

Assignment 8 – Part A

SE1 Proseminar, winter 2007/2008

Stefan Resmerita
stefan.resmerita@cs.uni-salzburg.at

Exercise 1

Implement the following command:

Undo

This reverses the effect of the last successfully executed command in the system, if the command is undoable. You have to decide which commands are undoable and which are not, and to explain your decision in this respect. After a successful execution, a message should be displayed at the console, including the command that has been reversed. If the execution of the Undo fails, (e.g., because the last executed command is not undoable), nothing is reversed and a corresponding message is displayed on the console, including the command that could not be undone.

Exercise 2

Implement the following command:

Redo

If the last successfully executed command in the system was an Undo, then Redo executes again the command that has been undone by the Undo. Otherwise, Redo executes again the last successfully executed command in the system, if the command is repeatable. You should decide which commands are repeatable and which are not, and you should explain your decision. The output to the console should be given similarly as for Undo.

You should specify an *undo level*, which is the maximum number of commands that can be undone during an execution of the system without performing any Redo. This number is given in the configuration file. The above commands can be issued only from the console, they are not used in scripts.

The checking package follows. Please make sure to implement the interface in this package.

```
package checking;

public interface CheckerInterface {

    /*
     * Starts operation of the system.
     */
    public void start();

    /*
     * Terminates system operation: releases all resources.
     * Make sure that this method returns (no System.exit here).
     */
    public void stop();

    /*
     * Determines execution of the command given by the string s,
     * as if s were entered by the operator at the console. Returns
     * true if the command execution completed successfully, false
     * otherwise.
     * Examples: s="PowerOn TransportRobot", s="Undo", etc.
     */
    public boolean sendCommand(String s);

    /*
     * Returns the student numbers of the authors. Each number is
     * represented as a String
     */
    public String[] getStudentIds();
}

-----

package checking;

public class CheckerImplement implements CheckerInterface{

    /* Place your implementation here */

}
```