Prof. A. Podelski, **Prof. P. Thiemann**,
M. Mehlmann and **S. Wehr**

Summer term 2008

Software Engineering

http://swt.informatik.uni-freiburg.de/node/94
http://proglang.informatik.uni-freiburg.de/teaching/swt/2008/

Exercise Sheet 4

2008-05-30

Exercise 1 (Design by Contract; 7 Points)
The homepage of the lecture provides the code of a class implementing stacks. Unfortunately, all pre- and postconditions as well as the invariants are missing. Please add them to the code.

Exercise 2 (Design by Contract and inheritance; (5+4) Points)
Assume the following interface, which specifies mappings from keys (of type `K`) to values (of type `V`).

```
interface Map<K,V> {
    boolean containsKey(K key);
    V get(K key);
    void put(K key, V value);
}
```

Beyond the usual requirements for the methods of `Map` (see `http://java.sun.com/j2se/1.5.0/docs/api/java/util/Map.html`), we impose additional restrictions:

- Keys equal to `null` are not supported.
- The method `get` returns `null` if there is no mapping for the given key.

(a) Add suitable pre- and postconditions to the methods of the interface.

(b) Devise two subinterfaces of `Map`. Both subinterfaces should override existing pre- / post-conditions of methods of `Map`. One of the subinterfaces should specialize the methods correctly, the other should contain one illegal method specialization. Please justify your results.

Exercise 3 (Deadlocks; 4 Punkte)
Write a Java program containing a deadlock. Explain how the deadlock occurs.

Submission: 2008-06-06, 12pm **before** the exercise session in HS 00-036, building 101.