Prof. Dr. A. Poetzsch-Heffter
Dipl.-Inform. Kathrin Geilmann

University of Kaiserslautern
Department of Computer Science
Software Technology Group

Advanced Aspects of Object-Oriented Programming (SS 2011)

Practice Sheet 7

Date of Issue: 24.05.11
Deadline: 31.05.11
(until 10 a.m. as PDF via E-Mail)

Exercise 1 Questions

a) Add one or more questions to the document at `http://bit.ly/lGbs2v`.

Exercise 2 Aliasing

a) Give two examples for dynamic aliasing, one where aliasing is desired and one where aliasing has undesired effects.

b) Define the relationship between capturing, leaking and aliasing.

c) Give a solution to the signers issue on slide 8 of chapter 4.

Exercise 3 Immutable Classes in the JDK

Select five classes from the JDK which conform to the definition of immutability as given in the lecture and explain the reason for their immutability.

Exercise 4 Confined Types

Examine the code, available with this practice sheet on the web, with respect to confinedness. The classes `ProofTreeNodeIt` und `ProofContainer` should provide the externally visible interfaces and are thus not confined.

a) Examine the class `ConfinedList`. Can we declare this class as confined? Can we modify the implementation as to make this class confined?

b) Examine the class `ProofTreeNode` and answer the same questions as above.

c) Examine the class `PTNIterator` and answer the same questions as above.

Exercise 5 Encapsulation on Object Level

Consider the class `ConfinedList` from the exercise above. The array stored in the field `data` should be part of the representation of a `ConfinedList`-object, i.e. in the we annotate it with `rep`.

```
class ConfinedList<T> extends AbstractList<T> implements List<T>, RandomAccess, Cloneable, Serializable {
...
 private /*rep*/ transient Object[] data;
 ...
}
```

a) Is it possible to annotate the remaining parts of the class `ConfinedList` , such that it conforms to the programming discipline presented in the lecture (slide 4.27)? If not, what is the problem with the implementation.

b) Which properties can the programming discipline guarantee? Compare them with the guarantees given by confined types.