

Dice game

The following project is a small program where you are able to input your name and be represented as a player of the game. You are then generated a personal die that will be rolled after you guess what number it will have.

You may choose how many rounds you wish to play and each round the die will be rerolled so you can have a fresh guess. In the end the program will exit and count the total amount of points that you got, one for each correct guess.

Structure

The program is divided into three classes. Die, player, main.

The Die class

simulates the die and can have a current value and a max value.

The class has a roll method which sets the current value to a random integer between 1-6 as a result from the random library.

The class also uses a getter for the current value so it can provide other classes with its current value.

The constructor takes a maxdicevalue. If one wanted to, one could input a larger number and the guess would be harder. E.g. a 20-sided die.

The private random rand = new Random makes it so that only through the setter can one use this variable.

The player class

simulates a player of the game and has a name string variable, points/score integer variable and dice variable of the Die type from the Die class.

The constructor takes only a name since score is increased by calling the increase score method, and it should always start at 0 points.

The name variable has a getter to display the name of the player that was input. As does the points variable.

The rollDice method rolls the dice that was created/instanced by the addDie method further down. It calls the method of the instanced dice to roll the dice and assign a value between 1-6 as listed in the Die class.

There is also a getter for current value of instanced dice. Calling a getter from Die class.

The main class

Here we declare a new instance of scanner class for future input from player.

We also declare two variables that help us run the main gameplay loop, rounds & maxrounds. Rounds will increase with each loop until it has reached the same value as maxrounds. At that point the while loop will exit.

We ask player for # of rounds and use nextInt to gather the int.

Next we consume a line so we don't skip any inputs.

A name is asked for and creates an instance of player1 using the player class.

Following this a 'personal' die is created when we call the addDie method from the player class.

Now the main loop is started, as we have a value for both maxrounds and rounds has been declared as always starting at 0.

Rounds are increased by one and we ask the player to guess a number between 1 and 6.

We record his guess in usrGuess variable and then call the roll dice method to create a value to compare with. The values are compared and if the player guessed correctly we call the getpoint method from the player class to increase the player's score.

If not, we display the correct number and restart the loop. The loop logic checks if rounds has reached the value of maxrounds and we start again if that is not the case.

After the set amount of rounds has been played out the loop exits. A message is printed displaying player name and their total score.

And finally we close out the scanner

Henrik Bergman
MT24