Formulating algorithm Guessing Your Number HW2 (Bonus1.java)

1 I import a class Scanner. Scanner is in **java.util** package used for obtaining the input of the primitive types like int, double, etc. and strings.

2 Declare an integer data type to the **lowRange** variable and set to zero.

3 Declare an integer data type to the **highRange** variable and set to one hundred.

4 Initialize an integer data type to the **UserGuess** and set it to zero, declare **UserInput** variables.

5 Declare a boolean data type to the **CorrectGuess** variable and set to **false**.

6 Initialize an integer data type to the **attempts** variable and set to zero.

7 Declare **Scanner** variable **scan** which is used to specify the type of data that the user will enter.

8 Create a list with the data type of an integer named **list** and fill it with dimensions from zero to one hundred cells.

9 Create a **for** loop in which we declare an integer **i** and assign it to zero. Create a limit to which the **for** loop will work with an addition to plus one until it reaches one hundred and fills our list.

10 Create a **while** statement which should continue looping as long as **CorrectGuess**'s value will not equal true.

11 I declare the variable **midRange** to the data type integer and equate it to created formula **midRange**, which will calculate the step of one number from another, thus it will gradually decrease and the program will provide the correct answer depending on what the user clicks in the console.

12 Create a variable **UserGuess** to which I have equated **list**, which will be filled with **midRange** values. Initially there will be a value of fifty so that it is convenient for the user to start calculating his hidden number.

13 I applied a postfix increment (**++**) to the variable attempts (**attempts++**), which means that with each iteration of the **while** loop, one will be added to the counter variable, which was initially zero, and so on, one by one will be added until the **while** loop exits its loop body (until **CorrectGuess** is true). This will show the user how many attempts it took for the program to guess its unknown value.

14 I will write pair of outputs to show the user what to click to understand and what result to show.

15 I apply the **UserInput** variable to the **Scanner**, that is, this variable will take on the value that the user enters. Using the previously created **scan** variable, I say values will be accepted as input by the user.

16 Create **if** statement. If user input equal one, he/she will see output in the form of text and **while** loop will finish, because **CorrectGuess** will be equal **true**.

17 Create **else if** statement. If user input equal two, he/she will see on the console a number greater than the previous one and user will be closer to the true number that he/she guessed. This new number which is higher than the last one will be generated in the formula **midRange**.

18 Create **else if** statement. If user input equal three, he/she will see on the console a number lower than the previous one and user will be closer to the true number that he/she guessed. This new number which is lower than the last one will be generated in the formula **midRange**.

19 Declare a boolean data type to the **validAnswer** variable and set to **false**.

20 Create a **while** statement which should continue looping as long as **validAnswer**'s value will not equal true.

21 I will write pair of outputs to show the user what to click to understand and what result to show.

22 I apply the **UserInput** variable at second time to the **Scanner**, that is, this variable will take on the value that the user enters. Using the previously created **scan** variable, I say values will be accepted as input by the user.

23 Create **if** statement. If user input equal one, he/she will see output in the form of text and **while** loop will finish, because **validAnswer** will be equal **true**.

24 Create **else if** statement. If user input equal two, program will restart yourself and user will again try to guess a number.

25 Create **else if** statement. If user input equal any integer value, but program prefer three, it will show him text that he/she needs to type '2' to restart program. It will be used if user's number was wrong, therefore program make validation check to user at any time when he/she type the input.