Dalarna University
Introduction to Object-oriented Programming: 7,5hp
Exam. Course Module 2
Part: Programming Exercises

16 Oct. 2018

Contents

Individual Hand-in Assignment

Part I
Formalia

All code must be **commented** using your own words in the java doc notation. So, at whom am I writing comments for?

Answer: Think of it as explaning the code to a student which is one course module behind you!

1 Time available

See the Learn (Blackboard)-directory opening hours

2 Aid

This is an open book/note exam, which means that you can use any book, notes, or Internet references which can support your solutions. You are not allowed to communicate (in any form) with another human being.

Exercises must be solved individually.

Beware of plagiarism. Plagiarism can be defined as the use of words and ideas and works that belong to somebody else that one presents as one's own. Plagiarism is an academic crime that can lead to expulsion from a course or a programme. It is therefore important that you know how to cite and reference correctly. Therefore you have to read more under section 3, below.

3 Submitting Your Answers

BEWARE OF PLAGIARISM
By your act of submitting this exam **you also confirm** that you have **read and understood** the following: **https://www.du.se/en/library/academic-writing/** including the hyperlink and especially (at the bottom of the previous link target) 'Refero - Helps you understand what plagiarism is and how to avoid it.

- Hand in the Eclipse project(s) in **one** zip-file in Learn (Blackboard)
- The project must be named as: `YourUserName_YourTargetGrade_Exam_CM2_courseCode.zip`
  - The *target grade* indicates which grade you are targeting in this exam
  - Examples: `h18abcde_VG_Exam_CM2_ik1052.zip`, or `h18abcde_G_Exam_CM2_ik1052.zip`

4 Grading

For the grade it is:

Godkänd (G) To *pass* (G) this you must complete **every assignment** marked **[M]** (= Mandatory). In effect, this means that you have reached the target learnings outcomes according to the course syllabus. You must also answer all the questions in the reflection protocol (see the final task in this exam)

Väl Godkänd (VG) To *pass with distinction* (VG) you must first complete every assignment marked **[M]** and then continue on the additional **[*VG*]** exercises. This means that you have fulfilled the criteria for G, and you have also shown deeper knowledge and skills. You must also answer all the questions in the reflection protocol (see the final task in this exam)

Part II
Exercises

5 Mandatory Exercises for grade G and VG

0. When you submit your solutions make sure that you:
   (a) Submit all files, including the **complete** Netbeans (or Eclipse) project, in **one** zip file.
   (b) For the sake of our assessment of your work: After extraction of the zip-file, the project must be ready for execution in the Netbeans (or Eclipse) environment.
   (c) name the zip-file correctly (see section 3 above)
   (d) state your real world name or the user name at Dalarna University as the author at the top in each file

1. **[M]** This assignment is about tossing a coin with randomised outcome
   (a) Design a class called `Coin` , and create the instance variables `sideUp` and `totalTosses`
   (b) In the `Coin` class design a method called `tossNow` that simulates the tossing of the coin. When this method is called, it **randomly** determines the side of the coin, that is facing up ("heads" or "tails") and sets the `sideUp` field accordingly.
   (c) In the `Coin` class design a method called `getSideUp` which returns the value of the `sideUp` instance variable.
   (d) Additionally, in the `Coin` class, design method called `getTotalTosses` that returns the value of the instance variable `totalTosses` . (i.e. the total number of tosses since the tossing started)
   (e) Demonstrate the `Coin` class, by tossing a coin 50 times. Each time the coin is tossed, display the side that is facing up. The program should keep count of the number of times heads is facing up and the number of times tails is facing up, and display those values after each toss, as in the textbox below:

```
1.    Heads       [1,0]
2.    Tails       [1,1]
3.    Tails       [1,2]
4.    Tails       [1,3]
5.    Heads       [2,3]
...
49.   Tails       [21,28]
50.   Tails     [21,29]
Out of 50 tosses: Heads= 41% Tails = 59%
```

2. **[*VG*]** Continue on your solution of assignment #1

   (a) During the execution of your program, write the output to a textfile named *coindemo.txt*. The content should be the same as the displayed output in the textbox.

   (b) When demonstrating the `Coin` class, you **must** start the program from another class, thus create the `main` method in another class than `Coin`. Name this class `Demo`.

   (c) Describe your application in this exercise by using **UML Class** diagram(s), as presented in the course book, (chapter 6). Append it (them) in your hand-in

3. **[M]** Fill in this reflection protocol**: Students Reflection Protocol**

GOOD LUCK! :-)