

Tortoise And Hare Race: Java Simulation of a Race Game

Introduction

The tortoise and hare race is probably one of the most famous stories from our childhood. The story is about an arrogant hare competing with a slow tortoise. However, the hare, believing he could easily beat the slow tortoise, falls asleep halfway through the race. Meanwhile, the diligent tortoise slowly crawls to the finishing line and wins.

This time, we will design a program to simulate a tortoise competing with the hare. Unlike the original story, in this Java simulation program, the hare won't always lose because he falls asleep halfway. Instead, the simulation of the tortoise and hare race has unique rules for how they will advance or fall behind. The race ends when one of the players reaches position 50 on the track.

Program Organization

The program is designed using a while loop. In each iteration, the movements of the tortoise and the hare are updated based on two random numbers, one for the tortoise and one for the hare. The random number is used to simulate the probability of different movements. For example, there is a 50% chance for the tortoise to perform a fast plod, which happens when the random number is between 1 and 5. The loop ends when one of the players reaches the finishing line.

The program also takes account into that when the players fall behind due to certain movements, they won't be allowed to go past the starting line. By using an if statement, we make sure that players always restart from position 1 if they have a slip. Similarly, both players won't go beyond the finishing line when performing a fast plod. I also use an if statement to control that. Moreover, the program displays some results. If the tortoise wins, display "TORTOISE

WINS!!”. If the hare wins, display “HARE WINS!!”. If the race is a tie, display “IT’S A TIE!!”.

At the beginning of the race, display “AND THEY’RE OFF!!”.

Major Data Structures

One set of data variables is the tortoise and hare position. They are used to track the positions of the two players in each iteration. Another important variable is the random number. Using the Random class in Java, two random numbers are generated in each iteration, one number for the tortoise, and another for the hare, to simulate different movements. The race track is visually represented by constructing a string that shows the current position of each player.

Other Considerations

After reading the design of this program, I realized that a loop is necessary to simulate the progression of the race. In the beginning, I also considered using a for loop. However, when I tried to write the conditions, I found out that a for loop is more appropriate when we know the number of iterations in advance. After analyzing this problem more closely, I decided to use a while loop instead. This way, the loop will continue until one of the players reaches the finishing line.

Conclusion

In this project, I learned how to simulate real-world scenarios using random numbers. It reinforced my understanding of control structures – loops and conditions. It is a good practice to apply the knowledge we learned to real-world situations and carefully construct the correct logic for coding.

Future Improvement

One improvement I could think of is to build a graphical user interface for displaying. It will offer a better user experience. So it will be interesting to learn how to build GUI using Java.