Object Oriented Programming in Java

Boris Milašinović

boris.milasinovic@fer.hr

Associate Professor at the University of Zagreb, Faculty of Electrical Engineering and Computing Zagreb, Croatia (克罗地亚)

Materials

- Public *Github* repository

  https://github.com/swu-oopj/Lectures

- PDFs of Lectures slides:
  - https://github.com/swu-oopj/Lectures/Slides
- Source code of examples:
  - https://github.com/swu-oopj/Lectures/Samples

- Homework and Lab tests
  - private repositories created for each student upon accepting an assignment using GitHub Classroom
  - Assignments' URLs would be announced in lecture slides [and/or] in the main README file.

Prerequisites

- Github account and basic knowledge of Git
  - clone repository, commit, push, checkout, …
- Knowledge of basic programming constructs (i.e. loops, functions)
  - preferably (but not necessary) in C or some C style language

Course outline (1)

- **Introduction**: Java syntax and types, source file organization and compilation, program memory organization (stack, heap), garbage collector
- **Definition of classes**: Fields, methods, encapsulation, constructors, static methods and attributes.
- **Polymorphism and inheritance**: Subclasses, inheritance, method overriding, virtual methods and dynamic dispatch.
- **Abstract classes and interfaces**
- **Exceptions**
- **Generics**: Generics in Java, type erasure, type inference, bounded type parameters, wildcards for upper and lower bounds.

Course outline (2)

- **Collections**: Lists, sets, maps and their implementations
- **Nested and inner classes, anonymous classes, iterator pattern**
- **Lambda expressions**
- **Custom classes and collections**: Use of custom classes in collections, simple and complex comparators, composite comparator as a decorator pattern example.
- **Advanced collection manipulation**: Java Stream API.

Course plan (Week 1)

- Topics per days

Date	Id	Title	# of Periods
Mon, July 8	T0	About the course, rules…	1
	T1	Introduction (how organize, compile and run simple Java programs)	1,5
	T2	Object creation (arrays, strings)	1,5
Tue, July 9	T3	Encapsulations, constructors, Variable number of arguments, static class variables and methods	4
Wed, July 10	T4	Inheritance and polymorphism	2
	T5	Abstract classes and interfaces	2
Thu, July 11	T6	Short recap	1
	T7	Exceptions	3
Fri, July 12	T8	Generics	4

Course plan (Week 2)

- Topics per days

Date	Id	Title	# of Periods
Mon, July 15	T9	Java Collections (lists, sets and maps)	4
Tue, July 16	T10	Inner and nested classes. Iterator pattern. Anonymous classes. Lambda expression	4
Wed, July 17	T11	Using custom classes in collections. Comparators. Composite comparators	2
Wed, July 17 / Thu, July 18	T12	Advanced collection manipulation: Java Stream API and collections	4
Thu, July 18	T13	Recap of T7-T12	2

Homework (50%) and tests (50%)

- Test (solving tasks, 2 hours in lab)
  - Monday, July 15: 25 points
  - Friday, July 19: 25 points
- Homework
  - Dates of assignments (deadline usually within 1-2 days)
    - Monday, July 8: 5 points
    - Tuesday, July 9: 5 points
    - Wednesday, July 10: 5 points
    - Thursday, July 11: 5 points
    - Friday, July 12: 10 points
    - Monday, July 15: 5 points
    - Tuesday, July 16: 5 points
    - Wednesday, July 17: 5 points
    - Thursday, July 18: 5 points

Homework/Test evaluation and grades

- Homework and test evaluated automatically
  - Set of tests available in advance
    - part of an assignment’s startup code
  - Undisclosed tests run after assignment's deadline on a dedicated virtual machine
    - Results available after test runs
- Grades
  - 90% – 100% = A
  - 80% – 89% = B
  - 70% – 79% = C
  - 60% – 69% = D
  - < 60% = F
- Grades and other administrative jobs: Friday, July 19