GeoLoacation Project

Uses the GeoCoordinatePortable API to compare the location data from a CSV file containing all the locations and store names of all the Taco Bell's in Alabama.

The program selects the 2 Taco Bells furthest apart and returns their name and the distance between the 2 stores in miles.

Input: A CVS file of stores in the form of Latitude,Longitude,Name

Output: The name of the 2 stores that were the furthest away and the distance in miles.

```
//create file variable
const string cvsPath = "TacoBell-US-AL.csv";

static void Main(string[] args)
{
    // use File.ReadAllLines(path) to grab all the lines from your csv
    var lines = File.ReadAllLines(cvsPath);

    // Create a new instance of your TacoParser class
    var parser = new TacoParser();

    // Grabs an IEnumerable of locations using the Select command.
    var locations = lines.Select(parser.Parse).ToArray();

    //Variables to track 2 stores and a distance
    ITrackable one = null;
    ITrackable two = null;
    double distance = 0.0;
```

First, I created an interface (`ITrackable`) that would handle the details of how the information on each line of the CSV files was stored – I then made a class (`TacoBell`), that would handle the an instance of each store. Another class (`TacoParser`) was created to handle how the data was extracted from the CSV file to create the instance of the store. The `locations` reference is an array of `ITrackable's` that represent each store in the CSV file.

```
namespace GeoLocationProject
{
    public interface ITrackable
    {
        public string Name { get; set; }
        public Point Location { get; set; }
    }
}
```

```
namespace GeoLocationProject
{
    class TacoBell : ITrackable
    {
        public string Name { get; set; }
        public Point Location { get; set; }

        public TacoBell(string name, double lat, double longitude)
        {
            Name = name;
            Location = new Point(lat, longitude);
        }
```

```
    public struct Point
    {
        public double Longitude { get; set; }
        public double Latitude { get; set; }

        public Point(double latitude, double longitude)
        {
            Longitude = longitude;
            Latitude = latitude;
        }
    }
```

```
    public class TacoParser
    {
        public ITrackable Parse(string line)
        {
            var cells = line.Split(',');

            //makes sure there are enough locations to compare
            if (cells.Length < 3)
            {
                return null;
            }
            //Extract out the data from the array
            var latitude = double.Parse(cells[0]);
            var longitude = double.Parse(cells[1]);
            var name = cells[2];
            //Create the individual stroe from the above data extracted
            var store = new TacoBell(name, latitude, longitude);

            return store;
        }
    }
```

Two references that would temporarily hold the stores with the furthest distance and a variable to stare that distance.

```
for (int i = 0; i < locations.Length; i++)
{
    //Outer Location
    Point corA = new Point(locations[i].Location.Latitude, locations[i].Location.Longitude);
    for (int j = i + 1; j < locations.Length; j++)
    {
        //Inner Location
        Point corB = new Point(locations[j].Location.Latitude, locations[j].Location.Longitude);
        var s1 = new GeoCoordinate(corA.Latitude, corA.Longitude);
        var s2 = new GeoCoordinate(corB.Latitude, corB.Longitude);
        //Updates the distance and stores if next stores have greater distance
        if (s1.GetDistanceTo(s2) > distance)
        {
            distance = s1.GetDistanceTo(s2);
            one = locations[i];
            two = locations[j];
        }
    }
}
```

A nested for loop was used to keep compute the distances using the API mentioned earlier and the two furthest stores we tracked as the loops iterated.

```
Console.WriteLine("These are the 2 Taco Bells the farthest apart in Alabama.");
Console.WriteLine($"Taco Bell #1: {one.Name}");
Console.WriteLine($"Taco Bell #2: {two.Name}");
distance = Math.Round(distance * 0.000621371, 2); // Converts Distance to miles (round 2 places)
Console.WriteLine($"They are {distance} miles from each other.");
```

The data was then formatted for output.

```
These are the 2 Taco Bells the farthest apart in Alabama.
Taco Bell #1: Taco Bell Blue Ridg...
Taco Bell #2: Taco Bell Fole...
They are 367.82 miles from each other.

C:\Users\jeffr\Desktop\repos\GeoLocationProject\GeoLocatio
0) exited with code 0.
To automatically close the console when debugging stops, e
le when debugging stops.
Press any key to close this window . . .
```

I used xUnit to test and make sure I was getting expected data from each parsed line of data.