

Week 02 - Objects

Topics covered in this week

- Initialization & Cleanup
- equals vs. hashCode (light version)
- Inner classes (incl. anonymous)
- Access control
- Reusing classes
- Interfaces & Abstract classes
- Interfaces in Java 8+ (default methods)
- Interfaces in Java 9+ (private methods)

Reading material

- <https://docs.oracle.com/javase/tutorial/java/javaOO/classes.html>
- <https://docs.oracle.com/javase/tutorial/java/landl/index.html>
- <https://www.journaldev.com/12850/java-9-private-methods-interfaces>
- <https://www.journaldev.com/21095/java-equals-hashcode>
- <https://www.javatpoint.com/Garbage-Collection>

Homework

| Difficulty | Problem | Notes |
|------------|--|---|
| EASY | Create a class that holds the <i>firstname</i> and the <i>surname</i> of a person. Make it have two <i>constructors</i> , one with both <i>firstname</i> and <i>surname</i> , and one with <i>full name</i> (we assume each individual has only one <i>surname</i> and at least one <i>firstname</i> separated by spaces). Test in main class. | |
| EASY | Create a class called Tank that can be filled and emptied (let's say it has acts like a stack), and has a termination condition that it must be empty when the object is cleaned up. Write a <code>finalize()</code> that verifies this termination condition. In <code>main()</code> , test the possible scenarios that can occur when your Tank is used. | Add objects to the tank and then call the garbage collector. Solution: week2p2.zip . |
| EASY | Create a small application that demonstrates (with <code>sysouts</code>) the order in which <i>constructors</i> , <i>local variables</i> , <i>fields</i> , <i>static blocks</i> are initialized / called - consider using <i>superclasses</i> as well. | |

Kahoot