

КОМПЬЮТЕРНЫЕ
ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ

Конспект лекций

Витебский государственный технологический университет

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ»

КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Конспект лекций

для студентов специальностей 1-25 01 04 «Финансы и кредит»,
1-25 01 07 «Экономика и управление на предприятии»,
1-25 01 08 «Бухгалтерский учет, анализ и аудит»,
1-25 01 10 «Коммерческая деятельность», 1-26 02 03 «Маркетинг»

ВИТЕБСК
2013

УДК 004
ББК 32.97
К63

Рецензенты:

д.э.н., заведующая кафедрой экономической теории и маркетинга УО «Витебский государственный технологический университет» Г.А. Яшева;
к.т.н., заведующий кафедрой информатики УО «Витебский государственный технологический университет» В.Е. Казаков.

Рекомендовано к изданию редакционно-издательским советом УО «ВГТУ», протокол № 7 от 30.10.2013.

Компьютерные информационные технологии : конспект лекций / сост.:
Е. Ю. Вардомацкая ; УО «ВГТУ». - Витебск, 2013. - 116 с.

ISBN 978-985-481-331-8

Конспект лекций составлен в соответствии с действующей учебной программой дисциплины «Компьютерные информационные технологии» для специальностей 1-25 01 04 «Финансы и кредит», 1-25 01 07 «Экономика и управление на предприятии», 1-25 01 08 «Бухгалтерский учет, анализ и аудит», 1-25 01 10 «Коммерческая деятельность», 1-26 02 03 «Маркетинг» заочной формы 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 в профессиональной деятельности специалистов экономического профиля. Значительное внимание уделено изучению возможностей использования пакетов прикладных программ для анализа и компьютерного моделирования задач экономики и управления.

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 978-985-481-331-8

© Вардомацкая Е.Ю., 2013
© УО «ВГТУ», 2013

СОДЕРЖАНИЕ

Лекция 1. ПРЕДМЕТ И ОСНОВНЫЕ ПОНЯТИЯ КУРСА «КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ».....	5
1 Предмет и задачи курса «Компьютерные информационные технологии»...	5
2 Понятие информационной технологии.....	6
3 Составляющие информационных технологий.....	7
4 Свойства информационных технологий.....	8
5 Классификация информационных технологий.....	9
6 Проблемы и перспективы развития информационных технологий.....	12
7 Литература к лекции 1.....	14
Лекция 2. СЕТЕВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.....	15
1 Классификация компьютерных сетей.....	15
2 Глобальная сеть Интернет.....	16
3 Стек протоколов TCP/IP.....	17
4 Протоколы прикладного уровня сети Internet.....	18
5 Сервисы Internet, их назначение и особенности.....	19
6 World Wide Web. Протокол HTTP. URL-адресация web-ресурсов.....	19
7 Виды информационных ресурсов в WWW.....	21
8 Литература к лекции 2.....	23
Лекция 3. ТЕХНОЛОГИИ И СИСТЕМЫ ОБРАБОТКИ ТАБЛИЧНОЙ ИНФОРМАЦИИ.....	24
1 Табличные процессоры и их функциональные возможности.....	24
2 Основные понятия табличного процессора MS Excel.....	25
3 Технология разработки электронной таблицы.....	28
4 Вычисления в таблицах.....	28
5 Встроенные функции рабочего листа ТП MS Excel.....	29
6 Средства деловой графики.....	43
7 Литература к лекции 3.....	45
Лекция 4. ПАКЕТЫ ДЛЯ МАТЕМАТИЧЕСКОЙ ОБРАБОТКИ ДАННЫХ.....	46
1 Характеристика и функциональные возможности пакетов для математической обработки данных.....	46
2 Технологии работы с системой компьютерной математики Maple.....	50
2.1 Общие сведения о СКМ Maple.....	50
2.2 Функции.....	51
2.3 Типовые средства графики.....	53
2.4 Решение уравнений.....	55
2.5 Решение систем линейных алгебраических уравнений.....	56
2.6 Вычисление интегралов и производных.....	57
3 Использование СКМ Maple для решения задач экономического профиля... ..	59
4 Литература к лекции 4.....	60

Лекция 5. ПОНЯТИЕ БАЗЫ ДАННЫХ. ВВЕДЕНИЕ В СТРУКТУРИРОВАННЫЙ ЯЗЫК ЗАПРОСОВ SQL.....	61
1 Язык SQL в СУБД.....	61
2 Структура команды SQL. Типы данных. Выражения.....	62
3 Описание команд языка SQL.....	63
4 Литература к лекции 5.....	67
Лекция 6. ТЕХНОЛОГИИ ВЕДЕНИЯ БАЗ ДАННЫХ И ВЫБОРА ДАННЫХ ИЗ БАЗЫ.....	68
1 Оператор выбора данных SELECT.....	68
2 Устранение избыточности данных.....	69
3 Определение условия выбора WHERE.....	69
4 Специальные операторы в условиях (IN, BETWEEN, LIKE).....	71
5 Функции агрегирования. Параметры GROUP BY и HAVING.....	73
6 Упорядочивание выходных полей.....	75
7 Литература к лекции 6.....	76
Лекция 7. ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ ДЛЯ МОДЕЛИРОВАНИЯ И ЧИСЛЕННОГО РЕШЕНИЯ ЗАДАЧ ЭКОНОМИКИ И УПРАВЛЕНИЯ.....	78
1 Основные понятия и задачи моделирования.....	78
2 Модули стандартных офисных пакетов для работы с корреляционно-регрессионными моделями.....	79
3 Анализ и решение задач межотраслевого баланса	86
4 Анализ и решение задач оптимизации плана производства продукции.....	88
5 Анализ и решение задач оптимизации плана транспортных перевозок.....	92
6 Литература к лекции 7.....	97
Лекция 8. ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ ДЛЯ МОДЕЛИРОВАНИЯ И СИМВОЛЬНОГО РЕШЕНИЯ ЗАДАЧ ЭКОНОМИКИ И УПРАВЛЕНИЯ.....	98
1 Стандартные библиотеки пакетов символьной математики для решения задач моделирования и оптимизации.....	98
2 Корреляционно-регрессионное моделирование.....	99
3 Анализ и решение задач межотраслевого баланса.....	102
4 Анализ и решение задач оптимального планирования.....	103
5 Литература к лекции 8.....	106
Лекция 9. КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ.....	107
1 Понятие информационной системы.....	107
2 Классификация и виды обеспечения информационных систем.....	108
3 Принципы организации корпоративных информационных систем.....	110
4 Современная структура корпоративной информационной системы.....	111
5 Информационное обеспечение корпоративных информационных систем....	113
6 Требования к корпоративным информационным системам.....	114
7 Литература к лекции 9.....	115

Лекция 1

ПРЕДМЕТ И ОСНОВНЫЕ ПОНЯТИЯ КУРСА «КОМПЬЮТЕРНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

План лекции

1. Предмет и задачи курса “Компьютерные информационные технологии” (КИТ) .
2. Понятие информационной технологии.
3. Составляющие информационных технологий.
4. Свойства информационных технологий.
5. Классификация информационных технологий.
6. Проблемы и перспективы развития информационных технологий.
7. Литература.

1 Предмет и задачи курса “Компьютерные информационные технологии” (КИТ)

Курс “Компьютерные информационные технологии” (КИТ) является базовым в цикле подготовки студентов экономических специальностей в области использования информационных технологий в профессиональной деятельности. В этой дисциплине 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, основы компьютерного моделирования практических задач экономики, основы бизнес-моделирования задач управления. Этот курс предназначен не только систематизировать получаемые студентами знания, но и должен привести их в соответствии 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

контроля за ее решением, что так необходимо для современного руководителя.

В результате изучения курса КИТ 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 видеопроизведений;
- 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, усваиваемые 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 киноленте 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 (рисунок 1.1).

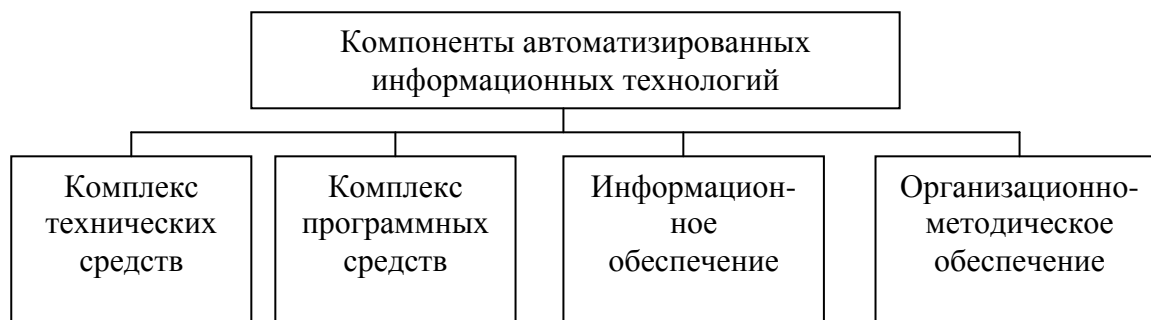


Рисунок 1.1 – Структура автоматизированных информационных технологий

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.

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

Возможные способы классификации информационных технологий по другим классификационным признакам представлены в таблице 1.1.

Таблица 1.1– Классификация информационных технологий

Классификационный признак	Вид информационной технологии
Степень автоматизации задач управления	<ul style="list-style-type: none"> – электронная обработка данных; – автоматизация функций управления; – электронный офис; – поддержка принятия решений; – экспертная поддержка
Тип пользовательского интерфейса	<ul style="list-style-type: none"> – 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	<ul style="list-style-type: none"> – локальные; – многоуровневые; – распределенные
Тип носителя	<ul style="list-style-type: none"> – бумажная (входные и выходные документы); – безбумажная (сетевая технология, современная оргтехника, электронные деньги, документы)
Степень типизации операций	<ul style="list-style-type: none"> – пооперационная – за каждой операцией закрепляется рабочее место с техническим средством, она присуща пакетной технологии; – попредметная – подразумевает выполнение всех операций на одном рабочем месте, например, автоматизированное рабочее место (АРМ)
Вид информационного процесса	<p>информационная технология обработки данных (см. рисунок 1.3); информационная технология управления (см. рисунок 1.4)</p>

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.

На рисунках 1.3 и 1.4 для примера представлены схемы информационной технологии обработки данных и информационной технологии управления.

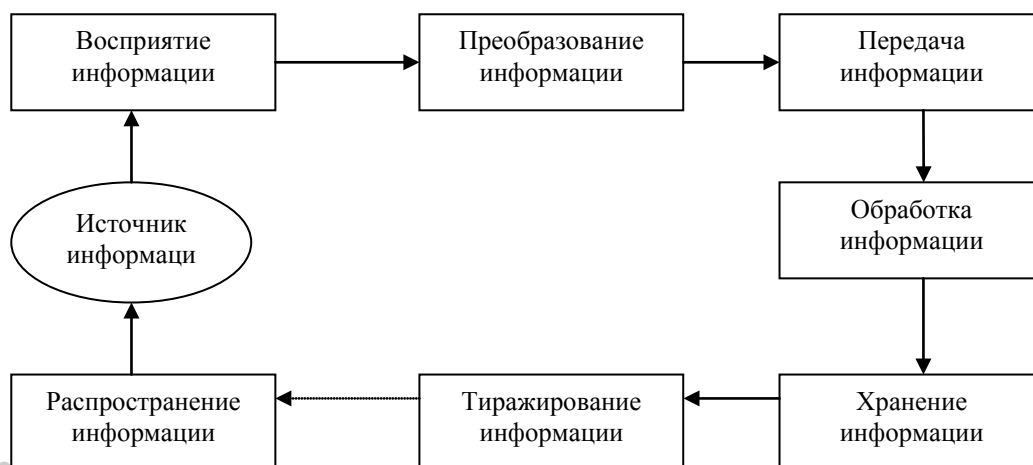


Рисунок 1.3 – Информационная технология обработки данных



Рисунок 1.4 – Информационная технология управления

6 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 и т. д.

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. Основной 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.

7 Литература к лекции 1

1. Алексеев, А. П. Информатика 2007: учебное пособие для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов 210400 (654400) – Телекоммуникации / А. П. Алексеев. – Москва: Солон-Пресс, 2007. – 608 с.

2. Информатика для юристов и экономистов: учебник для вузов / под ред. С. В. Симоновича. – Санкт-Петербург: Питер, 2006.

3. Коуров, Л. В. Информационные технологии / Л. В. Коуров. – Минск: Амалфея, 2000. – 192 с.

4. Трещалин, М. Ю. Основы информационных технологий: учебное пособие для студентов вузов / М. Ю. Трещалин. – Москва: Элит, 2007. – 108 с.

5. Шарстнев, В. Л. Компьютерные информационные технологии: курс лекций / В. Л. Шарстнев. – Витебск: УО ВГТУ, 2008. – 350 с.

6. Internet-ресурс <http://www.alfcomp.ru>.

Лекция 2

СЕТЕВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

План лекции

1. Классификация компьютерных сетей.
2. Глобальная сеть Интернет.
3. Стек протоколов TCP/IP.
4. Протоколы прикладного уровня сети Internet.
5. Сервисы Internet, их назначение и особенности.
6. World Wide Web. Протокол HTTP. URL-адресация web-ресурсов.
7. Виды информационных ресурсов в WWW.
8. Литература.

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).

- **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 следующими признаками.

- **Методами передачи данных.** В локальных сетях используются методы, не требующие предварительной установки соединения. Глобальные сети ориентированы на соединение.

- *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 вышеперечисленных 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:

- *Компьютеры.* Разделяются на компьютеры, предоставляющие ресурсы (серверы), и компьютеры, потребляющие ресурсы (рабочие станции – клиенты).
- *Линии связи или каналы передачи данных.*
- *Коммутационное оборудование.* Используется для связи сегментов сети.
- *Соединительное оборудование.*

Программный компонент КС определяют:

- *Сетевая операционная система (ОС)* – связывает и координирует функции всех компьютеров и периферийных устройств в сети.
- *Сетевые программные приложения* – это прикладные программы, которые расширяют возможности сетевых ОС.

2 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.

Первая в мире компьютерная сеть ARPAnet была создана в конце 1969 г.

и стала прототипом сети Интернет¹.

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

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

Протоколы транспортного уровня 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.

¹ Название сети ARPAnet происходит от названия ее учредителя *Advanced Research Project Agency (ARPA* — Управление перспективных исследовательских программ США).

Протокол TCP сам разбивает данные на пакеты, обеспечивает 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.

4 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 частей программы. Выделяют следующие наиболее известные прикладные протоколы:

1. **HTTP** (Hyper Text Transfer Protocol) – протокол передачи гипертекста. Используется в WWW (World Wide Web – всемирная паутина) 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 каждый пользователь Web-страниц 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.

2. **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.

3. **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.

4. **TELNET** – используется для подключения и управления удаленным компьютером. После подключения каждый символ, введенный на локальном компьютере, обрабатывается так, как если бы он был введен на удаленном компьютере. Фактически TELNET – это протокол эмуляции терминала.

5 Сервисы Internet, их назначение и особенности

Способы использования Интернета определяются предоставляемыми посетителям сервисами, в число которых входят традиционные, специальные и новые виды сервиса.

К **традиционным видам сервиса** относятся:

- электронная почта (e-mail);
- телеконференции (news);
- живое общение (chat);
- использование вычислительных и информационных ресурсов компьютера в режиме удаленного терминала (telnet);
 - служба хранения, поиска и пересылки файлов – FTP (File Transfer Protocol);
 - WWW (World Wide Web) – всемирная паутина (в ее составе в июле 1993 г. было 130 хост² – компьютеров, а в июне 1996 г. – 230000).

К **специальным сервисным средствам** относятся:

- информационная система Gopher;
- информационная система WAIS, ведущая поиск по ключевым словам;
- библиографические (диспетчерские) системы;
- Archie – поисковая система FTP;
- поисковые системы WWW.

Новые виды сервиса включают в себя:

- интернет-банк;
- системы продажи ж/д и авиа-билетов;
- системы для заказа товаров по сети (цветы, пицца, авиабилеты, номер в гостинице и т. д.);
- библиотечные системы;
- электронные издания газет и журналов, блоги;
- сервисные представительства фирм;
- коммерческие информационные системы по производителям товаров и услуг, котировкам акций на фондовых биржах и др.

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

Служба **WWW. World Wide Web** (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,

² Хост – это определенное подключенное к Интернету устройство – компьютер, с которого сделан вход на определенный сайт. Сколько бы страниц посетитель ни просмотрел на сайте, сколько бы времени он на нем ни оставался, все это будет считаться одним хостом.

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 **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, был разработан универсальный формат для веб-документов — 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://ru.wikipedia.org/wiki/URL>.

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/mails/redirect?page=40>.

Чаще используются простые URL-адреса. Например: <http://www.pfu.edu.ru>, <http://www.imeb.ru>.

7 Виды информационных ресурсов 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** — это веб-сайты, обычно служащие начальной точкой поиска информации, содержащие 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. Для поиска информационных ресурсов в сети предназначены поисковые системы, которые можно разделить на три группы:

- **Тематические каталоги**, которые представляют собой иерархическую базу данных, организованную 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). С 1994 г. в сети Интернет функционирует каталог *Lycos* (www.lycos.com). Он осуществляет рейтинг страниц, зарегистрированных в этом каталоге. Несколько позже, в 1995 г., появился целый ряд поисковых систем, имеющих и тематические каталоги. Одна из таких систем — *InfoSeek* (www.infoseek.com). Часть страниц, попавших в этот каталог, получают рейтинг сотрудников компании. Система *Excite* (www.excite.com) также появилась в 1995 г. Она имеет две особенности: возможность поиска по серверам телеконференций; на некоторые страницы имеются рецензии сотрудников сервера. Поисковая система *WebCrawler* (www.webcrawler.com) имеет также свой каталог рецензированных страниц. Один из старейших русскоязычных тематических каталогов — это каталог *WebList* (www.weblist.ru).

- **Поисковые системы** – это специально созданные системы для поиска информации в Интернете по заданным условиям. 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, осуществляющие контекстный поиск по индексным базам данных. Список наиболее популярных поисковых систем приведен в таблице 2.1.

Таблица 2.1 – Популярные 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 точные результаты поиска и меньшая затрата времени на поиск. 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. К числу мегапоисковых систем относятся www.dogpile.com, www.37.com, meta360.com и т. д.

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.

8 Литература к лекции 2

1. Шарстнев, В. Л. Компьютерные информационные технологии: курс лекций / В. Л. Шарстнев. – Витебск: УО ВГТУ, 2008. – 350 с.
2. Информатика для юристов и экономистов: учебник для вузов / под ред. С. В. Симоновича. – Санкт-Петербург: Питер, 2006.
3. Велихов, А. В. Основы информатики и компьютерной техники: учебное пособие для студентов ссузов и вузов по дисциплине "Основы информатики" / А. В. Велихов. – Москва: СОЛОН-ПРЕСС, 2007.
4. Информатика. Базовый курс: учебное пособие для студентов высших технических учебных заведений / под ред. С. В. Симоновича. – 2-е изд. – Санкт-Петербург: Питер, 2007. – 640 с.
5. Морозевич, А. Н. Прикладная информатика: учебное пособие / А. Н. Морозевич. – Минск: Выш. школа, 2003. – 335 с.: ил
6. <http://lessons-tva.info/favorite.html>.

Лекция 3

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.
7. Литература.

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.), SuperCulk, 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, 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 (2^8) столбцов и 65536 (2^{16}) строк.

- **Ячейка** – область, определяемая пересечением столбца и строки электронной таблицы (ЭТ). На рабочем листе расположено **16 777 216** ($2^8 \times 2^{16} = 2^{24}$) 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 C65536.³

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

³ Изменить формат адресации можно, выбрав из меню Офис (Файл) ==> Параметры Excel ==> Формулы ==> Стиль ссылок R1C (Excel 2010).

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 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**⁴ – 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 3.1.

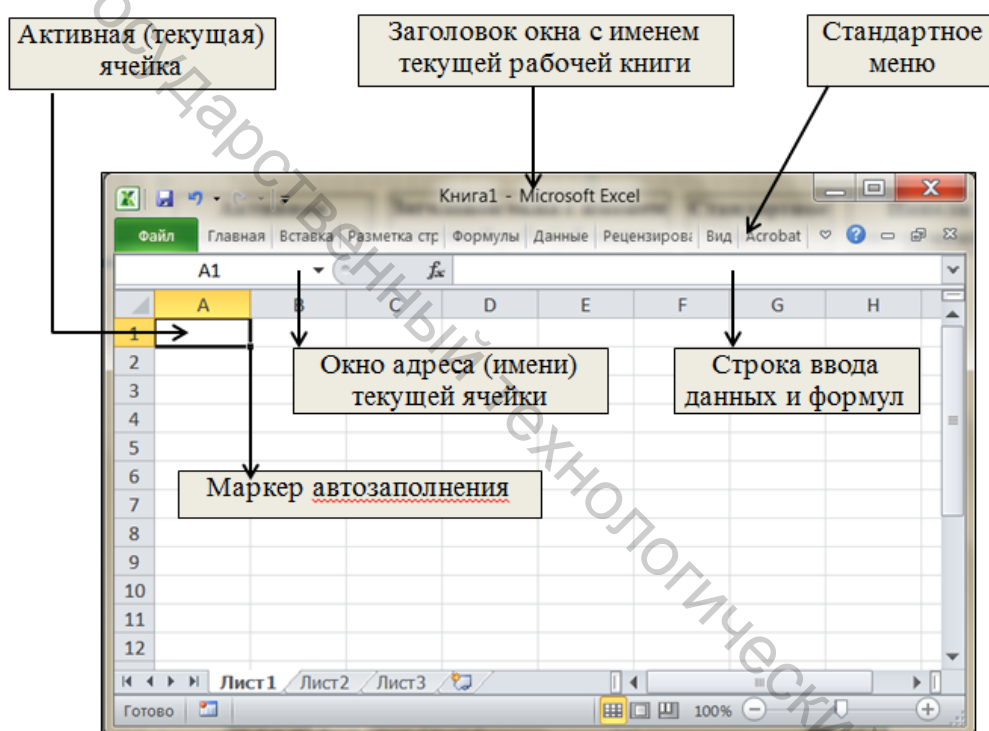


Рисунок 3.1 – Окно Microsoft Excel 2010

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:

XLS – файл рабочей книги;

XLC – файл деловой графики;

XLT – файл шаблона;

XLA – файл дополнительных макрок команд;

XLB – файл описания пиктографического меню и пр.

⁴ 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*.

Арифметические операторы служат для выполнения арифметических действий над числами. В таблице 3.1 представлены арифметические операторы, которые могут использоваться в Excel.

Таблица 3.1 – 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

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).

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 на ячейку*, *абсолютную ссылку на ячейку* 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 [2].

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 функции⁵

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

⁵ С форматами и примерами использования всех встроенных функций можно подробно ознакомиться в справочной системе ТП MS Excel.

решения определенной задачи.

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 3.2.

Таблица 3.2 – 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: <ul style="list-style-type: none"> ▪ 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. 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.

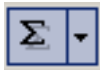
- **СУММ(число1; число2; . . . ; число n)** – вычисляет сумму аргументов.

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.

На рисунке 3.2 представлена таблица, в которой вычисляется общая сумма премии высокооплачиваемых сотрудников (тех, у кого оклад больше 4 млн. руб.). Для этого анализируется столбец *Оклад* (диапазон ячеек A2:A7), и суммируются те ячейки из столбца *Премия* (диапазон ячеек C2:C7), для которых значения в соответствующих ячейках столбца *Оклад* больше 4000000. Результат помещен в ячейку C8.

	A	B	C	D	E
1	Оклад	% премии	Премия		
2	4250000	10%	425000		
3	3300000	15%	495000		
4	3350000	15%	502500		
5	4200000	15%	630000		
6	4700000	20%	940000		
7	2200000	10%	220000		
8			1995000		

Рисунок 3.2 – Пример использования функции СУММЕСЛИ

▪ **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 8293 рубля, функция =ОКРВВЕРХ(8293;10) округлит цену до 8300 руб., а функция =ОКРВНИЗ(8293;10) округлит цену до 8290 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)**

Массив1; массив2; ... ; массив n – от 2 до 30 массивов, чьи компоненты нужно перемножить, а затем сложить.

На рисунке 3.3 показано, как в ячейке С9 с использованием функции СУММПРОИЗВ может быть вычислена общая сумма премии по ведомости.

- **СУММКВРАЗН(массив_x; массив_y)**

- *массив_x* – первый массив или интервал значений.

- *массив_y* – второй массив или интервал значений.

- **СУММРАЗНКВ(массив_x; массив_y)** возвращает сумму разностей квадратов соответствующих значений в двух массивах.

- **СУММСУММКВ(массив_x; массив_y)** 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.

	A	B	C	D
1	Оклад	% премии	Премия	
2	4250000	10%	425000	
3	3300000	15%	495000	
4	3350000	15%	502500	
5	4200000	15%	630000	
6	4700000	20%	940000	
7	2200000	10%	220000	
8		3212500	1995000	

Рисунок 3.3 – Пример использования функции СУММПРОИЗВ

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 3.1.

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 3.2.

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 воспользуемся функцией

СРЗНАЧА:

=СРЗНАЧА(С2:С5)⇒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(А1:А5, >100) – возвратит число ячеек, содержимое которых будет более 100.

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

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 3.3:

СЧЁТ(А1:А7) возвращает 3

СЧЁТ(А4:А7) возвращает 2

СЧЁТ(А1:А7, 2) возвращает 4

	А
1	Продажи
2	21.08.09
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 3.4:

СЧЕТЗ (А1:А7) возвращает 6

СЧЕТЗ (А4:А7) возвращает 4

СЧЕТЗ (А1:А7, 2) возвращает 7

	А
1	Продажи
2	21.08.09
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(значение1;значение2,...), МИНА(значение1;значение2,...)** – 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.

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 3.5:

Поиск данных в таблице (см. рисунок 3.4).

Таблица для выполнения поиска расположена в ячейках D2:F9. 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 3.4 – 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-ти) 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") равняется "2-ой"

СУММ(A1:ВЫБОР(3;A10;A20;A30)) равняется СУММ(A1:A30)

Если ячейка A10 содержит 4, то:

ВЫБОР(A10;"Гвозди";"Винты";"Гайки";"Болты") равняется "Болты"

Если A10-3 равняется 3, то:

ВЫБОР(A10-3;"1-ый";"2-ой";"3-ий";"Последний") равняется "3-ий"

Если ПрoшлыеПрoдaжи - это имя, ссылающееся на значение 10 000, то:

ВЫБОР(2;НовыеПрoдaжи;ПрoшлыеПрoдaжи;БюджетПрoдaж) равняется 10 000.

Пример 3.6. 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. рисунок 3.5).

3	A	B	C	D	E	F	G	H	I
4	Объем продаж	Комисс %		ФИО	Стаж	Объем продаж	Комисс. %	Сумма комисс	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 3.5 – Результаты расчета 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 Н5: =F5*G5/100.
- Скопируем эту формулу в ячейки Н6:Н14.
- Для определения суммы комиссионных с учетом стажа воспользуемся формулой, которую внесем в ячейку I5: =Н5+Н5*Е5/100.
- Скопируем эту формулу в ячейки I6:I14.

Вариант 2. С использованием функции ЕСЛИ (см. рисунок 3.6):

	К	Л	М	Н	О
			Объем продаж	Сумма комисс	С учетом стажа
4	ФИО	Стаж			
5	Тормозов	5	5000	400	420
6	Иванов	12	7000	560	627,2
7	Кругой	10	11000	1155	1270,5
8	Умный	1	15000	1575	1590,75
9	Простой	8	21500	2580	2786,4
10	Кошкин	4	25000	3000	3120
11	Фунтиков	6	12000	1260	1335,6
12	Зайчик	7	15000	1575	1685,25
13	Волк	5	39900	4788	5027,4
14	Баранов	4	45000	6300	6552

В ячейку N5 внесем формулу:

- =ЕСЛИ(И(M5>=0;M5<=9999,99);M5*0,08;ЕСЛИ(M5<=19999,99;M5*0,105;ЕСЛИ(M5<=39999,99;M5*0,12;M5*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;

Рисунок 3.6 – Результаты расчета комиссионных с использованием функции ЕСЛИ

5.5 Функции категории «Текстовые»

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

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.

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 3.7

▪ **ЗАМЕНИТЬ("абвгдежзийк";6,5;"*")** равняется **"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")** равняется **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.

▪ **T** – 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.

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. Числовой формат даты — это код дата-время, используемый в Microsoft Excel для вычислений с датами и периодами времени. 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,

накладывать 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 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.

7 Литература к лекции 3

1. Вардомацкая, Е. Ю. Информатика. В 2 ч. Ч. II. Excel : учебное пособие / Е. Ю. Вардомацкая, Т. Н. Окишева. – Витебск, 2007. – 237 с.
2. Вардомацкая, Е. Ю. Компьютерные информационные технологии: лабораторный практикум. Часть 1 для студентов экономических специальностей / Е. Ю. Вардомацкая. – Витебск, 2009. – 95 с.
3. Велихов, А. В. Основы информатики и компьютерной техники: учебное пособие для студентов ссузов и вузов по дисциплине "Основы информатики" / А. В. Велихов. – Москва: СОЛОН-ПРЕСС, 2007.
4. Информатика. Базовый курс: учебное пособие для студентов высших технических учебных заведений / под ред. С. В. Симоновича. – 2-е изд. – Санкт-Петербург: Питер, 2007. – 640 с.
5. Информатика для юристов и экономистов: учебник для вузов / под ред. С. В. Симоновича. – Санкт-Петербург: Питер, 2006.
6. Морозевич, А. Н. Прикладная информатика: учебное пособие / А. Н. Морозевич. – Минск: Выш. школа, 2003. – 335 с.: ил.

Лекция 4

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.
4. Литература.

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 визуализация;
- 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 [6].

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 A до Z и от a 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. 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	-->	– функциональный оператор
-	– оператор вычитания	<	– менее чем
*	– оператор умножение	>	– более чем
/	– деление	=	– равно
** , ^	– возведение в степень	<=	– менее чем и равно
!	– факториал	>=	– более чем и равно
.	– десятичная точка	<>	– неравно
:=	– оператор присваивания	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 расширений⁶. 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)`, `abs(x)`,
`ilog10`, `ilog` – целочисленные логарифмы (`ilog10(25)=1`);
`ln(x)`, `log[a](x)`, `log10(x)` – логарифмы по разному основанию.

Некоторые целочисленные функции:

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

Функции с элементами сравнения:

`ceil` – наименьшее целое $\geq x$;
`floor` – наибольшее целое $\leq x$;
`frac` – дробная часть числа x;
`trunc` – меньшее целое, округленное в направлении к нулю;

⁶ С перечнем пакетов расширений можно ознакомиться в справочной системе СКМ Maple.

round – округленное значение числа;

signum – функция знака (-1, 1);

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

Пример 4.1. 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 опций⁷, задающий стиль построения графика функции.

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

Пример 4.2. 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);
```

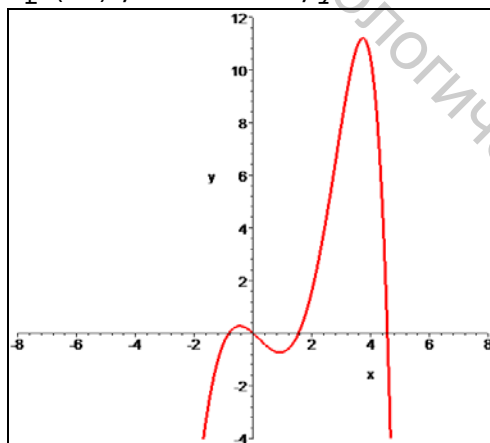


Рисунок 4.1 – График функции $y = x^3 + 1 - e^x$

Пример 4.3. 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.

⁷ Полный перечень опций для построения двумерных графиков приведен в справочной системе СКМ Maple.

Решение (см. рисунок 4.2):

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

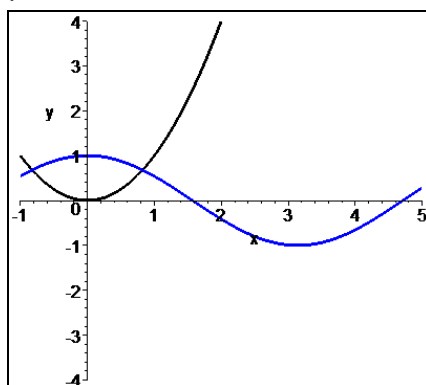


Рисунок 4.2 – Графики двух зависимостей на одной диаграмме

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 4.4. 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 (т. е. $x = xmin..xmax$, $y = ymin..ymax$).

Решение (см. рисунок 4.3):

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

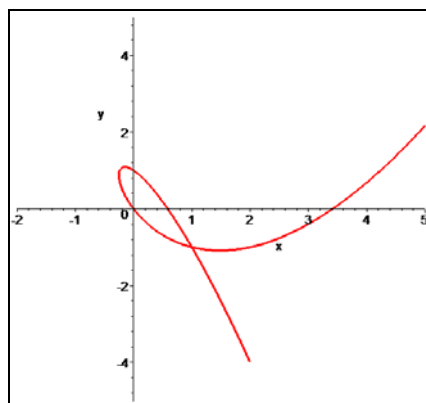


Рисунок 4.3 – График кривой, заданной параметрическими зависимостями

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 проекты 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 4.5. Построить поверхность h^2 в цилиндрической системе координат.

Решение (см. рисунок 4.4):

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

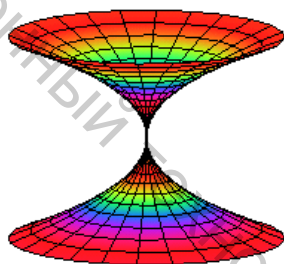


Рисунок 4.4 – Пример трехмерного графика

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 4.6. 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 4.7. 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 4.8. 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:

$\text{Int}(f, x)$ – отложенного действия, $\text{int}(f, x)$ – 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:

$\text{Int}(f, x=a..b, \text{continuous})$ – отложенного действия,

$\text{int}(f, x=a..b, \text{continuous})$ – 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 :

$\text{evalf}(\text{int}(f, x=a..b))$.

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

Пример 4.9. 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\$m, x3], ..., xj, [xk\$m])</code>	Производная n-го порядка функции n переменных $\frac{d^n}{dx_k^m dx_j \dots dx_3 dx_2 dx_1^n} f$

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`⁸ 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 4.10. Вычислить значение производной первого порядка

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

Решение :

⁸ Иначе функцию `diff` называют командой прямого действия, а функцию `Diff` – командой отложенного действия.

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

#Определяем функцию f:

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

> Diff(f, x) = diff(f, x);

#Вычисляем производную:

$$\frac{d}{dx} \left(\frac{x^{27}}{27} + 5x^{0.2} \right) = x^{26} + \frac{1.0}{x^{0.8}};$$

Пример 4.11. 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 4.12. Объем выпуска ковровых изделий, млн. м², 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:

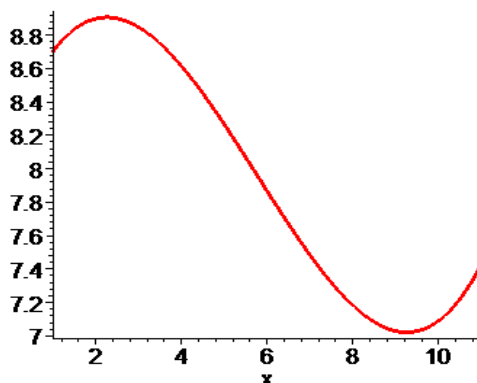
Определяем функцию y:

> 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-ой период) и 2005 (11 период) г.г.

```
> CH_2000 := subs(x=6, y);  
CH_2000 := 7.866
```

```
> CH_2005 := subs(x=11, y);  
CH_2005 := 7.431
```

4 Литература к лекции 4

1. Вардомацкая, Е. Ю. Информатика в прикладных задачах легкой промышленности: пособие / Е. Ю. Вардомацкая, Т. Н. Окишева. – Витебск, 2007. – 187 с.

2. Вардомацкая, Е. Ю. Информатика. В 2 ч. : учебное пособие / Е. Ю. Вардомацкая, Т. Н. Окишева. – Витебск, 2007. – 237 с.

3. Туркина, Е. П. Основы информатики и вычислительной техники: учебно-практическое пособие. Дистанционное обучение / Е. П. Туркина, Н. А. Кочетова. – Минск: БГЭУ, 2005.

4. Шарстнев, В. Л. Компьютерные информационные технологии: курс лекций / В. Л. Шарстнев. – Витебск: УО ВГТУ, 2008. – 350 с.

5. Шарстнев, В. Л. Компьютерные информационные технологии: лабораторный практикум : пособие / В. Л. Шарстнев, Е. Ю. Вардомацкая. Витебск: УО «ВГТУ», 2008. – 170 с.

6. http://ndo.sibsutis.ru/magistr/courses_work/ktnp_work1/lec9-11.htm.

7. Дьяконов, В. Maple : учебный курс / В. Дьяконов. – Санкт-Петербург: Питер, 2010. – 608 с. : ил.

Лекция 5

ПОНЯТИЕ БАЗЫ ДАННЫХ. ВВЕДЕНИЕ В СТРУКТУРИРОВАННЫЙ ЯЗЫК ЗАПРОСОВ SQL

План лекции

1. Язык SQL в СУБД.
2. Структура команды SQL. Типы данных. Выражения.
3. Описание команд языка SQL.
4. Литература.

1 Язык SQL в СУБД

База данных (БД) – это информационная модель объекта – именованная совокупность данных, отображающая состояние объектов, их свойства и взаимоотношения в некоторой предметной области.

Язык SQL (*structured query language – структурированный язык запросов*) представляет собой наиболее распространенный язык управления базами данных клиент/сервер. Он позволяет создавать не только реляционные базы данных, заполнять их, но и оперировать данными, хранящимися в этих базах. Причиной возникновения SQL является то, что в начале 70-х годов фирме IBM потребовался метод, с помощью которого простые пользователи могли извлекать из базы данных и выводить нужную им информацию. Языки, которые могут применять пользователи, называли *языками 4-го поколения* или *структурированным английским*.

Благодаря своей элегантности и независимости от специфики ЭВМ, а также поддержке всеми лидерами современных информационных технологий, **SQL** стал общемировым стандартом. Набор инструкций **SQL** можно использовать в любой СУБД, которая совместима с **SQL**. Стандарт **SQL** определен американским национальным институтом стандартов **ANSI** (American National Standarts Institute) и принят в качестве международного стандарта международной организацией стандартов **ISO** (International Standarts Organization) в 1986 г.

1989 год – ISO (Международный комитет по стандартизации) опубликовал первый международный стандарт языка SQL.

1992 год – версия SQL/92 или SQL2.

1999 год – стандарт SQL3, в который введены новые стандартные типы данных, появилась возможность конструирования сложных структурированных типов данных.

2003 г. – стандарт SQL:2003.

Каждый стандарт представляет собой многостраничный документ сложной структуры. К примеру, текст стандарта SQL-92 составляет около 600 страниц.

Язык SQL позволяет создавать реляционные базы данных, заполнять их и оперировать данными, хранящимися в этих базах.

SQL состоит из инструкций, которые передаются программе, управляющей работой БД, предлагая ей выполнить определенные действия. Эти инструкции в общем виде называются предложениями или командами. **SQL** – это язык, ориентированный специально на реляционные БД. Он позволяет исключить большую работу, выполняемую при использовании языков программирования общего назначения.

Существуют два **SQL**: интерактивный и встроенный.

Интерактивный **SQL** – применяется для выполнения действий непосредственно в БД с целью получить результат, который будет использован человеком.

Встроенный **SQL** состоит из команд **SQL**, включенных в программы, которые в большинстве случаев написаны на каком-то другом языке программирования.

2 Структура команды SQL. Типы данных. Выражения

Каждая команда SQL начинается с действия – *ключевого слова* или группы слов, описывающих выполняемую операцию. В языках программирования *ключевое слово* – название, зарезервированное для определенных целей, например, названий команд, устройств и т. п.

Например, INSERT (добавить), DELETE (удалить), COMMIT (завершить), CREATE TABLE (создать таблицу). После действия может следовать одно или несколько *предложений*.

Предложение описывает данные, с которыми работает команда, или содержит уточняющую информацию о действии, выполняемом командой. Каждое предложение начинается с ключевого слова, такого как, например, WHERE (где), FROM (откуда), INTO (куда), HAVING (имеющий). Многие предложения содержат имена таблиц и полей БД; некоторые – константы и выражения.

Имена таблиц, полей и пользователей должны содержать от 1 до 18 символов, начинаться с буквы и не содержать пробелов или специальных символов пунктуации. В качестве имен нельзя использовать ключевые слова SQL.

В языке SQL определен ряд типов, которые представлены в таблице 5.1.

Таблица 5.1 – Базовые типы данных SQL

Тип	Описание
CHARACTER(n) или CHAR(n)	Символьные строки длиной в n символов. Максимальная длина – 255 символов
DATE, TIME, TIMESTAMP	Дата, время
NUMERIC[(n,m)]	Точные числа, здесь n – общее количество цифр в числе, m – количество цифр слева от десятичной точки
DECIMAL[(n,m)] или DEC [(n,m)]	Точные числа, здесь n – общее количество цифр в числе, m – количество цифр слева от десятичной точки
INTEGER или INT	Целые числа от –2147483648 до 2147483647
SMALLINT	Целые числа от –32768 до 32767
FLOAT[(n)]	Числа большой точности, хранимые в форме с плавающей точкой. Здесь n – число байтов, резервируемое под хранение одного числа
REAL	Число с плавающей точкой и одинарной точностью от –3.402823E38 до 3.403823E38
DOUBLE PRECISION	Число с плавающей точкой и двойной точностью от –1.79769313486232E308 до 1.797...E308

3 Описание команд языка SQL

Язык SQL, соответствующий последним стандартам SQL:2003, SQL:1999, представляет собой очень богатый и сложный язык, все возможности которого трудно сразу осознать и тем более понять. Поэтому приходится разбивать язык на уровни. В одной из классификаций, предусмотренных стандартом SQL, этот язык разбивается на «базовый» (entry), «промежуточный» (intermediate) и «полный» (full) уровни. Базовый уровень содержит около сорока команд, которые можно сгруппировать в категории по их функциональному назначению.

Для примера создадим базу данных предметной области **Банк**, отражающую следующие сведения:

1. Вид вклада.
2. Код вклада.
3. Ставка %.
4. ФИО вкладчика.
5. Дата.
6. Сумма вклада.
7. % в месяц у. е.

Руководствуясь правилами нормализации, данные оформим в виде двух таблиц: в таблицу **Справочник** включим данные 1, 2 и 3, а в таблицу **Сведения** – 2 и 4-7. Предусмотрим не менее четырех видов вкладов, на каждом из которых хранят деньги не менее трех вкладчиков.

Таким образом, таблицы базы данных могут иметь вид:

Справочник

Вид вклада	Код	Ставка, %
Валютный депозит	ВД	12
Валютный депозит накопительный	ВДН	10
Валютный депозит срочный	ВДС	12
Валютный текущий	ТВ	3

Сведения

№	ФИО	Код вклада	Дата	Сумма вклада	% в месяц, у. е.
1	Сергеев И.	ВД	02.02.2013	5000	
2	Костин В.	ВД	05.01.2013	10000	
3	Суворов А.	ВД	10.01.2013	8000	
4	Петрова И.	ВДС	02.02.2013	3000	
5	Соколова Н.	ВДС	01.03.2013	12000	
6	Михайлова А.	ВДС	02.02.2013	7000	
7	Орлова И.	ВДН	02.02.2013	15000	
8	Богданова О.	ВДН	05.01.2013	12000	
9	Купцова С.	ВДН	05.01.2013	10000	
10	Архипов Н.	ТВ	04.03.2013	20000	
11	Разин А.	ТВ	02.02.2013	1000	
12	Пугачева А.	ТВ	02.02.2013	50000	

Для того, чтобы с таблицами можно было работать как с единым целым, они должны быть связаны. Связь в данном случае устанавливается по полям **[Код]** таблицы **Справочник** (поле первичного ключа) и **[Код вклада]** таблицы **Сведения** (внешний ключ), так как показано на рисунке 5.1.

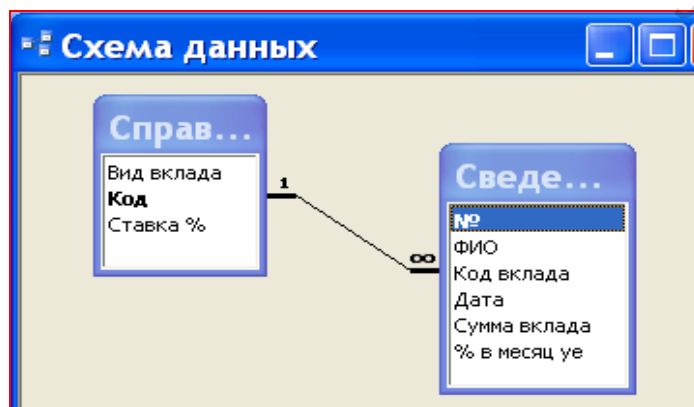


Рисунок 5.1 – Схема данных базы данных Банк

Рассмотрим команды SQL базового уровня по категориям.

К категории **Описание данных** относятся команды, позволяющие создавать, изменять и уничтожать БД и объекты БД (таблицы и представления).

- **CREATE** – предназначена для *создания* таблиц. Имя запроса не должно совпадать с именем создаваемой таблицы.

Формат:

```
CREATE TABLE <имя таблицы>  
(<имя столбца 1>< тип данных>[(<размер>)]),  
(<имя столбца 2>< тип данных>[(<размер>)]...);
```

Например, по команде

```
CREATE TABLE Справочник ([Вид вклада] CHAR(30), Код CHAR(5),  
[Ставка %] INT)) – в БД создается структура таблицы Справочник,  
включающая поля: Вид вклада, Код, Ставка %. Поля Вид вклада и Код имеют  
символьный тип с размерами 30 и 5 символа соответственно, поле Ставка %  
имеет целый тип.
```

- **ALTER** – предназначена для *изменения структуры* таблицы.

Формат команды, предназначенный для добавления столбца в таблицу:

```
ALTER TABLE <имя таблицы> ADD < имя столбца>  
<тип данных>[(<размер>)];
```

Например, по команде

```
ALTER TABLE Сведения ADD Адрес CHAR(30)
```

в структуру таблицы **Сведения** добавляется новое поле **Адрес**,
символьного типа размером 30 символов.

- **DROP** – предназначена для удаления одной или нескольких таблиц.

Формат:

```
DROP TABLE <имя таблицы>;
```

Например, по команде **DROP TABLE Справочник** – таблица
Справочник удаляется из БД.

К категории **Внесение изменений в БД** относятся команды, позволяющие добавлять, удалять и модифицировать данные в таблицах.

- Команда **INSERT** – добавляет новые записи в таблицу

Формат:

```
INSERT INTO <имя таблицы> [{<имя поля>}, ...]  
VALUES({<значение>},...)
```

Например, по команде

```
INSERT INTO Справочник
```

VALUES (“Текущий рублевый ”, "TP", 3) – в конец таблицы Справочник добавляется новая запись, в первое поле которой заносится значение “Текущий рублевый ”, во второе поле – значение "TP", в третье поле – 3.

- Команда **DELETE** – удаляет записи из таблицы.

Формат:

DELETE FROM <имя таблицы> [**WHERE** <условие>]

Например, по команде

DELETE FROM Сведения WHERE [Код вклада]=”ТВ” – из таблицы **Сведения** удаляются все записи, у которых в поле *Код вклада* хранится значение ТВ.

- Команда **UPDATE** – обновляет данные таблицы.

Формат:

UPDATE <имя таблицы> **SET**

{<имя поля>=<выражение>},... [**WHERE** <условие>]

В предложении **UPDATE** указывается имя обновляемой таблицы, в предложении **SET** указываются имена столбцов и выражение, изменяющее данные указанного поля. Часть **WHERE** является необязательной. Она позволяет отобрать строки, к которым будет применена модификация. Если условие отбора не задается, то операция модификации будет применена ко всем строкам таблицы.

Например, по команде

UPDATE Справочник SET [Ставка %] = [Ставка %] – 2 – значения процентных ставок по всем видам вкладов будут понижены на 2 процентных пункта.

А по команде

UPDATE Справочник SET [Ставка %] = [Ставка %] + 3

WHERE Код = “ВДС” – значение процентной ставки по вкладам вида ВДС будет повышено на 3 процентных пункта.

Для того, чтобы рассчитать значения поля [% в месяц уе] таблицы **Сведения** в зависимости от значения %-ой ставки, размещенной в таблице **Справочник**, необходимо команду **UPDATE** (запрос на обновление) составлять на основании двух таблиц, установив между ними связь.

UPDATE Справочник, Сведения SET [% в месяц уе] = [Ставка %]/100*[Сумма вклада]/12

WHERE Справочник.Код=Сведения.[Код вклада];

В результате выполнения этой команды произойдет автоматический расчет значений поля [% в месяц уе].

4 Литература к лекции 5

1. Бекаревич, Ю. Б. Microsoft Access за 21 занятие для студента / Ю. Б. Бекаревич, Н. В. Пушкина. – Санкт-Петербург: БХВ-Петербург, 2005. – 544 с.
2. Вардомацкая, Е. Ю. Компьютерные информационные технологии: лабораторный практикум. Часть 2 / Е. Ю. Вардомацкая, Н. А. Багрецова. – Витебск: УО «ВГТУ», 2010. – 75 с.
3. Вардомацкая, Е. Ю. «Технологии обработки экономической информации в среде MS Office и СКМ Maple»: методические указания и контрольные задания по курсу «Компьютерные информационные технологии» для студентов экономических специальностей заочной формы обучения / Е. Ю. Вардомацкая. – Витебск, 2010.
4. Информатика. Базовый курс : учебное пособие для студентов высших технических учебных заведений / под ред. С. В. Симоновича. – 2-е изд. – Санкт-Петербург: Питер, 2007. – 640 с.
5. Малыхина, М. П. Базы данных: основы, проектирование, использование: учебное пособие для студентов вузов, обучающихся по направлению подготовки "Информатика и вычислительная техника" / М. П. Малыхина. – Санкт-Петербург: БХВ-Петербург, 2006. – 528 с.
6. Советов, Б. Я. Базы данных. Теория и практика: учебник для студентов вузов, обучающихся по направлениям "Информатика и вычислительная техника" и "Информационные системы" / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – 2-е изд., стер. – Москва: Высшая школа, 2007. – 463 с.
7. Фрост, Р. Проектирование и разработка баз данных. Визуальный подход Database Design and Development / Р. Фрост, Д. Дей, Ван Слайк К. – Москва: NT Press, 2007. – 592 с.

Лекция 6

ТЕХНОЛОГИИ ВЕДЕНИЯ БАЗ ДАННЫХ И ВЫБОРА ДАННЫХ ИЗ БАЗЫ

План лекции

1. Оператор выбора данных SELECT.
2. Устранение избыточности данных.
3. Определение условия выбора WHERE.
4. Специальные операторы в условиях (IN, BETWEEN, LIKE).
5. Функции агрегирования. Параметры GROUP BY и HAVING.
6. Упорядочивание выходных полей.
7. Литература.

1 Оператор выбора данных SELECT

Категория Извлечение данных состоит из одной команды **SELECT**, являющейся основной, наиболее часто используемой командой языка SQL. Структура этой команды проста, потому что её можно расширять для того, чтобы выполнять очень сложные вычисления и обработку данных. Эта команда применяется при формировании всех запросов выбора. Запрос – команда, которая формулируется для СУБД и требует предоставить определённую указанную информацию. Эта информация обычно выводится на экран, может быть направлена на принтер, сохранена в файле или использована в качестве исходных данных для другой команды или процесса

Команда SELECT – извлекает данные из БД

Формат: **SELECT** <что выводится>

FROM <откуда> [**INTO** <куда>

WHERE<каким условиям должно отвечать>

GROUP BY <колонки(поле), по которым осуществляется группирование>

HAVING<условие группирования записей в одну строку>

ORDER BY<в каком порядке выводить данные>]

Для того, чтобы из таблицы выбрать определенные столбцы, в команде **SELECT** достаточно перечислить их имена.

Пример 6.1. Из таблицы **Сведения** выбрать сведения о клиентах.

```
SELECT ФИО, [Код вклада], [Сумма вклада]
FROM Сведения;
```

2 Устранение избыточных данных

Ключевое слово **DISTINCT** – аргумент, дающий возможность исключить дублирующие значения из результата выполнения предложения **SELECT**.

DISTINCT можно задать только один раз для данного **SELECT**. Если **SELECT** извлекает множество полей, то он исключает строки, в которых все выбранные поля идентичны. Строки, в которых некоторые значения одинаковы, а другие различны, он включает в результат.

Пример 6.2. Пользуясь таблицей **Сведения**, вывести список видов счетов, предоставляемых банком.

SELECT [Код вклада] FROM Сведения;	SELECT DISTINCT [Код вклада] FROM Сведения;																		
Результат	Результат																		
<table border="1"><thead><tr><th>Код вклада</th></tr></thead><tbody><tr><td>ВД</td></tr><tr><td>ВД</td></tr><tr><td>ВД</td></tr><tr><td>ВДС</td></tr><tr><td>ВДС</td></tr><tr><td>ВДС</td></tr><tr><td>ВДН</td></tr><tr><td>ВДН</td></tr><tr><td>ВДН</td></tr><tr><td>ТВ</td></tr><tr><td>ТВ</td></tr><tr><td>ТВ</td></tr></tbody></table>	Код вклада	ВД	ВД	ВД	ВДС	ВДС	ВДС	ВДН	ВДН	ВДН	ТВ	ТВ	ТВ	<table border="1"><thead><tr><th>Код вклада</th></tr></thead><tbody><tr><td>ВД</td></tr><tr><td>ВДН</td></tr><tr><td>ВДС</td></tr><tr><td>ТВ</td></tr></tbody></table>	Код вклада	ВД	ВДН	ВДС	ТВ
Код вклада																			
ВД																			
ВД																			
ВД																			
ВДС																			
ВДС																			
ВДС																			
ВДН																			
ВДН																			
ВДН																			
ТВ																			
ТВ																			
ТВ																			
Код вклада																			
ВД																			
ВДН																			
ВДС																			
ТВ																			

3 Определение условия выбора WHERE

Предложение **WHERE** команды **SELECT** позволяет определить предикат (условие), который может быть либо истинным, либо ложным для каждой строки таблицы. Команда извлекает только те строки из таблицы, для которых предикат имеет значение «истина». При наличии предложения **WHERE** программа обработки БД просматривает таблицу строка за строкой и для каждой строки проверяет, истинен ли предикат.

Формат: WHERE критерий поиска,

где *критерий поиска* является выражением, включающим текстовый оператор сравнения для текстовых полей или арифметический оператор сравнения для числовых полей. В критерии поиска могут использоваться реляционные и булевы операторы.

Реляционный оператор – это математический символ, который задает определенный тип сравнения между двумя значениями: = (равно), >= (больше или равно), > (больше, чем), <= (меньше или равно), < (меньше, чем), <> (не равно).

Булевы операторы (выражения) – это те выражения, относительно которых, подобно предикатам, можно сказать истинны они или ложны. Булевы операторы связывают одно или несколько значений (истина/ложь) и в результате получают единственное значение (истина/ложь).

Стандартные булевы операторы: AND, OR, NOT.

В сложных булевых выражениях можно использовать скобки. SQL понимает круглые скобки так: всё то, что расположено внутри круглых скобок, вычисляется прежде всего и рассматривается как единственное выражение по отношению к тому, что расположено за пределами круглых скобок.

Пример 6.3. Выбрать сведения о клиентах, у которых на текущем счете размещена сумма не меньше 10000 у.е.

```
SELECT ФИО, [Код вклада], [Сумма вклада]
FROM Сведения
WHERE [Код вклада]="ТВ" AND [Сумма вклада]>10000;
```

ФИО	Код вклада	Сумма вклада
Архипов Н.	ТВ	20000
Пугачева А.	ТВ	50000

Кроме того, в предложении WHERE может указываться условие связывания таблиц, если выбор производится не из одной таблицы, а из нескольких. Таким образом, SQL-запрос обеспечивает возможность определять связи между множеством таблиц и отображать содержащуюся в них информацию в терминах этих связей в рамках единственной команды. Операции такого рода называются соединением (join) и являются одним из самых мощных преимуществ реляционных БД.

При операции соединения имена таблиц перечисляются через запятую в предложении FROM. Предикат (условие) запроса может ссылаться на любой столбец любой из соединяемых таблиц и использоваться для установления связей между ними. Записи из двух таблиц объединяются при обнаружении совпадающих значений в объединяемых полях. При этом поля должны иметь одинаковый тип данных, но могут иметь разные имена. Обычно предикат

сравнивает значения в столбцах различных таблиц для того, чтобы определить, выполняется ли условие **WHERE**.

Пример 6.4. Выбор данных из двух таблиц.

Вывести информацию о вкладчиках с полным названием вида вклада. Выводимым полям присвоить собственные имена.

```
SELECT ФИО AS [ФИО вкладчика], [Сумма вклада], [Вид вклада] AS
[Название вклада]
FROM Справочник, Сведения
WHERE Справочник.Код = Сведения.[Код вклада];
```

ФИО вкладчика	Сумма вклада	Название вклада
Сергеев И.	5000	Валютный депозит
Костин В.	10000	Валютный депозит
Суворов А.	8000	Валютный депозит
Петрова И.	3000	Валютный депозит срочный
Соколова Н.	12000	Валютный депозит срочный
Михайлова А.	7000	Валютный депозит срочный
Орлова И.	15000	Валютный депозит накопительный
Богданова О.	12000	Валютный депозит накопительный
Купцова С.	10000	Валютный депозит накопительный
Архипов Н.	20000	Валютный текущий
Разин А.	1000	Валютный текущий
Пугачева А.	50000	Валютный текущий

4 Специальные операторы в условиях (**IN**, **BETWEEN**, **LIKE**)

Оператор IN. Определяет множество, элементы которого перечисляются в круглых скобках и разделяются запятыми. Если в поле, имя которого указано слева от **IN**, есть одно из перечисленных в списке значений (требуется точное совпадение), то предикат считается истинным.

Пример 6.5. Выбрать клиентов, у которых на счетах размещены суммы в 5 или 10 тыс. у. е.

1 способ

2 способ

SELECT ФИО, [Код вклада], [Сумма вклада] FROM Сведения WHERE [Сумма вклада]=5000 OR [Сумма вклада]=10000;	SELECT ФИО,[Код вклада], [Сумма вклада] FROM Сведения WHERE [Сумма вклада] In (5000,10000);
---	--

Оператор BETWEEN. Сходен с **IN**. Вместо перечисления элементов в множестве, как это делается в **IN**, **BETWEEN** задает границы, в которые должно попадать значение, чтобы предикат был истинным. Используется ключевое слово **BETWEEN**, за которым следует начальное значение, ключевое слово **AND** и конечное значение. **BETWEEN** требует: первое значение в предложении должно быть первым в соответствии с алфавитным или числовым порядком.

Оператор **BETWEEN** является включающим, т. е. граничные условия входят в диапазон и делают предикат истинным. Для того, чтобы исключить граничные условия, можно к примеру воспользоваться следующим выражением:

Пример 6.6. Выбрать сведения о клиентах, у которых на вкладе размещена сумма от 5000 до 10000 у. е.

```
SELECT ФИО, [Код вклада], [Сумма вклада]
FROM Сведения
WHERE [Сумма вклада] Between 5000 And 10000;
```

ФИО	Код вклада	Сумма вклада
Сергеев И.	ВД	5000
Костин В.	ВД	10000
Суворов А.	ВД	8000
Петрова И.	ВДС	3000
Михайлова А.	ВДС	7000
Купцова С.	ВДН	10000

Оператор LIKE. **LIKE** применим только к полям типа **CHAR**, поскольку он используется для поиска подстрок. Этот оператор осуществляет просмотр строки для выяснения, входит ли заданная подстрока в указанное поле. С этой целью используются шаблоны: символ «подчеркивания» (_), символ процент (%) – специальные символы, которые могут заменить собой определенные комбинации из других символов. В СУБД Access используется один символ-заменитель «звездочка» (*).

Пример 6.7. Выбрать сведения о клиентах, заключивших договора 02.02.13, фамилии которых начинаются на букву П.

```
SELECT ФИО, Дата, [Сумма вклада]
FROM Сведения
WHERE ФИО Like "П*" AND Дата=#2/2/2013#;
```

ФИО	Дата	Сумма вклада
Петрова И.	02.02.2013	3000
Пугачева А.	02.02.2013	50000

5 Функции агрегирования. Предложения GROUP BY и HAVING

Запросы могут обобщать не только группы значений, но и значения одного поля. Для этого применяются агрегатные функции. Они дают единственное значение для целой группы строк таблицы.

В стандарте языка SQL определено несколько функций агрегирования:

- **COUNT** – определяет количество строк или значений поля, выбранных посредством запроса;
- **SUM** – вычисляет сумму всех выбранных значений данного поля;
- **AVG** – вычисляет среднее арифметическое выбранных значений данного поля;
- **MAX** – вычисляет наибольшее из всех выбранных значений данного поля;
- **MIN** – вычисляет наименьшее из всех выбранных значений данного поля.

Функции агрегирования используются как имена полей в предложении запроса SELECT, причем имена полей применяются как аргументы.

Функции SUM, AVG применимы только к числовым полям. Функции COUNT, MAX, MIN могут использоваться для числовых и символьных полей.

Пример 6.8. Проанализировать популярность у вкладчиков видов вкладов предоставляемых банком.

```
SELECT [Код вклада], Sum([Сумма вклада]) AS [Сумма вкладов],  
Avg([Сумма вклада]) AS [Среднее значение вклада], Count([Сумма  
вклада]) AS [Количество]  
FROM Сведения  
GROUP BY [Код вклада];
```

Код вклада	Сумма вкладов	Среднее значение вклада	Количество
ВД	23000	7666,67	3
ВДН	37000	12333,33	3
ВДС	22000	7333,33	3
ТВ	71000	23666,67	3

Здесь группировка данных с вычислением итогов производится по одному полю [Код вклада].

Каждая группа состоит из всех тех строк, которые имеют одно и то же значение поля [Код вклада], а функции SUM, AVG и COUNT применяются отдельно к каждой такой группе. Это означает, что поле, к которому применяется GROUP BY, по определению имеет на выходе только одно значение на каждую группу, что соответствует применению агрегатных функций. Таким образом, все записи базы, для которых значения полей совпадают, отображаются в выборке единственной строкой.

Возможна группировка по нескольким полям.

Пример 6.9. Проанализировать популярность у вкладчиков видов вкладов, предоставляемых банком. Вывести полную информацию по видам вкладов.

```
SELECT [Вид вклада], [Ставка %] AS [Процентная ставка], Sum([Сумма вклада]) AS [Общая сумма вкладов], Avg([Сумма вклада]) AS [Среднее значение вклада]
FROM Справочник, Сведения
WHERE (((Справочник.Код)=[Сведения].[Код вклада]))
GROUP BY [Вид вклада], [Ставка %];
```

Вид вклада	Процентная ставка	Общая сумма вкладов	Среднее значение вклада
Валютный депозит	12	23000	7666,67
Валютный депозит накопительный	10	37000	12333,33
Валютный депозит срочный	12	22000	7333,33
Валютный текущий	3	71000	23666,67

Пример 6.10. Проанализировать вклады за истекшие месяцы.

В этом случае нужно сгруппировать данные по дате, выделив из дат параметр «месяц».

Для работы с данными типа Дата/время SQL располагает специальными функциями:

Day – выделяет число – номер дня месяца введенной даты.

Month – выделяет число – номер месяца даты.

Year – выделяет число – номер года даты.

Date() – возвращает текущую дату.

```
SELECT Month(Дата) AS Месяц, Count([Сумма вклада]) AS [Количество вкладов], Avg([Сумма вклада]) AS [В среднем на вкладе], Sum([Сумма вклада]) AS [Итого за месяц]
FROM Сведения
GROUP BY Month(Дата);
```

Месяц	Количество вкладов	В среднем на вкладе	Итого за месяц
1	4	10000	40000
2	6	13500	81000
3	2	16000	32000

Пример 6.11. Проанализировать вклады за январь 2013 г.

В этом случае нужно сгруппировать данные по дате, а затем из получившихся групп выбрать ту, у которой значение даты = 1, что соответствует месяцу январь.

Функцию отбора данных из групп выполняет параметр **HAVING**. Он определяет критерий, согласно которому определенные группы исключаются из числа выходных данных так же, как параметр **WHERE** делает это для отдельных строк. **HAVING** является необязательным параметром, но если он задан, то должен следовать за параметром **GROUP BY**.

```
SELECT Month(Дата) AS Месяц, Count([Сумма вклада]) AS
[Количество вкладов], Avg([Сумма вклада]) AS [В среднем на вкладе],
Sum([Сумма вклада]) AS [Итого за январь]
FROM Сведения
GROUP BY Month(Дата)
HAVING (Month(Дата)=1);
```

Месяц	Количество вкладов	В среднем на вкладе	Итого за январь
1	4	10000	40000

Пример 6.12. Проанализировать вклады, открытые в период с января по март 2013 г.

```
SELECT Month(Дата) AS Месяц, Sum([Сумма вклада]) AS [В среднем за
месяц], Min([Сумма вклада]) AS [Min вклад], Max([Сумма вклада]) AS [Max
вклад]
FROM Сведения
GROUP BY Month(Дата)
HAVING (Month([Дата]) Between 1 And 3);
```

Месяц	В среднем за месяц	Min вклад	Max вклад
1	40000	8000	12000
2	81000	1000	50000
3	32000	12000	20000

6 Упорядочивание выходных полей

Таблицы базы данных являются неупорядоченными множествами, и выбираемые из них данные необязательно представлены в какой-либо определенной последовательности.

Команда **ORDER BY** позволяет упорядочить выходные данные запроса в соответствии со значениями одного или нескольких выбранных столбцов.

Можно задать возрастающую (**ASC**) или убывающую (**DESC**) последовательность сортировки для каждого из столбцов. По умолчанию установлена сортировка по возрастанию, т. е. параметр ASC можно не указывать. Внутри уже упорядоченной по одному полю таблицы можно провести упорядочение и по другому столбцу.

Пример 6.13. Вывести сведения о вкладчиках, упорядочив выходной набор по полю Дата по возрастанию, а по полю Сумма вклада – по убыванию:

```
SELECT Дата, [Сумма вклада], ФИО
FROM Сведения
ORDER BY Дата, [Сумма вклада] DESC;
```

Дата	Сумма вклада	ФИО
05.01.2013	12000	Богданова О.
05.01.2013	10000	Купцова С.
05.01.2013	10000	Костин В.
10.01.2013	8000	Суворов А.
02.02.2013	50000	Пугачева А.
02.02.2013	15000	Орлова И.
02.02.2013	7000	Михайлова А.
02.02.2013	5000	Сергеев И.
02.02.2013	3000	Петрова И.
02.02.2013	1000	Разин А.
01.03.2013	12000	Соколова Н.
04.03.2013	20000	Архипов Н.

7 Литература к лекции 6

1. Бекаревич, Ю. Б. Microsoft Access за 21 занятие для студента / Ю. Б. Бекаревич, Н. В. Пушкина. – Санкт-Петербург: БХВ-Петербург, 2005. – 544 с.

2. Вардомацкая, Е. Ю. Компьютерные информационные технологии: лабораторный практикум. Часть 2 / Е. Ю. Вардомацкая, Н. А. Багрецова. – Витебск: УО «ВГТУ», 2010. – 75 с.

3. Вардомацкая, Е. Ю. «Технологии обработки экономической информации в среде MS Office и СКМ Maple»: методические указания и контрольные задания по курсу «Компьютерные информационные технологии» для студентов экономических специальностей заочной формы обучения / Е. Ю. Вардомацкая. – Витебск, 2010.

4. Информатика. Базовый курс : учебное пособие для студентов высших технических учебных заведений / под ред. С. В. Симоновича. – 2-е изд. – Санкт-Петербург: Питер, 2007. – 640 с.

5. Малыгина, М. П. Базы данных: основы, проектирование, использование: учебное пособие для студентов вузов, обучающихся по направлению подготовки "Информатика и вычислительная техника" / М. П. Малыгина. – Санкт-Петербург: БХВ-Петербург, 2006. – 528 с.

6. Советов, Б. Я. Базы данных. Теория и практика: учебник для студентов вузов, обучающихся по направлениям "Информатика и вычислительная техника" и "Информационные системы" / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. – 2-е изд., стер. – Москва: Высшая школа, 2007. – 463 с.

7. Фрост, Р. Проектирование и разработка баз данных. Визуальный подход Database Design and Development / Р. Фрост, Д. Дей, Ван Слайк К. – Москва: NT Press, 2007. – 592 с.

Лекция 7

ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ ДЛЯ МОДЕЛИРОВАНИЯ И ЧИСЛЕННОГО РЕШЕНИЯ ЗАДАЧ ЭКОНОМИКИ И УПРАВЛЕНИЯ

План лекции

1. Основные понятия и задачи моделирования.
2. Модули стандартных офисных пакетов для работы с корреляционно-регрессионными моделями.
3. Анализ и решение задач межотраслевого баланса.
4. Анализ и решение задач оптимизации плана производства продукции.
5. Анализ и решение задач оптимизации плана транспортных перевозок.
6. Литература.

1 Основные понятия и задачи моделирования

Поведение любой системы (экономической, технологической, социальной и т. п.) зависит от множества факторов. Чем больше факторов, влияющих на поведение системы, выявлено, тем более точно (достоверно) можно регулировать и предсказывать ее поведение.

Процесс выявления факторов, влияющих на поведение системы, установления степени и характера взаимосвязи и влияния факторов друг на друга и на поведение системы в целом и есть **моделирование**.

Таким образом, **модель** – это упрощенное описание поведения реальной системы.

Экономико-математическая модель – математическое описание экономического процесса или явления, произведенное в целях его исследования.

В общем виде любая модель представляет собой зависимость вида

$$Y=f(x),$$

где x – набор независимых параметров, оказывающих влияние на поведение модели,

Y – выходной параметр (зависимая переменная), изменяющая свое значение при изменении независимых параметров.

Установление связи между зависимыми и независимыми параметрами и есть задача моделирования. Сформулировав модель, можно прогнозировать

поведение системы.

Для расчета любой модели необходима информационная база и возможность использования определенных методов моделирования.

В зависимости от направления и сути решаемых задач при экономическом анализе можно выделить следующие виды моделей:

- эконометрические (регрессионные) – в основе расчета таких моделей лежит теория вероятности и математическая статистика,
- балансовые – методы матричного исчисления,
- финансовые – методы финансового анализа,
- оптимизационные – методы математического программирования,
- модели исследования операций – теория вероятности,
- имитационные – теория вероятности и математическая статистика.

Любая модель всегда будет отличаться от реального поведения рассматриваемой системы. Это может быть связано

- со спецификацией (видом) модели,
- с невозможностью учета всех влияющих факторов,
- с вмешательством случайных величин (рисков),
- с влиянием человеческого фактора.

Расчет любой экономической модели и построение прогнозов является процессом достаточно трудоемким, основанным на большом количестве вычислений. Поэтому в настоящее время для этих целей используются пакеты прикладных программ (ППП) и возможности компьютерного моделирования.

ППП для моделирования и анализа задач экономики можно разбить на 2 типа:

- использующие численные методы – пакеты общего назначения (из семейства MS Office (OpenOffice, MS Office, NeoOffice, Numbers и др.),
- использующие точные аналитические решения – аналитические пакеты специального назначения (Statistica, STATA, SPSS и др.).

Возможности стандартных офисных пакетов для моделирования и численного решения типовых задач экономики и управления рассмотрим на примере табличного процессора MS Excel.

2 Модули стандартных офисных пакетов для работы с корреляционно-регрессионными моделями

При построении (расчете) эконометрических (регрессионных) моделей задачами моделирования являются⁹:

- выявление и расчет тесноты и вида связи между зависимыми и независимыми параметрами;
- расчет уравнения регрессии;

⁹ Теоретические основы корреляционно-регрессионного анализа подробно рассматриваются в курсе «Статистика».

- оценка адекватности модели;
- выполнение прогноза проведения экономической системы.

В ТП MS Excel существует возможность моделирования задач экономики на базе следующих технологий.

- С использованием стандартных функций категории Статистические. Встроенные функции ТП MS Excel этой категории с достаточной степенью вероятности позволяют рассчитывать и оценивать модели только двух видов: линейную и экспоненциальную.

- Графическим способом. При помощи графических средств можно получить прогнозы на будущие периоды для полиномиальной, логарифмической и степенной модели. Но рассчитать эти модели и оценить их адекватность по всем статистическим критериям в этом случае не представляется возможным.

- С использованием специальных надстроек «Пакет анализа» и «Поиск решения».

Таким образом, в ТП MS Excel существует возможность расчета и анализа регрессионных моделей двух видов: **линейных** и **экспоненциальных**. Некоторые другие виды моделей – степенная, логарифмическая, полиномиальная – могут быть получены графически.

В практической работе наибольшее распространение получили модели линейной зависимости. Все остальные виды нелинейных зависимостей путем стандартных математических преобразований рекомендуется приводить к линейным.

Линейная однофакторная модель – это уравнение прямой на плоскости, которая может быть описана уравнением

$$Y = m \cdot x + b, \quad (7.1)$$

где m – коэффициент уравнения регрессии, представляющий собой тангенс угла наклона прямой $Y = f(x)$ к оси OX ;

b – свободный член, равный значению точки пересечения прямой $Y=f(x)$ с осью OY .

Линейная многофакторная модель представляет собой уравнение прямой в многомерном пространстве и имеет вид:

$$Y = b + m_1 \cdot x_1 + m_2 \cdot x_2 + \dots + m_n \cdot x_n. \quad (7.2)$$

Экспоненциальная однофакторная модель имеет вид

$$Y = b \cdot m^x. \quad (7.3)$$

Экспоненциальная многофакторная модель имеет вид

$$Y = b \cdot m_1 x_1 \cdot m_2 x_2 \cdot \dots \cdot m_n x_n. \quad (7.4)$$

Здесь

Y – зависимая переменная,

x_1, \dots, x_n – независимые переменные (факторы),

m_0, \dots, m_n – коэффициенты уравнения регрессии,

b – свободный член.

Для расчета **линейных моделей** Excel располагает рядом функций, работа которых основана на методе наименьших квадратов: ЛИНЕЙН, НАКЛОН, ОТРЕЗОК, ПРЕДСКАЗ, ТЕНДЕНЦИЯ. Регрессионный анализ предполагает не только расчет параметров модели, но и оценку ее адекватности по некоторым статистическим характеристикам. Часть этих характеристик можно получить как результат работы некоторых стандартных функций, например, ЛИНЕЙН, КОРРЕЛ. Для проведения полного корреляционно-регрессионного анализа Excel предлагает специальную надстройку *Пакет Анализа*.

Рассмотрим поочередно перечисленные функции.

▪ **Функция ЛИНЕЙН** – рассчитывает статистику ряда с применением метода наименьших квадратов для вычисления уравнения прямой линии, которое наилучшим образом описывает исходные данные. Результатом работы функции является массив, который описывает полученную теоретическую прямую. Эта функция может использоваться как для расчета однофакторных, так и многофакторных моделей (что определяется размером массива независимых переменных x_i), кроме того, в качестве результата, кроме коэффициентов уравнения регрессии, можно получить и статистические характеристики, характеризующие построенную модель.

Формат:

ЛИНЕЙН(известные значения Y , известные значения X , константа; статистика),

где *известные значения Y* – это известные значения Y , для которых параметры X по уравнению определены.

Известные значения X – это известные значения независимой переменной X . Массив *известные значения X* может быть многомерным в отличие от массива *известные значения Y* , который является одномерным. Массив X может быть опущен, тогда значения X устанавливаются автоматически как предварительный ряд чисел, начиная с 1. Но обязательно должно быть соответствие между размерностями массива X и Y , если массив X задан.

Константа – это логическое значение, которое указывает функции, каким образом должен быть определен коэффициент b . Если логическое значение ИСТИНА или оно опущено, то b определяется в обычном порядке. Если константа равна ЛОЖЬ, то коэффициенты подбираются таким образом, чтобы выполнялось равенство $y = m \cdot x$ ($b=0$).

Статистика – логическое значение, которое может принимать значение ИСТИНА или ЛОЖЬ. Если статистика имеет значение ИСТИНА, то будет представлена дополнительная регрессионная статистика по регрессии, если ЛОЖЬ или опущено, то выходным массивом будет основная статистика, т. е. коэффициенты m_1, m_2, \dots, m_n и b .

В зависимости от количества независимых переменных уравнение получаемой теоретической прямой может иметь вид (7.1) для однофакторной

модели или (7.2) для многофакторной модели.

В качестве результата функция **ЛИНЕЙН** возвращает массив коэффициентов уравнения регрессии и дополнительную статистику по регрессии, как показано в таблице 7.1:

Таблица 7.1 – Результаты, возвращаемые функцией **ЛИНЕЙН**

m_n	m_{n-1}	...	m_2	m_1	b
Se_n	Se_{n-1}	...	Se_2	Se_1	Se_b
r^2	Se_y				
F	d_f				
SS_{reg}	SS_{resid}				

Здесь $m_1, m_2 \dots m_n, b$ – коэффициенты уравнения регрессии.

Все остальное – дополнительная статистика по регрессии:

Se_1, Se_2, Se_n – стандартные ошибки для коэффициентов $m_1, m_2 \dots m_n$

;

Se_b – стандартная ошибка для свободного члена b ;

R^2 – коэффициент детерминированности, который показывает, как близко теоретическое уравнение описывает исходные данные;

Se_y – стандартная ошибка для Y ;

F – критерий Фишера используется для определения того, является ли наблюдаемая взаимосвязь между зависимой и независимой переменными случайной или нет;

Df – степень свободы системы (уравнение надежности);

SS_{reg} – регрессионная сумма квадратов;

SS_{resid} – остаточная сумма квадратов.

Для выполнения прогноза нового значения Y в полученное уравнение нужно в уравнение регрессии подставить новое значение x .

▪ **Функция ПРЕДСКАЗ** - на основании линейного тренда вычисляет или предсказывает будущее значение зависимой переменной Y , соответствующее заданному X -значению, по существующим X - и Y -значениям.

Формат:

ПРЕДСКАЗ(X ; известные значения Y ; известные значения X),

где X – это точка данных, для которой предсказывается значение;

известные значения Y – это зависимый массив или интервал данных;

известные значения X – это независимый массив или интервал данных.

▪ **Функция ТЕНДЕНЦИЯ** – на основании линейного тренда вычисляет или предсказывает будущее значение зависимой переменной Y , соответствующее заданному массиву X -значений по существующим X - и Y -значениям.

В отличие от функции **ПРЕДСКАЗ**, функция **ТЕНДЕНЦИЯ** позволяет рассчитать теоретические значения Y для массива новых значений X .

Формат:

ТЕНДЕНЦИЯ(*известные значения Y* ; *известные значения X* ; *новые значения X* ; *константа*),

где *известные значения Y* – это зависимый массив или интервал данных; *новые значения X* – массив (или интервал данных), который должен содержать столбец (или строку) для каждой независимой переменной, как и *известные значения X* . Если аргумент *новые значения X* опущен, то предполагается, что он совпадает с аргументом *известные значения X* ;

константа – логическое значение, которое указывает, требуется ли, чтобы константа b была равна 0.

Пример 7.1. В таблице 7.2 приведены показатели хозяйственной деятельности предприятия легкой промышленности. Требуется построить многофакторную регрессионную модель и провести анализ зависимости производительности труда (Y) от следующих факторов:

- X_1 – удельный вес рабочих в составе ППП;
- X_2 – коэффициент сменности оборудования;
- X_3 – премии и вознаграждения на одного рабочего;
- X_4 – среднегодовая стоимость ОПС;
- X_5 – фондовооруженность труда.

Таблица 7.2 – Показатели хозяйственной деятельности предприятия.

	Производительность труда	Удельный вес покупных изделий	Коэффициент сменности оборудования	Премии и вознаграждения на одного рабочего	Среднегодовая стоимость ОПС	Фондовооруженность труда млн. руб.
№	Y	x_1	x_2	x_3	x_4	x_5
1	9,26	0,40	1,37	1,23	167,69	6,40
2	9,38	0,26	1,49	1,04	186,10	7,80
3	12,11	0,40	1,44	1,80	220,45	9,76

29	10,02	0,54	1,22	0,03	36,96	4,02
30	8,16	0,4	1,28	0,99	91,43	5,23
31	?	0,35	1,25	1,35	95	7,5

Решение:

Таблица с исходными данными размещена в ячейках A3:G33 рабочего листа Excel (см. рисунок 7.1).

В ячейках С34–G34 внесены значения показателей x_1-x_5 , для которых требуется выполнить прогноз уровня производительности труда (Y).

Для расчета линейной регрессионной многофакторной модели вида (7.2) в ячейки I5 :N9 введена функция ЛИНЕЙН в формате:

=ЛИНЕЙН(В4:В33;С4:G33;1;1)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
1	Многофакторная регрессионная модель																
2	№	Y	x1	x2	x3	x4	x5		ЛИНЕЙНАЯ МОДЕЛЬ							Y=9,82x1+5,39x2+1x3-0,00074x4+0,33x5-4,89	
3	1	9,26	0,40	1,37	1,23	167,69	6,40		m5	m4	m3	m2	m1	b			
4	2	9,38	0,26	1,49	1,04	186,10	7,80		0,3377	-0,0007	1,0041	5,3912	9,8228	-4,894			
5	3	12,11	0,40	1,44	1,80	220,45	9,76		0,12269	0,0033	0,46188	2,003	1,7075	2,55027			
6	4	10,81	0,50	1,42	0,43	169,30	7,90		0,79509	1,04052	#Н/Д	#Н/Д	#Н/Д	#Н/Д			
7	5	9,35	0,40	1,35	0,88	39,53	5,35		18,6244	24	#Н/Д	#Н/Д	#Н/Д	#Н/Д			
8	6	9,87	0,19	1,39	0,57	40,41	9,90		100,821	25,9844	#Н/Д	#Н/Д	#Н/Д	#Н/Д			
9	7	8,17	0,25	1,16	1,72	102,96	4,50										
10	8	9,12	0,44	1,27	1,70	37,02	4,88	Fтабл=	2,62065	Y=1.44*3.52^x1*2.11^x2*1.156^x3*0.99^x4*1/37^x5							
11	9	5,88	0,17	1,16	0,84	45,74	3,46		ЭКСПОНЕНЦИАЛЬНАЯ МОДЕЛЬ								
12	11	6,22	0,33	1,13	0,82	45,44	3,56		m5	m4	m3	m2	m1	b			
13	12	5,49	0,25	1,10	0,84	41,08	5,65		1,03695	0,99967	1,15626	2,11063	3,52864	1,44371			
14	13	6,50	0,32	1,15	0,67	136,14	4,28		0,01795	0,00048	0,06758	0,29309	0,24985	0,37317			
15	14	6,61	0,02	1,23	1,04	42,39	8,85		0,72396	0,15225	#Н/Д	#Н/Д	#Н/Д	#Н/Д			
16	15	4,32	0,06	1,39	0,66	37,39	8,52		12,5886	24	#Н/Д	#Н/Д	#Н/Д	#Н/Д			
17	16	7,37	0,15	1,38	0,86	101,78	7,19		1,4591	0,55635	#Н/Д	#Н/Д	#Н/Д	#Н/Д			
18	17	7,02	0,08	1,35	0,79	47,55	4,82										
19	29	10,02	0,54	1,22	0,03	36,96	4,02										
20	30	8,16	0,4	1,28	0,99	91,43	5,23										
21	31	9,0997	0,35	1,25	1,35	95	7,5										
22	ТЕНДЕНЦИЯ	9,0997															
23		8,83547															
24	РОСТ	8,83547															

Рисунок 7.1 – Таблица с исходными данными и результатами расчетов

Результатом работы функции ЛИНЕЙН является массив значений размера 5 x 6 (т. к. 5 переменных), представленный на рисунке 7.2.

	H	I	J	K	L	M	N	O	P	Q	
1											
3		ЛИНЕЙНАЯ МОДЕЛЬ					Y=9,82x1+5,39x2+1x3-0,00074x4+0,33x5-4,89				
4		m5	m4	m3	m2	m1	b				
5		0,337658	-0,00074	1,004112	5,391192	9,822783	-4,89449				
6		0,12269	0,0033	0,461878	2,003002	1,7075	2,550268				
7		0,795085	1,04052	#Н/Д	#Н/Д	#Н/Д	#Н/Д				
8		18,62437	24	#Н/Д	#Н/Д	#Н/Д	#Н/Д				
9		100,8214	25,98437	#Н/Д	#Н/Д	#Н/Д	#Н/Д				

Рисунок 7.2 – Результат расчета линейной модели

- В ячейках I5 – N5 – находятся коэффициенты уравнения регрессии m_1-m_5 и свободный член b ;
- в ячейке I7 – коэффициент детерминированности $R^2 = 0.795$;
- в ячейке I8 – критерий Фишера $F=18.62$.

Таким образом, многофакторная регрессионная модель, оценивающая влияние коэффициента сменности оборудования, премий и вознаграждений

на одного рабочего, среднегодовой стоимости ОПС, фондовооруженности труда на производительность труда, имеет вид:

$$Y=9,82 \cdot x_1+5,39 \cdot x_2+1 \cdot x_3-0,00074 \cdot x_4+0,33 \cdot x_5-4,89$$

Вывод: поскольку коэффициент детерминированности $R^2=0.795$ лежит в пределах $0,75 - 1$, расчетное значение критерия Фишера $F=18,6$ больше табличного ($=F_{\text{РАСПОБР}}(0,05; 5; 18) = 2,621$), модель следует признать адекватной, а значит с достаточной степенью точности ее можно использовать для прогнозирования.

Прогнозное значение производительности труда (Y) рассчитывается по формуле $=M5 * C34 + L5 * D34 + K5 * E34 + J5 * F34 + I5 * G34 + N5$, введенной в ячейку В34.

В ячейке В35 рассчитано прогнозное значение производительности труда с использованием функции ТЕНДЕНЦИЯ:

$$=\text{ТЕНДЕНЦИЯ}(B4 : B33 ; C4 : G33 ; C34 : G34 ; 1) .$$

Для построения моделей на основании **экспоненциальной зависимости** и дальнейшего расчета прогнозов Excel предлагает функции **ЛГРФПРИБЛ** и **РОСТ**, работа которых основана на вычислении экспоненциальной кривой, аппроксимирующей данные. Эти функции могут использоваться как для расчета однофакторных, так и многофакторных моделей.

▪ **Функция ЛГРФПРИБЛ** в регрессионном анализе вычисляет экспоненциальную кривую, аппроксимирующую данные, и возвращает массив значений, описывающий эту кривую. Она является универсальной для расчета параметров экспоненциальных моделей, так как кроме коэффициентов уравнения регрессии может возвращать и дополнительную статистику по регрессии.

Формат:

ЛГРФПРИБЛ(известные значения_Y; известные значения_X; константа; статистика).

В отличие от функции **ЛИНЕЙН** аргумент *константа* — это логическое значение, которое указывает, требуется ли, чтобы константа b была равна 1.

Если константа имеет значение **ИСТИНА** или опущено, то b вычисляется обычным образом.

Если константа имеет значение **ЛОЖЬ**, то b полагается равным 1 и значения m подбираются так, чтобы удовлетворить соотношению $y = m^x$.

Способ использования функции **ЛГРФПРИБЛ** аналогичен функции **ЛИНЕЙН**, с той лишь разницей, что исходный набор данных аппроксимируется не прямой линией, а экспонентой. В соответствии с этим и количеством независимых переменных уравнение регрессии соответствует виду (7.3) или виду (7.4).

▪ **Функция РОСТ** аналогична функции ТЕНДЕНЦИЯ и используется для расчета прогнозируемого экспоненциального роста на основании имеющихся данных. Она возвращает значения Y для последовательности новых значений X , задаваемых с помощью существующих X - и Y -значений.

Формат :

РОСТ(известные значения $_Y$; известные значения $_X$; новые значения $_X$; константа).

В соответствии с условием *примера 7.1* построим экспоненциальную многофакторную регрессионную модель, оценивающую влияние вышеперечисленных факторов на производительность труда.

Для этого в ячейки I15:N19 (см. рисунок 7.1) введем функцию ЛГРФПРИБЛ в формате: =ЛГРФПРИБЛ (B4:V33;C4:G33;1;1) .

Результатом работы функции является массив значений (ячейки I15:N19):

- ячейки I15:N11 – коэффициенты уравнения регрессии m и свободный член b ;
- ячейка I17 – коэффициент детерминированности $R^2 = 0,724$;
- ячейка I18 – критерий Фишера $F=12,59$.

Таким образом, экспоненциальная многофакторная регрессионная модель, оценивающая влияние коэффициента сменности оборудования, премий и вознаграждений на одного рабочего, среднегодовой стоимости ОПС, фондовооруженности труда на производительность труда, имеет вид:

$$Y = 1.44 \cdot 3.52^{x_1} \cdot 2.11^{x_2} \cdot 1.156^{x_3} \cdot 0.99^{x_4} \cdot 1.037^{x_5} \quad (7.5)$$

Вывод: поскольку коэффициент детерминированности $R^2=0.724$ лежит в пределах $0,75 - 1$, расчетное значение критерия Фишера 12.59 больше табличного (2.62), модель следует признать адекватной, и следовательно ее можно использовать для прогнозирования.

Прогнозное значение производительности труда в соответствии с экспоненциальной моделью (7.5) может быть вычислено по формуле =N15*M15^C34*L15^D34*K15^E34*J15^F34*I15^G34 или с использованием функции РОСТ: =РОСТ (B4:V33;C4:G33;C34:G34;1) .

3 Анализ и решение задач межотраслевого баланса

Рассмотрим на примере расчета межотраслевого баланса (МОБ)¹⁰.

Матричная форма записи межотраслевого баланса имеет следующий вид:

$$X - A \cdot X = Y, \quad (7.6)$$

где $Y (Y_j)$ – вектор конечного продукта,

¹⁰ Теоретические основы расчета МОБ подробно рассматриваются при изучении специальных экономических дисциплин и курса «Эконометрика и экономико-математические методы и модели».

$X(i)$ – вектор объемов производства,

$A(a_{ij})$ – матрица коэффициентов прямых затрат¹¹

После подсчета коэффициентов прямых затрат соотношение (7.6) можно использовать для анализа и планирования и решить следующие задачи:

1. **Определить объемы валовой продукции отраслей** X_1, X_2, \dots, X_n по заданным объемам конечной продукции $Y(j)$.

Выполнив несложные преобразования соотношения 7.6

$$E \cdot X - A \cdot X = Y \implies (E - A) \cdot X = Y \implies X = Y / (E - A) \implies X = (E - A)^{-1} \cdot Y, \quad (7.7)$$

объемы валовой продукции можно рассчитать по формуле

$$X = B \cdot Y,$$

где $B = (E - A)^{-1}$ – матрица, обратная матрице $(E - A)$ – матрица полных затрат¹², представляющая собой сумму прямых затрат и косвенных затрат.

2. **Определить объемы конечного продукта отраслей** Y_1, Y_2, \dots, Y_n по заданным объемам валовой продукции X_1, X_2, \dots, X_n по формуле

$$Y = (E - A) \cdot X \quad (7.8)$$

Реализация матричного метода решения модели МОБ (6.6) по формуле (6.7) предполагает:

- определение матрицы коэффициентов прямых затрат $A(a_{ij})$,
- определение единичной матрицы E ,
- вычисление разности матриц E и A ,
- нахождение матрицы, обратной к матрице $(E - A)$ – расчет матрицы полных затрат B ,
- выполнение операции умножения матрицы полных затрат B и вектора-столбца конечного продукта Y .

Пример 7.2. По известным данным о коэффициентах прямых затрат a_{ij} и конечном продукте Y в межотраслевом балансе для трех отраслей (см. таблицу 7.3) определить общий выпуск продукции по каждой отрасли x_{ij} .

Таблица 7.3. Исходные данные для расчета МОБ

Отрасль	Коэффициенты прямых затрат a_{ij}			Конечный продукт Y , млн. руб
	1	2	3	
1	0,2	0,3	0,3	35
2	0,1	0,1	0,1	50
3	0,4	0,5	0,4	45

¹¹ В натуральном балансе коэффициенты прямых затрат a_{ij} означают расход i -той продукции на изготовление j -той продукции.

В стоимостном балансе коэффициенты a_{ij} означают затраты i -той отрасли на каждый рубль валовой продукции j -той отрасли.

¹² Коэффициенты полных затрат b_{ij} показывают объем валовой продукции i -той отрасли, необходимый для производств единицы конечной продукции j -той отрасли.

По данным таблицы можно записать следующую систему уравнений:

$$\begin{cases} X_1 = 0.2X_1 + 0.3X_2 + 0.3X_3 + 35 \\ X_2 = 0.1X_1 + 0.1X_2 + 0.1X_3 + 50 \\ X_3 = 0.4X_1 + 0.5X_2 + 0.4X_3 + 45 \end{cases}$$

Превратив конечные продукты отраслей в свободные члены, получим:

$$\begin{cases} X_1 - 0.2X_1 - 0.3X_2 - 0.3X_3 = 35 \\ X_2 - 0.1X_1 - 0.1X_2 - 0.1X_3 = 50 \\ X_3 - 0.4X_1 - 0.5X_2 - 0.4X_3 = 45 \end{cases}$$

или

$$\begin{cases} 0.8X_1 - 0.3X_2 - 0.3X_3 = 35 \\ -0.1X_1 + 0.9X_2 - 0.1X_3 = 50 \\ -0.4X_1 - 0.5X_2 + 0.6X_3 = 45 \end{cases}$$

Решение данной системы уравнений осуществляется известными методами линейной алгебры, например, методом Крамера или матричным методом. При реализации решения в ТП MS Excel могут использоваться встроенные функции категории «Математические» МОБР, МУМНОЖ, МОПРЕД, рассмотренные в лекции 3.

4 Анализ и решение задач оптимизации плана производства продукции

Среди задач экономики и управления производством наиболее известны задачи оптимального планирования, к которым относится достаточно широкий круг задач оптимизации.

Задача оптимизации – это задача выбора таких условий и зависящих от них факторов, при которых критерий эффективности достигает экстремального значения.

Под **решением задач оптимизации** понимается процесс выбора таких значений переменных x , принадлежащих допустимой области D , которые обеспечивают оптимальное значение некоторой функции $F(x)$, называемой целевой.

Если целевая функция линейна, а область допустимых значений задается системой линейных уравнений или неравенств, то такая задача является задачей линейного программирования.

Модель задачи линейного программирования должна иметь вполне определенный вид: требуется найти максимум (минимум) значения целевой функции L при переменных x_1, x_2, \dots, x_n

$$L(x) = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \max(\min), \quad (7.9)$$

Максимально допустимое время работы на устройствах I, II, III составляет соответственно 428, 672 и 672 часа. Требуется рассчитать план производства, обеспечивающий максимальную прибыль.

Решение

Решение задачи оптимизации в среде ТП MS Excel предполагает:

1. Математическую постановку задачи.
2. Размещение на рабочем листе ТП MS Excel исходных данных, расчет значений ограничений, расчет значения целевой функции.
3. Формулировка математической модели задачи в терминах ячеек рабочего листа ТП MS Excel.
4. Поиск оптимального решения поставленной задачи средствами надстройки «Поиск решения».
5. Анализ результатов.

1. Математическая постановка задачи.

Под производственной программой понимается номенклатура продукции и объем выпуска каждого вида изделий, входящих в данную номенклатуру (ассортимент). В данной задаче в качестве критерия эффективности производственной программы выбрана прибыль от реализации, которую нужно максимизировать.

Общий вид экономико-математической модели оптимизации производственной программы по критерию максимизации прибыли:

целевая функция :
$$L(x) = \sum_{j=1}^n p_j \cdot x_j \rightarrow \max ,$$

при ограничениях на запас i-го ресурса:
$$\sum_{j=1}^n a_{ij} x_j \leq b_i ,$$

и неотрицательности переменных $x_j \geq 0$.

Если обозначить через x_1 – количество изделий вида А;

x_2 – количество изделий вида В;

то математическая модель данной задачи будет иметь вид:

Целевая функция: $F = 3x_1 + 8x_2 \rightarrow \max$

Система ограничений:

$$\begin{cases} 2x_1 + 3x_2 \leq 428 \\ 3x_1 + 6x_2 \leq 672 \\ 2x_1 + 8x_2 \leq 672 \\ x_1, x_2 \geq 0, \quad x_1, x_2 - \text{целое} \end{cases}$$

2. Размещение данных на рабочем листе ТП MS Excel.

Разместим исходные данные в ячейках А3:F6 рабочего листа ТП MS Excel, как показано на рисунке 7.3.

	A	B	C	D	E	F
1	Приложение 3					
2						
3	Вид продукта	Время обработки, ч.			Прибыль, у.е.	Количество
4		I	II	III		
5	A	2	3	2	3	112
6	B	3	6	8	8	56
7					=E5*F5+E6*F6	
8	Время	=B5*\$F\$5+B6*\$F\$6	=C5*\$F\$5+C6*\$F\$6	=D5*\$F\$5+D6*\$F\$6		
9						
10	Ограничения	428	672	672		

Рисунок 7.3 – Исходные данные и результаты решения

В ячейки F5 : F6 внесем начальное значение параметров x_1 и x_2 (примем их равными нулю).

В ячейки B10:D10 внесем значения ограничений на использование оборудования каждого вида 428, 672 и 672 соответственно.

В ячейках B8:D8 рассчитаем значения ограничений на использование оборудования каждого вида соответственно:

$$\text{В ячейке B8} = B5 * \$F\$5 + B6 * \$F\$6.$$

$$\text{В ячейке C8} = C5 * \$F\$5 + C6 * \$F\$6.$$

$$\text{В ячейке D8} = D5 * \$F\$5 + D6 * \$F\$6.$$

$$\text{В ячейке E7 рассчитаем значение целевой функции} = E5 * F5 + E6 * F6.$$

3. *Формулировка математической модели задачи в терминах ячеек рабочего листа ТП MS Excel.*

Целевая функция: ячейка E7 \rightarrow max

Система ограничений:

$$\left\{ \begin{array}{l} B8 \leq B10 \\ C8 \leq C10 \\ D8 \leq D10 \\ F5 : F6 \geq 0, F5 : F6 \text{ - целое} \end{array} \right.$$

Таким образом, в терминах ячеек рабочего листа ТП MS Excel математическая модель задачи может быть сформулирована следующим образом: добиться максимального значения в ячейке E7, изменяя значения ячеек F5:F6 так, чтобы значения в ячейках B8:D8 были бы не больше значений в ячейках B10:D10 при неотрицательных и целых значениях в ячейках F5:F6.

4. *Поиск оптимального решения.*

Окно надстройки «Поиск решения» (меню Данные \rightarrow Поиск решения) с постановкой задачи в терминах ячеек рабочего листа Excel приведено на рисунке 7.4.

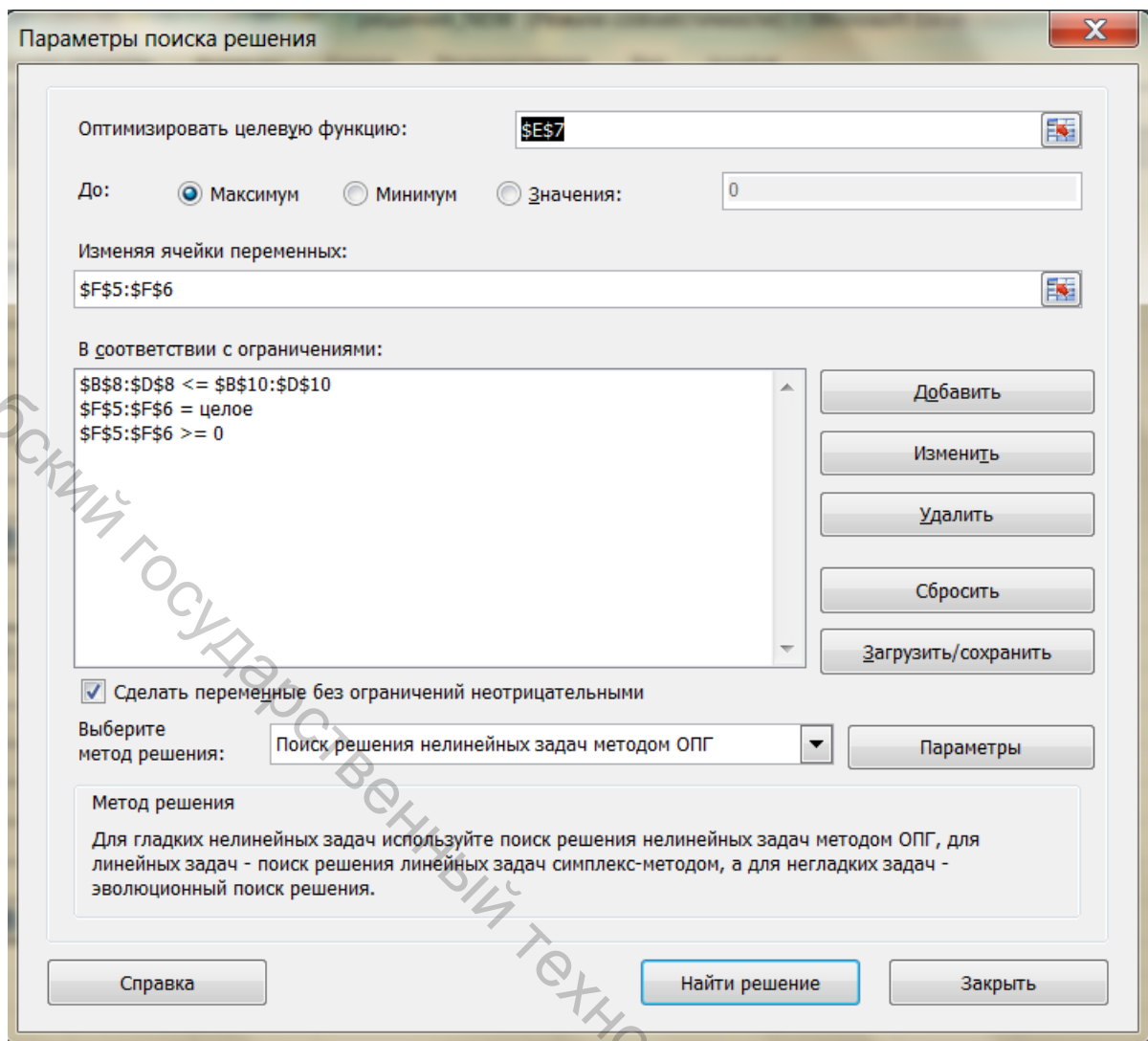


Рисунок 7.4 – Окно утилиты «Поиск решения»

5. Анализ результатов.

В результате оптимизации получено:

максимальная прибыль в размере 784 у.е. будет получена при производстве 112 шт. изделий вида А и 56 шт. изделия вида В, время использования оборудования каждого вида при оптимальном плане составит 392, 672 и 672 час. соответственно.

Результаты оптимизации можно посмотреть в Отчете по результатам, сформированном «Поиском решения».

5 Анализ и решение задач оптимизации плана транспортных перевозок

Сущность задачи оптимизации плана транспортных перевозок (транспортной задачи линейного программирования) состоит в наивыгоднейшем прикреплении поставщиков однородного продукта ко многим потребителям этого продукта. Условие транспортной задачи обычно

записывается в виде матрицы, в которой потребители однородного груза размещаются по столбцам, а поставщики – по строкам. В последнем столбце матрицы проставляют запас груза, имеющийся у каждого поставщика, а в последней строке – потребность в нем потребителей. На пересечении строк со столбцами записывают размер поставки, а также расстояние пробега по всем возможным маршрутам, время доставки груза или затраты на перевозку единицы груза по этим маршрутам.

Математически *транспортная задача по критерию стоимости* формулируется следующим образом.

Имеется n потребителей и m поставщиков однородного груза.

Мощность i -го поставщика ($i = \overline{1, m}$) обозначается a_i ,

спрос j -го потребителя ($j = \overline{1, n}$) обозначается b_j .

Затраты на перевозку одной тонны груза от i -го поставщика до j -го потребителя обозначаются c_{ij} .

Размер поставки продукции поставщиком i потребителю j обозначим x_{ij} ; общую сумму затрат на перевозку груза обозначим через L .

Математическую модель задачи имеет вид:

Целевая функция – общая сумма затрат на перевозку должна быть минимальной:

$$L = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min.$$

Ограничения:

объем поставок i -го поставщика должен равняться количеству имеющегося у него груза:

$$\sum_{j=1}^n x_{ij} = a_i \quad (i = \overline{1, m});$$

объем поставок j -му потребителю должен быть равен его спросу:

$$\sum_{i=1}^m x_{ij} = b_j \quad (j = \overline{1, n});$$

запас груза у поставщиков должен равняться суммарному спросу потребителей:

$$a_i = b_j$$

размер поставок должен выражаться неотрицательным числом:

$$x_{ij} \geq 0;$$

В случае выполнения равенства $a_i = b_j$, когда запас груза у поставщиков равняется суммарному спросу потребителей, имеет место закрытая транспортная модель. Если это равенство не соблюдается, запас груза у поставщиков не равен суммарному спросу потребителей, вводится фиктивный потребитель или фиктивный поставщик и модель изменяется.

Пример 7.4.

Имеются 5 пунктов производства и 4 пункта распределения продукции.

Стоимость перевозки единицы продукции с i -го пункта производства в j -й центр распределения c_{ij} приведена в таблице, где под строкой понимается пункт производства, а под столбцом – пункт распределения. Кроме того, в этой таблице в i -той строке указан объем производства в i -м пункте производства, а в j -м столбце указан спрос в j -м центре распределения. Необходимо составить план перевозок по доставке требуемой продукции в пункты распределения, минимизирующий суммарные транспортные расходы.

		Стоимость перевозки единицы продукции				Объемы производства
		5	9	3	10	10
		3	10	5	9	30
		7	2	3	8	20
		8	5	11	2	20
		5	9	10	5	20
Объемы потребления		50	10	30	10	

Рисунок 7.5 – Условие транспортной задачи

Решение

1. Математическая постановка задачи.

Обозначим через x_{ij} – объемы перевозок от i -го поставщика j -му потребителю. Тогда математическая модель задачи будет иметь вид:

- Целевая функция (стоимость перевозок):

$$F = 5x_{11} + 9x_{12} + 3x_{13} + 10x_{14} + 3x_{21} + 10x_{22} + 5x_{23} + 9x_{24} + 7x_{31} + 2x_{32} + 3x_{33} + 8x_{34} + 8x_{41} + 5x_{42} + 11x_{43} + 2x_{44} + 5x_{51} + 9x_{52} + 10x_{53} + 5x_{54} \rightarrow \min$$

- Система ограничений на объемы производства:

$$\begin{cases} x_{11} + x_{12} + x_{13} + x_{14} = 10 \\ x_{21} + x_{22} + x_{23} + x_{24} = 30 \\ x_{31} + x_{32} + x_{33} + x_{34} = 20 \\ x_{41} + x_{42} + x_{43} + x_{44} = 20 \\ x_{51} + x_{52} + x_{53} + x_{54} = 20 \end{cases}$$

- Система ограничений на объемы потребления:

$$\begin{cases} x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 50 \\ x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 10 \\ x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 30 \\ x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 10 \end{cases}$$

- Ограничения целочисленности и неотрицательности переменных:

$$x_{ij} \geq 0, \quad x_{ij} \text{ – целое}$$

2. Размещение данных на рабочем листе ТП MS Excel.

- Разместим исходные данные в ячейках В3:Е7 (рисунок 7.6).
- Отведем ячейки В12:Е16 под значения неизвестных (объемов перевозок).
- Введем в ячейки G12:G16 объемы производства на фабриках.
- Введем в ячейки В18:Е18 потребность в продукции в пунктах распределения.
- В ячейку В21 (рисунок 7.7) введем функцию цели =СУММПРОИЗВ(В3:Е7;В12:Е16).
- В ячейки F12:F16 введем формулы, вычисляющие объемы производства на фабриках, в ячейки В17:Е17 – объемы доставляемой продукции в пункты распределения (рисунок 7.7).

	A	B	C	D	E	F	G
1	Стоимости перевозок						
2		Стоимость перевозки единицы продукции					
3		5	9	3	10		
4		3	10	5	9		
5		7	2	3	8		
6		8	5	11	2		
7		5	9	10	5		
8							
9							
10	неизвестные						
11		Объемы перевозок продукции					Объемы производства
12		0	0	0	0	0	10
13		0	0	0	0	0	30
14		0	0	0	0	0	20
15		0	0	0	0	0	20
16		0	0	0	0	0	20
17		0	0	0	0		
18	Объемы потребления	50	10	30	10		
19							
20		Функция цели					
21		0					

Рисунок 7.6 – Исходные данные транспортной задачи

	A	B	C	D	E	F	G
1	Стоимости перевозок						
2	Стоимость перевозки единицы продукции						
3	5	9	3	10			
4	3	10	5	9			
5	7	2	3	8			
6	8	5	11	2			
7	5	9	10	5			
8							
9							
10	неизвестные						
11	Объемы перевозок продукции						
12	0	0	0	0	=СУММ(B12:E12)		Объемы производст ва
13	0	0	0	0	=СУММ(B13:E13)		10
14	0	0	0	0	=СУММ(B14:E14)		30
15	0	0	0	0	=СУММ(B15:E15)		20
16	0	0	0	0	=СУММ(B16:E16)		20
17	=СУММ(B12:B16)		=СУММ(C12:C16)	=СУММ(D12:D16)	=СУММ(E12:E16)		
18	Объемы потребления	50	10	30	10		
19							
20	Функция цели						
21	=СУММПРОИЗВ(B3:E7;B12:E16)						

Рисунок 7.7 – Исходные данные в режиме формул

3. Постановка задачи в терминах рабочего листа Excel и использование утилиты «Поиск решения».

В окне утилиты «Поиск решения» зададим целевую ячейку, изменяемые ячейки и ограничения (рисунок 7.8).

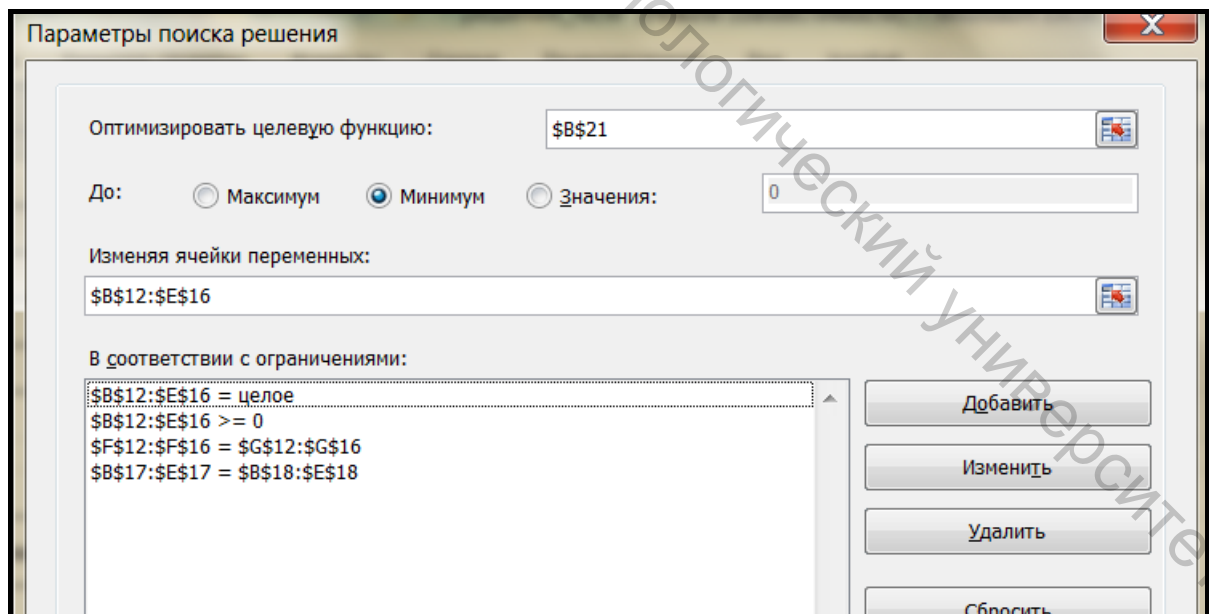


Рисунок 7.8 – Параметры окна «Поиск решения»

Оптимальный план, обеспечивающий минимальные затраты на перевозку продукции от производителей к потребителям, найденный с

помощью утилиты «Поиск решения», представлен на рисунке 7.9.

	A	B	C	D	E	F	G
1	Стоимости перевозок						
2		Стоимость перевозки единицы продукции					
3		5	9	3	10		
4		3	10	5	9		
5		7	2	3	8		
6		8	5	11	2		
7		5	9	10	5		
8							
9							
10	неизвестные						
11		Объемы перевозок продукции					Объемы производства
12		0	0	10	0	10	10
13		30	0	0	0	30	30
14		0	0	20	0	20	20
15		0	10	0	10	20	20
16		20	0	0	0	20	20
17		50	10	30	10		
18	Объемы потребления	50	10	30	10		
19							
20		Функция цели					
21		350					

Рисунок 7.9 – Результаты решения транспортной задачи

3. Анализ результатов.

Минимальная стоимость перевозок груза от поставщиков к потребителям составляет 350 ден. ед. при плане перевозок, представленном на рисунке 7.9.

6 Литература к лекции 7

1. Миксюк, С. Экономико-математические методы и модели : учебно-практическое пособие / С. Ф. Миксюк, В. Н. Комкова. – Минск: БГЭУ, 2006. – 219 с.

2. Шарстнев, В. Л. Компьютерные информационные технологии: курс лекций / В. Л. Шарстнев. – Витебск: УО ВГТУ, 2008. – 350 с.

3. Шарстнев, В. Л. Компьютерные информационные технологии: лабораторный практикум : пособие / В. Л. Шарстнев, Е. Ю. Вардомацкая. – Витебск: УО «ВГТУ», 2008. – 170 с.

4. Шарстнев, В. Л. Компьютерные информационные технологии. Пакеты прикладных программ для моделирования и анализа задач экономики: пособие / В. Л. Шарстнев, Е. Ю. Вардомацкая. – Витебск: УО «ВГТУ», 2008. – 138 с.

Лекция 8

ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ ДЛЯ МОДЕЛИРОВАНИЯ И СИМВОЛЬНОГО РЕШЕНИЯ ЗАДАЧ ЭКОНОМИКИ И УПРАВЛЕНИЯ

План лекции

1. Стандартные библиотеки (модули) пакетов символьной математики для решения задач моделирования и оптимизации.
2. Корреляционно-регрессионное моделирование.
3. Анализ и решение задач межотраслевого баланса.
4. Анализ и решение задач оптимального планирования.
5. Литература.

1 Стандартные библиотеки (модули) пакетов символьной математики для решения задач моделирования и оптимизации

Возможности пакетов для моделирования и символьного решения типовых задач экономики и управления рассмотрим на примере системы компьютерной математики Maple. Как говорилось выше, значительная часть встроенных функций СКМ Maple доступна непосредственно из его ядра. Чтобы использовать встроенные функции, предназначенные для специализированных расчетов, следует подключать соответствующие пакеты расширений.

В частности, для проведения статистического анализа, решения задач моделирования и оптимизации можно использовать функции из пакетов расширений (библиотек), перечисленных в таблице 8.1.

Таблица 8.1 – Основные пакеты расширений СКМ Maple для решения задач моделирования и оптимизации

Пакет	Назначение	Пакет	Назначение
linalg	Линейная алгебра	LinearAlgebra	Линейная алгебра
simplex	Линейная оптимизация	Optimization	Нелинейная оптимизация
stats	Статистика	Statistics	Статистика
networks	Теория графов	GraphTheory	Теория графов
CurveFitting	Приближение кривых	finance	Финансовые расчеты

2 Корреляционно-регрессионное моделирование

При использовании системы символьной математики Maple для построения экономико-математической модели выбор вида модели не предопределен разработчиками пакета. СКМ Maple позволяет пользователю произвольно, по своему усмотрению, задавать вид модели и при необходимости оперативно его изменять.

Для проведения корреляционно-регрессионного анализа в среде СКМ Maple можно использовать функции библиотеки Statistics. Для кусочной аппроксимации удобнее пользоваться функциями библиотеки CurveFitting.

Рассмотрим функции библиотеки Statistics, с помощью которых можно рассчитывать разные виды экономико-математических регрессионных моделей.

- `Fit` – описывает данные в виде введенной зависимости.

Формат:

`Fit(f, X, Y, v, options)`

`f` – вид моделируемой зависимости,

`X` – вектор значений независимой переменной,

`Y` – вектор значений зависимой переменной,

`v` – имя независимой переменной,

`options` – выражение в формате: `option=value`, где `value` – значение одного из возможных дополнительных параметров команды `Fit` (статистические характеристики уравнения регрессии).

- `LinearFit` – описывает данные в виде линейной зависимости

Формат:

`LinearFit(falg, X, Y, v, options)`

`LinearFit(fop, X, Y, options)`

`falg` – список или вектор параметров функции в алгебраической форме,

`X` – вектор или матрица значений независимой (`yx`) переменной,

`Y` – вектор значений зависимой переменной,

`v` – имя независимой переменной – параметра функции,

`fop` – список или вектор параметров функции в операторной форме,

`options` – выражение в формате: `option=value`, где `option` один из возможных параметров команды `LinearFit`,

`value` – значения указанного параметра (статистические характеристики уравнения регрессии).

- `ExponentialFit` – описывает данные в виде экспоненциальной зависимости

Формат:

`ExponentialFit(X, Y, v)`

X – вектор значений независимой переменной,
Y – вектор значений зависимой переменной,
v – имя независимой переменной.

▪ `LogarithmicFit` – описывает данные в виде логарифмической зависимости.

Формат:

`LogarithmicFit(X, Y, v)`

X – вектор или матрица значений независимой (ых) переменной,

Y – вектор значений зависимой переменной,

v – имя независимой переменной – параметра функции.

▪ `NonlinearFit` – описывает данные в виде введенной нелинейной зависимости.

Формат:

`NonlinearFit(falg, X, Y, v, options)` или

`NonlinearFit(fop, X, Y, options)`

▪ `PolynomialFit` – описывает данные в виде полиномиальной зависимости.

Формат:

`PolynomialFit(d, X, Y, v, options)` или

`PolynomialFit(d, X, Y, v, options)`

d – степень полинома.

▪ `PowerFit` – описывает данные в виде степенной зависимости.

Формат:

`PowerFit(X, Y, v)`.

Пример 8.1. Построить регрессионные модели разных видов для набора статистических данных X и Y, представляющих собой значения *прибыли* (зависимая переменная Y) и значения *производительности труда* (независимый фактор X). Оценить ошибки моделирования.

Исходные данные приведены в таблице 8.2:

Таблица 8.2 – Исходные данные к примеру 8.1

Y	60	75	41	82	90	129	145	160	140	250	262
X	1,7	2,2	0,6	2,3	3,4	3,9	4,7	5,8	5,6	6,4	7,2

Решение:

> restart;

1. Определяем исходные данные: массивы X-производительность труда, Y-прибыль:

> Y := [60, 75, 41, 82, 90, 129, 145, 160, 140, 250, 262];

> X := [1.7, 2.2, 0.6, 2.3, 3.4, 3.9, 4.7, 5.8, 5.6, 6.4, 7.2];

2. Формируем график $Y=f(x)$:

```
>gr:=plot(X,Y,x=0..10,y=0..300,style=point,thickness=3):
```

3. Подключаем библиотеку *Statistics*:

```
> with(Statistics):
```

4. Вычисляем коэффициент корреляции и коэффициент детерминированности:

```
> Coeff_Cor:=Correlation(Y,X);  
Coeff_Cor := 0.9400911641
```

```
> Coeff_Det:=evalf(Coeff_Cor^2,3);  
Coeff_Det := 0.884
```

5. Рассчитываем линейную модель:

```
> LinM:=evalf(Fit(m*x+b,X,Y,x),3);  
LinM := 3.04 + 32.0 x
```

```
> Lin_M:=evalf(LinearFit([1, x],X,Y,x),3);  
Lin_M := 3.04 + 32.0 x
```

6. Выполняем прогноз по линейной модели для $x=10$:

```
> Prib_Lin:=subs(x=10,LinM);  
Prib_Lin := 323.04
```

7. Формируем график линейной модели:

```
>gr1:=plot(LinM,x=1..10,y=0..300,color=blue,thickness=4):
```

8. Рассчитываем экспоненциальную модель:

```
> ExpM:=evalf(Fit(b*m^x,X,Y,x),3);  
ExpM := 39.7 1.30x
```

```
> Exp_M:=evalf(ExponentialFit(X,Y,x),3);  
Exp_M := 39.3 e(0.265 x)
```

9. Выполняем прогноз по экспоненциальной модели для $x=10$:

```
> Prib_Exp:=evalf(subs(x=10,ExpM),5);  
Prib_Exp := 547.30
```

10. Формируем график экспоненциальной модели:

```
>gr2:=plot(ExpM,x=1..10,y=0..300,color=green,thickness=4)  
:
```

11. Рассчитываем логарифмическую модель:

```
> LogM:=evalf(Fit(a+b*ln(x),X,Y,x),4);  
LogM := 32.75 + 82.06 ln(x)
```

```
> Log_M:=evalf(LogarithmicFit(X,Y,x),4);  
Log_M := 32.75 + 82.06 ln(x)
```

12. Прогноз:

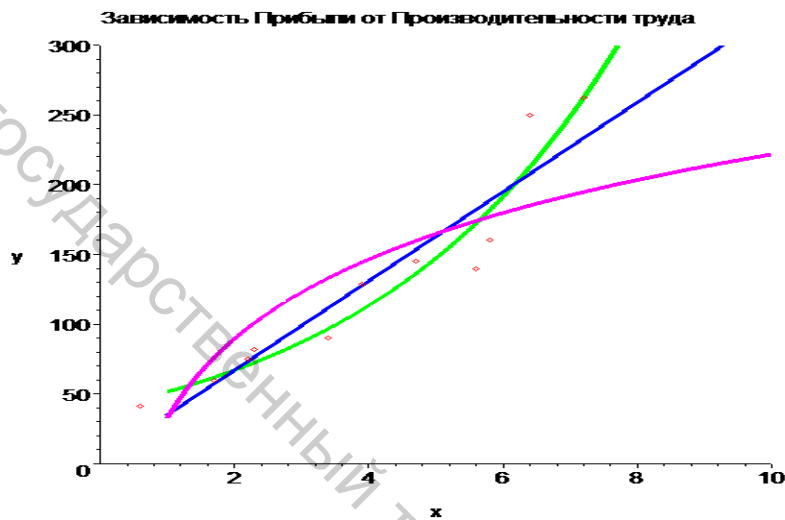
```
> Pr_Log:=evalf(subs(x=10,LogM),5);
Pr_Log := 221.70
```

13. Формируем график логарифмической модели:

```
>gr3:=plot(LogM,x=1..10,y=0..300,color=magenta,thickness=4):
```

14. Подключаем библиотеку plots и выводим графики на экран:

```
>with(plots):
>display({gr,gr1,gr2,gr3}, title = "Зависимость Прибыли от Производительности труда"13);
```



3 Анализ и решение задач межотраслевого баланса

Для построения модели межотраслевого баланса необходимо уметь пользоваться функциями работы с матрицами. Часть этих функций находится в ядре системы, часть в библиотеках `linalg` и `LinearAlgebra`. Форматы этих функций интуитивно понятны. Полную информацию по возможностям их использования можно получить в справочной системе СКМ Maple.

Пример 8.2. (см. условие примера 7.2). По известным данным о коэффициентах прямых затрат (a_{ij}) и конечном продукте (Y) в межотраслевом балансе для трех отраслей (промышленность, строительство, сфера услуг) определить общий выпуск продукции по каждой отрасли (x_{ij}).

Отрасль	Коэффициенты прямых затрат a_{ij}			Конечный продукт Y , млн. руб.
	1	2	3	
1	0,2	0,3	0,3	35
2	0,1	0,1	0,1	50

¹³ Команда `display(lst,opns)` из графического пакета расширений `plots` выводит на экран уже сформированные графики, заданные списком `lst`.

3	0,4	0,5	0,4	45
---	-----	-----	-----	----

Решение:

1. Определяем матрицу коэффициентов прямых затрат:

```
> A:=matrix([[0.2,0.3,0.3],[0.1,0.1,0.1],[0.4,0.5,0.4]]);
```

$$A := \begin{bmatrix} 0.2 & 0.3 & 0.3 \\ 0.1 & 0.1 & 0.1 \\ 0.4 & 0.5 & 0.4 \end{bmatrix}$$

2. Определяем единичную матрицу:

```
> E:=Matrix(3,3,shape=identity);
```

$$E := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3. Находим матрицу E-A:

```
> K:=evalm(E-A);
```

$$K := \begin{bmatrix} 0.8 & -0.3 & -0.3 \\ -0.1 & 0.9 & -0.1 \\ -0.4 & -0.5 & 0.6 \end{bmatrix}$$

4. Определяем вектор-столбец свободных членов:

```
> B:=vector([35,50,45]);
```

$$B := [35, 50, 45]$$

5. Подключаем библиотеку linalg:

```
> with(linalg);
```

6. Вычисляем общий выпуск продукции по отраслям:

```
> Pr:=linsolve(K,B);
```

$$Pr := [197.2803348, 110.6694561, 298.7447700]$$

7. Задаем количество значащих цифр:

```
> Pr:=evalf(%,4);
```

$$Pr := [197.3, 110.7, 298.7]$$

Результаты расчета МОБ совпали с результатами, полученными при решении в среде ТП MS Excel.

4 Анализ и решение задач оптимального планирования

Задачи оптимального планирования в СКМ Maple решаются при помощи функций библиотек simplex (только линейное программирование) и

Optimization (линейное и нелинейное программирование).

В частности, стандартная Библиотека Optimization позволяет отыскивать оптимальные решения для задач следующего вида:

- линейного программирования LPSolve,

Формат:

LPSolve(obj, constr, bd, opts),

где obj – алгебраическое выражение для целевой функции,

constr – линейные ограничения { },

bd – maximize или minimize,

opts – необязательные дополнительные параметры – одно из дополнительных ключевых слов, которое позволяет установить дополнительные ограничения на отыскание значений переменных. Эти параметры вводятся специальным словом assume = binary, integer or nonnegint:

binary – логическое значение переменной (0 или 1),

nonnegative – неотрицательное значение переменной,

integer – целочисленное значение переменной,

nonnegint – неотрицательное целочисленное значение переменной.

- квадратического программирования QPSolve,
- нелинейного программирования NLPsolve,
- среднеквадратического отклонения LSSolve.

Пример 7.3. Оптимизация плана производства продукции.

Вернемся к условию примера 7.3, стр. 85 (исходные данные в таблице ниже) и реализуем решение в СКМ Maple средствами библиотеки Optimization.

Вид продукции	Время обработки, ч			Прибыль, у.е.
	I	II	III	
A	2	3	2	3
B	3	6	8	8
Ограничения на ресурсы оборудования, час.	428	672	672	?

Решение:

Экономико-математическая модель задачи представлена на стр. 77.

```
> restart;
```

```
> with(Optimization);
```

```
> F:=3*A+8*B;
```

$$F := 3x_1 + 8x_2$$

```
> ogran:={2*A+3*B<=428, 3*A+6*B<=672, 2*A+8*B<=672};
```

$$\text{ogran} := \{2x_1 + 3x_2 \leq 428, 2x_1 + 8x_2 \leq 672, 3x_1 + 6x_2 \leq 672\}$$

```
> rez:= LPSolve(F, ogran, assume={nonnegint},
maximize)
[784, [x1 = 112, x2 = 56]]
```

Результаты оптимизации совпали с результатами, полученными в ТП MS Excel.

Следует отметить, что в последних версиях Maple библиотека Optimization по сути поглотила библиотеку simplex. Поскольку библиотека simplex не предусматривает параметра для наложения ограничения целочисленности на переменные, результаты оптимизации при одной и той же постановке задачи всегда будут различаться. Таким образом, при необходимости получения целочисленного решения оптимизационных задач использование библиотеки Optimization предпочтительно.

Пример 7.4. Оптимизации плана транспортных перевозок.

Вернемся к условию примера 7.4, стр. 90 (исходные данные в таблице ниже) и реализуем решение в СКМ Maple средствами библиотеки Optimization.

		Стоимость перевозки единицы продукции				Объемы производства	
		5	9	3	10	10	
		3	10	5	9	30	
		7	2	3	8	20	
		8	5	11	2	20	
		5	9	10	5	20	
Объемы потребления		50	10	30	10		

Решение:

```
> restart;
> with(Optimization):
> F:=5*x11+9*x12+3*x13+10*x14+3*x21+10*x22+5*x23+9*x24+7*x
31+2*x32+3*x33+8*x34+8*x41+5*x42+11*x43+2*x44+5*x51+9*x52
+10*x53+5*x54;
F:=5 x11 + 9 x12 + 3 x13 + 10 x14 + 3 x21 + 10 x22 + 5 x23 + 9 x24 + 7 x31 + 2 x32
+ 3 x33 + 8 x34 + 8 x41 + 5 x42 + 11 x43 + 2 x44 + 5 x51 + 9 x52 + 10 x53 + 5 x54
> ogran:={x11+x12+x13+x14=10,
x21+x22+x23+x24=30, x31+x32+x33+x34=20, x41+x42+x43+x44=20,
x51+x52+x53+x54=20, x11+x21+x31+x41+x51=50,
x12+x22+x32+x42+x52=10, x13+x23+x33+x43+x53=30, x14+x24+x34
+x44+x54=10};
```

```

ogran := { x11 + x12 + x13 + x14 = 10, x21 + x22 + x23 + x24 = 30,
           x31 + x32 + x33 + x34 = 20, x41 + x42 + x43 + x44 = 20, x51 + x52 + x53 + x54 = 20,
           x11 + x21 + x31 + x41 + x51 = 50, x12 + x22 + x32 + x42 + x52 = 10,
           x13 + x23 + x33 + x43 + x53 = 30, x14 + x24 + x34 + x44 + x54 = 10 }

```

```
> rez:=Minimize(F, ogran, assume=nonnegint);
```

```

[350, [x11=0, x12=0, x13=10, x14=0, x21=30,
       x22=0, x23=0, x24=0, x31=0, x32=0, x33
       =20, x34=0, x41=0, x42=10, x43=0, x44
       =10, x51=20, x52=0, x53=0, x54=0]]

```

Для представления результата в матричной форме:

```
>V:=matrix([[0,0,10,0],[30,0,0,0],[0,0,20,0],[0,10,0,10],
            [20,0,0,0]]);
```

$$V := \begin{bmatrix} 0 & 0 & 10 & 0 \\ 30 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 10 & 0 & 10 \\ 20 & 0 & 0 & 0 \end{bmatrix}$$

5 Литература к лекции 8

1. Миксюк, С. Экономико-математические методы и модели : учебно-практическое пособие / С. Ф. Миксюк, В. Н. Комкова. – Минск: БГЭУ, 2006. – 219 с.

2. Шарстнев, В. Л. Компьютерные информационные технологии: курс лекций / В. Л. Шарстнев. – Витебск: УО ВГТУ, 2008. – 350 с.

3. Шарстнев, В. Л. Компьютерные информационные технологии: лабораторный практикум : пособие / В. Л. Шарстнев, Е. Ю. Вардомацкая. – Витебск: УО «ВГТУ», 2008. – 170 с.

4. Шарстнев, В. Л. Компьютерные информационные технологии. Пакеты прикладных программ для моделирования и анализа задач экономики: пособие / В. Л. Шарстнев, Е. Ю. Вардомацкая. – Витебск: УО «ВГТУ», 2008. – 138 с.

Лекция 9

КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

План лекции

1. Понятие информационной системы.
2. Классификация и виды обеспечения информационных систем.
Корпоративные информационные системы (КИС).
3. Принципы организации корпоративных информационных систем.
4. Современная структура корпоративной информационной системы.
5. Информационное обеспечение корпоративных информационных систем.
6. Требования к корпоративным информационным системам.
7. Литература.

1 Понятие информационной системы

Информационная система (ИС) – это вся инфраструктура предприятия, задействованная в процессе управления всеми информационно-документальными потоками, включающая в себя следующие обязательные элементы:

- Информационная модель, представляющая собой совокупность правил и алгоритмов функционирования ИС. Информационная модель включает в себя все формы документов, структуру справочников и данных и т. д.
- Регламент развития информационной модели и правила внесения в неё изменений.
- Кадровые ресурсы, отвечающие за формирование и развитие информационной модели.
- Программное обеспечение, конфигурация которого соответствует требованиям информационной модели (программное обеспечение является основным двигателем и, одновременно, механизмом управления ИС).
- Кадровые ресурсы, отвечающие за настройку и адаптацию программного обеспечения, и его соответствие утвержденной информационной модели.
- Регламент внесения изменений в настраиваемые структуры (специфические настройки, структуры баз данных и т. д.) и конфигурацию программного обеспечения и состав его функциональных модулей.
- Аппаратно-техническая база, соответствующая требованиям по

эксплуатации программного обеспечения (компьютеры на рабочих местах, периферия, каналы телекоммуникаций, системное программное обеспечение и СУБД).

- Эксплуатационно-технические кадровые ресурсы, включая персонал по обслуживанию аппаратно-технической базы.

- Правила использования программного обеспечения и пользовательские инструкции, регламент обучения и сертификацию пользователей.

Ресурсы корпораций включают:

1. Материальные (материалы, готовая продукция, основные средства).
2. Финансовые.
3. Людские (персонал).
4. Знания (ноу-хау).
5. КИС.

Система управления любой компании включает три основные *подсистемы*:

1. Планирование продаж и операций. Это общий план функционирования предприятия, устанавливающий объемы изготовления готовой продукции.

2. Детальное планирование необходимых ресурсов (материалов, производственных мощностей, трудовых ресурсов и т. д.). Составленный план определяет время и объем заказов для всех материалов и комплектующих, необходимых для реализации основного производственного плана.

3. Управление исполнением планов в процессе производства и закупок (снабжения).

Все эти подсистемы реализуются на основе КИС.

2 Классификация и виды обеспечения информационных систем. Корпоративные информационные системы (КИС)

Многочисленные и разнообразные ИС, которые существуют сегодня, можно классифицировать по разным признакам.

Используемая техническая база.

- **ИС, работающая на одном компьютере (ПК, мини-ЭВМ или большая ЭВМ).** Вся информация сосредоточена в памяти этой машины и на ней же функционирует все программное обеспечение системы.

- **ИС на базе локальной сети.** Это ИС системы, обслуживающие учреждение, предприятие, фирму. В такой ИС информация может передаваться по сети между разными пользователями; разные части общедоступных данных могут храниться на разных компьютерах сети.

- **ИС на базе глобальных компьютерных сетей.** Это все службы Интернета и наиболее масштабная из них – World Wide Web.

- **Корпоративные ИС.** Они объединяют между собой ИС, функционирующие на базе локальных сетей предприятий одного ведомства,

региона и пр. Корпоративная система управления предприятием (КИС) – это управленческая идеология, объединяющая бизнес-стратегию предприятия и современные информационные технологии. Ведущую роль здесь играет система управления, а автоматизация выполняет второстепенную роль.

По назначению (по выполняемым функциям).

▪ **Информационно-справочные, или информационно-поисковые системы (ИПС)**, являющиеся наиболее старым и традиционным видом ИС. Основная цель в использовании таких систем — оперативное получение ответов на запросы пользователей в диалоговом режиме. Характерным свойством ИПС является большой объем хранимых данных, их постоянная обновляемость. Качество системы определяется скоростью поиска данных в хранилище данных (базе данных) и выдачи ответа. Примером справочной системы является ИПС крупной библиотеки, ИПС в профессиональной области. Поисковые серверы Интернета — это информационно-справочные системы сетевых ресурсов.

▪ **Управляющие системы.** Основное назначение таких систем — выработка управляющих решений. Управляющие системы бывают полностью автоматическими или автоматизированными.

Системы автоматического управления (САУ) работают без участия человека. Это системы управления техническими устройствами, производственными установками, технологическими процессами. Управление в САУ происходит в режиме реального времени. Это значит, что управляющие команды вырабатываются синхронно с управляемым физическим процессом. Поэтому с ростом скорости работы управляемого объекта должно повышаться быстродействие управляющего компьютера.

Автоматизированные системы управления (АСУ) можно назвать человеко-машинными системами. В них компьютер выступает в роли помощника человека-управляющего. В АСУ задача компьютера состоит в оперативном предоставлении человеку необходимой информации для принятия решения. При этом компьютер может выполнять достаточно сложную обработку данных на основании заложенных в него математических моделей. Ограничения на ресурсы времени не такие жесткие, как в автоматических системах. Часто в автоматизированных системах управления в качестве подсистем присутствуют ИПС. Крупные АСУ обеспечивают управление предприятиями, энергосистемами и даже целыми отраслями производства.

▪ **Обучающие системы на базе ЭВМ.**

Простейший вариант такой системы — обучающая программа на ПК или в локальной сети, с которой пользователь работает в индивидуальном режиме. Наиболее сложными и масштабными обучающими системами являются системы дистанционного обучения (образование XXI века), работающие в глобальных сетях. Высокоскоростные системы связи в сочетании с технологией мультимедиа позволяют организовывать обучение в режиме реального времени

(on line), проводить дистанционные лекции, семинары, конференции, принимать зачеты и экзамены.

▪ **Экспертные системы** — системы, основанные на моделях знаний из определенных предметных областей. Экспертные системы (ЭС) иначе называют «Системами искусственного интеллекта». ЭС включает в себе знания высококвалифицированного специалиста в определенной предметной области и используется для консультаций помощи в принятии сложных решений или для решения плохо формализуемых задач (установление диагноза болезни; определение причин неисправности сложной техники (например, космического корабля); выдача рекомендаций по ликвидации неисправности; определение вероятных последствий принятого управляющего решения и так далее). Подобно ИПС, ЭС часто входят в состав АСУ в качестве подсистем.

- **Автоматизированные системы научных исследований (АСНИ).**
- **Справочно-аналитические системы.**
- **Системы автоматизированного проектирования (САПР).**
- **Геоинформационные системы (ГИС) и др.**

3 Принципы организации корпоративных информационных систем

Принципы организации КИС базируются на **стандартах интеграции систем**. Эти стандарты, разработанные американским обществом по контролю за производством и запасами (American Production and Inventory Control Society, **APICS**), обеспечивают интегрированное компьютерное управление.

MRP (Material Requirement Planning) – планирование потребностей в материалах и ресурсах,

MRP II (Manufacturing Resource Planning) – планирование производственных ресурсов,

ERP (Enterprise Resource Planning) – система планирования ресурсов организации,

CRM (Customer Relations Management) – управление отношениями с клиентами,

PRM (Partner Relationship Management) – управление взаимоотношениями с партнерами),

CSRP (Customer Synchronized Resource Planning) – планирование ресурсов организации, синхронизированное на потребителя,

ERP II (Enterprise Resource and Relationship Processing) – управление внутренними ресурсами и внешними связями организации.

В настоящее время большинство КИС строятся на основе стандарта ERP и ERP II, поскольку в этом случае единая интегрированная программа вместо нескольких разрозненных может управлять обработкой, логистикой, дистрибуцией, запасами, доставкой, выставлением счёт-фактур и бухгалтерским учётом. Единая система безопасности, включенная в ERP, позволяет противостоять как внешним угрозам (например,

шпионаж), так и внутренним (например, хищения). Совместно в связке с CRM-системой и системой контроля качества, ERP позволяют максимально удовлетворять потребности клиентов.

Таким образом, можно сделать вывод, что с момента введения первых стандартов интеграции КИС до настоящего времени объем функций, поддерживаемых КИС, значительно расширился, что показано на рисунке 9.1.

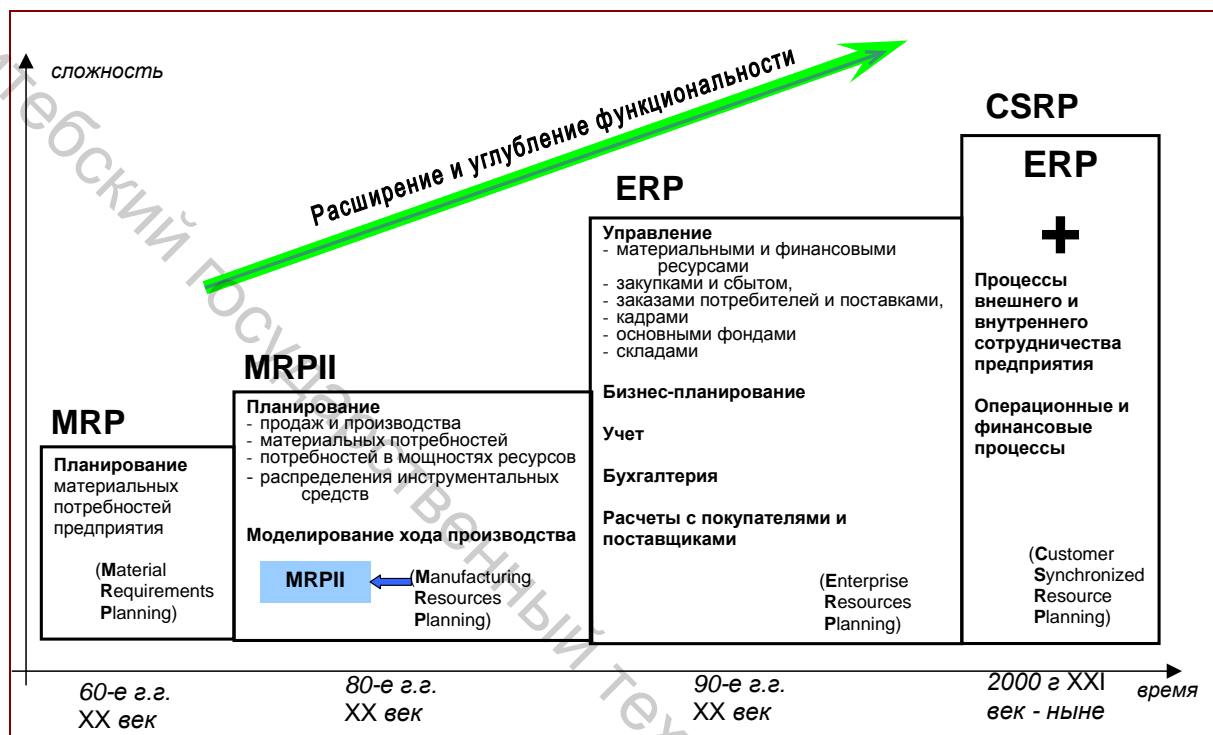


Рисунок 9.1 – Эволюция КИС

4 Современная структура корпоративной информационной системы

Большинство современных КИС строятся на базе модели MRP/ERP. Сегодня модель MRP/ERP включает в себя следующие подсистемы, которые часто называют также блоками или сериями:

Основные модули MRP/ERP-систем (стандарт APICS):

- управление финансами;
- управление материальными потоками;
- управление производством;
- управление проектами;
- управление сервисным обслуживанием;
- управление качеством;
- управление персоналом.

Дополнительные модули ERP (стандарт APICS):

- управление логистическими цепочками – SCM (Supply Chain Management);
- усовершенствованное планирование и составление производственных

графиков – APS (Advanced Planning and Scheduling);

- управление взаимоотношениями с клиентами – CRM (Customer Relationship Management);
- электронная коммерция – EC (Electronic Commerce);
- управления данными об изделии – PDM (Product Data Management) или PLM (Product Lifecycle Management);
- настройка Business Intelligence, включающая инструменты в области интеллектуальной бизнес-аналитики (построение хранилищ данных, OLAP, data mining, визуализация и отчетность) и другие модули.

Пример структуры КИС на базе модели MRP/ERP приведен на рисунке 9.2.

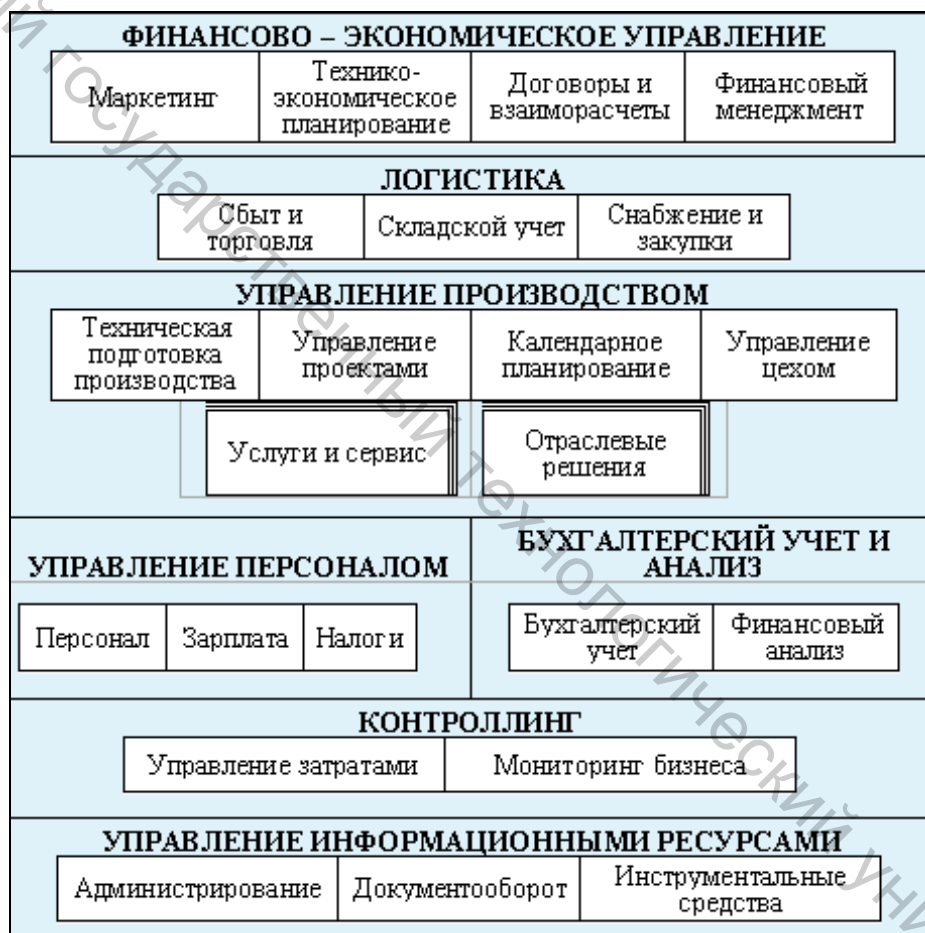


Рисунок 9.2 – Пример структуры КИС на базе модели MRP/ERP

Согласно анализу **Panorama Consulting**¹⁴ по состоянию на 2010 год поставщики ERP-систем разделены на три группы по мере уменьшения доли присутствия на рынке:

- SAP (24 %, системы mySAP ERP, MySAP All-in-One и SAP BusinessOne), Oracle (18 %, Oracle E-Business Suite, JD Edwards и PeopleSoft Enterprise), Microsoft (11 %, Microsoft Dynamics AX (Axapta) и NAV (Navision));

¹⁴ Ведущая мировая независимая консалтинговая фирма по работе с ERP-системами. Расположена в США.

- Epicor, Sage, Infor (COM, MAX+, SSA ERP LN (Baan), SyteLine), IFS (IFS Applications), QAD, Lawson, Ross — 11 % на всех;
- ABAS, Activant Solutions, Baan, Bowen and Groves, Compiere, Exact, Netsuite, Visibility, Blue Cherry, HansaWorld, Intuitive, Syspro.

Российские ERP-системы

- 1С:Управление производственным предприятием 8.0,
- ИС «Галактика»,
- Инталев – Корпорация,
- Фрегат – Корпорация,
- АВА Системы.

Что касается сугубо белорусской специфики ERP – актуальную информацию можно найти на белорусском тематическом сайте www.belerp.com.

5 Информационное обеспечение корпоративных информационных систем

К информационному обеспечению КИС можно отнести разные виды информации, которые хранятся в форме удобной для быстрого доступа, обработки, анализа и передачи.

Отчеты (документы) – для хранения информации, представленной в тестовой и табличной форме.

Автоматизированные банки данных – организационно-техническая система, представляющая собой совокупность баз данных пользователей, технических и программных средств формирования и ведения этих баз и коллектива специалистов, обеспечивающих функционирование системы. К основным функциям банка данных можно отнести: адекватное информационное отображение предметной области, обеспечение хранения, обновления и выдачи необходимых данных пользователям. Составными частями любого банка данных являются база данных, система управления базой данных (СУБД), администратор базы данных, ППО.

Монопольные базы данных (БД) – это поименованная совокупность структурированных данных, относящихся к определенной предметной области, размещенных с возможностью монопольной обработки.

Централизованные базы данных. Информация хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе. Такой способ использования баз данных часто применяют в локальных сетях ПК.

Системы централизованных баз данных с сетевым доступом предполагают различные *архитектуры* подобных систем:

Файл-сервер. Архитектура систем БД с сетевым доступом

предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На сервере хранится совместно используемая централизованная БД. Все другие компьютеры сети выполняют функции рабочих станций, которые поддерживают доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает.

Клиент-сервер. Помимо хранения централизованной базы данных сервер базы данных должен обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные (но не файлы БД) транспортируются по сети от сервера к клиенту. Спецификой архитектуры *клиент-сервер* является использование языка запросов SQL.

Распределенные базы данных. Такие БД состоят из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых на различных компьютерах вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

6 Требования к корпоративным информационным системам

Корпоративная информационная система – это информационная система организации, отвечающая следующему минимальному перечню требований:

1. Функциональная полнота системы:

- выполнение международных стандартов управленческого учета MRP II, ERP, CSRP;
- автоматизация в рамках системы решения задач планирования, бюджетирования, прогнозирования, оперативного (управленческого) учета, бухгалтерского учета, статистического учета и финансового-экономического анализа;
- формирование и ведение учета одновременно по внутренним и международным стандартам;
- количество однократно учитываемых параметров деятельности организации до 1000, количество формируемых таблиц баз данных – до 3000.

2. Надежная система защиты информации:

- парольная система разграничения доступа к данным и реализуемым функциям управления,
- многоуровневая система защиты данных.

3. Наличие инструментальных средств адаптации и сопровождения системы:

- изменение структуры и функций бизнес-процессов,

- изменение информационного пространства,
- изменение интерфейсов ввода, просмотра и корректировки информации,
- изменение организационного и функционального наполнения рабочего места пользователя,
- генератор произвольных отчетов,
- генератор сложных хозяйственных операций,
- генератор стандартных форм.

4. *Реализация удаленного доступа и работы в распределенных сетях.*

5. *Обеспечение обмена данными между разработанными информационными системами и другими программными продуктами, функционирующими в организации.*

6. *Возможность консолидации информации:*

- на уровне организации – объединение информации филиалов, холдингов, дочерних компаний и т. д.,
- на уровне отдельных задач – планирования, учета, контроля и т. д.,
- на уровне временных периодов – для выполнения анализа финансово-экономических показателей за период, превышающий отчетный.

7. *Наличие специальных средств анализа состояния системы в процессе эксплуатации:*

- анализ архитектуры баз данных,
- анализ алгоритмов,
- анализ статистики количества обработанной информации,
- журнал выполненных операций,
- список работающих станций серверов,
- анализ внутрисистемной почты.

Наиболее развитые КИС предназначены для автоматизации всех функций управления корпорацией: от научно-технической и маркетинговой подготовки ее деятельности до реализации ее продукции и услуг. В настоящее время КИС имеют в основном экономическую и производственную направленность.

7 Литература к лекции 9

1. Информационные системы в экономике: учебник для вузов / под ред. Г. А. Титоренко. — 2-е изд., перераб. и доп. — Москва: ЮНИТИ-ДАНА, 2008.

2. Информационные технологии управления: учеб. пособие / под ред. Г. А. Титоренко. — Москва: ЮНИТИ, 2003.

3. Романов, А. Н. Информационные системы в экономике (лекции, упражнения и задачи): учеб. пособие / А. Н. Романов, Б. Е. Одинцов. — Москва: Вузовский учебник, 2006.

4. <http://panorama-consulting.com>.

5. Шарстнев, В. Л. Компьютерные информационные технологии: курс лекций / В. Л. Шарстнев. – Витебск: УО ВГТУ, 2008. – 350 с.