

Языки программирования

Теоретические основы:

Парадигмы программирования

Элементы теории трансляции

Практика программирования:

ООП подход на основе C++

Литература

- 1) И.А.Волкова, А.В.Иванов, Л.Е.Карпов. Основы объектно-ориентированного программирования. Язык программирования С++
- 2) И.А. Волкова, А.А. Вылиток, Т.В. Руденко. Формальные грамматики и языки. Элементы теории трансляции
- 3) В.Ш. Кауфман. Языки программирования. Концепции и принципы
- 4) Т. Пратт. Языки программирования. Разработка и реализация
- 5) Е.И. Большакова, Н.В. Груздева. Основы программирования на языке Лисп
- 6) <http://cmcmsu.no-ip.info/2course/>

Примеры языков программирования

- Паскаль
- язык ассемблера
- Си
- Фортран
- Ява
- Оберон
- Ада
- Питон
- Лисп

Язык программирования (определение)

- 1) инструмент для планирования поведения некоторого исполнителя
- 2) инструмент для планирования поведения некоторого устройства-исполнителя
- 3) нотация (знаковая система) для планирования поведения компьютеров

Виды ЯП:

Специализированные языки (SQL, html и др.)

Универсальные языки (Паскаль, Си др.)

Зачем могут понадобиться знания о ЯП

- для разработки новых языков программирования (соглашения, определяющие способ общения программы с внешним миром тоже можно считать ЯП)
- для понимания технологии программирования, отраженной в ЯП
- для более глубокого усвоения принципов и понятий программирования

Три составляющие языка

- синтаксис
- семантика
- прагматика

Схема рассмотрения ЯП

- Базис - это понятия и конструкции, встроенные в язык, иначе говоря, это то, что «понимает» транслятор. Базис подразделяется на скалярный и структурный.
- Средства развития — это аппарат, позволяющий добавлять в программы новые понятия (абстракции), которых не было в базисе.
- Аппарат защиты — это контроль за соблюдением ограничений, управление реакцией на нарушение объявленного поведения

Парадигма программирования

- это совокупность идей и понятий,
определяющих стиль программирования

Примеры:

- императивная парадигма
- объектная парадигма
- функциональная парадигма
- логическая парадигма

Императивная парадигма

модель компьютера (автор - математик Фон Нейман) :

- центральное процессорное устройство (ЦПУ)
- оперативная память (ОП)
- устройства ввода/вывода (УВВ)

Команды ЦПУ

- пересылки между ОП и регистрами ЦПУ;
- арифметико-логические команды
- команды управления, включающие в себя команды перехода и некоторые специальные команды (например, команда останова)
- команды ввода/вывода

Основные понятия императивных языков программирования

- Переменная – абстрагирует понятие ячейки ОП
- Операция – обобщает команды АЛУ ЦПУ
- Присваивание – обобщает команды пересылки
- Операторы управления – обобщают команды перехода
- Операторы ввода/вывода – обобщают машинные команды ввода/вывода

Объектная парадигма

- Объект = состояние + поведение
- Поведение = посылка сообщений себе и другим объектам.
- Для каждого вида сообщения существуют «обработчики», которые могут модифицировать состояние объекта и посылать сообщения другим объектам.

пример

запись $X = X + Y$

трактуется в объектной парадигме так:

объекту X посылается сообщение

«складывайся с объектом Y и измени свое состояние на новое – результат сложения»

Объекты с одинаковым поведением и
возможными состояниями объединяются в
классы

Классы образуют *иерархии*

Иерархия – это способ борьбы со
сложностью систем

Иерархия «часть/целое»



В ООП это называется структурой объектов

Иерархия «общее/частное»

Транспортное средство

— грузовое ТС

— самосвал

— бензовоз

— пассажирское ТС

— автобус

— легкой автомобиль

В ООП это называется структурой классов

Объектно-ориентированное программирование

это метод программирования, основанный на представлении программы в виде совокупности взаимодействующих объектов, каждый из которых является экземпляром определенного класса, а классы являются членами определенной иерархии наследования

Наследование

- механизм, с помощью которого один объект (производного класса) приобретает свойства другого объекта (родительского, базового класса).

Полиморфизм

- механизм, позволяющий использовать один и тот же интерфейс для похожих действий.

Общая концепция полиморфизма: **один интерфейс – много методов.**

Пример: Имеются 3 типа стека для хранения:

- целых чисел - `push_int()`
- чисел с плавающей точкой - `push_float()`
- символов - `push_char()`

Вместо трех подпрограмм управления в объектно-ориентированной программе требуется всего одна операция (один интерфейс) `push()` и одна операция `pop()`

Виды полиморфизма

- **статический** (реализуется на этапе компиляции с помощью перегрузки функций и операций)
- **динамический** (реализуется во время выполнения программы с помощью механизма виртуальных функций)
- **параметрический** (реализуется на этапе компиляции с использованием механизма шаблонов)