

Томский политехнический университет
Юргинский технологический институт

Кафедра информационных систем

Разработка программных приложений.

Лекция 1. Проектирование. Современные парадигмы

Понятия программного обеспечения

- Программирование – это сфера действий, направленная на создание программ.
- Программа – это последовательность команд компьютера, приводящая к решению задачи.
- Приложение – это программная реализация на компьютере решения задачи.
- Программное обеспечение (ПО) - это программные продукты и техническая документация к ним.
- Программный продукт (ПП) – это комплекс взаимосвязанных программ, предназначенный для реализации определенной задачи массового спроса.
- Программы являются критерием развития вычислительной техники.
- Проблемы, возникающие при создании программных продуктов:
 - Быстрая смена вычислительной техники и алгоритмических языков.
 - Не стыковка машин друг с другом (VAX и IBM).
 - Отсутствие полного взаимопонимания между заказчиком и исполнителем к разработанному программному продукту.

Классификация и сравнение языков программирования

ЭВМ исполняет программу в машинных кодах.

Составляют программу люди на удобном для себя языке.

Различают языки:

- низкого уровня (машинно-ориентированные),
- высокого уровня (на зависят от типа ЭВМ).

Языки высокого уровня бывают:

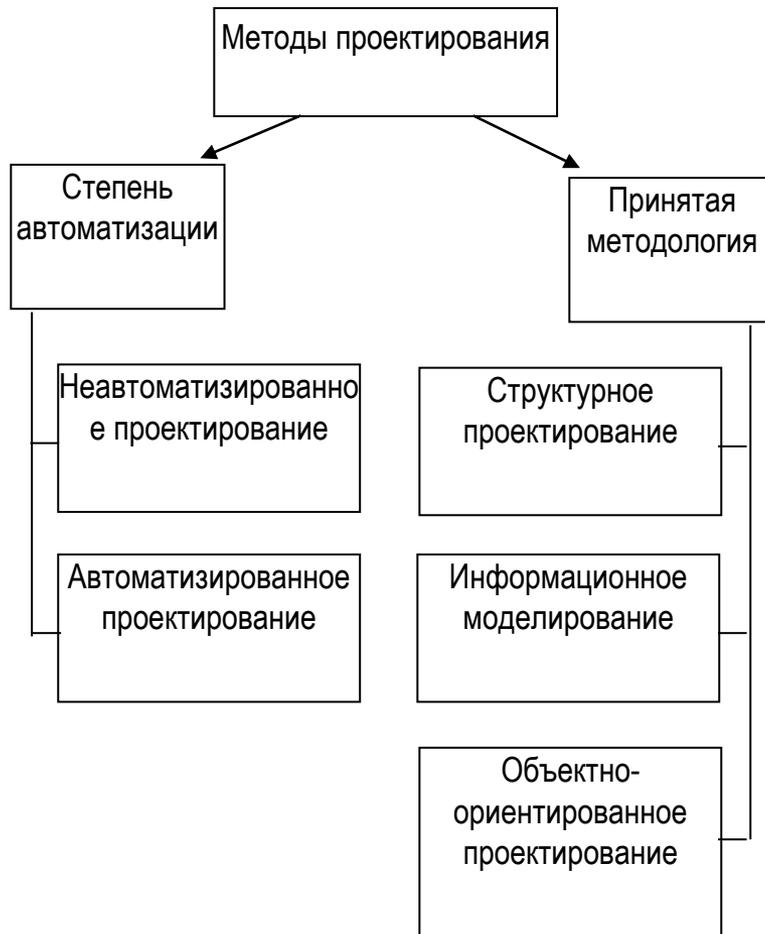
- процедурно-ориентированные (Паскаль),
- проблемно-ориентированные (MathCAD),
- объектно-ориентированные (C++).

Трансляторы переводят программу с языка программирования в машинные коды.

Виды трансляторов:

- Интерпретатор – построчный перевод и исполнение,
- Компилятор – перевод всей программы с последующим исполнением.

Классификация методов проектирования программных продуктов



Неавтоматизированное проектирование алгоритмов и программ используется при разработке небольших по трудоемкости и структурной сложности ПП.

Автоматизированное проектирование используется в крупных фирмах при разработке определенного класса ПП большого коллектива разработчиков.

Структурное проектирование – это последовательная декомпозиция, целенаправленное разбиение на отдельные составляющие.

Структурное проектирование включает в себя:

- нисходящее проектирование ("сверху вниз"),
- модульное программирование,
- структурное программирование (кодирование).

Информационное моделирование – построение комплекса взаимосвязанных моделей данных.

Объектно-ориентированный подход основан на:

- выделении классов объектов;
- установлении свойств объектов и методов их обработки;
- создании иерархии классов, наследовании свойств объектов и методов их обработки.

Каждый объект объединяет данные и программу обработки этих данных и относится к определенному классу.

Основная цель ООП - преодолеть следующие недостатки проектирования "сверху вниз":

- - недостаточное внимание к структурам данных,
- - слабая связь структур данных с процессами их обработки.

Этапы создания ПП

1. Составление технического задания на программирование.

На этом этапе требуется:

- определить тип ОС - MS DOS, Windows, Windows NT,
- оценить необходимость сетевого варианта работы программы,
- определить необходимость разработки программы,
- обосновать необходимость работы с БД под управлением СУБД,
- выбрать методы решения задачи,
- разработать обобщенный алгоритм решения комплекса задач и структуры данных,
- определить требования к интерфейсу пользователя.

2. Технический проект.

На этом этапе выполняется:

- разработка детального алгоритма обработки данных,
- определение состава общесистемного программного обеспечения, включающего базовые средства (ОС, модель БД, электронные таблицы),
- разработка внутренней структуры ПП, образованной отдельными программными модулями,
- выбор средств разработки программных модулей.

3. Рабочая документация (рабочий проект).

На этом этапе осуществляется:

- разработка программных модулей,
- программирование или создание программного кода,
- отладка программного продукта,
- испытание работоспособности программных модулей и базовых программных средств,
- подготовка *контрольного* примера, для проверки соответствия ПП заданию,
- создание эксплуатационной документации на программный продукт:

1. *Описание применения;*
2. *Руководство пользователя;*
3. *Руководство программиста (оператора).*
4. *Обучающей системы (для ПП массового применения)*
4. Ввод в действие.

Готовый программный продукт сначала проходит *опытную эксплуатацию*, а затем сдается в *промышленную эксплуатацию*.

Структура программных продуктов

Архитектура ПП представляет собой совокупность и взаимосвязь программных модулей.

Модуль – это самостоятельная часть программы, имеющая определенное назначение и обеспечивающая заданные функции обработки автономно от других программных модулей.

Программный продукт обладает внутренней структурой. Структуризация программы выполняется для удобства

- разработки,
- программирования,
- отладки,
- внесения изменения в ПП.

Структуризация ПП преследует следующие цели:

- Распределить работы по исполнителям, обеспечив их загрузку и требуемые сроки разработки;
- Построить календарные графики проектных работ и осуществлять их координацию в процессе создания программных изделий;
- Контролировать трудозатраты и стоимость проектных работ.



Нисходящее проектирование

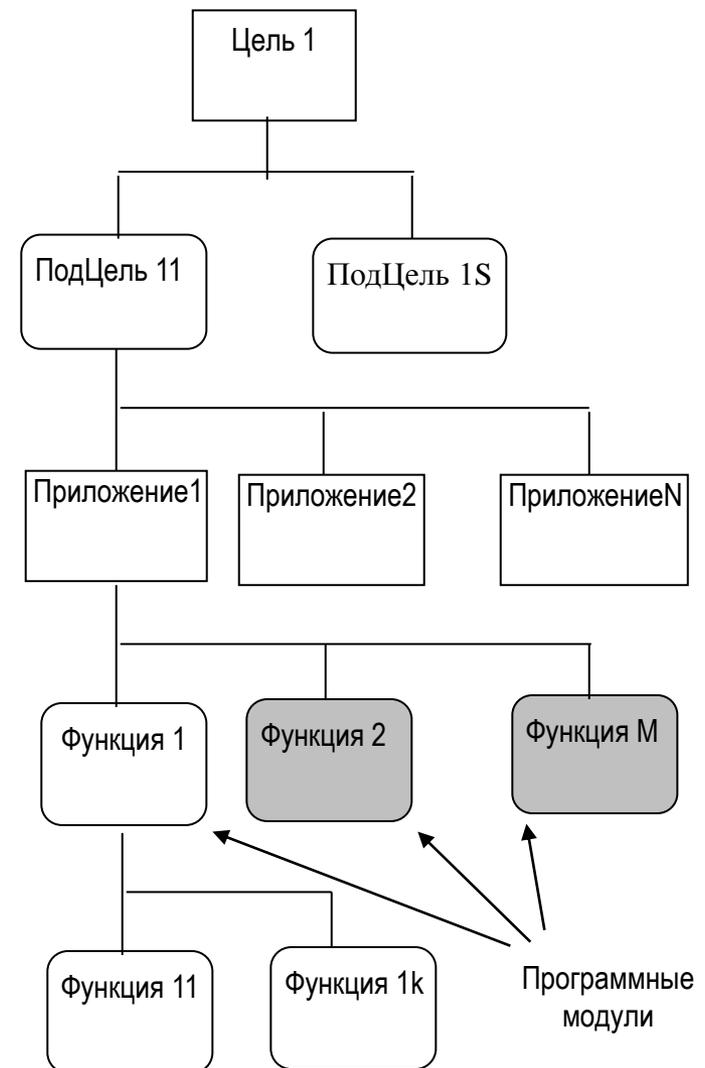
Метод предполагает последовательное разложение функции обработки данных на простые функциональные элементы ("сверху вниз").

В результате строится функциональная структура алгоритма (ФСА) приложения, в которой отражаются:

- цели предметной области (*цель-подцель*);
- состав приложений (*задач обработки*), обеспечивающих реализацию поставленных целей;
- характер взаимосвязи приложений с их основными характеристиками;
- функции обработки данных.

По частоте использования функции делятся на:

- однократно выполняемые;
- повторяющиеся.



Стили программирования

- **Событийное программирование** является развитием идей нисходящего проектирования, когда постепенно определяются и детализируются реакции программы на различные события. Под событием понимают действие над визуальным объектом: щелчок кнопки мыши, изменение размеров объекта, изменение видимости объекта и т.д.
- **В Визуальном программировании** разработка программы или приложения производится с помощью выбора визуальных компонент и размещения их на главное окно приложения, которое представляет собой интерфейс приложения. Визуальное программирование поддерживает идеологию автоматического написания программного кода. Программирование сводится к тому, что программист пишет только обработчик событий, которые связаны с соответствующими визуальными компонентами.
- **Объектное программирование** не отменяет, а дополняет принципы структурного подхода, позволяет разрабатывать и модернизировать программный код с меньшими затратами времени, распределять задачу программирования между большим количеством программистов.
- **Автоматное программирование** – подход к разработке программных систем со сложным поведением, основанный на модели автоматизированного объекта управления (расширении конечного автомата). Этот подход позволяет создавать качественное программное обеспечение для ответственных систем, охватывая все этапы его жизненного цикла и поддерживая его спецификацию, проектирование, реализацию, тестирование, верификацию и документирование.

Модульное программирование

Модуль представляет собой совокупность логически связанных элементов, предназначенных для использования другими модулями и программами.

Модуль сам по себе не является выполняемой программой – **его объекты используются другими** программными единицами (процедурами, функциями).

Модуль имеет:

- один вход и один выход – на входе программный модуль получает определенный набор исходных данных, выполняет обработку данных и возвращает один набор результатных данных, т.е. реализует принцип IPO (Input – Process – Output) – вход-процесс-выход;
- функциональную завершенность – модуль выполняет перечень операций для реализации каждой отдельной функции в полном составе, достаточных для завершения начатой обработки;
- логическую независимость – результат работы модуля зависит только от исходных данных, и не зависит от работы других модулей;
- слабые информационные связи с другими программными модулями – обмен информации между модулями должен быть по возможности минимизирован;
- обозримый по размеру и сложности программный элемент.

Каждый модуль состоит из *Спецификации* – правила использования модуля и *Тела* – способ реализации процесса обработки.

Модульная структура приложения.

Принцип модульного программирования ПП:

- определение состава и подчиненность функций,
- определение набора программных модулей, реализующих эти функции.

При составлении алгоритма необходимо учитывать:

- Каждый модуль вызывается на выполнение вышестоящим модулем и, закончив работу, возвращает управление вызвавшему его модулю;
- Принятие основных решений в алгоритме выносится на максимально «высокий» по иерархии уровень;
- Для использования одной и той же функции в разных местах алгоритма создается один модуль, который вызывается на выполнение по мере необходимости.

Структурное программирование

Основано на модульной структуре ПП и типовых управляющих (базовых) структурах алгоритмов обработки данных различных программных модулей.

Типы управляющей структуры:

- Последовательность;
- Альтернатива (условие выбора);
- Цикл.

