ECS705/ECS717

Lab Sheet 3: Methods and strings

Essential exercises:

Exercise 15.

a) Browse the `String` class and the `Scanner` class in Java API. Check those methods mentioned in the lecture notes and other available methods.

b) Write a Java program `StringTest.java` with a `main` method. This program should ask the user to input a string, and print out the following: 1. the String; 2. the length of the String; 3. change all letters to uppercase; 4. change all letters to lowercase; 5. the first character; 6. the last character. 7 remove all white spaces.

Exercise 16.

Write a Java program named `Positive.java` that reads a double value from the user. Take the following actions, depending on the value:

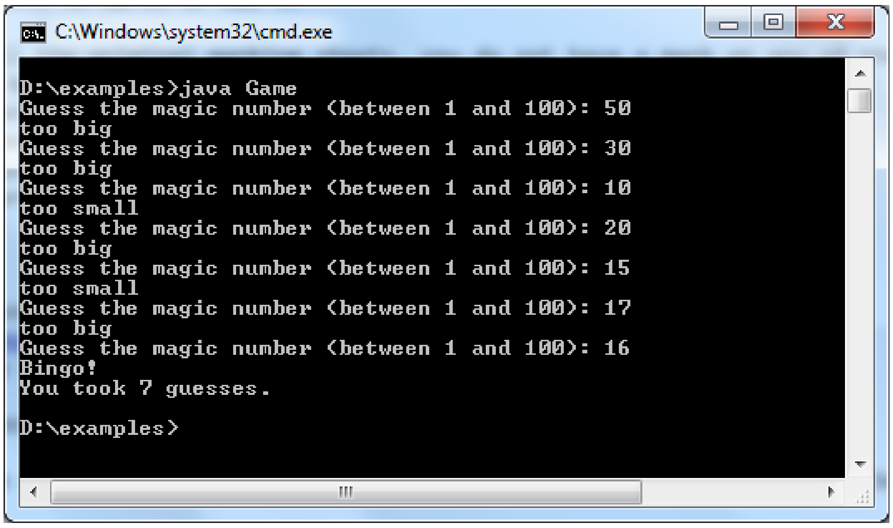
- If the value is positive, request a second value. Print the difference between these two numbers so that the difference is always positive. For instance, if the first value is 10.3 and the second is 4.1, you would print the result of 6.2. If the first value is 3.8 and the second is 13.4 you would print the result 9.6.

- If the first value read is negative, print its positive equivalent. For instance if its value is –89.6 you should print 89.6.

- If the first value is not a number, give appropriate error message.

Exercise 17.

Using methods, write a simple ***Magic Number Game***: The computer randomly generates a magic number which is an integer between 1 and 100 (inclusive). The computer asks the player to guess the magic number and tells the player whether the guess is too big, too small or it is right. Continue asking until the player enters the magic number. When the game is over, it should show the player the number of guesses s/he took. An example of the output:



```
D:\examples>java Game
Guess the magic number (between 1 and 100): 50
too big
Guess the magic number (between 1 and 100): 30
too big
Guess the magic number (between 1 and 100): 10
too small
Guess the magic number (between 1 and 100): 20
too big
Guess the magic number (between 1 and 100): 15
too small
Guess the magic number (between 1 and 100): 17
too big
Guess the magic number (between 1 and 100): 16
Bingo!
You took 7 guesses.

D:\examples>
```

- Use methods (this means you **should NOT** have just one class and write everything in the main method). You should have at least two methods besides main: one for generating the magic number, one for the game. The `main` method should be short.

- Comment your code; follow Java naming conventions for variables and methods.

- [hint] The following code generates a random number between 0 and 9 and stores it in variable randomInt. You will need to import java.util.Random from the API

```
// at beginning of program
import java.util.Random;

// in your method - tweak as required
Random randomGenerator = new Random();
int randomInt = randomGenerator.nextInt(10);
```

Exercise 18.

Download the file `CountDownExample.java` from the course web (next to the link for this lab).

a) Compile and run it.

b) Generate JavaDoc. In the directory in which you saved `CountDownExample.java` , create a directory called `doc`. (`mkdir doc` from the command line in your directory is faster than using windows to do it!). Type the following command on the command line:

```
javadoc –d doc CountDownExample.java
```

This command creates a whole bunch of different files in the `doc` directory. Open up `index.html` in your browser and take a look at what it has produced.

c) Alter the program `CountDownExample.java` so that instead of counting down it counts up. When it gets to the end of the count, it should print `All done!` (rather than `Time up!`) You also

need to change the JavaDoc comments to your own comments.
Name your program `CountUpExample.java`.

d) Generate JavaDoc for `CountUpExample.java.`

Desirable exercises:

Exercise 19.

Write a program counting how many vowels are contained in a sentence given by the user. We count ‘a’, ‘e’, ‘i’, ‘o’, ‘u’ as vowels (both upper-case and lower-case). Consider two different solutions.

Exercise 20.

Write a Java program which takes a sentence from user input, remove all non-letters, reverse it and change case of every letter. For example: Welcome to C4DM -> mdcOTEMOCLEw.

Optional exercises:

Exercise 21.

Improve the game from Ex18. When the game is over, ask the player whether s/he wants to play it again. If the player enters ‘y’, start a new game, if the player enters ‘n’, quit the game. Otherwise, tell user “Please enter y or n”.