Seminar 1: Designing a class
24292-Object Oriented Programming

1 Introduction

The objective of this seminar is to learn how to design a class, the fundamental unit in object-oriented programming. Each class has a number of attributes and methods that are to be set to complete the design. The seminar will consist of two design exercises each requires the definition of a new class. We will first design the class that represents a geometric point in space. Then we will design a distance matrix class. For each attribute that you define you must specify:

- its visibility (public or private)
- its type (integer, real, char, string, ...)
- its name

For each method that you define you must specify:

- its visibility (public or private)
- its name
- its parameters including name and type
- its return type (the method may not return anything)

A constructor method is called each time an instance of that class is created. Each instance can have different attribute values and the constructor method must initialize / assign them. A "getter" method returns the value of an attribute and a setter changes it.

The solution to these exercises will be implemented in Java in the next lab session.

2 Geometric Point class

A geometric point is defined by the coordinates x and y which we will consider to be real numbers. Some examples of points: (0,14; 1,25), (−1,3; 0,44). The distance between two points can be computed by the Pythagorean theorem, given two points $(x_1; y_1)$ and $(x_2; y_2)$ its distance is equal to: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

We are going to design a class that represents geometric points. That is, its instances (or exemplars) will be all the geometric points: each particular instance will assign its own values to the attributes of the class. Define the attributes of the class and its methods (constructor, getters and setters) including an additional function to compute the distance to another point and a method to print in a string format the point itself: for example "(0,14; 1,25)".

3 Matrix Distance class

The second exercise consists in designing a class which represents the matrices of distances between points. It is common to include distances between important cities in maps in a matrix form like this:

	Madrid	Barcelona	Valencia
Madrid	0	750	600
Barcelona	750	0	300
Valencia	600	300	0

We will design a class for dealing with this kind of matrices. For the class you will need to specify its attributes and its methods.

The matrix is initialized empty and points can be added one by one. It can be useful that the distances between points are pre-computed: for this purpose we will need a method to construct the matrix. This method can be called each time a point is added to update the matrix. We will also need a method to retrieve the distance between two particular points. To store the points you can use the list data structure. To store the pre-computed distance values you can use a matrix (a two dimensional array).