

JAVA Cheat Sheet 3

Basic OO

Reference types

In JAVA only primitive types are handled **by value**, because their representation does have a fixed size.

All other objects are handled **by reference**.

Thus **only** their **references are passed**, not their values → Operations on variables storing references will affect other variables holding the same reference.

Objects, classes, instances

Every JAVA program is a set of **classes**.

Every piece of code has to be part of a class.

A class is the **concept** or **blueprint** of an object.

Classes can be **instantiated**, creating a concrete object from those blueprints - an **instance**.

Every class is inherited of the class **Object**.

Attributes (fields, members)

A class can have several **attributes** describing it. Those attributes are typically declared at the top of the class and their scope is the whole class. They are called **fields** or **members**.

Advanced OO

Modifiers (class, attribute, method)

Visibility

- public → Accessible from everywhere
- private → Accessible only from within this class
- protected → Accessible from this class, all sub classes and all classes in the same package.
- package private (no keyword) → Accessible from this class and all classes in the same package

static

Can be accessed without having an instance of the class.
Holds for every instance.

final

Final **classes** may **not be subclassed**.

Final **methods cannot be overridden** by subclasses.

Final **attributes cannot change their values** once initialized.

Constructors, initializers, memory allocation

Destructors, garbage collection

Inheritance

Polymorphism

Local, inner, anonymous classes

Modularization, packages, archives