Aidan Lewis

Software Engineering

Homework 3

Reflection

To begin, I decided my highest-level class would a Race class, in which the actual race simulation takes place. Since the horses are a part of this race, this race would be composed of horses. Any information about the horse is stored within the Horse object, so within the Race class, I simply have a list of all the horses who will be racing. The control loop for the race occurs in this Race class, since this loop should apply to all the horses. The strategy was a private member of this Horse class, since each horse in the race has a jockey with a particular strategy. Each strategy was implemented from a JockeyStrategy interface, since all will use a different implementation of the getSpeed method. When a horse is constructed, the implemented strategy is set so that whenever the horse’s update method is called, it can take into account this strategy to update the horse’s distance. The JockeyStrategy interface only contains this single method because this strategy should only really matter when calculating the speed (and thus the updated distance). The Horse class, meanwhile, has an update method for updating the horse’s location throughout a race, and a display method for outputting a horse’s progress.