Prof. Dr. Peter Thiemann
Sergio Feo-Arenis
Sergiy Bogomolov

Summer Term 2014

Software Engineering

http://proglang.informatik.uni-freiburg.de/teaching/swt/2014/

Exercise Sheet 7

Exercise 1: Battleships (10 Points)

Consider the game battleships: http://en.wikipedia.org/wiki/Battleship_(game). A software engineer is commisioned to write a program that validates a ship configuration according to the rules. A configuration is valid if and only if:

1. It contains exactly ten ships:
   - One aircraft carrier (5 squares)
   - Two cruisers (4 squares)
   - Three destroyers (3 squares)
   - Four submarines (2 squares)
2. The ships are not placed adjacent to each other (there is at least one square free between any two ships).
3. The ships are straight, have no corners or indentations.
4. The ships cannot be placed diagonally.

The code should implement the following interface:

```
public enum GridState {Invalid, Valid, Viable}
public interface BattleshipValidator {
  // returns: Valid if the grid contains a valid configuration,
  // Viable if the configuration does not violate the rules but does
  // not yet contain all ships, Invalid otherwise.
  public GridState validate(boolean[][] grid);

  // adds a ship of the specified size to a grid at position (x,y)
  // with the specified orientation. Returns true if placing the ship
  // respects the rules.
  public boolean addShip(boolean[][] grid, int shipSize, int x, int y,
      bool isHorizontal);
}
```

- Provide a specification (in the form of requires and ensures clauses) for each of the interface methods. Try to be as formal and precise as possible.

Exercise 2: Roman numerals (10 Points)

Consider code that converts between arabic and roman numerals:

```java
public interface RomanNumeralConverter {
  // returns the arabic numeral representation of the input string
  // or -1 if invalid
  public int toArabic(String roman);

  // returns a string with the roman numeral representation of the
  // input number, the input number should be positive and
  // no greater than 3000
  public String toRoman(int arabic);
}
```

- Provide test cases according to the *black box* testing principle for both interface methods.
- How many test cases are necessary for each method?

Submission

- Submit this sheet *before* the lecture of Thursdays.
- Late submissions will not be accepted.
- Deadline: Thursday 11:59 a.m.