**MiniProject1 Report**

Nathaniel Foster

Department of Engineering, Johns Hopkins University

EN.605.201.84.SP20: Intro to Programming Using Java

Dr. Jamal Simms

March 9, 2020

General Program Design:

The program is organized into different classes defining various objects involved in the race. Originally it was written as different files which were part of the default package and then put into a single file for submission. The classes created were ***MyRace***, ***Race***, ***Racer***, ***Hare*** and ***Tortoise***. ***MyRace*** was renamed to ***OneFileOneRace*** for submission.

***MyRace*** contains the ***main()*** method and made calls to the other class files. All user inputs were taken from the ***main()*** method. It allows the user to perform races of varying lengths (input by the user) between the hare and the tortoise until the user decides he or she is done performing races. ***MyRace*** prompts the user for a race length, simulates a race, displays the race in progress, shows a brief history to race if prompted to, and lastly repeats to perform another race if prompted.

***Race*** defines an object of type ***Race***. A ***Race*** object has two racers which are the ***hare*** and the ***tortoise***, an indication of elapsed time or progress made it in the race, ***raceProgress***, and a total length of the race***, raceLength***. The total length of the race had to be stored for each race since it the race length is input by the user. ***Race*** also contains methods for simulating a race to determine a winner, showing the progress of the race, and the function to show the history of a simulated race.

***Racer*** is the parent class of both ***Hare*** and ***Tortoise***. It contains the data that is shared between every racer (in this case the hare and the tortoise) but is specific to an individual racer and not a ***Race*** object. This data includes current position in the race, ***racePosition***, a ***name***, a history of the racer’s moves and position in the race, ***raceHistoryMoves*** and ***raceHistoryPositions*** respectively. The racer’s history data had to be stored in ***ArrayList*** objects since the race could be of varying length.

Both ***Hare*** and ***Tortoise*** extend the ***Racer*** class. These children classes do not contain any further data but do name themselves accordingly on concatenation. The ***Hare*** and ***Tortoise*** classes define methods for the different movements of the hare and the tortoise.

Alternative Approaches:

The first iteration of the design only contained a class similar to ***MyRace*** that would contain the ***main()*** method and the class ***Racer*** to define the participants of the race***.*** It became quickly evident that a third class ***Race*** would be needed to store all the status data that was quickly making the ***main()*** method difficult to read.

The next iteration was significantly closer in design to the final iteration but it attempted to define movement patterns for both the hare and tortoise depending on a stored racer type. Again this quickly became overwhelming to the ***Racer*** class to define seemingly duplicate methods within the same class. Thus, the decision to make children classes to the ***Racer*** class was made in order to separate the two racer types.

Takeaways:

While design was primarily focused on class design. This project was also an exercise in control structures using both ***if*** and ***switch*** decision statements as well as numerous looping structures controlled by variables of variable length. This made the ***ArrayList*** objects the perfect tool for storing a large part of the ***Racer*** data.

The most difficult hurdle was actually designing the display of the race progress in the terminal window. If doing this again, I would create a display function that perhaps leverages some kind of GUI to display race progress rather than displaying progress from the terminal. Another feature I would also like to add is a variable number of racers and the types of racers.

Additionally I would change my code design so that only the main function outputs any outputs to the terminal unless an error condition is met. This practice to me seems more robust. I would also like to explore the idea of trying to make better use of overriding the ***toString()*** function in order to better enhance some of the readability of the code. This change however may also produce limited returns on investments since the object data for ***Racer***, ***Hare***, ***Tortoise***, and ***Race*** are displayed in many different ways.

Citations/References:

1. Java, Ninth Edition

Herbert Schildt - 2014