Homework Assignment #9

CS5004 – Object-Oriented Design
Northeastern University – Silicon Valley
Summer 2019

Due Sunday 08/04 at 11:00pm PDT

Grading: Each programming problem is graded as follows

- A submission which does not compile gets 0.
- A submission which compiles but does something completely irrelevant gets 0.
- A submission which works (partially) correctly, gets (up to) %80 of the total credit.
- %20 is reserved for the coding style. Follow the coding style described in the book.

Formatting: Each class must reside in its own file. If you need to instantiate objects of class `A` in your class `B`, your `B.java` file must import `A.java` first. The names you choose for your classes should match the ones specified in each problem. Also, for each problem i, you must have a `Problem_i.java` file containing (only) the class which has the `main` method. Finally, you must put all files related to problem i into a folder named `problem_i`. The files *must be placed in the right sub-folder*. Make sure that your code complies. See the note in Problem 1 for an example.
Submission: For each problem i submit one `problem_i.zip` file to Blackboard. The zip file should maintain the structure of the sub-folders. See the note in Problem 1 for an example.

Problem 1 [30pts]. Many Global Positioning Satellite (GPS) units can record waypoints. The waypoint marks the coordinates of a location on a map along with a timestamp. Consider a GPS unit that stores waypoints in terms of an (x, y) coordinate on a map together with a timestamp t that records the number of seconds that have elapsed since the unit was turned on. Write a program that allows the user to enter as many waypoints as desired, storing each waypoint in an `ArrayList`, where each waypoint is represented by a class that you design. Each waypoint represents a successive sample point during a hike along some route. The coordinates should be input as doubles, and the timestamp as an integer. Have your program compute the total distance traveled and the average speed in miles per hour. Use the map scaling factor of $1 = 0.1$ miles. For example, if the only two waypoints are $(x = 1, y = 1, t = 0)$ and $(x = 2, y = 1, t = 3600)$, then the hiker traveled a distance of 0.1 miles in 3,600 seconds, or 0.1 miles per hour.

Problem 2 [60pts]. Implement a priority queue capable of holding objects of an arbitrary type, `T`, by defining a `PriorityQueue` class that implements the queue with an `ArrayList`. A priority queue is a type of list where every item added to the queue also has an associated priority. Define priority in your application so that those items with the largest numerical value have the highest priority. Your class should support the following methods:

- `Add(item, priority)` — Adds a new item to the queue with the associated priority.
- `Remove()` — Returns the item with the highest priority and removes it from the queue. If the user attempts to remove from an empty queue, return `null`.

For example, if q is a priority queue defined to take `Strings`

```
q.add("X", 10);
q.add("Y", 1);
q.add("Z", 3);
System.out.println(q.remove()); // Returns X
System.out.println(q.remove()); // Returns Z
System.out.println(q.remove()); // Returns Y
```

Test your queue on data with priorities in various orders (e.g., ascending, descending, mixed). You can implement the priority queue by performing a linear search through the `ArrayList`. In future courses, you will study a data structure called a heap that is a more efficient way to implement a priority queue.