**CS 210**

**Lab 11: Factory and Adapter Pattern**

**Lab Date:** Monday, April 20th

**Due Date**: Wednesday April 29th, 11:59 PM

**Domain:** Your client is a racehorse breeder who would like a system that keeps track of the races in which they are considering entering their horses. The client would like the ability to print a listing of all possible races, the total number of races and the average length of the races.

A race has a name, length, purse, and direction. However, your client has a mix of races in the US and races in the UK. Races in the US are run counterclockwise while races in the UK are run clockwise. In addition, there are flat races which have no jumps (e.g., Kentucky Derby) and are run on three different surfaces (dirt, turf, and synthetic). There are also jump races which have fences and are always run on turf.

Note that races in the UK have their purses represented in pounds while in the US, purses are represented in US Dollars (USD). You may assume a conversion rate where 1 pound = 1.23 USD. In addition, races in the UK are measured in furlongs while races in the US are measured in miles. A furlong is exactly 201.168 meters, or 0.125 miles so there are 8 furlongs in a mile. The data file contains information stored in the measures for the race country. For instance, the Queen Mother Champion Chase has a length represented in furlongs while the Kentucky Derby has a length represented in miles.

**Application:**

This client has asked you to develop a system that reads in races contained in a file. The data is stored in units of the country in which the race is located. For instance, the Epsom Derby is located in England so the length is measured in furlongs and the purse is in pounds. A sample input file has been provided with this lab. The general format is

FLAT, Kentucky Derby, 1.25, 20000000.0, Counterclockwise, dirt

FLAT, Epsom Derby, 12.0, 20000000.0, Clockwise, turf

You must write a program that reads in the race information and prints out the race information for each race in U.S. units (e.g., miles and dollars), the total number of races and the average race length (in miles). You know that you may have other clients so your solution must be as general as possible. In addition, you must implement the Factory and Adapter patterns. You have been provided with two existing interfaces:

* The JumpHeight interface which allows the height of jumps to be set and retrieved. (Already implemented from a prior project for hunters and jumpers for another client.)
* The PriceWinnings interface which supports retrieving a length, a purse, and also has a toString method.

You must create an abstract Race class that represents a generic horse race. This class implements the PriceWinnings interface and contains a name, length, purse and race direction. You should provide setters and getters for all attributes.

As there are two types of races, you must create two subtypes:

* The FlatRace represents a race run on a flat track and has an additional surface attribute. You should provide the getter and setter for this attribute as well as an updated toString method that includes the surface that the race is run on.
* The JumpRace represents a race run over fences. It should implement the JumpHeight interface and has an additional height attribute. You should provide the getter and setter for this attribute as well as an updated toString method that includes the height of the jumps used in the race.

You must also provide a RaceFactory class. This class must have a getRace method that is passed the information for a race and returns either a JumpRace or FlatRace instance.

Your application must print out the race information in U.S. units. Therefore, you must implement a RaceAdapter class. This class must implement the PrizeWinnings interface will wrap instances of the Race class and return values for the length of the race in miles and the purse in dollars. The toString method must also accurately print out the contents of the race in U.S. units.

Your Lab11.Driver class should contain a collection of PrizeWinnings. You should have a method that reads the races in from file and adds them to this collection. You should also have a method that prints out the race information for all entries in the PrizeWinnings collection as well as the total number of races and the average race purse.

**Hint:**

* Create a UML diagram of the solution before starting.

**What to Submit:**

* A zip file containing all source files.