

Softwarekonstruktion - Exercise 7

7 Simple Java Components

This exercise should be solved until Wednesday (23:59 latest), December 8th, 2010.

You have to submit your solution to your tutor by email:

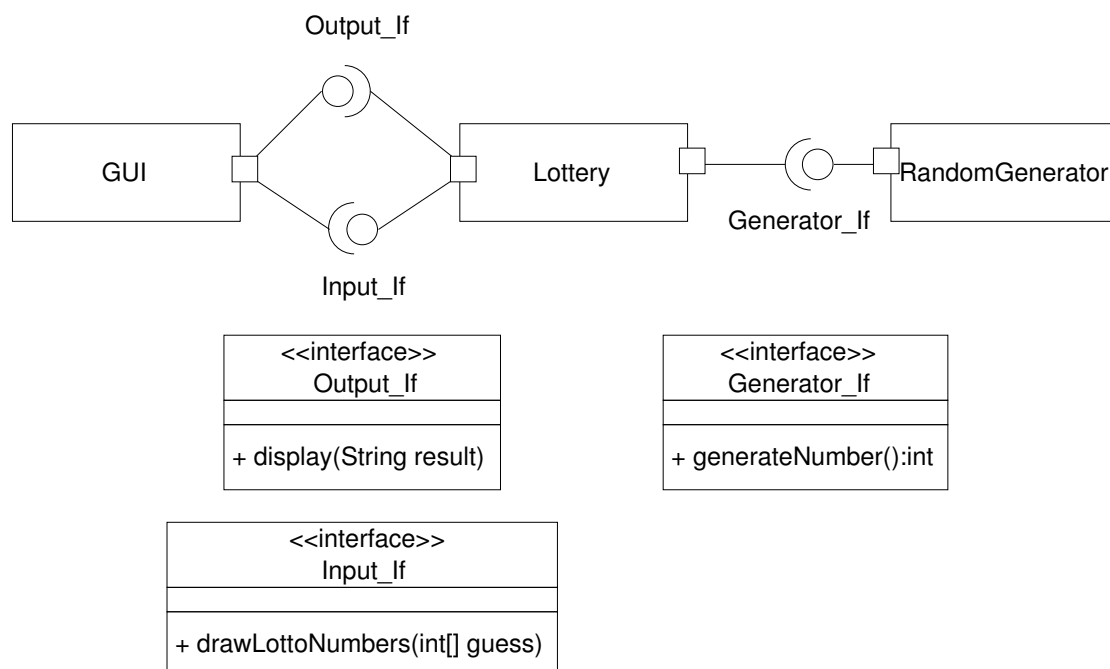
Holger Schmidt: [holger.schmidt \[at\] cs.tu-dortmund.de](mailto:holger.schmidt@cs.tu-dortmund.de)

Gregor Kotainy: [gregor.kotainy \[at\] tu-dortmund.de](mailto:gregor.kotainy@tu-dortmund.de)

You have to work in groups of two or three persons. Only one person per group has to submit a group's solution. State the names and matriculation numbers of the group members in your email and as a comment in each of your source code files.

7.1 Lottery Simulation

The following composite structure diagram and UML interface classes describe a simple lottery simulation:



- **GUI:** As soon as the component **GUI** is connected to the component **Lottery**, a user is prompted for 6 numbers between 1 and 49 (you can assume that the user only enters integer values between 1 and 49). Use the method `nextInt()` of the Java class **Scanner** (<http://java.sun.com/javase/6/docs/api/java/util/Scanner.html>) to prompt a user for input. When all numbers have been entered the drawing of the lotto numbers starts.
- **Lottery:** The component **Lottery** uses the component **RandomGenerator** to draw 6 different lotto numbers (if a number is drawn twice a new lotto number has to be generated). The drawn lotto numbers are compared to the numbers entered by the user and the component **GUI** is used to display the drawn numbers and to inform the user how many numbers he/she guessed correctly.
- **RandomGenerator:** Use the method `nextInt(int n)` of the Java class **Random** (<http://java.sun.com/javase/6/docs/api/java/util/Random.html>) to generate random integer values between 1 and 49.

Implement this lottery simulation using the approach for simple Java components presented in the lecture. To test your implementation write a Java class **Main** containing a **main** method which connects the software components.