

Companion exercises
Objektorientierte Programmierung: Wintersemester 2021/2022

No. 4, due until 22.11.2021

Task 4.1: Klaus' billboards

4 Points

```

      \ | | | | /
      (  O  O  )
|-----ooO----- ( _ ) -----|
|
| Testing your own code is important! |
|
|-----Ooo-----|
      | _ | | _ |
      | | | | |
      ooO Ooo

```

This is Klaus. Klaus likes to make billboards with important messages. Messages that appear on Klaus' billboards can have any length up to 84 characters.

We want to help Klaus with his billboards by writing a function that outputs Klaus with his latest billboard to the console.

- a) Write a function `void showBillboard(String message)` that outputs Klaus with his message to the console. Use a `while`-loop to achieve this.

Attention: You may copy the above ASCII art of Klaus. Keep in mind that, though his billboards may grow in size, his arms should not. You need to figure out where the fingers should be placed when a message is sent. You may use **more than one** `while`-loop in your code.

- b) Let Klaus make a billboard with `"Testing your own code is important"` and `"Testing your program with all kinds of different input values is important!"`. Does Klaus' billboard look the way it is supposed to?
- c) Write test cases for `showBillboard` with `" "` and any String that is longer than 84 characters.

Task 4.2: Count Count

4 Points

Implement the following functions that take a natural number as a parameter and convert it into the given format.

You are not allowed to use baked in functions that take care of said conversion.

- a) String `toBinary(int n)` that takes a positive number and returns it in binary notation.
- b) String `toOctal(int n)` that takes a positive number and returns it in octal notation.
- c) String `toTwosComplement(int n)` that takes a number between -128 and 127 and returns it in 8 bit two's complement notation.
- d) `void my_tests()` that tests every function from this task sufficiently. Write as many tests as you deem necessary and output their result to the console.

Example:

```
1  ...
2  String binaryTest1 = "1";
3  String binaryTest1Result = toBinary(1);
4  if(binaryTest1.equals(binaryTest1Result)){
5      System.out.println("Binary test 1 passed");
6  }
7  else{
8      System.out.println("Binary test 1 failed");
9      System.out.println("Expected: "+binaryTest1);
10     System.out.println("Actual: "+binaryTest1Result);
11 }
12 ...
```

Attention: Should you be having any trouble to convert your number into two's complement for testing purposes, you may use

```
Integer.toBinaryString(((<Number> & 0xFF) + 256).substring(1))
```

to convert a number into the right format.

Task 4.3: I wanna play a game

4 Points

In this task, you are supposed to implement a game with the following rules:

There are exactly two piles of pebbles with varying numbers of pebbles. Both players have to pick up pebbles from the biggest pile each round they play. Every round, each player can decide if they wish to pick up two or three pebbles from the largest pile. You can not pick up more pebbles than are within the largest pile. The game ends if one player is unable to pick up pebbles which means that both piles contain less than two pebbles. The player who was not able to pick up any more pebbles lost.

- a) Write a method `boolean canIWin(int pileA, int pileB)` that returns `true` if there is a way for the player who's turn it is to win.

Attention: Implement this method recursively: I can win if I take x from pile y and my opponent can not win.

- b) Write a method `void whatShouldIDo(int pileA, int pileB)` that recommends a turn to the player. The output should occur on the console.

Attention: You may assume that it is your turn when the function is called.

- c) Test your implementation by calling your functions with suitable values and simulate a game with at least 5 turns.

Include your tests and the simulation in your submission.