

Advanced Aspects of Object-Oriented Programming (SS 2015)

Practice Sheet 0

Date of Issue: –
Deadline: –
(before the lecture as PDF via E-Mail)

Exercise 1 Java Programming – The Basics

We want to give you an impression what we think you should already know. You should understand the concepts mentioned in the exercise and be able to implement the task in Java.

a) Implement classes and interfaces for the following szenario:

We want to model animals and their relation to persons.

- Animals have a species (modeled as a `String`) and a sex.
- The sex may be male, female or intersex.
- A pet is an animal that has a name and you can play with it.
- A person has a name and a date of birth (use `java.util.Date`).
- A person can have multiple pets.
- A dog is a pet.
- When playing with a dog it wags its tail (output to console "`<<name>>_wags_<<his/her/its>>_tail.`" with `<<name>>` replaced by the dog's name and the pronoun matching the dogs sex).

Implement the required classes in Java without the support of an IDE (e.g. on a sheet of paper). All attributes should be accessible only using getters and setters.

b) Not only pets and persons can have names. Implement an interface `Named` which gives read and write access to the name. Refactor your implementation of a) to use the interface `Named` where appropriate.

c) Implement a main method that creates the person Bob born on 1 April 1990. Bob has two dogs Pluto (a male) and Bella (a female) which are both German Shepherds. Let Bob play with both his dogs once.

Remark: For setting the date you can use the constructor of `Date` which takes three integers, one for the year, one for the month and one for the day.

Exercise 2 Java Programming – Synchronization

a) Implement a producer and consumer implementation. The producer should add the numbers from 1 to 100 to the queue (infinitely often). The consumer should take one number at a time from the queue and print it on a line to the console. Use a `java.util.LinkedList<Integer>` as the queue. This class has (among others) the methods `addFirst(Integer)`, `removeLast() : Integer` and `isEmpty() : boolean`.

b) Write a main method that starts threads for the producer and consumer and connects them using a `LinkedList`.