

6. OOP. JAVA

Conversions, instanceof, linking, nested classes.
Serialization

Преобразование типов между классами

- Между классами, которые находятся в отношении наследования возможно преобразование типов двух видов: **восходящее** и **нисходящее**.
- **Восходящее** преобразование **от потомка к предку**. Может выполняться неявно и иллюстрирует одну из основных особенностей ООП: потомок может заменить предка в любом контексте.
- **Нисходящее** преобразование **от предка к потомку**, который находится на любом уровне иерархии наследования.

Оператор instanceof

Первым операндом (`RelationalExpression`) бинарного оператора `instanceof` является `объект` или `null`, вторым (`ReferenceType`) – имя ссылочного типа (класса, интерфейса или типа массива).

`RelationalExpression instanceof ReferenceType`

Связывание

Linking

Это процесс определения, какой именно метод надо вызывать. Различают два типа:

- **раннее** – **на этапе компиляции** для того, чтобы определить, какой именно метод надо вызывать, исходя из имени и набора его параметров (компилятор разбирается с перегрузкой).
- **позднее** – **во время исполнения**, зная сигнатуру метода, виртуальная машина анализирует объект, на котором этот метод вызывается, чтобы определить, в каком именно классе брать определение вызываемого метода (JVM разбирается с переопределением).

Связывание

Linking

- 1) Компилятор выполнит ранне связывание, т.к. в классе **C** есть метод, соответствующий формальному типу параметра (**A**, хотя реальный тип – **B**). Т.о. ввиду раннего связывания на объекте **c** вызывается метод **f(A)**.
- 2) Поскольку у класса **A** такой метод есть – компилятор ограничивается только контролем, но виртуальная машина перед вызовом **a.m()** выполнит позднее связывание – определит, что реальный тип этого объекта – **B**, в результате чего вызывается метод, определенный в классе **B**.

Вложенные классы

<https://docs.oracle.com/javase/tutorial/java/javaOO/nested.html>

Nested Classes (вложенные классы)

- Static Nested Classes (статические)
- Inner Classes (non-static, нестатические)
 - Внутренние классы
 - Local Classes (локальные классы)
 - Anonymous Classes (анонимные классы)

Java Object Serialization

<https://docs.oracle.com/javase/8/docs/technotes/guides/serialization/index.html>

- **Сериализация** – это процесс сохранения состояния объекта в последовательность байт.
- **Десериализация** – это процесс восстановления объекта из последовательность байт.
- **Java Serialization API** – стандартный механизм управления сериализацией объектов – легок в применении, а его классы и методы просты для понимания.

Два способа сериализации

- 1) Реализация интерфейса `java.io.Serializable`
- 2) Реализация интерфейса `java.io.Externalizable`

В отличие от `java.io.Serializable`, он содержит два метода, которые необходимо реализовать (в них размещается логика сериализации и десериализации):

`writeExternal(ObjectOutput)`

`readExternal(ObjectInput)`