Assignment 2: Human vs Computer

2IP90, Programming - Q1 (2023)

1 | Human vs Computer

This assignment consists of two parts. For part A, you will earn at most 7.5 points. If you want a higher grade than that, do also part B, which will earn you up to 2.5 points.

Startup

1. Create a new directory `human_vs_computer` and open it in VSCode.
2. Download files `HumanGame.java` and `ComputerGame.java` and put them in this folder. You use the file `HumanGame.java` in part A of this assignment and the file `ComputerGame.java` in part B.
3. Open both files, and fill in your names and student IDs in each file.

Part A: User guesses a number generated by the computer

For part A, work in file `HumanGame.java`.

Problem Description

Write a program to let a user play a simple number guessing game. The program asks the user to guess an integer number between 0 and 99, called the code. This code is randomly chosen by the computer. The user gets a maximum of seven guesses. After the game is finished, the program shows an overview of what happened in the game.

Generate random numbers

You are required to use the random number generator of the class `java.util.Random`. Follow the given template `HumanGame.java` and create a random number generator with the code:

```
long seed = 12323434; // actually read from input
randomGenerator = new Random(seed);
```

To generate a random number, call the method `nextInt(n)` on `randomGenerator`. This method `nextInt(n)` returns a random number between 0 (included) and n (excluded). For example, after running the following line of code, `n` has a value that is ≥ 0 and < 5:

```
int n = randomGenerator.nextInt(6);
```

Although the numbers that you generate in this way seem totally random, the sequence of generated "random" numbers is completely determined by the starting value, or seed, that you used to create the random generator. We use this seed to be able to properly test your program automatically with Momotor.

Input

Your program should ask the user first to enter this seed. Furthermore, the user inputs their guesses as integer numbers. You may assume that the input is in compliance with this description. Thus when you expect the user to input an integer number, you don't have to check if the input is actually an integer number or not.

Output

- The first line of output is:
  `Type an arbitrary number`
  The program uses this arbitrary number as the seed for the random generator.
  Note: Its type is long, not int.
- Then the game starts with the line:
  `Start guessing!`
- The program replies to each guess:
  - When the guess is higher than the code:
    `lower`
  - When the guess is lower than the code:
    `higher`
  - When the player guesses right:
    `Good guess! You won.`
- Furthermore, when the maximum number of seven (7) guesses has been reached without a good guess, your program prints:
  `No more guesses, you lost.`
  This will be printed after the `lower`/`higher` reply to the last guess.
- Afterwards, the guessing history is printed.

Guessing History

The guessing history consists of:

1. The number of guesses followed by the string `guesses:`.
2. For each guess, a line that shows how close your guess was. The line consists of 100 characters: 98 or 99 dots, an `X` at the position of the guess, and a `|` at the position of the code. For example, when the code is 77 and the guess is 33, the line will be:

```
.................................X...........................................|......................
```

If the two positions coincide, print only an `X` at this position; don't print a `|` on that line at all. This is the only case in which 99 dots are printed.

Example Runs

In the example runs below, we indicate user input with a `>`. This symbol is not actually part of the input; the user doesn't enter it. Nor is it part of the output; your program doesn't print it.
Example Run 1: Player wins

```
Type an arbitrary number
1234567890
Start guessing!
33
higher
66
higher
99
lower
88
lower
77
Good guess! You won.
5 guesses:
.................................X...........................................|......................
..................................................................X..........|......................
.............................................................................|.....................X
.............................................................................|..........X...........
.............................................................................X......................
```

Example Run 2: Player loses

```
Type an arbitrary number
>23444
Start guessing!
>5
higher
>10
higher
>15
higher
>20
higher
>25
higher
>30
higher
>35
higher
No more guesses, you lost.
7 guesses:
.....X............................................................................................|.
..........X.......................................................................................|.
...............X..................................................................................|.
....................X.............................................................................|.
.........................X........................................................................|.
..............................X...................................................................|.
...................................X..............................................................|.
```

Example Run 3: Player wins in seven guesses

```
Type an arbitrary number
>12
Start guessing!
>12
higher
>88
lower
>24
higher
>76
lower
>36
higher
>64
higher
>66
Good guess! You won.
7 guesses:
............X.....................................................|.................................
..................................................................|.....................X...........
........................X.........................................|.................................
..................................................................|.........X.......................
....................................X.............................|.................................
................................................................X.|.................................
..................................................................X.................................
```

Part B: Computer guesses a number thought of by the user

For part B, work in file ComputerGame.java.

Problem description

Write a program that lets the computer play a simple number guessing game. The program asks the user to think of an integer number between 0 and 999 (yes, 999, not 99), and the computer tries to guess it.

Input

Once the user had thought of a number to guess, again called the code, they input the text "go". After each guess made by the computer, the user answers "higher", "lower", or "guessed", depending on whether the computer's guess was, respectively, smaller than, greater than, or equal to the code.
You may again assume that the input is in compliance with this description. There is no reaction required when the users enters text that is not in the description.

Output

• The first line of the output is:
Think of a secret number not smaller than 0 and not greater than 999. Type 'go' when you have one.
• After the user has entered "go", the program outputs a number as the first guess, and continues with guesses after the user's reaction.
• The user replies to each guess: – When the computer's guess is greater than the code:

```
lower
```

– When the computer's guess is lower than the code:

```
higher
```

– When the computer guesses right:

```
guessed
```

• Furthermore, when the maximum number of ten (10) guesses has been reached without a good guess, the program prints:

```
I give up
```

• However, this should not happen! Your program should always find the code within the maximum number of ten (10) guesses.
• In this version, there is no recording or printing of the guessing history.

Example run

There is only one example run: The computer always wins! As before, we indicate user input with an >.

```
Example run 1: Computer wins, secret code = 13
Think of a secret number not smaller than 0 and not greater than 999. Type 'go' when you have one.
go
500
>lower
250
>lower
125
>lower
62
>lower
31
>lower
15
>lower
7
>higher
11
>higher
13
>guessed
```