3. OOP. JAVA

Packages, classes, objects, this, super

Пакеты. Структура программы

- Обычно программа состоит из нескольких пакетов.
- Каждый пакет имеет собственное пространство имен для типов, объявленных в пакете.
- Верхнеуровневый тип доступен извне пакета только если он объявлен со спецификатором public.
- Пакеты образуют иерархическую структуру имен.

Элементы пакета
Package Members

- Верхнеуровневые классы и интерфейсы, объявленные во всех единицах компиляции пакета.
- Подпакеты, которые имеют свои собственные подпакеты и единицы компиляции.

Единицы компиляции
Compilation Units

CompilationUnit:
[PackageDeclaration] [ImportDeclaration] [TypeDeclaration]

- Может содержать тоьлко один тип, объявленный public и совпадающий по имени с именем файла единицы компиляции.
- Типы внутри единицы компиляции имеют доступ ко всем верхнеуровневым типам, объявленным в других единицах компиляции данного пакета а также к public типам пакета java.lang.

Полное квалифицированное имя
Fully Qualified Name

- Для примитивных типов – ключевое слово типа.
- Для именованного пакета первого уровня – простое имя этого пакета.
- Для именованного пакета уровня N – полное квалифицированное имя объемлющего пакета плюс простое имя пакета:

outer1.outer2.outerN-1.packagename

Полное квалифицированное имя
Fully Qualified Name

- Для класса или интерфейса в безымянном пакете – простое имя этого класса или интерфейса.
- Полное квалифицированное имя класса или интерфейса в именованном пакете – это полное квалифицированное имя пакета плюс простое имя класса или интерфейса:
  outerpackage1…outerpackageN-1.packagename.ClassName

Полное квалифицированное имя
Fully Qualified Name

- класс (интерфейс) – элеиент другого класса имеет полное квалифицированное имя только если таковое имеется у класса его содержащего:

  outerpackage1…..outerpackageN.OuterClass1….
  .OuterClassM.Member

- Полное квалифицированное имя массива – это полное квалифицированное имя компонентного типа с последующим []

Виды классов по объявлению

- class
- enum

Виды классов по расположению

- Верхнего уровня
- Вложенные
  - Анонимные
  - Локальные
  - Внутренние
  - Элементы классов

Экземпляр класса

Класс - это тип.

Экземпляр класса - реализация типа, объект.

Что может содержать класс (элементы/члены класса)

- Конструкторы
- Блоки инициализации
- Методы
- Поля
- Вложенные классы

Наследование

class A extends B {…}

Потомок - **всегда** частный случай предка.

Наследуются **все** элементы класса B.

Потомок может заменить предка в любом контексте.

Инкапсуляция

- Ограничение доступа к элементам класса.
- Сокрытие деталей внутренней реализации.
- **Цель**: целостность объекта.

Полиморфизм

```
class Base { void m() {...} }
class A extends Base { void m() {...} }
class B extends Base { void m() {...} }
```

Потомок может переопределить функциональность предка

```
Base base = new A();
base.m();
```

```
Base base = new B();
base.m();
```

Уровни доступа к элементам класса

- private внутри класса
- default внутри пакета
- protected внутри пакета и потомков
- public любой внешний код

default - по умолчанию

Уровни доступа к классам

- Классы верхнего уровня:
  - public default
- Вложенные классы:
  - public protected default private
- Локальные классы:
  - default