



Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Институт повышения квалификации
и переподготовки кадров

Кафедра «Информатика»

Н. В. Водополова

**МЕНЕДЖМЕНТ
В ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ**

ПОСОБИЕ

**по одноименной дисциплине
для слушателей специальности
1-40 01 73 «Программное обеспечение
информационных систем»
заочной формы обучения**

Гомель 2015

УДК 004:005.311.6(075.8)
ББК 32.81:65.291.21я73
В62

*Рекомендовано научно-методическим советом
факультета автоматизированных систем
ИПК и ПК ГГТУ им. П. О. Сухого
(протокол № 8 от 31.03.2014 г.)*

Рецензенты: доц. каф. «Информационные технологии» ГГТУ им. П. О. Сухого
канд. техн. наук, доц. *В. В. Комраков*

Водополова, Н. В.
В62 Менеджмент в информационных технологиях : пособие по одной дисциплине для слушателей специальности 1-40 01 73 «Программное обеспечение информационных систем» заоч. формы обучения / Н. В. Водополова. – Гомель : ГГТУ им. П. О. Сухого, 2015. – 100 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц; 32 Mb RAM; свободное место на HDD 16 Mb; Windows 98 и выше; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

Содержит учебно-методический материал для выполнения лабораторных работ по разработке автоматизированных рабочих мест специалистов в области управления и принятия решений.

Для слушателей специальности 1-40 01 73 «Программное обеспечение информационных систем» заочной формы обучения ИПК и ПК.

**УДК 004:005.311.6(075.8)
ББК 32.81:65.291.21я73**

© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2015

Оглавление

ВВЕДЕНИЕ	5
РАЗДЕЛ 1. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ПРИНЯТИЯ УПРАВЛЕНЧЕСКИХ РЕШЕНИЙ	5
ИНФОРМАЦИОННАЯ ПОДДЕРЖКА БИЗНЕСА	5
<i>Базовые элементы информационной индустрии</i>	5
<i>Определение и назначение информационных систем</i>	6
<i>Классификация информационных систем</i>	7
<i>Основные принципы создания информационных систем</i>	10
<i>Структура информационных систем</i>	12
СТРУКТУРНО-ФУНКЦИОНАЛЬНЫЙ АНАЛИЗ ТЕХНОЛОГИИ	15
ОРГАНИЗАЦИОННОГО УПРАВЛЕНИЯ	15
<i>Структура системы организационного управления</i>	15
<i>Функциональная модель системы управления и жизненный цикл принятия решения</i>	17
ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ	18
РАЗДЕЛ 2. ПРЕДСТАВЛЕНИЕ ЭКОНОМИЧЕСКИХ ДАННЫХ	19
ОБЩИЕ ИНФОРМАЦИОННЫЕ МЕТОДЫ ОБРАБОТКИ ДАННЫХ В EXCEL	19
ИСТОЧНИКИ И ОРГАНИЗАЦИЯ ДАННЫХ В EXCEL	20
<i>Внутренние источники данных</i>	20
<i>Организация данных</i>	23
МЕТОДИКИ ПОДГОТОВКИ ДАННЫХ К АНАЛИЗУ СРЕДСТВАМИ EXCEL	25
<i>Обработка списков типа "один объект" – "одна запись"</i>	25
<i>Обработка списков типа "один объект" – "несколько записей"</i>	25
ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ	28
РАЗДЕЛ 3. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА АНАЛИЗА И ОБРАБОТКИ ДАННЫХ	29
ПОИСК И ВЫБОРКА ДАННЫХ	29
<i>Критерии поиска и фильтрации данных</i>	32
<i>Автофильтр</i>	35
<i>Расширенный фильтр</i>	38
ЛОГИЧЕСКИЙ АНАЛИЗ ДАННЫХ	46
<i>Логические функции</i>	46
<i>Статистическая функция СЧЕТЕСЛИ()</i>	50
<i>Математическая функция СУММЕСЛИ()</i>	52
РАЗДЕЛ 4. РАЗРАБОТКА АРМ МЕНЕДЖЕРА	55
ИНТЕРФЕЙС «ЧЕЛОВЕК-КОМПЬЮТЕР»	55
<i>Пользовательский интерфейс</i>	55
<i>Структурные элементы интерфейса</i>	56
<i>Критерии качества диалога</i>	58
<i>Типы диалогов и их реализация</i>	59
ФОРМАТИРОВАНИЕ ЭКРАНА	61
<i>Этапы размещения данных и общие принципы расположения информации на экране</i>	61
<i>Выбор выводимой на экран информации</i>	62
<i>Формат вывода данных</i>	63
<i>Общие принципы расположения информации</i>	63
<i>Выделение информации на экране</i>	65
АВТОМАТИЗАЦИЯ ПРОЕКТА С ПОМОЩЬЮ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ	66
<i>Организация работы с элементами управления</i>	66
<i>Элемент управления Кнопка</i>	72
ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ	74
РАЗДЕЛ 5. МЕНЕДЖМЕНТ ПРОГРАММНЫХ ИЗДЕЛИЙ	75
МЕНЕДЖЕР В РАЗРАБОТКЕ ПРОГРАММНЫХ ИЗДЕЛИЙ	75
<i>Жизненный цикл программного изделия</i>	76
<i>Модели жизненного цикла программного обеспечения</i>	77

РАЗРАБОТКА ПРОЕКТА	83
<i>Менеджмент проекта</i>	83
<i>Планирование задач проекта</i>	86
<i>Работа с таблицами и представлениями проекта</i>	95

Библиотека ГГТУ им. П.О.Суворова

Введение

Курс лекций по дисциплине «Менеджмент в информационных технологиях» разработан для слушателей факультета повышения и переподготовки кадров. Данный курс ориентирован на лиц, которые в своей профессиональной деятельности постоянно сталкиваются с решением большого количества и разнообразных задач в области принятия управленческих решений, а также занимаются проблемами организации взаимодействия групп разработчиков, совместно реализующих одну большую задачу.

Управление любого предприятия основано на фундаментальных принципах организационного управления. С точки зрения информационных технологий и их применения в решении управленческих задач различают оперативные, тактические и стратегические задачи, которые реализуют соответствующие методы, технологии и инструменты. В соответствии с этим в данном курсе рассмотрены инструментальные средства обработки и анализа оперативной информации, основные принципы разработки автоматизированных рабочих мест специалистов, технологии и подходы интеллектуального анализа данных и, наконец, программный продукт, который может быть полезен при управлении работой большого коллектива разработчиков или специалистов.

Раздел 1. Информационные технологии принятия управленческих решений

Информационная поддержка бизнеса

Базовые элементы информационной индустрии

В истории развития цивилизации произошло несколько информационных революций – преобразований общественных отношений из-за кардинальных изменений в сфере обработки информации. Следствием подобных преобразований являлось приобретение человеческим обществом нового качества.

Первая революция связана с изобретением письменности, что позволило передавать знания.

Вторая (середина XVIв.) - вызвана изобретением книгопечатания, которое радикально изменило организацию деятельности человека.

Третья (конец XIXв.) обусловлена изобретением электричества, благодаря которому появились телеграф, телефон, радио, позволяющие оперативно и передавать и накапливать информацию в любом объеме.

Четвертая (70-е гг. XXв.) связана с изобретением микропроцессорной технологии и появлением персонального компьютера. Это послужило толчком для развития новой отрасли – *информационной индустрии*. Важнейшими составляющими ее являются *информационные технологии* и *телекоммуникации*. В свою очередь развитие информационных технологий послужило толчком для развития *информационное общество*.

Информационная технология – процесс, использующий совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления.

Телекоммуникации – дистанционная передача данных на базе компьютерных сетей и современных технических средств связи.

Информационное общество – общество, в котором большинство работающих занято производством, хранением, переработкой и реализацией информации, особенно высшей ее формы – знаний.

Для создания информационного общества необходимы три составляющие:

1. *Материальная база;*
2. *Технологическая база*, элементом которой является информационная система;
3. *Информационная культура*, предполагающая
 - умение работать с информацией, т.е. умение выбирать именно ту ее часть, которая необходима и самодостаточна для решения поставленной задачи;
 - владение техническими средствами и методами для получения, обработки и передачи информации;
 - использование в профессиональной деятельности компьютерных технологий.

Определение и назначение информационных систем

Информационная система – взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Современное понимание информационной системы предполагает использование в качестве основного технического средства переработки информации вычислительную технику. Однако, техническое воплощение информационной системы само по себе ничего не значит, если не учтена роль человека, для которого собственно и предназначена производимая информация и без которого невозможно ее получение и представление.

Процессы, обеспечивающие работу информационной системы любого назначения, условно можно представить в виде схемы (рис.1.1).

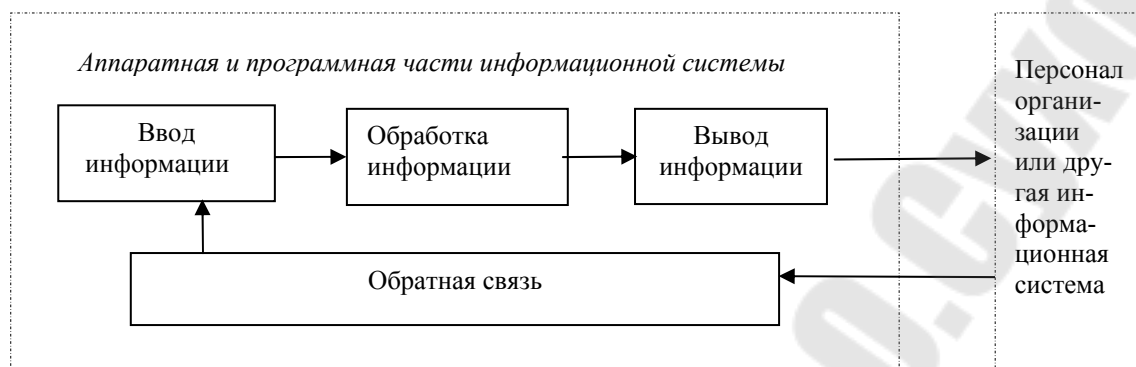


Рис.1.1

Назначение основных блоков:

- ввод информации из внешних или внутренних источников;
- обработка входной информации и представление ее в удобном виде;
- вывод информации для представления потребителям или передачи в другую информационную систему;
- обратная связь – это информация, переработанная людьми данной организации или другой информационной системой для коррекции входной информации.

Классификация информационных систем

Вся совокупность информационных систем по степени автоматизации процессов обработки информации в них может быть представлена тремя основными группами:

- ручные,
- автоматические и
- автоматизированные информационные системы.

Ручные информационные системы характеризуются отсутствием современных технических средств переработки информации и выполнением всех операций человеком. Например, о деятельности менеджера в фирме, где отсутствуют компьютеры, можно говорить, что он работает с ручной информационной системой.

Автоматические информационные системы выполняют все операции по переработке информации без участия человека. Например, система автоматического пилотирования самолета, системы управления атомными реакторами и др.

Автоматизированные информационные системы предполагают участие в процессе обработки информации и человека, и технических средств,

причем главная роль отводится компьютеру. В современном толковании в термин «Информационная система» вкладывается обязательно понятие автоматизированной системы.

Автоматизированные информационные системы, учитывая их широкое использование в организации процессов управления, имеют различные модификации, например, по характеру использования информации и по сфере применения.

Классификация информационных систем по характеру использования информации

По характеру использования информации выделяют *информационно-поисковые* и *информационно-решающие* системы.

Информационно-поисковые системы производят ввод, систематизацию, хранение, выдачу информации по запросу пользователя без сложных преобразований данных. Например, информационно-поисковая система в библиотеке, в железнодорожных и авиационных кассах продажи билетов.

Информационно-решающие системы осуществляют все операции переработки информации по определенному алгоритму. Среди них по степени воздействия выработанной информации на процесс принятия решений выделяют *управляющие* и *советующие* системы.

Управляющие информационные системы вырабатывают информацию, на основании которой человек принимает решения. Для этих систем характерны тип задач расчетного характера и обработка больших объемов данных. Например, система оперативного планирования выпуска продукции, система бухгалтерского учета и др.

Советующие информационные системы вырабатывают информацию, которая принимается человеком к сведению и не превращается немедленно в серию конкретных действий. Эти системы обладают более высокой степенью интеллекта, т.к. для них характерна обработка знаний, а не данных. Например, медицинская информационная система для постановки диагноза больного, система анализа кредитоспособности заемщика и др.

Человек при работе с такой системой может принять к сведению полученную информацию, но предложить иное по сравнению с рекомендуемым решение.

Классификация информационных систем по сфере применения

Различают

- интегрированные системы;
- системы организационного управления;
- системы управления технологическими процессами и

- системы автоматизированного проектирования (САПР).

Интегрированные (корпоративные) информационные системы используются для автоматизации всех функций фирмы и охватывают весь цикл работ от проектирования до сбыта продукции. Создание таких систем весьма затруднительно, т.к. требует системного подхода с позиций главной цели, например, получения максимальной прибыли, завоевания рынка сбыта и т.д. Такой подход может привести к существенным изменениям в самой структуре управления фирмой, на что может решиться не каждый управляющий.

Информационные системы организационного управления предназначены для автоматизации функций управленческого персонала. Широкое применение и разнообразие этого класса систем привело к тому, что часто любые информационные системы понимают именно в данном толковании. К этому классу относятся информационные системы управления как промышленными фирмами, так и непромышленными объектами: гостиницами, банками, торговыми организациями и др.

Основными функциями этих систем являются:

- оперативный контроль и регулирование;
 - оперативный учет и анализ;
 - перспективное и оперативное планирование;
 - бухгалтерский учет;
 - учет сбытом и снабжением
- и другие экономические и организационные задачи.

Информационные системы управления технологическими процессами служат для автоматизации функций производственного персонала. Они широко используются при организации поточных линий, на сборке, для поддержания технологического процесса в металлургической и машиностроительной промышленности.

Информационные системы автоматизированного проектирования предназначены для автоматизации функций инженеров-проектировщиков, конструкторов, архитекторов, дизайнеров при создании новой техники или технологии. Основными функциями подобных систем являются:

- инженерные расчеты;
- создание графической документации (схем, чертежей, планов);
- создание проектной документации;
- моделирование проектируемых объектов.

Основные принципы создания информационных систем

Разработчики информационных систем придерживаются следующих принципов их создания.

1. *Принцип первого руководителя.* Если задача автоматизации ставится локально, например, необходимо автоматизировать труд в бухгалтерии, то, естественно, такая задача может быть поручена бухгалтерам, а главный руководитель может не вникать непосредственно в суть работ. Если же речь идет о принципиально новых возможностях, которые открываются перед организацией управления, такие задачи не могут быть решены без первого руководителя.

2. *Принцип системного подхода к управлению.* Он предполагает тщательный анализ функционирования управляемой системы (завода, отрасли и т.д.) в целом. Особое внимание следует уделять согласованию критериев, методов работы органов управления и задач, решаемых в отдельных автоматизированных и информационных системах.

3. *Принцип этапности создания и внедрения систем* управления предполагает необходимость предусмотреть с самого начала работ определенную этапность их создания и внедрения. Это объясняется большими сроками, значительными людскими и финансовыми затратами, сложностью выполнения работ и невозможностью в ограниченные сроки перестройки всей организации управления.

Обычно внедрение первых этапов автоматизированной системы управления осуществляется в недрах существующей системы управления, что допускает в течение некоторого времени параллельное функционирование двух систем. На этих этапах проверяется и дорабатывается автоматизированная система, переподготавливается административный персонал, преодолевается недоверие к новой системе.

Создание автоматизированной системы административно-организационного управления целесообразно начинать с объектов, относящихся к верхнему уровню иерархии управления, постепенно распространяя ее на объекты нижних уровней. То есть используется иерархический подход. При этом

- облегчается согласование дерева целей (задач), возлагаемых на систему, с иерархией объектов, входящих в систему;
- обеспечивается унификация, стандартизация и сокращение документооборота;
- разрабатываются общие принципы кодирования информации;
- достигается единство технической базы и унификации математического, программного и других видов обеспечения информационной системы.

4. *Принцип повышения оперативности управления и решения новых задач* предусматривает определение главных путей достижения эффективности внедрения автоматизированных систем управления.

Повышение оперативности управления путем сокращения времени сбора и анализа информации, разработки вариантов решений и планов, доведения решений до исполнителей и контроля исполнения позволяет

- улучшить реакцию системы на возможные отклонения условий функционирования объектов управления;
- сократить всевозможные потери и объем необходимых запасов;
- повысить эффективность функционирования системы в целом.

Решение новых задач предполагает широкое внедрение современных методов математического моделирования, экономико-математических методов планирования, методов математической оптимизации, реализации которых невозможно без использования современных средств обработки данных. Решение новых задач должно повысить качество принимаемых решений и планов их выполнения.

5. *Принцип унификации и стандартизации документооборота* и всех видов обеспечения автоматизированной системы управления отражает одно из важнейших условий минимизации затрат на создание системы и ее эксплуатацию. Он связан с требованием разработки единых способов кодирования информации, формализации документов, способов адресации, представления и обращения к информации.

Перечисленные принципы создания и внедрения автоматизированных систем управления являются основными в руководстве работами со стороны заказывающих органов и будущих потребителей этих систем. Основными критериями выбора информационной системы являются их свойства и эффект от внедрения.

Информационная система должна характеризоваться и обладать следующими свойствами:

- любая информационная система может быть подвергнута анализу, должна быть построена и управляема на основе общих принципов построения систем;
- информационная система должна быть динамичной и развивающейся;
- выходной продукцией информационной системы должна быть информация, на основе которой могут быть приняты решения;
- информационную систему следует воспринимать как человеко-компьютерную систему обработки информации, т.е. у нее должен быть хороший и удобный пользовательский интерфейс.

Внедрение информационных систем должно способствовать

- получению более рациональных вариантов решения управленческих задач за счет внедрения математических методов и интеллектуальных систем и т.д.;
- освобождению работников от рутинной работы за счет ее автоматизации;
- обеспечению достоверности и надежности информации;
- рациональной организации переработки информации и снижению объемов документов на бумаге и документооборота;
- совершенствованию структуры потоков информации и системы документооборота;
- уменьшению затрат на производство продуктов и услуг и т.д.

Структура информационных систем

Структуру информационной системы составляет совокупность отдельных ее частей, называемых *подсистемами*.

Подсистема – это часть системы, выделенная по какому-либо признаку.

Общую структуру информационной системы можно рассматривать как совокупность подсистем независимо от сферы применения. В этом случае говорят о *структурном признаке* классификации, а подсистемы называют *обеспечивающими*. Таким образом, структура любой информационной систем может быть представлена совокупностью обеспечивающих систем.

Среди обеспечивающих подсистем обычно выделяют

- информационное;
- техническое;
- математическое;
- программное;
- организационное;
- правовое обеспечение.

Информационное обеспечение

Назначение подсистемы – своевременное формирование и выдача достоверной информации для принятия управленческих решений.

Информационное обеспечение – совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, а также методология построения баз данных.

Для создания информационного обеспечения необходимо:

- ясное понимание целей, задач, функций всей системы управления организацией;
- выявление движения информации от момента возникновения и до ее использования на различных уровнях управления, представленной для анализа в виде схем информационных потоков;
- совершенствование системы документооборота;
- наличие и использование системы классификации и кодирования;
- владение методологией создания концептуальных информационно-логических моделей, отражающих взаимосвязь информации;
- создание массивов информации на машинных носителях, что требует наличия современного технического обеспечения.

Техническое обеспечение

Техническое обеспечение – комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Комплекс технических средств составляют:

- компьютеры любых моделей;
- устройства передачи данных и линий связи;
- оргтехника и устройства автоматического съема информации;
- эксплуатационные материалы и др.

К настоящему времени сложились две основные формы организации технического обеспечения (формы использования) технических средств: централизованная и частично или полностью децентрализованная.

Централизованное техническое обеспечение базируется на использовании в информационной системе больших ЭВМ и вычислительных центров.

Децентрализация технических средств предполагает реализацию функциональных подсистем на персональных компьютерах непосредственно на рабочих местах.

Частично децентрализованный подход – новая перспективная форма организации технического обеспечения на базе распределенных сетей, состоящих из персональных компьютеров и большой ЭВМ для хранения баз данных, общих для любых функциональных подсистем.

Математическое и программное обеспечение

Математическое и программное обеспечение – совокупность математических методов, моделей, алгоритмов и программ для реализации це-

лей и задач информационной системы, а также нормального функционирования комплекса технических средств.

К средствам математического обеспечения относятся:

- средства моделирования процессов управления;
- типовые задачи управления;
- методы математического программирования, математической статистики, теории массового обслуживания и др.

В состав программного обеспечения входят

- общесистемные и специальные программные продукты;
- техническая документация.

Общесистемное программное обеспечение – это комплексы программ, ориентированных на пользователей и предназначенных для решения типовых задач обработки информации. Они служат для расширения функциональных возможностей компьютеров, контроля и управления процессом обработки данных. Число всех разновидностей системных программ очень велико. Это операционные системы, драйверы, программы-оболочки, операционные оболочки.

Специальное программное обеспечение представляет собой совокупность программ, разработанных при создании конкретной информационной системы. В его состав входят пакеты прикладных программ. Наиболее широко применяются редакторы текстов, табличные процессоры, системы управления базами данных, графические редакторы и т.д.

Техническая документация на разработку программных средств должна содержать описание задач, задание на алгоритмизацию, экономико-математическую модель задачи, контрольные примеры.

Организационное обеспечение

Организационное обеспечение – совокупность методов и средств, регламентирующих взаимодействие работников с техническими средствами и между собой в процессе разработки и эксплуатации информационной системы.

Организационное обеспечение реализует следующие функции:

- анализ существующей системы управления организацией, где будет использоваться информационная система, и выявление задач, подлежащих автоматизации;
- подготовку задач к решению на компьютере, включая техническое задание на проектирование информационной системы и технико-экономическое обоснование ее эффективности;

– разработку управленческих решений по составу и структуре организации, методологии решения задач, направленных на повышение эффективности системы управления.

Организационное обеспечение создается по результатам предпроектного обследования на первом этапе построения баз данных.

Правовое обеспечение

Правовое обеспечение – совокупность правовых норм, определяющих создание, юридический статус и функционирование информационных систем регламентирующих порядок получения, преобразования и использования информации.

В состав правового обеспечения входят законы, указы, постановления, приказы и т.д. В правовом обеспечении можно выделить общую часть, регулирующую функционирование любой информационной системы, и локальную часть, регламентирующую функционирование конкретной системы.

С другой точки зрения в правовом обеспечении можно выделить обеспечение

– этапов разработки информационной системы (включает нормативные акты, связанные с договорными отношениями разработчика и заказчика и правовом регулировании отклонений от договора);

– этапов функционирования информационной системы. Оно включает статус информационной системы; права, обязанности и ответственность персонала; правовые положения отдельных видов процесса управления; порядок создания и использования информации и др.

Структурно-функциональный анализ технологии организационного управления

Структура системы организационного управления

Построение информационной системы можно сравнить с постройкой дома. Кирпичи, гвозди, цемент и прочие материалы, сложенные вместе, не дают дома. Нужны проект, землеустройство, строительство и др., чтобы появился дом.

Аналогично для создания и использования информационной системы необходимо сначала понять структуру, функции и политику организации, цели управления и принимаемых решений, возможности компьютерной технологии. Информационная система является частью организации, а ключевые элементы любой организации – структура и органы управления, стандартные процедуры, персонал.

Построение информационных систем должны начинаться с анализа структуры управления организацией.

Структура управления любой организации (рис.1.2) традиционно делится на три уровня:

- операционный;
- функциональный;
- стратегический.

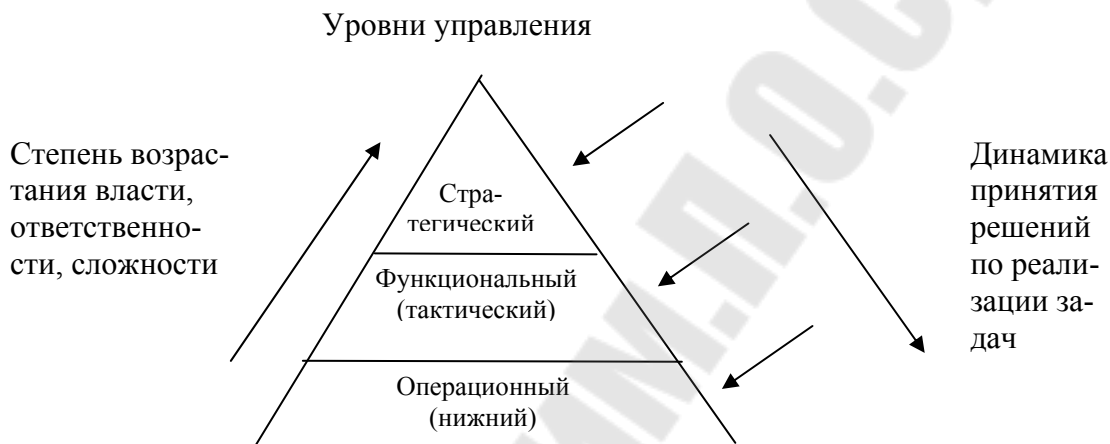


Рис.1.2.

Уровни управления (вид управленческой деятельности) определяются сложностью решаемых задач. Чем сложнее задача, тем более высокий уровень управления требуется для ее решения. При этом следует понимать, что более простых задач, требующих немедленного (оперативного) решения, возникает значительно большее количество, следовательно, уровень управления для них нужен уже другой – более низкий, где принимаются решения оперативно. При управлении необходимо также учитывать динамику реализации принимаемых решений, что позволяет рассматривать управление под углом временного фактора.

Операционный (нижний) уровень управления обеспечивает решение многократно повторяющихся задач и операций и предполагает быстрое реагирование на изменения входной текущей информации. На этом уровне достаточно велики как объем выполняемых операций, так и динамика принятия управленческих решений. Этот уровень управления часто называют *оперативным* из-за необходимости быстрого реагирования на изменение ситуации. На этом уровне большой объем занимают учетные задачи (учет произведенной продукции, складской учет, бухгалтерский учет и т.д.).

Функциональный (тактический) уровень управления обеспечивает решение задач, требующих предварительного анализа информации, подготовленной на первом уровне. На этом уровне большое значение приобретает такая функция управления, как анализ. Объем решаемых задач уменьшается, но возрастает их сложность. При этом не всегда удается вы-

работать нужное решение оперативно, требуется дополнительное время на анализ, осмысление, сбор недостающих сведений и т.д. Управление связано с некоторой задержкой от момента поступления информации до принятия решений и их реализации, а также от момента реализации решений до получения реакции на них.

Пример. *На основании анализа статистических данных по спросу на продукцию, о ценах конкурентов и пр. прогнозируется прибыль и разрабатывается план выпуска продукции на неделю, месяц, квартал. Результаты принимаемых управленческих решений проявляются спустя некоторое время.*

Стратегический уровень обеспечивает выработку управленческих решений, направленных на достижение долгосрочных стратегических целей организации. Поскольку результаты принимаемых решений проявляются спустя длительное время, особое значение на этом уровне имеет такая функция управления, как стратегическое планирование. Часто этот уровень называют *стратегическим* или *долгосрочным планированием*. Правомерность принятого на этом уровне решения может быть подтверждена спустя достаточно длительное время. Ответственность за принятие управленческих решений чрезвычайно велика и определяется не только результатами анализа с использованием математического и специального аппарата, но и профессиональной интуиции менеджеров.

Пример. *На основании анализа финансового состояния фирмы принимаются решения об увеличении (уменьшении, снятии с продажи) производимой продукции, о привлечении дополнительных работников или об их сокращении.*

Функциональная модель системы управления и жизненный цикл принятия решения

Координация работы всех подразделений организации осуществляется через органы управления разного уровня. Под *управлением* понимают обеспечение поставленной цели при условии реализации следующих функций: организационной, плановой, учетной, анализа, контрольной, стимулирования.

Организационная функция заключается в разработке организационной структуры и нормативных документов: штатного расписания, должностных инструкций и т.д.

Планирование состоит в разработке и реализации планов по выполнению поставленных задач. Например, бизнес-план, план производства, финансовый план и т.д.

Учетная функция заключается в разработке или использовании готовых форм и методов учета показателей деятельности фирмы: бухгалтерский учет, финансовый учет и т.д.

Анализ или *аналитическая функция* – изучение итогов выполнения планов и заказов; определение влияющих факторов, резервов; изучение тенденций и т.д.

Контрольная функция осуществляется менеджером: контроль за выполнением планов, расходованием материальных ресурсов и финансовых средств и т.д.

Стимулирование или *мотивационная функция* предполагает разработку и применение различных методов стимулирования труда работников: финансовые стимулы (зарплата, премия, повышение в должности и т.д.); психологические (благодарности, грамоты, звания и т.д.).

В последнее время в сфере управления все активнее применяется понятие «*принятие решения*».

Принятие решения – акт целенаправленного воздействия на объект управления, основанный на анализе ситуации, определении цели, разработке программы достижения этой цели.

Таким образом, в жизненном цикле принятия любого решения можно выделить следующие основные моменты:

- планирование;
- организация;
- контроль (учет, оценка, анализ)
- регулирование.

То есть в данном случае добавляется новая функция: регулирование. Ее назначение – на основании результатов анализа делается вывод: будет достигнут результат, если нет, то что необходимо изменить, чтобы цель была достигнута.

Но в основе управляющего воздействия на объект управления лежат данные анализа.

Вопросы для самоконтроля

1. Базовые элементы информационной индустрии
2. Определение и назначение информационных систем
3. Классификация информационных систем
4. Основные принципы создания информационных систем
5. Структура информационной системы
6. Структура системы организационного управления
7. Функциональная модель системы управления жизненным циклом принятия решений

Раздел 2. Представление экономических данных

Общие информационные методы обработки данных в Excel

Анализ – это, прежде всего, ежедневная кропотливая, трудоемкая и очень ответственная работа. Качественно выполнять ее помогает одно из самых распространенных в последнее время программных продуктов: табличный процессор Excel.

Информационные методы обработки данных напрямую связаны с основными компонентами информационной технологии обработки данных. Это:

- сбор и хранение данных;
- обработка данных;
- получение отчетов.

В соответствии с ними все информационные методы обработки данных могут быть условно классифицированы как методы

- 1) организации и хранения данных;
- 2) подготовки данных к анализу;
- 3) анализа данных;
- 4) создания качественных отчетов.

Проблемы организации и хранения данных в Excel связаны с решением вопросов а) об источниках данных (внутренних и внешних) и б) о способах их организации.

Методы и способы подготовки данных к анализу зависят от способа их организации и осуществляются путем

а) сжатия данных (вертикальное удаление) с помощью

- сортировки;
- фильтрации;
- механизма "Промежуточные итоги";
- механизма "Сводная таблица";
- группировки.

б) дополнения данных с помощью формул

в) удаления лишних данных (горизонтальное удаление) с помощью

- простого удаления;
- скрытия деталей;
- расширенного фильтра;
- механизма "Сводная таблица".

Анализ данных предполагает использование

- логических, математических и статистических функций;
- вычисляемого критерия расширенного фильтра;
- фильтрации данных;

Вопросы организации работы с внешними данными рассматриваются как самостоятельная тема в следующем разделе.

Внутренними являются данные, размещенные в одной рабочей книге Excel. Они могут быть организованы в виде:

- базы данных;
- списка;
- таблицы;
- диапазона ячеек.

С точки зрения организации данных дадим следующие определения.

База данных – это специальным образом организованные исходные данные.

Список – специальным образом организованные исходные и расчетные данные.

Как видим из определений, и база данных, и список должны быть организованы специальным образом, т.е. в соответствии со следующими правилами:

1. Верхняя строка должна содержать заголовки полей. Заголовок поля может располагаться только в одной ячейке.

Соблюдение этого правила необходимо для правильной работы таких механизмов Excel, как "Сортировка", "Форма данных", "Автофильтр", "Расширенный фильтр", "Промежуточные итоги", "Сводная таблица", "Мастер диаграмм".

2. Диапазон ячеек, составляющих базу данных или список, должен быть ограничен пустыми ячейками или границами рабочего места. Т.е. не допускается:

- наличие внутри базы данных или списка пустых строк или столбцов;
- примыкание к базе данных или списку других данных.

Соблюдение этого правила позволяет при работе с механизмами "Промежуточные итоги", "Сводная таблица" и сортировки получать автоматическое выделение всех ячеек, составляющих базу данных или список.

На рис. 2.2 и 2.3 представлены фрагменты базы данных и списка соответственно. В данном примере список отличается от базы тех же данных наличием расчетных показателей "Прирост" и "Доля".

	A	B	C	D
1	Средства производства	На начало года, млн.руб.	На конец года, млн.руб.	
2	Основные средства	10150	10800	
3	Долгосрочные финансовые вложения	2500	3000	
4	Незавершенное строительство	2850	3300	
5	Нематериальные активы	2200	1700	
6				
7				
8				

Рис.2.2. Пример базы данных "Состав и динамика основного капитала"

	A	B	C	D	E	F	G	H
1	Средства производства	На начало года, млн.руб.	Доля на начало года,%	На конец года, млн.руб.	Доля на конец года,%	Прирост, млн.руб.	Прирост, %	
2	Основные средства	10150	57,3	10800	57,4	650	0,1	
3	Долгосрочные финансовые вложения	2500	14,1	3000	16,0	500	1,9	
4	Незавершенное строительство	2850	16,1	3300	17,6	450	1,5	
5	Нематериальные активы	2200	12,5	1700	9,0	-500	-3,5	
6								
7								

Рис.2.3. Пример списка "Состав и динамика основного капитала"

Таблица – это исходные и расчетные данные специальным образом оформленные.

Специальное оформление данных может быть связано с введением совмещенных заголовков граф, строки с номерами граф и графы "Номер по порядку", подножия таблицы, специального форматирования данных, (выравнивание, представление, выделение и т.д.).

Пример таблицы с теми же данными, что и в списке (рис.2.3) представлен на рис.2.4.

	A	B	C	D	E	F	G	H
1								
2								Таблица
3	Состав и динамика основного капитала							
4		На начало года		На конец года		Прирост		
5	Средства производства	млн.руб.	доля,%	млн.руб.	доля,%	млн.руб.	доля,%	
6	A	1	2	3	4	5	6	
7	Основные средства	10150	57,3	10800	57,4	650	0,1	
8	Долгосрочные финансовые вложения	2500	14,1	3000	16,0	500	1,9	
9	Незавершенное строительство	2850	16,1	3300	17,6	450	1,5	
10	Нематериальные активы	2200	12,5	1700	9,0	-500	-3,5	
11	Итого	17700	100	18800	100	1100		
12								
13								

Рис.2.4. Пример таблицы "Состав и динамика основного капитала"

Диапазон ячеек – совокупность смежных ячеек, расположенных в прямоугольной области рабочего листа, которые содержат какие-либо данные.

Как правило, данные диапазона ячеек в анализе не используются, поскольку их организация не соответствует указанным выше требованиям.

Итак, организация данных всех четырех типов внутренних источников данных имеет следующие общие особенности: в строке могут располагаться разнородные данные, а в столбце (поле, графе) – только однородные. Причем, только в диапазоне ячеек имя поля или заголовков отсутствует.

Во-вторых, для хранения данных используют организацию данных в виде базы данных, т.к. хранение расчетных данных расточительно.

В-третьих, для анализа могут быть использованы данные базы данных, списка и за редким исключением таблицы.

В-четвертых, для оформления результатов анализа данные представляют в виде таблицы, а для наглядности в некоторых случаях иллюстрируют их с помощью графиков и диаграмм.

Поскольку база данных является частным случаем списка, в дальнейшем будем ссылаться только на список.

Организация данных

Возможны два варианта организации данных в Excel:

- "один объект" – "одна запись"
- "один объект" – несколько записей"

Определим следующие понятия.

Информационный объект – это совокупность сведений об объекте материального или нематериального мира. Каждый объект характеризуется свойствами (реквизитами), которые позволяют выделить его из множества других. Например, объект "Автомобиль" может иметь реквизиты: цвет, вес, марка, номер, фамилия владельца.

Запись – строка списка.

Первичный (главный) ключ списка – это поле или группа полей, с помощью которых можно однозначно идентифицировать запись. Значение первичного ключа не должно повторяться у разных записей.

Для списка, представленного на рис.2.5, главным ключом является поле "Студент", поскольку значения его однозначно идентифицируют отдельную запись списка. В этом случае один объект описывается одной записью.

	А	В	С	Д	Е
1	Студент	Математика	Информатика	История	
2	Иванов И.И.	4	3	4	
3	Петров П.П.	5	5	5	
4	Сидоров С.С.	3	3	4	
5					
6					
7					

Рис.2.5. Пример списка типа "один объект" –

В списке рис.2.6. главным ключом является группа полей: "Студент" + "Предмет". Объект – это студент и информация о нем. Каждому объекту (студенту) в этом случае соответствует несколько записей: столько, сколько предметов изучал студент. В этом случае имеет место тип организации списка "один объект" – "несколько записей". Следует отметить, что этот список может быть обработан по тому же ключу, но в качестве объекта выбраны значения поля "Предмет". В этом случае каждому объекту будет соответствовать столько записей, сколько студентов изучало конкретный предмет.

	А	В	С	Д
1	Студент	Предмет	Оценка	
2	Иванов И.И.	математика	4	
3	Петров П.П.	математика	5	
4	Иванов И.И.	информатика	5	
5	Смирнов С.И.	экономика	4	
6	Сидоров С.С.	история	4	
7	Петров П.П.	информатика	5	
8				

Рис.2.6. Пример списки типа "один объект" - "несколько записей"

Добавим в список рис.2.5 новое поле "Группа" (рис.2.7). В данном случае главным ключом будет группа полей: "Группа" + "Студент". Если в качестве объекта взять значения того же поля "Студент", что и в примере рис.2.5, то получим с точки зрения организации списка тот же тип: "один объект" – "одна запись". Но поскольку главным ключом является группа полей, то в качестве объекта можно взять значения и другого поля: "Группа". В этом случае объекту по полю "Группа" будет соответствовать несколько записей – столько, сколько студентов учится в данной группе. Т.е. присутствует тип организации "один объект" – "несколько записей".

Таким образом, если в качестве главного ключа выступает группа полей, то возможна организация данных смешанного типа. И зависеть она будет от выбора объекта анализа.

	А	В	С	Д	Е	Ф
1	Группа	Студент	Математика	Информатика	История	
2	МГ-11	Иванов И.И.	4	3	4	
3	МГ-11	Петров П.П.	5	5	5	
4	МГ-12	Сидоров С.С	3	3	4	
5						
6						
7						

Рис.7. Пример списка комбинированного типа

Примеры организации и способы представления данных приведены в дополнительном файле «*Организация данных.xls*».

Методики подготовки данных к анализу средствами Excel

Обработка списков типа "один объект" – "одна запись"

В данном случае подготовка к анализу в основном сводится к сжатию с помощью фильтрации, сортировке и дополнению исходных данных расчетными значениями. Исключением является случай использования механизма "Сводная таблица", поскольку он сам осуществляет фильтрацию данных (область СТРАНИЦА) и их автоматическую сортировку при составлении отчета.

Пример. Найти средний балл каждого студента. Информация о результатах сдачи сессии представлена в списке рис.2.5.

Решить эту задачу возможно только с помощью ввода расчетной формулы среднего балла.

Последовательность действий:

- в строке заголовков полей вводится новый заголовок "Средний балл" (ячейка E1);

- в ячейку E2 вводится формула:
=СРЗНАЧ (B2:D2)

- формула копируется с ячейки E3 и до конца списка.

Этот вариант отличает очевидная простота. Однако он требует достаточно большого объема ручной работы.

Обработка списков типа "один объект" – "несколько записей"

Для экономических данных это самый распространенный способ их организации и хранения. Однако он диктует практически всегда выполнение некоторых дополнительных действий по подготовке данных к анализу. Прежде всего, это – сортировка и фильтрация данных.

Решим пример 1 с учетом организации данных в виде списка, представленного на рис.2.6.

Вариант а). Ввод расчетной формулы "средний балл" невыполним.

Вариант б). Использование механизма "Промежуточные итоги".

Необходимо:

- отсортировать данные по двум ключам "Студент" и "Предмет".

Если этого не сделать, получим неверный ответ: по ст.Иванову И.И. и Петрову П.П. будет выдано 2 средних оценки – после 1-ой и 3-ей, 2-ой и 6-ой записями соответственно;

- обратиться к механизму "*Промежуточные итоги*"

- заполнить диалоговое окно:

"При каждом изменении в " - *Студент*

"Операция" - *среднее*

"Добавить итоги по " - *Оценка*

Результат представлен на рис.2.8.



	А	В	С	Д
1	Студент	Предмет	Оценка	
2	Иванов И.И.	математика	4	
3	Иванов И.И.	информатика	5	
4	Иванов И.И. Среднее		4,5	
5	Петров П.П.	информатика	5	
6	Петров П.П.	математика	5	
7	Петров П.П. Среднее		5	
8	Сидоров С.С.	история	4	
9	Сидоров С.С. Среднее		4	
10	Смирнов С.И.	экономика	4	
11	Смирнов С.И. Среднее		4	
12	Общее среднее		4,5	

Рис.2.8. Результат решения примера 1 с помощью механизма "Промежуточные итоги"

Вариант в). Использование механизма "*Сводная таблица*".

Необходимо:

- вызвать Мастер сводной таблица (выделить любую ячейку списка, *Данные - Сводная таблица*);

- последовательно заполнить диалоговые окна;

- структура таблицы: область Строка – *Студент*
область Данные – *Оценка*

- Операция – *среднее*

Результат представлен на рис.2.9.

Книга1			
	А	В	С
1	Среднее по полю Оценка		
2	Студент	Всего	
3	Иванов И.И.	4,5	
4	Петров П.П.	5	
5	Сидоров С.С.	4	
6	Смирнов С.И.	4	
7	Общий итог	4,5	
8			

Рис.9. Результат решения примера 1 с помощью механизма "Сводная таблица"

Пример. Данные организованы в виде списка, представленного на рис.2.10. Необходимо составить отчет о результатах сдачи сессии студентами по семестрам.

Условие задачи предполагает нахождение среднего балла по группе за каждый семестр.

	А	В	С	Д	Е	Ф
1	семестр	группа	студент	предмет	оценка	
2	1	мг-11	Иванов И.И.	математика	4	
3	1	мг-11	Петров П.П.	математика	5	
4	1	мг-12	Иванов И.И.	информатика	5	
5	2	уп-21	Смирнов С.И.	экономика	4	
6	2	уп-11	Сидоров С.С.	история	4	
7	1	мг-11	Петров П.П.	информатика	5	
8	2	мг-11	Иванов И.И.	история	3	
9						
10						

Рис.2.10.Список для примера 2

Вариант а). Использование механизма "Промежуточные итоги".

Действия:

- отсортировать список по полям : "Семестр", "Группа", "Предмет".
- обратиться к механизму "Промежуточные итоги". Заполнить диалоговое окно:
 - "При каждом изменении в " – Семестр
 - "Операция" – среднее
 - "Добавить итоги по " – Оценка
- еще раз обратиться к механизму "Промежуточные итоги". Заполнить диалоговое окно:
 - "При каждом изменении в " – Группа
 - "Операция" – среднее
 - "Добавить итоги по" – Оценка
 - "Заменить текущие итоги" – снять флажок.
- скрыть детальные данные.

Двойное обращение к механизму "Промежуточные итоги" связано с необходимостью получения итогов по двум полям: *Семестр* и *Группа* – а снятие флажка в опции "Заменить текущие итоги" – с необходимостью сохранения результатов предыдущего подведения итогов. Результат представлен на рис.2.11.

Вариант б). Использование механизма "Сводная таблица".

Действия:

- обратиться к механизму "Сводная таблица"
- последовательно заполнить диалоговые окна. Структура таблицы:

область СТРОКА: *Семестр, Группа*

область ДАННЫЕ: *Оценка*

Функция: *среднее*.

Результат представлен на рис.2.12.

1	2	3	4	А	В	С	Д	Е	Ф
1	семестр	группа	студент	предмет	оценка				
2	1	мг-11	Иванов И.И.	математика	4,0				
3	1	мг-11	Петров П.П.	математика	5,0				
4	1	мг-11	Петров П.П.	информатика	5,0				
5		мг-11 Среднее			4,7				
6	1	мг-12	Иванов И.И.	информатика	5,0				
7		мг-12 Среднее			5,0				
8	1 Среднее				4,8				
9	2	мг-11	Иванов И.И.	история	3,0				
10		мг-11 Среднее			3,0				
11	2	уп-11	Сидоров С.С.	история	4,0				
12		уп-11 Среднее			4,0				
13	2	уп-21	Смирнов С.И.	экономика	4,0				
14		уп-21 Среднее			4,0				
15	2 Среднее				3,7				
16		Общее среднее			4,3				
17	Общее среднее				4,3				
18									

	А	В	С	Д
1	Среднее по полю оценка			
2	семестр	группа	Всего	
3		1 мг-11	4,7	
4		мг-12	5,0	
5	1	Всего	4,8	
6		2 мг-11	3,0	
7		уп-11	4,0	
8		уп-21	4,0	
9	2	Всего	3,7	
10	Общий итог			4,3
11				
12				

Рис.2.12.Результат решения примера 2 с помощью механизма "Сводная таблица"

Рис.2.11. Результат решения примера 2 с помощью механизма "Промежуточные итоги"

Вопросы для самоконтроля

1. Информационные методы обработки данных
2. Компоненты информационной технологии обработки данных в Excel
3. Источники данных в Excel
4. База данных. Список. Таблица
5. Способы организации данных в Excel
6. Обработка списков типа "один объект" – "одна запись"
7. Обработка списков типа "один объект" – "несколько записей"

Раздел 3. Инструментальные средства анализа и обработки данных

Поиск и выборка данных

Поиск и анализ данных – это наиболее частые действия, производимые со списком или базой данных. Каждая строка списка является *записью*, а каждый столбец строки – *полем*. Первая строка списка содержит имена полей списка и называется *строкой заголовков*.

Поиск производится на основе задаваемых пользователем *критериев* – требований, налагаемых на информацию. В Excel результат поиска информации может быть представлен двумя способами: в отдельной области рабочего листа и в области данных с помощью временного скрывания информации, не удовлетворяющей условию поиска. В первом случае говорят о *выборке* данных, а во втором – о *фильтрации*.

В Excel существует три средства для поиска данных: ***Форма данных***, ***Автофильтр*** и ***Расширенный фильтр***.

Форма данных позволяет с легкостью осуществлять поиск и редактирование отдельных записей. Этот механизм чаще всего и используется для редактирования данных списка.

Автофильтр обеспечивает простой и быстрый способ скрывания лишних записей, оставляя на экране только те, что удовлетворяют критериям поиска. Он позволяет выполнить только фильтрацию данных.

Расширенный фильтр предоставляет более широкие возможности, с его помощью можно выполнить и фильтрацию, и выборку данных.

При работе с *Автофильтром* и *Расширенным фильтром* используются три составляющие полнофункционального списка Excel: *Диапазон базы данных*, *Диапазон критериев* и *Диапазон для извлечения*. Причем, только *Диапазон базы данных* используется обоими механизмами, а два последние диапазона предназначены для организации работы только *Расширенного фильтра*.

Диапазон базы данных

Диапазон базы данных – это область на рабочем листе для хранения данных списка. Является обязательным для всех трех средств фильтрации и поиска данных. Для работы со списком необходимо указать эту область. Сделать это можно явным образом или автоматически. Существует несколько правил для определения местоположения списка:

- если какому-либо диапазону ячеек не текущем листе присвоено имя База_данных, Excel будет считать этот диапазон списком;

- если имя База_данных не определено, но перед выбором команды из меню *Данные* выделен диапазон ячеек, то он и будет воспринят как список;

- если вышеперечисленные правила неприменимы, но активной ячейкой является любая ячейка внутри списка. Excel определит диапазон базы данных следующим образом. Будет сделана попытка найти самую верхнюю строку, содержащую текст. В случае успеха эта строка станет строкой заголовков, а все непустые строки под заголовками будут рассмотрены как список. Если текстовая строка не найдена, в качестве списка будут выделены все соседние непустые ячейки вокруг активной.

В любом случае самая верхняя строка найденного диапазона интерпретируется как строка имен полей, нижележащие строки – как строки данных.

При создании списков следует придерживаться следующих рекомендаций:

- над списком необходимо располагать строку заголовков с уникальными именами для всех полей;

- если в рабочей книге содержится один список, присвойте ему имя *База_данных* (прямо в *Поле имен* или с помощью команды *Вставка-Имя-Присвоить*).

- располагайте каждый создаваемый список так, чтобы он был ограничен краями рабочего листа и пустыми ячейками, либо так, чтобы со всех сторон списка были пустые ячейки. В этом случае по одной выделенной ячейке Excel сможет автоматически определить весь список.

Пример *Диапазона базы данных* приведен на рис.3.1

	А	В	С	Д	Е	
1	Дата	Автор	Изд-во	Гонорар	Аванс	
2	01.01.94	Иванов	Мысль	150	80	
3	06.01.93	Петров	Финансы	220	100	
4	07.02.95	Комаров	Мысль	160	60	
5	01.01.94	Сидоров	Наука	180	90	
6	01.01.94	Сыч	Финансы	215	150	
7	06.01.93	Пырх	Мысль	90	40	
8	15.03.94	Иванов	Мысль	100	50	
9	06.01.93	Комаров	Финансы	400	150	
10						

Рис.3.1. Пример базы данных (списка)

Диапазон критериев

Условия отбора информации – критерии для всех средств фильтрации и поиска информации в Excel задаются по-разному. В *Форме данных* для этого используется диалоговое окно (см.рис.3.2), в *Автофильтре* – раскрывающийся список *Автофильтра* (см.рис.3.6), в *Расширенном фильтре* создается *Диапазон критериев* (см.рис.3.11).

Диапазон критериев применяется для задания условий отбора записей из списка и состоит как минимум из двух строк: строки с заголовками полей и строк с критериями списка.

Верхняя строка диапазона критериев должна содержать имена, в точности совпадающие с заголовками столбцов списка. Точное соответствие имен полей списка и имен, указанных в *Диапазоне критериев*, проще всего обеспечить с помощью команд *Копировать* и *Вставить*.

При создании строки заголовков диапазона критериев можно:

- перечислять все имена, но это не обязательно;
- указать только те заголовки полей, которые определяют условия отбора;
- дублировать заголовки полей при создании сложных условий поиска;
- менять взаиморасположение заголовков полей.

Строка, а в случае сложного условия поиска - строки, с задаваемыми критериями должны располагаться непосредственно за строкой с заголовками полей.

Диапазон для извлечения

Диапазон для извлечения – область ячеек, куда Excel помещает записи, отобранные согласно заданным критериям. Используется только в *Расширенном фильтре* и является необязательным. Если он отсутствует, то в *Диапазоне базы данных* скрывается информация, не удовлетворяющая условию поиска. Этот способ фильтрации имеет один существенный недостаток: поскольку исходные данные и результат поиска расположены в одной области рабочего листа, т.е. результат есть по сути часть исходной информации, то проверить правильность фильтрации практически невозможно.

Кроме того, с помощью *Диапазона для извлечения* можно быстро создавать новые списки, имеющие структуру полей записи, отличную от исходной, и при этом основной список остается нетронутым.

Диапазон для извлечения может находиться как на том же рабочем листе, где расположен исходный список, так и на другом.

Как и *Диапазон критериев*, *Диапазон для извлечения* в обязательном порядке начинается со строки с заголовками полей. Правила ее создания аналогичны созданию строки заголовков *Диапазона критериев*.

Различают ограниченный и неограниченный *Диапазон для извлечения*.

Ограниченный диапазон для извлечения - диапазон, для которого указано максимально допустимое количество строк, чтобы задать такой диапазон, следует создать строку с заголовками полей, а затем выделить ее и нужное количество пустых строк под ней.

Неограниченный диапазон для извлечения – диапазон, содержащий только строку заголовков полей. При отборе записей в этом случае Excel заполняет столько строк, сколько записей исходного списка удовлетворяет условию поиска.

Критерии поиска и фильтрации данных

Различают:

- критерии поиска на основе сравнения;
- поиск соответствия по дате и времени;
- поиск близкого соответствия с использованием образца;
- поиск соответствия с использованием множественного критерия с операциями И/ИЛИ;
- вычисляемый критерий.

Критерии поиска на основе сравнения

Критерии на основе сравнения позволяют:

- а) находить точные соответствия;
- б) отбирать данные, соответствующие простому условию отбора.

При этом математические вычисления и логические операции (такие, как *И*, *ИЛИ*) не используются. Критерии на основе сравнения можно использовать в Excel при любом способе поиска или фильтрации.

Поиск данных, точно соответствующих критерию, является простейшим. Для ввода критерия достаточно ввести в соответствующее поле искомое значение (*Форма данных*- рис.3.3, *Расширенный фильтр* – рис.3.11 или выбрать его из раскрывающегося списка *Автофильтра* – рис.3.6,3.7).

Отбор данных, соответствующих простому условию отбора, осуществляется на основе различных операций сравнения, представленных в таблице 1.

Таблица 1

Операция	Смысл	Пример	Результат поиска
=	Равняется	=150	Поля, равные 150
=	Равняется	=	Пустые поля
>	Больше	>150	Поля, больше 150
<	Меньше	<150	Поля меньше 150
>=	Больше или равно	>=150	Поля больше или равно 150
<=	Меньше или равно	<=150	Поля меньше или равно 150
<>	Не равно	<>150	Поля, не равные 150
<>	Не равно	<>	Непустые поля

Перечисленные операции сравнения можно использовать не только с числовыми аргументами. Например, чтобы отобрать издательства, начинающиеся на букву К и далее по алфавиту, следует ввести критерий >=К. На рис.3.3,3.4,3.5,3.8 и т.д. приведены примеры критериев с операциями сравнения.

Поиск соответствия по дате и времени

При поиске и отборе информации по датам можно пользоваться всеми операциями сравнения, перечисленными в табл.1. При составлении критерия дата и время вводится так же, как это делается в ячейках рабочего листа. Можно использовать любой существующий в Excel формат даты и времени. Например, >=11.04.99 или <19.авг. (См. пример рис.3.17).

Поиск близкого соответствия с использованием образца

Образец или маску поиска можно задать с помощью специальных символов, называемых символами шаблона. К текстовым символам шаблона относятся звездочка (*) и знак вопроса (?). Они обозначают:

? – любой одиночный символ в той же позиции;

* - любая последовательность символов, может указываться в любом месте: в начале, в середине или в конце текста.

Используется образец поиска в случае неуверенности в правильности написания слова или, если необходимо найти записи, содержащие похожий, но не совпадающий полностью текст. Например, если вы не уверены в написании фамилии: *Коротич* или *Каротич*, в качестве критерия можно ввести – *К?ротич*. Чтобы узнать, к примеру, какие издательства, имеющие в названии слово "Наука", перечислены в базе данных, подходит критерий: *Наука **. С его помощью будут обнаружены такие записи, как

Наука

Наука и жизнь,

а по критерию: *наука, – будут обнаружены такие записи:

Академическая наука.

Если необходимо найти поля, содержащие сам символ ? или *, следует в образце перед знаками «?» или «*» поставить знак «Тильда» (~).

Поиск соответствия с использованием множественного критерия с операциями И/ИЛИ

Для поиска записей, удовлетворяющих нескольким условиям, можно задавать множественные критерии, используя логические операции *И* и *ИЛИ*. Такие сложные критерии вычисляются по следующим правилам:

И Чтобы запись была найдена или отобрана с помощью фильтра, она должна отвечать всем критериям, соединенным операцией *И*. (Множественный критерий с операцией *И* читается как: это должно быть истиной, *И* это должно быть истиной, *И* ...)

Используется:

- чтобы задать допустимый интервал для **одного поля** (Доход >100 *И* Доход <300);
- чтобы задать условия, которым **должны отвечать различные поля одной записи** (Фамилия=Иванов *И* Доход >500).

ИЛИ Чтобы запись была найдена, она должна отвечать хотя бы одному из критериев, соединенных операций *ИЛИ*. (Множественный критерий с операцией *ИЛИ* читается как: *ИЛИ* это должно быть истиной, *ИЛИ* это должно быть истиной, *ИЛИ*...)

Используется:

- чтобы задать условия, которым **может отвечать одно поле** (Фамилия=Иванов *ИЛИ* Фамилия=Петров);
- чтобы задать условия на **различные поля одной записи** (Фамилия=Иванов *ИЛИ* Доход >=500).

Для различных средств фильтрации и поиска данных (*Форма данных, Автофильтр, Расширенный Фильтр*) существуют различные правила использования множественных критериев и записи. Они рассматриваются ниже.

Вычисляемые критерии

Вычисляемые критерии – это условия поиска и фильтрации данных, описанные с помощью формул. Могут быть использованы только в *Расширенном фильтре*.

Автофильтр

Автофильтр является мощным средством фильтрации данных. С его помощью можно отфильтровать данные по сложному условию, состоящему из условий, которым должны отвечать различные поля одной записи. В свою очередь условие поиска по отдельному полю может состоять из одного или двух условий, связанных логической операцией *И()* (для задания интервала) или *ИЛИ()* (для задания двух альтернативных операций сравнения). Например, по издательствам «Мысль» или «Наука», найти всех авторов, гонорар которых находится в диапазоне [200; 500] и договор заключен после 1998 года.

Записи, не удовлетворяющие заданному критерию фильтрации данных, скрываются. *Автофильтр* скрывает строки рабочего листа целиком. Если около боковых границ списка имеются другие данные, они могут оказаться скрытыми вместе с записями списка. Это надо помнить при организации списка.

Чтобы воспользоваться *Автофильтром*:

1. Выделите любую ячейку внутри списка или выделите его целиком.

2. Выберите команду *Данные – Фильтр – Автофильтр*.

Как видно на рис.3.6 ячейки с названиями полей дополняются элементами структуры «Раскрывающийся список».

	А	В	С	Д	Е
1	Дата ▾	Автор ▾	Изд-во ▾	Гонора ▾	Аванс ▾
2	01.01.94	Иванов	(Все)	150	80
3	06.01.93	Петров	(Первые 10...) (Условие...)	220	100
4	07.02.95	Комаров	Мысль	160	60
5	01.01.94	Сидоров	Наука	180	90
6	01.01.94	Сыч	Финансы	215	150
7	06.01.93	Пырь	Мысль	90	40
8	15.03.94	Иванов	Мысль	100	50
9	06.01.93	Комаров	Финансы	400	150
10					

Рис.3.6. Раскрывающийся список Автофильтра

Чтобы отфильтровать записи:

1. Раскройте список, соответствующий полю, которое следует включить в критерий. (На рис.3.6 это поле «Изд-во»).

2. Выберите желаемый критерий. Существуют следующие возможности:

<i>Все</i>	Вывод на экран всех записей (снятие ограничений на поле)
Первые 10	Вывод на экран заданного числа или процента первых или последних элементов списка
Условие	Вывод на экран диалогового окна <i>Пользовательский автофильтр</i> , чтобы задать одно или два условия с использованием шаблонов, операций сравнения и логических операций <i>И()</i> , <i>ИЛИ()</i>
<i>Точное значение</i>	Вывод на экран записей, поля которых в точности совпадают с выбранным значением
<i>Пустые</i>	Вывод на экран записей с пустым полем
<i>Непустые</i>	Вывод на экран записей, содержащих в поле какие-либо данные

3. Задайте параметры в диалоговом окне выбранной команды или выберите интересующее вас точное значение поля.

4. Вернитесь к шагу 1, если необходимо включить в критерий другое поле.

Результаты работы *Автофильтра* отображаются моментально. В результате фильтрации данных часть записей списка скрывается вместе с соответствующими номерами строк (см.рис.3.7). Нумерация записей отображается голубым цветом для индикации того, что на листе выведено только подмножество списка. Также меняют цвет на голубой стрелки кнопок раскрывающихся списков тех полей, которые задействованы в критерии.

Можно исключить любое поле из критерия. Для этого раскройте соответствующий список и выберите команду *Все*. Чтобы снять критерий целиком, выберите команду *Данные – Фильт – Показать все*. Чтобы отменить режим *Автофильтра*, выберите команду *Данные– Фильт–, Автофильтр*.

При работе с Автофильтром имеется три метода фильтрации данных:

- поиск точного соответствия;
- поиск по команде *Условие*;
- поиск по команде *Первые 10*.

	А	В	С	Д	Е
1	Дата ▾	Автор ▾	Изд-во ▾	Гонора ▾	Аванс ▾
2	01.01.94	Иванов	Мысль	150	80
4	07.02.95	Комаров	Мысль	160	60
7	06.01.93	Пырх	Мысль	90	40
8	15.03.94	Иванов	Мысль	100	50
10					

Рис.3.7. Поиск информации по издательству «Мысль» средствами Автофильтра

Поиск точного соответствия

Раскрывающийся список поля содержит полный перечень значений этого поля, имеющих место в обрабатываемой базе данных (см.рис.3.6). При выборе в раскрывающемся списке значения «Мысль» был получен результат рис.3.7.

Поиск по команде «Условие»

По команде Условие можно отфильтровать данные списка на базе операций сравнения или задать более сложное условие с помощью логических операций И() и/или ИЛИ().

Выбор в раскрывающемся списке Автофильтра команды Условие вызывает появление на экране диалогового окна Пользовательский автофильтр (см.рис.3.8). В левом раскрывающемся списке диалогового окна можно выбрать операцию сравнения, а в правом – ввести или выбрать значение. Затем, если необходимо, выбрать один из переключателей И, ИЛИ, и задать вторую операцию и значение. Завершение ввода условия поиска – нажатие кнопки ОК.

На рис.3.9 приведены результаты по критерию поиска рис.3.8.

Рис.3.8. Диалоговое окно Пользовательский автофильтр

	А	В	С	Д	Е
1	Дата ▾	Автор ▾	Изд-во ▾	Гонора ▾	Аванс ▾
3	06.01.93	Петров	Финансы	220	100
4	07.02.95	Комаров	Мысль	160	60
5	01.01.94	Сидоров	Наука	180	90
6	01.01.94	Сыч	Финансы	215	150
10					

Рис.3.9. Результат поиска данных по сложному условию

Поиск по команде Первые 10

Команда *Первые 10* позволяет выбирать ряд записей из начала или конца списка. Число отбираемых записей можно задать непосредственно либо указать необходимый процент от всех записей. Например, можно отобрать записи по тем издательствам, выданный авторам аванс которых попадает в 10% наибольших.

Чтобы воспользоваться этим средством:

1. Раскройте список необходимого поля
2. Выберите команду *Первые 10*. Откроется диалоговое окно *Наложение условия по списку* (рис.3.10).
3. Выберите 3 параметра: число, выбор из наибольших или из наименьших элементов, единицы отсчета (количество или процент).
4. Нажмите кнопку *ОК*.

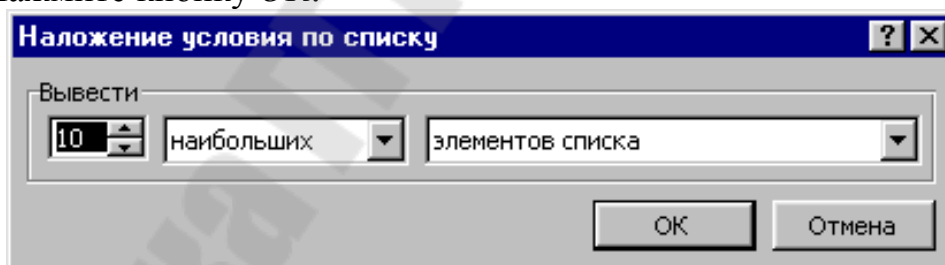


Рис.3.10. Диалоговое окно Автофильтра «Наложение условия по списку»

Расширенный фильтр

Расширенный фильтр требует большей работы, чем *Автофильтр* или *Форма данных*, но и предоставляемые возможности поиска и фильтрации значительно шире:

- можно применять сложные критерии, составленные с помощью логических операций *И()*, *ИЛИ()*, количество которых не регламентируется;

- отфильтрованные данные могут быть скопированы в другой диапазон ячеек, при этом не требуется включать все имеющиеся заголовки и сохранять их порядок.

Расширенный фильтр требует создания всех трех составляющих полнофункционального списка Excel: *Диапазона базы данных*, *Диапазона критериев* и *Диапазона для извлечения*. Они должны быть подготовлены заранее.

Расширенный фильтр может работать в двух режимах: фильтрация и выборка данных. В первом случае не удовлетворяющие критерию записи будут скрыты на месте исходного списка, а во втором – сформирован новый список из записей, удовлетворяющих условию и занимающих новый диапазон ячеек. Для фильтрации обязательны *Диапазон базы данных* и *Диапазон критериев*, а выбор требует формирования еще и *Диапазона для извлечения*.

Чтобы отфильтровать список:

1. Выделите любую ячейку списка или выделите список целиком.
2. Выберите команду *Данные – Фильтр – Расширенный фильтр*. Появится диалоговое окно (рис.3.11).
3. В группе *Обработка* установите параметр *Фильтровать список на месте*.
4. Если необходимо, перейдите к полю *Исходный диапазон* и введите ссылку или имя диапазона списка.
5. Если необходимо, перейдите к полю *Диапазон условий* и введите ссылку или имя *Диапазона критериев*.
6. Установите флажок *Только уникальные записи*, если не хотите, чтобы одинаковые записи повторялись (будет выводиться первая из всех удовлетворяющих критерию одинаковых записей).
7. Нажмите кнопку *ОК*. Результат фильтрации данных рис.3.11 представлен на рис.3.12.

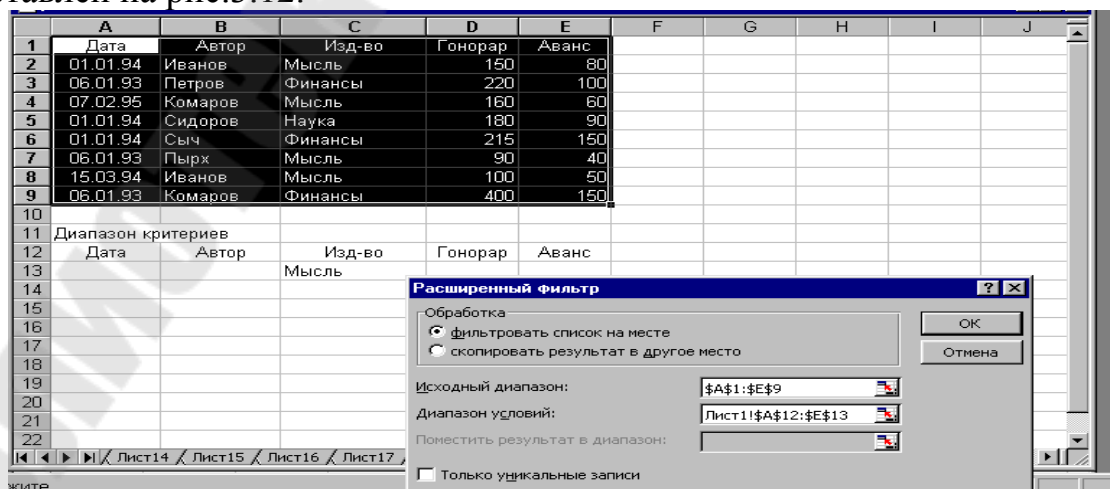


Рис 3.11 Диалоговое окно *Расширенного фильтра*

	А	В	С	Д	Е
1	Дата	Автор	Изд-во	Гонорар	Аванс
2	01.01.94	Иванов	Мысль	150	80
4	07.02.95	Комаров	Мысль	160	60
7	06.01.93	Пырх	Мысль	90	40
8	15.03.94	Иванов	Мысль	100	50
10					
11	Диапазон критериев				
12	Дата	Автор	Изд-во	Гонорар	Аванс
13			Мысль		
14					

Рис.3.12. Фильтрация данных по издательству «Мысль»

Для возвращения к данным полного списка необходимо выбрать команды *Данные – Фильтр – Отобразить все*.

Выборка данных выполняется аналогично фильтрации.

Изменения:

- в группе *Обработка* установите параметр *Скопировать результат в другое место*.
- если необходимо, в поле *Поместить результат в диапазон* введите ссылку или имя *Диапазона назначения*.

Результаты работы *Расширенного фильтра* сразу же отобразятся на рабочем листе.

Пример простого поиска данных изображен на рис.3.13, 3.14.

	А	В	С	Д	Е	Ф	Г	Н	И	Ж
1	Дата	Автор	Изд-во	Гонорар	Аванс					
2	01.01.94	Иванов	Мысль	150	80					
3	06.01.93	Петров	Финансы	220	100					
4	07.02.95	Комаров	Мысль	160	60					
5	01.01.94	Сидоров	Наука	180	90					
6	01.01.94	Сыч	Финансы	215	150					
7	06.01.93	Пырх	Мысль	90	40					
8	15.03.94	Иванов	Мысль	100	50					
9	06.01.93	Комаров	Финансы	400	150					
10										
11	Диапазон критериев									
12	Дата	Автор	Изд-во	Гонорар	Аванс					
13			Мысль							
14										
15	Диапазон для извлечения									
16	Дата	Автор	Изд-во	Гонорар	Аванс					
17										
18										
19										

Расширенный фильтр

Обработка

фильтровать список на месте

скопировать результат в другое место

Исходный диапазон:

Диапазон условий:

Поместить результат в диапазон:

Только уникальные записи

Рис.3.13.

	А	В	С	Д	Е
1	Дата	Автор	Изд-во	Гонорар	Аванс
2	01.01.94	Иванов	Мысль	150	80
3	06.01.93	Петров	Финансы	220	100
4	07.02.95	Комаров	Мысль	160	60
5	01.01.94	Сидоров	Наука	180	90
6	01.01.94	Сыч	Финансы	215	150
7	06.01.93	Пырх	Мысль	90	40
8	15.03.94	Иванов	Мысль	100	50
9	06.01.93	Комаров	Финансы	400	150
10					
11	Диапазон критериев				
12	Дата	Автор	Изд-во	Гонорар	Аванс
13			Мысль		
14					
15	Диапазон для извлечения				
16	Дата	Автор	Изд-во	Гонорар	Аванс
17	01.01.94	Иванов	Мысль	150	80
18	07.02.95	Комаров	Мысль	160	60
19	06.01.93	Пырх	Мысль	90	40
20	15.03.94	Иванов	Мысль	100	50
21					

Рис.3.14.

Расширенный фильтр отключает *Автофильтр*, поэтому его можно использовать независимо от того, включен или нет *Автофильтр*.

Использование множественного сравнения

В диапазоне критериев можно ввести несколько условий как в строке,

так и в столбце. Условия в каждой строке соединяются логической операцией *И*, а строки между собой соединяются операцией *ИЛИ*. Поясним на примерах.

Несколько условий введено в одной строке. На рис.3.15 критерий поиска будет читаться так: найди данные по издательству «Мысль» *И* гонорар автора не должен превышать 150 руб., т.е. будут отфильтрованы записи, удовлетворяющие всем условиям одновременно.

	A	B	C	D	E	F
1	Дата	Автор	Изд-во	Гонорар	Аванс	
2	01.01.94	Иванов	Мысль	150	80	
3	06.01.93	Петров	Финансы	220	100	
4	07.02.95	Комаров	Мысль	160	60	
5	01.01.94	Сидоров	Наука	180	90	
6	01.01.94	Сыч	Финансы	215	150	
7	06.01.93	Пырх	Мысль	90	40	
8	15.03.94	Иванов	Мысль	100	50	
9	06.01.93	Комаров	Финансы	400	150	
10						
11	Диапазон критериев					
12	Дата	Автор	Изд-во	Гонорар	Аванс	
13			Мысль	<=150		
14						
15	Диапазон для извлечения					
16	Дата	Автор	Изд-во	Гонорар	Аванс	
17	01.01.94	Иванов	Мысль	150	80	
18	06.01.93	Пырх	Мысль	90	40	
19	15.03.94	Иванов	Мысль	100	50	
20						

Рис.3.15. Пример множественного сравнения: два условия для двух полей

Если в одной строке задублировать заголовок одного и того же поля и для каждого из них ввести простое условие поиска, например, как на рис.3.16, можно отобрать данные, попадающие в заданный промежуток, т.к. оба условия объединяются логической функцией И.

Пример. Необходимо найти авторов, заключивших договор с издательствами во второй половине 1993г. и в первой половине 1994 г..

В Диапазоне критериев вводится дополнительный столбец "Дата" (рис.3.16). В одном указывается значение нижней, а во втором – верхней границы диапазона.

	А	В	С	Д	Е	Ф
1	Дата	Автор	Изд-во	Гонорар	Аванс	
2	01.01.94	Иванов	Мысль	150	80	
3	06.01.93	Петров	Финансы	220	100	
4	07.02.95	Комаров	Мысль	160	60	
5	01.01.94	Сидоров	Наука	180	90	
6	01.01.94	Сыч	Финансы	215	150	
7	06.01.93	Пырх	Мысль	90	40	
8	15.03.94	Иванов	Мысль	100	50	
9	06.01.93	Комаров	Финансы	400	150	
10						
11	Диапазон критериев					
12	Дата	Дата	Изд-во			
13	>31.05.93	<=31.05.94				
14						
15	Диапазон для извлечения					
16	Дата	Автор	Изд-во	Гонорар	Аванс	
17	01.01.94	Иванов	Мысль	150	80	
18	01.01.94	Сидоров	Наука	180	90	
19	01.01.94	Сыч	Финансы	215	150	
20	15.03.94	Иванов	Мысль	100	50	
21						

Рис.3.16. Пример поиска данных из заданного промежутка

По одному условию введено в разных строках. В этом случае будут отфильтрованы записи, которые удовлетворяют хотя бы одному условию. На рис.3.17 приведены критерии и результаты поиска данных об издательствах «Мысль» или «Наука».

	А	В	С	Д	Е	Ф
1	Дата	Автор	Изд-во	Гонорар	Аванс	
2	01.01.94	Иванов	Мысль	150	80	
3	06.01.93	Петров	Финансы	220	100	
4	07.02.95	Комаров	Мысль	160	60	
5	01.01.94	Сидоров	Наука	180	90	
6	01.01.94	Сыч	Финансы	215	150	
7	06.01.93	Пырх	Мысль	90	40	
8	15.03.94	Иванов	Мысль	100	50	
9	06.01.93	Комаров	Финансы	400	150	
10						
11	Диапазон критериев					
12	Дата	Автор	Изд-во	Гонорар	Аванс	
13			Финансы			
14			Наука			
15						
16	Диапазон для извлечения					
17	Дата	Автор	Изд-во	Гонорар	Аванс	
18	06.01.93	Петров	Финансы	220	100	
19	01.01.94	Сидоров	Наука	180	90	
20	01.01.94	Сыч	Финансы	215	150	
21	06.01.93	Комаров	Финансы	400	150	
22						

Рис.3.17. Пример множественного сравнения: два условия для одного поля

Для числовых значений данный вариант формирования *Диапазона критериев* связан с поиском информации вне заданного промежутка.

Несколько условий введено в нескольких строках и столбцах. Результатом поиска будут записи, которые удовлетворяют всем условиям первой строки или всем условиям второй строки или всем условиям третьей строки и т.д. На рис.3.18 видны записи об издательстве «Мысль» по авторам с гонорарами, большими 100, или о договорах издательства «Финансы», заключенные после 31.12.93г.

	A	B	C	D	E	F
1	Дата	Автор	Изд-во	Гонорар	Аванс	
2	01.01.94	Иванов	Мысль	150	80	
3	06.01.93	Петров	Финансы	220	100	
4	07.02.95	Комаров	Мысль	160	60	
5	01.01.94	Сидоров	Наука	180	90	
6	01.01.94	Сыч	Финансы	215	150	
7	06.01.93	Пырх	Мысль	90	40	
8	15.03.94	Иванов	Мысль	100	50	
9	06.01.93	Комаров	Финансы	400	150	
10						
11	Диапазон критериев					
12	Дата	Автор	Изд-во	Гонорар	Аванс	
13			Мысль	>100		
14	>31/12/93		Финансы			
15						
16	Диапазон для извлечения					
17	Дата	Автор	Изд-во	Гонорар	Аванс	
18	01.01.94	Иванов	Мысль	150	80	
19	07.02.95	Комаров	Мысль	160	60	
20	01.01.94	Сыч	Финансы	215	150	
21						

Рис.3.18.Пример поиска по сложному условию: несколько условий введено в нескольких строках и столбцах

Вычисляемые критерии

Вычисляемые критерии позволяют создавать наиболее интересные условия поиска информации. С их помощью можно

- сравнивать значения разных полей;
- сравнивать значения со значением, зафиксированным в конкретной ячейке;
- сравнивать значения с результатами вычислений встроенных функций.

В любом из этих случаев следует прибегнуть к формулам для описания условий поиска данных.

При составлении вычисляемого критерия необходимо пользоваться следующими правилами:

1. В диапазоне критериев вводится дополнительный столбец. Причем заголовок этого поля не должен совпадать с заголовком ни одного поля. Располагаться этот столбец может в любом месте *Диапазона критериев*.

2. Поскольку критерий поиска – формула, он должен начинаться со знака "равно" (=).

3. Так как формула поиска логическая, то она в обязательном порядке должна содержать операцию отношения или логическую функцию И(), ИЛИ(), НЕ().

4. Ссылки в формуле критерия могут указывать как на записи списка, так и на другие ячейки рабочего листа. Для ссылок на список следует использовать относительные ссылки, причем они должны указывать на первую запись в *Диапазоне базы данных*. Ссылки на ячейки вне списка или ссылка на весь диапазон ячеек списка во встроенных функциях должны быть абсолютными (со знаком \$).

Приведем несколько примеров сравнения полей одной записи:

= B12 = E12

= B12 < 12/2

= B12 - E12 > 10

= B12 <> \$C\$5

= C2 < CPЗНАЧ(\$C2:\$C\$200)

Результат выполнения формулы – значение *ИСТИНА* или *ЛОЖЬ*, что соответствует выполнению или невыполнению условия поиска, заданного формулой, для первой записи *Диапазона базы данных*. Все записи, удовлетворяющие заданному условию, отображаются либо в области *Диапазона базы данных*, либо в *Диапазоне для извлечения*.

Пример. *Отобрать данные по выплатам гонораров для авторов, после выплаты аванса у которых остаток гонорара превышает 100.*

Данные и *Диапазон критериев* представлены на рис.3.19,а результат поиска – на рис.3.20.

	А	В	С	Д	Е	Ф
1	Дата	Автор	Изд-во	Гонорар	Аванс	
2	01.01.94	Иванов	Мысль	150	80	
3	06.01.93	Петров	Финансы	220	100	
4	07.02.95	Комаров	Мысль	160	60	
5	01.01.94	Сидоров	Наука	180	90	
6	01.01.94	Сыч	Финансы	215	150	
7	06.01.93	Пырь	Мысль	90	40	
8	15.03.94	Иванов	Мысль	100	50	
9	06.01.93	Комаров	Финансы	400	150	
10						
11	Диапазон критериев					
12	Гонорар	Аванс	Остаток			
13			=D2-E2>100			
14	Диапазон для извлечения					
15	Дата	Автор	Изд-во	Гонорар	Аванс	
16						

Рис.3.19. Пример сравнения значений разных полей

10					
11	Диапазон критериев				
12	Гонорар	Аванс	Остаток		
13			ЛОЖЬ		
14	Диапазон для извлечения				
15	Дата	Автор	Изд-во	Гонорар	Аванс
16	06.01.93	Петров	Финансы	220	100
17	06.01.93	Комаров	Финансы	400	150
18					

Рис. 3.20. Результат поиска информации по вычисляемому критерию рис.3.19.

Примеры использования *Расширенного фильтра* приведены в дополнительном файле «*Расширенный фильтр.xls*»

Логический анализ данных

Логический анализ данных основан на применении встроенных логических функций *И()*, *ИЛИ()*, *НЕ()* и *ЕСЛИ()*, а также статистической функции *СЧЕТЕСЛИ()* и математической функции *СУММЕСЛИ()*.

Логические функции

Функция *И()*. Синтаксис:

$$И(арг.1; арг2; …; аргN),$$

где *арг.1; арг2; …; аргN* – логические выражения, принимающие значения *ИСТИНА* или *ЛОЖЬ*. Максимальное количество аргументов – 30.

Функция возвращает значение *ИСТИНА*, если все аргументы имеют значение *ИСТИНА*; возвращает значение *ЛОЖЬ*, если хотя бы один аргумент имеет значение *ЛОЖЬ*.

Функция *ИЛИ()*. Синтаксис:

$$ИЛИ(арг.1; арг2; …; аргN),$$

где *арг.1; арг2; …; аргN* – логические выражения, принимающие значения *ИСТИНА* или *ЛОЖЬ*. Максимальное количество аргументов – 30.

Функция возвращает значение *ИСТИНА*, если хотя бы один из аргументов имеет значение *ИСТИНА*; возвращает значение *ЛОЖЬ*, если все аргументы имеют значение *ЛОЖЬ*.

Функция *НЕ()*. Синтаксис:

$$НЕ(логический аргумент),$$

где *логический аргумент* – величина или выражение, принимающее одно из значений: *ИСТИНА* или *ЛОЖЬ*.

Функция меняет на противоположное логическое значение своего аргумента. Функция *НЕ()* используется в тех случаях, когда необходимо быть уверенным в том, что значение не равно некоторой конкретной величин. Например, *=НЕ(ЛОЖЬ)*, т.е. *ИСТИНА*;
=НЕ(=A3), т.е. не совпадает со значением ячейки *A3*.

Пример. На основании данных об аттестации студентов (рис.3.21) найти информацию о гомельчанах, получивших по информатике оценку, большую 3.

I4		=И(C4="Гомель";E4>3)							
	A	B	C	D	E	F	G	H	I
1	Результаты аттестации								
2									
3	Группа	Факультет	Город	Ф.И.О.	Информатика	Математика	Физика		
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	3	4	4		ЛОЖЬ
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	2	5			ЛОЖЬ
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	3	3	3		ЛОЖЬ
7	МГ-23	ГЭФ	Корма	Петров П.П.	4		4		ЛОЖЬ
8	МТ-11	ГЭФ	Гомель	Котов А.А.	3	2			ЛОЖЬ
9	МТ-21	ГЭФ	Брагин	Петрович С.И.	4	4	4		ЛОЖЬ
10	МТ-11	ГЭФ	Брагин	Сергеев С.С.	5	5	5		ЛОЖЬ
11	МТ-22	ГЭФ	Брагин	Игорев И.И.	4	4	4		ЛОЖЬ
12	МТ-12	ГЭФ	Корма	Пушкин А.С.	3	3	3		ЛОЖЬ
13	Э-11	ЭФ	Гомель	Сидоров С.С.	5	4	4		ИСТИНА
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	5	5	5		ИСТИНА
15									

Рис.3.21. Пример использования функции *И()*

Пример. Отобразить данные о студентах из Гомеля и Брагина, получивших одинаковые оценки по информатике и физике.

Решение осуществляется с помощью формулы
=И(ИЛИ(C4="Гомель";C4="Брагин");E4=G4)

Функция ЕСЛИ(). Синтаксис:

ЕСЛИ(условие;если_да;если_нет),

где *условие* – любое значение или выражение, принимающее значение *ИСТИНА* или *ЛОЖЬ*;

если_да – значение, которое возвращается функцией, если *условие* выполняется;

если_нет – значение, которое возвращается функцией, если условие не выполняется. Этот аргумент может отсутствовать. В этом случае функция возвращает значение *ЛОЖЬ*. Этот аргумент может быть представлен формулой. В одной функции можно использовать до 7 функций *ЕСЛИ()*, вложенных друг в друга, что позволяет конструировать более сложные условия.

Пример. Отметить константой «хорошо» записи списка с результатами аттестации студентов, получивших по информатике оценку 4 и выше.

Решение представлено на рис. 3.22.

		=ЕСЛИ(E7>=4;"хорошо";" ")						
	A	B	C	D	E	F	G	K
1	Результаты аттестации							
2								
3	Группа	Факультет	Город	Ф.И.О.	Информатика	Математика	Физика	
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	3	4	4	
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	2	5		
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	3	3	3	
7	МГ-23	ГЭФ	Корма	Петров П.П.	4		4	хорошо
8	МТ-11	ГЭФ	Гомель	Котов А.А.	3	2		
9	МТ-21	ГЭФ	Брагин	Петрович С.И.	4	4	4	хорошо
10	МТ-11	ГЭФ	Брагин	Сергеев С.С.	5	5	5	хорошо
11	МТ-22	ГЭФ	Брагин	Игорев И.И.	4	4	4	хорошо
12	МТ-12	ГЭФ	Корма	Пушкин А.С.	3	3	3	
13	Э-11	ЭФ	Гомель	Сидоров С.С.	5	4	4	хорошо
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	5	5	5	хорошо
15								

Рис.3.22. Пример использования функции *ЕСЛИ()*

Пример. Отметить константой «хорошо» записи списка с результатами аттестации студентов, получивших по всем предметам оценку 4 и выше.

Решение представлено на рис. 3.23.

L4		=ЕСЛИ(И(Е4>=4;F4>=4;G4>=4);"хорошо";" ")						
	A	B	C	D	E	F	G	L
1	Результаты аттестации							
2								
3	Группа	Факультет	Город	Ф.И.О.	Информатика	Математика	Физика	
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	3	4	4	
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	2	5		
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	3	3	3	
7	МГ-23	ГЭФ	Корма	Петров П.П.	4		4	
8	МТ-11	ГЭФ	Гомель	Котов А.А.	3	2		
9	МТ-21	ГЭФ	Брагин	Петрович С.И.	4	4	4	хорошо
10	МТ-11	ГЭФ	Брагин	Сергеев С.С.	5	5	5	хорошо
11	МТ-22	ГЭФ	Брагин	Игорев И.И.	4	4	4	хорошо
12	МТ-12	ГЭФ	Корма	Пушкин А.С.	3	3	3	
13	Э-11	ЭФ	Гомель	Сидоров С.С.	5	4	4	хорошо
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	5	5	5	хорошо

Рис.3.23. Пример использования функции ЕСЛИ()

Пример. Отметить константой «неявка» студентов, не явившихся на экзамен.

Решение представлено на рис. 3.24.

M4		=ЕСЛИ(ИЛИ(Е4<1;F4<1;G4<1);"Неявка";" ")						
	A	B	C	D	E	F	G	M
1	Результаты аттестации							
2								
3	Группа	Факультет	Город	Ф.И.О.	Информатика	Математика	Физика	
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	3	4	4	
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	2	5		Неявка
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	3	3	3	
7	МГ-23	ГЭФ	Корма	Петров П.П.	4		4	Неявка
8	МТ-11	ГЭФ	Гомель	Котов А.А.	3	2		Неявка
9	МТ-21	ГЭФ	Брагин	Петрович С.И.	4	4	4	
10	МТ-11	ГЭФ	Брагин	Сергеев С.С.	5	5	5	
11	МТ-22	ГЭФ	Брагин	Игорев И.И.	4	4	4	
12	МТ-12	ГЭФ	Корма	Пушкин А.С.	3	3	3	
13	Э-11	ЭФ	Гомель	Сидоров С.С.	5	4	4	
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	5	5	5	

Рис.3.24. Пример использования ЕСЛИ()

Пример. Отметить константой «выше среднего» студентов, сдавших экзамен по математике лучше среднего уровня.

Решение представлено на рис. 3.25.

Пример. Отметить константами «отличник», «хорошист», «троечник» и «двоечник» студентов, сдавших экзамен по математике на соответствующую оценку.

Решение осуществляется с помощью формулы

=ЕСЛИ(F4=5;"отличник";ЕСЛИ(F4=4;"хорошист";ЕСЛИ(F4=3;"троечник";"двоечник")))

Результат представлен на рис.3.26.

N4								
=ЕСЛИ(F4>CPЗНАЧ(\$F\$4:\$F\$14);"выше среднего";" ")								
	A	B	C	D	E	F	G	N
1	Результаты аттестации							
2								
3	Группа	Факультет	Город	Ф.И.О.	Информатика	Математика	Физика	
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	3	4	4	выше среднего
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	2	5		выше среднего
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	3	3	3	
7	МГ-23	ГЭФ	Корма	Петров П.П.	4		4	
8	МТ-11	ГЭФ	Гомель	Котов А.А.	3	2		
9	МТ-21	ГЭФ	Брагин	Петрович С.И.	4	4	4	выше среднего
10	МТ-11	ГЭФ	Брагин	Сергеев С.С.	5	5	5	выше среднего
11	МТ-22	ГЭФ	Брагин	Игорев И.И.	4	4	4	выше среднего
12	МТ-12	ГЭФ	Корма	Пушкин А.С.	3	3	3	
13	Э-11	ЭФ	Гомель	Сидоров С.С.	5	4	4	выше среднего
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	5	5	5	выше среднего

Рис.3.25. Пример использования функции ЕСЛИ()

O4								
=ЕСЛИ(F4=5;"отличник";ЕСЛИ(F4=4;"хорошист";ЕСЛИ(F4=3;"троечник";"двоечник";" ")								
	A	B	C	D	E	F	G	O
1	Результаты аттестации							
2								
3	Группа	Факультет	Город	Ф.И.О.	Информатика	Математика	Физика	
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	3	4	4	хорошист
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	2	5		отличник
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	3	3	3	троечник
7	МГ-23	ГЭФ	Корма	Петров П.П.	4		4	двоечник
8	МТ-11	ГЭФ	Гомель	Котов А.А.	3	2		двоечник
9	МТ-21	ГЭФ	Брагин	Петрович С.И.	4	4	4	хорошист
10	МТ-11	ГЭФ	Брагин	Сергеев С.С.	5	5	5	отличник
11	МТ-22	ГЭФ	Брагин	Игорев И.И.	4	4	4	хорошист
12	МТ-12	ГЭФ	Корма	Пушкин А.С.	3	3	3	троечник
13	Э-11	ЭФ	Гомель	Сидоров С.С.	5	4	4	хорошист
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	5	5	5	отличник

Рис. 3.26. Пример использования функции ЕСЛИ()

Статистическая функция СЧЕТЕСЛИ()

Синтаксис:

СЧЕТЕСЛИ(диапазон; условие),

где *диапазон* – диапазон, в котором подсчитываются ячейки;
условие – простое логическое выражение, которое определяет, какие
 ячейки надо подсчитывать.

Функция подсчитывает количество ячеек из *диапазона*, удовлетво-
 ряющих заданному условию.

Пример. *Определить количество студентов, сдавших экзамен по
 физике на 4 и 5.*

Решение представлено на рис.3.27.

G16		fx =СЧЁТЕСЛИ(G4:G14;">=4")						
	A	B	C	D	E	F	G	
1	Результаты аттестации							
2								
3	Группа	Факультет	Город	Ф.И.О.	Инфор- матика	Матема- тика	Физика	
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	3	4	4	
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	2	5		
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	3	3	3	
7	МГ-23	ГЭФ	Корма	Петров П.П.	4		4	
8	МТ-11	ГЭФ	Гомель	Котов А.А.	3	2		
9	МТ-21	ГЭФ	Брагин	Петрович С.И.	4	4	4	
10	МТ-11	ГЭФ	Брагин	Сергеев С.С.	5	5	5	
11	МТ-22	ГЭФ	Брагин	Игорев И.И.	4	4	4	
12	МТ-12	ГЭФ	Корма	Пушкин А.С.	3	3	3	
13	Э-11	ЭФ	Гомель	Сидоров С.С.	5	4	4	
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	5	5	5	
15								
16	Количество студентов, сдавших экзамен по физике на 4 и 5						7	
17								

Рис.3.27. Пример использования функции СЧЕТЕСЛИ()

Поскольку в качестве условия в функции СЧЕТЕСЛИ() можно ис-
 пользовать только простое логическое выражение, то в тех случаях, когда
 условие сложное, решение задач осуществляется в два этапа:

1. С помощью функции ЕСЛИ() формируется дополнительный стол-
 бец, в котором отмечаются записи списка, удовлетворяющие сложному ло-
 гическому выражению.

2. В качестве аргумента «диапазон» функции СЧЕТЕСЛИ() исполь-
 зуется сформированный на предыдущем этапе дополнительный столбец, а
 аргумент «условие» – соответствие аргументу «если_да» функции ЕСЛИ().

Пример. *Подсчитать количество студентов группы МТ-11, сдав-
 ших экзамен по физике на 4 и 5.*

Решение осуществляется в два этапа. Сначала формируется дополни-
 тельный столбец с помощью функции ЕСЛИ() (рис.3.28). Результат рас-
 считывается с помощью формулы

=СЧЁТЕСЛИ(J4:J14;"="+")

J4		fx =ЕСЛИ(И(A4="MT-11";G4>=4);"+";"-")							
	A	B	C	D	E	F	G	J	
1	Результаты аттестации								
2									
3	Группа	Факультет	Город	Ф.И.О.	Информатика	Математика	Физика		
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	3	4	4	-	
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	2	5		-	
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	3	3	3	-	
7	МГ-23	ГЭФ	Корма	Петров П.П.	4		4	-	
8	MT-11	ГЭФ	Гомель	Котов А.А.	3	2		-	
9	MT-21	ГЭФ	Брагин	Петрович С.И.	4	4	4	-	
10	MT-11	ГЭФ	Брагин	Сергеев С.С.	5	5	5	+	
11	MT-22	ГЭФ	Брагин	Игорев И.И.	4	4	4	-	
12	MT-12	ГЭФ	Корма	Пушкин А.С.	3	3	3	-	
13	Э-11	ЭФ	Гомель	Сидоров С.С.	5	4	4	-	
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	5	5	5	-	
15									
16	К-во студентов, сдавших экзамен по физике на 4 и 5 из гр. MT11								1
17									

Рис.3.28. Пример использования функции *ЕСЛИ()* для формирования дополнительного столбца

Математическая функция СУММЕСЛИ()

Функция *СУММЕСЛИ()* может быть реализована с помощью краткой и полной формы синтаксиса.

Краткая форма синтаксиса:

где *диапазон* – диапазон, в котором подсчитываются ячейки;
условие – простое логическое выражение, которое определяет, какие ячейки надо подсчитывать.

Функция суммирует значения ячеек из *диапазона*, удовлетворяющих заданному условию.

Пример. Определить суммарное число пропусков тех студентов, имеющих более 10 часов пропусков.

Решение задачи представлено на рис. 3.29.

K17	fx =СУММЕСЛИ(K4:K14;">10")				
	A	B	C	D	K
1	Результаты аттестации				
2					
3	Группа	Факультет	Город	Ф.И.О.	Пропущено, час
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	12
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	6
7	МГ-23	ГЭФ	Корма	Петров П.П.	10
8	МТ-11	ГЭФ	Гомель	Котов А.А.	8
9	МТ-21	ГЭФ	Брагин	Петрович С.И.	14
10	МТ-11	ГЭФ	Брагин	Сергеев С.С.	
11	МТ-22	ГЭФ	Брагин	Игорев И.И.	4
12	МТ-12	ГЭФ	Корма	Пушкин А.С.	8
13	Э-11	ЭФ	Гомель	Сидоров С.С.	6
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	
15					
16	Суммарное число пропусков студентов,				
17	пропустивших более 10 часов занятий				26
18					

Рис.3.29. Пример использования функции СУММЕСЛИ()

Полная форма синтаксиса:

СУММЕСЛИ(диапазон; условие; диапазон суммирования),

где *диапазон* – диапазон, в котором ячейки проверяются на соответствие *условию*;

условие – простое логическое выражение, которое определяет, какие ячейки надо подсчитывать;

диапазон суммирования – диапазон, в котором значения ячеек суммируются, если соответствующим им ячейки в аргументе «диапазон» удовлетворяют *условию*.

Функция суммирует значения ячеек из *диапазона суммирования*, если соответствующие им ячейки из *диапазона* удовлетворяют заданному *условию*.

Пример. *Определить среднюю оценку студентов ГЭФ по информатике.*

Решение задачи представляет собой результат деления суммы оценок студентов ГЭФ, полученных по информатике, на количество студентов факультета ГЭФ, т.е. в формуле расчета используется две функции: *СУММЕСЛИ()* и *СЧЕТЕСЛИ()*. Решение представлено на рис.3.30.

К16		=СУММЕСЛИ(В4:В14;"ГЭФ";Е4:Е14)/СЧЕТЕСЛИ(В4:В14;"ГЭФ")							
	A	B	C	D	E	F	G	K	
1	Результаты аттестации								
2									
3	Группа	Факультет	Город	Ф.И.О.	Информатика	Математика	Физика		
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	3	4	4		
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	2	5			
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	3	3	3		
7	МГ-23	ГЭФ	Корма	Петров П.П.	4		4		
8	МТ-11	ГЭФ	Гомель	Котов А.А.	3	2			
9	МТ-21	ГЭФ	Брагин	Петрович С.И.	4	4	4		
10	МТ-11	ГЭФ	Брагин	Сергеев С.С.	5	5	5		
11	МТ-22	ГЭФ	Брагин	Игорев И.И.	4	4	4		
12	МТ-12	ГЭФ	Корма	Пушкин А.С.	3	3	3		
13	Э-11	ЭФ	Гомель	Сидоров С.С.	5	4	4		
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	5	5	5		
15									
16			Ср.оценка студентов ГЭФ по информатике						3,4
17									

Рис.3.30. Пример использования функции СУММЕСЛИ() и СЧЕТЕСЛИ()

Так же, как и у функции СЧЕТЕСЛИ(), аргумент «условие» функции СУММЕСЛИ() может быть только простым логическим выражением. В силу этого решение задач с использованием сложного логического выражения осуществляется в два этапа с применением функции ЕСЛИ().

Пример. Определить среднюю оценку студентов ГЭФ по математике, получивших по информатике оценку ниже 4.

Поскольку условие отбора данных сложное, решение осуществляется в два этапа. Сначала формируется дополнительный столбец (рис.3.31).

M4		=ЕСЛИ(И(Е4<4;В4="ГЭФ");"+";"-")							
	A	B	C	D	E	F	G	M	
1	Результаты аттестации								
2									
3	Группа	Факультет	Город	Ф.И.О.	Информатика	Математика	Физика		
4	МГ-11	ГЭФ	Гомель	Иванов И.И.	3	4	4	+	
5	МГ-21	ГЭФ	Брагин	Собакин И.И.	2	5		+	
6	МГ-23	ГЭФ	Гомель	Иванков М.И.	3	3	3	+	
7	МГ-23	ГЭФ	Корма	Петров П.П.	4		4	-	
8	МТ-11	ГЭФ	Гомель	Котов А.А.	3	2		+	
9	МТ-21	ГЭФ	Брагин	Петрович С.И.	4	4	4	-	
10	МТ-11	ГЭФ	Брагин	Сергеев С.С.	5	5	5	-	
11	МТ-22	ГЭФ	Брагин	Игорев И.И.	4	4	4	-	
12	МТ-12	ГЭФ	Корма	Пушкин А.С.	3	3	3	+	
13	Э-11	ЭФ	Гомель	Сидоров С.С.	5	4	4	-	
14	Э-11	ЭФ	Гомель	Федоров Ф.Ф.	5	5	5	-	
15									
16	Ср.оценка студентов ГЭФ по математике, получивших по информатике								
17	оценку, ниже 4								3,4

Рис.3.31. Пример использования функции ЕСЛИ()

Раздел 4. Разработка АРМ менеджера

Интерфейс «человек-компьютер»

Пользовательский интерфейс

Любую информационную систему можно оценить с помощью двух критериев: *точность* и *удобство*.

Традиционно критерий точности (при поступлении на вход системы заданных значений на ее выходе получают заданные результаты) всегда преобладал. Гораздо меньше внимания уделялось концепции удобства работы с вычислительной системой. Более того, в 60-х и начале 70-х гг. прошлого столетия удобство работы пользователя игнорировалось большинством разработчиков систем.

Расширение числа пользователей ЭВМ благодаря развитию персональных компьютеров, переход к пакетной обработке данных, внедрение сетевых технологий и развитие технологий хранения данных поставило критерий удобства работы как с данными, так и с программными продуктами в один ряд с точностью получаемых результатов.

Современный пользователь информационных систем имеет право ожидать не только точных результатов обработки, но и удобства в использовании системы. Прежде всего это значит, что при использовании программного продукта человек не должен существенно менять стиль своей работы. Поскольку процесс обработки данных пользователь непосредственно не чувствует, то он и не может существенно повлиять на стиль и качество работы пользователя. Наоборот интерфейс «человек-компьютер», как среда, через которую пользователь взаимодействует с системой, становится мерилем удобства и качества ее работы.

Чтобы работа с компьютером была удобной, пользователь должен при взаимодействии с системой ощущать комфорт. Таким образом, на удобство работы влияют факторы, которые вызывают чувство комфорта.

Основными *эргономическими характеристиками* реальной информационной системы, существенно улучшающими или ухудшающими отношение к ней, являются

- конструктивные особенности оборудования;
- качество разработки диалоги;
- доступность и надежность системы;
- чувствительность системы.

Конструктивные особенности оборудования (как компьютера, так и дополнительных устройств) и размещение его на рабочем месте могут повлиять на чувство *физического комфорта* пользователя при работе с системой. *Психологический дискомфорт* пользователь ощущает, если

1. нет доступа к системе в любое время и в удобном для пользователя месте;
2. система ненадежно, т.е. имеют место регулярные сбои в работе;
3. имеет место длительное ожидание ответа на запрос (в течение 20с и более);
4. пользователь может прочесть текст, но не может понять его смысл из-за непонятных слов или непривычной формы представления текста, т.е. качество разработки диалога не соответствует *эмоциональному и умственному* комфорту.

На этот аспект работы системы, получившей название *диалога*, разработчики программного обеспечения могут повлиять как в положительную, так и в отрицательную сторону, т.к. эргономические (как физические, так и психологические) характеристики интерфейса оказывает существенное влияние на производительность пользователя.

Разработка автоматизированного рабочего места специалиста прежде всего направлена на повышение физического комфорта пользователя путем согласования физических возможностей системы и человека. Разработка интерфейса связана с повышением психологического комфорта пользователя путем согласования стиля диалога с потребностями и представлениями человека.

Структурные элементы интерфейса

Интерфейс человек-компьютер включает два основных компонента:

1. процесс диалога, который связывает все фоновые процессы в единую систему;
2. набор процессов ввода-вывода, которые обеспечивают физическую связь между пользователем и процессом диалога (а данной работе не рассматривается).

Диалог между человеком и компьютером можно определить как обмен информацией между вычислительной системой и пользователем, проводимый с помощью интерактивного терминала и по определенным правилам.

Процесс диалога – это механизм обмена информацией, который можно рассматривать как некую оболочку, включающую все процессы системы по выполнению конкретных заданий.

Задачи диалогового процесса:

1. определение задания, которое пользователь возлагает на систему;
2. прием логически связанных входных данных от пользователя и размещение их внутри системы;
3. вызов процесса выполнения требуемого задания;

4. вывод результатов обработки по окончании процесса в подходящем для пользователя формате.

Правила ведения диалога с компьютером должны быть такими, чтобы облегчалась работа пользователя, а не упрощался процесс вычислений.

Во время диалога происходит обмен информацией между его участниками. Информация передается в виде сообщений. Существует несколько типов сообщений:

➤ *подсказка* – выходное сообщение системы, побуждающее пользователя вводить данные. Реакция пользователя на подсказку может вызвать процесс выполнения некоторого задания (например, задать имя файла), ввести исходные данные и быть комбинированной, т.е. за один сеанс обмена вызывается нужный процесс и вводятся требуемые данные. Соответственно различают *входные управляющие сообщения*, сообщения *ввода данных* и *сложные сообщения*;

➤ *сообщение об ошибке* – это сигнал диалогового процесса о том, что невозможно дальнейшее выполнение работы, т.к. он или вызванный процесс выполнения задания не может обработать сообщение, введенное пользователем;

➤ *выходные данные* – это данные, которые возвращает процесс по окончании обработки, причем в формате, удобном пользователем;

➤ *сообщение о состоянии системы* – это информация для пользователя о том, что произошло или происходит в системе;

➤ при затруднении пользователя диалоговый процесс может вывести *справочную информацию*, поясняющую, что делать дальше и почему.

Классификация сообщений в соответствии с выполняемыми ими функциями представлена на рис.5.7.

Диалоговые процессы можно классифицировать

1. по способу выделения диалога;
2. по формату поддерживаемых входных сообщений.

Классификацию можно уточнять с учетом вида подсказок, инициализирующих запрос на вход.

По способу ведения диалога различают

1. *диалог, управляемый системой*, – это диалог, в котором процесс жестко задает, какое задание можно выбрать и какие данные вводить. Это осуществляется с помощью системы подсказок пользователю, которые регламентируют ввод требуемой информации на каждом этапе.

2. *диалог, управляемый пользователем*, – это диалог, в котором инициатива принадлежит пользователю, т.е. он непосредственно задает команду на выполнение нужного на данном этапе задания. Для этого пользователь, как правило, вводит комбинированные данные, с помощью кото-

рых он одновременно выбирает процесс и вводит необходимые для обработки данные.

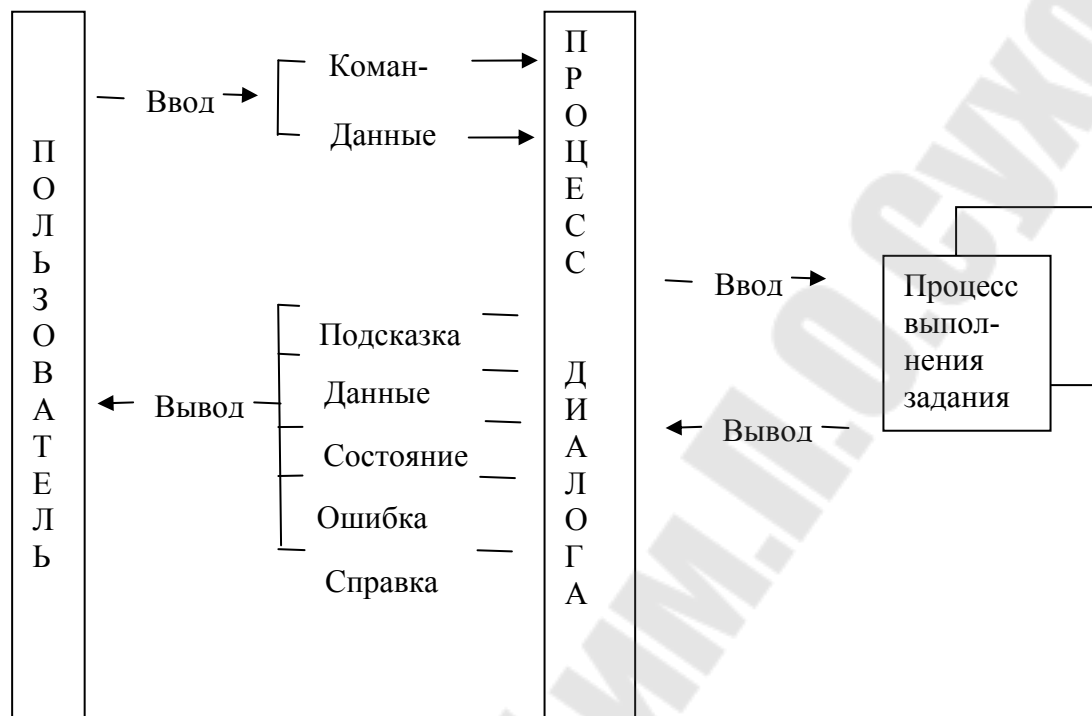


Рис.5.7.

Диалог, управляемый системой, более удобен, т.к. он лучше подстраивается под пользователя, но при этом имеет больше ограничений, чем диалог, управляемый пользователем. Также предполагается, что диалог ведется в соответствии со строго структурированными, как правило иерархическими, заданиями.

Формат, в соответствии с которым пользователь вводит свои сообщения, называют *грамматикой диалога*.

Различают следующие форматы поддерживаемых входных сообщений:

- коды (например, ввод не фамилии, имени и отчества, кода работника и т.д.);
- цепочки ключевых слов, напоминающих программу (формат командной строки);
- ограниченный естественный язык;
- естественный язык.

Критерии качества диалога

Качество любого диалога можно оценивать, руководствуясь пять основными критериями:

1. *Естественность.* Естественный диалог – это диалог, который не вынуждает пользователя, взаимодействующего с системой, существенно изменять свои традиционные способы решения задачи.

2. *Последовательность.* Диалог, отличающийся логической последовательностью, гарантирует, что пользователь, освоивший работу одной части системы, не запутается, разбираясь в особенностях описания и работы другой части системы.

3. *Краткость.* Краткий диалог требует ввода только минимума информации, необходимой для работы системы. Такой диалог обладает тем достоинством, что взаимодействие с ним становится более быстрым и осуществляется с меньшим числом ошибок, т.к. оно напрямую зависит от количества требуемых нажатий на клавиши.

4. *Поддержка пользователя.* Это мера помощи, которую диалог оказывает пользователю при его работе с системой. Выделяют три основных аспекта такой поддержки:

- количество и качество имеющихся инструкций;
- характер выдаваемых сообщений об ошибках;
- подтверждение каких-либо действий системы.

5. *Гибкость.* Гибкость – это мера того, насколько хорошо диалог соответствует различным уровням подготовки и производительности труда пользователя.

Типы диалогов и их реализация

Различают четыре основные структуры диалога, каждый из которых соответствует определенному типу взаимодействия между людьми:

1. Вопрос-ответ;
2. Меню;
3. Экранная форма;
4. Структура на базе командного языка.

Структура диалога типа «Вопрос-ответ»

В каждой точке диалога система выводит в качестве подсказки один вопрос, на который пользователь дает один ответ. В зависимости от полученного ответа система может решить, какой следующий вопрос задавать. Если дан некорректный ответ, система обычно повторяет вопрос, сообщая, что ответ не подходит.

Данный тип диалога в MS Excel реализуется с помощью стандартных встроенных окон ввода информации и вывода сообщений, с помощью графического объекта *Кнопка*, а также гиперссылок.

Структура устарела с появлением современных устройств отображения информации. Однако имеет ряд достоинств:

- это промежуточная структура, которая может удовлетворить требования различных пользователей и типов данных;
- структура гибка и в достаточной степени обеспечивает поддержку пользователя.

Область использования – при реализации диалога со множеством «ответвлений», т.е. в тех случаях, когда на каждый вопрос предусматривается большое число ответов, каждый из которых влияет на то, какой вопрос будет задан следующим. Поэтому структура часто используется в качестве диалоговой структуры в экспертных системах.

Структура диалога типа Меню

Сущность структуры меню – у пользователя есть список возможных вариантов данных для ввода, из которого нужно выбрать то, что требуется.

Меню можно использовать как для ввода управляющих сообщений, так и данных. Приемлемая структура меню зависит от размера и организации меню (в виде списка, строки меню, меню в виде блока данных, в виде пиктограмм или строки данных и т.д.) и реальной потребности пользователя в поддержке со стороны меню.

При большом числе возможных вариантов выбора их следует сгруппировать в иерархию небольших меню.

Реализация структуры типа меню осуществляется в два этапа:

1. меню отображается на экране;
2. запрос на ввод повторяется до тех пор, пока не будет сделан правильный выбор.

Для реализации диалога этого типа может использоваться пользовательская панель управления (один пункт – одно действие); элементы управления *Список*, *Счетчик*; графические объекты (выбор с указанием на объект) и др.

Меню – структура, удобная для неподготовленного пользователя, однако может утомить подготовленного пользователя. Поэтому должна содержать так называемые элементы опережающего ввода.

Структура диалога на основе экранных форм

Основная идея такого диалога – получение всей информации от пользователя путем заполнения формы. Важно, чтобы форма, отображаемая на экране, была похожа на ту форму на бумаге, которая является источником информации. Все вводимые элементы данных должны располагаться в том же относительном порядке и иметь такой же формат, что и в исходном документе.

Структура диалога типа экранной формы строится в два этапа:

1. форма отображается целиком;
2. вопросы повторяются до тех пор, пока не завершится заполнение формы. Заполнение формы заканчивается вводом знака завершения, либо ответами на все вопросы.

Многие интерактивные системы используют для ввода данных стандартные группы вопросов. В этих случаях удобна экранная форма, т.к. она работает быстрее по сравнению со структурой «Ответ - вопрос», может манипулировать более широким диапазоном входных данных, нежели меню, и ее могут использовать пользователи любой квалификации.

Структура диалога на основе командного языка

Структура диалога на основе командного языка в информационных системах для обычных пользователей используется крайне редко, т.к. требует от него дополнительных знаний.

Итак, четыре структуры диалога различаются весьма незначительно и в действительности являются просто разновидностью структуры типа «Вопрос - ответ».

Структура диалога типа меню – это такая модификация структуры диалога типа «Вопрос - ответ», когда справочная информация первого уровня, т.е. само меню, автоматически отображается до запроса возможных вариантов ответа. Структура типа экранных форм отображает сразу весь комплекс вопросов (собственно форму), а затем по очереди запрашивают ответы. Структура на основе языка команд по сути структура типа «Вопрос -ответ», в рамках которого пользователь широко применяет метод «опережающего ввода».

Второй вывод: для различных частей диалога нужны различные структуры диалога. Поэтому в большинстве прикладных программ организация диалога должна поддерживать различные его типы.

Третий вывод: не существует единственной структуры диалога, которая одинаково хорошо подходила бы для всех вариантов диалога.

Поэтому основные структуры комбинируют друг с другом. Такие комбинации называются *смешанной* структурой диалога.

Форматирование экрана

Этапы размещения данных и общие принципы расположения информации на экране

Для эффективного взаимодействия между пользователем и системой нужен диалог, который

- естественен;
- последователен;
- неизбыточен;
- обеспечивает поддержку пользователя;
- гибок.

Эти критерии применимы не только к основным структурам диалога и содержимому отображаемых сообщений, но и к тому, как они выглядят на экране. Внешний вид экрана зависит от того, какие поля сообщений отображаются, в каком месте и с какими атрибутами.

Процесс размещения данных на экране включает следующие этапы:

- решить, какая информация, т.е. какие поля должны появляться на экране;
- определить основной формат этой информации;
- решить, где она должна появляться на экране, т.е. определить область вывода для каждого поля;
- решить, какие требуются средства для выделения полей, т.е. какие атрибуты необходимы для каждого поля;
- разработать проект размещения данных на экране;
- оценить эффективность этого размещения.

Процесс повторяется до тех пор, пока разработчик и потенциальный пользователь не будут удовлетворены.

Для сокращения временных и материальных расходов при разработке диалога следует придерживаться следующих правил:

1. Данные должны располагаться так, чтобы пользователь мог просматривать экран в логической последовательности.
2. Пользователь легко мог бы
 - выводить нужную информацию;
 - идентифицировать связанные группы информации;
 - различать исключительные ситуации (сообщения об ошибках или предупреждения);
 - определять, какое действие с его стороны требуется (и требуется ли вообще) для продолжения выполнения задания.

Выбор выводимой на экран информации

1. Экран не должен содержать избыточной информации. Важно, чтобы на экране отображалась вся информация, но относящаяся к решаемой на данном, конкретном этапе задачи.

2. Не следует заставлять пользователя запоминать информацию на одном экране, чтобы позднее воспользоваться ею для обработки информации на следующем экране. Если вся информация исходного документа не помещается на одном экране, некоторые элементы данных могут

повторяться на других экранах для сохранения последовательности обработки. Однако такая информация должна располагаться в одном и том же месте на всех используемых экранах.

3. Группа логически связанных данных, т.е. группа данных, которые необходимо видеть сразу, чтобы решить поставленную задачу, не должны разделяться по разным экранам. (ряды данных могут быть логически связанными, если они описывают один аспект задачи либо вводятся из одного документа и т.д.).

Формат вывода данных

На этом этапе разработчик должен указать размер области вывода и атрибуты, связанные с каждым полем.

1. Естественность диалога предполагает, что информация представлена таким образом, что ее сразу же можно использовать, т.е. не требуется дополнительных справочников, использования калькулятора для получения суммарных итогов по строкам и столбцам и т.д.

2. Формат для ввода даты и времени должен быть стандартным, а не выводиться во внутреннем представлении системы.

3. Восприятие информации повышается с использованием общепринятой системой сочетания прописных и строчных букв (Гомель – общепринятое написание в отличие ГОМЕЛЬ).

4. Бухгалтеры привыкли отрицательные числа выделять красным цветом или заключать их в скобки, т.е. знак «минус» не используется.

5. Использование наименований полей необходимо для правильной интерпретации входных и выходных данных. Они должны точно определять содержимое соответствующего поля и отделяться от данных.

Общие принципы расположения информации

1. Плотность расположения информации зависит от конкретного пользователя и задачи. Однако существуют правила, регулирующие плотность расположения данных на экране с помощью интервалов:

- Приблизительно половина экрана должна быть пустой;
- После каждой пятой строки таблицы должна быть пустая строка или каждая пятая строка выделена цветом;
- Между данными столбцов таблицы должно быть 4-5 пробелов.

Это общие правила, которые определяют лишь общие принципы.

2. Если данные выводятся на несколько экранов, то делить их нужно так, чтобы сохранялась четкая граница раздела (если таковая имеется).

3. Нельзя нарушать логические связи в группе данных.
4. На одном экране должны находиться все необходимые данные для принятия конкретного решения.
5. На любом экране логически связанные данные должны представляться отдельной группой. Этого можно достичь, оставляя по несколько пробелов с каждой стороны группы, либо проводя горизонтальные или вертикальные линии, либо используя различные атрибуты для полей разных групп.
6. Фрагменты текста должны располагаться на экране так, чтобы взгляд пользователя сам перемещался по экрану в нужном направлении. Содержимое полей не должно «прижиматься» к краю экрана, а располагаться около горизонтальных и вертикальных осей. Меню, содержащее относительно небольшой объем информации, должно смещаться в левую верхнюю часть экрана. Чтобы подчеркнуть симметрию, содержимое и наименование полей, относящихся к одной группе, должны выравниваться по вертикале. По возможности необходимо выравнивать все логически связанные группы данных.
7. Необходимо определить *точку начала отсчета*. Естественнее всего ее связать с левым верхним углом экрана и перемещаться слева направо и сверху вниз.
8. Один и тот же тип информации должен проявляться всегда в одном и том же месте экрана.
9. Важно учитывать и эстетические характеристики.
10. Если расположенные данные необходимо согласовывать в пределах одной программы или между несколькими программами, пользуются единым шаблоном. Например:

<i>Заголовок и данные о состоянии системы</i>
<i>Верхняя область для вывода сообщений</i>
<i>Основная область</i>
<i>Нижняя область для вывода сообщений</i>
<i>Описание функциональных клавиш и «запасной люк»</i>

Заголовок и состояние системы – выводятся в верхние 2-3 строки экрана. *Заголовок* – название активной подсистемы. *Состояние системы* – подтверждение о том, что система работоспособна.

Области в верхних и нижних частях экрана, предназначенные для вывода подсказок или индикации исключительных состояний. Подсказки, определяющие способы работы с информацией на экране, располагаются в верхней части, а подсказки, поясняющие, что делать дальше, – в нижней.

Любую из этих областей можно использовать для вывода справочной информации и сообщений об ошибках. Принято выводить справочные сообщения в верхнюю часть экрана, а сообщения об ошибках – в нижнюю. Сообщения, требующие подтверждения пользователя, лучше располагать в нижней части экрана.

Основная область экрана содержит главную информацию. Для меню – это список вариантов. Заголовок меню может располагаться либо в верхней части экрана либо основной. При заполнении формы это – часть экрана для вывода наименований и значений полей.

В нижней части экрана («запасной люк») иногда фиксируется место для описания средств прерывания процесса обработки.

Выделение информации на экране

Выделение информации – это использование таких атрибутов, которые позволяют привлечь внимание пользователя к некоторой области экрана.

Выделение слишком многих областей не допустимо, т.к. может не помочь, а запутать пользователя.

Выделение данных осуществляется с помощью цвета символов, цвета фона, уровня яркости и режима мерцания.

Мерцающий фон – самое сильное средство привлечения внимания. И в силу этого и самый отвлекающий фактор. Используется поэтому редко.

Цвет – второй по значимости фактор привлечения внимания. Разные цвета спектра воспринимаются по-разному. Области, фон которых представлен более теплыми оттенками в красной области спектра, кажутся крупнее, чем области, цвет фона которых смещен в голубу. Области экрана на белом фоне или на фоне цвета, который находится в средней области спектра, кажутся ярче и легче воспринимаются. Наибольший контраст между двумя областями экрана достигается, когда цвет фона одной из них – черный или близок к любому из концов спектра, а цвет фона другой – белый или взят из средней части спектра. Те же рассуждения справедливы для соотношения между цветами переднего плана и фона.

Яркость изображения – наименее назойливый способ привлечения внимания. Чаще используются для вывода наименований групп полей.

Для выделения данных можно использовать эффект подчеркивания значения поля или вывода другим шрифтом, а также звуковой сигнал.

Автоматизация проекта с помощью элементов управления

Практически любая программа имеет две основные компоненты: визуальную, с помощью которой организуется пользовательский интерфейс, и программную, реализующую поставленную задачу.

Визуальные компоненты программ в VBA могут быть созданы с помощью встроенных окон; элементов управления; меню и форм.

Интересные проекты создаются с помощью элементов управления. Они позволяют, не нарушая структуры рабочих листов, создавать вспомогательные структурные компоненты, с помощью которых можно «передвигаться» по рабочей книге, выполнять моделирование изучаемого процесса и т.д.

Организация работы с элементами управления

Элемент управления – это объект графического интерфейса, позволяющий пользователю управлять проектом. С помощью элементов управления можно отображать данные, осуществлять выбор параметров, автоматизировать выполнение действий и т.д.

Элементы управления могут размещаться на рабочем листе Excel и в форме. В первом случае говорят об элементах управления *ActiveX* (панель инструментов *Элементы управления* - рис.5.8), во втором – об элементах управления панели инструментов *Форма* (рассматривается в соответствующем разделе пособия). В дальнейшем будем именовать их как «элементы управления» и «элементы управления форм» соответственно.

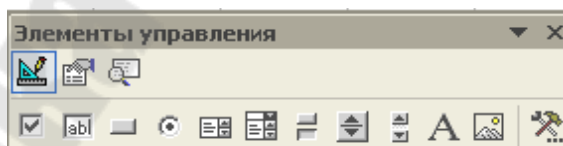


Рис.5.8

Список основных элементов управления и их краткое описание:

Кнопка	Элемент управления	Имя	Префикс	Использование/Описание
--------	--------------------	-----	---------	------------------------



Поле

TextBox

txt

Отображает данные и используется для ввода и вывода информации











Надпись

Label

lbl

Отображает неизменяемый текст, например, за

	Кнопка	<i>Command-Button</i>	<i>cmd</i>
	Список	<i>ListBox</i>	<i>lst</i>
	Поле со списком	<i>ComboBox</i>	<i>cbo</i>
	Счетчик	<i>SpinButton</i>	<i>spn</i>
	Флажок	<i>CheckBox</i>	<i>chk</i>
	Переключатель	<i>OptionButton</i>	<i>opt</i>
	Выключатель	<i>ToggleButton</i>	<i>tgl</i>
	Полоса прокрутки	<i>ScrollBar</i>	<i>scr</i>

головки других элементов управления

Создает кнопку, при нажатии которой выполняется запрограммированное действие

Позволяет создать список из требуемых значений

Текстовое поле с раскрывающимся списком. Используется либо для выбора элемента из списка, либо для ввода значения в поле

Кнопка, которая может быть вложена в ячейку или текстовое поле. Стрелка вверх служит для увеличения, а стрелка вниз — для уменьшения значения

Включает или выключает действие определенного параметра

Кнопка, используемая для выбора только одного параметра из группы

Создает кнопку, имеющую два состояния: включено и выключено

Прокручивает список значений при нажатии стрелок прокрутки или перемещении ползунка

Элементы управления являются объектами. Поэтому, как и любой объект VBA, они обладают уникальными свойствами, методами и событиями. С помощью, например, свойств элементов управления можно присваивать им уникальные имена, задавать размер, устанавливать режим отображения: видимый или невидимый, - и т.д. Методы позволяют, например, установить фокус на вызвавшем этот метод элементе управления, перемещать элементы управления. Однако управление проектом осуществляется благодаря событиям, связанным с элементами управления. На-

пример, щелчок на кнопке активизирует конкретную цепочку действий, на переключателе – позволяет сделать выбор конкретного параметра и т.д. Для этого в проекте должны быть предусмотрены специальные процедуры.

Процедуры, реализующие управление проектом с помощью элементов управления, называются *событийными процедурами* или *процедурами обработки событий*.

В VBA существует целая коллекция возможных событий, связанных с элементами управления. Очевидно, что

1. Конкретному элементу управления соответствует свой набор событий. Например, элемент управления **Кнопка** может активизировать событие **Click (Щелчок)**, но никак не связан, например, с событиями **Error (Ошибка)** или **KeyPress (Нажатие клавиши)**.

2. Одно и то же событие реализует разные действия для разных элементов управления, даже если они являются однотипными. Например, событие **Click (Щелчок)**, связанное с одной кнопкой, может активизировать вывод на рабочий лист встроенного окна с конкретным сообщением и это же событие **Click (Щелчок)**, но связанное с другой кнопкой – переход к новому рабочему листу.

Создание процедуры обработки события осуществляется в три этапа:


- размещение на рабочем листе Excel элемента управления;
- присваивание требуемых значений свойствам элемента управления;
- ввод инструкций и операторов, реализующих конкретные действия конкретного события.

Размещение на рабочем листе Excel элемента управления:

1. Вывести панель управления **Элементы управления**, активизировав команды

Вид – Панели инструментов – Элементы управления

2. Включить режим конструктора элементов управления с помощью

нажатия кнопки **Режим конструктора**  на панели инструментов **Элементы управления**. Помимо включения требуемого режима работы происходит замена назначения кнопки **Режим конструктора** противоположным: теперь кнопка предназначена для выхода из режима конструктора и называется **Выход из режима конструктора**. По умолчанию режим конструктора включается автоматически при выводе панели инструментов **Элементы управления**.

3. Создать требуемый элемент управления. Для этого на панели инструментов необходимо нажать соответствующую кнопку и с помощью мыши перетащить рамку элемента управления в нужное место на рабочем листе. Поскольку любой элемент управления является графическим объектом, при его создании можно воспользоваться клавишами **Shift** или **Alt**. В

первом случае будет реализована возможность пропорционального изменения размеров, а во втором – привязки элемента управления к сетке рабочего листа.

4. Выйти из режима конструктора с помощью кнопки ***Выход из режима конструктора*** панели инструментов ***Элементы управления***. Если на рабочий лист необходимо вывести несколько элементов управления, то выход из режима конструктора выполняется после вывода последнего элемента управления.

Созданный элемент управления на рабочем листе можно перемещать, изменять его размеры, копировать и извлекать из буфера обмена, удалять с рабочего листа.

Перемещение элемента управления: указатель мыши разместить в области элемента управления – появится четырехнаправленная стрелка (маркер перемещения), нажать левую кнопку мыши и, не отжимая ее, переместить элемент на новое место рабочего листа.

Изменение размера элемента управления осуществляется с помощью маркеров изменения размеров объекта, которые появляются при его выделении.

Копирование и вставка из буфера обмена, а также *удаление* элемента управления осуществляются стандартными способами.

Присваивание требуемых значений свойствам элемента управления и ввод инструкций и операторов, реализующих конкретные действия конкретного события, можно выполнять как после размещения всех элементов управления на рабочем листе, так и непосредственно после вывода каждого элемента управления, но только в режиме работы конструктора. В первом случае требуемый элемент управления следует выделить, т.е. щелкнуть по нему левой кнопкой мыши.


Присваивание требуемых значений свойствам элемента управления рассмотрим на примере свойства *Name* (*Наименование*) т.к. оно присуще всем без исключения элементам управления.

Свойство *Name* (*Наименование*) задает уникальный идентификатор, с помощью которого к конкретному объекту можно обращаться из программы. Очевидно, что каждый элемент управления должен иметь уникальное имя. При создании элемента управления на рабочем листе VBA присваивает ему имя по умолчанию (табл.13.1.). Например, *CommonButton1* – имя первой кнопки, расположенной на конкретном рабочем листе. При выводе на рабочий лист второй кнопки ей автоматически будет присвоено имя *CommonButton2*. Однако объектам дают так называемые значащие имена, которые не только однозначно идентифицируют элемент управления, но поясняют его назначение. Например, кнопка *Прибыль*, список *Сотрудники*.

Для того чтобы по имени объекта можно было определить и его тип: кнопка, список, текстовое поле и т.д. – хороший стиль программирования

предполагает использования префикса. Например, с объектом кнопка может быть связано такое имя - *cmdПрибыль*, со списком – *lstСотрудники*.

Для того чтобы задать свойство *Name*, следует выбрать это свойство в окне свойств и ввести новое имя. На этапе конструирования элемента управления для активизации окна свойств объекта необходимо нажать

кнопку **Свойства**  панели инструментов **Элементы управления**.

Окно свойств (*Properties*) показано на рис.5.9. Слева – окно свойств объекта «Кнопка» с именем, присвоенным по умолчанию, справа – окно свойств этого же объекта с новым именем.

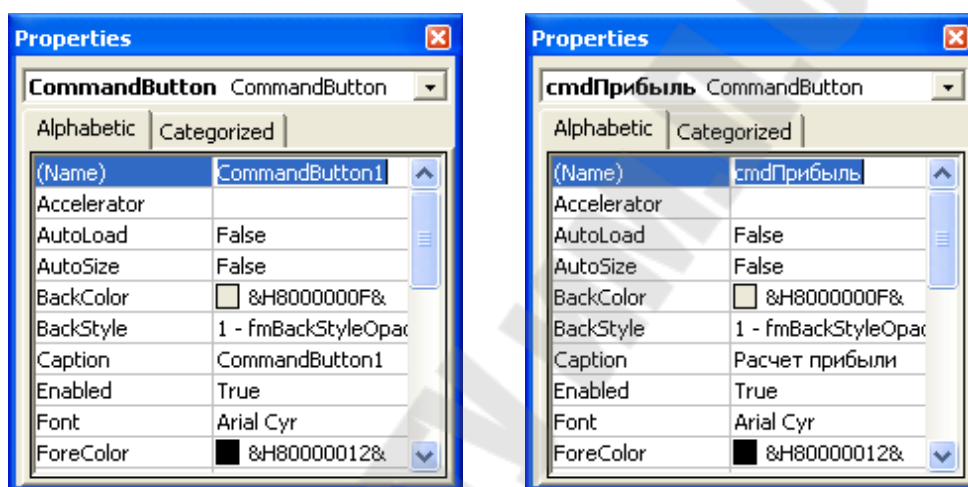


Рис.5.9

Значение свойства *Name* объекта «элемент управления» помимо окна свойств отображается в поле вывода элементов списка объектов *Object* окна модуля рабочего листа, на котором помещен элемент управления (рис.5.10).

Ввод инструкций и операторов, реализующих конкретные действия конкретного события, осуществляется также в режиме конструктора после

активизации кнопки **Исходный текст**  панели инструментов **Элементы управления**. При этом

1. Происходит переход в редактор VBA.
2. Автоматически загружается модуль того рабочего листа, на котором выведен элемент управления (рис.5.10).

3. В окне модуля выводится заготовка будущей процедуры обработки события: строка заголовка и окончания процедуры. Тип процедуры – *Sub*. Область действия процедуры – модуль рабочего листа, поэтому перед типом процедуры записывается служебное слово *Private*. Имя процедуры формируется автоматически и состоит из двух частей: значение свойства *Name* объекта «элемент управления» и идентификатор события (табл.5.11), разделенных символом подчеркивания.

4. С помощью выпадающего списка процедур окна модуля можно сформировать заготовку процедуры обработки любого события, связанного с конкретным элементом управления.

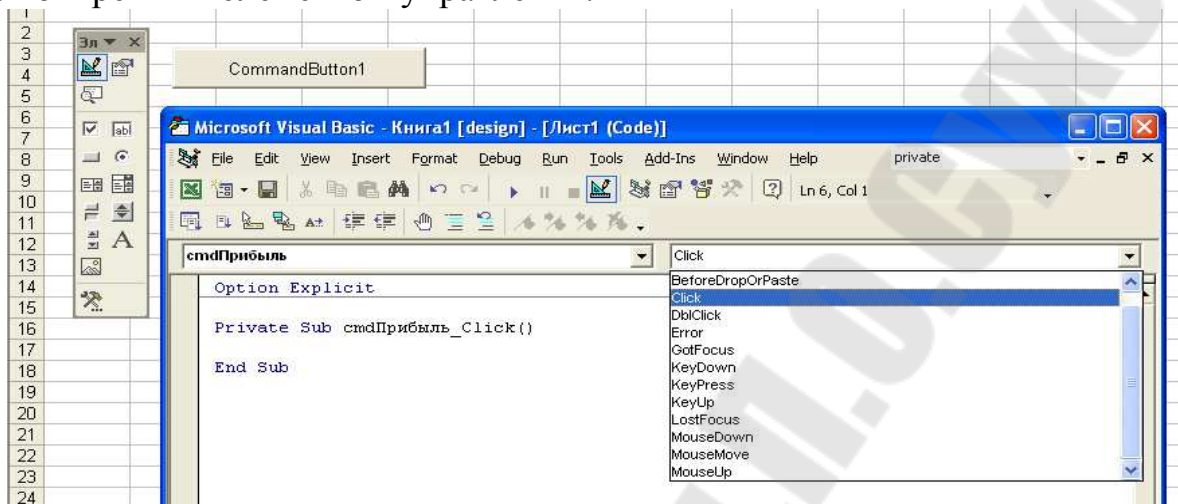


Рис.5.10

Поясним рис.13.3. На рабочем листе с именем *Лист1* были выведены панель управления *Элементы управления* и элемент управления *Кнопка*. Текст на кнопке - *CommandButton1* – определен в соответствии с соглашением «по умолчанию». В модуле рабочего листа «*Лист1*» создана заготовка процедуры обработки события *Click (Щелчок)*, область действия которой распространяется на рабочий лист «*Лист1*». Имя процедуры *cmdПрибыль_Click()* состоит из двух частей: имя объекта *cmdПрибыль* и название события *Click*. Поскольку в процедуру и из нее не передаются параметры, то после имени процедуры в круглых скобках ничего не указывается.

Напишем процедуру обработки события *Click* для кнопки с именем *cmdПрибыль*. Щелчок левой кнопки мыши по этой кнопке должен активизировать встроенное окно вывода сообщения «*Нажата кнопка*». Для этого текст процедуры *cmdПрибыль_Click()* должен быть изменен так, как показано на рис.5.11.



Рис.5.11

Для проверки работы процедуры обработки события необходимо выйти из режима конструирования. Кнопка на рабочем листе примет вид, как на рис.13.5 слева. Затем щелкнуть левой кнопкой мыши по кнопке *CommandButton1*. Произойдет вывод окна сообщения как на рис.5.12 спра-

ва. Отсутствие в кнопке пунктирной рамки – признак активизации этого элемента управления.

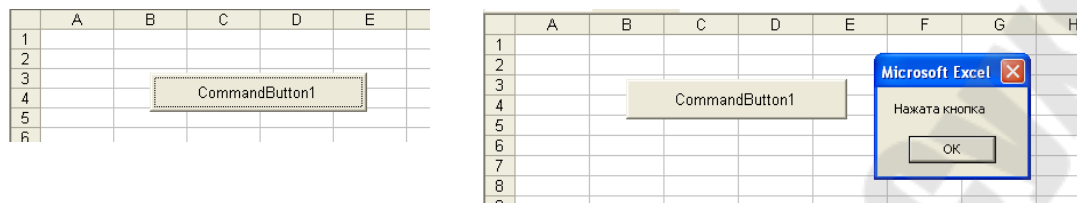



Рис.5.12

Как отмечалось выше, в данном разделе рассматриваются вопросы использования элементов управления как самостоятельных элементов управления проектом. Чаще всего для этих целей используются элементы управления *Кнопка*, *Счетчик*, *Список*, *Поле со списком*, *Переключатель*.

Элемент управления *Кнопка*

Элемент управления *CommandButton* (*Кнопка*)  позволяет инициировать выполнение некоторого конкретного действия. Пожалуй, это самый распространенный управляющий элемент в приложениях Windows.

Объект *CommandButton* () характеризуется большим количеством свойств. С полным перечнем их можно познакомиться помощью окна свойств. Наиболее часто используемые свойства следующие:

<i>Name</i>	<i>Enabled</i>
<i>Caption</i>	<i>Visible</i>

Свойство *Name* подробно описывалось в предыдущем разделе.

Как отмечалось выше, по умолчанию на объекте *Кнопка* размещается текст «*CommandButton1*», т.е. *Кнопка1* или «*CommandButton2*» и т.д. Очевидно, что такой текст ни коим образом не раскрывает предназначение кнопки. Для размещения на кнопке текста, несущего смысловую нагрузку, необходимо свойству *Caption* объекта *Кнопка* присвоить в качестве его значения значение текстовой константы. Проще всего задать значение свойству *Caption* можно с помощью окна свойств в режиме конструирования элемента управления. На рис.5.13 на объекте *Кнопка*, которому присвоено имя *stdПрибыль*, выведен текст «Прибыль». Для этого свойству *Caption* объекта *stdПрибыль* присвоено значение соответствующей константы.

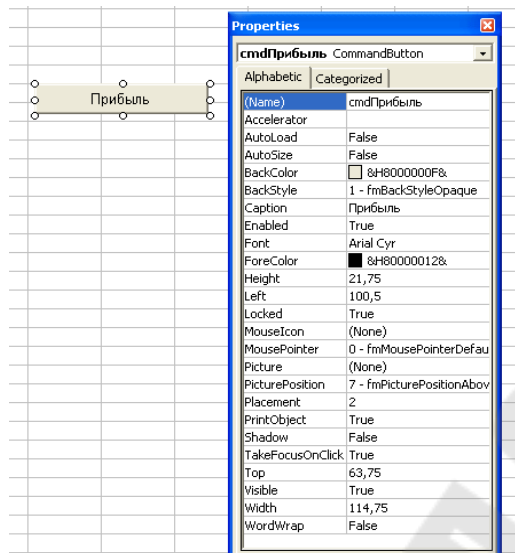


Рис 5 13

Свойства *Enabled* и *Visible* позволяют ограничить (блокировать) доступ к кнопке. Это может быть полезным в тех случаях, когда пользователь перед активизацией кнопки должен выполнить некоторые действия. Например, расчет значения прибыли, осуществляемый с помощью кнопки «Прибыль», возможен только тогда, когда введены исходные данные для расчета.

Свойства *Enabled* и *Visible* являются логическими, т.е. в качестве значений они могут принимать только одно из двух значений: *True* или *False*. Если значение любого из них – *False*, кнопка становится недоступной. Однако если значение *False* присвоить свойству *Enabled*, то заблокированная кнопка остается на экране и поэтому пользователь может догадаться, что при выполнении некоторых действий, она станет доступной. Если же кнопку заблокировать с помощью свойства *Visible*, то она просто исчезает с экрана.

Например, фрагмент следующей процедуры позволит в случае отсутствия на рабочем листе числовых значений, соответствующих расходам и доходам, заблокировать кнопку (рис.5.14а) и рассчитать полученную прибыль в случае наличия таких данных (рис.5.14б).

```
Private Sub cmdПрибыль_Click()
    Dim Rashod As Integer, Dohod As Integer
    Rashod = Range("C4"): Dohod = Range("C5")
    If Rashod = 0 Or Dohod = 0 Then
        cmdПрибыль.Enabled = False
    Else
        MsgBox "Прибыль =" & (Dohod - Rashod)
    End If
End Sub
```

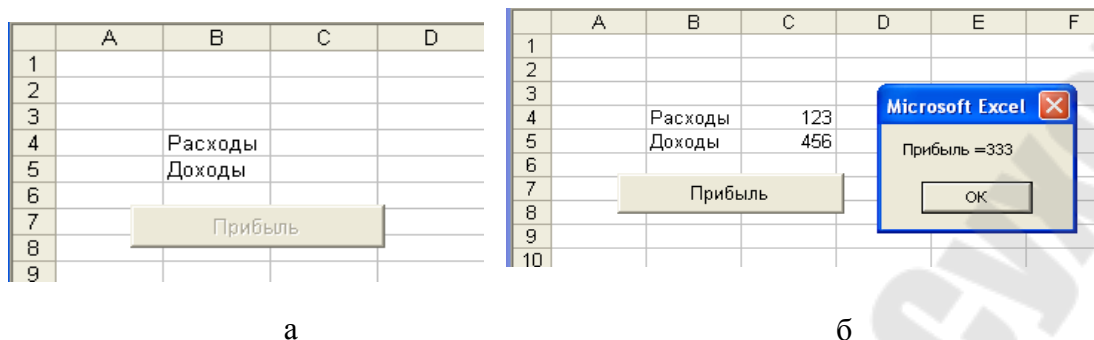


Рис. 5.14

С объектом *Кнопка* связано несколько событий (рис.5.15). Чаще всего используется событие *Click* (*Щелчок*). Процедура *stdПрибыль_Click*, текст которой приведен чуть выше, – реализация события *Click* (*Щелчок*) по кнопке «Прибыль». Остальные события используются в основном после усвоения программирования на VBA.

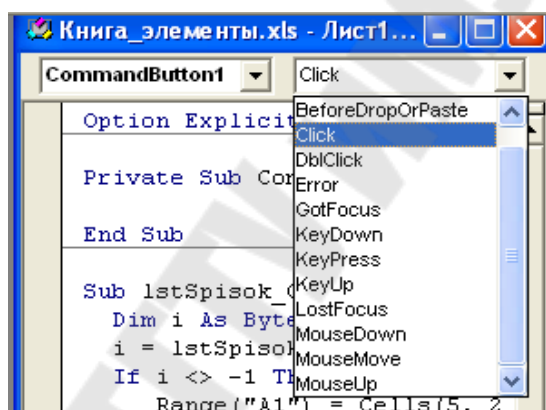


Рис.5.15

Пример разработки проекта с использованием элементов управления *Кнопка*, *Счетчик*, *Переключатель*, *Список* представлен в дополнительном файле «*Элементы управления.xls*»/

Вопросы для самоконтроля

1. Пользовательский интерфейс
2. Структурные элементы интерфейса
3. Критерии качества диалога
4. Структура диалога типа «*Вопрос-ответ*»
5. Структура диалога типа *Меню*
6. Структура диалога на основе экранных форм
7. Выбор выводимой на экран информации
8. Формат вывода данных
9. Общие принципы расположения информации
10. Выделение информации на экране

11. Автоматизация проекта с помощью элементов управления
12. Организация работы с элементами управления
13. Автоматизация проекта с помощью макросов
14. Способы запуска макросов на выполнение

Раздел 5. Менеджмент программных изделий **Менеджер в разработке программных изделий**

Как правило, *разработка программного обеспечения* – это коллективный труд специалистов, направленный на удовлетворение потребности пользователей в автоматизации их деятельности. Как и любой другой коллективный труд, разработка программного обеспечения требует специальной организованности.

Управление разработкой программного обеспечения имеет свои специфические особенности:

1. Как и любой другой труд, тесно связанный с неоднозначными потребностями тех, кто будет использовать продукты труда, необходимым элементом разработки программ является изучение всего круга задач, решаемых пользователем.
2. Обеспечение обратной связи с пользователями.
3. Стремление к переиспользованию ранее созданных программных компонентов.
4. Распространение созданного программного продукта.

Перечисленные аспекты разработки программного продукта определяют специфику деятельности *менеджера программного проекта*, а задачи, решаемые им, зависят от таких аспектов, как *кто* являются участниками разработки и *какие* этапы проходит проект в своем развитии.

Выполнение проекта – это совокупность работ, различных как по своему назначению, так и по квалификационным требованиям, предъявляемым к разработчикам. То есть в ходе развития проекта командой разработчиков выполняются те или иные *функции*. Распределение функций между исполнителями является распределением *ролей* в команде, выполняющей проект.

Значимость ролей разработчиков зависит от того, какой проект выполняется. Однако можно выделить наиболее типичные роли, игнорирование которых приводит к потерям производительности труда команды в целом или, что хуже, к неуспеху развития проекта. Центр объектно-ориентированной технологии фирмы IBM предлагает следующий список ролей разработчиков проекта.

- *Заказчик* – инициатор разработки или уполномоченный принимать решения;
- *Планировщик ресурсов* – выдвигает и координирует требования к проектам, развивает и направляет план выполнения проекта;

- Менеджер проекта – отвечает за развитие проекта в целом; гарантирует, что распределение заданий и ресурсов позволяет выполнить проект; отвечает за своевременность выполнения работ и соответствия их графику, а также предъявляемых требований.

В рамках своих функций менеджер проекта взаимодействует с *Заказчиком* и *Планировщиком ресурсов*.

- Руководитель команды – производит техническое руководство командой в процессе выполнения проекта. Для больших проектов возможно привлечение нескольких руководителей подкоманд, отвечающих за решение частных задач.
- Архитектор – отвечает за проектирование архитектуры системы, согласовывает развитие работ, связанных с проектом.
- Проектировщик подсистемы – отвечает за проектирование подсистемы или категории классов, определяет реализацию и интерфейсы с другими подсистемами.
- Эксперт предметной области – изучает сферу приложения, поддерживает направленность проекта на решение задач данной области.
- Разработчик – реализует проектируемые компоненты; создает специфические классы и методы; осуществляет кодирование и автоматическое тестирование; komponует продукт.

Термин «Разработчик» очень широкий. Он может подразделяться на более узкие роли, например, *Разработчик классов*. Количество разработчиков зависит от сложности проекта.

- Разработчик информационной поддержки – создает документацию, сопровождающую продукт, когда выпускается версия.
- Специалист по пользовательскому интерфейсу – отвечает за удобство применения системы. Работает с *Заказчиком*.
- Тестировщик – проверяет функциональность, качество и эффективность продукта; создает и исполняет тесты для каждой фазы развития проекта.
- Библиотекарь – отвечает за создание и ведение общей библиотеки проекта, которая содержит все проектные рабочие продукты; проверяет и отвечает за соответствие рабочих продуктов стандартам.

В зависимости от проекта и условий его выполнения указанные роли могут совмещаться.

Жизненный цикл программного изделия

Понятие жизненного цикла занимает центральное место в технологии программирования, образуя базу для естественной систематизации инструментов и методов, ресурсов и результатов на различных этапах разработки и использования программных систем.

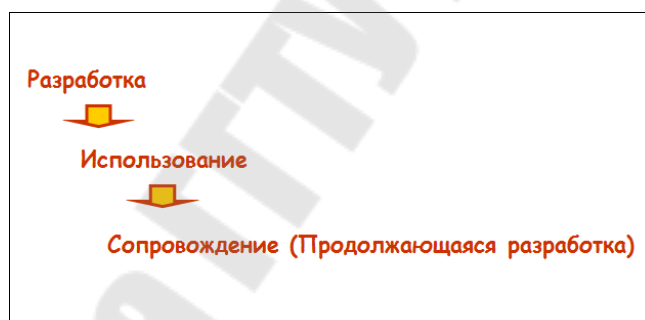
Происхождение термина «*Жизненный цикл программного обеспечения*» объясняется комбинацией термина «*Жизненный цикл*», т.е. «*время жизни*» и термина «*технологический цикл*» или «*цикл разработки*».

Программы не подвержены физическому износу, но в ходе их эксплуатации обнаруживаются ошибки (неисправности), требующие исправления. Ошибки возникают также от изменения условий использования программы. Это принципиальное свойство программного обеспечения, иначе оно теряет свой смысл. Поэтому говорят о моральном старении программ.

Необходимость внесения изменений в действующие программы есть по сути дела продолжение разработки программного обеспечения после передачи его пользователю и в течение всего времени жизни программ.

Деятельность, связанная с решением довольно многочисленных задач такой продолжающейся разработки, получила название *сопровождения программного обеспечения*.

Любой программный продукт развивается по схеме:

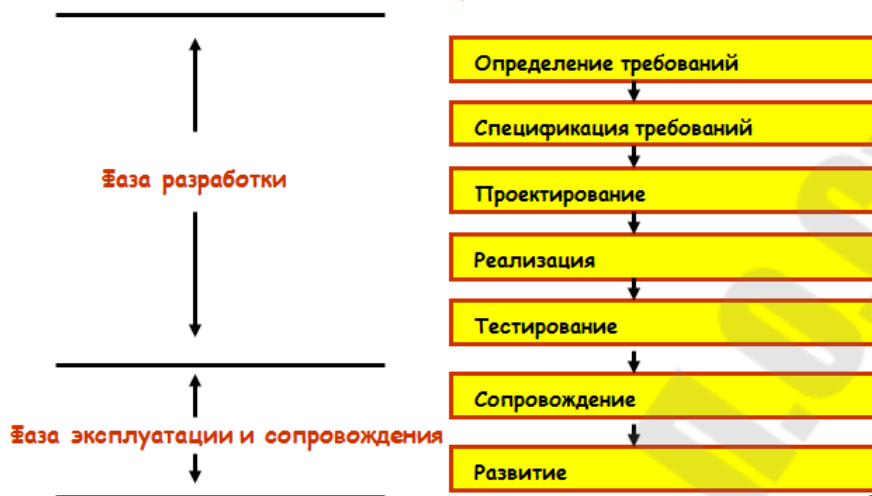


Развитие концепций жизненного цикла связано с поиском для него адекватных моделей. Любая модель является абстракцией реального процесса, в которой опущены детали, несущественные с точки зрения назначения модели. Это позволяет акцентировать внимание на главном. Разнообразие назначений определяет разнообразие моделей.

Модели жизненного цикла программного обеспечения

Самым распространенным мотивом обращения к понятию жизненного цикла является потребность в систематизации работ в соответствии с технологическим процессом. Этому в полной мере соответствует так называемая *общепринятая модель* жизненного цикла программного обеспечения. Согласно ей программные системы проходят в своем развитии 2 фазы: разработка и сопровождение. Фазы разбиваются на ряд этапов.

Общепринятая модель ЖЦПО



Фаза разработки начинается с *идентификации потребности* в новом приложении, а заканчивается передачей продукта разработки в эксплуатацию.

Первый этап фазы разработки – *постановка задачи и определение требований*, – включает описание общего контекста задачи, ожидаемых функций системы и ее ограничений. *Заказчик* совместно с *Разработчиками* принимают решение о необходимости проектирования системы.

Спецификация требований – это результат осмысления разработчиками программного обеспечения выдвигаемых *Заказчиком* требований, которые фиксируются в виде спецификаций системы. Назначение их – описать внешнее поведение разрабатываемой системы, т.е. что должна делать система.

Разработка проектных решений, отвечающих на вопрос как должна быть реализована система, чтобы она могла удовлетворять специфицированным требованиям, выполняется на этапе *проектирования*. Главная задача этого этапа – последовательная декомпозиция системы до уровня очевидно реализуемых моделей или процедур.

На этапе *реализации* или *кодирования* каждый из моделей программируется на наиболее подходящем для данного приложения языке.

Фаза разработки заканчивается этапом *тестирования* (автономного и комплексного) и передачей системы в эксплуатацию.

Фаза эксплуатации и сопровождения включает в себя всю деятельность по обеспечению нормального функционирования программных систем, в том числе фиксирование вскрытых во время эксплуатации программных ошибок, поиск причин и их исправление, повышение эксплуатационных характеристик системы, адаптацию системы к новым требованиям. Поэтому эта фаза разбивается на 2 этапа: собственно сопровождение и развитие.

В ряде случаев на данную фазу приходится большая часть средств, расходуемых в процессе жизненного цикла программного обеспечения.

Общепринятая модель жизненного цикла программного обеспечения является идеальной, т.к. только очень простые задачи проходят все этапы без каких-либо *итераций* – возвратов на предыдущие шаги технологического процесса. Однако при разработке больших и нетрадиционных систем, в связи с выявлением неэффективности реализуемых функций, ошибок, неточностей, допущенных на более ранних стадиях такая ситуация невозможна. Решить данную проблему позволяет *классическая итерационная модель* жизненного цикла, которая дает возможность возвратов на предыдущие этапы разработки и эксплуатации программного продукта.

Классическая итерационная модель ЖЦПО



В настоящее время наиболее востребованы две технологии программирования: традиционная и объектно-ориентированная.

Все традиционные технологии программирования направлены на то, чтобы минимизировать возвраты, т.к. при возврате всегда приходится повторять построение того, что уже считалось готовым.

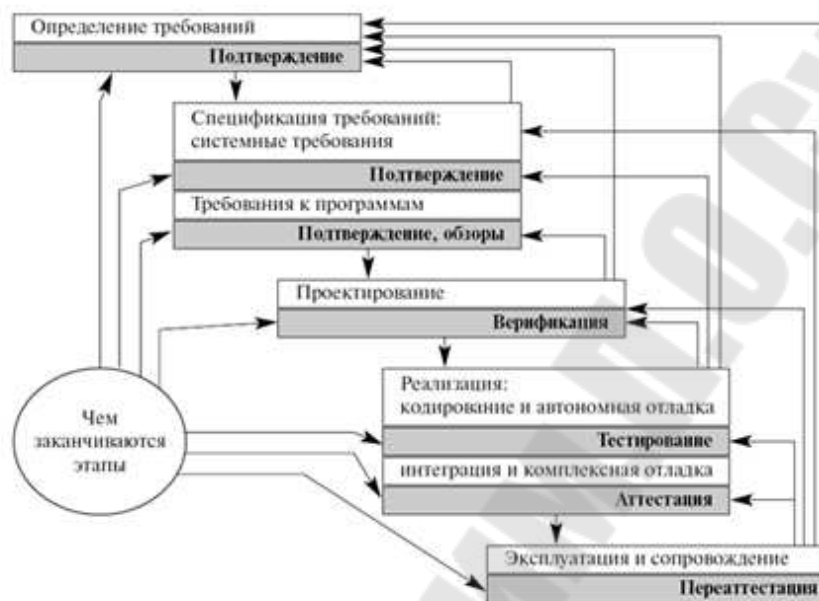
Объектно-ориентированные технологии позволяют отказаться от завершенности фаз и этапов и перейти к наращиванию функциональности и интерфейсных возможностей по итерациям, что в свою очередь позволяет ослабить требования переделки старого при возвратах.

По существу, классическая схема остается верной, но только в рамках одной итерации и с важной поправкой: все полезное, что было сделано ранее, сохраняется.

Существуют несколько моделей.

Каскадная модель – завершение каждого этапа проверкой полученных результатов с целью устранения как можно большего числа проблем, связанных с разработкой изделия; циклическое повторение пройденных этапов.

Каскадная итерационная модель ЖЦТО



Мотивация каскадной модели связана с так называемым управлением качеством программного обеспечения. В связи с ним уточняются понятия этапов, некоторые из них структурируются (*Спецификация* требований и *Реализация*).

Результат проектирования верифицируется, т.е. проверяется, обеспечивают ли принятая структура системы и реализованные механизмы выполнимость специфицированных функций.

Реализация контролируется путем тестирования компонентов, а после интеграции компонентов в систему и комплексной отладки проводится аттестация, т.е. проверка-фиксация фактически реализованных функций системы, описание ограничений реализации и т.д.

В каскадной модели верификация и аттестация приписаны к различным этапам, т.к. верификация – проверка правильности создания программной системы, а аттестация – проверка правильности работы программной системы.

Модель фазы – функции. На классическую итерационную модель накладываются контрольные точки и функции, задающие организационно-временные рамки проекта.

Модель ЖЦТО «Фазы-функции»

	← Исследования →									
	← Анализ осуществимости →									
Фазы (этапы):	← Конструирование →									
	← Программирование →									
					← Оценка →					
									← Использование →	
Контрольные точки (события):					← Спецификации утверждены					
					← 4 Спецификации составлены					
					← 3 Требования утверждены					
					← 2 Требования сформулированы					
					← 1 Ресурсы распределены					
					← 0 Необходимость разработки признама					
					Компоновка завершена 6→					
				Независимые испытания начались 7→						
				Начато изготовление изделия 8→						
				Изделие передано на распространение 9→						
									Изделие снято с производства 10→	

В наиболее последовательном виде такое дополнение классической схемы реализовано в модели Гантера в виде матрицы «Фазы - функции».

Модель Гантера имеет два измерения:

1. *Фазовое*, отражающее этапы выполнения проекта и сопутствующие им события.
2. *Функциональное*, показывающее, какие организационные функции выполняются в ходе развития проекта, и какова их интенсивность на каждом из этапов.

Три итерации проекта, развиваемые одновременно:

Трехитерационный проект

I.	Пл	Ан	Ко	Пр	Те	Оц						
II.				Пл	Ан	Ко	Пр	Те	Оц			
III.							Пл	Ан	Ко	Пр	Те	Оц

Вторая итерация начинается после завершения программирования первой итерации, а третья – в ходе этапа конструирования второй итерации.

Пределы совмещения двух итераций в проекте:

Пл	Ан	Ко	Пр	Те	Оц		
Совмещение не допустимо		Совмещение возможно			Совмещение рационально		
				Последовательное выполнение			

Недопустимость совмещения означает, что для планирования очередной итерации нет достаточно полной информации, как следствие, оно не может быть выполнено эффективно. В ходе конструирования наступает момент, когда такая информация появляется, т.е. есть возможность активизации работ над новой итерацией.

Определение области рационального совмещения работ двух итераций отражает факт неразумного начала этапа программирования новой итерации пока рабочий продукт предыдущей итерации не протестирован.

Область последовательного выполнения указывает на то время, которое соответствует началу следующей итерации после завершения работ над предыдущей, т.е. совмещения нет.

Определение перечисленных областей повышает гибкость распределения времени выполнения проекта. Однако, планируя работы, лучше не рассчитывать на совмещение итераций, а оставлять эту возможность как резерв временного ресурса проекта.

Разработка проекта Менеджмент проекта

Проект – это временная служба, которая пытается предпринять что-либо для создания уникального продукта в области информационных технологий.

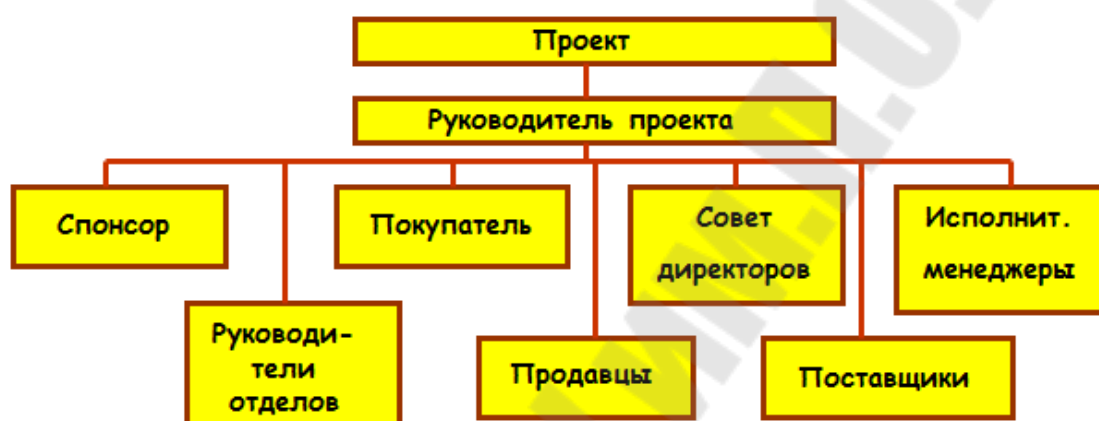
По сути своей проекты временны, для них всегда устанавливается дата старта и окончания. Считается, что проект закончен, если поставленные

перед ним цели достигнуты. Иногда проекты заканчиваются, если установлено, что цели и задачи не могут быть выполнены и проект отменяется.

Проект разрабатывается для того, чтобы создать какой-либо определенный продукт или то, чего до сих пор не существовало. В этом случае проект является уникальным.

Проект считается удачным в том случае, если он оправдывает ожидания его участников. *Участники* – это люди, заинтересованные в проекте.

Схема участников типового проекта



Очень часто участники проекта руководствуются противоречивыми интересами. Основная задача руководителя проекта – понять эти противоречия, конфликты и решить их.

Таким образом, проект имеет несколько характеристик:

- проект сам по себе уникален;
- проекты временны по сути своей и имеют определенную дату начала и завершения;
- проект завершен, если цели проекта достигнуты;
- успешный проект – это проект, который оправдывает ожидания его участников.

Управление проектом – это процесс, который включает в себя несколько составляющих, в том числе планирование, внедрение плана проекта в действие, отслеживание прогресса и исполнение.

Основные функции или задачи:

1. *Планирование* – это одна из наиболее важных функций, которую выполняет руководитель проекта. Это закладывает основу для осуществления проекта и используется для того, чтобы следить за его исполнением.

Руководители проекта – это люди, ответственные за процесс управления проектом и использующие различные методы и средства для его реализации.

2. Для реализации проекта необходимо сочетание трех условий: время, деньги, качество. Как правило, все эти требования носят ограниченный характер. Основная задача менеджера проекта – сбалансировать все три условия с целью удовлетворить или превзойти ожидания участников проекта.

Таким образом, *менеджмент проекта* – это обеспечение поставок продукта, разработка которого требует выполнения определенного объема работ (область действий) с привлечением затрат, не выходящих за определенные пределы, укладывающаяся в заданные рамки времени и удовлетворяющая приемлемому уровню качества.

«Треугольник менеджмента проектов» интерпретирует задачу менеджера: нужно уравновесить производительность работ проекта (область действия), время (план-график) и расход ресурсов (затраты), удовлетворяя требованиям качества.



График проекта. Цель составления – определить сроки начала и завершения операций проекта, в также получение информации о распределении ресурсов.

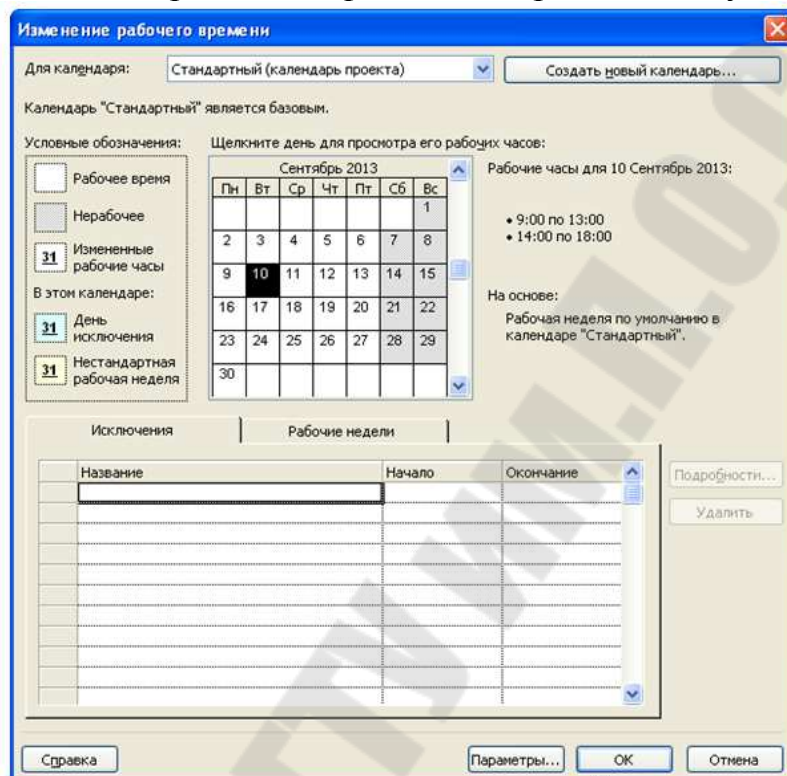
Практически всегда операции проекта могут быть объединены в группы операций или вехи проекта. Вехи отмечают главные достижения или важнейшие события в ходе выполнения проекта.

Для представления операций и вех широко используется диаграмма в виде линий (диаграмма Ганта), которая легко воспринимается. С помощью такой диаграммы можно отследить последовательность операций и вех, сроки их выполнения, применения ресурсов, зависимость между операциями и критический путь.

2. Открыть файл проекта

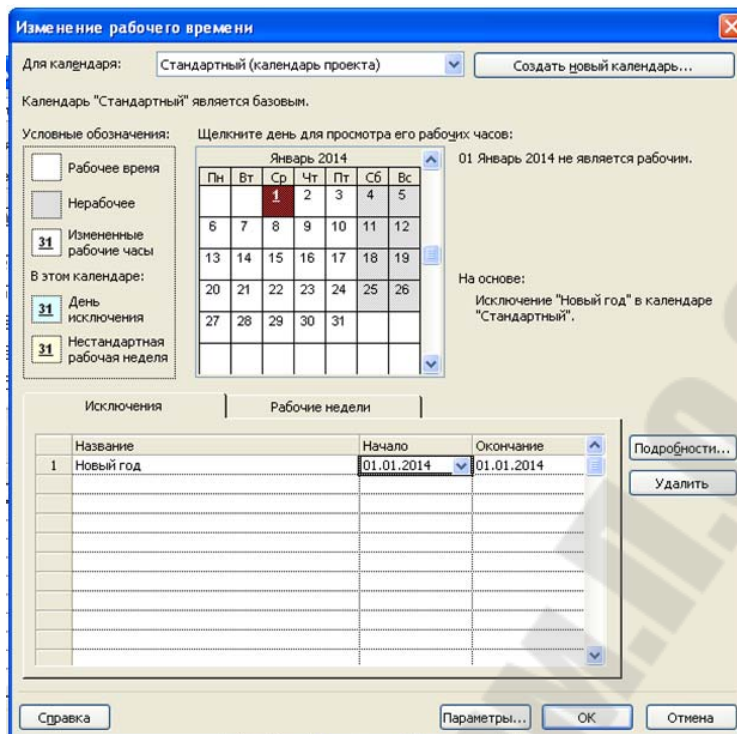
3. Настройка календаря (исключение праздничных дней):

- открыть окно изменения рабочего времени (команда *Сервис – Изменить рабочее время*);
- из календаря *Стандартный* выбрать вкладку *Исключения*:

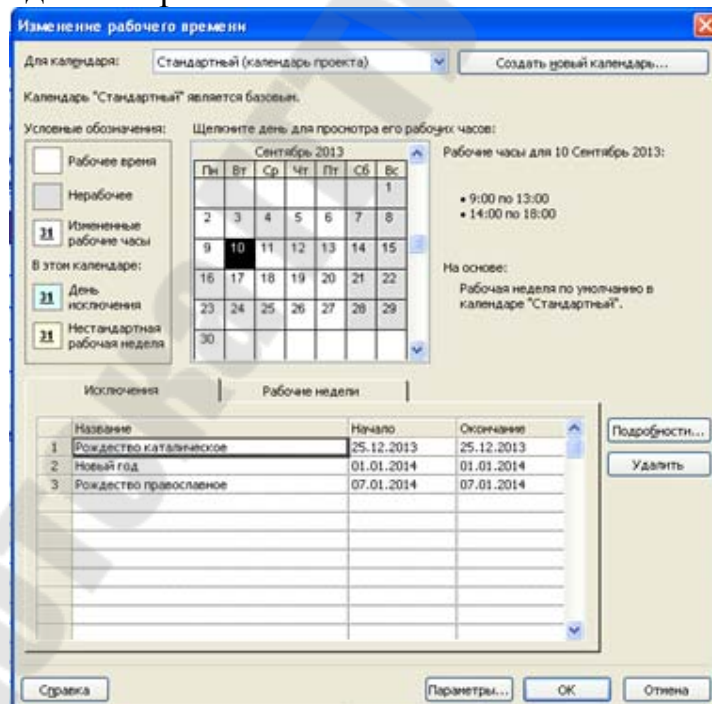


а). в поле *Название* первой пустой строки таблицы ввести «Новый год»;

б). в поле *Начало* – (щелчок) – кнопка выбора – *Календарь* – выбрать дату *1 января 2014г.* По умолчанию устанавливается эта же дата окончания исключения. Исключение считается нерабочим днем.



Задание: исключить из рабочего времени праздничные дни:
Говый год,
Рождество православное.



4. Ввод перечня задач проекта:

- составить список задач проекта, содержащий вехи, фазы и обычные задачи. Расположение задач должно соответствовать последовательности выполнения, а после каждой фазы должны быть перечислены входящие в нее вехи и задачи.

Ввод перечня задач проекта

Начало реализации проекта

Программирование

Постановка задачи

Разработка интерфейса

Разработка модулей обработки данных

Разработка структуры БД

Заполнение БД

Программирование завершено

Отладка

Отладка программного комплекса

Тестирование и исправление ошибок

Составление программной документации

Отладка завершена

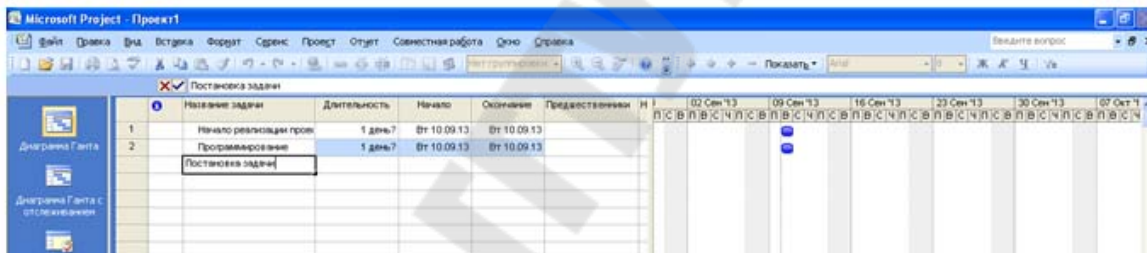
Конец проекта

Веха

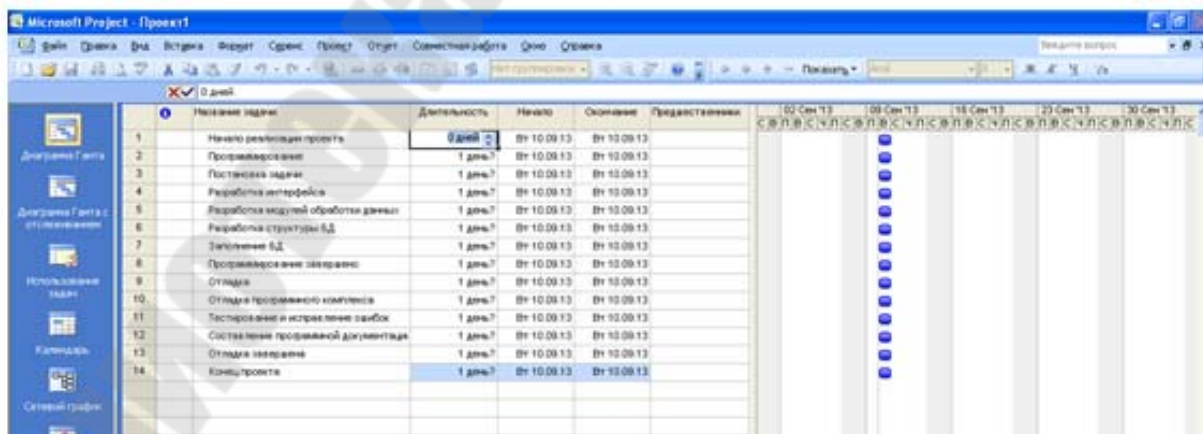
Фаза

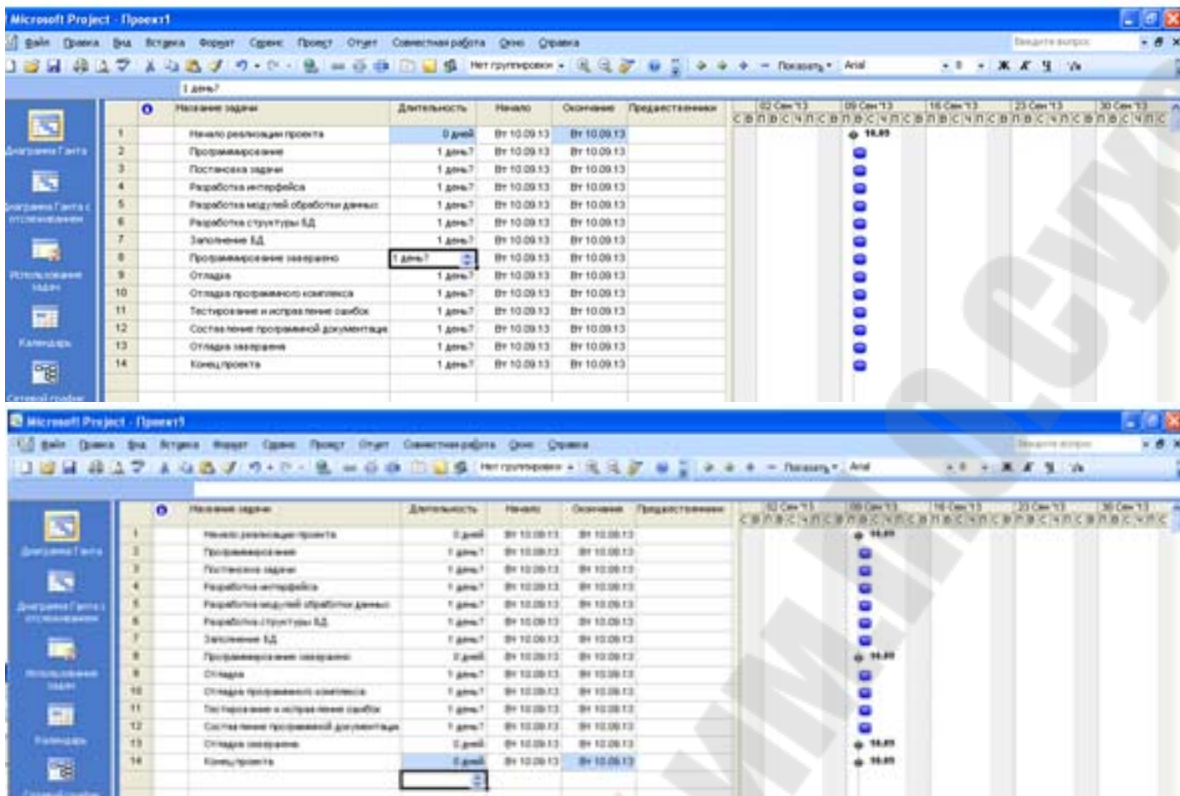
Задача

- сформировать Диаграмму Ганта (Панель представления):
 - а). *Название задачи* – ввести названия вех, фаз и задач. По умолчанию длительность их выполнения 1 день;



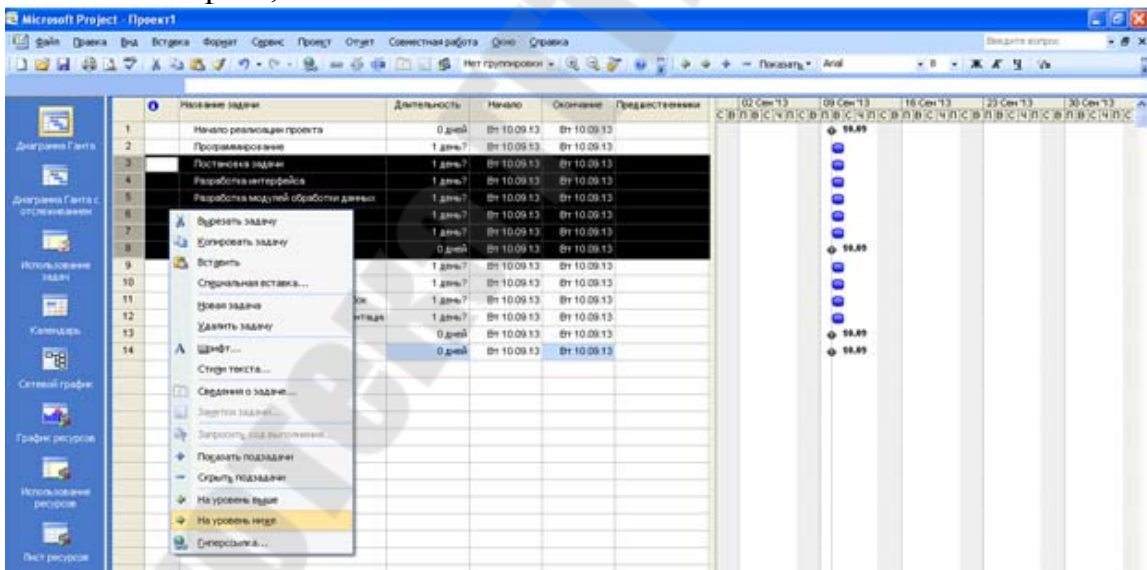
- б). Установка вех. В столбце *Длительность* для всех вех ввести 0 дней. На диаграмме появятся ромбики – символ обозначения вехи;





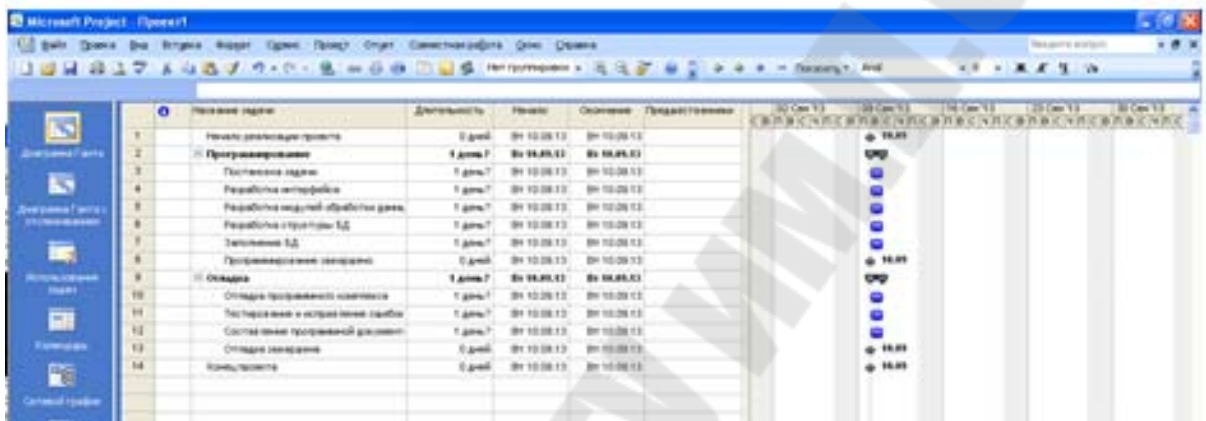
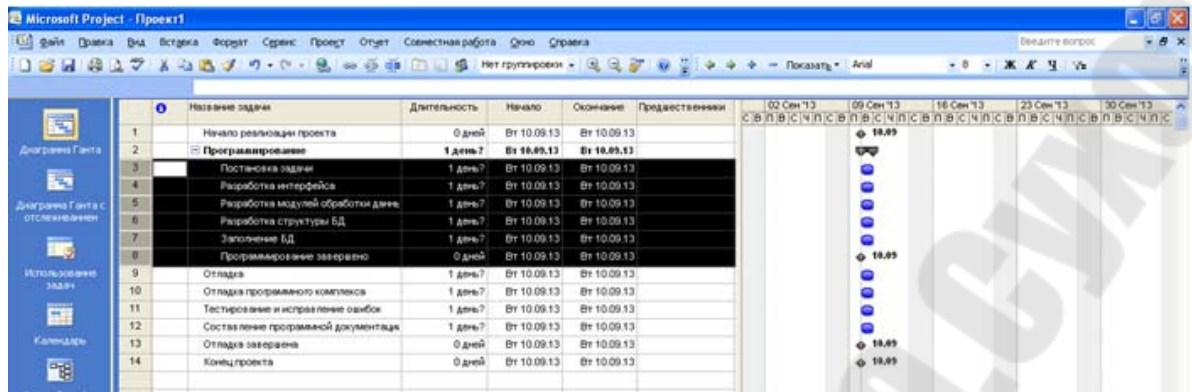
в). Установка фаз:

- в строке задач выделить номера задач, относящиеся к одной фазе;



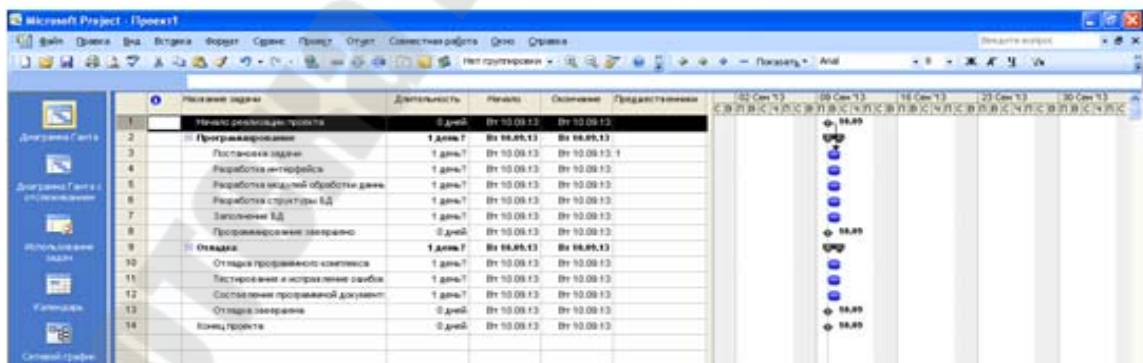
- присвоить статус выделенным задачам *На уровень ниже* с помощью контекстного меню или соответствующей кнопки на *Панели форматирования*.

Фаза – это составная задача. Для преобразования задачи в фазу все подзадачи, входящие в нее должны следовать в таблице задач непосредственно после нее. На диаграмме Ганта фаза изображается отрезком в виде горизонтальной скобки.



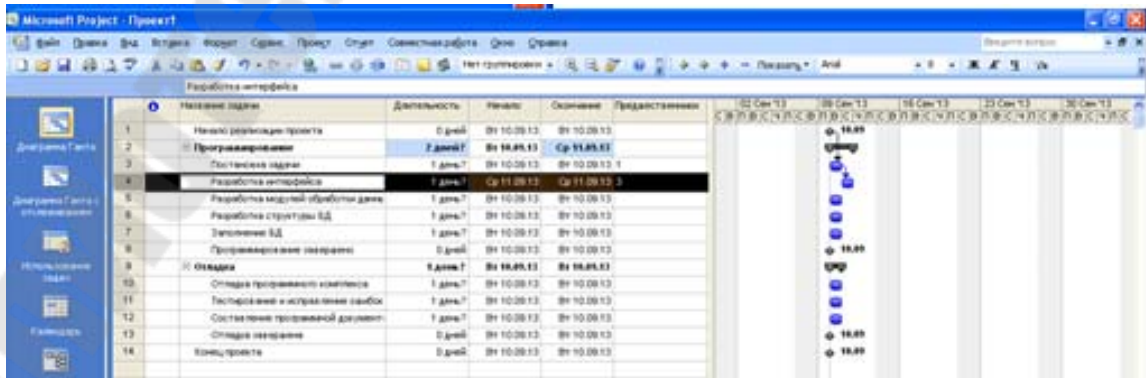
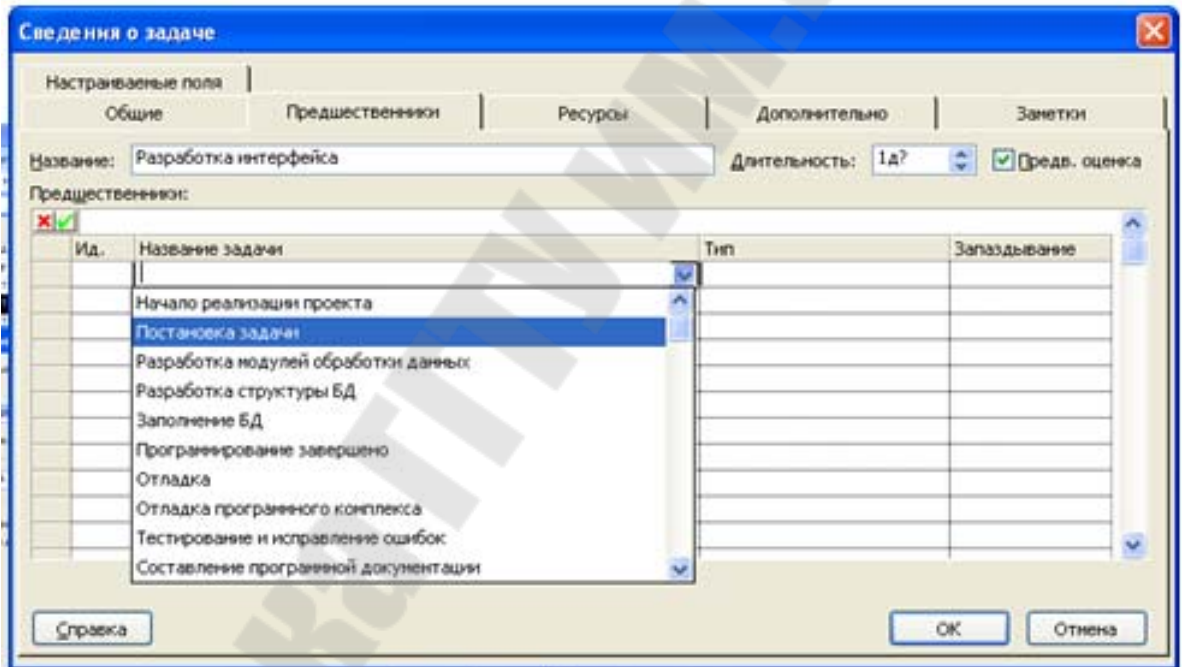
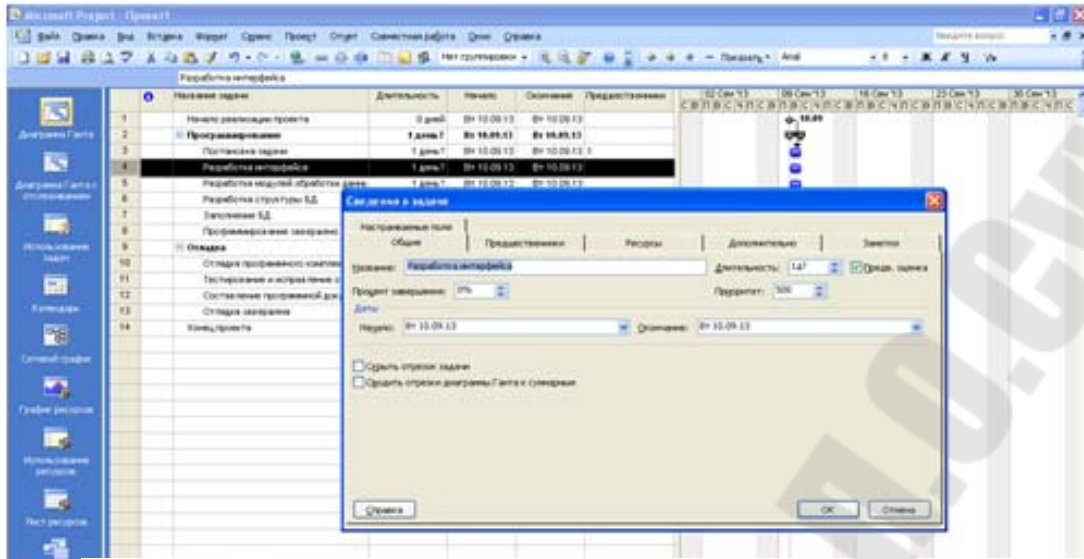
г). Создание связей. Связи указывают последовательность выполнения задач. На диаграмме Ганта связи изображаются в виде стрелки. Создать связь между задачами можно несколькими способами:

С помощью мыши. Ромбик вехи *Начало проекта* выделить и удерживая левую кнопку мыши переместить указатель мыши на *Постановка задачи*.

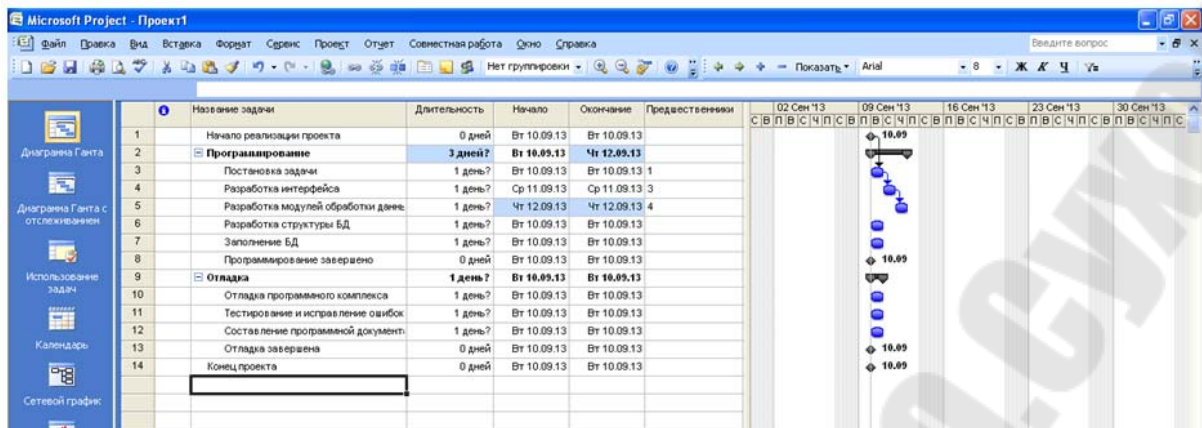


С помощью окна сведений о задаче. Выполнить двойной щелчок по задаче *Разработка интерфейса*. В поле *Название задачи* вкладки *Предше-*

стенники диалогового окна ввести *Постановка задачи.*



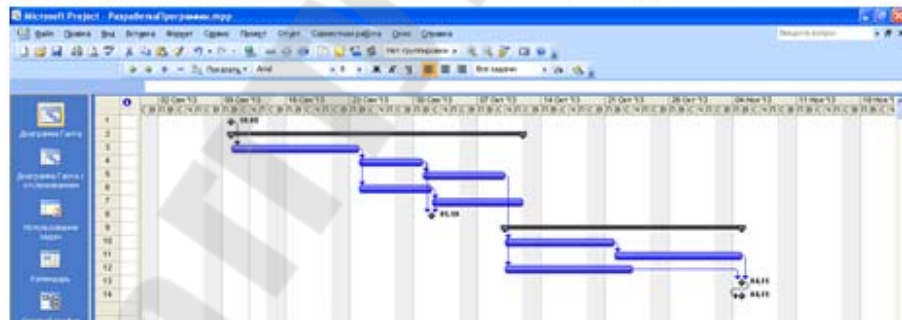
С помощью заполнения столбца *Предшественники* в таблице задач.



Задание: используя рассматриваемые методы создать остальные связи проекта в соответствии с данными таблицы.

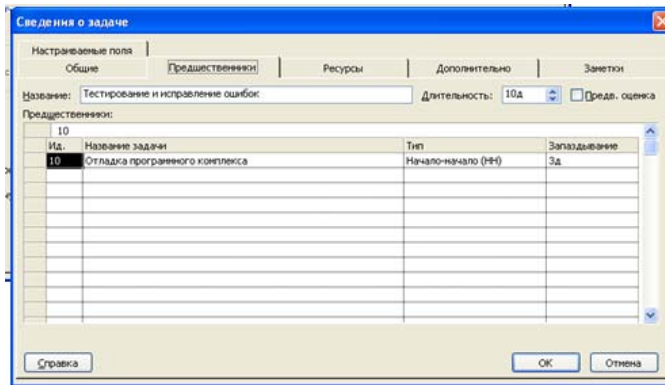
№	Название	Предшественники	Длительность
1	Начало реализации проекта	-	-
2	Программирование	-	-
3	Постановка задачи	1	10
4	Разработка интерфейса	3	5
5	Разработка модулей обработки данных	4	7
6	Разработка структуры базы данных	3	6
7	Заполнение базы	6	8

№	Название	Предшественники	Длительность
8	Программирование завершено	4;6	-
9	Отладка	-	-
10	Отладка программного комплекса	8	5
11	Тестирование и исправление ошибок	10	10
12	Составление программной документации	10	5
13	Отладка завершена	11;12	-
14	Конец проекта	13	-



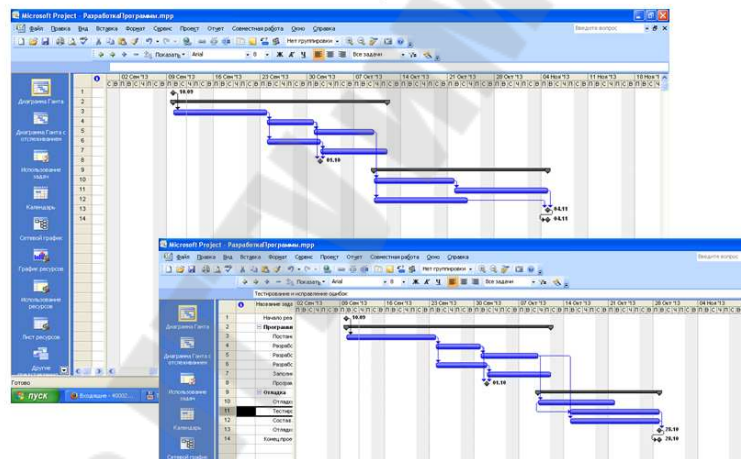
д). Типы связей – задержки, опережения, ограничения. Для указания задержки и опережения необходимо:

- выполнить 2ЛКМ по строке задачи *Тестирование и исправление ошибок* в таблице задач;
- в окне *Сведения о задаче* выбрать вкладку *Предшественники*;



- в столбце *Название задачи* указать *Отладка программного комплекса*, изменить значение поля *Тип* на *Начало-начало*, а в поле *Запаздывание* установить 3д (отрицательное значение означает задержку на 3 дня, а положительное – опережение).

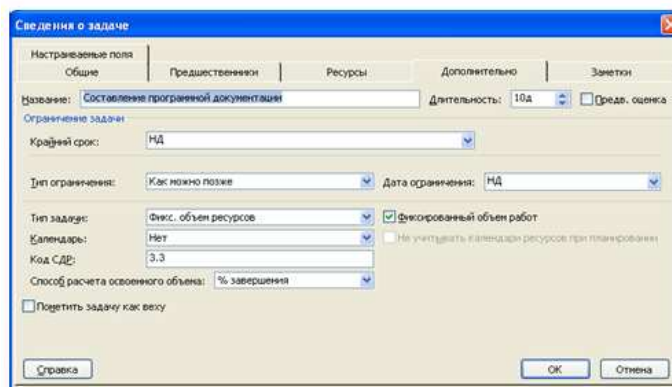
Результат ввода опережения



Для ввода ограничений необходимо:

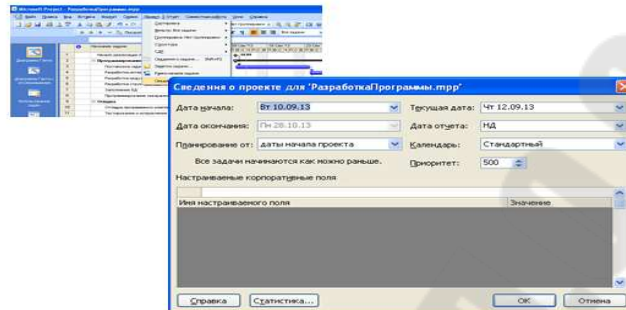
- выполнить 2ЛКМ по строке задачи, например *Составление программной документации*;
- выбрать вкладку *Дополнительно*;
- в поле *Тип ограничений* ввести *Как можно позже*.

Ограничения



- е). Ввод даты начала проекта:
- меню *Проект – Сведения о проекте*;
 - в поле *ДатаНачала* ввести любую дату.

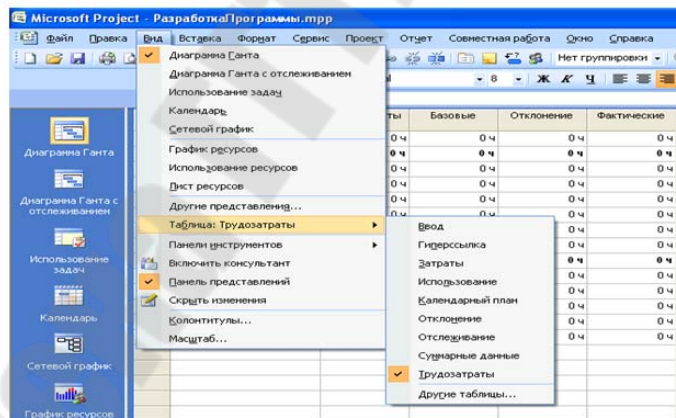
Установка даты начала проекта



Работа с таблицами и представлениями проекта

Таблица задач автоматически может быть преобразована в несколько представлений проекта. Например, в таблицу стоимостных затрат, календарного плана выполнения работ, отклонений, трудозатрат и т.д.

Все виды представлений собраны в списке меню команды *Вид – Таблица*.

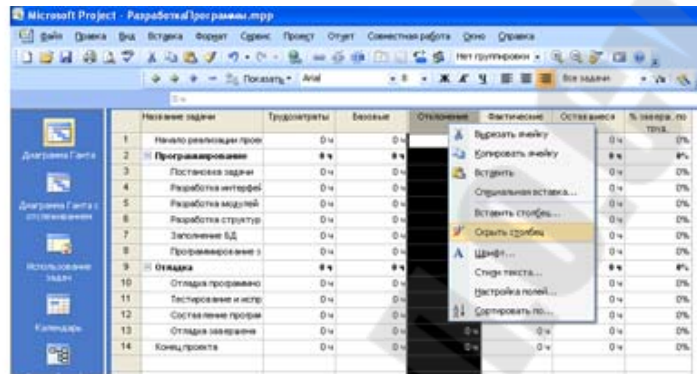


Имя задачи	Трудозатраты	Базовые	Отклонение	Фактические	Оставшиеся	% заверш. по тред.
1 Начало реализации прое	0 ч	0 ч	0 ч	0 ч	0 ч	0%
2 Программирование	0 ч	0 ч	0 ч	0 ч	0 ч	0%
3 Постановка задачи	0 ч	0 ч	0 ч	0 ч	0 ч	0%
4 Разработка интерфей	0 ч	0 ч	0 ч	0 ч	0 ч	0%
5 Разработка модулей	0 ч	0 ч	0 ч	0 ч	0 ч	0%
6 Разработка структур	0 ч	0 ч	0 ч	0 ч	0 ч	0%
7 Заполнение БД	0 ч	0 ч	0 ч	0 ч	0 ч	0%
8 Программирование з	0 ч	0 ч	0 ч	0 ч	0 ч	0%
9 Отладка	0 ч	0 ч	0 ч	0 ч	0 ч	0%
10 Отладка программно	0 ч	0 ч	0 ч	0 ч	0 ч	0%
11 Тестирование и испр	0 ч	0 ч	0 ч	0 ч	0 ч	0%
12 Составление програм	0 ч	0 ч	0 ч	0 ч	0 ч	0%
13 Отладка завершена	0 ч	0 ч	0 ч	0 ч	0 ч	0%
14 Конец проекта	0 ч	0 ч	0 ч	0 ч	0 ч	0%

Например, *Таблица: Ввод* – таблица задач, остальные элементы списка соответствуют конкретным представлениям этой таблицы.

Удаление (скрытие) столбца. Для этого действия достаточно в области конкретного столбца вызвать контекстное меню и выбрать команду *Скрыть столбец*.

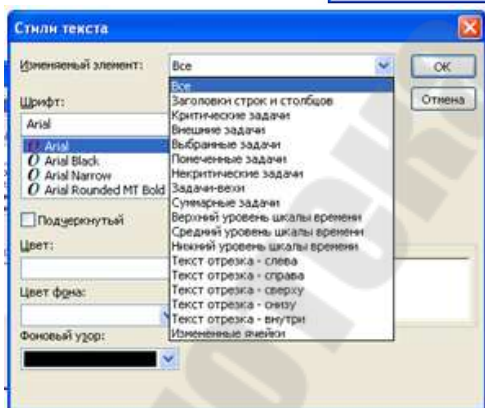
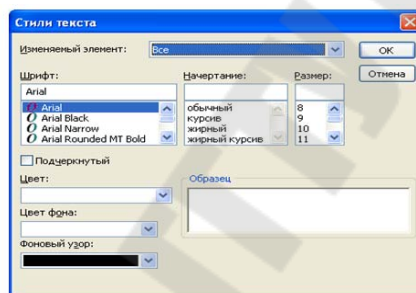
**Удаление
(скрытие) столбца**



Добавление столбца выполняется аналогично.

Назначение стилей текста осуществляется с помощью команды *Формат – Стили текста*.

**Назначение стилей
текста**



Сортировка выполняется с помощью команды *Проект – Сортировка*. Разделяют сортировку predetermined и сортировку пользователя.

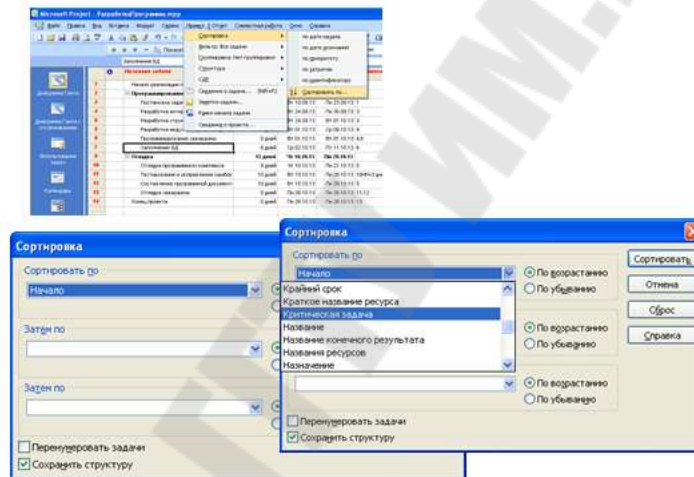
Предetermined сортировка выполняется по дате начала, окончания, по приоритету, по затратам. Отмена сортировки – сортировка по идентификатору.

Сортировка predetermined

Имя задачи	Длительность	Начало	Критическая задача
Начало реализации проекта	0	01.10.13	нет
Проектирование	8	01.10.13	да
Поступление заказа	8	01.10.13	нет
Разработка чертежа	8	01.10.13	нет
Разработка спецификации	8	01.10.13	нет
Разработка модели	8	01.10.13	нет
Подготовка сметы	8	01.10.13	нет
Закупка материалов	8	01.10.13	нет
Отладка программного обеспечения	8	01.10.13	нет
Тестирование и исправление ошибок	10	01.10.13	да
Составление проектной документации	10	01.10.13	нет
Отладка аппаратуры	8	01.10.13	нет
Компьютерная симуляция	8	01.10.13	нет

Сортировка пользователя выполняется по команде *Проект – Сортировка-Сортировать по..* Позволяет отсортировать информацию таблицы или ее представления по трем признакам, например, *Крайний срок, Начало, Критическая задача*.

Сортировка пользователя



Задание: Отсортировать данные таблицы задач по 2 полям: *Критическая задача* и *Длительность*.

Вставить столбец *Критическая задача*. Первая группа задач имеет значение *нет*, вторая – *да*. Результат сортировки можно представить с сохранением структуры или без нее (убирается или остается соответствующий флажок).

Результат сортировки без сохранения структуры

Имя задачи	логически задача	Длительность	Начало	Окончание	Предшественники	Имя задачи
6	Программирование заголовка	Нет	0 дней	Вт 01.10.13	Вт 01.10.13 4,6	
8	Разработка структуры БД	Нет	8 дней	Вт 24.09.13	Вт 01.10.13 3	
7	Заполнение БД	Нет	8 дней	Ср 02.10.13	Пн 11.10.13 6	
7	Начало реализации проекта	Да	0 дней	Вт 10.09.13	Вт 10.09.13	
63	Отладка заголовка	Да	0 дней	Пн 28.10.13	Пн 28.10.13 11,12	
64	Конец проекта	Да	0 дней	Пн 28.10.13	Пн 28.10.13 13	
4	Разработка интерфейса	Да	5 дней	Вт 24.09.13	Пн 30.09.13 3	
8	Разработка модулей обработки данных	Да	7 дней	Вт 01.10.13	Ср 09.10.13 4	
10	Отладка программного комплекса	Да	8 дней	Чт 10.10.13	Пн 21.10.13 5	
3	Постановка задачи	Да	10 дней	Вт 10.09.13	Пн 23.09.13 1	
11	Тестирование и исправление ошибок	Да	10 дней	Вт 15.10.13	Пн 28.10.13 10,11-13 дней	
12	Составление программной документации	Да	10 дней	Вт 15.10.13	Пн 28.10.13 5	
2	Отладка	Да	13 дней	Чт 10.10.13	Пн 25.10.13	
2	Программирование	Да	24 дней	Вт 10.09.13	Пн 11.10.13	

Результат сортировки с сохранением структуры

Имя задачи	логически задача	Длительность	Начало	Окончание	Предшественники	Имя задачи
7	Начало реализации проекта	Да	0 дней	Вт 10.09.13	Вт 10.09.13	
64	Конец проекта	Да	0 дней	Пн 28.10.13	Пн 28.10.13 13	
9	Отладка	Да	13 дней	Чт 10.10.13	Пн 25.10.13	
63	Отладка заголовка	Да	0 дней	Пн 28.10.13	Пн 28.10.13 11,12	
10	Отладка программного комплекса	Да	8 дней	Чт 10.10.13	Пн 21.10.13 5	
11	Тестирование и исправление ошибок	Да	10 дней	Вт 15.10.13	Пн 28.10.13 10,11-13 дней	
12	Составление программной документации	Да	10 дней	Вт 15.10.13	Пн 28.10.13 5	
2	Программирование	Да	24 дней	Вт 10.09.13	Пн 11.10.13	
6	Программирование заголовка	Нет	0 дней	Вт 01.10.13	Вт 01.10.13 4,6	
8	Разработка структуры БД	Нет	8 дней	Вт 24.09.13	Вт 01.10.13 3	
7	Заполнение БД	Нет	8 дней	Ср 02.10.13	Пн 11.10.13 6	
4	Разработка интерфейса	Да	5 дней	Вт 24.09.13	Пн 30.09.13 3	
8	Разработка модулей обработки данных	Да	7 дней	Вт 01.10.13	Ср 09.10.13 4	
3	Постановка задачи	Да	10 дней	Вт 10.09.13	Пн 23.09.13 1	

Фильтрация. Можно выполнить фильтрацию несколькими способами: с помощью структурной фильтрации, автофильтра, предопределенного фильтра и фильтра пользователя.

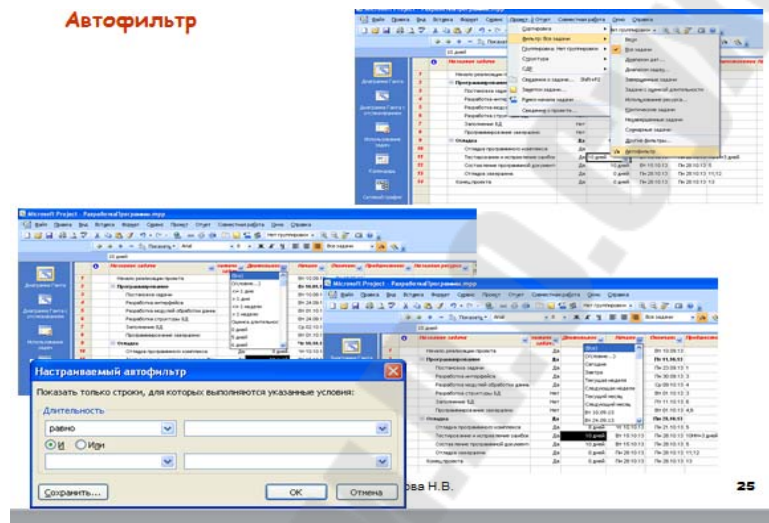
Структурный фильтр активизируется с помощью команды *Проект – Структура – Показать – ..*. Отмена структурной фильтрация осуществляется с помощью команды *Проект – Структура – Показать – Все подзадачи*.

Структурная фильтрация

Имя задачи	логически задача	Длительность	Начало	Окончание	Предшественники	Имя задачи
7	Начало реализации проекта	Да	0 дней	Вт 10.09.13	Вт 10.09.13	
2	Программирование	Да	24 дней	Вт 10.09.13	Пн 11.10.13	
9	Отладка	Да	13 дней	Чт 10.10.13	Пн 25.10.13	
64	Конец проекта	Да	0 дней	Пн 28.10.13	Пн 28.10.13 13	

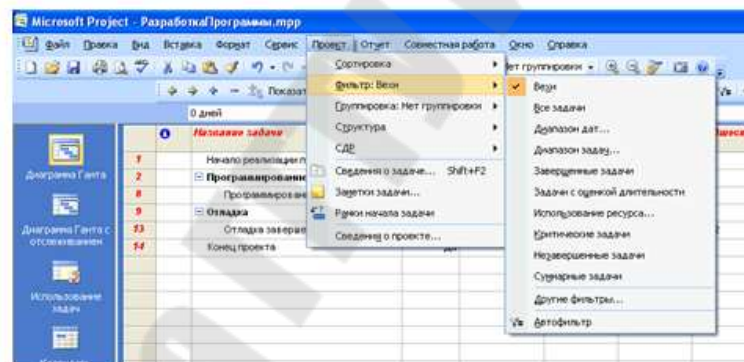
Автофильтр активизируется с помощью команды *Проект – Фильтр – Автофильтр*. В заголовках столбцов появляются элементы раскрывающихся списков, причем перечень элементов списков связан со значениями конкретного столбца. Для указания сложного условия фильтрации исполь-

зуются элемент списка *Условие*. Отмена фильтрации осуществляется после выбора в списке *Автофильтра* пункта *Все*. Для отмены действия *Автофильтра* по всем столбцам необходимо активизировать команду *Проект – Фильтр – Автофильтр*.



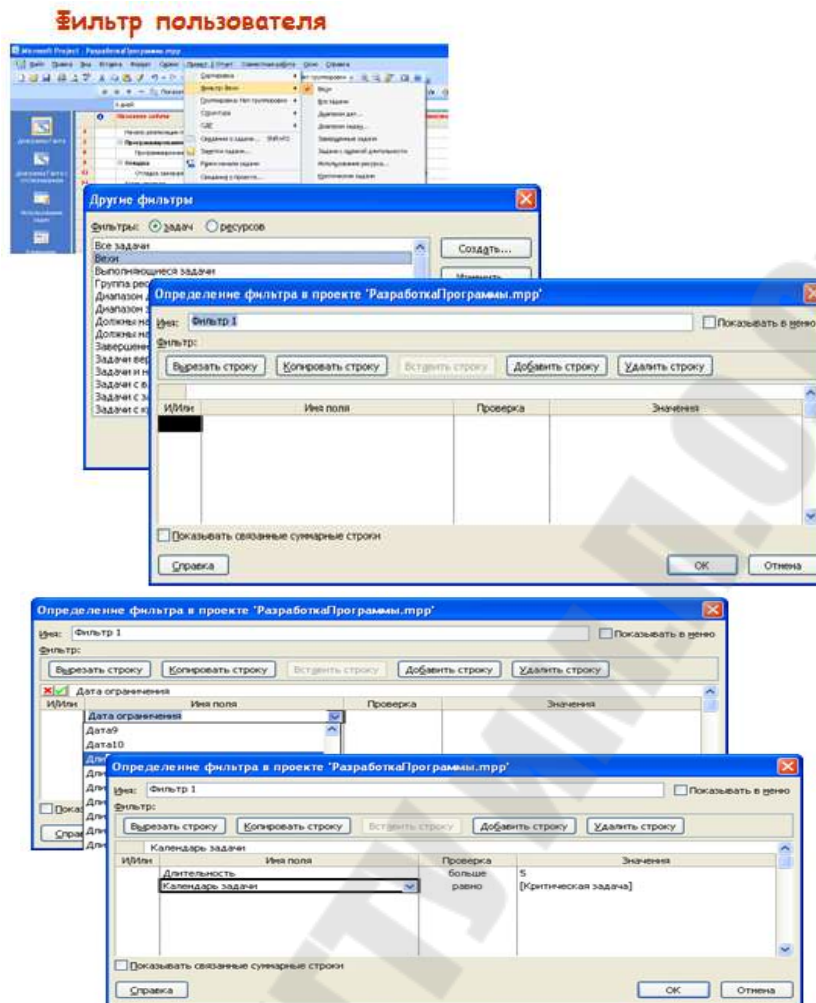
Предопределенный фильтр:

Предопределенный фильтр



Отмена действия предопределенного фильтра активизируется команда *Проект – Фильтр – Все задачи*.

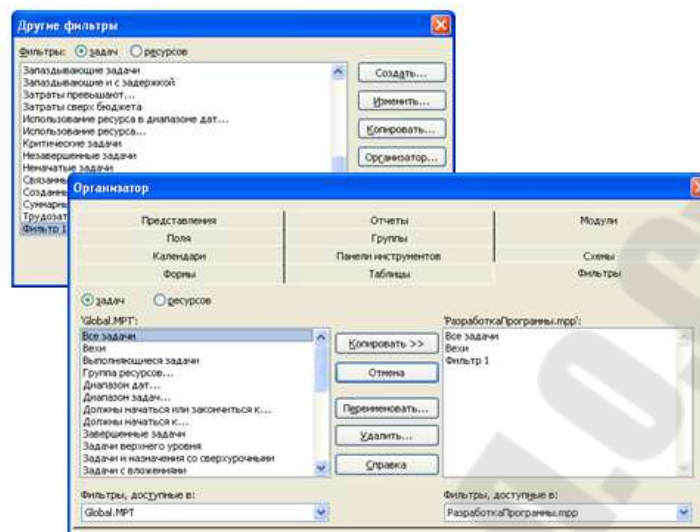
Создание фильтра пользователя активизируется командой *Проект – Фильтр – Другие фильтры*. Откроется окно со списком определенных в системе фильтров. При нажатии кнопки *Создать* откроется окно *Определение фильтра*. Затем заполняются поля с использованием списков выбора. По нажатии кнопки *OK* в списке фильтров окна *Другие фильтры* появится строка с названием созданного фильтра.



Удаление созданного фильтра из системы:

- Проект – Фильтр – Другие фильтры;
- выделить удаляемый фильтр в *Фильтр1*;
- активизировать кнопку *Организатор*;
- в окне *Организатор* в правом поле *РазработкаПрограммы.mpr* выделить *Фильтр1*;
- активизировать кнопку *Удалить*.

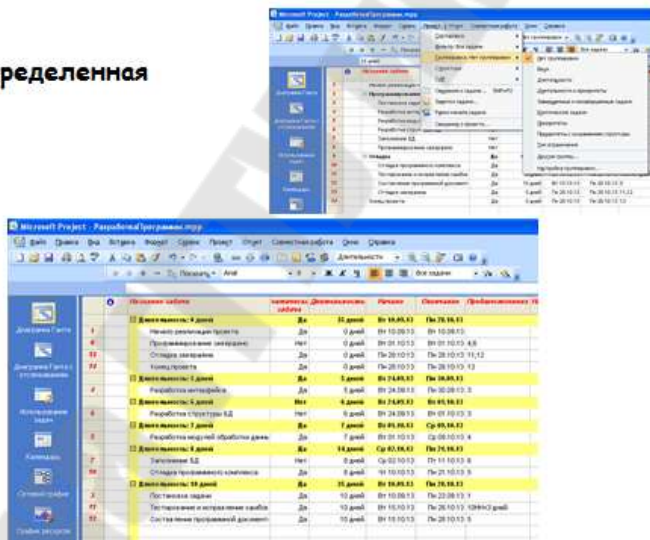
Удаление фильтра пользователя



Группировка. Можно использовать predetermined и разработать пользовательскую группировку.

Группировка

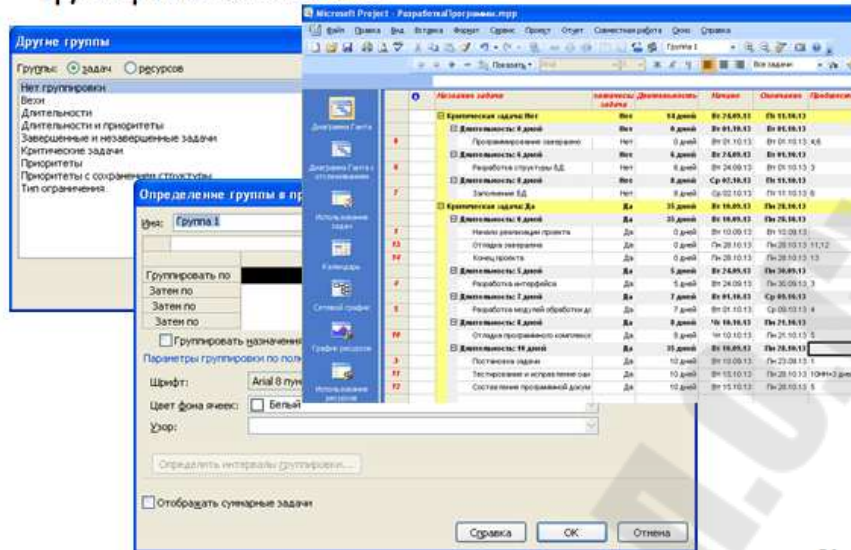
Предопределенная



Для отмены predetermined группировки следует активизировать команду *Проект – Группировка – Нет группировки*.

Для создания пользовательской группировки (*Проект – Группировка – Другие группы*) в окне со списком определенных в системе группировок активизировать кнопку *Создать*, заполнить поля окна *Определение группы*.

Группировка пользователя



Для удаления созданной группировки необходимо активизировать команду *Проект – Группировка – Другие группы*, затем кнопку *Организатор*, в окне проекта выделить группировку и щелкнуть по кнопке *Удалить*.

Список использованной литературы

Основная литература

1. Буза М.К., Певзнер Л.В. Windows – приложения: от операции к реализации: Учебн.пособие. – Минск : Выш.шк., 1998. – 158 с.
2. Додж М., К. Стивенсон. Excel 2002. – СПб.: Питер, 2003. – 992с.
3. Миллхоллон М. К.Мюрей. Word 2002. – СПб.: Питер, 2003. – 944с.
4. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. – М. : ДИАЛОГ- МИФИ, 2003.
5. Черемных С. В. Моделирование и анализ систем. IDEF-технологии: практикум / С. В. Черемных, И. О. Семенов, В. С. Ручкин. – М. : Финансы и статистика, 2006. – 189 с.
6. Автоматизированные информационные технологии в экономике: Учеб. / Под ред. Г. Л. Титоренко. – М. : ЮНИТИ, 2006
7. Благодатских В. А. Стандартизация разработки программных средств. Москва: Финансы и статистика, 2006 Карлберг К. Бизнес-анализ с помощью Excel.: Пер.с англ. – К. : Диалектика, 1997. – 448 с.
8. Самаль С.А. Интеллектуальные модели социально-экономических систем. – Минск, БГЭУ, 2003. – 161 с.
9. Скопин И. Менеджмент программных проектов (материалы к специальному курсу в электронном виде)
10. Хелдман К. Профессиональное управление проектом / К.Хелдман; Пер.с англ. – М. : БИНОМ. Лаборатория знаний, 2005. – 517 с.

Дополнительная литература

11. Современный самоучитель работы на персональном компьютере. Быстрый старт.: Практ.пособие. М.:Триумф, 2007.
12. Информатика. Практикум по технологии работы на компьютере / Под ред. проф. Н. В. Макаровой. – 4-ое изд. – М. : Финансы и статистика. 2008. – 348 с.
13. Браун С. Visual Basic 5 с самого начала. – СПб.: Питер, 1998. – 320 с.
14. Савицкая Г.В. Анализ хозяйственной деятельности предприятия: 5-ое изд./Г.В.Савицкая. – Минск : Новое знание, 2001. – 688 с.

Учебно-методические комплексы

Учебно-методический комплекс по дисциплине «АРМ менеджера»

Перечень компьютерных программ, наглядных и других пособий, методических указаний и материалов и технических средств обучения

15. М/ук № 2534 Водополова Н.В., Шибeko В.Н. Сводные таблицы.
16. М/ук № 2556 Водополова Н.В., Шибeko В.Н. Промежуточные итоги.
17. М/ук № 2535 Водополова Н.В., Шибeko В.Н. Фильтрация данных в MS Excel.
18. М/ук № 2792 Водополова Н.В., Шибeko В.Н., Косинов Г.П. Получение данных в Excel из внешних данных
19. М/ук №2766 Водополова Н.В., Шибeko В.Н., Косинов Г.П. Excel – как средство анализа данных
20. М/ук № Водополова Н.В., Шибeko В.Н., Косинов Г.П. Консолидация и связывание данных
21. М/ук №2980 Водополова Н.В., Чабуркина С.А. Основы программирования на VBA
22. М/ук. №2761 Чабуркина С.А. Использование функций в Excel
23. М/ук. №2607 Асенчик О.Д., Стародубцев Е.Г. Работа с объектами Excel и Access
24. М/ук. №2580 Кравченко О.А. Решение задач оптимизации в MS Excel. Задачи линейного программирования
25. М/ук. Водополова Н.В. Использование основных приемов и методов анализа временных рядов и прогнозирования данных в профессиональной деятельности экономистов

Водополова Наталья Витальевна

**МЕНЕДЖМЕНТ
В ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЯХ**

**Пособие
по одноименной дисциплине
для слушателей специальности
1-40 01 73 «Программное обеспечение
информационных систем»
заочной формы обучения**

Подписано к размещению в электронную библиотеку
ГГТУ им. П. О. Сухого в качестве электронного
учебно-методического комплекса 22.05.15.

Рег. № 17Е.

<http://www.gstu.by>