

# Advanced Aspects of Object-Oriented Programming (SS 2013)

## Practice Sheet 9

Date of Issue: 11.06.13  
Deadline: 18.06.13  
(before the lecture as PDF via E-Mail)

### Exercise 1 Behavioral Subtyping

In this exercise we look at the `Reader` example found at the end of slide set 5 (specification and checking).

- Give a detailed explanation why the execution of method `BufferedReader.getChar()` cannot lead to an `ArrayIndexOutOfBoundsException`.
- Implement a `StringReader`. First add an interface equivalent to the `BlankReader` including the specification of the behavior. Second add an implementation of `StringReader` that fulfills the specification.
- Add a method `void unread(int c)` to `BufferedReader`. The method pushes back a single character by copying it to the front of the buffer. After the method returns, the next character to be read is `c`. Add a specification and an implementation of the method in `BufferedReader`. Is it easily possible to move the method to the `Reader` interface instead?

### Exercise 2 Concurrent Access to Shared State

Read and write operations on fields are atomic except for fields of type `long` and `double` (§17.7). Analyze the following code (taken and adapted from the book “Effective Java Second Edition” by Joshua Bloch):

```
package concurrency.mt;

import java.util.concurrent.TimeUnit;

public class StopThread {
    private static boolean stopRequested;

    public static void main(String[] args) throws InterruptedException {
        Thread backgroundThread = new Thread(new Runnable() {
            public void run() {
                int i = 0;
                while(!stopRequested) {
                    i++;
                }
                System.out.println("Thread_terminated.");
            }
        });
        backgroundThread.start();

        TimeUnit.SECONDS.sleep(1);
        stopRequested = true;
        System.out.println("Program_sould_terminate_now...");
    }
}
```

The expected behavior is that the program terminates after one second by telling the background thread to stop working. On some machines, this program does not terminate. Explain this unintended behavior. How can the program be fixed?