Dalarna University

Introduction to Object-oriented Programming 7.5 ECTS-credits

Re - Examination

Course Module 1 (CM1)

Innehåll

Part 2:2 Programming

Remember to also complete Part 1, the Multiple Choice Questions, in case you need to.

- All exercises are solved individually and independentently by you.
- All code must be commented
- Hand in the Netbeans project(s) in Learn as a **zipfile** (that means with the *extension* **.zip** in this context) named: YourUserName_ExamModule1_.zip

What's needed to pass ?

- To pass this part of the exam with ...
  - grade G:
    - all exercises marked **[G]** must work when the teacher run them.
  - grade VG:
    - all exercises marked **[G] and [VG]** must work when the teacher run them.

Beware of plagiarism

Plagiarism can be defined as the deliberate use of words and ideas and works that belong to somebody else that one presents as one's own.

- 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 targets 'Refero'.

[G] Exercise 1

The date June 10, 1960, is special because when we write it in the following format (6/10/60), the month times the day equals the year. Write a program that asks the user to enter ...

1. a month (in numeric form)
2. a day
3. a year using two digits

The program should then determine whether the month times the day is equal to the year. If so ... it should display the message saying that " The date is magic ". Otherwise it should display the message " The date is not magic "

Use JOptionPane dialogs in the exercise

[G] Exercise 2

Write a program that asks the user to input the name of *three* different cities and then ...

a) display the citynames in ***ascending*** order

For ***example*** if the user entered "Oslo, "Stockholm" and "Berlin", the program should display Berlin, Oslo, Stockholm

Note: It must be possible to enter different citynames than theese.

b) for the ***last*** **city** in the list, after the citynames have been displayed in ascending order above, do the following :

1. display the cityname in all UPPERCASES.
2. display the first three letters and the last letter of the city.
3. let the user input a character and then *replace* all occurrences of that character with an "X".
    - In the coursebook there are a *few* String class methods listed, but there are more methods that can be used for String objects. You can figure out which method that can be suitable to use in this case by, for instance, reading in the API for the String class (Java API 8 - String class) to replace the occurrences.
4. let the user enter a character and then ...
    - ... search through the cityname and display whether the entered character from the user is in the city-name or not
      – If in fact the character is in the city, also display at what position(s) the occurance of the that character is in the city's name
      – If the character is *not found* in the city - state with the text "Didn't found character"

c) count the *number of characters in each cityname* and display the ...

- the *sum* of all characters
- the *average* number of characters

Use the Scanner class in the exercise

[VG] Exercise 3

Redo the payroll application from the book so that a worker gets his overtime paid according to this contract below.

Let the user input the following, by using dialogs with JOptionPane, ...

1. ... how many hours he/she worked for one week
2. ... the hourly pay rate.

The calculation is then done based on this contract, if ...

- ... worked hours > 40 and $\leqq$50 then 15% should be added to the hourly pay rate
- ... worked hours > 50 and $\leqq$60 then 25% should be added to the hourly pay rate
- ... worked hours > 60 then 50% should be added to the hourly pay rate

Display the gross pay, using a dialog, including the overtime that he/she gets for the week.

Read the example below !

Example:

Steven works 58 hours one week and the hourly pay rate is = 100
This will be:
40 * 100 = 4000
10 * (100*1,15) = 1150
 8 * (100*1,25) = 1000

Gross pay including overtime in this case is : 6150

Good Luck !