

Advanced Aspects of Object-Oriented Programming (SS 2010)

Practice Sheet 3

Date of Issue: 22.04.10
Deadline: 18.04.10
(until 10 a.m. as PDF via E-Mail)

Exercise 1 Introspection and Reflection

With the techniques of introspection and reflection, it is possible to store objects or whole object graphs in a file.

- a) Implement the class `PersistencyMangager` that supports writing and reading of arbitrary objects to and from files. Your class should implement the following Interface

```
interface PersistencyManager {
    public void writeObject(Object o, String toFileName);
    public Object readObject(String fromFileName);
}
```

`writeObject` writes the state of the object `o`, the states of the objects in the object graph reachable by `o` and the reference structure to a newly created file with the given filename. `readObject` reads an object graph from the file and reconstructs it, the return value is the object, which acted as entry in the graph during the storage.

- b) Look at the following code snippet:

```
IPersistencyManager pm;
Object o, o2;
...
pm.writeObject(o, "File");
o2 = pm.readObject("File");
```

Is the object graph referenced to by `o` equal to the one referenced by `o2`? If not, what is the difference?

- c) Look into the Java-Documentation, which possibilities are offered for storing and retrieving objects? What are the advantages and disadvantages of the different solutions? Compare them with your solution.

Exercise 2 Prototype-based Programming

The programming language *JavaScript* uses the prototype approach to create objects.

- a) Download the sources of the current Mozilla JavaScript implementation (see <http://ftp.mozilla.org/pub/mozilla.org/js>) and build the standalone interpreter `js`. On UNIX-Systems, this is done by a simple `make -f Makefile.ref`. The executable `js` is stored in a subdirectory, that is named dependent on your system architecture.
- b) Inform yourself at http://developer.mozilla.org/en/docs/Introduction_to_the_JavaScript_shell about the usage of the interpreter and about its features.
- c) Write an object-oriented JavaScript version of the University Administration System of the first lecture. Your implementation should have the following features:
1. Attributes are set by an appropriate constructor.
 2. Attributes are not accessible from other objects.
 3. The attribute `name` should be part of `Person`, the `print`-method of `Person` prints the current name. The `print`-implementations of `Student` and `Professor` should reuse the implementation of `Person`.
 4. Which advantages has the prototype-based version over the class-based one?

Exercise 3 Structural vs. Nominal Typing

Type systems for programming languages can be based on the structure or on the names of types. In nominal type systems types are considered to be equal if they have the same name, whereas in structural type systems types are equal if they have the same structure. In a structural type systems usually a type T is a subtype of S if it offers at least all “features” of S, where features are e.g. attributes and methods.

The following assignment would be legal in structural typed languages, because B has all the attributes A has plus an additional one. It is of course but illegal in nominal type systems.

```
class A {
    int x;
    int y;
};

class B {
    int x;
    int y;
    int z;
};

A a = new B();
```

Most widely used languages use a nominal type system. An example for a structural typed OO-programming language is Go (see <http://golang.org>). C++, Scala and others use a hybrid type system.

- a) What is the advantage of structural typed languages?
- b) The University Administration System is now implemented in a language with a structural type system and it is extended with two new modules. One for handling the exams and one for the handling of employes.

```
class Person {
    String name;
    void print(){...}
}

class Student {
    String name;
    int reg_num;
    void print() {...}
}

class Professor {
    string name;
    string room;
    void print(){...}
}

class Employee {
    string name;
    int reg_num;
    int salary;
    void print() {...}
}

class ...
    public void registerForExam(Student s) {...}
```

Which kind of errors can occur now?