

Министерство науки и образования Украины
Донбасская государственная машиностроительная академия

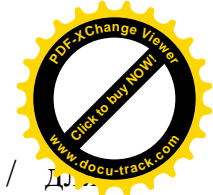
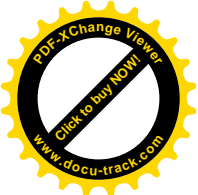
Е.Ю. Ивченкова

ЭКОНОМИЧЕСКАЯ ИНФОРМАТИКА

конспект лекций

Утверждено
на заседании кафедры ИСПР
Протокол № 2 от 09.09.2014 г.

Краматорск 2014

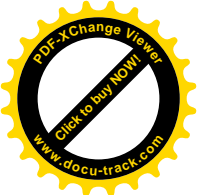


Конспект лекций по дисциплине «Экономическая информатика» / для студентов направления 6.040303 «Системный анализ» очной и заочной формы обучения / Составители: Е.Ю.Ивченкова. – Краматорск: ДГМА, 2014.

Конспект по дисциплине «Экономическая информатика» предназначены для студентов направления 6.040303 «Системный анализ» всех форм обучения, и всех специалистов имеющих дело с обработкой экономической информации при помощи компьютерной техники.

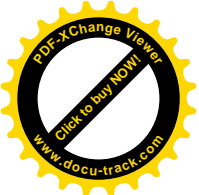
Составитель: Ивченкова Е.Ю., к.э.н., ст. преподаватель

Отв. за выпуск Мельников А.Ю., к.т.н., доц., и.о. зав. кафедрой



СОДЕРЖАНИЕ

ТЕМА 1 Теоретические основы экономической информатики	3
ТЕМА 2 Работа в редакторе электронных таблиц MS Excel	15
ТЕМА 3 Основы программирования на Visual Basic for Applications	31
ТЕМА 4 Работа в MS Access	72
ЛИТЕРАТУРА	122



ТЕМА 1 Теоретические основы экономической информатики

1.1.1. Объект, предмет, методы и задачи экономической информатики

Интенсивное внедрение информационных технологий в экономику привело к появлению одного из направлений в информатике – экономической информатики, которая является интегрированной прикладной дисциплиной, основанной на межпредметных связях информатики, экономики и математики.

Теоретической основой для изучения экономической информатики является информатика. Слово "информатика" (informatique) происходит от слияния двух французских слов: information (информация) и automatique (автоматика), введено во Франции для определения сферы деятельности, занимающейся автоматизированной обработкой информации.

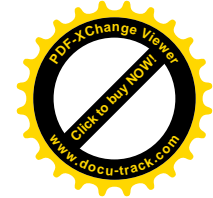
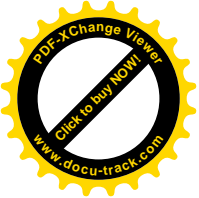
Существует много определений информатики. Информатика - это наука об информации, способах ее сбора, хранения, обработки и предоставления с помощью компьютерной техники. Информатика — это прикладная дисциплина, изучающая структуру и общие свойства научной информации и т.д. Информатика состоит из трех взаимосвязанных составляющих: информатика как фундаментальная наука, как прикладная дисциплина и как отрасль производства.

Основными объектами информатики выступают:

- информация;
- компьютеры;
- информационные системы.

Общие теоретические основы информатики:

- информация;
- системы счисления;



- кодирование;
- алгоритмы.

Структура современной информатики:

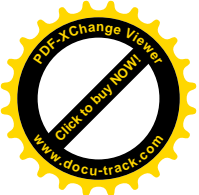
1. Теоретическая информатика.
2. Вычислительная техника.
3. Программирование.
4. Информационные системы.
5. Искусственный интеллект.

Экономическая информатика - это наука об информационных системах, используемых для подготовки и принятия решений в управлении, экономике и бизнесе.

Объектом экономической информатики выступают информационные системы, которые обеспечивают решение предпринимательских и организационных задач, возникающих в экономических системах (экономических объектах). То есть, объектом экономической информатики выступают экономические информационные системы, конечная цель функционирования которых является эффективное управление экономической системой.

Информационная система – это совокупность программно-аппаратных средств, способов и людей, которые обеспечивают сбор, хранение, обработку и выдачу информации для обеспечения подготовки и принятия решений. К основным компонентам информационных систем, используемых в экономике, относятся: программно-аппаратные средства, бизнес-приложения и управление информационными системами. Назначение информационных систем - создание современной информационной инфраструктуры для управления компанией.

Предмет дисциплины "Экономическая информатика" - способы автоматизации информационных процессов с применением экономических данных.



Задача дисциплины "Экономическая информатика" - изучение теоретических основ информатики и приобретение навыков использования прикладных систем обработки экономических данных и систем программирования для персональных компьютеров и компьютерных сетей.

1.1.2. Данные, информация и знания

Основные понятия данных, информации, знаний.

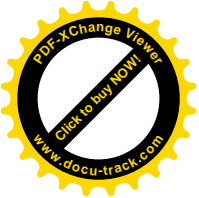
К базовым понятиям, которые используются в экономической информатике, относятся: данные, информация и знания. Эти понятия часто используются как синонимы, однако между этими понятиями существуют принципиальные различия.

Термин данные происходит от слова data - факт, а информация (informatio) означает разъяснение, изложение, т.е. сведения или сообщение.

Данные - это совокупность сведений, зафиксированных на определенном носителе в форме, пригодной для постоянного хранения, передачи и обработки. Преобразование и обработка данных позволяет получить информацию.

Информация - это результат преобразования и анализа данных. Отличие информации от данных состоит в том, что данные - это фиксированные сведения о событиях и явлениях, которые хранятся на определенных носителях, а информация появляется в результате обработки данных при решении конкретных задач. Например, в базах данных хранятся различные данные, а по определенному запросу система управления базой данных выдает требуемую информацию.

Существуют и другие определения информации, например, информация – это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся о них степень неопределенности, неполноты знаний.



Знания – это зафиксированная и проверенная практикой обработанная информация, которая использовалась и может многократно использоваться для принятия решений.

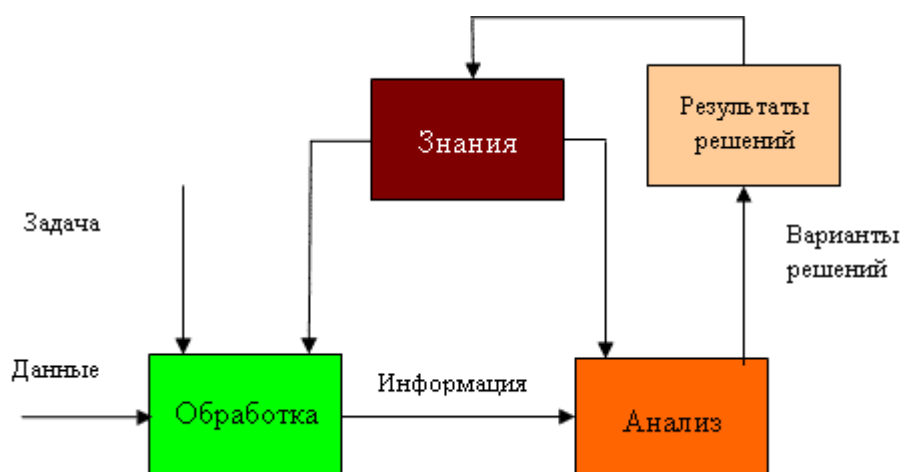
Знания – это вид информации, которая хранится в базе знаний и отображает знания специалиста в конкретной предметной области. Знания – это интеллектуальный капитал.

Формальные знания могут быть в виде документов (стандартов, нормативов), регламентирующих принятие решений или учебников, инструкций с описанием решения задач. Неформальные знания – это знания и опыт специалистов в определенной предметной области.

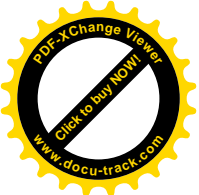
Необходимо отметить, что универсальных определений этих понятий (данных, информации, знаний) нет, они трактуются по-разному. Принятия решений осуществляются на основе полученной информации и имеющихся знаний.

Принятие решений – это выбор наилучшего в некотором смысле варианта решения из множества допустимых на основании имеющейся информации.

Взаимосвязь данных, информации и знаний в процессе принятия решений представлена на рисунке.



Для решения поставленной задачи фиксированные данные обрабатываются на основании имеющихся знаний, далее полученная



информация анализируется с помощью имеющихся знаний. На основании анализа, предлагаются все допустимые решения, а в результате выбора принимается одно наилучшее в некотором смысле решение. Результаты решения пополняют знания.

В зависимости от сферы использования информация может быть различной: научной, технической, управляющей, экономической и т.д. Для экономической информатики интерес представляет экономическая информация.

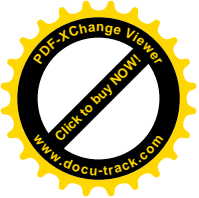
1.1.3. Экономическая информация и информационные технологии

Экономическая информация – это преобразованная и обработанная совокупность сведений, отражающая состояние и ход экономических процессов. Экономическая информация циркулирует в экономической системе и сопровождает процессы производства, распределения, обмена и потребления материальных благ и услуг. Экономическую информацию следует рассматривать как одну из разновидностей управленческой информации.

Экономическая информация может быть:

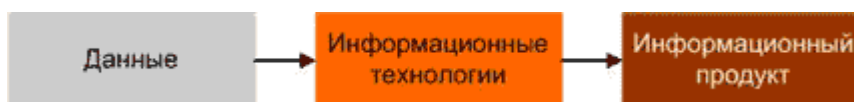
- *управляющая* (в форме прямых приказов, плановых заданий и т.д.);
- *осведомляющая* (в отчетных показателях, выполняет в экономической системе функцию обратной связи).

Информацию можно рассматривать как ресурс, аналогичный материальным, трудовым и денежным ресурсам. Информационные ресурсы – совокупность накопленной информации, зафиксированной на материальных носителях в любой форме, обеспечивающей ее передачу во времени и пространстве для решения научных, производственных, управленческих и других задач.



1.1.3.1 Информационные технологии

Сбор, хранение, обработка, передача информации в числовой форме осуществляется с помощью информационных технологий. Особенностью информационных технологий является то, что в них и предметом и продуктом труда является информация, а орудиями труда – средства вычислительной техники и связи. Основная цель информационных технологий - производство необходимой пользователю информации в результате целенаправленных действий по ее переработке.

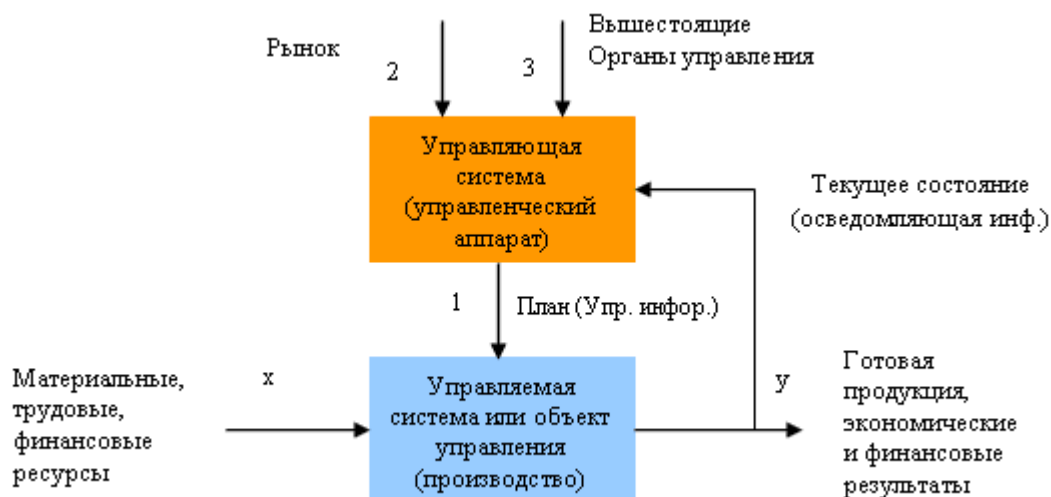


Известно, что информационная технология – это совокупность методов, производственных и программно-технологических средств, объединенных в технологическую цепочку, обеспечивающую сбор, хранение, обработку, вывод и распространение информации.

С точки зрения информационных технологий для информации необходим материальный носитель в качестве источника информации, передатчик, канал связи, приемник и получатель информации. Сообщение от источника к получателю передается через каналы связи или посредством среды.



Информация является формой связи между управляемыми и управляющими объектами в любой системе управления. В соответствии с общей теорией управления, процесс управления можно представить как взаимодействие двух систем - управляющей и управляемой.

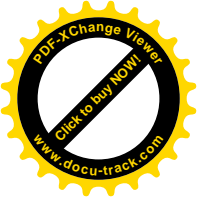


Система управления предприятием функционирует на базе информации о состоянии объекта, его входов X (материальные, трудовые, финансовые ресурсы) и выходов Y (готовая продукция, экономические и финансовые результаты) в соответствии с поставленной целью (обеспечить выпуск необходимой продукции).

Управление осуществляется путем подачи управленческого воздействия 1 (план выпуска продукции) с учетом обратной связи - текущего состояния управляемой системы (производства) и внешней среды (2, 3) - рынок, вышестоящие органы управления.

Назначение управляющей системы - формировать такие воздействия на управляемую систему, которые побуждали бы последнюю принять состояние, определяемое целью управления.

Применительно к промышленному предприятию с некоторой долей условности можно считать, что цель управления - это выполнение производственной программы в рамках технико-экономических ограничений; управляющие воздействия - это планы работ подразделений, обратная связь данные о ходе производства: выпуске и перемещении изделия, состоянии оборудования, запасах на складе и т.д.



Очевидно, что планы и содержание обратной связи - не что иное, как информация. Поэтому процессы формирования управляющих воздействий как раз и являются процессами преобразования экономической информации. Реализация этих процессов и составляет основное содержание управленческих служб, в том числе экономических. К экономической информации предъявляются следующие требования: точность, достоверность, оперативность.

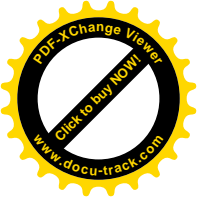
Точность информации обеспечивает ее однозначное восприятие всеми потребителями. Достоверность определяет допустимый уровень искажения как поступающей, так и результатной информации, при котором сохраняется эффективность функционирования системы. Оперативность отражает актуальность информации для необходимых расчетов и принятия решений в изменившихся условиях.

1.1.4. Информатика и информационные системы

Слово «система» происходит от греческого *systema*, что означает целое, составленное из частей или множества элементов. Система - это совокупность взаимосвязанных элементов, которые функционируют для достижения определенной цели.

Основные характеристики систем: цель, входы, выходы, обратная связь и внешняя среда. Системы значительно отличаются между собой как по составу, так и по главным целям. К системам относятся аппаратные и программные средства компьютеров, телекоммуникации, системы жизнеобеспечения, системы образования и т.д.

К экономическим системам относятся: промышленные предприятия, торговые организации, коммерческие банки, государственные учреждения и т.д.



Итак, объектом экономической информатики выступают экономические информационные системы, конечная цель функционирования которых - эффективное управление экономической системой. Таким образом, основное назначение информационной системы – создание современной инфраструктуры для управления предприятием, организацией, учреждением.

Разнообразие задач, решаемых с помощью ИС, привело к появлению множества разнотипных систем, отличающихся принципами построения и заложенными в них правилами обработки информации. Информационные системы можно **классифицировать** по целому ряду различных признаков.

Классификация информационных систем по признаку структурированности задач.

Различают три типа задач, для которых создаются информационные системы:

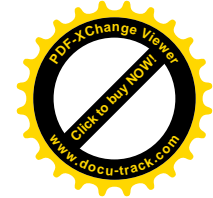
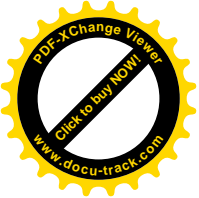
- структурированные (формализуемые);
- неструктурированные (неформализуемые);
- частично структурированные.

Структурированная (формализуемая) задача - задача, где известны все ее элементы и взаимосвязи между ними. Неструктурированная (неформализуемая) задача - задача, в которой невозможно выделить элементы и установить между ними связи.

Информационные системы для частично структурированных задач. Информационные системы, используемые для решения частично структурированных задач, подразделяются на два вида создающие управленческие отчеты и ориентированные главным образом на обработку данных; разрабатывающие возможные альтернативы решения.

Классификация рынка информационных систем по масштабности системы:

- Локальные системы (1С, БЕСТ, Инфо - Бухгалтер и т.д.)
- Малые интегрированные системы (Skala, Парус, Галактика и другие)



- Средние интегрированные системы (MFG-PRO и другие)
- Крупные интегрированные системы (SAP/R3 другие)

Классификация систем, которая основывается на классификации бизнес-задач.

Принципы классификации управленческих информационных систем:

1. Уровень стратегического управления (3 – 5 лет)
2. Уровень среднесрочного управления (1 – 1,5 года)
3. Уровень операционного управления (месяц – квартал - полугодие)
4. Уровень оперативного управления (день - неделя)
5. Уровень управления реального времени

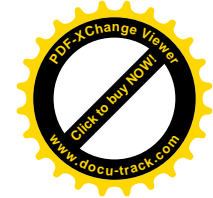
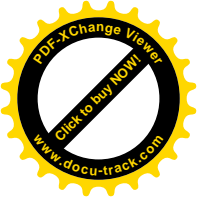
Существуют и другие типы классификации информационных систем. За рубежом были разработаны специальные программы Стандарты информационных систем управления предприятиями системы MRP, MRP-II, ERP, ERP II.

MRP – это системы планирования потребностей в материальных ресурсах (обеспечивает необходимый объем остатков материалов на складе). MRP-II – предназначены для планирования производственных ресурсов, т.е. ресурсов, используемых для производства продукции.

ERP – предназначена для планирования и управления материальными, производственными и людскими ресурсами. SAP R/3 - это ERP система (Enterprise Resource Planning) управления ресурсами предприятия или SAP ER. ERP II – предназначена для управление ресурсами и внешними связями предприятий.

Информационные системы, применяемые для планирования и управления различными ресурсами, называются интегрированными системами управления или корпоративными информационными системами.

К основным компонентам информационных систем, используемых в экономике, относятся: программно-аппаратные средства, бизнес-приложения и управление информационными системами.



1. Программно-аппаратные средства информационных систем:

- технические средства обработки информации (компьютеры и периферийные устройства);
- системное и сервисное программное обеспечение (операционные системы и утилиты)
- Прикладное программное обеспечение офисного назначения (MS Office);
- компьютерные сети (коммуникационное оборудование, сетевое ПО и сетевые приложения);
- базы и банки данных.

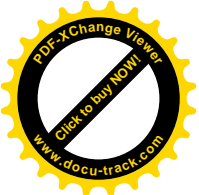
2. Бизнес-приложения (прикладные программы):

- локальные информационные системы (1С: Бухгалтерия, Инфин, Парус и т.д.);
- малые информационные системы (1С: Предприятие, Парус, Галактика и т.д.);
- средние информационные системы (PEOPLE SOFT, BAAN, SCALA и т.д.);
- интегрированные системы управления (ERP).

3. Управление информационными системами предназначено для управления и поддержки информационных процессов предприятия (управление персоналом, развитием, качеством, безопасностью, оперативное управление и т.д.)

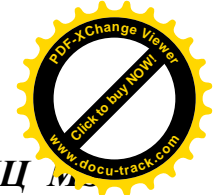
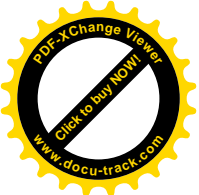
Таким образом, **информационные системы**, которые рассматриваются в экономической информатике, **состоят из трех основных компонентов:**

- **информационные технологии** (аппаратные и программные средства компьютеров, телекоммуникации, данные);



- **функциональные подсистемы** (производство, бухгалтерия, финансы, сбыт, маркетинг, кадры) и бизнес приложения (прикладные программы для решения бизнес задач);
- **управление информационными системами** (персонал, пользователи, развитие ИС, финансы).

В настоящее время наиболее целесообразным путем построения экономической информационной системы является применения готовых решений, которые реализованы в виде готовых прикладных программ.



ТЕМА 2 РАБОТА В РЕДАКТОРЕ ЭЛЕКТРОННЫХ ТАБЛИЦ MS EXCEL

MS Excel относится к классу систем обработки числовой информации, называемых *spreadsheet*. Буквальный перевод термина “spreadsheet” с английского языка означает “расстеленный лист (бумаги)”. В компьютерном мире под этим термином подразумевают класс программных средств, именуемых у нас “электронными таблицами”.

Области применения электронных таблиц:

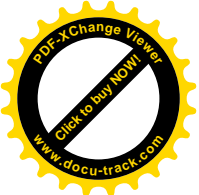
- бухгалтерский и банковский учет;
- планирование распределение ресурсов;
- проектно-сметные работы;
- инженерно-технические расчеты;
- обработка больших массивов информации;
- исследование динамических процессов.

Основные возможности электронных таблиц:

- анализ и моделирование на основе выполнения вычислений и обработки данных;
- оформление таблиц, отчетов;
- форматирование содержащихся в таблице данных;
- построение диаграмм требуемого вида;
- создание и ведение баз данных с возможностью выбора записей по заданному критерию и сортировки по любому параметру;
- перенесение (вставка) в таблицу информации из документов, созданных в других приложениях, работающих в среде Windows;
- печать итогового документа целиком или частично.

Преимущества использования ЭТ при решении задач:

- Решение задач с помощью электронных таблиц освобождает от составления алгоритма и отладки программы. Нужно только



определенным образом записать в таблицу исходные данные математические соотношения, входящие в модель.

- При использовании однотипных формул нет необходимости вводить их многократно, можно скопировать формулу в нужную ячейку. При этом произойдет автоматический пересчет относительных адресов, встречающихся в формуле. Если же необходимо, чтобы при копировании формулы ссылка на какую-то ячейку не изменилась, то существует возможность задания абсолютного (неизменяемого) адреса ячейки.

Структура электронной таблицы.

В таблице используются *столбцы* (256) и *строки* (16384).

Строки пронумерованы от 1 до 16384, столбцы помечаются латинскими буквами от A до Z, и комбинациями букв AA, AB,..., IV,

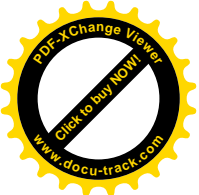
Элемент, находящийся на пересечении столбца и строки называется - *ячейкой* (клеткой).

Прямоугольная область таблицы называется *диапазоном* (интервалом, блоком) ячеек. Она задается адресами верхней левой и правой нижней ячеек блока, перечисленными через двоеточие.

2.1. Модель ячейки в MS Excel

Каждая ячейка таблицы имеет следующие характеристики:

- адрес;
- содержимое;
- изображение;
- формат;
- имя;
- примечание (комментарий).



Адрес ячейки - номер столбца и строки. Используется в формулах в виде относительной, абсолютной или смешанной ссылки, а также для быстрого перемещения по таблице.

MS Excel позволяет использовать два стиля ссылок: стиль A1 и стиль R1C1. Переключение стилей осуществляется в меню: *Сервис/ Параметры/ Общие* опцией “*Стиль ссылок R1C1*”.

Например. Пусть в ячейке D3 нужно получить произведение чисел, находящихся в ячейках A2 (второй ряд, первая колонка) и B1 (первый ряд, вторая колонка). Это может быть записано одним из следующих способов:

Таблица 1.

Вид ссылок	Стиль A1	Стиль R1C1
относительный	=A2 * B1 имя столбца, имя строки, которые будут относительно изменяться, при копировании формулы в другую ячейку.	=R[-1]C[-3] * R[-2]C[-2] смещение по строке, смещение по столбцу, относительно ссылающейся ячейки. Сама формула при копировании не изменяет вид, но ссылается уже на другие ячейки.
абсолютный	=\$A\$2 * \$B\$1 имя столбца, имя строки, которые останутся неизменным, при копировании формулы .	=R2C1 * R1C2 номер строки и номер столбца, относительно начала координат.
смешанный	=A2 * B\$1 =A\$2 * \$B1	=R[-1]C1 * R1C[-2] =R2C[-3] * R[-2]C2

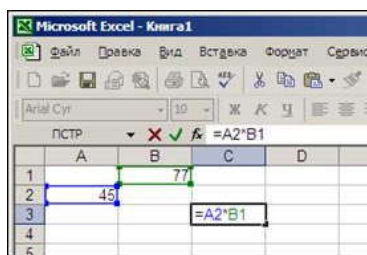


Рис.1а. Стиль ссылок A1
(имя столбца, имя строки).

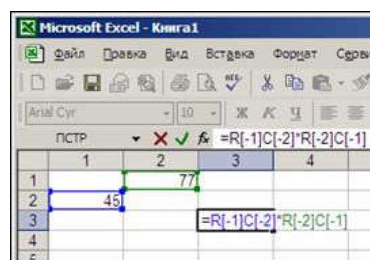


Рис.1b. Стиль ссылок R1C1
(смещение по строке,
смещение по столбцу).

Содержимым ячейки может быть:

- число (целое со знаком или без (-345), дробное с фиксированной точкой (253,62) или с плавающей точкой (2,5362e+2));
- текст;
- формула.

Формула - всегда начинается со знака “=” и может содержать: числовые константы, абсолютные или относительные ссылки на адреса ячеек, встроенные функции.

Аргументы функций всегда заключаются в круглые скобки. Стандартные функции можно как ввести с клавиатуры, так и воспользоваться меню *Вставка/Функция* или соответствующей кнопкой на панели инструментов.

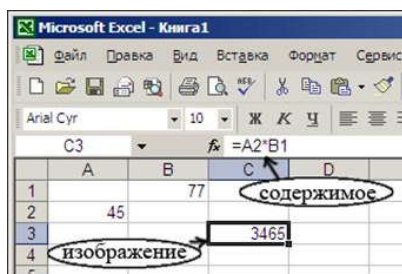
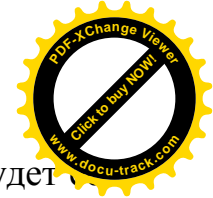
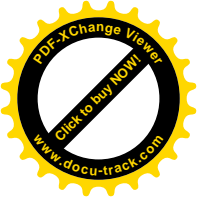


Рис. 2. Изображение не всегда совпадает с содержимым ячейки

Изображение - то, что пользователь видит на экране монитора.



Если содержимым ячейки является формула, то изображением будет значение.

Текст, помещенный в ячейку, может быть “виден” целиком, либо (если соседняя ячейка не пуста), из него видно столько символов, сколько позволяет ширина ячейки.

Изображение числа зависит от выбранного формата. Одно и то же число в разных форматах (дата, процент, денежный и т.д.) будет иметь различное изображение.

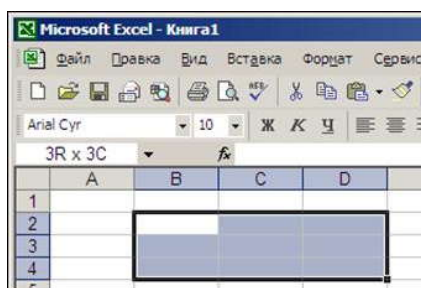
Формат ячейки - формат чисел, шрифт, цвет символов, вид рамки, цвет фона, выравнивание по границам ячейки, защита ячейки.

Имя - используется в формулах, как замена *абсолютного* адреса ячейки. Например, назначив ячейке C3 имя “Произведение” в ячейку D3 можно поместить формулу: =Произведение/3 (вместо формулы =C3/3). В этом случае, при копировании формулы, адрес ячейки меняться не будет.

Примечание - сопроводительный текст к содержимому ячейки. Ввести примечание в ячейку можно с помощью меню Вставка / Примечание. Ячейка, имеющая примечание, отмечается в рабочем листе точкой в правом верхнем углу.

2.2. Ввод и редактирование данных

Основными объектами, над которыми производятся действия в электронных таблицах, являются ячейки и диапазоны ячеек (блоки).



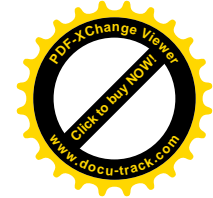
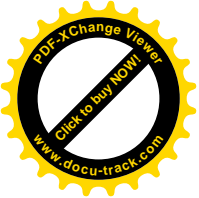


Рис. 3. Выделенный диапазон ячеек (блок).

Блок - любая прямоугольная область таблицы, в минимальном случае - одна ячейка. Адрес блока задается так: адрес верхней левой ячейки блока, двоеточие, адрес правой нижней ячейки блока.

Примеры блоков: A1 (ячейка); A1:A9 (столбец); B2:Z2 (строка); B2:D4 (прямоугольная область).

Неотъемлемым элементом рабочего поля таблицы является курсор. В ЭТ термин “курсор” используется в следующих случаях:

- курсор ЭТ - жирная рамка вокруг текущей ячейки, перемещается с помощью клавиш управления курсором;
- текстовый курсор - мигающая (или не мигающая) черточка, отмечающая положение текущего символа при редактировании содержимого ячейки.

2.2.1. Ввод данных

1. Установить курсор ЭТ в ячейку, в которой должны быть размещены данные.
2. Набрать данные.
3. Для завершения ввода нажать клавишу <Enter> (при этом курсор ЭТ переместится на строку ниже), либо нажать «зеленую галочку» на панели инструментов (при этом курсор останется в текущей ячейке).

В ячейке могут размещаться данные одного из следующих типов:

1. число
2. формула
3. текст

2.2.1.1 Правила ввода текста и чисел

Текст можно вводить произвольной формы, но если он начинается с знака “=”, то перед ним следует поставить апостроф, чтобы он не воспринимался как формула.

Числа также вводятся в привычном виде. Следует только помнить, что дробные десятичные числа записываются через запятую: 3,5; -0,0045, либо через точку: 3.5; -0.0045, в зависимости от установленных параметров. Изменение вида разделителя целой и дробной части производится в меню *Сервис/ Параметры/ Международные*.

2.2.1.2 Форматирование текстовых полей

По умолчанию текстовые поля в MS Excel выводятся в одну строку. Для того чтобы текст переносился в ячейке в несколько строк:

1. Выделите ячейки, для которых необходимо разрешить перенос текста.
2. Выберите пункт меню *Формат/ Ячейки* вкладка *Выравнивание*.
3. Поставьте галочку в опции *Переносить по словам*.

Для таблиц со сложной структурой используйте объединение ячеек, но только там, где это действительно требуется.

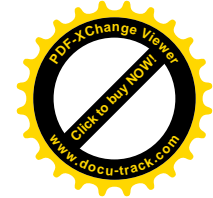
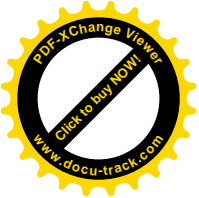
Не правильно

	A	B	C	D	E	F
1				январь		
2	Наименование		Цена, руб.	Количество	Стоимость, руб.	
3	Товар первый		20	2	40	
4	Товар второй		50	40	2000	
5	Товар третий		72	13	936	

Правильно

	A	B	C	D
1				январь
2				
	Наименование	Цена, руб.	Количество	Стоимость, руб.
3	Товар первый	20	2	40
4	Товар второй	50	40	2000
5	Товар третий	72	13	936

Рис. 4. Неправильно и правильно отформатированные текстовые поля.



2.2.1.3 Правила ввода формул

1. Убедитесь в том, что активна (выделена курсивной рамкой) та ячейка, в которой вы хотите получить результат вычислений.

2. Ввод формулы начинается со знака “=”. Этот знак вводится с клавиатуры.

3. После ввода знака “=” Excel переходит в режим ввода формулы. В этом режиме, при выделении какой-либо ячейки, ее адрес автоматически заносится в формулу. Это позволяет избавить пользователя от необходимости знать адреса ячеек и вводить их в формулу с клавиатуры.

4. Находясь в режиме ввода формулы, вы последовательно указываете левой кнопкой мыши на ячейки, хранящие некие числовые значения, и вводите с клавиатуры знаки операций между исходными значениями.

- Знаки операций должны вводиться между адресами ячеек.
- Удобнее вводить знаки операций с правого цифрового блока клавиатуры. Чтобы этот блок работал в нужном режиме, индикатор <Num Lock> должен быть включен.

5. Чтобы результат вычислений появился в активной ячейке, необходимо выйти из режима ввода формулы.

- <Enter> завершает ввод формулы, и переводит курсор в следующую ячейку.
- “Зеленая галочка” на панели ввода формулы завершает ввод формулы, и оставляют курсор в той же ячейке.

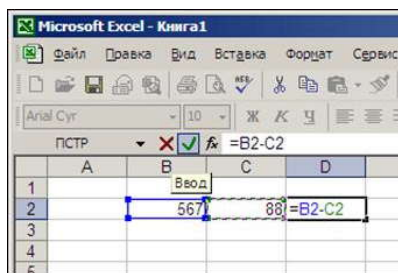
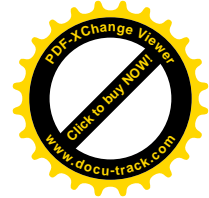
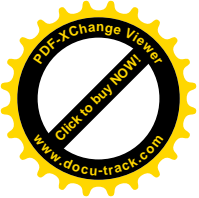


Рис. 5. Ввод формулы.



Например, если в ячейке D2 должна помещаться разность чисел из ячеек B2 и C2, то после установки курсора на D5 следует указать мышью на B2, ввести с клавиатуры знак “-”, указать мышью на C2 и нажать <Enter> или “зеленую галочку”.

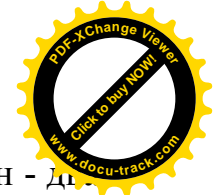
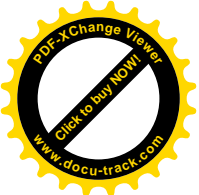
В формулах можно использовать числовые константы (-4,5), ссылки на блоки (D4), (A3:D8), знаки арифметических операций, встроенные функции (СУММ, МАКС, SIN и т.д.)

Таблица 2.

Операция	Обозначение формуле	Пример
Возведение в степень	^	=3^2
Умножение	*	=A8*C6
Деление	/	=D4/N5
Сложение	+	=B2+5
Вычитание	-	=9-G6
Равно	=	
Меньше	<	
Больше	>	
Меньше или равно	<=	
Больше или равно	>=	
Не равно	<>	
Диапазон	:	=СУММ(A1:C10)
Объединение диапазонов	;	=СУММ(A1;A2;A6)
Максимум	МАКС	=МАКС(A3:C5)
Минимум	МИН	=МИН(E2:P7)

2.2.2. Редактирование данных

При вводе данных Вы можете ошибиться и должны уметь исправлять ошибки. Конечно, Вы можете просто ввести в ячейку с ошибочными



данными новое правильное значение, но если исправить требуется один - два символа, то целесообразнее *отредактировать* содержимое ячейки.

Отредактировать данные Вы можете различными способами, но *курсор ЭТ должен стоять на редактируемой ячейке*.

1. Перейдите в режим редактирования содержимого ячейки. Это можно сделать одним из следующих способов:

- Щелкните левой клавишей мыши в строке формул.
- Нажмите <F2>.
- Дважды щелкните мышью на ячейке.

2. Текстовый курсор поставьте перед неверным символом, исправьте данные.

3. Нажмите <Enter> или “зеленую галочку” на панели инструментов, чтобы выйти из режима редактирования.

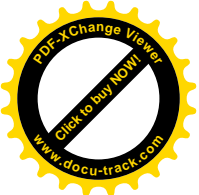
Неверный формат ячейки может быть изменен только выбором другого формата в меню *Формат / Ячейка*.

Если ошибка допущена *при вводе числа*, то так как компьютер не знает, что это ошибка, Excel автоматически пытается подобрать подходящий для данного изображения формат.

Например, Вы хотели ввести число 410. Интерпретация некоторых возможных при вводе ошибок приводится в таблице 4.

Таблица 3.

Ошибочный ввод	Распознанный формат	Изображение на экране
4:10	время	текст 4:10
4.10	дата	4.Окт.
410.	текст	410.



Не пытайтесь исправить ошибку непосредственно в ячейке, вряд ли это удастся, так как скрытый формат этой ячейки уже сформирован. Поэтому нужно исправлять сначала формат ячейки на правильный с помощью меню *Формат / Ячейка / Число*.

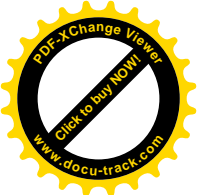
Если при вводе формул Вы забыли поставить знак “=”, то все, что было набрано, запишется в ячейку как текст. Если Вы поставили знак равенства, то компьютер распознал, что идет ввод формулы и не допустит записать формулу с ошибкой до тех пор, пока она не будет исправлена. Некоторые ошибки, которые могут быть допущены при вводе формулы, приведены в таблице 20.

Таблица 4.

Изображение на экране	Тип ошибки
b1*b2	нет знака =
#ИМЯ?	адрес ячейки введен с клавиатуры в режиме кириллицы
открывается диалоговое окно	введено два знака операции подряд
#ЗНАЧ!	в одной из ячеек, входящих в формулу, находится не числовое значение
#ДЕЛ/0!	значение знаменателя =0 , делить на 0 нельзя

2.2.3. Копирование формул

В электронных таблицах часто требуется проводить операции не просто над двумя *переменными* (ячейками), но и над *массивами* (столбцами






или строками) ячеек. Т.е. все формулы результирующего массива аналогичны и отличаются друг от друга только адресом строк или столбцов.

От проведения однотипных действий в каждой ячейки строки (или столбца) избавляет следующий прием копирования формулы:

1. Убедитесь, что активна (выделена курсорной рамкой) именно та ячейка, в которой находится предназначенная для копирования формула.
2. Не нажимая на кнопки мыши, подведите указатель мыши к нижнему правому углу курсорной рамки (этот угол специально выделен).
3. Отыщите положение, при котором указатель мыши превращается в тонкий черный крестик.
4. Нажмите на левую кнопку мыши и, удерживая ее, выделяйте диапазон ниже (при копировании по строкам) или правее (при копировании по столбцам) до тех пор, пока не выделятся все ячейки, в которые вы хотите скопировать данную формулу.
5. Отпустите левую кнопку мыши.

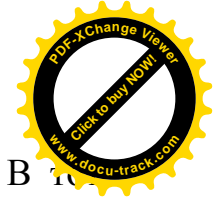
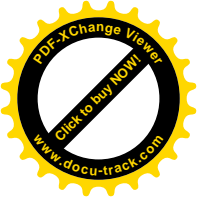
Таблица 5.

Типы указателей мыши

	Копирование содержимого выделенной ячейки, или блока ячеек.
	Перенос блока ячеек с одного места рабочего листа на другое. При нажатой клавише <Ctrl> произойдет не перенос, а копирование.
	Выделение блока ячеек.

2.2.4. Относительная и абсолютная адресация

Одно из преимуществ электронных таблиц в том, что в формулах можно использовать не только конкретные числовые значения (*константы*),



но *переменные* - ссылки на другие ячейки таблицы (адреса ячеек). В момент, когда Вы нажимаете клавишу <Enter>, в формулу вместо адреса ячейки подставляется число, находящееся в данный момент в указанной ячейке.

Другое достоинство в том, что *при копировании формул входящие в них ссылки изменяются (относительная адресация)*.

Однако иногда при решении задач требуется, чтобы *при копировании формулы ссылка на какую-либо ячейку не изменялась*. Для этого используется *абсолютная адресация*, или абсолютные ссылки.

При копировании приведенным выше способом адреса ячеек в формуле изменялись *относительно*.

Если необходимо, чтобы при копировании или перемещении данных адрес какой-либо ячейки в формуле не мог изменяться (например, при умножении всего столбца данных на значение одной и той же ячейки), нужно зафиксировать положение этой ячейки в формуле до того, как вы будете копировать или перемещать данные.

Для фиксации адреса ячейки используется знак “\$”.

Координата строки и координата столбца в адресе ячейки могут фиксироваться отдельно.

Чтобы относительный адрес ячейки в формуле стал *абсолютным*, после ввода в формулу адреса этой ячейки нажмите <F4>.

Таблица 6.

Адресация		По строке	
		Относительная	Абсолютная
По столбцу	Относительная	B1	B\$1
	Абсолютная	\$B1	\$B\$1

Изменение адреса ячейки в формуле, при ее копировании.

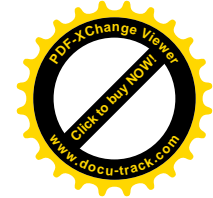
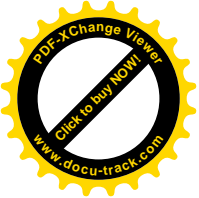


Таблица 7.

Значение адреса в исходной формуле	Вниз	Вправо
B1	B2	C1
\$B1	\$B2	\$B1
B\$1	B\$1	C\$1
\$B\$1	\$B\$1	\$B\$1

2.3. Построение диаграмм

Одним из самых впечатляющих достоинств MS Excel является способность превращать абстрактные ряды и столбцы чисел в привлекательные, информативные *графики* и *диаграммы*. Excel поддерживает 14 типов различных стандартных двух- и трехмерных диаграмм. При создании новой диаграммы по умолчанию в Excel установлена гистограмма.

Диаграммы - это удобное средство графического представления данных. Они позволяют оценить имеющиеся величины лучше, чем самое внимательное изучение каждой ячейки рабочего листа. Диаграмма может помочь обнаружить ошибку в данных.

Для того чтобы можно было построить диаграмму, необходимо иметь, по крайней мере, один ряд данных. Источником данных для диаграммы выступает таблица Excel.

Специальные термины, применяемые при построении диаграмм:

- Ось X называется осью *категорий* и значения, откладываемые на этой оси, называются категориями.
- Значения отображаемых в диаграмме функций и гистограмм составляют *ряды данных*. Ряд данных – последовательность числовых

значений. При построении диаграммы могут использоваться несколько рядов данных. Все ряды должны иметь одну и ту же размерность.

- *Легенда* – расшифровка обозначений рядов данных на диаграмме.

Тип диаграммы влияет на ее структуру и предъявляет определенные требования к рядам данных. Так, для построения круговой диаграммы всегда используется только один ряд данных.

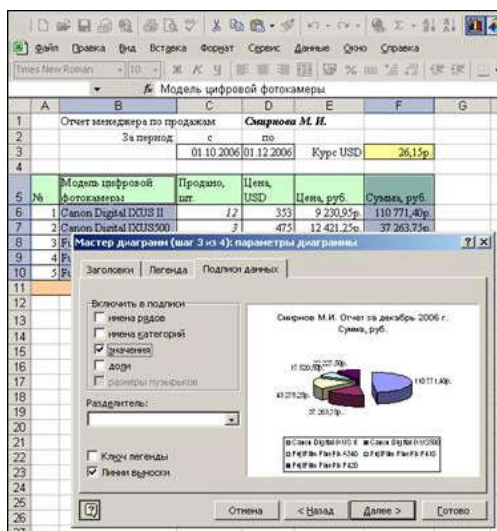
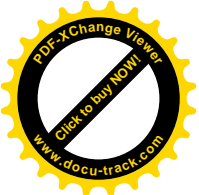


Рис. 21. Создание диаграммы.

Последовательность действий, при построении диаграммы

1. Выделите в таблице диапазон данных, по которым будет строиться диаграмма, включая, если это возможно, и диапазоны подписей к этим данным по строкам и столбцам.
2. Для того чтобы выделить несколько несмежных диапазонов данных, производите выделение, удерживая клавишу <Ctrl>.
3. Вызовите мастера построения диаграмм (пункт меню *Вставка/Диаграмма* или кнопка на стандартной панели инструментов).
4. Внимательно читая все закладки диалогового окна мастера построения диаграмм на каждом шаге, дойдите до конца (выбирайте “Далее”, если эта кнопка активна) и в итоге нажмите “Готово”.

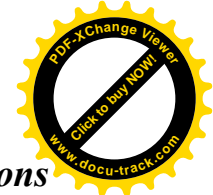
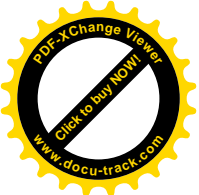
После построения диаграммы можно изменить:



- размеры диаграммы, потянув за габаритные обозначения, которые появляются тогда, когда диаграмма выделена;
- положение диаграммы на листе, путем перетаскивания объекта диаграммы мышью;
- шрифт, цвет, положение любого элемента диаграммы, дважды щелкнув по этому элементу левой кнопкой мыши;
- тип диаграммы, исходные данные, параметры диаграммы, выбрав соответствующие пункты из контекстного меню (правая кнопка мыши).

Диаграмму можно удалить: выделить и нажать <Delete>.

Диаграмму, как текст и любые другие объекты в MS Office, можно копировать в буфер обмена и вставлять в любой другой документ.



ТЕМА 3 Основы программирования на Visual Basic for Applications

3.1 Основы программирования на VBA

Visual Basic For Application (VBA – Visual Basic для приложений) реализован в четырех важнейших компонентах пакета Microsoft Office 97/2000. VBA представляет собой полностью унифицированный язык программирования и почти полностью унифицированную среду разработки.

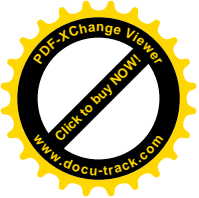
VBA включает в себя основные конструкции языка Visual Basic для Windows, встроен в MS Office и приложения Office. VBA является общим для всех приложений MS Office. В VBA применяется объектно - ориентированный подход к разработке приложений. Каждое приложение – Word, Excel, Access, PowerPoint и др.- дополняет средства VBA новыми функциональными возможностями с помощью объектов, свойств и методов, присущих этому приложению. Как объектно – ориентированный язык программирования VBA использует следующие основные понятия: объект, свойство, метод, событие, класс и семейство объектов. Введем определения.

Объект – это объединение данных с кодом, предназначенным для их обработки. Объекты применяются в программировании, потому что они представляют предметы, о внутренней организации которых и о том, как они сделаны, программисту не обязательно даже знать, но он может ими пользоваться и управлять.

Свойства – это параметры объекта, которые ему присущи. Они характеризуют объект. Все свойства объекта имеют свои имена. Различные объекты характеризуются различными свойствами, например, объект «линия» характеризуется длиной и цветом, а объект «окружность» радиусом и цветом. Синтаксис задания свойства объекту имеет вид

ОБЪЕКТ. СВОЙСТВО = ЗначениеСвойства.

Например,



.....
'изменить имя текущего листа

ActiveSheet . Name = "Январь"

.....
Метод – это программа, которая воздействует на объекты и их параметры. С помощью методов можно, например, закрыть объект, удалить его, или показать на экране. Синтаксис записи такой:

ОБЪЕКТ.МЕТОД

Например,

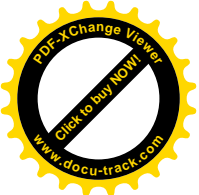
.....
'удалить лист

WorkSheets("Обзор") . Delete

.....
Событие – это действие, распознаваемое объектом, для которого можно запрограммировать отклик (например, щелчок мышью по строке меню или нажатие клавиши). При наступлении события происходит активизация метода, т.е. начинают выполняться связанные с объектом программы и функции.

Класс – это шаблон, на основе которого во время выполнения программы создается объект. Класс определяет свойства и методы объекта. Если взять пример из жизни, то в качестве класса объектов можно принять школьный класс, где объекты – это учащиеся, каждый со своими , присущими только ему свойствами. Также каждому объекту в классе присущи одно или несколько одинаковых свойств, например – возраст, по которому и определяется принадлежность объекта к классу. Объект, который создается на основе класса, называют *экземпляром* класса.

Итак, все приложения Microsoft Office имеют макроязык и средство записи макросов. Все макрокоманды VBA включаются в проекты, которые



могут принадлежать различным документам. При создании проекта используется один из трех методов:

- запись макроса при помощи макрорекодера, с последующим его редактированием,
- создание макроса в редакторе VBA,
- создание проекта в редакторе VBA.

3.2 Код VBA

Код VBA хранится в проектах. Проекты содержат модули, а модули включают процедуры и функции. Процедуры и функции содержат описания и инструкции на языке VBA.

Каждое приложение MS Office хранит проекты по своему:

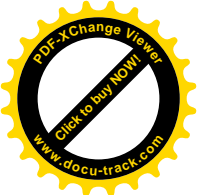
- Word сохраняет проекты в шаблонах документа (DOT) и в документах (DOC).
- Excel сохраняет проекты в рабочих книгах (XLS).
- Access сохраняет проекты в файлах базы данных (MDB).
- PowerPoint сохраняет проекты в презентациях (PPT).

Для программирования на VBA нет необходимости в обязательном порядке устанавливать какую-либо версию транслятора VB (Visual Basic). Вполне достаточно, если на компьютере будет установлен Microsoft Office 97 и выше. Для создания программы, которая называется макросом, необходимо загрузить Microsoft Word или Microsoft Excel. Далее: *-Сервис-Макрос-Редактор Visual Basic*. В окне проекта первой строкой записывается оператор объявления процедуры (программы)

```
Sub имя()
```

Это стандартный заголовок макроса. Завершающей строкой процедуры является строка

```
End Sub
```



Для запуска программы из редактора - клавиша F5. Для запуска программы из документа — alt+F8.

3.3 Элементы языка VBA

При написании кода VBA следует знать некоторые правила или соглашения, которых следует придерживаться, чтобы избежать ошибок выполнения кода. Первое соглашение — это соглашение по именам. Именовать нам придется переменные, константы, процедуры, функции, объекты и т.д., поэтому определим правила, которым будем следовать:

- Длина имен не должна превышать 255 символов.
- Имена должны начинаться с буквы и могут содержать комбинацию букв, цифр и символов.
- Имена не должны совпадать со стандартными именами Visual Basic.
- Имена не должны включать точек, пробелов и следующих символов: ! @ # \$ %. Символ пробела можно заменить символом подчеркивания _.

Регистр букв в именах не имеет значения, т.е. следующие имена PRICE, Price и price — эквивалентны.

Примеры правильной записи имен:

H123, Price, Sale

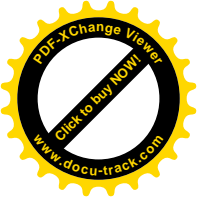
Неправильно записанные имена:

21days начинается с цифры;

Price Sale содержит символ пробела;

Dim совпадает со служебным словом языка.

Дадим также несколько определений элементам VBA.



Ключевые слова — это слово или символ, которые являются элементами языка программирования Visual Basic, это может быть инструкция, имя функции, оператор. Например: SUB, DIM, CONST.

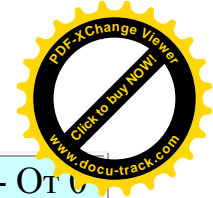
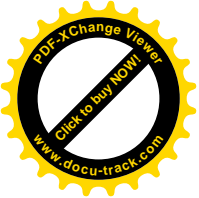
Выражение — это комбинация ключевых слов, операций, констант. Результатом выражения может быть строка, логическое значение или число. Примеры выражений: $X+3$; $Z \leq 0$; 11.

Операторы используются для объединения простых выражений в более сложные. Операторы VBA выполняют арифметические, логические операции, а также операции сравнения и конкатенации (сцепления строк).

3.4 Типы данных языка VBA

Visual Basic имеет 11 встроенных типов данных:

Тип данных	Размер в байтах	Разрядность цифр		Диапазон значений
Boolean	2	1	логический	True или False (да - нет, включен - отключен)
Integer	2	5	целое	От (-32768) до 32768
Long	4	10	длинное целое	От (-2147483648) до 2147483647
Single	4	7	действительное одинарной точности	От (-3.402823E+38) до (-1.401298E-45) и от 1.401298E-45 до 3.402823E+38
Double	8	15	действительное двойной точности	От $\pm 1.79769313486232E+308$ до $\pm 4.94065645841247E-324$
Currency	8	19	денежный	От (-922337203685477.5808) до 922337203685477.5807
Date	8	-	для дат и времени	От 01.01.100 до 31.12.9999



String	1	+1	символьный	на любой символ - От 0 до 65535 символов
Object	4	-		Любой объект
Array	Определяется количеством и размером элементов	-	массивы	-
Variant	Определяется записанными данными	-		Любой из встроенных типов данных

Тип данных определяет, каким образом задаваемая величина хранится в памяти компьютера; сколько разрядов может содержать присваиваемое ей значение; насколько оно может быть велико и может ли иметь дробную часть.

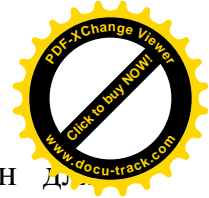
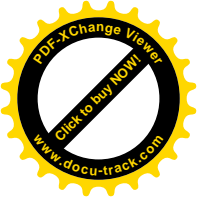
Boolean использует для хранения значения 2 байта памяти – как и тип **Integer**, хотя может принимать значения либо 0 (ложь - False), либо (True – правда). Для хранения этой информации достаточно 1 бита, а занимает 16 бит.

Integer предназначен для хранения смещенного целого числа, т.е. с указанием знака числа “+” или “-”. Из 16 бит 1 используется для хранения знака, а остальные – для значащих цифр.

Long используется для хранения смещенного целого числа с двойным разрешением по сравнению с **Integer** и, соответственно, занимает в два раза больше памяти. Из 32, занимаемых в памяти бит, 1 предназначается для хранения знака, 31 – для хранения самого числа.

Double – предназначен для хранения действительного числа с плавающей точкой двойной точности по отношению к **Single**. Для хранения требует вдвое больше памяти, но и диапазон значений расширяется в 8 раз.

Currency предназначен для хранения действительного числа с фиксированной точкой, которая всегда располагается между 4 и 5 цифрами справа.



ПРИМЕР: 0.0021 46.7280 Этот тип специально предназначен для получения минимальных ошибок округления и чаще всего используется в бухгалтерских расчетах.

Date предназначен для хранения дат и времени с точностью до секунды.

Машиной даты записываются следующим образом: целая часть – количество дней, прошедших от начала отсчета 01.01.1900, а дробная часть – время.

Пример: 05.03.1994 18.00 запишется как 34398.75. При работе с Excel следует помнить, что в нем диапазон используемых дат – от 01.01.1900 до 31.12.2078. Сам VBA позволяет оперировать датами в диапазоне 01.01.100 – 31.12.9999

String используется для хранения текста по одному байту на символ, добавляя к этому количеству еще один – для обозначения конца строки.

Object предназначен для ссылок на любой объект VBA

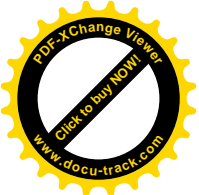
Array определяет индексный список величин одного и других типов - массив.

Variant используется по умолчанию. Он задается VBA, если вы сами не укажете этот тип. При этом производится проверка значения, вводимого в память и его идентификация, поэтому резко возрастает и время выполнения программы, и объем памяти.

Переменные

Переменная — это область памяти, предназначенная для временного хранения данных, и которой присвоено имя. Содержимое переменной может быть изменено в процессе выполнения программы. Данные, хранимые в переменных, могут быть различного типа: числовые данные, строковые, логические и т.д. Тип данных определяет тип переменной.

Область определения переменной определяет, где можно использовать переменную. В VBA есть три уровня определения переменных.



Уровень процедуры. Переменные используются только в процедурах, которой они описаны. Переменная описывается с помощью инструкции DIM, размещенной в теле процедуры.

Например:

Dim x As Single ‘переменная x описана как действительная одинарной точности

Dim I As Integer ‘переменная I описан как целая

Уровень модуля. Переменные используются только в процедурах модуля, в которых они описаны, но не в других модулях проекта. Переменная описывается с помощью инструкции DIM, размещенной в разделе описаний модуля.

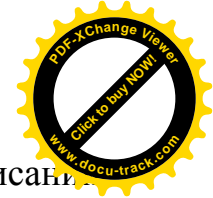
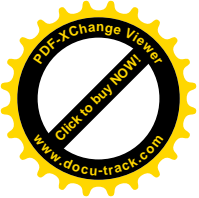
Общие переменные. Переменные используются во всех модулях и процедурах проекта. Переменная описывается с помощью инструкции PUBLIC, размещенной в разделе описаний модуля.

Переменные уровня процедуры сохраняют свое значение лишь при выполнении процедуры, при выходе из процедуры значение переменной теряется. Если процедура вызывает другие процедуры, переменная сохраняет свое значение, пока не закончится выполнение и этих процедур. Если переменная уровня процедуры описана с помощью ключевого слова Static, она сохраняет свое значение до тех пор, пока программа выполняется в каком-либо модуле. По завершении работы всей программы переменная теряет свою область определения и свое значение. Ее время жизни совпадает с временем жизни переменной уровня модуля.

Если переменная не описана, то ей присваивается тип Variant при вводе имени переменной.

Константы

Константа — это именованная область памяти для хранения данных, изменение которых во время работы программы не допускается. Константа может иметь числовой, логический или строковый тип.



Константа описывается ключевым словом CONST, и при описании константы требуется присвоить ей значение. Попытка изменить в ходе выполнения программы значение константы вызывает ошибку.

Примеры описания констант:

Const Pi as Double = 3.1415926 ‘описана константа Pi

Const Name = "Visual Basic" ‘описана константа строкового типа.

Операции

Операции производятся над выражениями, переменными, константами, литералами.

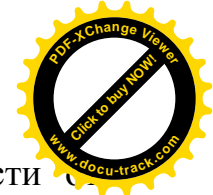
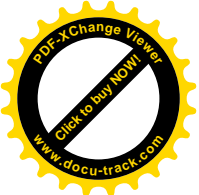
Арифметические операции

В качестве арифметических операций в языке Visual Basic применяются операции, описанные в таблице в порядке убывания приоритета.

Математические и строковые операции VBA в порядке убывания приоритета Операция	Выполняемое действие
^	Возведение в степень
-	Отрицание
*	Умножение
/	Деление 1
\	Целочисленное деление
Mod	Определение остатка от деления
+	Сложение
-	Вычитание
&	Объединение (конкатенация) строк

Правила записи арифметических выражений в языке Бейсик.

а) Все арифметические операции записываются в одну строку. Если в числителе или знаменателе дроби стоит многочлен, его следует заключить в скобки.



b) При записи десятичных дробей для отделения целой части дробной используется десятичная точка.

c) Знаки арифметических операций опускать нельзя.

d) Если в десятичной дроби нет целой части, то ноль перед десятичной точкой можно опустить.

Арифметические операции выполняются слева направо в порядке приоритета. Для изменения естественного приоритета операций применяются скобки (так же, как в арифметике). В Бейсике существуют стандартные функции, при применении которых не требуется никаких дополнительных библиотек или определений.

Логические операции

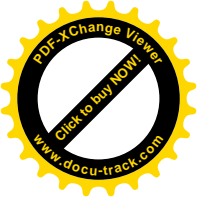
Логические выражения в математической логике оперируют с логическими переменными. Логические переменные имеют в современных языках программирования тип Boolean, что означает - булева(я) или логическая. Логическая переменная в процессе выполнения программы может принимать только два значения — "True" или "False". Определить логическую переменную можно, присвоив ей значение логического выражения.

Логическое выражение — выражение, в котором присутствуют операции отношения или знаки логических операций.

Операции отношения:

- = (равно)
- <> (не равно)
- < (меньше)
- > (больше)
- <= (меньше или равно)
- >= (больше или равно)

Например: $a > b$ и $b <= a + 2$.



Логические выражения могут быть связаны знаками *логически*
операций:

OR - Логическое сложение (ИЛИ),
AND - Логическое умножение (И),
NOT - Логическое отрицание (НЕ),
XOR - Исключающее ИЛИ,
EQV - Логическая эквивалентность,
IMP - Логическая импликация .

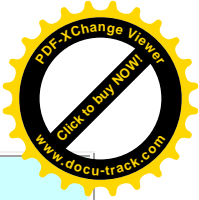
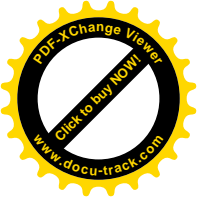
Операция конкатенации

Операция используется для сцепления двух или нескольких строк, обозначается знаком «+». Результатом будет строка, состоящая из строк – операндов.

Например,
операнд1="язык"
операнд2="Visual Basic"
результатом операции
"язык" + "Visual Basic"
будет строка
"язык Visual Basic".

Математические функции VBA

Функция	Выполняемое действие
Atn(x)	Возвращение арктангенса угла в радианах $\arctg x$
Sin(x)	Возвращение синуса угла в радианах $\sin x$
Cos(x)	Возвращение косинуса угла в радианах $\cos x$
Tan(x)	Возвращение тангенса угла в радианах $tg x$
Exp(x)	Возвращение значения e^x , где x - значение переменной или введенное число
Log(x)	Возвращение натурального логарифма числа $\ln x$



Sqr(x)	Возвращение квадратного корень числа
Abs(x)	Возвращение абсолютного значения числа (по модулю) x

Функции преобразования типов *Str()* и *Val()*

Str(*число*)

Возвращает значение типа Variant (String), являющееся строковым представлением числа. Обязательный аргумент *число* может задаваться любым допустимым числовым выражением.

При преобразовании числа в строку в начале строки обязательно резервируется позиция для знака числа. Если число является положительным, возвращенная строка будет содержать пробел на месте знака.

Примечание.

В качестве допустимого десятичного разделителя функция Str воспринимает только точку (.). При наличии другого десятичного разделителя (например, запятой) для преобразования чисел в строки следует использовать функцию CStr.

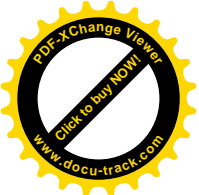
Val(*строка*)

Возвращает числа, содержащиеся в строке, как числовое значение соответствующего типа.

Обязательный аргумент *строка* является любым допустимым строковым выражением. Функция Val прекращает чтение строки на первом символе, который она не может распознать в качестве части числа. В качестве разделителя целой и дробной части распознается *только* точка (.). Пробелы, символы табуляции и символы перевода строк удаляются из значения аргумента.

В следующем примере возвращается числовое значение 1615198:

Val(" 1615 198-я ул. Н.Й.")



Организация ввода/вывода информации

Диалоговые функции ввода/вывода информации

При создании макрокоманд требуется организовать обмен информацией между макросом и пользователем, между макросом и документом. Эта задача решается с помощью функций InputBox, MsgBox.

Функция **InputBox**

Выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста пользователем или нажатия кнопки, а затем возвращает значение типа String, содержащее текст, введенный в поле.

InputBox (сообщение [, заголовок] [, текст по умолчанию])

сообщение — текст в диалоговом окне.

заголовок — строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот аргумент опущен, в строку заголовка помещается имя приложения.

текст по умолчанию — строковое выражение, отображаемое в поле ввода, если пользователь не введет другую строку. Если этот аргумент опущен, поле ввода изображается пустым.

Если пользователь нажимает кнопку ОК или клавишу ENTER, функция InputBox возвращает содержимое поля ввода.

Пример. Ввести значение X=0.5.

.....

X=Val(InputBox("Введите X"))

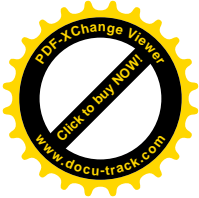
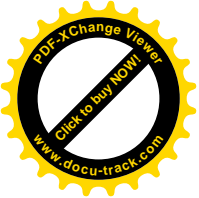
.....

В диалоговом окне ввести с клавиатуры 0.5.

Функция **MsgBox**

Функция **MsgBox** позволяет выводить результаты счета программы в диалоговом окне.

Синтаксис :



MsgBox (сообщение [, кнопки] [, заголовок])

сообщение — строковое выражение, отображаемое в диалоговом окне.

кнопки — числовое выражение, представляющее сумму значений, которые указывают число и тип отображаемых кнопок, тип используемого значка, основную кнопку. Значение по умолчанию этого аргумента равняется 0.

заголовок — строковое выражение, отображаемое в строке заголовка диалогового окна. Если этот аргумент опущен, в строку заголовка помещается имя приложения.

Будем использовать в качестве аргумента функции только *сообщение*, представляющее собой конкатенацию (сцепление)(+) двух строк: комментария к результату, взятого в кавычки, и строки, полученной из числового выражения с помощью функции Str.

Пример. Вывести результат на экран монитора.

```
.....  
MsgBox("Ответ=" + Str(y))  
.....
```

Использование окна отладки

Окно отладки можно вывести на экран при помощи команды *Вид - Окно отладки* или нажав клавиши *<Ctrl>+<G>* .

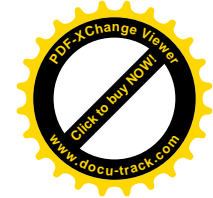
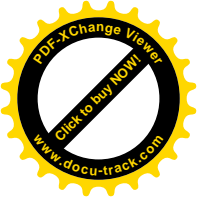
Для вывода текущих значений в окно Проверка непосредственно из макроса необходимо ввести в нужном месте кода макроса команду *Debug с* единственным методом *Print*. Синтаксис

Debug.Print выражение.

В месте, где установлена эта команда, произойдет прерывание выполнения макроса и в окне *Проверка* появится текущее значение переменной или выражения.

Пример.

```
.....
```



Debug.Print y

.....

Использование объектов Excel

Объект WorkSheet – это лист рабочей книги. Он входит в коллекцию всех листов рабочей книги - семейство Sheets. Этот объект можно использовать для ввода/вывода данных. В этом случае к этому объекту можно применить метод Select, позволяющий выбирать конкретный лист. Например,

Sheets (“Лист1”) . Select

Если макрос написан для определенного листа, то удобно исходные данные поместить в соответствующие ячейки данного листа. В этом случае метод Select можно не использовать. При этом результаты счета будут помещены на этот же листе автоматически..

Поскольку одиночная ячейка не представлена собственным объектом, то доступ к ячейке можно осуществить с помощью свойства Cells , указывая номер строки и столбца. Для присвоения некоторой переменной x значения, находящегося в ячейке A2 используем оператор:

X=Cells(2,1)

Пример.

Вывести значения переменных s и p во вторую и третью строки второй колонки (столбец B) третьего листа соответственно.

.....

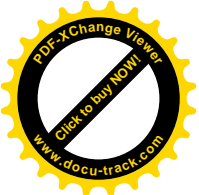
Sheets(“Лист3”).Select ‘переходим на третий лист

Cells(2,8)=s ‘выводим s

Cells(3,8)=p ‘выводим p

.....

Часто используется свойство Value, возвращающее содержимое ячейки. Например,



Cells(2,8).Value=s 'выводим s

3.5 Операторы языка VBA и реализация основных алгоритмических конструкций

В языке VBA используется множество инструкций или операторов. Они служат для выполнения каких-либо стандартных действий. Условно из них можно выделить группы.

Оператор комментариев не выполняет в программе никаких действий и может содержать любой текст. Инструкция комментариев имеет два вида: это инструкция, начинающаяся с ключевого слова REM, за которым следует произвольный текст, или знак апострофа ('), который ставится в начале строки или после любого оператора, и за ним также можно писать любой текст.

Например:

REM произвольный текст комментария

' произвольный текст комментария

A1=5 ' комментарий к строке оператора

Оператор присваивания

Присваивает выражение переменной или константе. Инструкции присвоения всегда включают знак равенства (=).

ПЕРЕМЕННАЯ = ВЫРАЖЕНИЕ

Вначале вычисляется выражение, а затем результат запоминается в переменной.

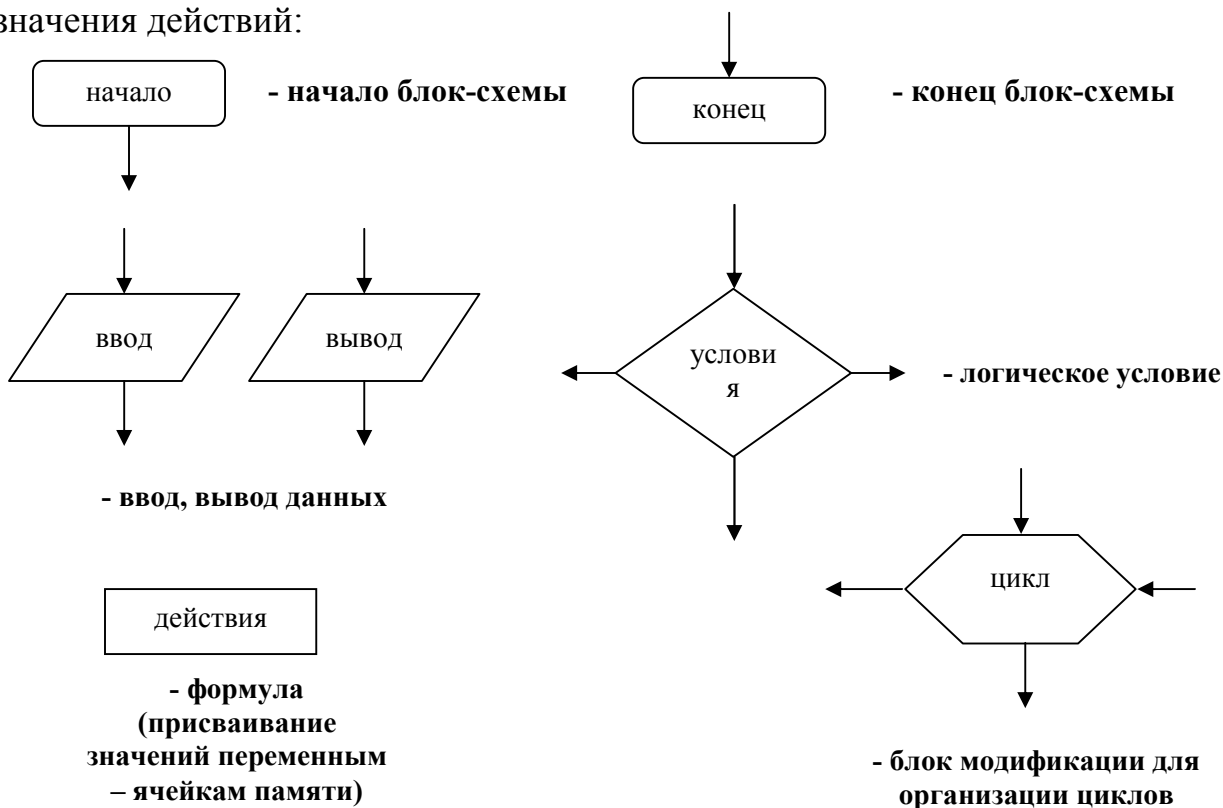
Выделяют следующие алгоритмические структуры:

1. линейная, когда все действия выполняются последовательно (без условий) от начала до конца;

2. разветвленная, когда на некотором шаге встречается условие, зависимости от которого процесс решения может пойти по одной из нескольких ветвей;

3. циклическая, когда некоторые фрагменты программы должны выполняться несколько раз.

ГОСТом установлены следующие геометрические фигуры для обозначения действий:

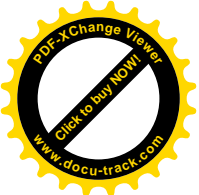


3.6 Программирование алгоритмов линейной структуры

Программа линейного вычислительного процесса представляет собой последовательность операторных строк, содержащих операторы присваивания, а также средства ввода и вывода.

При составлении линейных программ необходимо:

- выделить начальные данные;
- установить последовательность вычислений;
- определить исходные данные до момента их использования;



- желательно снабжать программу комментариями для удобства чтения.

Пример. Вычислить значение выражения для любых a и b .

Программа

```
Sub Lin()
```

```
Dim a As Double, x As Double
```

```
a = Val(InputBox("Введите a")) 'ввод значения переменной a
```

```
x = Val(InputBox("Введите x")) 'ввод значения переменной x
```

```
b = 1 / (x) ^ (1 / 4) 'вычисляем значение b'
```

```
c = Sin(a ^ 2 + b ^ 2) 'вычисляем значение C
```

```
MsgBox ("Ответ=" + Str(c))
```

```
End Sub
```

Пример. Составить программу для вычисления площади S треугольника ABC по заданным значениям сторон a, b, c .

Программа

```
Sub Geron()
```

```
Dim a, b, c, p, s As Double
```

```
a = Val(InputBox("Введите a")) 'ввод значения переменной a
```

```
b = Val(InputBox("Введите b")) 'ввод значения переменной b
```

```
c = Val(InputBox("Введите c")) 'ввод значения переменной c
```

```
'P - полупериметр, S - площадь
```

```
p = (a + b + c)
```

```
s = Sqr(p * (p - a) * (p - b) * (p - c))
```

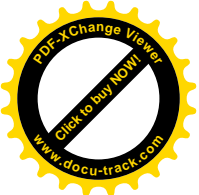
```
Cells(1, 1) = "Площадь="
```

```
Cells(1, 2) = s
```

```
End Sub
```

Операторы управления порядком выполнения команд

Обычно выполнение макроса начинается с выполнения инструкций SUB или FUNCTION и продолжается в порядке следования строк с



операторами до тех пор, пока не встретится инструкция END SUB или END FUNCTION.

Но часто, при проектировании макросов, необходимо выполнять различные варианты программы, в зависимости от значения каких-либо переменных или параметров (условий), задаваемых пользователем во время выполнения программы. На этот случай в VBA предусмотрен ряд операторов, предназначенных для изменения порядка выполнения операторов в программе.

*Оператор безусловного перехода **GoTo***

Позволяет изменить порядок выполнения команд. Задаёт безусловный переход на указанную строку внутри той процедуры, в которой она находится.

Синтаксис инструкции GoTo:

Гото строка

Обязательный аргумент *строка* может быть любой меткой строки или номером строки.

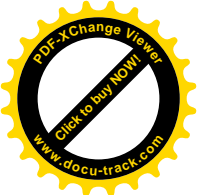
Метка строки — это строка программы, в начале которой указывается имя метки и двоеточие (:). Метки должны иметь уникальные имена в пределах одной процедуры.

Примечание.

Слишком большое число инструкций GoTo затрудняет чтение и отладку программ. По возможности следует использовать структурные управляющие инструкции (Do...Loop, For...Next, If...Then...Else, Select Case). Инструкция GoTo в VBA применяется, главным образом, для обработки ошибок.

*Оператор **If... Then... Else***

Условный оператор If... Then...Else выполняет определённые инструкции или наборы инструкций в зависимости от значения заданных условий.



Синтаксис, т.е. правила записи строчной инструкции If...then...else, следующий:

If *условие* Then [*оператор1*] [Else *оператор2*].

Различают условные операторы **строчный** и **блочный**. Строчный оператор используется в том случае, когда при разветвлении программы необходимо на каждой ветке выполнить по одному оператору, а блочный необходим тогда, когда операторов несколько.

Обязательный параметр *условие* — представляет собой логическое выражение, которое возвращает значение (истина) TRUE или (ложь) FALSE. Если *условие* равно TRUE, то выполняется *оператор1*, если *условие* равно FALSE, то выполняется *оператор2*. Далее выполняется, если иное не предусмотрено программой в *операторах1* или 2, инструкция, следующая по порядку за строкой с инструкцией If... Then...Else.

Часть оператора If, а именно ключевое слово Else и следующие за ним инструкции можно опустить, тогда при условии равном FALSE выполняется оператор, следующий за строкой с инструкцией If. Т.е. инструкция If выглядит так:

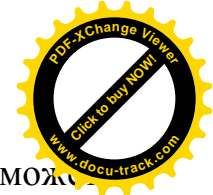
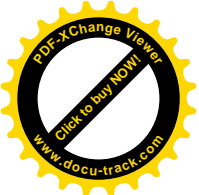
If *условие* Then [*оператор*]

Примечание.

В квадратных скобках обозначены те элементы инструкции, которые не являются обязательными для записи.

В реальных случаях часто требуется обработка нескольких инструкций при выполнении условия. В этих случаях предпочтительнее блочный вид инструкции If, синтаксис которой выглядит следующим образом:

If *условие* Then
[Блок операторов]
[Else
[Блок операторов]]
End If



Блочная структура If удобнее, так как каждый блок операторов может включать в себя произвольное число утверждений. Кроме того, допускается вложение структур If.

Оператор Select Case

Это оператор выбора из многих возможных вариантов. Оператор Select Case выполняет одну из нескольких групп операторов в зависимости от значения выражения. В языках высокого уровня такой оператор называется переключателем.

Синтаксис инструкции

Select Case *выражение*

[Case *список условий-1*

[*операторы-1*]]

[Case *список условий-2*

[*операторы-2*]]

...

[Case *список условий-n*

[*операторы-n*]]

[Case Else

[*операторы_ else*]]

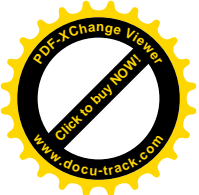
End Select

выражение — обязательный параметр, может быть любым выражением любого типа данных VBA, включая числовой, строковый, или логический.

Список условий — обязательный при наличии инструкции CASE. Содержит результаты вычисления выражения, записанные в одном из видов:

1. Case КОНСТАНТА1, КОНСТАНТА2, КОНСТАНТА3, ...
2. Case Is ЗНАК_ОТНОШЕНИЯ КОНСТАНТА
3. Case КОНСТАНТА1 to КОНСТАНТА2

Выполняется оператор Select Case следующим образом — сначала вычисляется значение выражения, стоящего после ключевых слов Select



Case, затем производится проверка на совпадение значения выражения одного из Case условий, в случае совпадения выполняются операторы, помещенные за этим Case условием. В случае, если совпадения не обнаружено ни в одном из Case условий, то выполняются операторы, находящиеся после ключевых слов Case Else. Если список выражений в Case условии записан в первой форме, то необходимо, чтобы результат вычислений выражения равнялся одной константе из списка, если Case условие записано во второй форме, то необходимо, чтобы выполнялось условие отношения между значением выражения и константой. Если же в третьей форме, то необходимо выполнение соотношения

$$\text{КОНСТАНТА1} \leq \text{выражение} \leq \text{КОНСТАНТА2}.$$

3.7 Программирование алгоритмов разветвляющейся структуры

Алгоритм разветвляющейся структуры – это алгоритм такого процесса, в котором его реализация осуществляется по одному из заранее предусмотренных (возможных) направлений в зависимости от исходных условий или промежуточных результатов. Каждое отдельное направление обработки информации в таком процессе называется ветвью. Выбор ветви определяется проверкой выполнения логического условия, определяющего свойства исходных данных или промежуточных результатов. В каждом конкретном случае процесс обработки данных выполняется лишь по одной из ветвей. Для их реализации используются операторы управления порядком выполнения команд, рассмотренные выше.

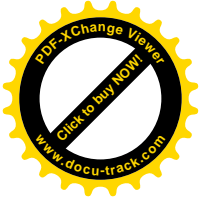
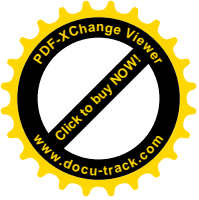
Пример. Дано действительное число x . Вычислить y .

Программа.

Sub Raz2()

Dim x, y As Double

$x = \text{Val}(\text{InputBox}(\text{"Введите } x\text{"}))$ 'ввод значения переменной x



If $x > 0$ Then $y = \sin(x)$ Else $y = 2 * x$

MsgBox ("Значение y=" + Str(y#))

End Sub

Пример. Дано действительное число x . Вычислить y .

Программа

Sub Raz3()

Dim x, y As Double

x = Val(InputBox("Введите x"))

If $x < 0.1$ Then $y = \cos(x^2)$ Else If $x > 0.1$ Then $y = \exp(x)$ Else $y = x^3$

- 2

MsgBox ("Значение y=" + Str(y#))

End Sub

Пример. Даны три числа. Выяснить, существует ли треугольник с такими сторонами.

Программа

Sub Treug()

Dim a, b, c As Double

a = Val(InputBox("Введите сторону a"))

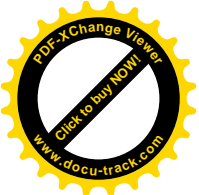
b = Val(InputBox("Введите сторону b"))

c = Val(InputBox("Введите сторону c"))

If $(a + b) > c$ And $(b + c) > a$ And $(a + c) > b$ Then MsgBox ("Треугольник существует") Else MsgBox ("Треугольник не существует") 'оператор
печатать в одной строке

End Sub

3.8 Программирование алгоритмов циклической структуры



Решение многих практических задач сводится к выполнению вычислений по одним и тем же зависимостям, но при разных значениях входящих в них величин. Такой вычислительный процесс называется циклическим, а многократно повторяющиеся участки этого процесса называются циклами.

Различают *регулярные* циклы с управляющим параметром (с известным числом повторений), условием окончания которого является достижение параметром цикла своего конечного значения; циклы *итерационные*, в которых условие повторения или окончания цикла задается по некоторому результату, например, пока не будет достигнута точность вычислений. Реализуются циклы с помощью специальных операторов цикла.

Операторы цикла

Оператор цикла While... Wend

Оператор имеет синтаксис

While *условие*

[*операторы*]

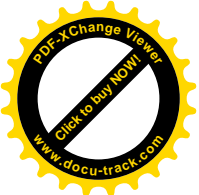
Wend

Синтаксис инструкции While... Wend содержит следующие элементы:

условие — обязательный элемент. Числовое выражение или строковое выражение, которое имеет значение True или False.

операторы — необязательный элемент. Один или несколько операторов, выполняемых, пока условие имеет значение True.

Этот оператор называют оператором цикла с предусловием. Выполняется оператор While... Wend следующим образом. Если *условие* имеет значение True, выполняются все операторы до инструкции Wend. Затем управление возвращается инструкции While и вновь проверяется *условие*. Если *условие* по-прежнему имеет значение True, процесс повторяется. Если оно не имеет значение True, выполнение возобновляется с инструкции, следующей за инструкцией Wend. В связи с этим элемент



условие здесь является условием выполнения цикла. Циклы While... *While* могут иметь любую глубину вложенности.

*Оператор цикла **Do...Loop***

Оператор Do...Loop также используется для выполнения наборов операторов неопределенное число раз. Оператор имеет синтаксис:

Do

[*операторы*]

Loop Until [*условие*]

Синтаксис инструкции Do Loop содержит следующие элементы:

условие — необязательный элемент. Числовое или строковое выражение, которое имеет значение True или False.

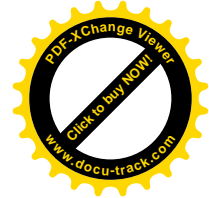
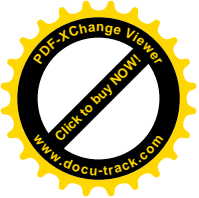
операторы— один или несколько операторов, выполнение которых повторяется, пока условие не приобретет значение True.

Этот оператор называют оператором цикла с постусловием. Выполняется оператор Do...Loop следующим образом. Если *условие* имеет значение False, выполняются все операторы после инструкции Do. Затем управление передается инструкции Until и вновь проверяется *условие*. Если *условие* по-прежнему имеет значение False, процесс повторяется. Если оно имеет значение True, управление передается следующему за инструкцией Loop Until оператору. В связи с этим элемент *условие* здесь является условием выхода из цикла.

*Оператор цикла **For...Next***

Часто при составлении макроса заранее известно количество повторений группы операторов, в таких случаях можно использовать инструкцию For...Next.

Оператор For...Next используется для выполнения наборов операторов указанное число раз. Циклы For используют в качестве счетчика переменную, значение которой увеличивается или уменьшается при каждом выполнении цикла на указанное значение.



Синтаксис инструкции For...Next

For *счетчик* = *начало* To *конец* [step *шаг*]

[*операторы*]

Next [*счетчик*]

счетчик — обязательный элемент. Это должна быть числовая переменная. Она не может иметь тип Boolean или быть элементом массива.

начало — обязательный элемент, содержит начальное значение переменной *счетчик*.

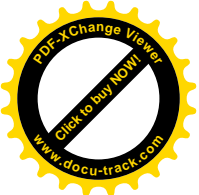
конец — обязательный элемент, содержит конечное значение переменной *счетчик*.

шаг — элемент необязательный, это значение, на которое изменяется счетчик при каждом выполнении тела цикла. Если это значение не задано, по умолчанию *шаг* равен единице. *Шаг* может быть как положительным, так и отрицательным.

операторы — необязательный элемент. Один или несколько операторов между For и Next, которые выполняются указанное число раз.

Инструкция For...Next работает следующим образом: начальное значение элемента *счетчик* сравнивается с конечным значением. Если *шаг* положителен и начальное значение меньше конечного или если шаг отрицателен и начальное значение больше конечного, то управление передается внутрь тела цикла. После выполнения всех операторов в теле цикла значение *шаг* добавляется к текущему значению переменной *счетчик*. После этого операторы тела цикла либо выполняются еще раз (на основе того же условия, которое привело к начальному выполнению цикла), либо цикл завершается и выполнение продолжается с оператора, следующей за Next.

Допускается вложение циклов For...Next (один цикл For...Next располагается внутри другого). Счетчик каждого цикла должен иметь уникальное имя.



Пример. Составить таблицу значений функции $y=x^2$ на отрезке $[2,8]$ шагом 0.5.

Программа

```
Sub Tablica()  
Dim x, y As Double, i As Integer  
i = 1  
Cells(1, 1) = "X": Cells(1, 2) = "Y"  
For x = 2 To 8 Step 0.5  
y = x ^ 2  
i = i + 1  
Cells(i, 1) = x: Cells(i, 2) = y  
Next x  
End Sub
```

Пример. Вычислить сумму квадратов n чисел: 1,4,9,16,25,... . Число n задать произвольно при вводе.

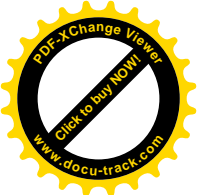
Программа

```
Sub sum()  
Dim n, i As Integer, s As Double  
n = Val(InputBox("Введите количество слагаемых n"))  
s = 0  
For i = 1 To n  
s = s + i ^ 2  
Next i  
MsgBox ("Сумма s=" + Str(s#))  
End Sub
```

Пример. Вычислить произведение квадратов n чисел: 1,4,9,16,25,... . Число n задать произвольно при вводе.

Программа

```
Sub Proiz()
```



```
Dim n, i As Integer, p As Double
n = Val(InputBox("Введите количество слагаемых n"))
p = 1
For i = 1 To n
p = p * i ^ 2
Next i
MsgBox ("Произведение p=" + Str(p#))
End Sub
```

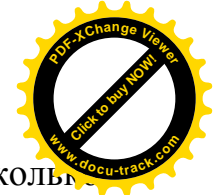
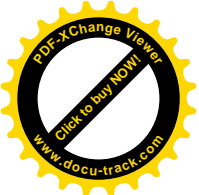
Пример. Дана числовая последовательность:

Вычислить сумму данной последовательности при $n=\infty$ с точностью 0.000001.

Программа

```
Sub Iter()
Dim a, e, s As Double
a = 1           'первый член ряда
s = a           'сумма ряда
e = Val(InputBox("Введите точность вычислений"))
Do              'начало цикла
a = -a / 2      'вычисляем очередной член последовательности
s = s + a       'накапливаем сумму
Loop Until Abs(a) < e 'конец цикла
MsgBox ("Сумма s=" + Str(s#))
End Sub
```

3.9 Вложенные циклы



Внутри одного цикла могут находиться один или несколько других циклов. В этом случае охватывающий цикл называется внешним, а вложенные в него циклы – внутренними. Правила организации как внешнего, так и внутренних циклов аналогичны правилам организации простого цикла. Параметры внешнего и внутреннего циклов изменяются не одновременно, т.е. при одном значении параметра внешнего цикла параметр внутреннего последовательно принимает все возможные значения. При организации вложенных циклов необходимо следить за тем, чтобы область действия внутреннего цикла не выходила за область действия внешнего цикла.

Пример. Составить таблицу значений функции $y = \sin(x)$ на отрезке $[1, 4]$ с шагом $h=0.5$.

Программа

```
Sub V1Circle()
```

```
Dim x, s, a, b, h As Double, i, n, k As Integer
```

```
a = Val(InputBox("Введите a"))
```

```
b = Val(InputBox("Введите b"))
```

```
h = Val(InputBox("Введите шаг h"))
```

```
n = Val(InputBox("Введите количество слагаемых n"))
```

```
k = 1
```

```
Cells(1, 1) = "X": Cells(1, 2) = "S"
```

```
For x = a To b Step h
```

```
s = 0
```

```
For i = 1 To n
```

```
s = s + x / i
```

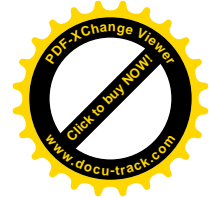
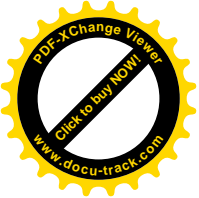
```
Next i
```

```
k = k + 1
```

```
Cells(k, 1) = x: Cells(k, 2) = s
```

```
Next x
```

```
End Sub
```



Массивы

Массивом называется упорядоченная последовательность величин, обозначаемая одним именем. Упорядоченность заключается в том, что элементы массива располагаются в последовательных ячейках памяти. Чтобы получить доступ к нужному элементу массива нужно указать имя массива и индекс этого элемента. Имя массива образуется также как имя переменных. Различают одномерные и двумерные массивы. Одномерный массив – это список переменных, двумерный массив – таблица, имеющая строки и столбцы. Элементы одномерного массива снабжаются одним индексом, заключенным в круглые скобки. Он определяет порядковый номер элемента в массиве. Элементы двумерного массива снабжаются двумя индексами, заключенными в квадратные скобки и разделенными запятой. Первый индекс номер строки, второй номер столбца, на пересечении которых расположен элемент в таблице (матрице). Например: $A(5)=3$, $B(2,3)=6$. Все используемые массивы должны быть описаны до их использования в программе.

Оператор описания DIM

Оператор Dim применяется для описания переменных и выделения для них памяти.

Синтаксис инструкции

Dim имя переменной [(индексы)] [As тип]

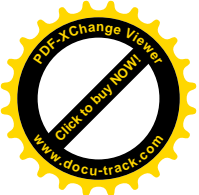
имя переменной — обязательный элемент, содержит имя переменной;

индексы — необязательный элемент. Если индексы отсутствуют, то описана простая переменная;

тип — необязательный элемент, это тип данных переменной. Если тип данных не указан, по умолчанию переменная получает тип Variant.

Например:

Dim Num As Integer явно объявляет переменную целого типа .



Если необходимо описать массив, то с помощью индексов указывается размерность массива. Например:

Dim B(3,3) As Single

Dim A(11) As Integer

Первый оператор объявляет двумерный массив 3*3 матрицу, состоящую из действительных чисел. Второй оператор объявляет одномерный массив (вектор) из 12 целых чисел, причем по умолчанию первый элемент массива будет A(0) а последний A(11). В этом случае говорят, что 0 - базовый индекс. Можно изменить базовый индекс, написав в области описания модуля инструкцию Option base 1. После этого индексы массивов будут нумероваться с единицы. Другим способом изменения базового индекса является использование ключевого слова To при объявлении массива:

Dim B(1 To 3, 1 To 3) As Single

Dim A(1 To 12) As Integer

Массив в программе определяется поэлементно. Например:

Dim B(1 To 2, 1 To 3) As Single

B(1,1)=2: B(1,2)=5: B(1,3)=4:

B(2,1)=7: B(2,2)=1: B(2,3)=3

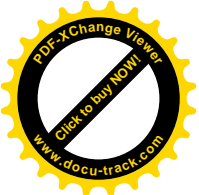
Для определения одномерных массивов можно использовать функцию *Array*, преобразующую список элементов, разделенных запятыми, в вектор из этих значений, и присваивающую им тип Variant. Например:

Dim A As Variant

A= Array(10,40,60)

Здесь A(0)=10, A(1)=40, A(2)=60.

Удобным способом ввода/вывода массивов является использование рабочего листа Excel. Исходный массив считывается из соответствующих ячеек рабочего листа, выводится массив также в ячейки рабочего листа. В этом случае используется оператор цикла *For...Next* и свойство Cells() для



указания конкретного местоположения элемента массива. Если макрос написан для того же рабочего листа, где располагаются элементы массива, то указание на выбор этого рабочего листа можно опускать.

Рассмотрим некоторые **алгоритмы обработки одномерных массивов**.

Поиск максимального элемента

Пример. Дан массив А из пяти чисел. Найти максимальный элемент массива и его индекс.

Программа

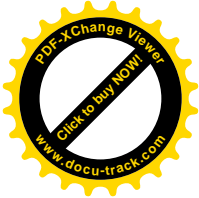
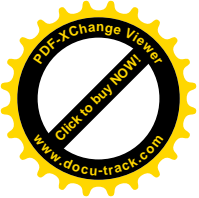
```
Sub Massiv()  
Dim a(5) As Single  
Dim max As Single, i, k As Integer  
'ввод массива  
For i = 1 To 5  
a(i) = Cells(1, i + 1)  
Next i  
max = a(1)  
k = 1          'индекс максимального элемента  
For i = 1 To 5  
If max < a(i) Then max = a(i): k = i  
Next i  
MsgBox ("Максимальный элемент=" + Str(max!))  
MsgBox ("Индекс максимального элемента k=" + Str(k))  
End Sub
```

Поиск количества элементов по данному правилу

Пример. Дан массив А из пяти чисел. Найти количество положительных элементов массива.

Программа

```
Sub Kol()
```



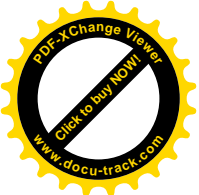
```
Dim a(5) As Single
Dim i, k As Integer
For i = 1 To 5
a(i) = Cells(1, i + 1)
Next i
k = 0
For i = 1 To 5
If a(i) > 0 Then k = k + 1
Next i
MsgBox ("Количество положительных элементов k=" + Str(k))
End Sub
```

Нахождение элементов массива по элементам другого массива

Пример. Дан вектор A(5). Найти элементы вектора B(5) по следующему правилу: $b_i = \sin(a_i), i = 1, \dots, 5$.

Программа

```
Sub NewMassiv()
Dim a(5), b(5) As Single
Dim i As Integer
'ввод массива a(5)
For i = 1 To 5
a(i) = Cells(1, i + 1)
Next i
For i = 1 To 5
b(i) = Sin(a(i))
Next i
'вывод нового массива b(5)
Cells(3, 1) = "Массив b(5)"
For i = 1 To 5
Cells(3, i + 1) = b(i)
```

Next i

End Sub

Умножение двух векторов

Пример. Найти скалярное произведение двух векторов A(5) и C(5).

Программа

Sub SkalProiz()

Dim a(5), c(5) As Single

Dim i As Integer, s As Single

'ввод массива a(5)

For i = 1 To 5

a(i) = Cells(1, i + 1).Value

Next i

'ввод массива c(5)

For i = 1 To 5

c(i) = Cells(7, i + 1).Value

Next i

s = 0

For i = 1 To 5

s = s + a(i) * c(i)

Next i

MsgBox ("Скалярное произведение s=" + Str(s))

End Sub

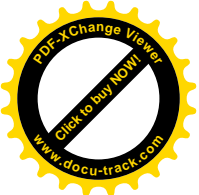
Некоторые алгоритмы обработки матриц

Поиск минимального элемента матрицы

Пример. Дана матрица A(2,3). Найти минимальный элемент.

Программа

Sub MinMat()



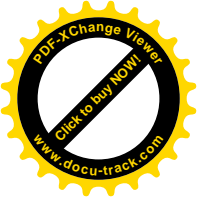
```
Dim a(2, 3) As Single
Dim min As Single, i, j As Integer
'ввод матрицы
For i = 1 To 2
For j = 1 To 3
a(i, j) = Cells(i + 1, j)
Next j
Next i
min = a(1, 1)
For i = 1 To 2
For j = 1 To 3
If a(i, j) < min Then min = a(i, j)
Next j
Next i
MsgBox ("Минимальный элемент min=" + Str(min))
End Sub
```

Умножение матрицы на вектор

Пример. Дана матрица A(2,3) и вектор B(3). Найти произведение A*B.

Программа

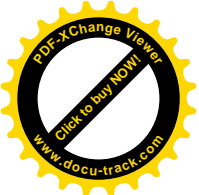
```
Sub UmnMatNaVec()
Dim a(2, 3) As Single, b(3) As Single, c(2) As Single
Dim s As Single, i, j As Integer
'ввод матрицы
For i = 1 To 2
For j = 1 To 3
a(i, j) = Cells(i + 1, j)
Next j
Next i
```



```
'ВВОД массива  
For i = 1 To 3  
b(i) = Cells(4, i + 1)  
Next i  
For i = 1 To 2  
s = 0  
For j = 1 To 3  
s = s + a(i, j) * b(j)  
Next j  
c(i) = s  
Next i  
'ВЫВОД нового массива b(5)  
Cells(6, 1) = "Массив c(2)"  
For i = 1 To 2  
Cells(6, i + 1) = c(i)  
Next i  
End Sub
```

Подпрограммы и функции

При разработке макросов часто требуется выполнять одни и те же законченные действия в различных его частях. Чтобы избежать многократного набора кода для этих действий, целесообразно описать их в виде подпрограммы и обращаться к ней по мере необходимости. Различают подпрограммы-процедуры и подпрограммы-функции. Описываются подпрограммы при помощи инструкций Sub...End Sub – это подпрограмма-процедура, инструкций Function...End Function – это подпрограмма-функция. Подпрограмма Function и подпрограмма Sub являются самостоятельной программой, которая может получать аргументы, выполнять последовательность операторов и изменять значения своих аргументов.



Различие между процедурами и функциями состоит в том, что функция всегда возвращает значение и процедура Function может применяться в правой части выражения, как и любая другая встроенная функция, например, Sqr, Cos или Chr, а процедура Sub не может. Для работы подпрограмм иногда необходимо передать данные, которые носят названия параметров или аргументов подпрограммы. Параметры передаются в подпрограммы при ее вызове при помощи перечисления их после имени подпрограммы. При этом описание действий в подпрограмме осуществляется с использованием формальных параметров. Обращение к подпрограмме осуществляется с фактическими параметрами, которые должны соответствовать формальным по числу, типу и месту расположения.

Оператор Sub...End Sub

Описывает имя, аргументы и текст программы, составляющий тело процедуры Sub.

Синтаксис инструкции

Sub *имя* [(списка аргументов)]

операторы

End Sub

Здесь

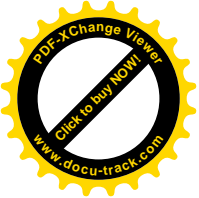
имя — обязательный элемент, это имя процедуры Sub, удовлетворяющее =: стандартным правилам именования переменных.

список аргументов — необязательный элемент. Список переменных, представляющий параметры, которые передаются в процедуру Sub при ее вызове (формальные параметры). Имена переменных разделяются запятой.

операторы — любая группа операторов, выполняемых в процедуре Sub, находятся между Sub и End Sub.

Не допускается определение процедуры Sub внутри другой процедуры Sub или Function.

Оператор вызова подпрограмм Call



Обращение к процедуре осуществляется с помощью оператора Call.

Синтаксис оператора:

Call *имя подпрограммы* [*параметры*]

Синтаксис инструкции Call содержит следующие элементы:

имя подпрограммы — обязательный элемент, содержит имя вызываемой процедуры.

параметры — необязательный элемент, это разделяемый запятыми список переменных, массивов или выражений, передаваемых в процедуру (фактические параметры).

Для передачи в процедуру полного массива следует воспользоваться именем массива с пустыми скобками.

Оператор Function...End Function

Описывает имя, аргументы и текст программы, составляющий тело подпрограммы- функции Function.

Синтаксис инструкции

Function *имя* [(*список аргументов*)] [As *тип*]

операторы

[*имя* = *выражение*]

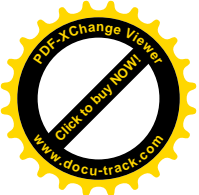
End Function

имя — обязательный элемент. Содержит имя подпрограммы-функции Function, удовлетворяющее стандартным правилам именования переменных;

список аргументов — необязательный элемент, это список переменных, представляющий параметры, которые передаются в подпрограмму Function при ее вызове (формальные параметры). Имена переменных разделяются запятой.

тип — необязательный элемент. Тип данных значения, возвращаемого подпрограммой Function.

операторы — элемент, содержащий любую группу операторов, выполняемых внутри процедуры Function.



выражение — возвращаемое значение подпрограммой Function.

Использование подпрограммы Function (нестандартной функции) аналогично использованию стандартных функций. Обращение к ней можно записать, например, в правой части оператора присваивания, при этом указывается имя подпрограммы- функции и в круглых скобках – фактические параметры т.е. *список аргументов*, заменяющих формальные параметры в операторе Function

Пример. Вычислить полярные координаты точки, если известны ее декартовы координаты. Связь полярных и декартовых координат определяется по формулам: $x=R \cos\alpha$; $y=R \sin\alpha$.

Программа

```
Sub PolKoord()
```

```
Dim x As Double, y As Double
```

```
Cells(1, 7) = "Полярные координаты"
```

```
Cells(1, 7).Interior.ColorIndex = Int(Rnd * 10)
```

```
x = Val(InputBox("Введите x"))
```

```
y = Val(InputBox("Введите y"))
```

```
Cells(2, 7) = "Радиус"
```

```
Cells(2, 8) = Radius(x, y)
```

```
Cells(3, 7) = "Угол"
```

```
Cells(3, 8) = Ugol(x, y)
```

```
End Sub
```

```
Function Radius(x As Double, y As Double) As Double
```

```
Radius = (x ^ 2 + y ^ 2) ^ (1 / 2)
```

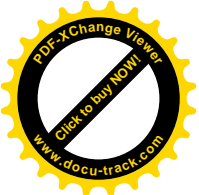
```
End Function
```

```
Function Ugol(x As Double, y As Double) As Double
```

```
Dim I1, I2, I3 As Boolean
```

```
Ugol = Atn(y / x) * 180 / (3.14152654)
```

```
I1 = x < 0 And y > 0
```



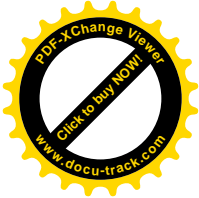
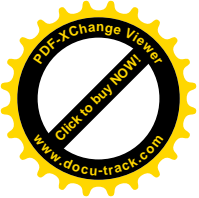
```
I2 = x < 0 And y < 0
I3 = x > 0 And y < 0
If I1 Or I2 Then Ugol = Ugol + 180
If I3 Then Ugol = Ugol + 360
End Function
```

Пример. Решить систему уравнений методом Гаусса.

Программа

```
Sub Method()
Dim a(3, 4), x(3), a1(3, 3)
n = 3
For i = 1 To n
For j = 1 To n + 1
a(i, j) = Cells(i + 1, j)
Next j
Next i
For k = 1 To n - 1
Call max(a(), n, k)
Call gauss(a(), n, k)
Next k
Call obr(a(), n, x())
Cells(6, 1) = "Вектор x"
For i = 1 To n
Cells(6, i + 1) = x(i)
Next i
End Sub

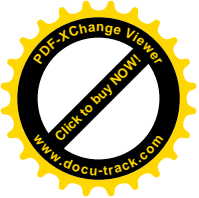
Sub max(a(), n, k)
h = Abs(a(k, k))
For i = k To n
If Abs(a(i, k)) > h Then h = Abs(a(i, k)): l = i
```



```
Next i
For j = 1 To n + 1
s = a(k, j): a(k, j) = a(l, j): a(l, j) = s
Next j
End Sub

Sub gauss(a(), n, k)
For i = k + 1 To n
d = a(k, k)
w = a(i, k)
For j = k To n + 1
a(i, j) = a(i, j) - a(k, j) / d * w
Next j
Next i
End Sub

Sub obr(a(), n, x())
x(n) = a(n, n + 1) / a(n, n)
For k = n - 1 To 1 Step -1
s = 0
For j = k + 1 To n
s = s + a(k, j) * x(j)
Next j
x(k) = (a(k, n + 1) - s) / a(k, k)
Next k
End Sub
```

ТЕМА 4 РАБОТА В MS ACCESS

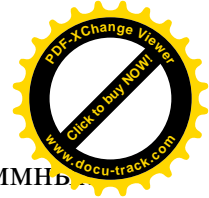
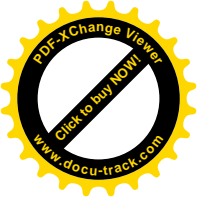
4.1. Что такое база данных?

В деловой или личной сфере часто приходится работать с данными из разных источников, каждый из которых связан с определенным видом деятельности. Для координации всех этих данных необходимы определенные знания и организационные навыки.

Microsoft Access объединяет сведения из разных источников в одной реляционной базе данных. Создаваемые формы, запросы и отчеты позволяют быстро и эффективно обновлять данные, получать ответы на вопросы, осуществлять поиск нужных данных, анализировать данные, печатать отчеты, диаграммы и почтовые наклейки.

В базе данных сведения из каждого источника сохраняются в отдельной таблице. При работе с данными из нескольких таблиц устанавливаются связи между таблицами. Для поиска и отбора данных, удовлетворяющих определенным условиям, создается запрос. Запросы позволяют также обновить или удалить одновременно несколько записей, выполнить встроенные или специальные вычисления. Для просмотра, ввода или изменения данных прямо в таблице применяются формы. Форма позволяет отобразить данные из одной или нескольких таблиц и вывести их на экран, используя стандартный или созданный пользователем макет. Для анализа данных или распечатки их определенным образом используется отчет. Например, можно создать и напечатать отчет, группирующий данные и вычисляющий итоги, или отчет для распечатки почтовых наклеек.

Современные информационные системы характеризуются большими объемами хранимых данных, их сложной организацией, а также высокими требованиями к скорости и эффективности обработки этих данных. Это



становится возможным при использовании специальных программных средств - систем управления базами данных (СУБД).

База данных (БД) - это поименованная совокупность данных, относящихся к определенной предметной области.

Система управления базами данных (СУБД) - это комплекс программных и языковых средств, необходимых для создания, обработки баз данных и поддержания их в актуальном состоянии организации поиска в них необходимой информации.

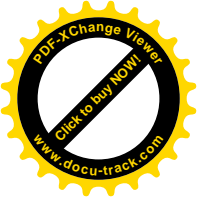
Почти все современные СУБД основаны на реляционной модели данных. Название "реляционная" связано с тем, что каждая запись в такой базе данных содержит информацию, относящуюся (related) только к одному объекту. Кроме того, с данными о двух объектах можно работать как с единым целым, основанным на значениях связанных между собой данных. Все данные в реляционной БД представлены в виде таблиц. Каждая строка таблицы содержит информацию только об одном объекте и называется записью. Столбец таблицы содержит однотипную для всех записей информацию и называется полем.

Для успешного функционирования базы данных важна правильная организация данных в ней. При определении структуры данных в базе выделяют следующие основные понятия.

Класс объектов - совокупность объектов, обладающих одинаковым набором свойств. Например, в базе данных о ВУЗе классами объектов являются студенты, преподаватели, предметы. Для каждого отдельного объекта из данного класса объектов в таблице создается отдельная запись.

Свойство (атрибут) - определенная часть информации о некотором объекте. Хранится в виде столбца (поля) таблицы. Например, фамилия, имя, отчество - это свойства для объекта Студент.

Связь (отношение) - способ, которым связана информация о разных объектах.



4.2. Проектирование базы данных

В Microsoft Access, прежде чем создавать таблицы, формы и другие объекты необходимо задать структуру базы данных. Хорошая структура базы данных является основой для создания адекватной требованиям, эффективной базы данных.

4.2.1. Этапы проектирования базы данных

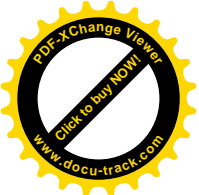
Ниже приведены основные этапы проектирования базы данных:

- 1. Определение цели создания базы данных.**
- 2. Определение таблиц, которые должна содержать база данных.**
- 3. Определение необходимых в таблице полей.**
- 4. Задание индивидуального значения каждому полю.**
- 5. Определение связей между таблицами.**
- 6. Обновление структуры базы данных.**
- 7. Добавление данных и создание других объектов базы данных.**
- 8. Использование средств анализа в Microsoft Access.**

1. Определение цели создания базы данных

На первом этапе проектирования базы данных необходимо определить цель создания базы данных, основные ее функции и информацию, которую она должна содержать. То есть нужно определить основные темы таблиц базы данных и информацию, которую будут содержать поля таблиц.

База данных должна отвечать требованиям тех, кто будет непосредственно с ней работать. Для этого нужно определить темы, которые должна покрывать база данных, отчеты, которые она должна выдавать, проанализировать формы, которые в настоящий момент используются для записи данных, сравнить создаваемую базу данных с хорошо спроектированной, подобной ей базой.



2. Определение таблиц, которые должна содержать база данных

Одним из наиболее сложных этапов в процессе проектирования базы данных является разработка таблиц, так как результаты, которые должна выдавать база данных (отчеты, выходные формы и др.) не всегда дают полное представление о структуре таблицы.

При проектировании таблиц вовсе не обязательно использовать Microsoft Access. Сначала лучше разработать структуру на бумаге. При проектировке таблиц, рекомендуется руководствоваться следующими основными принципами:

- Информация в таблице не должна дублироваться. Не должно быть повторений и между таблицами.

Когда определенная информация храниться только в одной таблице, то и изменять ее придется только в одном месте. Это делает работу более эффективной, а также исключает возможность несовпадения информации в разных таблицах. Например, в одной таблице должны содержаться адреса и телефоны клиентов.

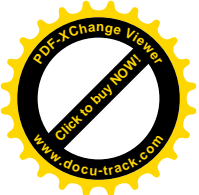
- Каждая таблица должна содержать информацию только на одну тему.

Сведения на каждую тему обрабатываются намного легче, если содержаться они в независимых друг от друга таблицах. Например, адреса и заказы клиентов хранятся в разных таблицах, с тем, чтобы при удалении заказа информация о клиенте осталась в базе данных.

3. Определение необходимых в таблице полей

Каждая таблица содержит информацию на отдельную тему, а каждое поле в таблице содержит отдельные сведения по теме таблицы. Например, в таблице с данными о клиенте могут содержаться поля с названием компании, адресом, городом, страной и номером телефона. При разработке полей для каждой таблицы необходимо помнить:

- Каждое поле должно быть связано с темой таблицы.



- Не рекомендуется включать в таблицу данные, которые являются результатом выражения.

- В таблице должна присутствовать вся необходимая информация.
- Информацию следует разбивать на наименьшие логические единицы (Например, поля «Имя» и «Фамилия», а не общее поле «Имя»).

4. Задание индивидуального значения каждому полю

С тем чтобы Microsoft Access мог связать данные из разных таблиц, например, данные о клиенте и его заказы, каждая таблица должна содержать поле или набор полей, которые будут задавать индивидуальное значение каждой записи в таблице. Такое поле или набор полей называют основным ключом.

5. Определение связей между таблицами

После распределения данных по таблицам и определения ключевых полей необходимо выбрать схему для связи данных в разных таблицах. Для этого нужно определить связи между таблицами.

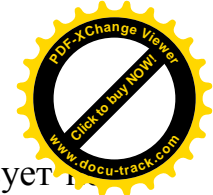
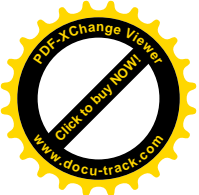
Типы связей между объектами

Основным структурным компонентом базы данных, как правило, является таблица. При определении состава таблиц следует руководствоваться правилом: в каждой таблице должны храниться данные только об одном классе объектов. Например, в одной таблице нельзя хранить анкетные данные студента и фамилии преподавателей, которым он сдавал экзамены, т.к. это свойства разных классов объектов.

Если в базе данных должна содержаться информация о разных классах объектов, то она должна быть разбита на отдельные таблицы. Связь между таблицами осуществляется с помощью общих полей.

Связи между любыми двумя таблицами относятся к одному из трех типов: один- к- одному (1:1), один- ко -многим (1:M) и много- ко- многим (M:M).

Связь типа "один- к- одному" (1:1)



При этом типе связи каждой записи в одной таблице соответствует не более одной записи в другой таблице. Этот вид связи встречается довольно редко. В основном в тех случаях, когда часть информации об объекте либо редко используется, либо является конфиденциальной (такая информация хранится в отдельной таблице, которая защищена от несанкционированного доступа).

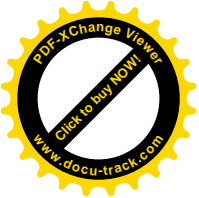
Связь типа "один- ко- многим" (1:M)

При таком типе связи каждой записи в одной таблице соответствует несколько записей в связанной таблице. Этот наиболее распространенный тип связей. Для его реализации используются две таблицы. Одна из них представляет сторону "один", другая - сторону "много". Например, нужно иметь информацию о студентах и результатах сдачи ими экзаменов (дата сдачи, предмет, оценка и т.д.). Если все это хранить в одной таблице, то ее объем неоправданно возрастет, т.к. в ней для каждой записи об очередном экзамене должны повторяться все анкетные сведения о студенте. Поскольку Студент и Экзамены - это разные классы объектов, то и свойства их должны храниться в разных таблицах.

Решением этой задачи является создание двух таблиц. Условно назовем их Студенты и Экзамены. В каждой из них хранятся соответствующие свойства. Для связи этих таблиц нужно использовать только часть информации о студенте, сдающем экзамен. Но она должна однозначно определять каждого студента среди всех. Такой информацией может явиться, например, номер зачетки (он уникален для каждого студента).

В таблице со стороны "один" (в нашем примере Студенты) такие поля называются ключевыми. Основное требование к значениям в ключевых полях - это их уникальность для каждой записи (т.е. они не должны повторяться).

Связь типа "много- ко- многим" (M:M)



При таком типе связи множеству записей в одной таблице соответствует множество записей в связанной таблице. Большинство современных СУБД непосредственно не поддерживают такой тип связи. Для его реализации такая связь разбивается на две связи типа один- ко- многим . Соответственно, для хранения информации потребуется уже три таблицы: две со стороны "много" и одна со стороны "один". Связь между этими тремя таблицами также осуществляется по общим полям.

6. Обновление структуры базы данных

После проектирования таблиц, полей и связей необходимо еще раз просмотреть структуру базы данных и выявить возможные недочеты. Желательно это сделать на данном этапе, пока таблицы не заполнены данными.

Для проверки необходимо создать несколько таблиц, определить связи между ними и ввести несколько записей в каждую таблицу, затем посмотреть, отвечает ли база данных поставленным требованиям. Рекомендуется также создать черновые выходные формы и отчеты и проверить, выдают ли они требуемую информацию. Кроме того, необходимо исключить из таблиц все возможные повторения данных.

7. Добавление данных и создание других объектов базы данных

Если структуры таблиц отвечают поставленным требованиям, то можно вводить все данные. Затем можно создавать любые запросы, формы, отчеты, макросы и модули.

8. Использование средств анализа в Microsoft Access

В Microsoft Access существует два инструмента для усовершенствования структуры баз данных. Мастер анализа таблиц исследует таблицу, в случае необходимости предлагает новую ее структуру и связи, а также переделывает ее.

Таблица хранит сведения по конкретному вопросу, например, таблица «Товары» содержит данные только о товарах, а таблица «Поставщики» содержит данные только о компаниях, поставляющих товары. Для того, чтобы данные не дублировались и не происходило ошибок при их выдаче, необходимо распределить данные разного рода по разным таблицам. Данные в таблице организованы в столбцы (называемые полями) и в строки (называемые записями).

Поля		Поставщики : таблица	
		Код поставщ	Название
		1	Exotic Liquids
		2	New Orleans Cajun Delights
		3	Grandma Kelly's Homestead
		4	Tokyo Traders

Товары : таблица		
Марка	Код поставщика	На складе
Индийский чай	1	39
Оливки	1	17
Анисовый сироп	1	13
Соевый сироп	2	53

Рис 23. Поля таблиц

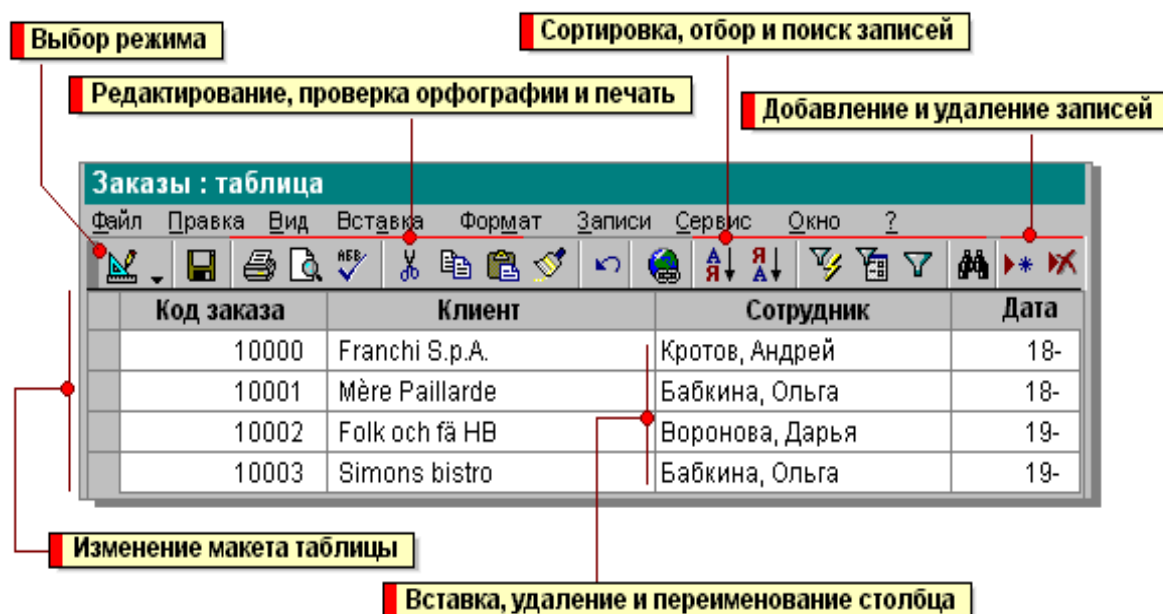


Рис. 24

Анализатор быстродействия исследует всю базу данных, дает рекомендации по ее улучшению, а также осуществляет их.

Общие сведения о таблицах

Каждое поле в таблице «Товары» содержит данные определенного рода о каждом товаре, например, название товара. Каждая запись в таблице «Товары» содержит все сведения об одном товаре, такие как название, код поставщика, количество товара на складе и т.д.

Код поставщика, название организации и т. д. вводятся один раз для каждого поставщика товаров в таблицу «Поставщики». Таблица «Товары» также имеет поле «Код Поставщика», чтобы при вводе нового товара указать код фирмы поставщика. Значение поля «Код Поставщика» из таблиц

«Товары» и «Поставщики» используется для редактирования, просмотра печати данных одновременно из двух таблиц.

В режиме таблицы добавляются, редактируются или просматриваются табличные данные. Также можно проверить орфографию и напечатать табличные данные, отфильтровать и отсортировать записи, изменить внешний вид таблицы или изменить структуру таблицы, добавив или удалив столбцы.

В режиме конструктора таблицы (Рис. 25) можно создать целую таблицу, добавляя новые поля или удаляя и настраивая существующие поля

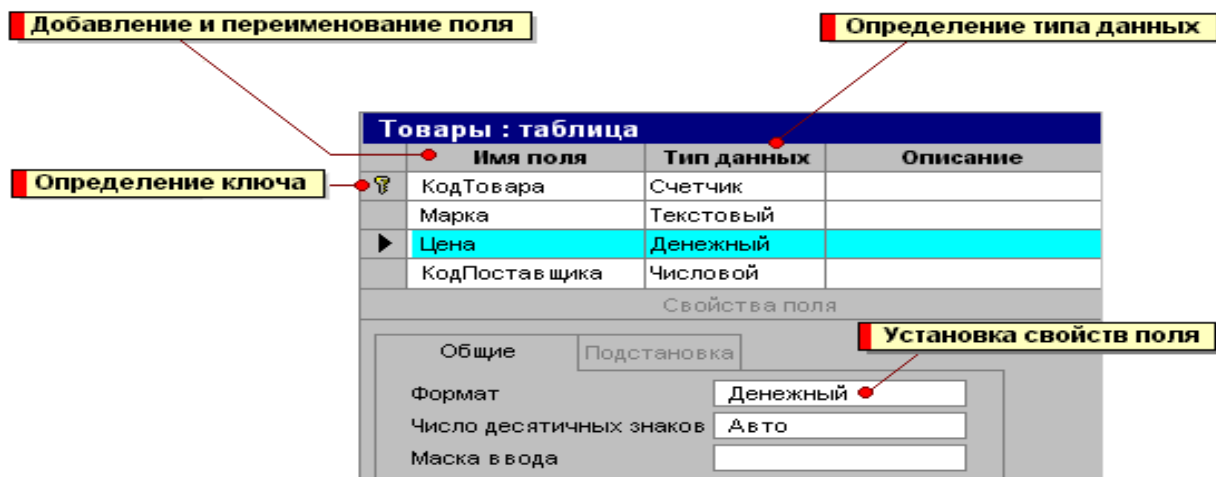



Рис.25

таблицы.

Чтобы добавить поле, в верхней части окна таблицы в режиме конструктора следует ввести имя поля и определить его тип. Чтобы переименовать поле, измените его имя в столбце Имя поля.

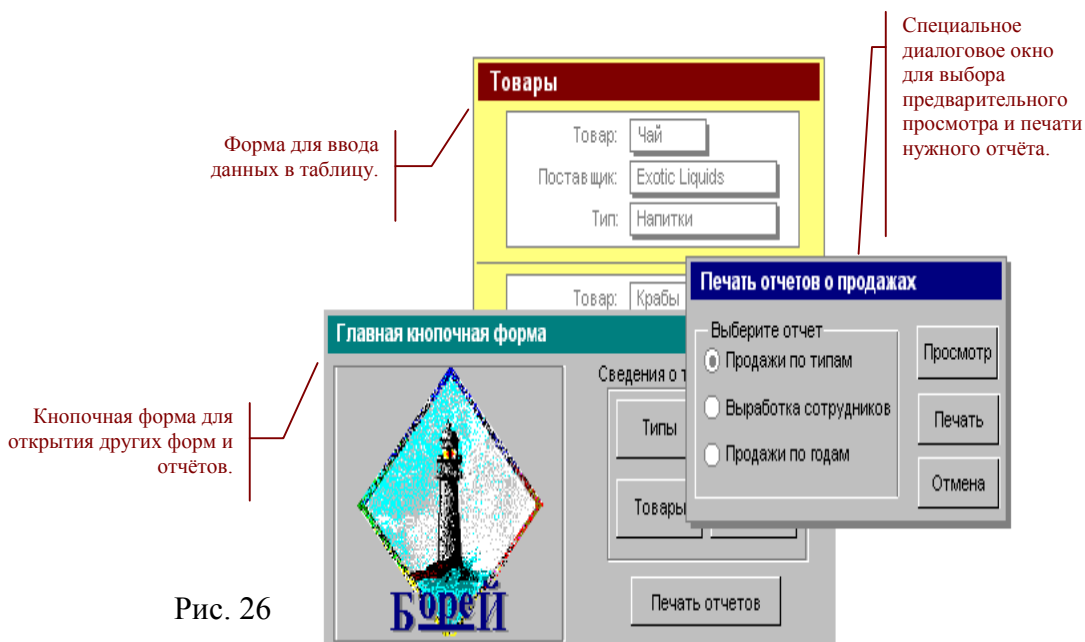
Уникальная метка, называемая ключом, используется для определения каждой записи таблицы. Подобно тому, как номерной знак однозначно определяет автомобиль, ключ определяет запись. Ключевые поля в таблицах используются для создания межтабличных связей. Например, поле «Код Товара» из таблицы «Товары» используется для объединения сведений о товарах и заказах из разных таблиц для их вывода на печать.

Чтобы определить ключ, выделите строку с описанием нужного поля, нажмите кнопку **Ключ**  на панели инструментов.

Каждое поле имеет свойства, изменяя которые, можно управлять сохранением, обработкой и отображением данных поля. Например, при задании свойству Формат (Format) значения Денежный для поля с числовым или денежным типом данных автоматически добавляется разделитель групп разрядов и символ рубля (1 234р.). Для задания свойства поля выберите его в верхней части окна конструктора таблиц и в нижней части окна введите значение этого свойства или выберите его из списка. Набор доступных свойств определяется типом данных поля.

Общие сведения о формах

Формы используются для различных целей (Рис. 26).



Большая часть данных, представленных в форме, берется из таблицы или запроса. Другая информация, несвязанная ни с таблицей, ни с запросом, хранится в макете формы (Рис 27).

Для создания связи между формой и записями таблицы, являющейся

Режим формы

Цены для сотрудников

Поясняющий текст хранится в макете формы.

Товар: 1 Марка: Индийский чай Цена: 18 000р.

Графические элементы хранятся в макете формы.

Данные берутся из полей базовой таблицы.

Цена для сотрудника: 13 500р.

Вычисленное значение выражения хранится в макете формы.

Пожалуйста, оплатите покупку.

Рис. 27

источником данных формы, используются графические объекты, называемые элементами управления. Чаще всего для отображения и ввода данных используется поле.

Общие сведения о запросах и их применение

С помощью запросов можно просматривать, анализировать и изменять данные из нескольких таблиц (Рис. 28). Они также используются в качестве источника данных для форм и отчетов.

Наиболее часто используется запрос на выборку (Рис. 29). При его

Режим конструктора

Эти надписи содержат поясняющий текст.

Эти поля отображают данные из таблицы «Товары».

Область данных

Товар: Марка: Цена:

Товар Марка Цена

Цена для сотрудника: $=[\text{Цена}] * 0,75$

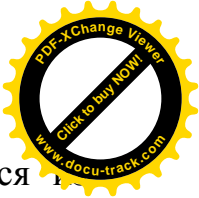
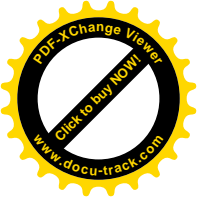
Пожалуйста, оплатите покупку.

Товары: таблица

Товар:	Марка:	Цена:
5	Зефир	21 350р.
6	Крабы	25 000р.
7	Икра	30 000р.

Это поле для вычисления итогов с помощью выражений.

Рис. 28



выполнении данные, удовлетворяющие условиям отбора, выбираются из одной или нескольких таблиц и выводятся в определенном порядке.

Отобразить данные из нескольких таблиц и отсортировать их в нужном порядке.

Товары и поставщики: запрос на выборку			Выработка сотрудников: запрос на выборку		
Марка	Поставщик	Телефон	Ф.И.О.	Сумма заказов	Общий объем
Анисовый сироп	Pavlova, Ltd.	(03) 44	Абрамов А.Н.	125	197 110 840р.
Ветчина	Exotic Liquids	(71) 55	Синица Е.А.	53	86 737 340р.
Зефир	Gai pâturage	38.7			
Икра	Pavlova, Ltd.	(03) 44			
Индийский чай	Exotic				

Выполнить вычисления над группами записей.

Распределение заказов по кварталам: перекрестный запрос				
Категория	Кв. 1	Кв. 2	Кв. 3	К
Кондитерские изделия	7 696 110р.	26 942 160р.	13 800 180р.	30
Молочные продукты	2 666 220р.	5 449 500р.	5 988 400р.	18
Мясо/птица	7 737 140р.	6 175 750р.	17 118 930р.	14
Напитки	13 665 870р.	10 494 940р.	15 921 140р.	21
Приправы	11 624 000р.	9 160 840р.	3 685 070р.	17

Вычислять итоговые значения и выводить их в компактном формате, подобном формату электронной таблицы.

Рис.29

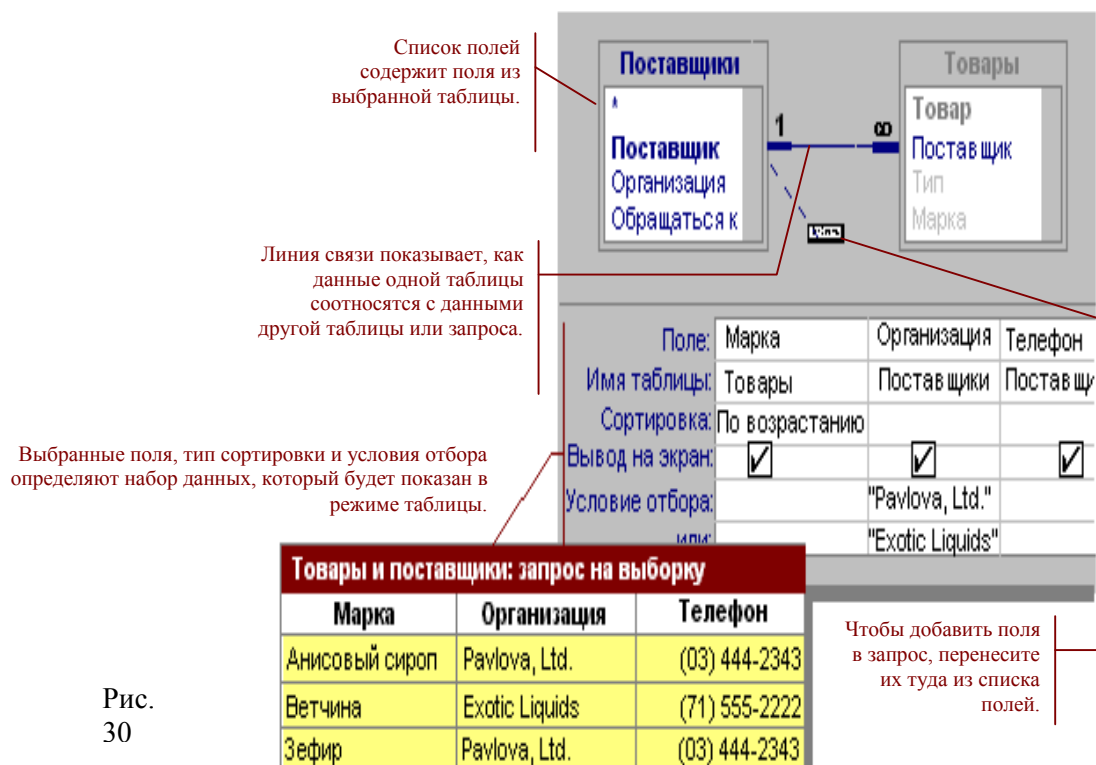


Рис.
30

Запрос можно создать с помощью мастера или самостоятельно. Во втором случае в режиме конструктора следует выбрать таблицы или запросы, содержащие нужные данные, и заполнить бланк запроса (Рис.30).

Общие сведения об отчетах

Отчет — это гибкое и эффективное средство для организации данных при выводе на печать. С помощью отчета имеется возможность вывести необходимые сведения в том виде, в котором требуется (Рис. 31).

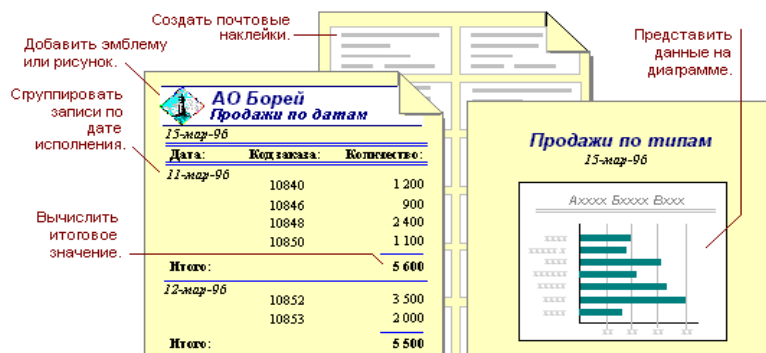


Рис. 31

Больше всего сведений в отчете берется из базовой таблицы, запроса или инструкции SQL, являющихся источниками данных для отчета (Рис. 32).



Рис. 32

Другие сведения вводятся при разработке отчета.

Для создания связи между отчётами и его исходными данными применяются элементы управления. Ими могут быть поля, содержащие имена или числа, надписи для заголовков, декоративные линии для графического оформления отчёта (Рис. 33).

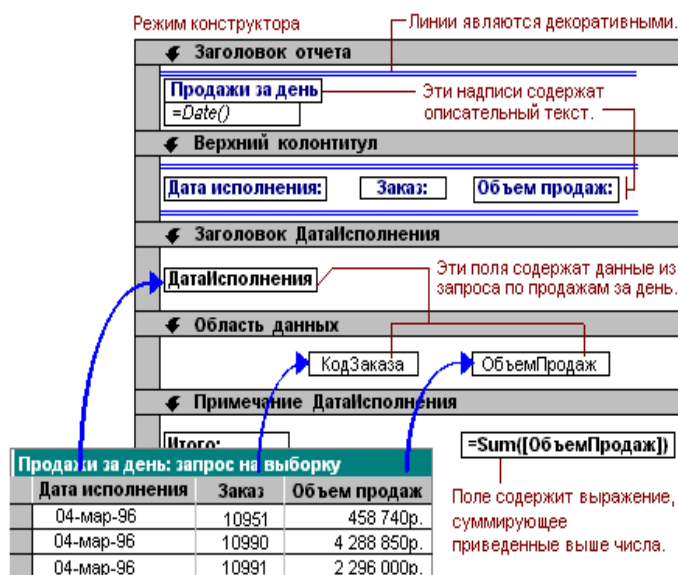



Рис. 33

4.3. Создание базы данных

4.3.1. Создание базы данных с помощью мастера

1. При запуске Microsoft Access открывается диалоговое окно, в котором предлагается создать новую базу данных или открыть существующую. Если появляется это диалоговое окно, выберите параметр **Запуск мастера**, а затем нажмите кнопку **ОК**.

Если база данных уже была открыта или если было закрыто первое диалоговое окно, нажмите кнопку **Создать базу данных**  на панели инструментов.

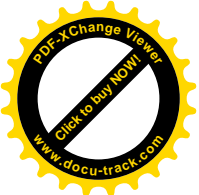
2. Установите указатель на нужный значок шаблона базы данных и дважды нажмите кнопку мыши.

3. Укажите имя и каталог для создаваемой базы данных.


4. Чтобы начать создание новой базы данных, нажмите кнопку **ОК**.

4.3.2. Создание базы данных без помощи мастера

1. При запуске Microsoft Access открывается диалоговое окно, в котором предлагается создать новую базу данных или открыть



существующую. Если появляется это диалоговое окно, выберите параметр Новая база данных, а затем нажмите кнопку ОК.

Если база данных уже была открыта или уже закрыто окно запуска, нажмите кнопку **Создать базу данных**  на панели инструментов и дважды нажмите кнопку мыши, установив указатель на значок новой базы данных.

2. Укажите имя и каталог базы данных и нажмите кнопку ОК.

После создания пустой базы данных необходимо самостоятельно создать объекты этой базы данных.

4.4. Создание таблицы

В Microsoft Access существует два способа создания таблицы. Для ввода собственных данных можно создать пустую таблицу. Можно также создать таблицу, используя уже существующие данные из другого источника.

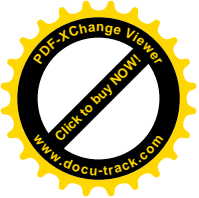
4.4.1. Создание новой пустой таблицы

В Microsoft Access существует четыре способа создания пустой таблицы.

1. Использование мастера баз данных для создания всей базы данных, содержащие все требуемые отчеты, таблицы и формы, за одну операцию. Мастера баз данных создает новую базу данных, его нельзя использовать для добавления новых таблиц, форм, отчетов в уже существующую базу данных.

2. Мастер таблиц позволяет выбрать поля для данной таблицы из множества определенных ранее таблиц, таких как деловые контакты, список личного имущества или рецепты.

3. Ввод данных непосредственно в пустую таблицу в режиме таблицы. При сохранении новой таблицы в Microsoft Access данные



анализируются и каждому полю присваивается необходимый тип данных формат.

4. Определение всех параметров макета таблицы в режиме конструктора.

Независимо от метода, примененного для создания таблицы, всегда имеется возможность использовать режим конструктора для дальнейшего изменения макета таблицы, например, для добавления новых полей, установки значений по умолчанию или для создания масок ввода.

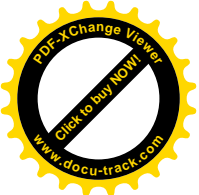
4.4.2. Создание таблицы при помощи мастера таблиц

1. Переключитесь в окно базы данных. Для переключения из другого окна в окно базы данных нажмите клавишу **F11**.
2. На вкладке **Таблица** нажмите кнопку **Создать**.
3. Дважды щелкните элемент «Мастер таблиц».
4. Следуйте инструкциям, выдаваемым в диалоговых окнах мастера таблиц.

✓ **Примечание.** В случае необходимости по окончании работы с мастером таблиц изменить или расширить полученную таблицу можно в режиме конструктора.

4.4.3. Создание таблиц путем ввода данных в таблицу

1. Переключитесь в окно базы данных. Для переключения из другого окна в окно базы данных нажмите клавишу **F11**.
2. На вкладке **Таблица** нажмите кнопку **Создать**.
3. Дважды щелкните элемент «Режим таблицы». На экране появится пустая таблица, состоящая из 20 столбцов и 30 строк. По умолчанию задаются следующие имена столбцов: «Поле1», «Поле2» и т. д.




4. Для переименования каждого столбца дважды щелкните название столбца, введите имя, следуя соглашениям об именах объектов Microsoft Access, и нажмите клавишу ENTER.

5. Если таблица должна содержать более 20 столбцов, то можно добавить дополнительные. Для этого нажмите кнопку мыши справа от столбца, рядом с которым необходимо разместить новый, и в меню **Вставка** выберите команду **Столбец**. Переименование столбцов описано в шаге 4.

6. Введите данные в таблицу.

➤ Вводите в каждый столбец данные определенного типа (в таблицах Microsoft Access каждый столбец называют полем). Например, вводите в один столбец фамилии, а в другой имена. При вводе чисел, значений дат и времени необходимо использовать согласованные форматы, чтобы Microsoft Access мог автоматически определить соответствующий тип данных и формат отображения столбца. Все столбцы с неизмененными именами, оставленные пустыми, будут удалены при сохранении таблицы.

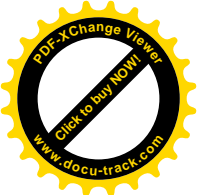
7. Закончив ввод данных в все нужные столбцы, нажмите кнопку **Сохранить**  на панели инструментов, чтобы сохранить таблицу.

8. При сохранении таблицы выводится приглашение создать ключевое поле. Если данные, которые могут однозначно идентифицировать каждую запись, такие как инвентарные номера или коды, еще не были введены, то рекомендуется нажать кнопку **Да**. Если данные, которые могут однозначно идентифицировать каждую запись, введены, то это поле можно определить как ключевое.

4.4.4. Быстрое создание таблицы в режиме конструктора


1. Переключитесь в окно базы данных. Для переключения из другого окна в окно базы данных нажмите клавишу **F11**.

2. На вкладке **Таблица** нажмите кнопку **Создать**.




3. Дважды щелкните элемент «Конструктор».
4. Определите в таблице каждое поле.
5. Определите ключевые поля до сохранения таблицы.

✓ **Примечание.** Рекомендуется определять ключевые поля, хотя это вовсе не обязательно. Если они не были определены, то при сохранении таблицы выдается вопрос, нужно ли их создавать.


6. Для сохранения таблицы нажмите кнопку **Сохранить**  на панели инструментов, а затем введите имя таблицы, следуя соглашениям об именах объектов Microsoft Access.

4.4.5. Добавление новых данных в режиме таблицы или формы

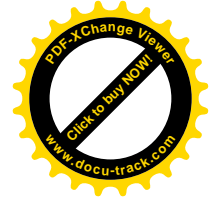
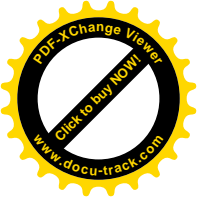
1. Откройте таблицу в режиме таблицы или форму в режиме формы.
2. Нажмите на панели инструментов кнопку **Новая запись** .
3. Введите данные и нажмите клавишу TAB для перехода к следующему полю.
4. В конце записи нажмите клавишу TAB для перехода к следующей записи.

4.4.6. Изменение данных в поле

1. Откройте таблицу в режиме таблицы или форму в режиме формы.
2. Чтобы изменять данные внутри поля, установите указатель в любое место в пределах этого поля и нажмите кнопку мыши.

Для выделения всего поля в режиме таблицы установите указатель на левый край поля. Когда указатель примет вид , нажмите кнопку мыши.

3. Введите текст с клавиатуры.



✓ **Примечания**

Если при вводе текста допущена ошибка, нажмите клавишу BACKSPACE. Для отмены изменений, внесенных в текущее поле, нажмите клавишу ESC. Для того чтобы отменить изменения, внесенные во всю запись, снова нажмите клавишу ESC перед выходом из поля.

➤ Microsoft Access сохраняет изменения в момент перехода к другой записи.

4.5. Объект: Запрос

Часто запросы в Microsoft Access создаются автоматически, и пользователю не приходится самостоятельно их создавать.

Для создания запроса, являющегося основой формы или отчета, попытайтесь использовать мастер форм или мастер отчетов. Они служат для создания форм и отчетов. Если отчет или форма основаны на нескольких таблицах, то с помощью мастера также создаются их базовые инструкции SQL. При желании инструкции SQL можно сохранить в качестве запроса.

Чтобы упростить создание запросов, которые можно выполнить независимо, либо использовать как базовые для нескольких форм или отчетов, пользуйтесь мастерами запросов. Мастера запросов автоматически выполняют основные действия в зависимости от ответов пользователя на поставленные вопросы. Если было создано несколько запросов, мастера можно также использовать для быстрого создания структуры запроса. Затем для его наладки переключитесь в режим конструктора.

Для создания запросов на основе обычного фильтра, фильтра по выделенному фрагменту или фильтра для поля, сохраните фильтр как запрос.

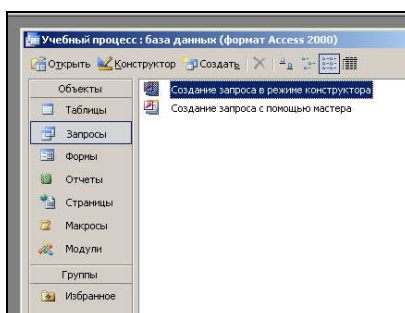
Если ни один из перечисленных методов не удовлетворяет требованиям, создайте самостоятельно запрос в режиме конструктора.

Запрос – объект базы данных, позволяющий манипулировать данными базовых таблиц. Запрос позволяет удалять, добавлять, обновлять записи,

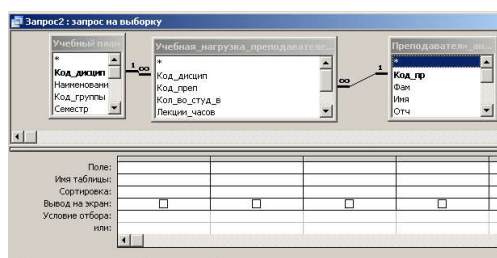
производить выборку данных, производить вычисления на основе данных таблиц. При запуске, запрос создает виртуальную таблицу, в которой отображается результат выполнения запроса.

Источников данных для запроса могут сложить таблицы и другие запросы.

Для создания запроса нужно в окне БД, на вкладке ЗАПРОСЫ, активизировать команду Создание запроса в режиме конструктора.



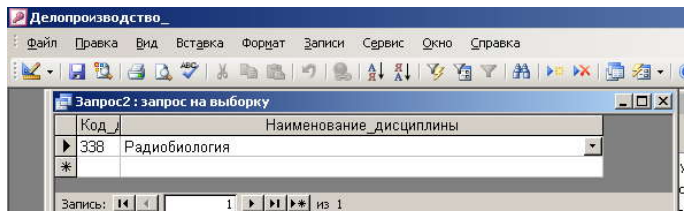
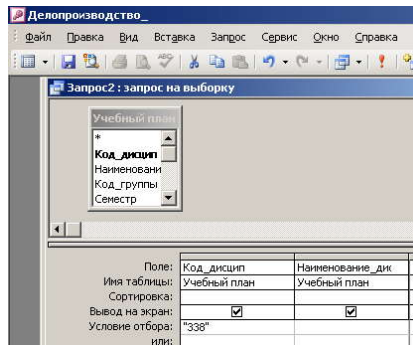
В верхней части окна конструктора запросов нужно добавить необходимые объекты-источники данных, а в нижней части – бланке запроса, в строке условие отбора, сформировать логическое выражение для выборки данных.



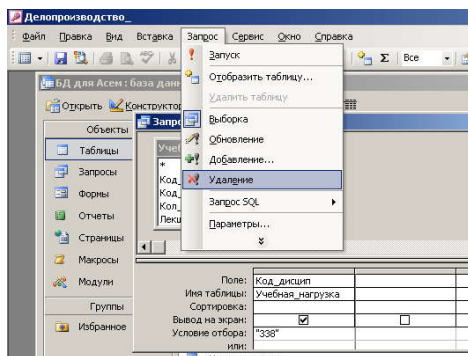
4.5.1. Виды запросов

1. Запрос на удаление. Данный вид запроса позволяет удалить записи из таблицы.

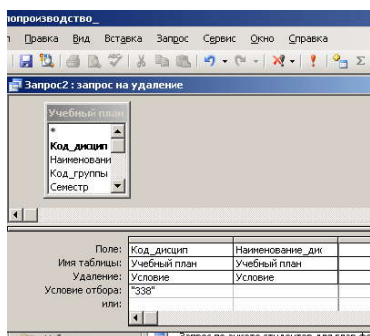
Для создания запроса нужно предварительно создать запрос на выборку записей подлежащих удалению.



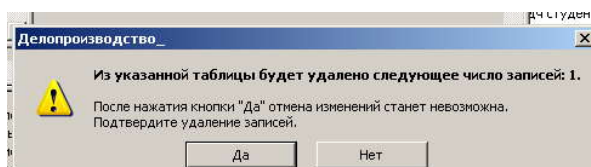
Затем преобразовать его в запрос на выборку, выполнив команду
Запрос-Удаление



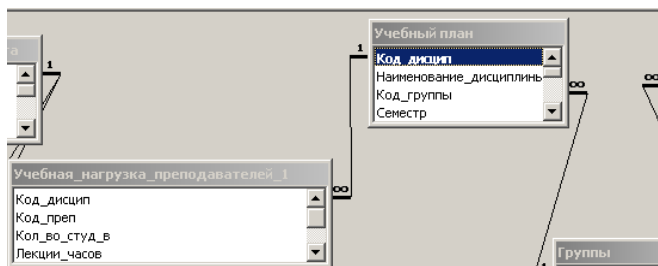
В результате в бланке запроса появится строка УДАЛЕНИЕ.



При запуске запроса на удаление на экране появится информационное
окно, сообщаемое о количестве удаляемых записей

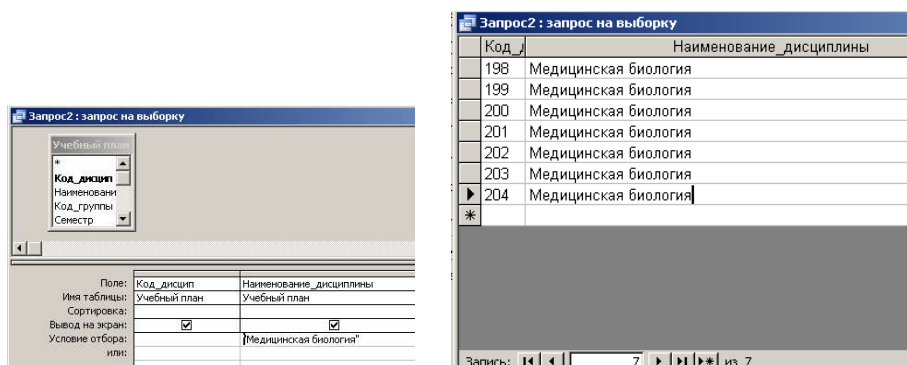


После удаления записей в таблице будут удалены все связанные записи в подчиненных таблицах

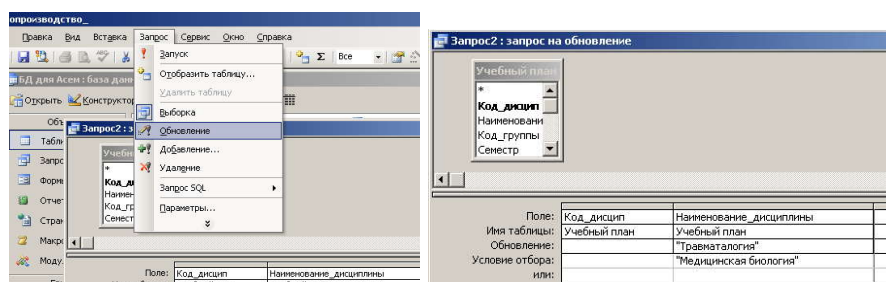


2. **Запрос на обновление.** Данный вид запроса позволяет изменить одно значение полей в выбранных записях на другое.

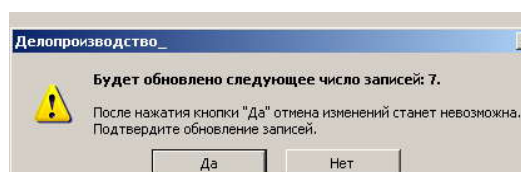
Предварительно нужно создать запрос на выборку для отбора записей, в которых нужно обновить значение поля.



Затем запрос преобразовать в запрос на обновление и в появившейся в бланке запроса строке **ОБНОВЛЕНИЕ** ввести новое значение поля



При запуске запроса на обновление на экране появится информационное окно, сообщаемое о количестве обновляемых записей.



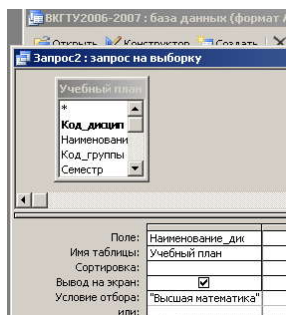
После удаления записей в таблице будут удалены все связанные записи в подчиненных таблицах

3. **Запрос на добавление.** Данный вид запроса предназначен для добавления записей из одной таблицы-источника в другую таблицу-приемник.

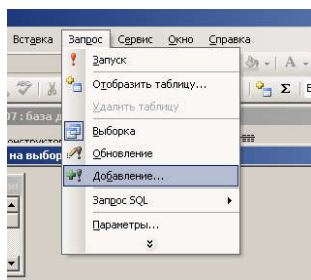
Имя	Размер	Тип	Изменен
ВКГТУ2006-2007.mdb	11 736 КБ	Приложение Micros...	15.11.2007 7:29
ВКГТУ 2007-2008.mdb	11 736 КБ	Приложение Micros...	15.11.2007 7:29
Папка 2.doc	1 300 КБ	Документ Microsoft...	15.11.2007 7:31

Например, существуют два файла с данными об учебном процессе за два учебных года. Требуется записи по дисциплине «Высшая математика» из таблицы «Учебный план» файла за 2006-2007 добавить в таблицу с таким же именем в файле с данными за 2007-2008 у.г.

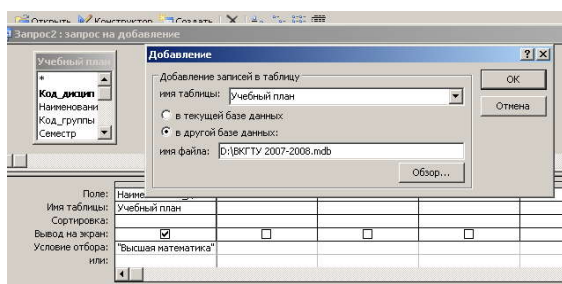
Для создания запроса на добавление нужно предварительно на основе таблицы-источника создать запрос на выборку записей, которые нужно добавить в таблицу-приемник.



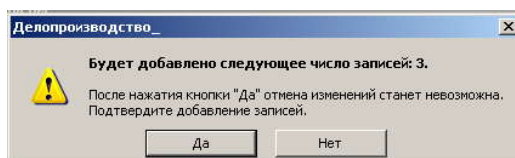
Затем преобразовать его в запрос на добавление



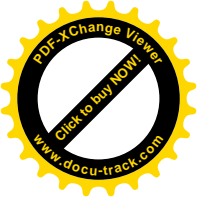
В результате появится диалоговое окно, запрашивающее имя таблицы-приемника. При необходимости нужно указать файл-приемник.



При запуске запроса на добавление, на экране появится информационное окно, сообщающее о количестве добавляемых записей.



4. Запрос на выборку. Данный вид запроса предназначен для выборки записей по определенному условию. Логическое выражение, являющееся критерием отбора записей записывается в строке Условие отбора бланка



запроса. При формировании условия отбора можно использовать следующие логические операторы:

1. Between ... and – позволяет произвести выборку записей, значения поля в которых принадлежат указанному диапазону.

Between 10 and 200

Between #10.05.2005# and #30.06.2005#

2. Not – отрицание

Not Between 10 and 200

4. Like – позволяет проверить соответствие значения поля указанному шаблону.

Специальные символы, используемые с оператором Like

СИМВОЛ	ОПИСАНИЕ
?	- любой одиночный символ
*	- любое количество символов
#	- любая цифра (от 0 до 9)
[A-P]	-любой символ, попадающий в указанный диапазон
[!A-P]	-любой символ, не попадающий в указанный диапазон
[A,Д,Ш]	- любой символ из указанного списка
[!A,Д,Ш]	- любой символ, не попадающий в указанный список

Примеры:

Like “a*a”

Like “[A-P]”

Like “[!A-P]”

Like “a#a”

Like “a[K]#[!к-т]a”

Like “B?T*”

Like "[А,Д,Ш]"

Like "![А,Д,Ш]"

Поле:	Фам	Лекции_часов	
Имя таблицы:	Преподаватели_а	Учебная_нагрузка	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:	Like "К*"	Between 15 And 40	
или:	Like "О*"		

5. Запрос с вычисляемым полем. Данный вид запроса позволяет произвести какие-либо вычисления на основе числовых данных объектов-источников. Также в вычисляемом поле можно сформировать комбинацию значений нескольких полей. Вычисляемое выражение можно сформировать вручную или с помощью построителя запросов.

Примеры вычисляемых полей:

[Преподаватели_анкета]![Фам] & " " &
 Left([Преподаватели_анкета]![Имя];1) & "." &
 Left([Преподаватели_анкета]![Отч];1) & ". "
 Иф ([Товар]![Количество]>500; [Товар]![Стоимость] *0.8;
 [Товар]![Стоимость])
 [Товар]![Стоимость] * [Товар]![Количество]

Запрос1 : запрос на выборку

Товар

Наименование т
Стоимость
Количество

Поле: Наименование тов Сумма: [Товар][Стоимость]*[Товар][Количество]
Имя таблицы: Товар
Сортировка: ☒
Вывод на экран: ☒
Условие отбора:
или:

6. Запрос с групповыми операциями. Данный запрос позволяет вычислить итоги по числовым полям для групп записей. В бланк конструктора запроса нужно включать только следующие поля: поле по которому группируются записи и числовые поля по которым для сгруппированных записей будут вычислены итоги.

Запрос5 : запрос на выборку

Группы

Курс
Код_спец
Год_поступ
Год_оконч

Специальности

Код_спец
Шифр
Наименование

Поле: Наименование_спе Кол-во_студ
Имя таблицы: Специальности Группы
Групповая операция: Групповая
Сортировка: ☒
Вывод на экран: ☒
Условие отбора:
или:

Запрос5 : запрос на выборку

Наименование_спец	Sum-Кол-во
Валеология	41
Медико-биологическое дело	144

Запись: 2 из 2

Запрос5 : запрос на выборку

Наименование_спец	Кол-во_студ
Валеология	7
Валеология	6
Валеология	11
Валеология	17
Медико-биологическое дело	13
Медико-биологическое дело	23
Медико-биологическое дело	20
Медико-биологическое дело	7
Медико-биологическое дело	26
Медико-биологическое дело	15
Медико-биологическое дело	10
Медико-биологическое дело	15
Медико-биологическое дело	15

Запись: 1 из 13

7. Параметрический запрос. Параметрический запрос при запуске запрашивает параметры условия отбора в диалоговом режиме. Для создания параметрического отбора запрос для диалога нужно задать в квадратных скобках в строке условия отбора.

Код	Код группы	Фамилия	Имя	Отчество	Год_рожд	Адрес_ро	Адрес_жительства
001	B-03-A	Бобров	Анастасия	Сергеевн	15.07.1986	Шемон-й р.г. У-Ка у. Образцовая 7-208	
002	B-03-A	Дакиева	Лейла	Болатовн	11.06.1984	г. Шемонаи г. У-Ка ул. 30-гв див. 40-118	
003	B-03-A	Крючков	Наталья	Игоревн	23.05.1985	Шемон-й р.г. У-Ка у. Образцовая 7-431	
004	B-03-A	Порожня	Анастасия	Владими	04.07.1986	г. Риддер у.г. У-Ка у. Образцовая 7-208	
005	B-03-A	Семенцо	Ирина	Александр	10.03.1986	К-Караг-й г. У-Ка у. Образцовая 7-208	
006	B-03-A	Толегенс	Дина	Толегено	02.05.1984	Улан-й р-н г. У-Ка у. Космическая 4/1-304	
007	B-03-A	Чешко	Надежда	Владими	09.02.1985	Глуб-й р-н г. У-Ка пр. Ленина 12/2-28	
008	B-03-B	Аканова	Назгул	Мараткы	22.10.1985	Тарбаг-й р.г. У-Ка у. Комсомольская 16-93	
009	B-03-B	Кусяиной	Айнул	Айтказие	10.03.1986	К-Караг-й г. У-Ка ул. 30-гв див. 40-98	
010	B-03-B	Мананов	Салтанат	Болаткыс	07.10.1985	г. Зайсан у.г. У-Ка ул. 30-гв див. 40-98	
011	B-03-B	Момкенс	Аулым	Сериковн	01.08.1986	Тарбаг-й р.г. У-Ка у. Образцовая 7-434	
012	B-03-B	Мухамад	Гулим	Сайлауби	24.09.1985	Тарбаг-й р.г. У-Ка ул. 30-гв див. 38-287	
013	B-03-B	Сыдыков	Эльмира	Дуйсенон	03.12.1985	Зайсанск.г. У-Ка у. Краснознаменная 28А	
014	B-03-B	Абильба	Лаура	Нурбеко	26.11.1984	Тарбаг-й р.г. У-Ка пр. Текстильщиков 12/2-67	
015	B-03-B	Бектуров	Алмагуль	Толегено	04.02.1984	К-Караг-й г. У-Ка ул. 30-гв див. 38-294	
016	B-03-B	Жекебае	Меруерт	Бакытбей	14.01.1984	Марк-й р-н г. У-Ка ул. 30-гв див. 38-294	
017	B-03-B	Калдыба	Мадина	Фаизрах	22.04.1985	Улан-й р-н г. У-Ка ул. 30-гв див. 40-156	
018	B-03-B	Калиаск	Назым	Айтхмие	01.12.1983	г. Зайсан у.г. У-Ка у. Серикбаева 23-118	

Запрос5 : запрос на выборку

Студент_анк

Код_группы
Фамилия
Имя
Отчество
Год_рожд

Поле: Код_группы
Имя таблицы: Студент_анкета

Сопоставка: Студент_анкета

Выходные поля: Студент_анкета

Условие отбора: (Введите номер группы)

Запрос5 : запрос на выборку

Код_группы	Фамилия	Имя	Отчество	Год_рожд
B-03-A	Дакиева	Лейла	Болатовна	11.06.1984
B-03-A	Крючков	Наталья	Игоревна	23.05.1985
B-03-A	Порожня	Анастасия	Владимировн	04.07.1986
B-03-A	Семенцо	Ирина	Александровн	10.03.1986
B-03-A	Толегенс	Дина	Толегеновна	02.05.1984
B-03-A	Чешко	Надежда	Владимировн	09.02.1985
B-03-A	Бобров	Анастасия	Сергеевна	15.07.1986

Введите значение параметра

Введите номер группы

B-03-A

OK Отмена

8. Перекрестный запрос. Данный запрос представляет собой аналитическую таблицу, в которой в качестве заголовков строк используются значения одного поля таблицы, в качестве заголовков столбцов – значения другого поля таблицы, а значения третьего поля (числового) отображаются в ячейках таблицы.

Перекрестный запрос создается с помощью мастера.

Шаг 1: Запрос источника данных

Создание перекрестных таблиц

Выберите таблицу или запрос, поля которых необходимо вывести в перекрестном запросе.

Для включения полей из нескольких таблиц сначала создайте обычный запрос, содержащий все необходимые поля.

Таблица: Вид ППС - ВЕР
Таблица: Вид соц-го статуса
Таблица: Виды родовых отношений
Таблица: виды семестров
Таблица: Группы
Таблица: Дисциплины
Таблица: должности в студенческой группе
Таблица: Должности кафедры

Показать
☒ Таблицы ☐ Запросы ☐ Таблицы и запросы

Образец:

	Заголовок1	Заголовок2	Заголовок3
ИТОГИ			

Отмена < Назад Далее > Готово

Шаг 2: Запрос имени поля, значения которого будут использоваться в качестве заголовков строк

Создание перекрестных таблиц

Выберите поля, значения которых будут использованы в качестве заголовков строк.

Допускается выбор не более трех полей.

Выберите поля по порядку сортировки данных. Например, можно сначала выполнить сортировку значений по странам, а затем по городам.

Доступные поля:

- Код_группы
- Наименование_дисциплин
- Всего часов

Выбранные поля:

Образец:

	Заголовок1	Заголовок2	Заголовок3
ИТОГИ			

Отмена < Назад Далее > Готово

Шаг 3: Запрос имени поля, значения которого будут использоваться в качестве заголовков столбцов

Создание перекрестных таблиц

Выберите поля для использования их значений в качестве заголовков столбцов.

Например, чтобы использовать имя каждого сотрудника в качестве заголовка столбца, выберите поле ИмяСотрудника.

Доступные поля:

- Наименование_дисциплины
- Всего часов

Образец:

Код_группы	Наименование_дисциплины	Наименование_1	Наименование_2	Наименование_3
Код_группы1	ИТОГИ			
Код_группы2				
Код_группы3				
Код_группы4				

Отмена < Назад Далее > Готово

Шаг 4: Запрос имени числового поля, значения которого будут отображены в ячейках таблицы и имя функции, с помощью которой будут обработаны эти данные.

Создание перекрестных таблиц

Какие вычисления необходимо провести для каждой ячейки на пересечении строк и столбцов?

Например, можно вычислить сумму заказов для каждого сотрудника (столбец) по странам и регионам (строка).

Вычислить итоговое значение для каждой строки?
☒ Да.

Поля:

Всего часов

Функции:

Дисперсия
Максимум
Минимум
Отклонение
Первый
Последний
Среднее
Сумма
Число

Образец:

Код_группы	Наименован	Наименован	Наименован
Код_группы1	Дисперсия(Всего часов)		
Код_группы2			
Код_группы3			
Код_группы4			

Отмена < Назад Далее > Готово

Шаг 5: Запрашивает имя запроса

Создание перекрестных таблиц

Задайте имя запроса.

Общее количество часов_перекрестный

Указаны все сведения, необходимые для создания запроса с помощью мастера.

Дальнейшие действия после создания запроса:

☒ Просмотреть результаты запроса.
☐ Применить структуру запроса.

☐ Вывести справку по работе с перекрестной таблицей.

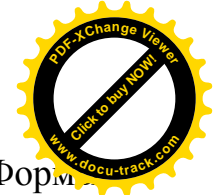
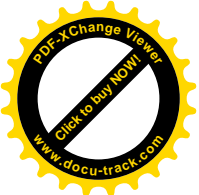
Отмена < Назад Далее > Готово

РЕЗУЛЬТАТ

Общее количество часов_перекрестный : перекрестный запрос

Код_группы	Итоговое значе	Адаптация	Биомеханика	БИР	Генетика повед	Микробиология	Радиобиология	Цитология
B-02-B	339			82			176	81
B-03-A	260		63	72	18	44		63
B-03-B	350	90	63	72	18	44		63

Запись: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445



Форма является основным объектом интерфейса базы данных. Формы позволяют организовать доступ ко всем остальным объектам базы данных – таблицам, запросам, другим формам, отчетам и т.д.

Для создания простой формы, содержащей один столбец, используется кнопка **Новый объект**.

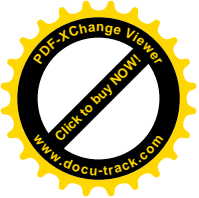
Формы позволяют:

- в удобном и наглядном виде отображать данные, хранящиеся в таблицах текущей и внешних баз данных;
- добавлять, удалять или модифицировать информацию в базовых таблицах;
- открывать для просмотра отчеты, другие формы или страницы доступа к ним;
- использоваться в качестве автоматически запускаемого элемента при открытии базы данных;
- запускать на выполнение макросы и функции;

Формы в зависимости от их функциональности можно классифицировать на три основные группы:

- Формы, предназначенные для манипулирования данными таблиц (отображение, добавление, удаление и модификация информации);
- Командные формы, служащие аналогом меню. В формах этой группы обычно расположены элементы управления, при помощи которых можно открывать другие формы, отчеты и т.д. или запускать на выполнение макросы и программы VB.
- Пользовательские диалоговые окна;

Форма представляет собой служебное окно MS Access, содержащее различные элементы, называемые элементами управления. Этими элементами могут быть стандартные элементы управления: кнопки, флажки, переключатели, метки, подчиненные формы и т.д.



Элементы управления в зависимости от их роли и предназначения условно можно разделить на группы:

- Различные графические элементы, используемые для оформления формы;
- Элементы управления, связанные с соответствующими полями базовых таблиц. Данные элементы управления, предназначены для отображения и модификации информации из набора, являющегося источником данных;
- Вычисляемые элементы управления, предназначенные для отображения различных итоговых, комбинированных или других вычисляемых данных;
- Элементы описательного характера, предназначенные для пояснения назначения других элементов;

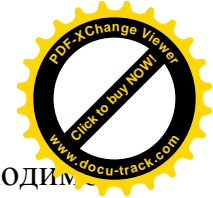
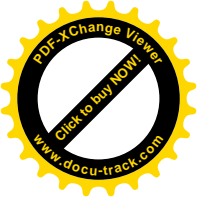
Формы можно создавать одним из следующих способов:

- Конструктор;
- Мастер форм;
- Автоформа: в столбец;
- Автоформа: ленточная;
- Автоформа: табличная;
- Автоформа: сводная таблица;
- Автоформа: сводная диаграмма;
- Диаграмма;
- Сводная таблица;

4.6.1. Создание формы с помощью автоформы

Автоформа создает форму, в которой отображаются все поля и записи выбранной таблицы или запроса. Каждое поле расположено на отдельной строке, с левой стороны от которой отображается надпись к данному полю.

1. В окне базы данных выберите вкладку ***Таблицы*** или ***Запросы***.



2. Выберите таблицу или запрос, на основе которых необходимо создать форму, или откройте их в любом режиме.

3. Нажмите кнопку раскрытия списка рядом с кнопкой **Новый объект** на панели инструментов и выберите элемент **Автоформа**.

4.6.2. Создание формы с помощью мастера

1. В окне базы данных выберите вкладку **Формы**.

2. Нажмите кнопку **Создать**.

3. В диалоговом окне **Новая форма** выберите нужного мастера.

Описание мастера появляется в левой части диалогового окна.

4. Выберите имя таблицы или запроса, содержащих данные, на основе которых будет создана форма.

✓ **Примечание.** При использовании мастера форм источник данных для формы следует указывать в диалоговом окне мастера.

5. Нажмите кнопку **ОК**.

6. Если на шаге 3 были выбраны **Мастер форм**, **Диаграмма** или **Сводная таблица**, то при создании формы следуйте инструкциям, выводимым в диалоговых окнах соответствующего мастера. При выборе элементов **Автоформа: в столбец**, **Автоформа: ленточная** или **Автоформа: табличная** форма создается автоматически.

➤ Изменить созданную форму можно в режиме конструктора.

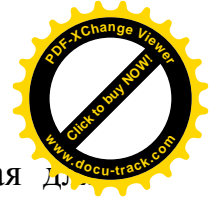
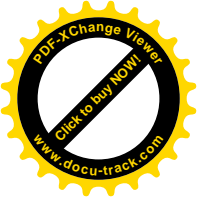
4.6.3. Создание формы без помощи мастера

1. В окне базы данных выберите вкладку **Формы**.

2. Нажмите кнопку **Создать**.

3. В диалоговом окне **Новая форма** выберите пункт **Конструктор**.

4. Выберите имя таблицы или запроса, содержащих данные, на основе которых будет создана форма. Если форма не будет содержать



данных (например, если форма будет использоваться как кнопочная для открытия других форм или отчетов или как специальное диалоговое окно), ничего не выбирайте в списке.

✓ **Примечание.** Форма, использующая данные из нескольких таблиц, должна быть основана на запросе, включающем эти таблицы.

5. Нажмите кнопку **ОК**.

4.6.4. Создание формы или отчета на основе записей, отобранных с помощью фильтра

1. Выведите на экран записи, отобранные с помощью фильтра, и нажмите кнопку раскрытия списка, расположенную рядом с кнопкой **Новый объект** на панели инструментов.

➤ Для автоматического создания стандартной формы или отчета с помощью мастера выберите команду **Автоформа** или **Автоотчет**.

➤ Для того чтобы указать тип создаваемой формы или отчета, выберите пункт Форма или Отчет, а затем выберите в списке мастера или режим.

Источником записей для нового объекта служит таблица или запрос, в которых был создан фильтр. Объект наследует этот фильтр.

2. Завершите разработку макета или внесите другие необходимые изменения в форму или отчет, а затем сохраните их.

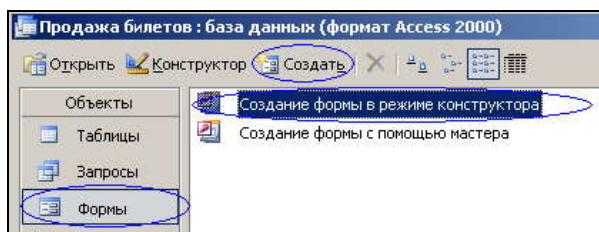
✓ **Примечания:**

В новой форме или отчете будет применен порядок сортировки, указанный в базовой таблице или запросе.

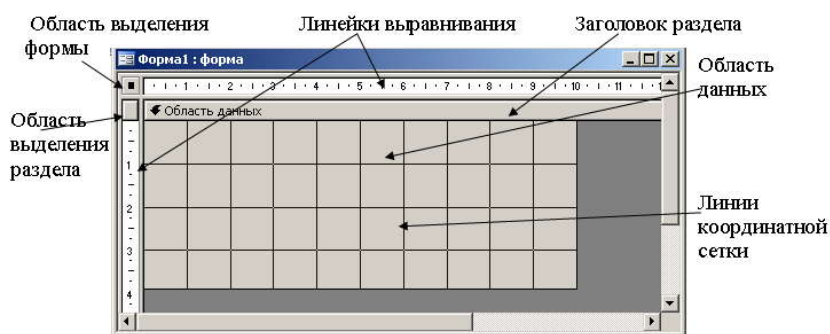
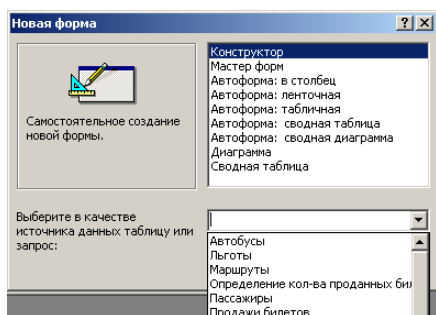
Последующее изменение фильтра в таблице или запросе не влияет на форму или отчет, созданные с помощью ранее заданного фильтра.

4.6.5. Режим конструктор

Для создания формы в режиме конструктора нужно в окне БД перейти на вкладку *Формы* и выбрать *Создание формы в режиме конструктора* или нажать кнопку *Создать*.



При втором способе на экране появится окно *Новая форма* в которой нужно выбрать способ создания формы и источник данных для формы.



При выборе режима Конструктор на экране появится окно с макетом формы

Область выделения формы (Form Selector). В каждый момент времени активным может быть только один элемент формы. Подавляющее большинство операций касается отдельного элемента. В случае, когда нужно настроить саму форму, нужно ее выделить. Для этого необходимо щелкнуть в области выделения формы. Признаком того, что в данный момент является

активной именно форма, служит черный квадрат в области выделения формы.



Область выделения раздела (Section Selector). Служит для выделения раздела формы.

Заголовок раздела (Section Header). Служит для обозначения начала раздела.

Горизонтальная и вертикальная линия прокрутки (Ruler). Используются для выравнивания элементов формы относительно друг друга.

Область данных (Detail). Один из разделов формы.

Линии координатной сетки (Grid). Служат для точного выравнивания элементов управления.

4.6.6. Разделы формы

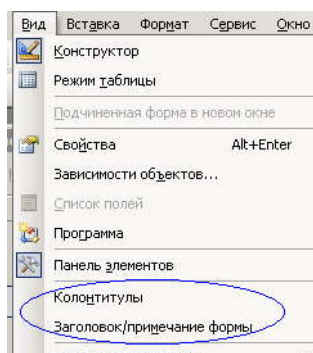
Форма может содержать до пяти разных разделов:

Заголовок формы (Form Header). В заголовке располагаются сведения, общие для всех записей. Заголовок отображается в верхней части окна в режиме формы и в начале первой страницы при печати.

Верхний колонтитул (Page Header). Здесь содержатся сведения, используемые при печати формы. Информация из верхнего колонтитула располагается в начале каждой страницы. В режиме формы верхний колонтитул не отображается.

Область данных (Detail). Это обязательный раздел формы, в котором располагается вся основная информация: значения полей из источника данных формы (таблицы или запроса), значения вычисляемых полей и др.

Нижний колонтитул (Page Footer). Аналогичен верхнему колонтитулу.



Примечание формы (Form Footer). Здесь содержатся данные, являющиеся общими для всех записей.

Новая форма, созданная в режиме конструктора обычно содержит только область данных. В случае необходимости остальные разделы можно добавить с помощью команды Вид/Заголовок/Примечание и Вид/Колонтитулы.

4.6.7. Панель инструментов управления

В процессе создания формы необходимо на макете формы расположить элементы управления и произвести их настройку. Это производится с помощью панели элементов управления



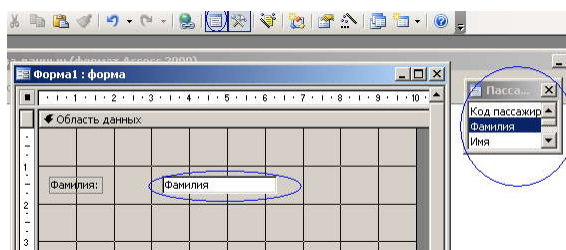
Выбор объектов - при выборе элемента управления, данная кнопка становится отжатой. Снова нажав на данной кнопке можно отменить выбор.

Мастера – Если эта кнопка активна, то выборе какого – либо элемента, автоматически запускается соответствующий мастер, позволяющий настроить этот элемент.

Элементы :

- Надпись;
- Поле (связанное и свободное)
- Группа переключателей;
- Переключатель;
- Флажок;

- Рисунок; - отображает содержимое графического файла
- Свободная рамка объекта;
- Присоединенная рамка объекта;
- Подчиненная форма/отчет;
- Линия;
- Прямоугольник;
- Набор вкладок;



4.7. Объект: Отчет

Пользователь имеет возможность разработать отчет самостоятельно или создать отчет с помощью мастера. Мастер по разработке отчетов Microsoft Access выполняет всю рутинную работу и позволяет быстро разработать отчет. После вызова мастера выводятся диалоговые окна с приглашением ввести необходимые данные, и отчет создается на основании ответов пользователя. Мастер окажется полезным даже для опытных пользователей, так как позволяет быстро разработать макет, служащий основой создаваемого отчета. После этого можно переключиться в режим конструктора и внести изменения в стандартный макет.

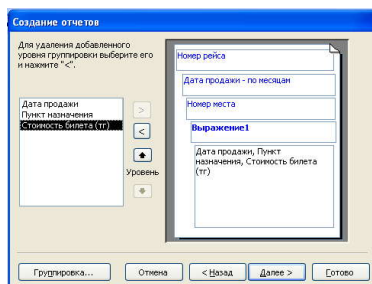
Существуют различные режимы создания отчетов: конструктор, мастера, автоотчет, мастер диаграмм, почтовые наклейки.

4.7.1. Режим мастера

1. В окне базы данных выберите вкладку **Отчеты**.

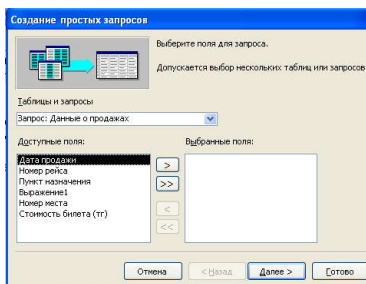
2. Нажмите кнопку **Создать**.
 3. В диалоговом окне **Новый отчет** выберите нужного мастера.
Описание действий, выполняемых мастером, выводится в левой половине диалогового окна.
 4. Выберите имя таблицы или запроса, содержащих данные, по которым строится отчет.
- ✓ **Примечание.** Microsoft Access по умолчанию использует эту таблицу или запрос как базовый источник данных для отчета. Однако мастер позволяет изменить источник данных, а также выбрать поля из других таблиц или запросов.
5. Нажмите кнопку **ОК**.
 6. Если на шаге 3 выбран мастер отчетов, мастер диаграмм или мастер наклеек, выполняйте инструкции мастера, выводящиеся в диалоговом окне. Если выбран один из мастеров автоотчетов, отчет создается автоматически.

Шаг1: Выбор источника
данных, и полей



Шаг3: Выбор полей для
сортировки записей

Шаг2: Добавление уровней
группировки



Шаг4: Выбор статистических
функций

Итоги

Какие итоговые значения необходимо вычислять?

Поле	Sum	Avg	Min	Max
Стоимость билета (tz)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Показать

☒ данные и итоги

☐ только итоги

☐ Вычислять проценты

ОК Отмена

Создание отчетов

Выберите порядок сортировки и вычисления, выполняемые для записей.

Допускается сортировка записей по возрастанию или по убыванию, включающая до 4 полей.

1. Дата продажи по возрастанию
2. Пункт назначения по возрастанию
3. по возрастанию
4. по возрастанию

Итоги...

Отмена < Назад Далее > Готово

Шаг5: Выбор вида макета для отчета

Шаг6: Выбор стиля

Создание отчетов

Выберите вид макета для отчета.

Макет

☐ ступенчатый

☐ блок

☐ структура 1

☐ структура 2

☒ по левому краю 1

☐ по левому краю 2

Ориентация

☒ фронтальная

☐ альбомная

☒ Настроить ширину полей для размещения на одной странице.

Отмена < Назад Далее > Готово

Создание отчетов

Выберите требуемый стиль.

Деловой

Обычный

Полужирный

Сжатый

Спокойный

Строгий

Отмена < Назад Далее > Готово

Шаг7: Выбор имени для отчета

Создание отчетов

Задайте имя отчета:

Продажи билетов

Указаны все сведения, необходимые для создания отчета с помощью мастера. Дальнейшие действия:

☒ Просмотреть отчет.

☐ Изменить макет отчета.

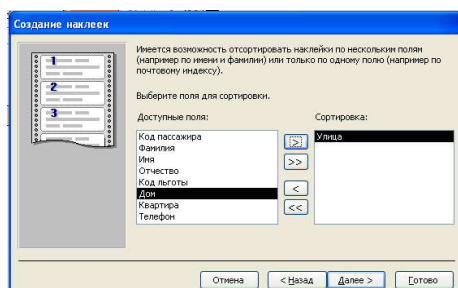
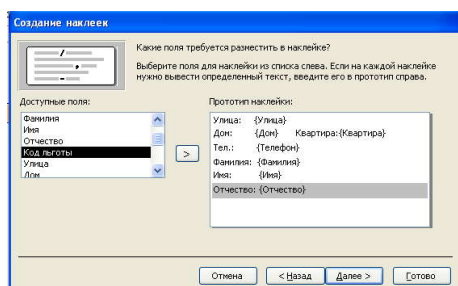
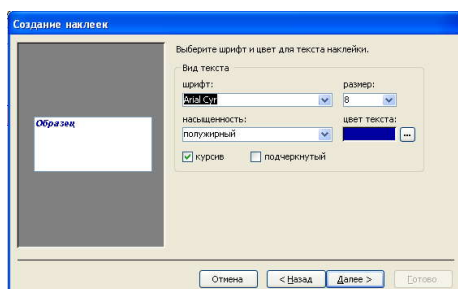
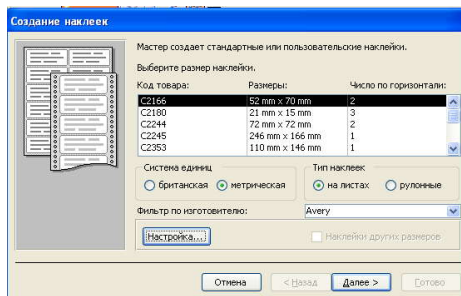
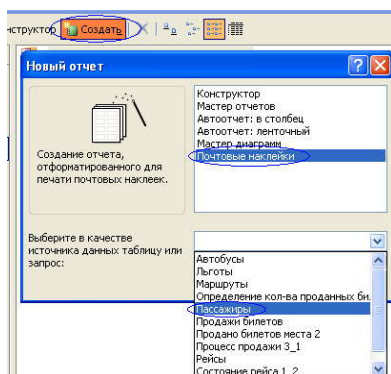
☐ Вывести справку по работе с отчетом?

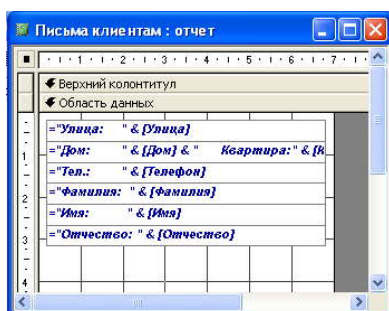
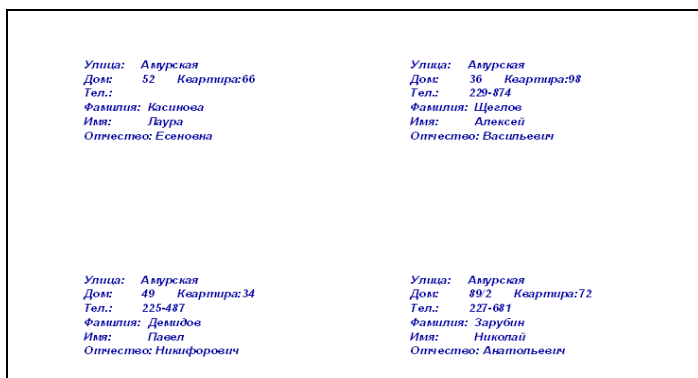
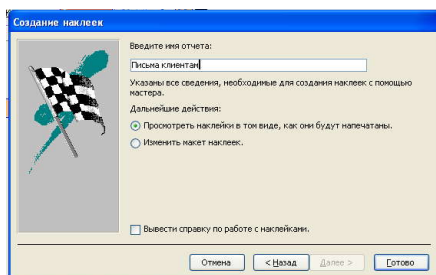
Отмена < Назад Далее > Готово

Продажи билетов								
Номер рейса		01						
Дата продажи - по		Сентябрь 2008						
Номер места		1						
Выражение 1		Касенова Г.С.						
<table><tr><th><u>Дата продажи</u></th><th><u>Пункт назначения</u></th><th><u>Стоимость билета (tz)</u></th></tr><tr><td>17.09.2008</td><td>Семей</td><td>3 640,00р.</td></tr></table>			<u>Дата продажи</u>	<u>Пункт назначения</u>	<u>Стоимость билета (tz)</u>	17.09.2008	Семей	3 640,00р.
<u>Дата продажи</u>	<u>Пункт назначения</u>	<u>Стоимость билета (tz)</u>						
17.09.2008	Семей	3 640,00р.						
Итого для "Выражение 1" = Касенова Г.С. (1 запись)								
Sum		3 640,00р.						
Итого для "Номер места" = 1 (1 запись)								
Sum		3 640,00р.						
Номер места		2						
Выражение 1		Самыхова О.П.						
<table><tr><th><u>Дата продажи</u></th><th><u>Пункт назначения</u></th><th><u>Стоимость билета (tz)</u></th></tr><tr><td>17.09.2008</td><td>Семей</td><td>3 120,00р.</td></tr></table>			<u>Дата продажи</u>	<u>Пункт назначения</u>	<u>Стоимость билета (tz)</u>	17.09.2008	Семей	3 120,00р.
<u>Дата продажи</u>	<u>Пункт назначения</u>	<u>Стоимость билета (tz)</u>						
17.09.2008	Семей	3 120,00р.						
Итого для "Выражение 1" = Самыхова О.П. (1 запись)								
Sum		3 120,00р.						

4.7.2. Создание почтовых наклеек

Почтовые наклейки – специальный вид многостолбцового отчета, предназначенные для печати на одной странице, нескольких групп информации.



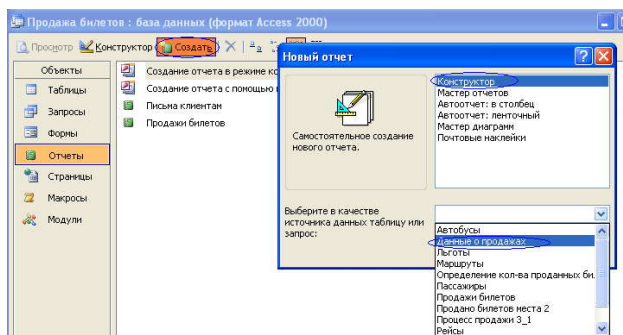


4.7.3. Режим конструктора

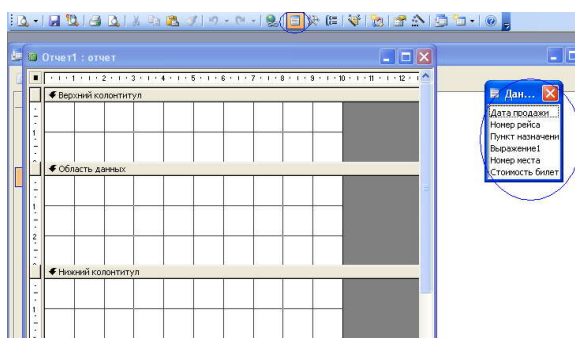
1. В окне базы данных выберите вкладку **Отчеты**.
2. Нажмите кнопку **Создать**.
3. В диалоговом окне **Новый отчет** выберите пункт **Конструктор**.
4. Выберите имя таблицы или запроса, содержащих данные, по которым строится отчет. (Если требуется создать свободный отчет, не выбирайте имя из списка).
 - Для того чтобы использовать в отчете данные из нескольких таблиц, создавайте отчет на основе запроса.


5. Нажмите кнопку **ОК**.

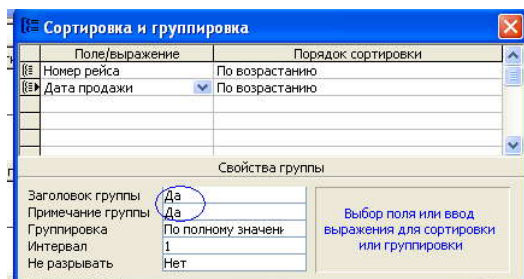
Для создания отчета в режиме конструктора нужно на вкладке Отчеты окна БД выбрать соответствующий режим.



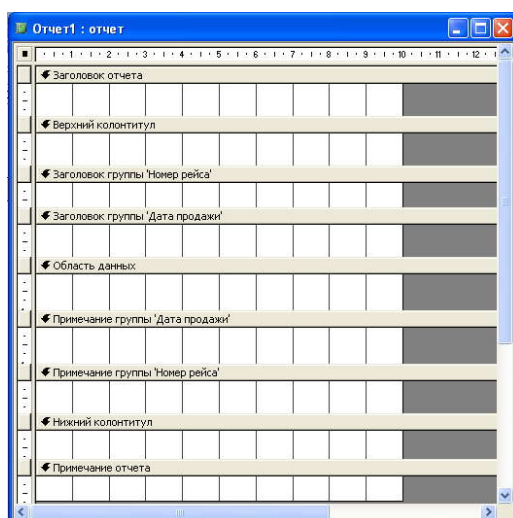
Окно конструктора отчета имеет такую же структуру, что и конструктор форм.



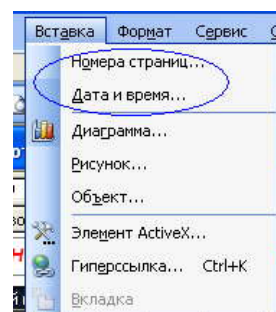
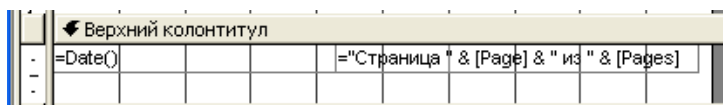
Кроме разделов область данных, заголовок, примечание, верхний и нижний колонтитулы, в макет отчета можно также включить также разделы заголовок и примечание группы. Данные разделы следует включать, если записи таблиц БД в отчете будут иметь иерархическую структуру. Для группировки записей следует выполнить команду Вид/Сортировка и группировка или нажать на кнопке . В появившемся окне выбрать нужные поля в зависимости от планируемой иерархии. Если необходимо, для каждой группы записей, в макет отчета включить заголовки и примечания групп.



Вид макета отчета после включения новых разделов



В колонтитулы номера страниц, текущую дату можно вставить используя команду Вставка /Номера страниц..., / Дата и время...



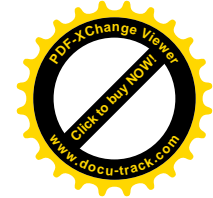
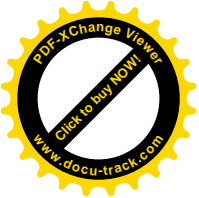
Для автоматической нумерации записей нужно настроить свободное поле

№	Макет	Данные	События	Другие	Все
Область	Данные	...	=1		
Примечание	Маска ввода	...			
	Сумма с накоплением	...			
	Смарт-теги	...			
			Отсутствует		
			Отсутствует		
			Для группы		
			Для всего		

Заголовок отчета			
Данные о продажах билетов на рейсы			
Верхний колонтитул			
=Date()		="Страница " & [Page] & " из " & [Pages]	
Заголовок группы 'Номер рейса'			
Номер рейса:		Номер рейса	
Заголовок группы 'Дата продажи'			
Дата продажи:		Дата продажи	
№	Фамилия И.О.	Номер места	Стоимость билета (тг.)
Область данных			
=1	Выражение1	Номер места	Стоимость билета (тг.)
Примечание группы 'Дата продажи'			
Общая сумма (тг.):		ки билетов][Стоимость	
Примечание группы 'Номер рейса'			
Общая сумма (тг.):		ки билетов][Стоимость	
Нижний колонтитул			
ТОО "Автостанция"			
Примечание отчета			
Общая сумма (тг.):		ки билетов][Стоимость	

Отчет с данными о продажах

Данные о продажах билетов на рейсы			
30.09.2008		Страница 1 из 1	
Номер рейса: 01			
Дата продажи: 16.09.2008			
№	Фамилия И.О.	Номер места	Стоимость билета (тг)
	Оспанов Е.М.	7	3 640,00р.
	Николаев Н.С.	4	3 640,00р.
	Махинов К.Е.	5	5 200,00р.
Общая сумма (тг.):			12480
Дата продажи: 17.09.2008			
№	Фамилия И.О.	Номер места	Стоимость билета (тг)
	Петров В.С.	9	5 200,00р.
	Санникова О.П.	2	3 120,00р.
	Касенова Г.С.	1	3 640,00р.
	Иванова Е.Н.	6	5 200,00р.
Общая сумма (тг.):			17160
Общая сумма (тг.):			29640
Номер рейса: 04			
Дата продажи: 15.09.2008			
№	Фамилия И.О.	Номер места	Стоимость билета (тг)
	Васильева И.А.	1	3 380,00р.
	Санникова О.П.	6	3 120,00р.
Общая сумма (тг.):			6500
Дата продажи: 16.09.2008			
№	Фамилия И.О.	Номер места	Стоимость билета (тг)
	Сайкенов М.Е.	2	3 640,00р.
	Оспанов Е.М.	5	3 640,00р.
Общая сумма (тг.):			7280
Общая сумма (тг.):			13780
Общая сумма (тг.):			43420



4.7.4. Печать отчета

Перед тем как открывать отчет в первый раз, рекомендуется проверить ориентацию страницы, размеры полей и другие параметры страницы.

1. Выберите отчет в окне базы данных или откройте отчет в режиме конструктора, режиме предварительного просмотра или режиме просмотра образца.

2. Выберите в меню **Файл** команду **Печать**.


3. Укажите требуемые значения параметров в диалоговом окне **Печать**.

- В группе **Принтер** выберите принтер.

- В группе **Печатать** выберите печать всего отчета или определенного диапазона страниц.

- В группе **Копии** укажите число печатающихся копий и, при необходимости, печать с раскладкой по копиям.

4. Нажмите кнопку **ОК**.

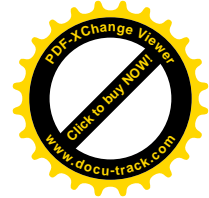
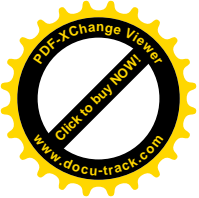
➤ Для вывода отчета на печать без открытия диалогового окна  нажмите кнопку **Печать** на панели инструментов.

Отчет – это объект базы данных, предназначенный для эффективного представления данных в печатной форме.

Отчеты предназначены только для просмотра и печати данных. Отчет невозможно просматривать в режимах таблицы, сводной таблицы и сводной диаграммы.

Окончательный внешний вид отчета во многом зависит от установленного драйвера принтера. Если в системе не установлен ни одного драйвера принтера, MS Access не позволит ни создать отчет, ни отредактировать уже имеющийся.

Отчет может быть открыт в одном из трех режимов: конструктора, предварительного просмотра, просмотр образца.



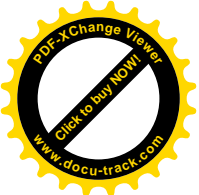
4.8. Защита базы данных

Microsoft Access обеспечивает два традиционных способа защиты базы данных: установка пароля, требуемого при открытии базы данных, и защита на уровне пользователей, которая позволяет ограничить, к какой части базы данных пользователь будет иметь доступ или какую ее часть он сможет изменять. Кроме того, можно удалить изменяемую программу Visual Basic из базы данных, чтобы предотвратить изменения структуры форм, отчетов и модулей, сохранив базу данных как файл MDE.

4.8.1. Установка пароля

Простейшим способом защиты является установка пароля для открытия базы данных. После того как пароль установлен, при каждом открытии базы данных будет появляться диалоговое окно, в которое требуется ввести пароль. Только те пользователи, которые введут правильный пароль, смогут открыть базу данных. Этот способ достаточно надежен (Microsoft Access шифрует пароль, так что к нему нет прямого доступа при чтении файла базы данных), но он применяется только при открытии базы данных. После открытия базы данных все объекты становятся доступными для пользователя (пока не определена защита на уровне пользователей, описанная ниже в этом разделе). Для базы данных, которая совместно используется небольшой группой пользователей или на автономном компьютере, установка пароля обычно оказывается достаточной.

✓ **Внимание!** Не используйте пароль базы данных, если предполагается выполнять репликацию базы данных. Реплицированные базы



данных не могут быть синхронизированы, если определен пароль баз данных.

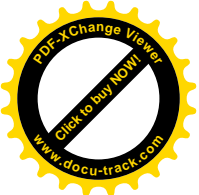
4.8.2. Защита на уровне пользователей

Наиболее гибкий и распространенный способ защиты базы данных называется защитой на уровне пользователей. Этот способ защиты подобен способам, используемым в большинстве сетевых систем. От пользователей требуется идентифицировать себя и ввести пароль, когда они запускают Microsoft Access. Внутри файла рабочей группы они идентифицируются как члены группы. Microsoft Access по умолчанию создает две группы: администраторы (группа «Admins») и пользователи (группа «Users»). Допускается также определение других групп.

Группам и пользователям предоставляются разрешения на доступ, определяющие возможность их доступа к каждому объекту базы данных. Например, члены группы «Users» могут иметь разрешения на просмотр, ввод или изменение данных в таблице «Клиенты», но им не будет разрешено изменять структуру этой таблицы. Группа «Users» может быть допущена только к просмотру данных в таблице, содержащей сведения о заказах, и не иметь доступа к таблице «Платежная ведомость». Члены группы «Admins» имеют все разрешения на доступ ко всем объектам базы данных. Имеется возможность установить более разветвленную структуру управления, создавая собственные учетные записи групп, предоставляя этим группам соответствующие разрешения и добавляя в них пользователей.

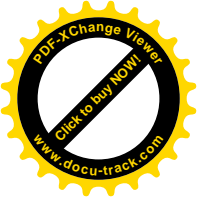
Следует отметить три главных преимущества защиты на уровне пользователей.

1. Защищается ваша программа как интеллектуальная собственность.



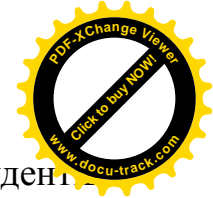
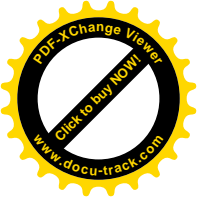
2. Приложение защищается от повреждения из-за неумышленного изменения пользователями программ или объектов, от которых зависит работа приложения.

3. Защищаются конфиденциальные сведения в базе данных.



ЛИТЕРАТУРА

1. Економічна інформатика: навч. посіб. / В.А. Ткаченко, Г.Ю. Під'ячий, В.А. Рябик. - Харків: НТУ "ХПИ" 2011. - 312 с.
2. Экономическая информатика: Учебник / Под ред. В.П. Косарева. - 2-е изд., перераб. и доп. - М.: Финансы и статистика, 2004. - 592 с.
3. Экономическая информатика: Введение в экономический анализ информационных систем: Учебник. - М.:ИНФРА - М, 2005.- 958с. - (Учебник экономического факультета МГУ им. М.В. Ломоносова).
4. Экономическая информатика: Учебник / Под ред. П.В. Конюховского Д.Н. Колесова - СПб: Питер. 2000. - 560 с.
5. Экономическая информатика: Учебник для вузов/ Под ред. д.т.н., проф. В.В. Евдокимова. - СПб: Питер. 1997. - 592 с.
6. Экономическая информатика: Учебное пособие / Н.И. Савицкий - М.: Экономистъ, 2004. - 429 с.
7. Корнейчук Б.В. Информационная экономика. Учебное пособие. - СПб: Питер. 2006. - 400 с.
8. Ивин Л.Н., Куклин В.М. Информационная экономика. - Харьков: изд-во Кроссруд, 2005. - 436 с.
9. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации: Учебник для вузов. 2-е изд. - СПб.: Питер, 2006 - 703 с.
10. Информатика: Базовый курс / С.В. Симонович и др.- 2-е изд., перераб. и доп. - СПб: Питер, 2006. - 640 с.
11. Курс: Информатика. (Юнита 1, 2, 3). Рабочий учебник (Учебное пособие для студентов Современного Гуманитарного Университета). - М. СГУ. 2003.



12. Макарова М.В. Електронна комерція: Посібник для студентів вищих навчальних закладів.- К.: Видавничий центр "Академія", 2002. - 272 с. (Альма-матер).
13. Экономическая информатика / Под ред. П.В. Конюховского и Д.Н. Колесова. – СПб: Питер, 2001. – 560 с.
14. Симонович С.В. Специальная информатика: Учеб. пособие / С.В. Симонович, Г.А. Евсеев, А.Г. Алексеев. – М.: АСТ-ПРЕСС: Инфоком-Пресс, 2000. – 480 с.
15. Гельман В.Я. Решение математических задач средствами Excel: Практикум. – СПб: Питер, 2003. – 240 с.
16. Інформатика: Комп'ютерна техніка. Комп'ютерні технології: Посібник / За ред. О.І. Пушкаря. – К.: Академія, 2001. – 696 с.
17. Медведева О.А. EXCEL в інженерних та економічних розрахунках: Навч. посіб. / О.А.Медведева, А.Г.Фокін. – Краматорськ: ДДМА, 2004.–100 с.
18. Гарнаев А.Ю. «Самоучитель VBA.» СПб.:БХВ-Петеребург,2004.
19. Уокенбах Д. «Профессиональное программирование на VBA в Excel 2002».: Перевод с английского. – М.: Издательский дом «Вильямс», 2003
20. Фокін А.Г. Система керування базами даних Access в економічних розрахунках: Навчальний посібник для студентів вищих навчальних закладів / А.Г. Фокін, І.А. Гетьман. – Краматорськ: ДДМА, 2005. – 132 с.