Institut for Matematik og Datalogi
Syddansk Universitet

November 20, 2017

Peter Schneider-Kamp

Introduction to Programming
12th Weekly Note (E17, Week 48)

3rd Project Qualification Assessment

The third and last assessment is only available to those, who did NOT score at least 150 points yet. The assessment will be performed as an oral exam. If you need points, please visit the following Doodle poll and indicate all possible slots on that day:
`https://doodle.com/poll/u9gspwb3qbdksi5b`

Reading for Week 48

- **Obligatory:** None :-)
- **Supplementary:** Android training "Adding Maps"
  `https://developer.android.com/training/maps/index.html`

  JavaFX; Getting Started with JavaFX "Hello World, JavaFX Style"
  `https://docs.oracle.com/javase/8/javafx/get-started-tutorial/hello_world.htm`

Lecture: Monday, November 27, 12-14 (U140)

We will learn how to use streams for input and output, both regarding files and regarding network connections.

Labs: see detailed schedule on course home page

- **Obligatory:** Build an Android app or a Swing GUI that asks the user for a positive integer n and shows a grid of n^2 fields, each labelled with a different positive integer from 1 to n^2.
- *Supplementary:* Use a random order for the integers. Let the user select two fields by touching/clicking them, and then swap those two fields (or their numbers). The game is won when all integers are sorted going top-to-bottom and lef-to-right.
- Challenge: Use only the numbers from 1 to $n^2 - 1$ and leave one field free. Then allow the user to swipe/drag neighbouring fields into the empty field. The winning condition is the one from the supplementary task.

Exercises: see detailed schedule on course home page

- **Obligatory:** Implement the BinTree ADT from the lecture, i.e., supporting the operations isEmpty, size, height, preOrder, inOrder, and postOrder using a recursive data structure BinTreeNode.
- *Supplementary:* Implement the SortTree ADT from the lecture using a design from the lecture or your own.
- Challenge: Implement the SortTree ADT using a self-balancing red-black tree (`https://en.wikipedia.org/wiki/Red-black_tree`) and let it implement the SortedSet interface from the Java collection classes.