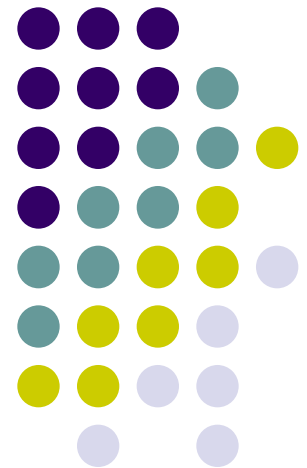
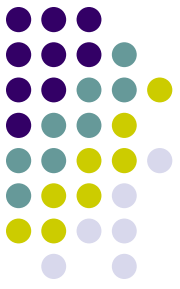


---

**Понятие информационные  
систем.  
История развития баз данных**

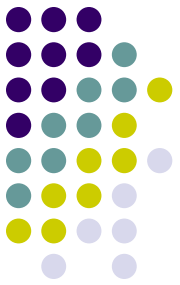




# Литература

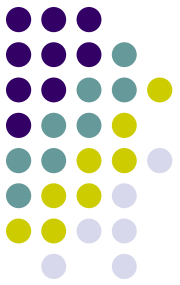
- **Базы данных: модели, разработка, реализация /** Т, С. Карпова. — СПб.: Питер,2001.
- М.Р. Когаловский. “Энциклопедия технологий баз данных”. М. Финансы и статистика, 2002.
- К. Дейт. “Введение в системы баз данных”. 7-е изд., М.; СПб.: Вильямс.- 2001
- Введение в системы управления базами данных *Пушников А.Ю.*

# Области использования ВТ



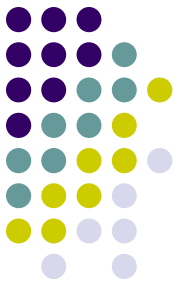
- Первая область — применение вычислительной техники для выполнения численных расчетов, которые слишком долго или вообще невозможно производить вручную.
- Характерной особенностью данной области применения вычислительной техники является
  - 1. наличие сложных алгоритмов обработки,
  - 2. простые по структуре данные,
  - 3. объем данных сравнительно невелик.

## 2-ая область применения



- Вторая область — это использование средств вычислительной техники в автоматизированных информационных системах

# Функции информационных систем



- Информационная система представляет собой программно-аппаратный комплекс, обеспечивающий выполнение следующих функций:
- надежное хранение информации в памяти компьютера;
- выполнение специфических для данного приложения преобразований информации и вычислений;
- Предоставление пользователям удобного и легко осваиваемого интерфейса.

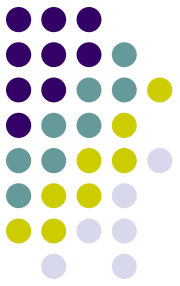


# Особенности ИС

- Большие объемы информации,
- Сложная структура данных

Классическими примерами информационных систем являются банковские системы, системы резервирования авиационных или железнодорожных билетов, мест в гостиницах и т. Д.

# Долговременная память старых компьютеров



- на заре вычислительной техники возможности компьютеров по хранению информации были очень ограниченными.
- В первых компьютерах использовались два вида устройств внешней памяти — магнитные ленты и барабаны.
- Емкость магнитных лент была достаточно велика, но по своей физической природе они обеспечивали последовательный доступ к данным.
- Магнитные барабаны давали возможность произвольного доступа к данным, но имели ограниченный объем хранимой информации.
- Эти ограничения не являлись слишком существенными для чисто численных расчетов

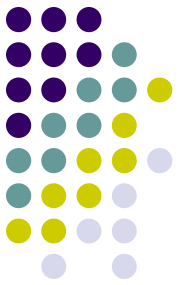


- Именно требования нечисловых приложений вызвали появление съемных магнитных дисков с подвижными головками, что явилось революцией в истории вычислительной техники.

Эти устройства внешней памяти обладали

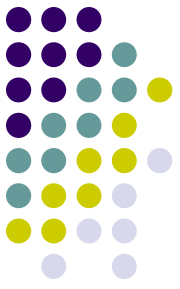
- существенно большей емкостью
- обеспечивали удовлетворительную скорость доступа к данным в режиме произвольной выборки,
- возможность смены дискового пакета на устройстве позволяла иметь практически неограниченный архив данных.





- С появлением магнитных дисков началась история систем управления данными во внешней памяти.

# Файлы и файловые системы

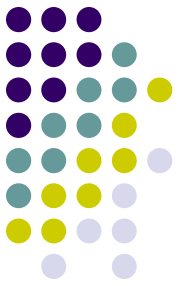


- С точки зрения прикладной программы, файл — это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные.

## Система управления файлами берет на себя

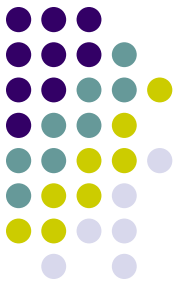
- распределение внешней памяти,
- отображение имен файлов в соответствующие адреса во внешней памяти
- и обеспечение доступа к данным.

# Зависимость от данных



- структура записи файла была известна только программе, которая с ним работала
- для того, чтобы извлечь некоторую информацию из файла, необходимо было точно знать структуру записи файла с точностью до бита.
- Каждая программа, работающая с файлом, должна была иметь у себя внутри структуру данных, соответствующую структуре этого файла.
- Поэтому при изменении структуры файла требовалось изменять структуру программы
- Такая ситуации характеризовалась как зависимость программ от данных.
- Это было первым существенным недостатком файловых систем, который явился толчком к созданию новых систем хранения и управления информацией.

# Авторизация доступа



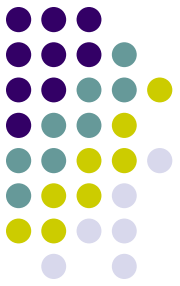
- В большинстве современных систем управления файлами применяется подход к защите файлов, впервые реализованный в ОС UNIX.
- Администрирование режимом доступа к файлу в основном выполняется его создателем-владельцем.
- Для множества файлов, отражающих информационную модель одной предметной области, такой децентрализованный принцип управления доступом вызывал дополнительные трудности.
- Отсутствие централизованных методов управления доступом к информации послужило еще одной причиной разработки СУБД.



# Параллельная работа

- необходимость обеспечения эффективной параллельной работы многих пользователей с одними и теми же файлами.
- Если все пользователи собираются только читать файл, то все нормально.
- Но если хотя бы один из них будет изменять файл, для корректной работы этих пользователей требуется взаимная синхронизация их действий по отношению к файлу.

# *Первый этап — базы данных на больших ЭВМ*

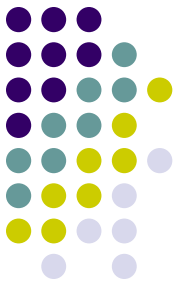


- История развития СУБД насчитывает более 30 лет.
- В 1968 году была введена в эксплуатацию первая промышленная СУБД система IMS фирмы IBM.
- В 1975 году появился первый стандарт ассоциации по языкам систем обработки данных - Conference of Data System Languages (CODASYL),



- В дальнейшее развитие теории баз данных большой вклад был сделан американским математиком Э. Ф. Коддом, который является создателем реляционной модели данных.
- В 1981 году Э. Ф. Кодд получил за создание реляционной модели и реляционной алгебры престижную премию Тьюринга Американской ассоциации по вычислительной технике.

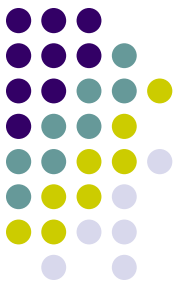
# четыре этапа в развитии обработки данных.



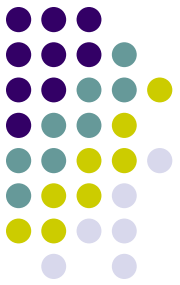
- 1. Первый этап развития СУБД связан с организацией баз данных на больших машинах типа IBM 360/370, ЕС-ЭВМ и мини-ЭВМ типа PDP11 (фирмы Digital Equipment Corporation – DEC)
- Базы данных хранились во внешней памяти центральной ЭВМ, пользователями этих баз данных были задачи, запускаемые в основном в пакетном режиме.
- Интерактивный режим доступа обеспечивался с помощью консольных терминалов, которые не обладали собственными вычислительными ресурсами (процессором, внешней памятью) и служили только устройствами ввода-вывода для центральной ЭВМ



# Особенности этого этапа развития выражаются в следующем

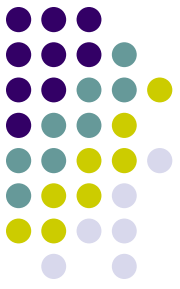


- Все СУБД базируются на мощных мультипрограммных операционных системах (MVS SVM, RTE, OSRV, RSX, UNIX), поэтому в основном поддерживается работа с централизованной базой данных в режиме распределенного доступа.
- Функции управления распределением ресурсов в основном осуществляются операционной системой (ОС),
- Поддерживаются языки низкого уровня манипулирования данными, ориентированные на навигационные методы доступа к данным.
- Значительная роль отводится администрированию данных.



- Проводятся серьезные работы по обоснованию и формализации реляционной модели данных, и была создана первая система (System R), реализующая идеологию реляционной модели данных.
- Проводятся теоретические работы по оптимизации запросов и управлению распределенным доступом к централизованной БД, было введено понятие транзакции.
- Результаты научных исследований открыто обсуждаются в печати, идет мощный поток общедоступных публикаций, касающихся всех аспектов теории и практики баз данных, и результаты теоретических исследований активно внедряются в коммерческие СУБД,
- Появляются первые языки высокого уровня для работы с реляционной моделью данных. Однако отсутствуют стандарты для этих первых языков.

# Эпоха персональных компьютеров



- Все СУБД были рассчитаны на создание БД в основном с монопольным доступом. В редких случаях предполагалась последовательная работа нескольких пользователей, например, сначала оператор, который вводил бухгалтерские документы, а потом главбух, который определял проводки, соответствующие первичным документам.
- Большинство СУБД имели развитый и удобный пользовательский интерфейс. Инструментальная среда состояла из готовых элементов приложения в виде шаблонов экранных форм, 'отчетов, (этикеток'(Labels), графических конструкторов запросов, которые достаточно просто могли быть собраны в единый комплекс.



- Во всех настольных СУБД поддерживался только внешний уровень представления реляционной модели, то есть только внешний, табличный вид структур данных.
- При наличии высокоуровневых языков манипулирования данными типа реляционной алгебры и SQL в настольных СУБД поддерживались низкоуровневые языки манипулирования данными на уровне отдельных строк таблиц.
- В настольных СУБД отсутствовали средства поддержки ссылочной и структурной целостности базы данных. Эти функции должны были выполнять приложения, однако скудость средств разработки приложений иногда не позволяла это сделать, и в этом случае эти функции должны были выполняться пользователем, требуя от него дополнительного контроля при вводе и изменении информации, хранящейся в БД.



- Наличие монопольного режима работы фактически привело к вырождению функций администрирования БД и в связи с этим - к отсутствию инструментальных средств администрирования БД.
- И, наконец, последняя и в настоящий момент весьма положительная особенность - это сравнительно скромные требования к аппаратному обеспечению со стороны настольных СУБД.



- В принципе, их даже трудно назвать полноценными СУБД,
- Яркие представители этого семейства — очень широко использовавшиеся до недавнего времени СУБД Dbase (DbaseIII+, DbaseIV), FoxPro, Clipper, Paradox.

# Распределенные базы данных



- Практически все современные СУБД обеспечивают поддержку полной реляционной модели, а именно:
- структурной целостности — допустимыми являются только данные, представленные в виде отношений реляционной модели;
- языковой целостности, то есть языков манипулирования данными высокого уровня (в основном SQL);
- ссылочной целостности, контроля за, соблюдением ссылочной целостности в течение всего времени функционирования системы, и гарантий не возможности со стороны СУБД нарушить эти ограничения,
- Большинство современных СУБД рассчитаны на многоплатформенную архитектуру,



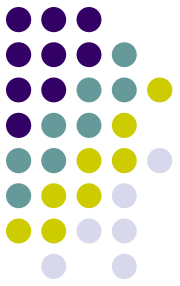
- Необходимость поддержки многопользовательской работы с базой данных и возможность децентрализованного хранения данных потребовали развития средств администрирования БД с реализацией общей концепции средств защиты данных.
- Потребность в новых реализациях вызвала создание серьезных теоретических трудов по оптимизации реализаций распределенных БД и работе с распределенными транзакциями и запросами с внедрением полученных результатов в коммерческие СУБД.
- Для того чтобы не потерять клиентов, которые ранее работали на настольных СУБД, практически все современные СУБД имеют средства подключения клиентских приложений, разработанных с использованием настольных СУБД, и средства экспорта данных из форматов настольных СУБД второго этапа развития.





- Именно к этому этапу можно отнести разработку ряда стандартов в рамках языков описания и манипулирования данными начиная с SQL89, SQL92, SQL99 и технологий по обмену данными между различными СУБД, к которым можно отнести и протокол ODBC (Open DataBase Connectivity), предложенный фирмой Microsoft.
- Именно к этому этапу можно отнести начало работ, связанных с концепцией объектно-ориентированных БД - СУБД.
- Представителями СУБД, относящимся ко второму этапу, можно считать MS Access 97 и все современные серверы баз данных Oracle7, Oracle 8.4, MS SQL6.5, MS SQL7.0, System 10, System 11, Informix, DB2, SQL Base и другие современные серверы баз данных, которых в настоящий момент насчитывается несколько десятков.

# Основные функции СУБД



1. создание (конструирование) базы данных путем описания структуры хранимой информации и взаимосвязей между ее частями;
2. занесение, хранение и удаление информации из базы данных - т. е. поддержание БД в актуальном состоянии;
3. обслуживание выборок и запросов пользователей

# Основные функции СУБД



## 1. Непосредственное управление данными во внешней памяти

Эта функция включает обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например, для убыстрения доступа к данным

## 2. Управление буферами оперативной памяти

СУБД обычно работают с БД значительного размера. Практически единственным способом реального увеличения скорости работы с такими БД является буферизация данных в оперативной памяти.

## 3. Управление транзакциями

Транзакция - это последовательность операций над БД, рассматриваемых СУБД как единое целое. Либо транзакция успешно выполняется, и СУБД фиксирует (COMMIT) изменения БД, произведенные этой транзакцией, во внешней памяти, либо ни одно из этих изменений никак не отражается на состоянии БД. Понятие транзакции необходимо для поддержания логической целостности БД.

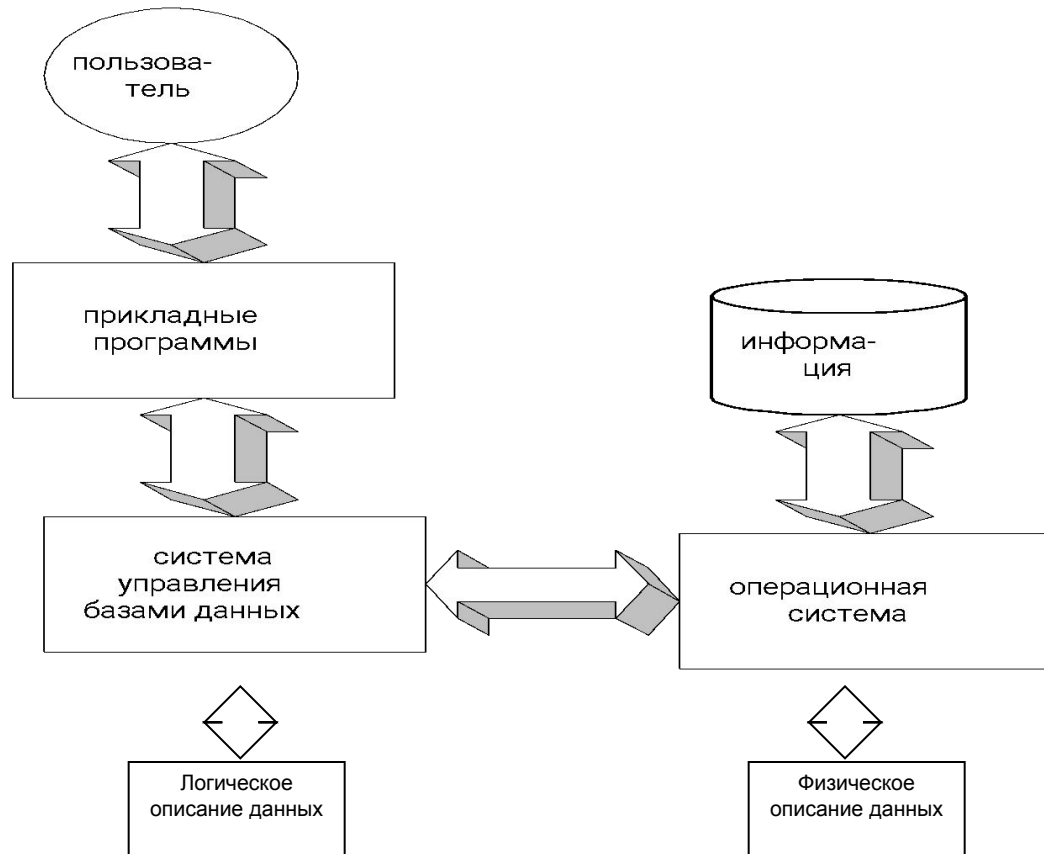
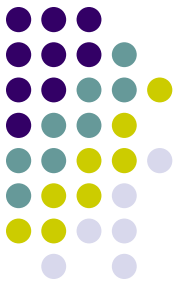
## 4. Поддержка языков БД

Для работы с базами данных используются специальные языки, в целом называемые *языками баз данных*.

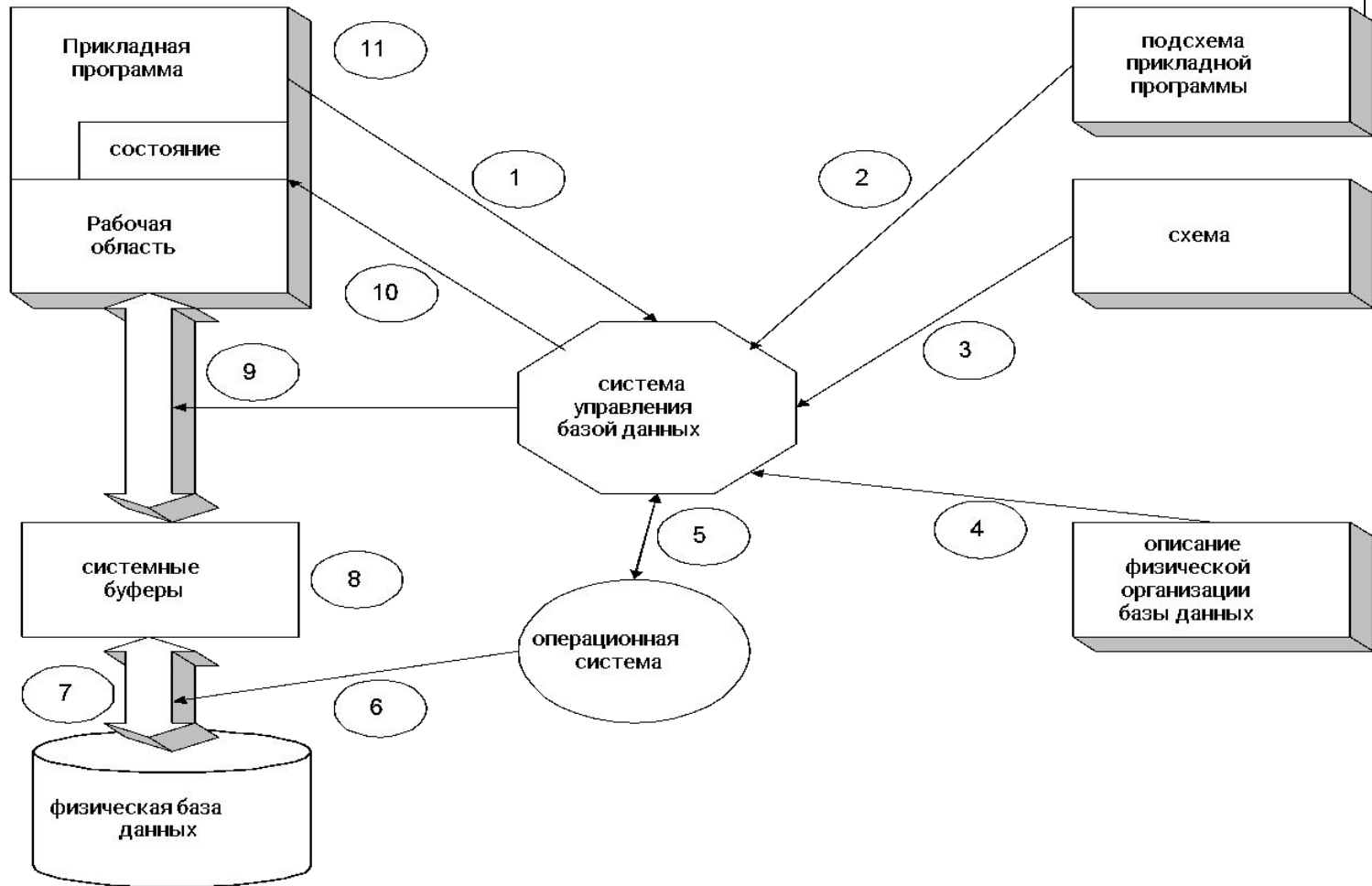
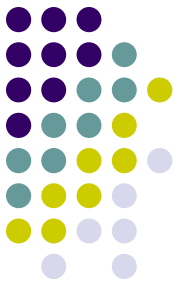
## 5. Журнализация

Одним из основных требований к СУБД является надежность хранения данных во внешней памяти, поэтому большинство СУБД на сегодняшний день являются журналируемыми, то есть все данные подлежат детальной «описи»

# Схема обработки данных в информационной системе



# Технологическая схема работы СУБД





1. Прикладная программа выдает запрос на чтение записи системе управления базой данных (1). Программа сообщает имя пользователя, затребовавшего данные и передает в запросе значение ключа сегмента или записи.
2. Система управления базой данных получает в распоряжение подсхему, используемую прикладной программой (описание данных для прикладной программы), и осуществляет в ней поиск описания данных на которые выдан запрос (2).
3. Система управления базой данных получает в распоряжение схему (глобальное логическое описание данных) и с ее помощью определяет какого типа или каких типов логические данные необходимы (3).
4. Система управления базой данных просматривает описание физической организации базы данных и определяет, какую физическую запись (или запись) требуется считать (4).



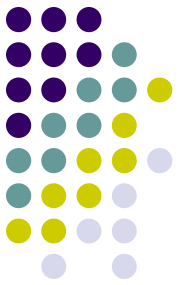
5. Система управления базой данных выдает операционной системе команду чтения (5) требуемой записи (или записей).
6. Операционная система взаимодействует с физической памятью, в которой хранятся данные (6).
7. Запрошенные данные передаются из памяти в системные буферы (7).
8. Система управления базой данных, осуществляя сравнение схемы и подсхемы, выделяет ту логическую запись, которая запрошена прикладной программой (8). Любое преобразование данных, необходимость в котором возникает из-за различия в описании одних тех же данных в схеме и подсхеме, выполняется системой управления базой данных.
9. Система управления базой данных передает данные из системных буферов в рабочую область прикладной программы (9).
10. Система управления базой данных передает прикладной программе информацию о результатах выполнения различных процедур по обслуживанию запроса (10). Эта информация содержит также сведения об ошибках, если они имеют место.
11. Прикладная программа обрабатывает данные, помещенные в ее рабочую область (11).

# Этапы разработки информационных систем

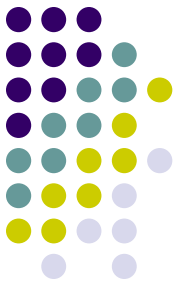


1. **Постановка задачи.** Необходимо четко сформулировать цель работы, предполагаемые результаты, объем работ, который предполагается выполнять на ЭВМ; оценить имеющееся программное обеспечение и стоимость закупки или разработки недостающего; решить вопрос о целесообразности разработки; подготовить **проект договора, техническое задание, календарный план, соглашение о цене.**
2. Обследование **предметной области**, то есть структуры, функциональных связей и документооборота предприятия. Разработка требований к блокам ИС.
3. Разбиение общих задач на отдельные блоки, **обоснование целесообразности** автоматизации отдельных задач, выбор состава и очередности решения задач
4. **Концептуальное проектирование:** исследование потоков и структуры информации, построение функционально-информационной схемы (**концептуальной схемы**) и структурных единиц информации (часто на основе реально используемых документов и нормативно-справочного обеспечения). Этап заканчивается построением **модели “сущность-связь”** – неформальной модели предметной области, позволяющей описывать объекты предметной области и их взаимоотношения. Основным достоинством модели является ее относительная простота и понятность за счет применения естественного языка, что позволяет использовать модель в качестве инструмента общения разработчика и будущего пользователя при сборе информации о предметной области базы данных. **Сущность** – это собирательное понятие, некоторая абстракция реально существующего объекта, процесса или явления, о котором необходимо хранить информацию в системе. В качестве сущностей в моделях предметной области рассматривают материальные (предприятия, изделия, сотрудники и т. п.) и нематериальные (явления, структуры) объекты реальной действительности. В моделях предметной области типа “сущность-связь” каждая рассматриваемая конкретная сущность является узловой точкой сбора информации об этой сущности. В модели используется также понятие “экземпляр сущности”. Графически модель “сущность-связь” представляется в виде **ER-диаграмм** (Entity-Relation Diagrams).





5. Проектирование реализации: **формальное описание** задач, разработка **блок-схем** программного обеспечения, составление спецификации программных модулей, подлежащих реализации.
6. Разработка алгоритма и отладка программ. **Алгоритм** – это конечная последовательность точно определенных действий, однозначно определяющая процесс преобразования исходных и промежуточных данных, приводящий к решению задачи.
7. Трансляция программ. **Транслятор** – это программа, переводящая текст программы, написанный на алгоритмическом языке, в машинные коды.
8. **Тестирование** программ. Программа, не имеющая синтаксических ошибок, может иметь логические ошибки и выдавать неверные результаты. Поэтому как отдельные блоки, так и программа в целом должны быть проверены с помощью тестовых задач с известными решениями.



9. Подготовка **инструкции** пользователю. Программа должна быть эргономичной, то есть обеспечивать удобный **диалог** с пользователем. Набор подсказок, как правило, включается в меню и специальный блок программы.
10. Обеспечение секретности информации и **администрирование** системы, то есть настройка системы допусков и паролей, а также ликвидация сбоев в работе системы.
11. **Совершенствование** системы. Серьезные фирмы-поставщики программного обеспечения не прерывают связей с клиентами и снабжают их новыми программными блоками по льготным ценам.