

I made the decision to use an interface to implement the 3 different racing strategies from one "Strategy" interface. To me, this made sense because I needed to pass these three different classes with three different functions in the same slot to my Race class, which means they all needed to be the same type. The only thing in the Strategy classes were the useStrategy method, which calculated a horse's new position if given its current position and max speed. I also made a Horse class to store information about each horse, in addition to the Race and Main classes. The control loop for the race went in my startRace method within race, and I stored the horses in an ArrayList of Horses. This was the best way I thought to do it, since each horse was easy to access this way, and it felt more organized, as that since ArrayList held all horse information, since all the information about each Horse, including its position in the race, are all stored in the Horse class.