Design Pattern Explanation

For RaceWeather I chose to use the Observer design pattern as I wanted other classes to be able to choose whether they should be notified of the changes in the weather from RaceWeather. I did this by Creating two new interfaces, "Observer" and "Subject" and implemented the Observer interface into RaceWeather and implemented the Subject interface into any classes that should be notified of changes.

For RaceCar I chose to use interfaces to create certain behaviours, and depending on the WeatherType stored within weather these behaviours would return a corresponding integer. I created an interface DrivingBehaviour and then created two different behaviours which implemented this interface, CautiousBehaviour and FastBehaviour.

For Race I used the existing design principles I had already implemented for the methods, to change the car Behaviour I called the changeDrivingBehaviour method on the particular racer with the new behaviour. I also implemented getter methods for the private fields in RaceCar so I could get these values to use in the methods for Race.java.