3 Instruction – Loops and Strings

Preliminaries

Exercises, or parts, that are marked with a * are extra material for the interested student.

Remember:

- Start every file you write with a line of comment with your name(s) and the date.
- Work in pairs, if possible.
- Keep your code neat from the beginning on, with proper indentation, etc.

3.1 Guessing game

1. Write a program that plays a guessing game with the user. (S)he should guess an integer number between 0 and 50 that is chosen randomly by the computer (see below). The program allows as many guesses as necessary and gives helpful replies to each guess ("lower", "higher"). When the player guesses right, the program should output: "Congratulations, ... was the right guess!" and then terminate.

   Use `Math.random()` to provide a random number in the interval $[0, 1\rangle$. If, for example, you want the variable `secretNumber` to get a value between between 0 and 1000 (not including 1000) you can use `secretNumber = (int)(Math.random() * 100)`. The effect of `(int)` is that the double `Math.random() * 1000` is rounded down to an int.

   On the wiki, you find a starting version `GuessingGame.java` that has to be completed.

2. Change the program to also count the number of guesses and output this number, adding "You used ... guesses."

3. Change the program such that after 10 guesses it informs the player that the player has lost the game, gives the number and terminates.

4. * Challenge: reverse roles and write a program that *guesses* a number that you have in mind, informing the computer at each guess whether it was too low, too high or correct.

3.2 Codingbat exercises

The following exercises can be found on the website `codingbat.com` . Go to this site and create an account (so your work is kept over sessions).

The first five are from a previous instruction class.

1. Warmup-2: stringTimes `http://codingbat.com/prob/p142270`
2. Warmup-2: stringBits `http://codingbat.com/prob/p165666`
3. Logic-1: lastDigit `http://codingbat.com/prob/p169213`
4. Logic-1: teaParty `http://codingbat.com/prob/p177181`
5. Warmup-2: stringMatch `http://codingbat.com/prob/p198640`
6. String-2: xyBalance `http://codingbat.com/prob/p134250`

7. Given an array of (possibly negative) integers, return the sum of the longest consecutive subsequence of positive numbers. The sum of the empty sequence (in the case there are no positive elements) is 0.

8. * Given an array of (possibly negative) integers, return the sum of the consecutive subsequence that has the greatest sum.