



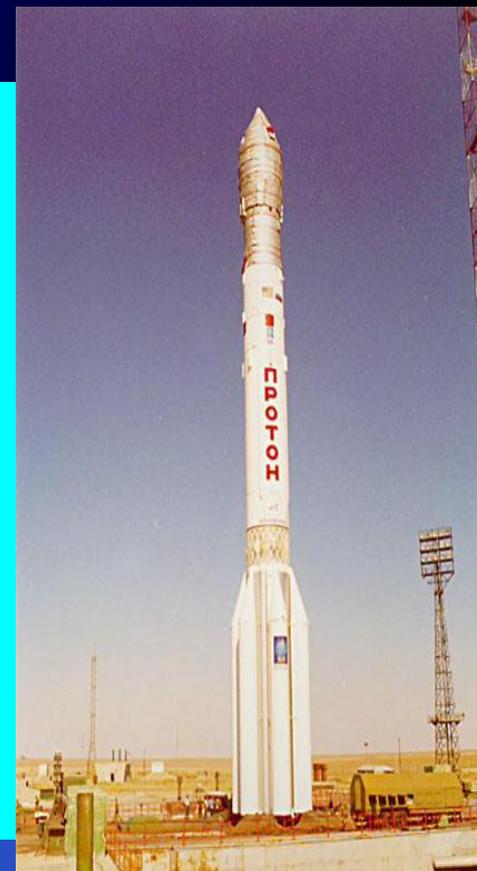
Институт энергоэффективности и водородных технологий
Кафедра инновационных технологий наукоемких отраслей



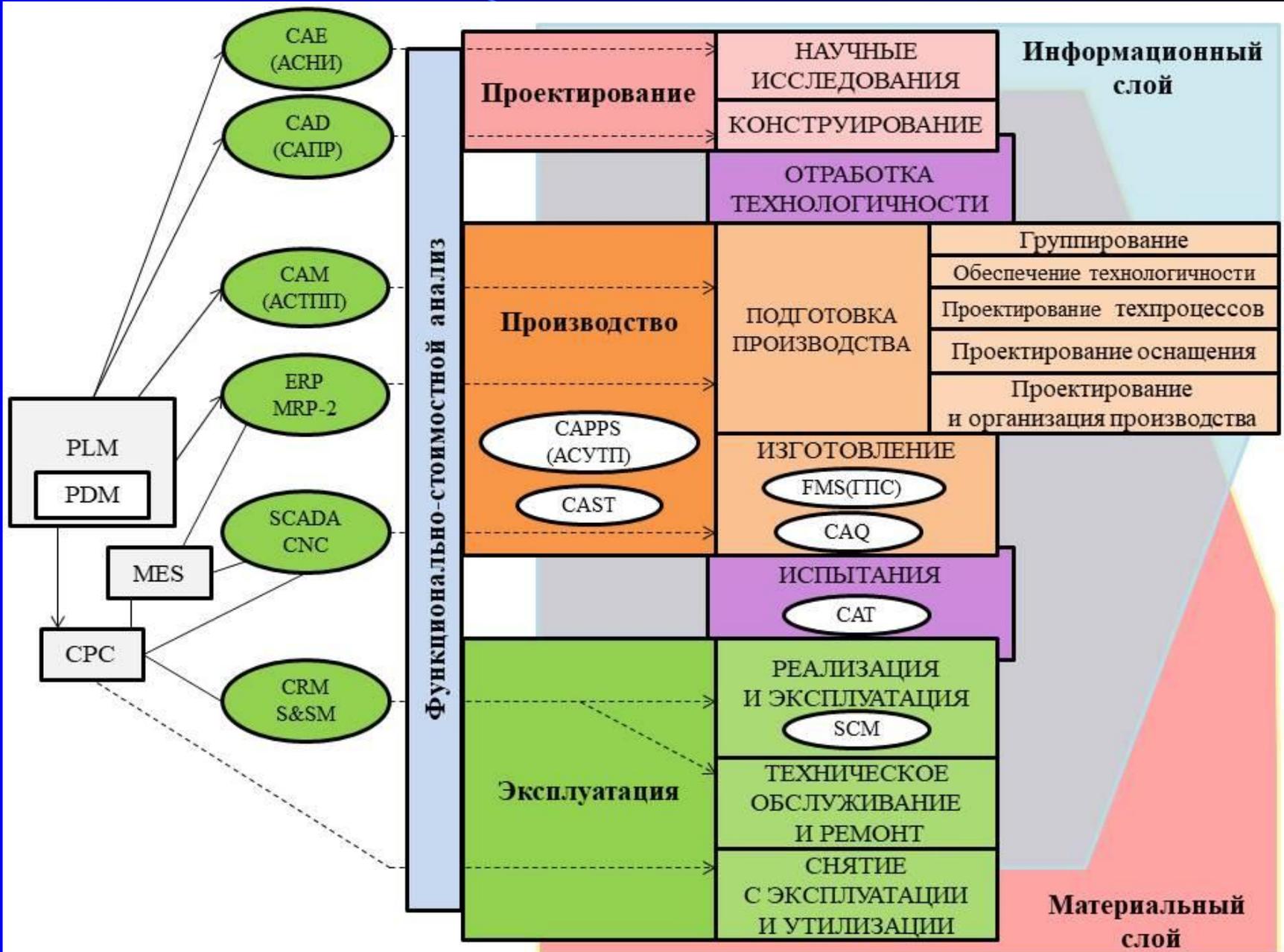
Информационное сопровождение наукоемкой продукции

Лекция 8

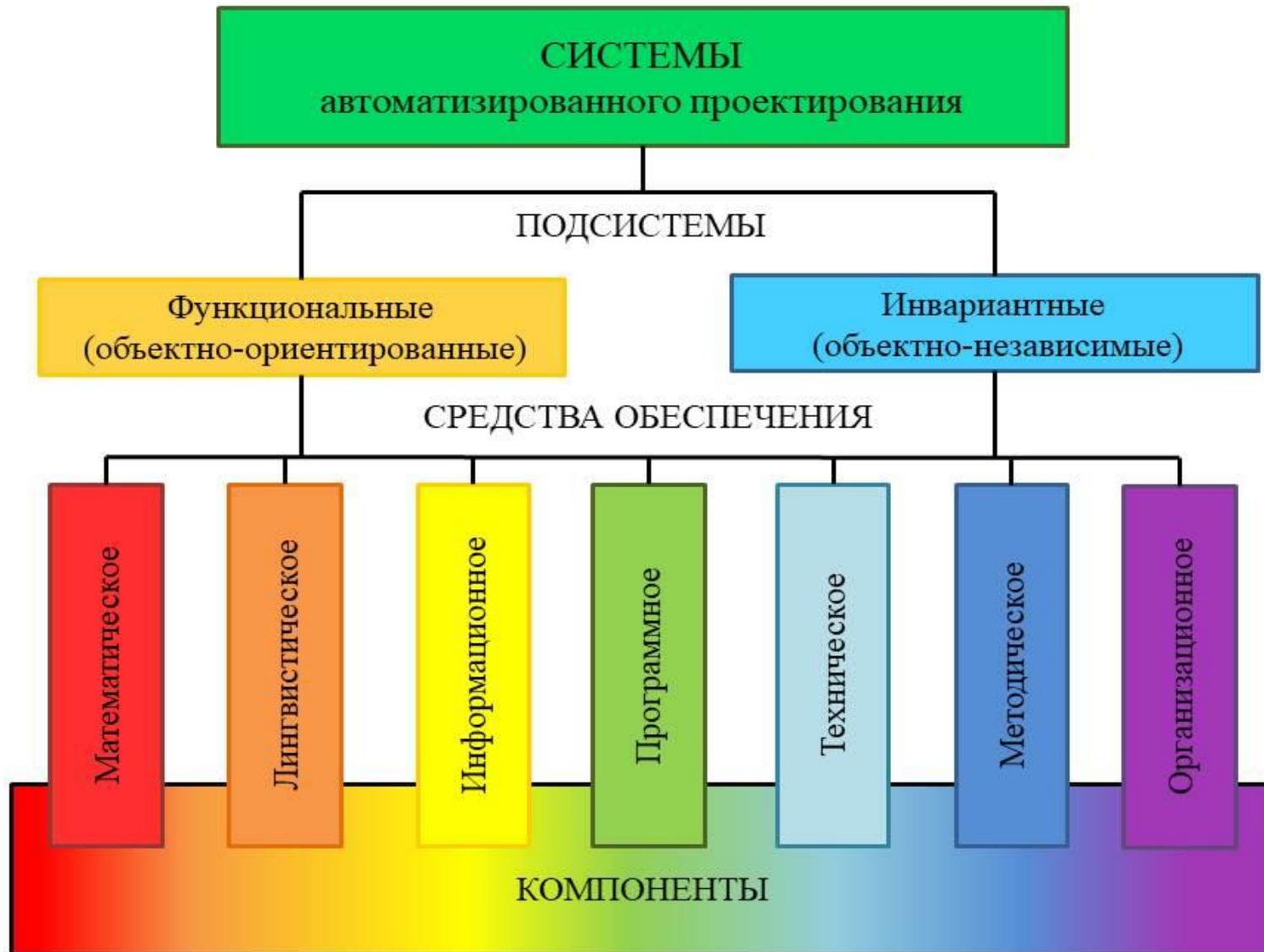
Лекции для группы ФП-07м-21
(2021/22 уч. г.)



Назначение информационных систем



Структура систем автоматизированного проектирования



Определение и назначение лингвистического обеспечения

Лингвистическое обеспечение информационных систем – это совокупность языков, используемых на различных стадиях создания и эксплуатации информационной системы и обеспечивающих взаимодействие персонала и пользователей информационной системы с информационным, программным и техническим обеспечением системы.

Состав лингвистического обеспечения

В состав лингвистического обеспечения включаются:

- языки программирования;
- языки, описания информационной базы, описания документов, описания данных;
- языки манипулирования данными;
- языки эксплуатации информационных систем:
 - информационно-поисковые языки;
 - языки проектирования;
- система терминов и определений конкретной предметной области, используемых в процессе разработки и функционирования автоматизированной информационной системы.

Определение и типы языков

Язык – это знаковая информационная система, выполняющая функции формирования, хранения и передачи информации в процессе познания действительности и общения между людьми.

Различают естественные и искусственные языки.

Естественный язык представляет собой информационную систему звуков и графических знаков, исторически возникающую в обществе в процессе общения.

Искусственный язык – система, специально созданная на основе естественного языка для более точного и экономного преобразования и передачи информации.

Проблемы ИСПОЛЬЗОВАНИЯ ЯЗЫКОВ

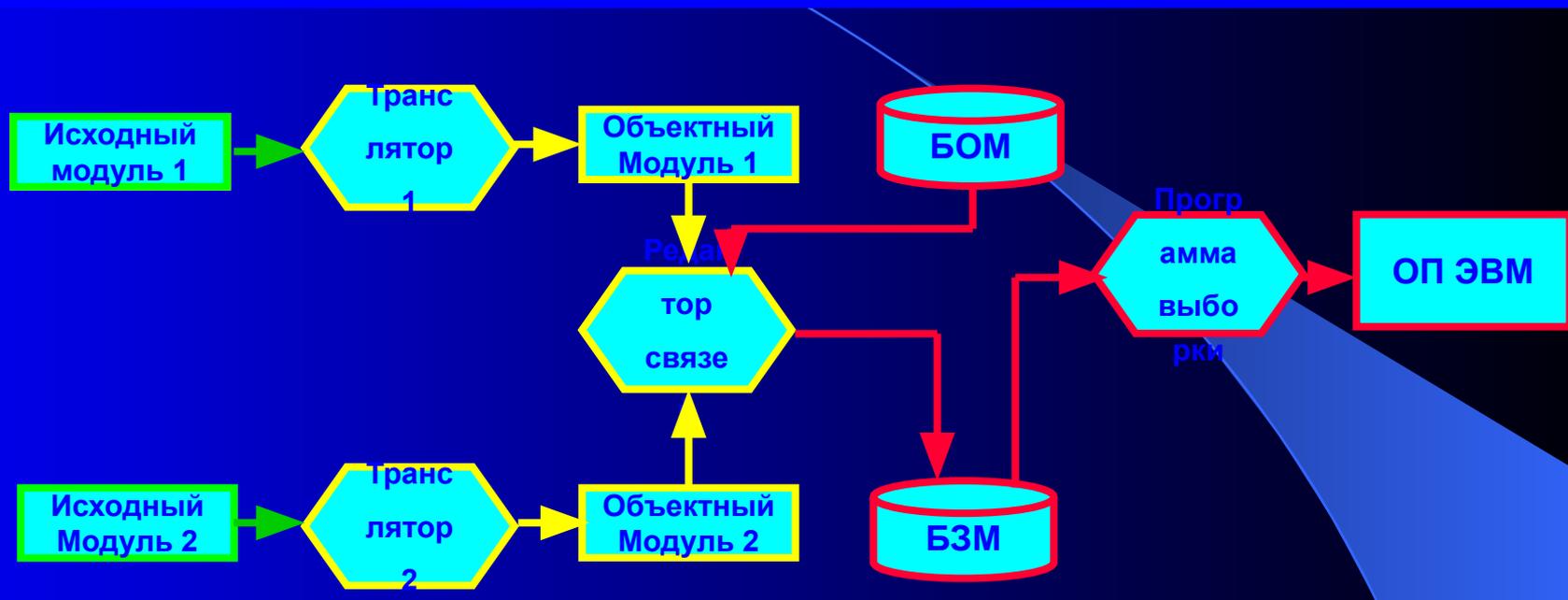
Проблемы использования естественного языка:

1. Многообразие средств передачи смысла
2. Семантическая неоднозначность
3. Эллипсность.

Проблемы использования искусственного языка:

1. Большой объем описаний
2. Трудность выражения потребности
(необходимость обучения).

Этапы обработки задания на ЭВМ



БОМ – библиотека объектных модулей;

БЗМ – библиотека загрузочных модулей;

ОП ЭВМ- оперативная память ЭВМ.

Всякий исходный модуль в результате трансляции преобразуется в стандартную, принятую для данного типа ЭВМ форму объектного модуля.

Структура лингвистического обеспечения



Автокоды
и ассемблеры

Языки
программирования

Алгоритмические
языки

Языки
проектирования

Языки логического
программирования

Языки
информационных
технологий

Основные понятия языка программирования

- **Язык программирования** – формальная знаковая система, предназначенная для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, задающих внешний вид программы и действия, которые выполнит исполнитель или исполнительное устройство (компьютер) под её управлением.
- Со времени создания первых программируемых вычислительных машин человечество придумало более восьми тысяч языков программирования (включая нестандартные, визуальные и эзотерические языки).

Проанализируем составные части определения языка программирования:

- **Формальный язык** – это множество конечных слов (строк, цепочек) над конечным алфавитом.
- **Знаковая система** – это система однозначно интерпретируемых и трактуемых сообщений/сигналов, которыми можно обмениваться в процессе общения. Иногда знаковые системы помогают структурировать процесс общения с целью придания ему некой адекватности в плане реакций его участников на те или иные «знаки». В качестве примера знаковой системы может быть рассмотрен язык общения (как в письменной форме так и, в случае естественных языков, в форме речи).
- **Компьютерная программа** – последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины.
- **Лексика** – совокупность слов того или иного языка, части языка или слов, которые знает тот или иной человек или группа людей.

Основные понятия языка программирования

- **Синтаксис** – составная часть языка программирования, которая описывает структуру программ как наборов символов безотносительно к содержанию.
Синтаксис языка дополняется его семантикой. Синтаксис языка описывает «чистый» язык, в то же время семантика приписывает значения (действия) различным синтаксическим конструкциям.
- **Семантика** в программировании – дисциплина, изучающая формализации значений конструкций языков программирования посредством построения их формальных математических моделей. В качестве инструментов построения таких моделей могут использоваться различные средства, например, математическая логика, теория множеств, теория категорий, теория моделей, универсальная алгебра. Формализация семантики языка программирования может использоваться как для описания языка, определения свойств языка, так и для целей формальной верификации программ на языке программирования.
- **Язык** – знаковая система, соотносящая понятийное содержание и типовое звучание (написание).

Особенности языков программирования

Язык программирования – множество заранее определенных, однообразных и понятных исполнительному устройству и исполнителю (интерпретатору, компилятору, компьютеру, программисту) инструкций, предназначенных для определения состава и последовательности действий с целью их исполнения устройством, являющимся частью вычислительной системы.

Язык программирования должен обладать рядом особенностей:

- включать конечное ограниченное число инструкций;**
- иметь определенность порядка построения и следования инструкций для получения определенных результатов;**
- должны быть определены правила написания и представления инструкций;**
- конструкции языка должны иметь однозначное толкование, соотношение того, что написано с тем, что требуется обозначить.**

Языки программирования разделяются на низкоуровневые и высокоуровневые языки. Эти виды отличаются «толщиной прослойки» между программистом и процессором.

В низкоуровневых языках каждая инструкция представляет собой одну или небольшое число процессорных команд,

а в высокоуровневых языках

каждая инструкция – это большой набор процессорных команд.

Подходы к программированию

Языки программирования

подразделяются на две группы:

- императивные;
- декларативные.

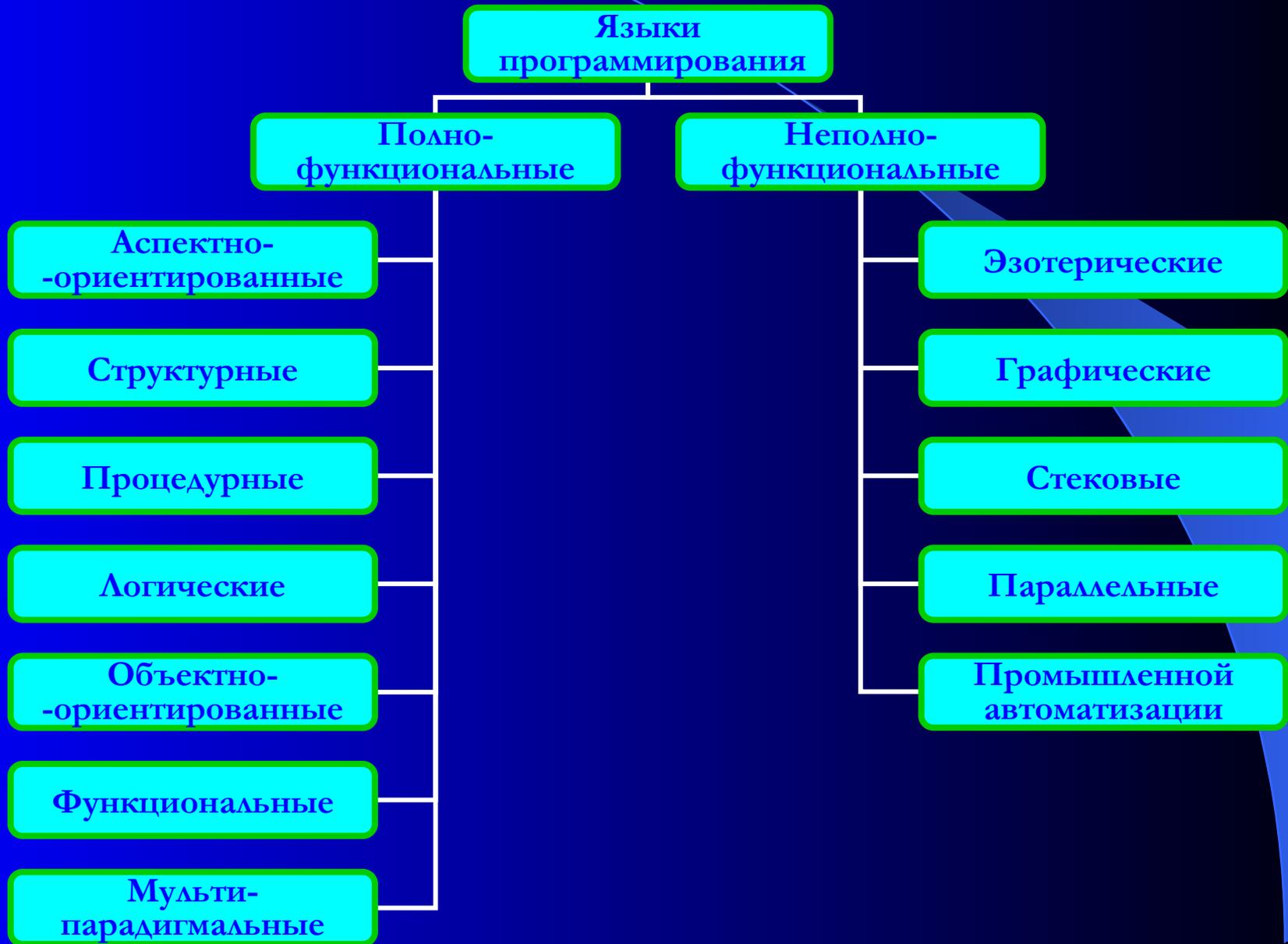
Программа на императивном языке программирования с математической точки зрения представляет собой общее решение поставленной задачи («как сделать?»). Это последовательность команд, выполняемая исполнителем.

Программа на декларативном языке программирования является сочетанием формализованной в рамках языка программирования задачей и всех необходимых для её решения теорем («что сделать?»).

Конкретную последовательность действий определяет компилятор или интерпретатор.



Классификация языков программирования



Определения классов языков программирования

Рассмотрим определения некоторых классов языков:

* **Парадигма программирования** – это совокупность идей и понятий, определяющих стиль (подход к программированию) написания компьютерных программ. Это способ концептуализации, определяющий организацию вычислений и структурирование работы вычислительной машины.

Аспектно-ориентированное программирование (АОП) – парадигма программирования, основанная на идее разделения функциональности для улучшения разбиения программы на модули.

Структурное программирование – методология разработки программного обеспечения, в основе которой лежит представление программы в виде иерархической структуры блоков.

Процедурное программирование – программирование на императивном языке, при котором последовательно выполняемые операторы можно собрать в подпрограммы, то есть более крупные целостные единицы кода, с помощью механизмов самого языка.

Логическое программирование – парадигма программирования, основанная на автоматическом доказательстве теорем, а также раздел дискретной математики, изучающий принципы логического вывода информации на основе заданных фактов и правил вывода. Логическое программирование основано на теории и аппарате математической логики с использованием математических принципов резолюций.

Определения классов языков программирования

Рассмотрим определения некоторых классов языков:

Объектно-ориентированное программирование (ООП) – парадигма программирования, в которой основными концепциями являются понятия объектов и классов. В случае языков с прототипированием вместо классов используются объекты-прототипы.

Функциональное программирование – раздел дискретной математики и парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании (в отличие от функций как подпрограмм в процедурном программировании).

Мультипарадигмальный язык программирования – как правило, язык программирования, который был разработан специально как инструмент мультипарадигмального программирования, то есть изобразительные возможности которого изначально предполагалось унаследовать от нескольких, чаще всего неродственных языков.

Эзотерический язык программирования – язык программирования, созданный для исследования границ возможностей разработки языков программирования, для доказательства потенциально возможной реализации некой идеи, в качестве произведения программного искусства.

Состав языков проектирования

Языки

Входные

Промежу-
точные

Внутрен-
ние

Диагнос-
тики

Выходны
е

Определения структурных элементов языков

Язык – система символов и правил представления информации, включает: алфавит, лексику и правила грамматики.

Алфавит – система знаков для записи слов и выражений.

Лексика или словарный состав – совокупность слов, словосочетаний и выражений, используемых для построения текстов. (Слова – лексические единицы).

Грамматика:

Морфология – совокупность средств и способов построения и изменения слов.

Синтаксис – совокупность средств и способов соединения слов в выражения и фразы.

Языки описания и процедурные языки

Языки описания (декларативные языки):

- предкоординатные (классификационные);
- посткоординатные (дескрипторные).

Процедурные языки:

языки запросов
и манипулирования данными.

Виды языков манипулирования данными

Языки манипулирования данными:

- информационно-логические языки:
 - языки, баз данных;
 - языки, представления знаний.
- информационно-поисковые языки:
 - классификационные языки:
 - иерархические;
 - фасетные.
 - дескрипторные языки:
 - с грамматикой и без грамматики;
 - с контролируемой и свободной логикой.

Понятия о базах и банках данных

Базы данных являются основой информационного обеспечения автоматизированных информационных систем. Именно в них хранятся данные, разделяемые многими пользователями и используемые в различных процедурах обработки информации.

Базой данных (БД) принято называть поименованную совокупность организованных (структурированных) определенным образом данных, отражающую состояние описываемых объектов и отношений между данными в рассматриваемой предметной области и хранящихся во внешней памяти.

Для управления данными (доступа к ним, обновления, пополнения и тому подобного) используют специальные лингвистические и программные средства, составляющие систему управления базами данных (СУБД).

Совокупность БД и СУБД называют банком данных.

Этапы работы с СУБД

Для компьютерной обработки баз данных используют СУБД.

СУБД - это набор средств программного обеспечения, необходимых для создания, обработки и вывода записей баз данных.

Различают несколько типов СУБД: иерархические, сетевые, реляционные.

При работе с СУБД выделяют несколько последовательных этапов:

- проектирование базы данных;
- создание структуры базы данных;
- заполнение базы данных;
- просмотр и редактирование базы данных;
- сортировку базы данных;
- поиск необходимой записи;
- выборку информации;
- создание отчетов.

Как правило, большинство популярных СУБД поддерживают эти этапы и предоставляют удобный инструментарий для их реализации.

Задачи проектирования баз данных

К основным задачам проектирования баз данных относятся:

- обеспечение хранения в БД всей необходимой информации;
- обеспечение возможности получения данных по запросам;
- сокращение избыточности и дублирования данных;
- обеспечение целостности баз данных.

Проектирование баз данных выполняется в три этапа:

- концептуальное (инфологическое) проектирование;
- логическое (даталогическое) проектирование;
- физическое проектирование.

В результате проектирования формируются концептуальная, логическая и физическая модели баз данных.

Концептуальное проектирование баз данных

Концептуальное (инфологическое) проектирование - построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстрагирования.

Такая модель создаётся без ориентации на конкретные СУБД, а также логическую и физическую модели данных.

Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться понятия «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является как образом реальности, так и образом проектируемой базы данных для этой реальности.

Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные ER- диаграммам.

Формализация процесса построения концептуальных моделей баз данных

В процессе проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной логической модели базы данных (иерархической, сетевой или реляционной).

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма «сущность-связь» (ER- диаграмма).

Согласно нотации, сущность изображается в виде прямоугольника, содержащего её имя, выражаемое существительным. Имя сущности должно быть уникальным в рамках одной модели. При этом, имя сущности - это имя типа, а не конкретного экземпляра данного типа. Экземпляром сущности называется конкретный представитель данной сущности.

Связи и атрибуты сущностей в формальной модели баз данных

Связь изображается линией, которая связывает две сущности, участвующие в отношении. Степень конца связи указывается графически, множественность связи изображается в виде «вилки» на конце связи. Модальность связи так же изображается графически - необязательность связи помечается кружком на конце связи. Именование обычно выражается одним глаголом в изъявительном наклонении настоящего времени: «имеет», «принадлежит» или глаголом с поясняющими словами: «включает в себя», и тому подобное. Наименование может быть одно для всей связи или два для каждого из концов связи. Во втором случае, название левого конца связи указывается над линией связи, а правого - под линией. Каждое из названий располагаются рядом с сущностью, к которой оно относится.

Атрибуты сущности записываются внутри прямоугольника, изображающего сущность и выражаются существительным в единственном числе (возможно, с уточняющими словами).

Пример ER-диаграммы (концептуальной модели базы данных)



Состав описаний концептуальной модели

Концептуальная модель - это модель предметной области, состоящей из перечня взаимосвязанных понятий, используемых для описания этой области, вместе со свойствами и характеристиками, классификацией этих понятий, по типам, ситуациям, признакам в данной области с учетом законов протекания процессов в ней.

Концептуальная (содержательная) модель – это абстрактная модель, определяющая структуру моделируемой системы, свойства её элементов и причинно-следственные связи, присущие системе и существенные для достижения цели моделирования.

Чаще всего концептуальная модель базы данных включает:

- описание информационных объектов или понятий предметной области и связей между ними;
- описание ограничений целостности, то есть требований к допустимым значениям данных и к связям между ними.

Логическое моделирование баз данных

Логическое (дatalogическое) проектирование – создание схемы базы данных на основе выбранной конкретной логической модели базы данных.

Логические модели баз данных подразделяются на: иерархические, сетевые и реляционные.

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

Иерархическая модель данных

Модель данных, в которой

для подчиненной записи

может существовать только одна исходная.

Графически представляется деревом, в котором связи

«один ко многим» распространяются только

в одном направлении – сверху вниз

от корневых вершин.

Эта модель данных наиболее проста в реализации

средствами СУБД, однако ее использование приводит к

избыточности и нарушению непротиворечивости.

Недостаток устраняется использованием

записей виртуального (фиктивного) типа.



Сетевая модель данных

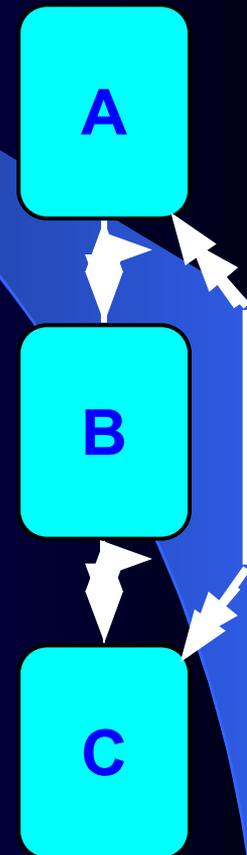
Модель данных, в которой

для подчиненной записи

может существовать более одной исходной.

В общем случае допускает связи между записями типа «многие ко многим», однако большинство сетевых СУБД автоматически поддерживают только связи типа «многие к одному».

При использовании таких СУБД связи типа «многие ко многим» должны быть представлены двумя связями типа «один ко многим» и «многие к одному» .



Реляционная модель данных

Модель данных, определяемая

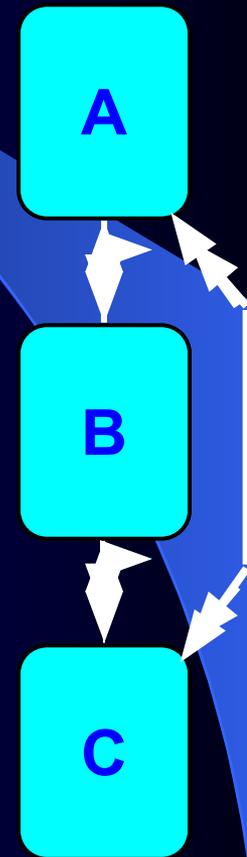
совокупностью плоских (двумерных) таблиц, где каждая таблица – некоторое отношение, состоящее из кортежей.

Понятие «кортеж»

в терминологии реляционной модели данных

аналогично понятию «запись»

в иерархических и сетевых моделях.



Физическое моделирование баз данных

Физическое проектирование – создание схемы баз данных для конкретной СУБД.

Специфика конкретной СУБД может включать в себя ограничения на идентификацию (наименования) объектов базы данных, ограничения на поддерживаемые типы данных.

Кроме того, специфика конкретной СУБД при физическом проектировании определяет выбор решений, связанных с физической средой хранения данных (выбор методов управления внешней памятью, разделение БД по файлам и устройствам, выбор методов доступа к данным), создание индексов.



Кафедра «Инновационные технологии наукоемких отраслей»

**ПРИГЛАШАЕМ
К СОТРУДНИЧЕСТВУ !**



**111250 г. Москва, ул. Красноказарменная, д.17. Ауд. А-238
Тел./факс: (495) 362-79-84**