данные любого из перечисленных выше типов	Зависит от контекста, но не менее 16

Язык VBA поддерживает разные способы определения данных. Описание типа каждой переменной ускоряет работу программы и делает её более надёжной. Переменную, в принципе, можно объявлять или не объявлять, и тогда тип ей будет присвоен по умолчанию или по первой букве имени. Явно объявить переменную можно как в начале блока, так и в том произвольном месте, где возникла необходимость ее использования. Но для повышения надежности программ рекомендуется все переменные объявлять явно в начале блока. При этом определяется тип переменной и область видимости – область, где возможен доступ к ее значению.

Объявление **простых переменных** имеет следующий синтаксис:

```
{Dim | Private | Public | Static } < имя переменной> [ As <имя типа > ]
[ < имя переменной> [ As <имя типа > ] ]
```

Например: Dim Key As String

Dim Rez As Single

Константы могут определяться как простые, если в момент объявления задается только значение, и типизированные, если в момент объявления задается значение константы и ее тип. В принципе объявление констант мало чем отличается от объявления переменных, кроме того, что в этот момент задается значение, которое уже нельзя изменить. Для объявления именованных констант используется оператор Const, который имеет следующий синтаксис:

```
[Public | Private ] Const < имя константы> [ As type] = < константное выражение>
```

Например: Public Const Stavka As Double = 0.12

Кроме простых типов данных, при программировании на VBA достаточно широко используется простейший и самый распространенный структурный тип – массив, который может быть одномерным и многомерным. Возможное число измерений массива очень велико – 60. Синтаксис объявления массива аналогичен синтаксису объявления констант и переменных с той лишь разницей, что должна быть указана размерность массива и границы изменения индексов.

Синтаксис:

{Dim | Private | Public | Static} <имя переменной> (<список размерностей>) [As <имя типа >]

Например: Dim Rezult (1 to 10) As Integer

Если тип данных или тип объекта не задан, то по умолчанию переменная получает тип **variant**.

Вводимые данные должны отражать суть объекта, чтобы сделать программу легко читаемой. В VBA имеются следующие ограничения на имена:

- длина имени не должна превышать 255 символов;
- имя не может содержать точек, пробелов, символов: %, \$, @, &, #, !;
- имя начинается с буквы и содержит любую комбинацию букв, цифр и символов;
- имена должны быть уникальны внутри области, в которой они определены;
- нельзя использовать имена, совпадающие с ключевыми словами VBA.

Примеры имён: n

k _ нач

процентная _ ставка

Массив – совокупность однотипных индексированных переменных. Как правило, используются массивы с одним, двумя и тремя индексами. Количество индексов массива говорит о его размерности.

Примеры: Dim A (3, 3) As Single

Dim B (10) As Integer

Инструкции VBA представляют собой полную команду языка VBA. Инструкция может содержать ключевые слова, операторы, переменные, константы и выражения. В VBA имеются три категории инструкций:

- инструкции описания;
- инструкции присваивания;
- исполняемые инструкции.

5 Операторы VBA

Программа представляет собой последовательность операторов. Помимо оператора описания существуют три основных типа операторов: операторы присваивания, условные операторы и операторы цикла. Перед оператором в строке может стоять метка – последовательность символов, начинающаяся с буквы и кончающаяся двоеточием.

– **Операторы присваивания** – это основное средство изменения состояния. Они служат для изменения значения переменных и свойств объектов. Оператор присваивания присваивает значение выражения переменной, константе или свойству объекта и всегда включает знак равенства (=). Синтаксис оператора присваивания:

[Let] Переменная (или Постоянная или Свойство) = Выражение.

Инструкция **Let** необязательна и чаще всего опускается. Оператор присваивания предписывает выполнить выражение, заданное в его правой части, и присвоить результат переменной, имя которой указано в левой части. В результате, например, действия пары операторов

$x = 2$ и $x = x + 2$ переменной x будет присвоено значение 4.

– **Управляющие операторы** представлены в таблице 9.3. В число управляющих операторов входят условный оператор, оператор выбора, оператор перехода и различные операторы задания циклов.

Таблица 9.3 – Операторы VBA

Оператор	Действие
Go To	Оператор безусловного перехода. Синтаксис: <i>Go To Строка.</i> Задаёт безусловный переход на указанную строку внутри процедуры. Обязательный аргумент «Строка» может быть любой меткой или номером строки
If Then Else	Оператор условного перехода. Синтаксис: <i>If Условие Then [Инструкции] [Else Инструкции _ else].</i> Если «Условие» принимает значение <i>True</i> , то выполняется инструкция (или инструкции) после <i>Then</i> , если <i>False</i> , то выполняется инструкция (или инструкции) после <i>Else</i> . Ветвь <i>Else</i> является необязательной. Допускается также использование формы синтаксиса в виде блока: <i>If Условие Then</i> <i>[Инструкции]</i> <i>[Else If Условие - n Then</i> <i>[Инструкции _ else if] ...</i> <i>[Else</i> <i>[Инструкции _ else]]</i> <i>End If</i>
Select Case	Оператор выбора. Синтаксис: <i>Select Case выражение – текст</i> <i>[Case список Выражений – 1</i> <i>[инструкции - 1]]</i> <i>[Case список Выражений – n</i> <i>[инструкции - n]]</i> <i>[Case Else</i> <i>[инструкции _ else]]</i> <i>End Select</i>

Окончание таблицы 9.3

For – Next	Синтаксис: <i>For Счётчик = Начало To Конец [Step Шаг]</i> <i>[Инструкции]</i> <i>Next [Счётчик]</i> Повторяет выполнение группы инструкций, пока «Счётчик» изменяется от начального значения до конечного с указанным шагом. Если шаг не указан, то он полагается равным 1. Альтернативный способ выхода из цикла предоставляет инструкция <i>Exit For</i>
For Each – Next	Синтаксис: <i>For Each Элемент In Группы</i> <i>[Инструкции]</i> <i>[Exit For]</i> <i>[Инструкции]</i> <i>Next [Элемент]</i> Повторяет выполнение группы инструкций для каждого элемента массива или семейства. Альтернативный способ выхода <i>Exit For</i>
Do – Loop While	Синтаксис: <i>Do</i> <i>[Инструкции]</i> <i>[Exit Do]</i> <i>[Инструкции]</i> <i>Loop [While Условие]</i> Повторяет выполнение набора инструкций, пока условие имеет значение True. Сначала выполняется инструкция, а потом проверяется условие. Альтернативный способ выхода из цикла предоставляет инструкция <i>Exit Do</i>
Do While – Loop	Синтаксис: <i>Do [While Условие]</i> <i>[Инструкции]</i> <i>[Exit Do]</i> <i>[Инструкции]</i> <i>Loop</i> Повторяет выполнение набора инструкций, пока условие не примет значение True. Условие проверяется после выполнения инструкции, по крайней мере, один раз. Альтернативный способ выхода из цикла предоставляет инструкция <i>Exit Do</i>

Примеры:

- Условный оператор в виде одной строки:

If Rez > 0 Then Eff = 20% : Zatr = 0 Else Eff = 0 : Zatr = 10000

- Тот же условный оператор в виде блока:

```
If Rez > 0 Then  
Eff = 20% : Zatr = 0  
Else  
Eff = 0 : Zatr = 10000  
End If
```

- Оператор выбора

```
Select Case Цена  
Select Is >5000  
    Скидка = 10%  
Select Is >2000  
    Скидка = 7%  
Select Is >1000  
    Скидка = 5%  
End Select
```

В зависимости от значения переменной «Цена» после выполнения оператора Case будет присвоено значение переменной «Скидка».

- Оператор цикла For Next.

Вычисление общего дохода за 10 периодов.

Rezult = 0

For i = 1 To 10 Rezult = Rezult + Dohod

В переменной Rezult вычисляется общий доход за 10 месяцев (без учета ставки дисконта и инфляции).

5 Встроенные функции VBA

Встроенные функции VBA образуют основу, позволяющую программисту не задумываться над реализацией некоторых часто встречающихся стандартных действий.

Все встроенные функции VBA можно условно разделить на следующие классы: функции проверки типов данных, функции преобразования типов данных, функции преобразования данных, математические функции, функции обработки строк, функции обработки дат и времени.

Подробно все эти функции можно изучить в специальной литературе. В таблице 9.4 приведен перечень наиболее часто используемых при вычислениях стандартных математических функций.

Таблица 9.4 – Математические функции VBA

Функция	Назначение
Abs(число)	Абсолютное значение числа
Atn(число)	Арктангенс (в радианах) аргумента, задающего тангенс числа
Cos(число)	Косинус угла. Аргумент <i>число</i> задает угол в радианах
Exp(число)	Экспонента, т.е. результат возведения числа <i>e</i> (основание натурального логарифма) в указанную степень
Log(число)	Натуральный логарифм числа
Rnd([число])	Случайное число
Sgn(число)	Знак числа (если число больше нуля, то Sgn(число) = 1, число равно нулю Sgn(число)= 0, число меньше нуля Sgn(число)= -1)
Sin(число)	Синус угла. Аргумент <i>число</i> задает угол в радианах
Sqr(число)	Квадратный корень
Tan(число)	Тангенс угла. Аргумент <i>число</i> задает угол в радианах

6 Создание пользовательских функций в ТП MS Excel

Знание рассмотренных операторов позволяет разрабатывать как простые, так и достаточно сложные функции пользователя на VBA.

Следует отметить, что для реализации разветвляющихся вычислительных процессов в функциях пользователя проще использовать условный оператор в виде одной строки или даже обходиться конструкцией If ...Then, избегая Else. Для реализации циклических вычислительных процессов наиболее прост оператор For-Next.

Любую функцию пользователя нужно создавать в модуле той же рабочей книги Excel, где производятся вычисления, для чего необходимо в главном меню последовательно выбрать пункты:

- Сервис;
- Макрос;
- Редактор Visual Basic → Insert Modul.

В результате на экране появится текстовый редактор, в котором и создается модуль новой функции. Причем один модуль может включать несколько функций пользователя. После создания и сохранения функция автоматически включается в раздел «Функции, определенные пользователем» Мастера функций ТП MS Excel и вызывается обычным способом. Для выхода из Visual Basic на панели задач следует выбрать лист с нужным номером.

Составим пользовательскую функцию, которая рассчитывала бы размер премии в зависимости от оклада на основании приведенной ниже шкалы:

До 300 ден.ед.	От 300 до 400 ден.ед.	От 400 до 600 ден.ед.	Свыше 600 ден.ед.
Премии нет	10%	15%	20%

Решение

Очевидно, что здесь необходимо использовать конструкцию If ... Then ... Elseif ... Else ... End If для реализации сложного логического условия. Синтаксис такой конструкции достаточно сложен, поэтому реализуем расчет премии более простым способом, используя конструкцию If ... Then. Причем следует учесть тот вариант (оклад меньше 300 ден.ед.), когда функции присваивается строка символов (или значение 0) в зависимости от значения параметра.

По приведенной выше методике в окно редактора VBA необходимо ввести следующий текст:

```
Function ПРЕМИЯ1 (Оклад As Single)
If Оклад < 300 Then ПРЕМИЯ1 = "---"
If (Оклад >= 300) And (Оклад < 400) Then ПРЕМИЯ1 = Оклад * 0.1
If (Оклад >= 400) And (Оклад < 600) Then ПРЕМИЯ1 = Оклад * 0.15
If Оклад > 600 Then ПРЕМИЯ1 = Оклад * 0.2
End Function
```

В составленной функции в качестве параметров-значений передается только значение тарифного оклада, а также управляющий параметр, указывающий, каким образом начисляется премия.

В функции ПРЕМИЯ1 в каждом логическом блоке анализируется приведенный в заданной шкале интервал окладов и в зависимости от результата анализа реализуется формула расчета значения премии.

Использование функции ПРЕМИЯ1 представлено на рисунке 9.1.

Для расчета значения столбца «К выдаче» использована логическая функция ЕСЛИ: значения оклада и премии суммируются только в том случае, если премия начислена. Если в строке «Премия» помещен текст (или, как в данном примере, значение 0), сумма к выдаче равна значению столбца «Тарифный оклад».

	A	B	C	D	E	F
	Табельный №	ФИО	Тарифный оклад ден.ед.	Премия %	Премия ден.ед.	К выдаче ден.ед.
15						
16	1001	Трифонов А.А.	450	15	=Премия1(C16)	=ЕСЛИ(E1>0;C16+E16;C16)
17	1002	Дубко В.И.	320	10	=Премия1(C17)	=ЕСЛИ(E2>0;C17+E17;C17)
18	1003	Савин Т.Б.	500	15	=Премия1(C18)	=ЕСЛИ(E3>0;C18+E18;C18)
19	1004	Данилова Т.Р.	750	20	=Премия1(C19)	=ЕСЛИ(E4>0;C19+E19;C19)
20	1005	Беляева Т.В.	750	20	=Премия1(C20)	=ЕСЛИ(E5>0;C20+E20;C20)
21	1005	Масько П.Т.	280	0	=Премия1(C21)	=ЕСЛИ(E6>0;C21+E21;C21)

Рисунок 9.1 – Использование функции ПРЕМИЯ1

Результаты расчета представлены на рисунке 9.2.

	A	B	C	D	E	F
	Табельный №	ФИО	Тарифный оклад ден.ед.	Премия %	Премия ден.ед.	К выдаче ден.ед.
15						
16	1001	Трифонов А.А.	450,0	15,0	67,5	450,0
17	1002	Дубко В.И.	320,0	10,0	32,0	320,0
18	1003	Савин Т.Б.	500,0	15,0	75,0	575,0
19	1004	Данилова Т.Р.	750,0	20,0	150,0	900,0
20	1005	Беляева Т.В.	750,0	20,0	150,0	900,0
21	1005	Масько П.Т.	280,0	0,0	---	280,0

Рисунок 9.2 – Результаты расчета с использованием функции ПРЕМИЯ1

Пример

Разработать пользовательскую функцию $Y = f(a,b)$ на VBA для вычисления значения функции и построить в Excel график функции на

интервале $[m,n]$. $Y = \begin{cases} \sqrt{a}, & 0 \leq a < b \\ \sqrt{b}, & a \geq b \end{cases}$

Решение

В этом примере также следует использовать оператор If ... Then. Текст функции пользователя на языке VBA:

```
Function Y(a As Integer, b As Integer)
If a < 0 Then Y = "Y не определен"
If (a >= 0) And (a < b) Then Y = Sqr(a)
If a >= b Then Y = Sqr(b)
End Function
```

На рисунке 9.3 приведены варианты расчета значений Y.

Формула с использованием встроенной функции ЕСЛИ вводится в ячейку C2 в формате:

=ЕСЛИ(A2<0;"Y не определен"; ЕСЛИ(A2<=\$H\$3; КОРЕНЬ(A2); КОРЕНЬ(\$H\$3))),

затем копируется в ячейки C3:C15.

В столбце B2:B15 Y вычисляется с использованием встроенной функции ЕСЛИ, в столбце C2:C15 – при помощи функции пользователя Y.

Таким образом, знание основ языка VBA дает возможность пользователю создавать собственные функции для автоматизации решения задач из любой профессиональной области.

К достоинствам языка VBA можно отнести сравнительную лёгкость освоения, благодаря которой приложения могут создавать даже пользователи, не программирующие профессионально. К особенностям VBA можно отнести выполнение скрипта именно в среде офисных приложений.

C2		fx =Y(A2;\$H\$3)						
	A	B	C	D	E	F	G	H
	Аргумент a	Y Функция ЕСЛИ	Y Функция пользователя					
1								
2	-3	Y не определен	Y не определен		m	n	h	b
3	-2	Y не определен	Y не определен		-3	10	1	5
4	-1	Y не определен	Y не определен					
5	0	0	0					
6	1	1	1					
7	2	1,41	1,41					
8	3	1,73	1,73					
9	4	2,00	2,00					
10	5	2,24	2,24					
11	6	2,24	2,24					
12	7	2,24	2,24					
13	8	2,24	2,24					
14	9	2,24	2,24					
15	10	2,24	2,24					

Рисунок 9.3 – Расчет функции Y

Недостатком являются проблемы с обратной совместимостью разных версий. Эти проблемы в основном связаны только с тем, что код программы обращается к функциональным возможностям, появившимся в новой версии программного продукта, которые отсутствуют в старой. Также к недостаткам часто относят и слишком высокую открытость кода для случайного изменения, тем не менее многие программные продукты (например, Microsoft Office и IBM Lotus Symphony) позволяют пользователю использовать шифрование исходного кода и установку пароля на его просмотр.

Литература

1. Вардомацкая, Е. Ю. Компьютерные информационные технологии: конспект лекций / Е. Ю. Вардомацкая. – Витебск: УО «ВГТУ», 2019. – 115 с.
2. Вардомацкая, Е. Ю. Информатика: учебное пособие. В двух частях. Часть I / Е. Ю. Вардомацкая, Т. Н. Окишева. – Витебск, 2007. – 220 с.
3. Вардомацкая, Е. Ю. Информатика: учебное пособие. В двух частях. Часть II. Excel / Е. Ю. Вардомацкая, Т. Н. Окишева. – Витебск, 2007. – 237 с.
4. Громов, Ю. Ю. Информационные технологии : учебник / Ю. Ю. Громов [и др.]. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2015. – 260 с.
5. Евсеев, Д. А. WEB-дизайн в примерах и задачах: учебное пособие для студентов вузов, обучающихся по спец. «Прикладная информатика» и другим экономическим спец. / Д. А. Евсеев, В. В. Трофимов; Санкт-Петербургский государственный университет экономики и финансов ; под ред. В. В. Трофимова. – Москва: Кнорус, 2018. – 263 с.
6. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы: учебное пособие для студентов высших учебных заведений, обучающихся по направлению «Информатика и вычислительная техника» и по специальностям «Вычислительные машины, комплексы, системы и сети», «Автоматизированные машины, комплексы, системы и сети», «Программное обеспечение вычислительной техники и автоматизированных систем» / В. Г. Олифер, Н. Олифер. – 4-е изд. – Санкт-Петербург: Питер, 2015. – 943 с.
7. Компьютерные информационные технологии. Задания для тестового контроля знаний: методические указания. – Витебск: УО «ВГТУ», 2018. – 58 с.
8. Круглова, О. В. Информационные технологии в управлении: учебное пособие / О. В. Круглова. – Дзержинск: изд-во «Конкорд», 2016. – 134 с.
9. Титоренко, Г. А. Информационные технологии в маркетинге : учебник / под ред. Г. А. Титоренко. – М.: ЮНИТИ-ДАНА, 2016. – 335 с.
10. Федосеева, Н. Н. Сущность и проблемы электронного документооборота в информационных технологиях / Н. Н. Федосеева // Юрист. – 2017. – № 6. – С. 61.
11. Internet-ресурс. Электронные учебные пособия кафедры «Математика и информационные технологии». – Режим доступа: <http://sdo.vstu.by/course/view.php?id=32>.
12. Internet-источник // iit.khsu.ru/isitc/informatika/6_2.htm
13. Internet-источник // www.stu.ru/inform/glaves2/glava8/gl_8_2.htm

Учебное издание

Елена Юрьевна Вардомацкая

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ТЕХНИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Конспект лекций

Редактор *Т.А. Осипова*

Корректор *Т.А. Осипова*

Компьютерная верстка *Е.Ю. Вардомацкая*

Подписано к печати 22.07.2022. Формат 60x90¹/₁₆. Усл. печ. листов 7,3.
Уч.-изд. листов 9,3. Тираж 70 экз. Заказ № 214.

Учреждение образования «Витебский государственный технологический университет»
210038, г. Витебск, Московский пр., 72.

Отпечатано на ризографе учреждения образования

«Витебский государственный технологический университет».

Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/172 от 12 февраля 2014 г.

Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 3/1497 от 30 мая 2017 г.