

5 Equality, Constructors, Extending Classes

Portfolio Problems

Work out as complete programs the following exercises from the textbook. Make sure you read the textbook to get a guidance how to proceed.

Problems:

1. Problem 19.5 on page 271
2. Problems 19.6 - 19.11 on page 276-279

You may complete this part with animation using the *draw* library.

Pair Programming Assignment

5.1 Problem

Complete the work on Problem 5.2 from Lab 5 and hand in your solution.

5.2 Problem

Complete the work on Problem 5.3 from Lab 5 and hand in your solution.

5.3 Problem

Design the class `CartPt` that extends the `geometry` library class `Posn`. We will not add any new fields, but by extending the library class we can add new methods to it.

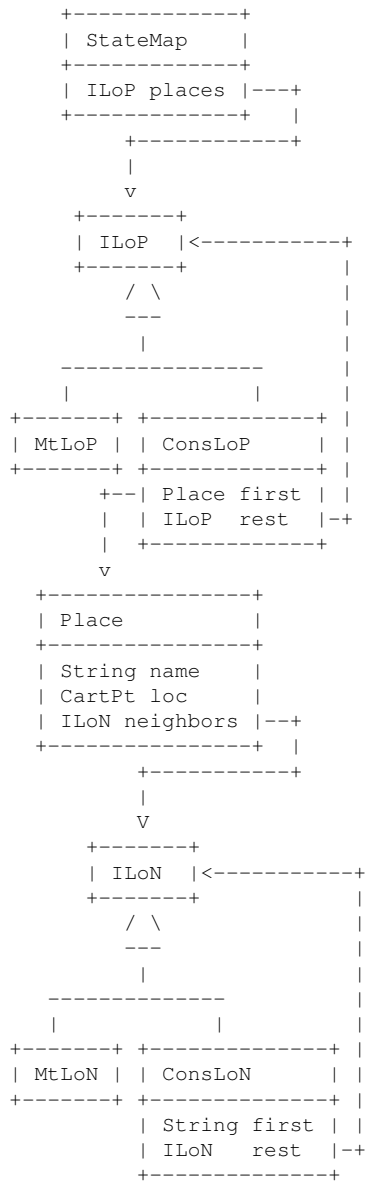
Design the following methods for the class `CartPt`:

- Method `distTo` that computes the distance from this point to the given one.
- Method `fromTo` that produces a `String` that tells us in which direction do we have to travel from this point to the given one. Make sure you cover the following possibilities:
going North, East, West, South, Northwest, Northeast, Southwest, and Southeast.

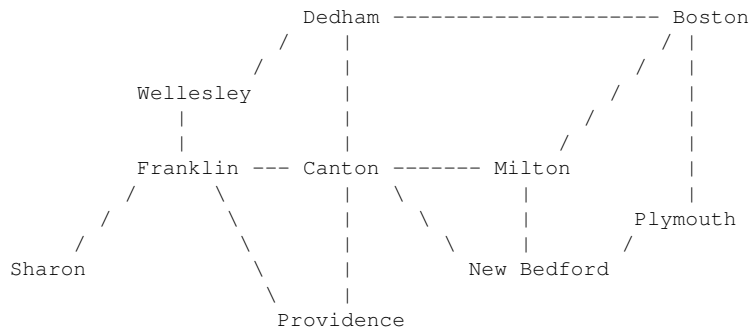
5.4 Problem

The following class diagram describes data definitions that represent a state map:

Each place has a unique name, a location, and a list of neighbors (only their names). The state map is a list of places.



- Make a simple map with at least ten places for your example and show how it looks. It can be *ASCII art* included in your code, or a *Canvas* drawing, or a sketch turned into a *.png* file. The following is a sample map:



Design the following methods for the classes that represent the state map:

- Method `totalDistance` that computes the total distance for the route that is given as a list of names. Throw `IllegalArgumentException` if the name is not a name of a place in the state maps list of places.
- Method `hasLoop` that determines whether the given route goes through one of the cities more than once.
- Method `directions` that produces a list of `Strings` that tell us how to get from one place to another one, following the chosen route.

For example, the directions could be:

```

"From Harvard Square to Boston Common"
"Go Southeast 1.2 miles to Central Square"
"Go Southeast 0.5 miles to Mass Ave and Main Street"
"Go East 1.3 miles to Longfellow Bridge"
"Go East 0.5 miles to MGH"
"Go South 0.5 miles to Boston Common"

```

- Display the route on the *Canvas*.
- Now animate the route using a slow tick rate, so the user can see where to go - possibly displaying the routing `String` at the bottom of the *Canvas*. Alternately, control the display by pressing the space bar (`String " "`).