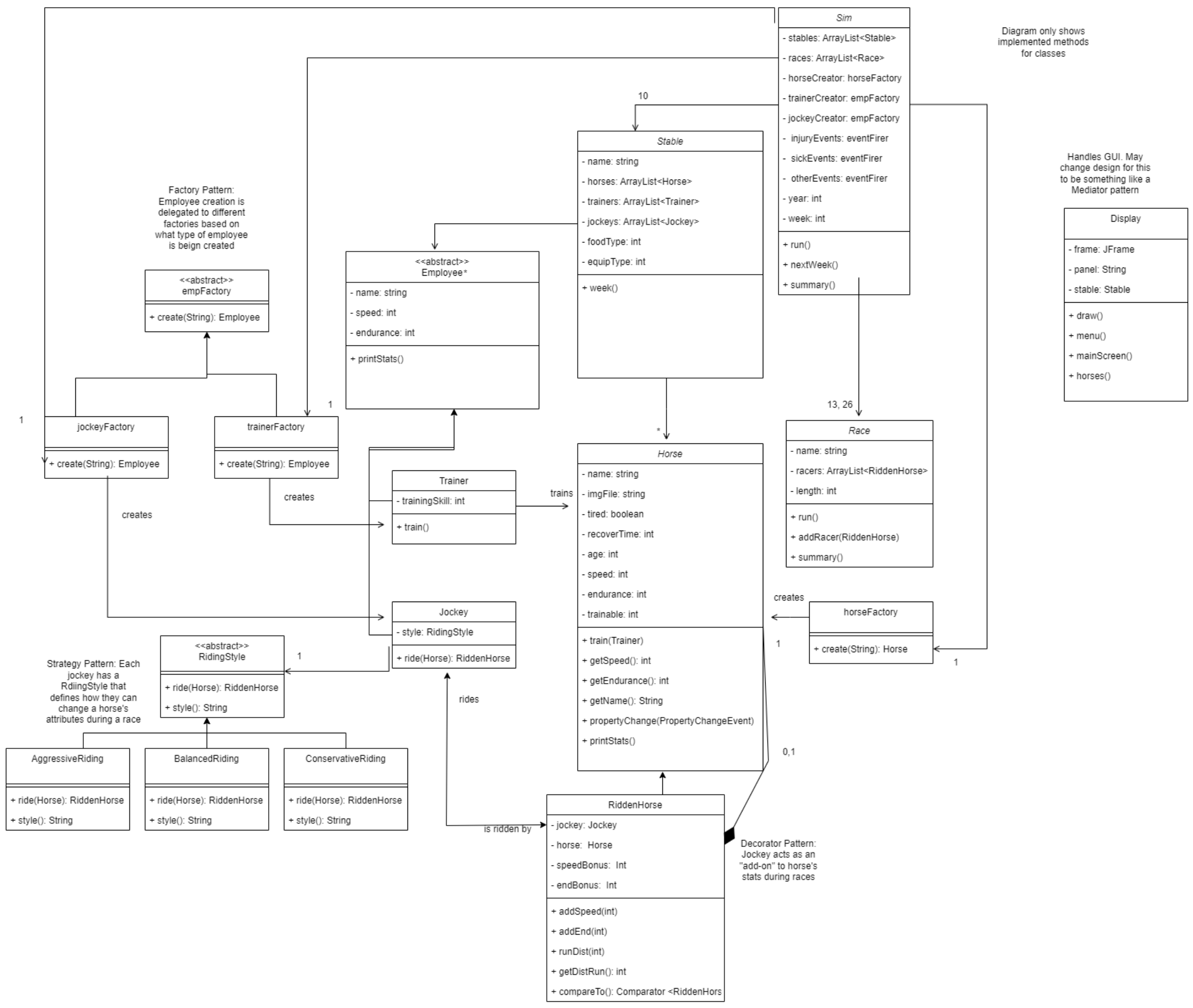
Status Summary

- Title: No Horsin' Around
- Team Members: Spencer Jackson and Jairo Guzman
- Work Done
    - Jairo - Helped by creating some of the skeleton code for the project. Wrote code for our Employee, Jockey, Horse, RidingStyle and some for RiddenHorse files.
    - Spencer - Created the logic for the Races, Trainers, and Horses, including edits to other classes needed for these to work. Also began a Display class which should maintain the GUI for the project

- Changes or issues encountered: We removed the acceleration stat for horses and employees since it seemed redundant with the speed stat. We also added a lot of getters needed for functionality in the race. The main issue encountered has been implementing the GUI, so for this check-in we just had a console app that let us test the capabilities of the simulation.
- Patterns Implemented So Far
    - Strategy: we used the strategy pattern to implement riding styles for different jockeys. When the jockey rides a horse (creating a ridden horse), it's riding style provides bonuses to speed and endurance depending on the style the jockey was assigned at initialization - aggressive gives a big boost to speed and a big detriment to endurance, balanced gives small boosts to both stats, and conservative takes away a lot of speed but heavily boosts endurance. This then affects the way horse's perform in races.
    - Factory: we used factory patterns to separate creation and implementation logic for horses and employees, which are created using different kinds of factories that we initialize with the simulation.
    - Decorator: when a Jockey rides a horse in a race, it acts as an add-on and creates a RiddenHorse, which contains both a jockey and horse. When we get stats for this new Horse subclass, we add the Jockey and Horse's stats (plus the bonus provided by the Strategy pattern). Thus we use a Decorator pattern to implement this additional behavior.

Class Diagram



Plan for next iteration

Currently we've implemented the simulation logic for races and training of horses. The main things we need to implement is the GUI, user input, events handling injuries and other random(which are handled through an Observer pattern), and saving the game. By the due date next week, all of these features should be implemented. The observer pattern should be relatively easy to implement, and after figuring out how to format the data the reading and writing for saving and loading should be fairly simple as well. The main challenge is the GUI, since neither of us have used the Swing library before. The worst case scenario is that we have to drop the GUI portion and just keep it as a console application, although we don't anticipate having to do this.