Steven Eisemann

9/10/18

Homework03 Design Reflection

I created Horse, Race, and RaceStrategy classes. In my Horse class, I have member variables that store information such as name, max speed, racing strategy, etc., as well as the associated getters and setters. This allows me to add Horse objects to the race instead of the individual data pieces, which I feel helps keep things more organized and understandable. The Race class keeps track of the horses in the race before and during it. The RaceStrategy class in an abstract class that contains the abstract method update, and each of the three horse racing strategies extend this class. My loop control takes place in the race class in the update method. Essentially, each iteration of the loop moves the horses forward one second in time, and loops until there is a horse that has run 10 miles. I used a HashSet to manage my horses, since I did not need to worry about data management in the Race class – it is all taken care of on the Horse side.