7 Assignment

Etudes: Stateful Classes, Stateful Methods.

7.1 Etude

Finish all the problems from the lab and include the solutions in your portfolio. ∎

7.2 Etude

Exercises 26.15 through 26.22 in the text (on pages 347 - 351). ∎

Main Assignment: Mapping the World Constructors, Exceptions, Circular Data

We continue designing classes that help us draw the city map and its attractions.

Introduction

We want to build a map program that records the streets in a city (as well as the intersections) and provides directions for a visitor.

We can get the information we need from various online resources. The most accurate representation of the locations that can easily be correlated with geographic maps is the knowledge of the latitude and longitude of the location. To get enough accuracy, we record each of these pieces of information as degrees, minutes, and seconds - with the first two being integers, and the third one represented as a *double*.

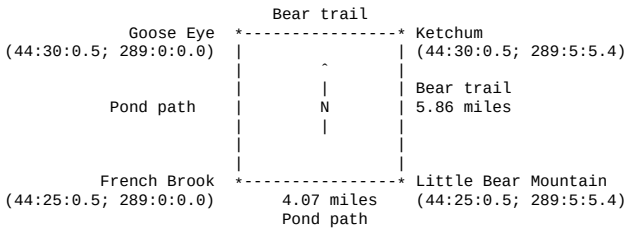
The first problem concerns the representation of a location and producing a travel advice for a travel from one location to another, such as "travel one mile North", or "at XYZ intersection turn left", or "arrive at destination".

The second problem examines one way that can be used to represent a route from the origin to destination.

7.3 Problem: Representing the Map Elements

A location is given by latitude and longitude. Both latitude and longitude are given by degrees, minutes, and seconds. Minutes are whole numbers in the range from 0 to less than 60. Seconds are *double*s in the range from 0 to less than 60. The latitude for our maps is in the range from 24 to less than 50 (the locations in the continental USA.) The longitude for our maps is in the range from 220 to 300 — again the continental USA - measured from the Greenwich meridian going East around the Earth, so that New York is at 286 and Boston is at 289.

Here is some sample data that you should use in your examples (of course, you shall add your own examples as well):



A. Design the classes that represent a location on our map. Make sure that the constructor does not accept invalid data and throws a *RuntimeException* if that happens.

B. Design the method that converts the distance from one latitude to another into a distance in miles.

C. Design the method that converts the distance from one longitude to another into a distance in miles.

D. Design the method that produces the angle that represents the direction of travel from one location to another one measured from North in a clockwise direction.

E. An intersection on a map is given by its name and its location. It also has a list of street segments that start at this intersection. A street segment has a name, the starting intersection, and the ending intersection. Design classes to represent intersections and street segments.

F. Design the method that produces the travel advice for a travel along one street segment.

Note: Design first the method for the advice that specifies the distance, then the advice that gives the direction, then combine the two. For example

"Travel along Bear trail North 5.1 miles to Ketchum"

Note: It is sufficient to give the directions as North, East, South, or West.

G. Design the method that tells the traveler which way to turn when moving from one street segment to another, e.g.:

"At French Brook turn left"

∎

7.4 Problem: Representing the Routing

It is clear that a list of street segments can be used to represent a route from the origin to the destination.

A. Design the classes to represent a route. Make sure that it is constructed in such was that the ending intersection of a street in the route is the starting intersection of the following street.

Hint: Design a constructor that only allows you to add a street that ends where this route starts.

B. Design the method that computes the total distance of the route.

C. Design the method that produces a *String* that represents the travel advice along this route. Use "\n" to separate the lines.

For example one route Ketchum to Goose Eye to Little Bear Mountain would result in:

Total distance from Ketchum to Little Bear Mountain is 14.0 miles.

From Ketchum travel West on Bear Trail 4.07 miles to Goose Eye.
At Goose Eye turn left.
From Goose Eye travel South on Pond Path 5.86 miles to French Brook.
At French Brook turn left.
From French Brook travel East on Pond Path 4.07 miles to Little Bear Mountain.
Arrive at destination at Little Bear Mountain.

∎