

```
package hanoi;

import hanoi.gui.JHanoi;

/**
 * Displays the execution of the Hanoi towers problem. If no argument is given, the gui will be
 * displayed, if the number of disks is given as argument the turns will be print in the console.
 * If more than 1 argument is given, the program will terminate.
 *
 * @author Kevin Farine, Timothee Van Hove
 */
public class HanoiDisplayer {

    /**
     * Main entry point of the program that determines the display mode based on the argument count.
     *
     * @param args The argument given to the program.
     */
    public static void main(String[] args){
        if (args.length == 0) {
            new JHanoi();
        } else {
            Hanoi hanoi = new Hanoi(checkArguments(args));
            hanoi.solve();
        }
    }

    /**
     * Displays the current state of the needles for the given hanoi instance.
     *
     * @param h The Hanoi instance.
     */
    public void display(Hanoi h){
        System.out.println(h);
    }

    /**
     * Checks if the arguments given to the program are correct.
     *
     * @param args The arguments given to the program.
     * @return The number of disks stacked on the first needle.
     * @throws RuntimeException If more than one argument are given, or if the argument < 1.
     * @throws NumberFormatException If the argument cannot be represented as an integer value.
     */
    private static int checkArguments(String[] args){
        int nbDisks;
        if (args == null || args.length > 1){
            throw new RuntimeException(
                "1 argument expected, " + (args == null ? "0" : args.length) + " given");
        }
        try {
            nbDisks = Integer.parseInt(args[0]);
        } catch (NumberFormatException e) {
            throw new NumberFormatException("Cannot parse " + "\"" + args[0] + "\"" + " to int");
        }
        return nbDisks;
    }
}
```