

Advanced Aspects of Object-Oriented Programming (SS 2009)

Practice Sheet 3

Date of Issue: 05.05.09
Deadline: 11.05.09
(until 10 a.m. as PDF via E-Mail)

Exercise 1 Inheritance and Subtyping

- a) Explain the relation between inheritance and subtyping.
b) Explain the output of the following program:

```
class Dog {
    public static void bark() {
        System.out.print("woof.");
    }
}

class Basenji extends Dog {
    public static void bark() {}
}

public class Bark {
    public static void main(String args[]) {
        Dog woofier = new Dog();
        Dog nipper = new Basenji();
        woofier.bark();
        nipper.bark();
    }
}
```

Bark.java

- c) Define the role and semantics of the `@Override` annotation in Java. Explain the relation between overriding and dynamic dispatch. *Hint: use the Java Language Specification*
d) Analyze the following program:

```
package p;

public class Person {
    void print() {}
}
```

Person.java

```
package q;
import p.Person;

public class Assistant extends Person {}
```

Assistant.java

```
package p;
import q.Assistant;

public class PhDAssistant extends Assistant {
    public void print() {}
}
```

PhDAssistant.java

Does the `print` method in class `PhDAssistant` override the `print` method in class `Person`? *Hint: look at the Java Language Specification in Section 8.4.8.*

Exercise 2 *StoJas* Extension

In this exercise, we are going to build extensions to the Java subset *StoJas* that was presented in the lecture.

- a) Extend the syntax as well as the semantics (static & dynamic) of *StoJas* to support a "`do { ... } while (...)`" statement and give a detailed explanation for the necessary adjustments.
- b) Extend the syntax as well as the semantics (static & dynamic) of *StoJas* to support interfaces and give a detailed explanation for the necessary adjustments. As for method declarations in classes, you can assume that method declarations in interfaces have exactly one parameter.

Exercise 3 Tiny Web Server

You can find the sources of a simple Java-based web server on the lecture's web page. *Important notice: The sources are meant to illustrate / practice concepts of the lecture. Students will refactor / redesign / extend parts of the software system, so currently existing deficiencies are intentional.*

- a) Download the ZIP file, unzip it, and start the server via `ant clean compile run`. Check if everything works by requesting the URL `http://localhost:8080/public_html/HelloWorld.html` using a web browser.
- b) Analyse the classes `SimpleWebServer` and `LoggableClass`. Introduce an appropriate interface and refactor the existing code of these two classes so forwarding / delegation is used instead of inheritance.