Introduction to Programming: Mini-Project 1

Following the American Psychological Association’s Guidelines

Markham M. Shofner

Johns Hopkins University

Introduction to Programming: Mini-Project 1

Design and Analysis:

1. General program design.
   1. The program is organized following the principles outlined in the first portion of modules in “Intro to Programming using Java”. The functionality of the program is written in an object-oriented manner, defining classes and instances, where the data contained therein in manipulated and managed by methods within those structures.
   2. The major data structures used are:
      * The main() method, which initiates an instance of the project class, and triggers the raceController() function.
      * The raceController() method, which initiates the contestants, the track, and manages the state of the race.
      * The contestants (Hare and Tortoise classes) which return moves based on random numbers.
      * The Track class, that keeps track of the Boolean status of the race, the winning condition, the position of the contestants, and can also display the current state of the race track.
2. Alternative approaches considered and reason for rejection:
   1. I considered using some sort of Contestant superclass that the Hare and the Tortoise would both branch off of, but ended up not using this option because the inheritance implementation become more complicated than I believe the situation needed the code to be.
3. Things learned from the project, and things that could have been done differently:
   1. Finding a way to properly manage the state of the program was a critical win in the early stages of writing the program (before getting into the deeper more specific parts of the code). There are probably more elegant ways to manage program state, and I look forward to building a more mature understanding of these nuances.
   2. I think I gained a better understanding of where certain functionality should reside within a program (track functions in the track class, move functions in the contestant classes, etc…) but at times it felt like functions could reside in more than one place (for example the display track function) but that it comes down to the coder’s decision of where is the most correct place for a function to be (and where it will be easiest to track down, expand, and debug in the future).

References

Schildt, Herbert (2014). Java The Complete Reference – Ninth Edition.