Department of Informatics

Professor: R. Ingold
Teaching Assistants: M. Alberti, L. Vögtlin
Assistant: R. Geissbühler

Assignment 01

1 Theory

In this exercise you have to answer the following theoretical questions. Keep your answers short and precise.

(a) How does the Object Oriented Programming paradigm differ from the Imperative one?

(b) What is a class?

(c) What is an object?

(d) What is the concept of visibility inside a class and why is it useful ? Make an example if necessary.

(e) Can private fields be accessed outside of the class (by any means) ? If yes, how?

(f) Can private constructors be accessed outside of the class (by any means) ? If yes, how?

(g) Can private methods be accessed outside of the class (by any means) ? If yes, how?

(h) How to create an object?

(i) How to destroy an object?

(j) What is the garbage collector?

2 Implementation

For this exercise you are required to submit two files: `Person.java` and `Main.java`, implemented as described below.

(a) Implement a class `Person` with private fields for name and age. Define constructor and other methods of the class as necessary.

(b) Implement a `Main` class with a `main()` method which creates a `Person` object and prints its name and age on the console.

3 Debugging

In this exercise we want to practice the use of the debugger and familiarize with the error messages of the compiler. Refer to the files `Main.java` and `Car.java` attached to this series for this exercise.

(a) Report the *exact* compiler error message and explain what is the problem for lines 5 and 6 of `Main.java`

(b) Implement two different solutions to fix the problems: once by modifying the code of `Car.java` and once by adding additional code without modifying the existing one (e.g adding new methods and make changes in the class Main).
Hint 1: there can be multiple constructors in a class.
Hint 2: static methods can be used as factory methods[1].

For reference (you don't have to rewrite them yourself, we provide the java files attached to the series):

```
1  package core;
2
3  public class Main {
4      public static void main(String[] args) {
5          Car car = new Car("MyCar");
6          System.out.println(car.model);
7      }
8  }
```

```
1  package core;
2
3  public class Car {
4
5      private String model;
6
7      private Car(String model){
8          this.model = model;
9      }
10 }
```

[1] http://www.javapractices.com/topic/TopicAction.do?Id=21

4 Project

The first step of your project consists in developing and testing a `Sprite.java` class which stores and paints the representation of an entity e.g. an image.
First, download the project data available on Moodle (`Tester.java` and `Sprite.java`). Then, create a new IntelliJ project using the provided sources ("File", "New project", "Java project with existing sources"). After this, add `OOP_2019_Lib.jar` to your project as external library. If you face any problems consider the *Get started with JAVA* tutorial available on Moodle.

4.1 Implementation of `Sprite.java`

Implement the `Sprite` class in the following way:

(a) Add `x` and `y` private fields for storing the coordinates as integers.

(b) Add a constructor allowing the creation of sprites with a specified image and coordinates.

(c) Add getters and setters for the coordinates fields and getters to return the width and the height of the image. *Hint: getWidth().*

(d) Complete the procedure `paint(Painting painting)`. The object `Painting` has a method drawImage() which you can use.

4.2 Test of `Sprite`

To test the `Sprite.java` class, you are required to complete the code in the class `Testing.java` such that an image of your choice is printed on the canvas, and its visible. Place your images in a package called `images` in the root directory of your project and mark it as resource folder[2] and consider the comments in the code.

[2] https://www.jetbrains.com/help/idea/creating-and-managing-modules.html