TP2 on Java Programming

Octobor 2024

1 Introduction

The objective of the first lab is to practice java operators, controlling executions, if-else, for/while loops. For this lab session and the following labs, you can use IntelliJ IDEA.

The requirement of your report:

- Submit everything in a .zip file named: `JAVA_TP2_prenom_nom.zip`;
- Include a .pdf file to answer open questions;
- Name your project folder `TP2_Prenom_Nom`, and inside this project folder, include the following:
  - The following files: `.idea`, `.gitignore`, `.iml`, `out`, and `src` (your code should be inside the `src` folder);
  - Write comments in your code specifying which question you are answering;
  - Inside the `src` folder, include:
    1. A package called `counter`, containing the following classes [70%]:
       1) `Counting.java` – 2.1
       2) `CountingAdvanced.java` – 2.2
       3) `CountingCyclic.java` – 2.3
       4) `Poker.java` – 3

Important: Write a report and submit it with code. The deadline of group1 is: 23:59:59, 10/10/2024; the deadline of group2 is: 23:59:59, 14/10/2024. Report should contains a pdf file with summary on what you have done and screenshot of codes. Zip of code should include multiple .java class files.

2 Counter

We would like to implement a class representing an integer counter, such that an object of this class is characterised by an integer value, **either positive or zero**. It should be noted that it only vary in step of 1 (both increment and decrement).

2.1 Class counting

Take the "Point" class from lab 1 as an example to create a class:

- Create a project "`TP2_Prenom_Nom`"
- In **src** folder, create a package "counter", create class "Counting", and in the class "Counting":
  - write incrementer() method, who will increase input integer by step 1
  - write decrementer() [1] method, who will decrease input integer by step 1

[1] Since it cannot be negative integer, here we will use if-else to make sure counter is always non-negative

  - write a constructor which initializes counter at value 0

- Create "main" method [2], in which, create an object of class "counting". The created object will
  - display its value
  - increment it 10 times, and then display its value again
  - decrement it 20 times, and display its value.

* You should get the display of results as: 0, 10, 0.

2.2 Improve the class "Counting" - part 1

Now we have the minimum value for counter, which is 0. What about the maximum?

- Modify the method increment(), if the (input value +1 > value_max), value will not increase, but remain at value_max
- create an object
  - display its value
  - increment it 10 times (value_max = 5), and then display its value.

2.3 Improve the class "Counting" - part 2

Now we have set the maximum and minimum of counter. What if we need to create a cyclic counter?

- Modify the method increment(), if the (input value +1 > value_max), value will be set to 0
- Modify the method decrement(), if the (input value -1 < 0), value will be set to value_max
- create an object (value_max = 5)
  - display its value
  - increment it 8 times, and then display its value.
  - decrement it 10 times, and then display its value.

3 Throwing Dice

- In the same package "counter": Create a class "Poker", it contains the following attributes and methods
- constructor [3]: initialize its integer value between 1 and 6.
- create method NewThrow(), generate a new random value between 1 and 6.
- in the "main" method, create a for/while loop to throw the dice 10 times. In each loop, call the method NewThrow(), and dispaly the new value.

[2] Or create another class "TestCount", and in the "main" method, implement the following
[3] Use "import import java.util.Random;"

4 Bonus

- Create a class "compute", it calculate the area of a triangle, given parameters: length of base and height.
- Following "compute", implement calculation of area for 3 scenarios, triangle, rectangle, and square, using switch.
- Use loops to simulate throwing a dice. A dice has 6 faces, containing value from 1 to 6. Assuming the dice has been thrown 10 times. Show the output of each throw.
- Use While to simulate: Exit when you get a value of 6. And print the number of times required to get 6. *Hint: use class Random. and Incremental Operator.
- Generate two variables, representing values obtained from two dices. Compare their value, and show which one wins.