Armand Sandiu
Eric Navarro

Report Lab 1 POO

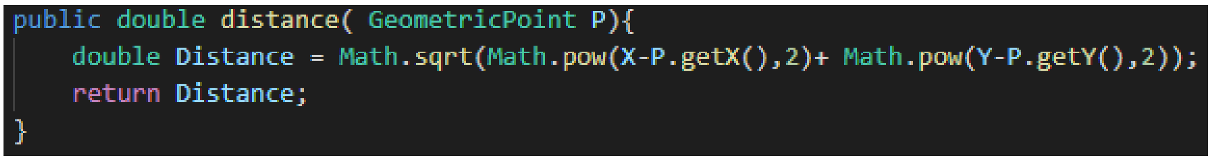
The aim of this lab is the creation of two classes. The first one is geometric point which creates points that will we used on the second class Distance matrix which will do a matrix of distances between the points.

2) Geometric Point:

For this question we were asked to create a class called Geometric Point which creates a Point that has coordinates and a name. This are the attributes that we declared and then we did some methods that are:

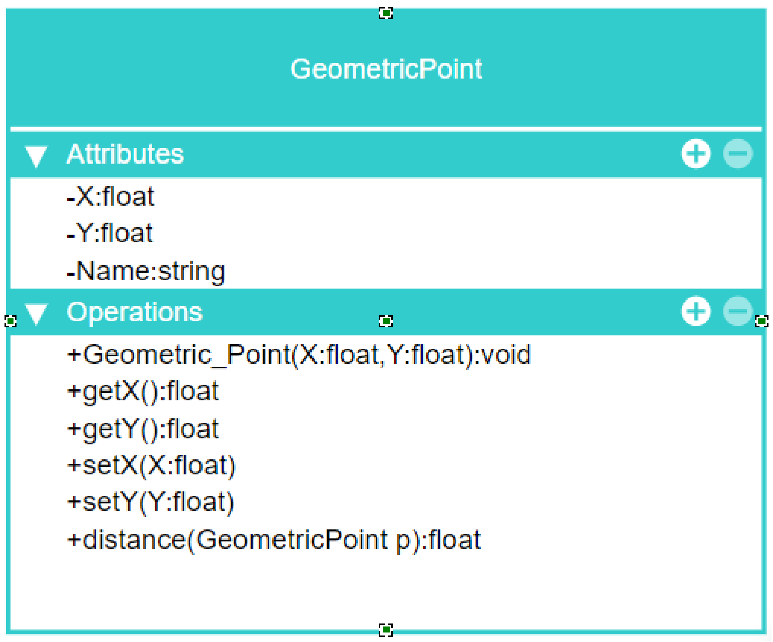
- The constructor:
  That initializes the attributes.
- Getters
- Setters
- Method that calculate the distance between two Geometric Points

For the last method, we had some issues in terms of how to do a square and a root. We realized that, with the import of *java.lang.Math* we could take functions like pow and sqrt that allowed us to keep on. Furthermore, we learned how to use getters for obtaining the coordinates of our points. The method has been done like this:



```
public double distance( GeometricPoint P){
    double Distance = Math.sqrt(Math.pow(X-P.getX(),2)+ Math.pow(Y-P.getY(),2));
    return Distance;
}
```

it follows the formula: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$ Which calculate the distance between two points.

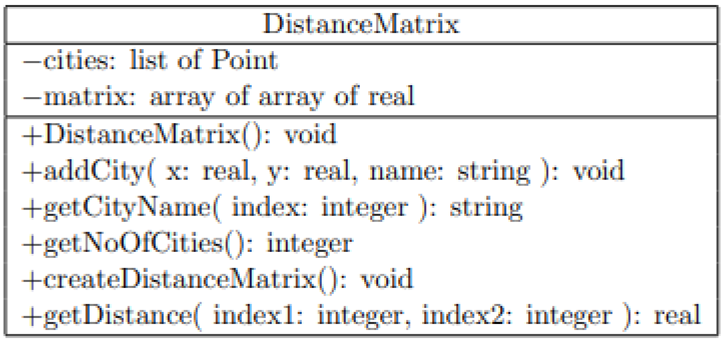


(This capture must include the getter and setter of the variable name)

Armand Sandiu
Eric Navarro

3 Distance Matrix:

To implement the class that represented distance matrices we were suggested to use the existing Java class LinkedList. To help with the implementation of the distance matrix class we were provided a sample design of the DistanceMatrix class:



```
DistanceMatrix
−cities: list of Point
−matrix: array of array of real
+DistanceMatrix(): void
+addCity( x: real, y: real, name: string ): void
+getCityName( index: integer ): string
+getNoOfCities(): integer
+createDistanceMatrix(): void
+getDistance( index1: integer, index2: integer ): real
```

It must be noted we were also allowed to implement our own design. However, the class DisplayMatrix would only work if the methods of DistanceMatrix were exactly those listed in the class Distance Matrix. In the end, we opted for choosing the class implementation that we were given since we believed it would cause less problems with the code provided (especially since we were warned the methods could cause unexpected problems if we implemented them ourselves.

For the implementation of the code, we started declaring the attributes as always. First the Linked List of cities, where cities represent geometric points with city names and then the double linked list that represents the matrix of distances between points. As you can see in our code, we finally declare a doubly array but it will be explained later the aim of this change. After the attributes, we start coding the methods one by one in the same order as the design. The explanation of each method:

- **distance matrix():** This is the constructor of our class, we decided to create a default constructor because both attributes are initialized in other functions.

- **addcity(double InitX, double InitY, String):** The use of this function is to put our geometric points on our linked list cities. To do this, we declared a Geometric point that has as parameters the coordinates and the name that we introduce through the test and then using the function add imported with “java.util.LinkedList” we introduce the point on the linked list.

- **getCityName(int index):** The aim of this method is to obtain the name of any city that we indicate with the index. Then it is as simple as using a get to indicate what city we are referring and then; call the getname function.

- **getNoOfCities():** This function must return the number of geometric points we have on our linked list of cities. We use the funcion .size which returns the integer we need.

- **CreateDistanceMatrix():** This is the method that more problems carried to us when we coded it. It was at this moment that we decided to change the type of the variable matrix to a double array. We didn't find the way to initialize the matrix on the size we needed since doubled Linked List can’t be indexed. We found it much easier to implement as an array because we initialized it with one code line for the matrix and then we did a loop for introducing the distances.

- **GetDistance(int index1, int index2):** For this method, we indexed on the position that we wanted of the matrix and there we calculated the distance between the points as well as selected for the indexes.