Imperial College
London

Software Engineering 2: Object Oriented Software Engineering

Lab 1 – Introduction to objects

Max Cattafi (`m.cattafi@imperial.ac.uk`)

class point

Write a class `point` featuring:

- Attributes for x and y coordinates and the distance from the origin.
- A default constructor with no parameters initializing the coordinates to 0 and an overloaded constructor which takes two parameters for the initial value of the coordinates.
- Setter methods for the coordinates (keeping the state consistent with respect to the distance from the origin).
- A method returning a `string` with some kind of representation of the point (e.g. `(3.3, 4.2)`).
- A method returning the distance of the point from the origin.
- A method which, given as argument another point, returns the distance between the two points.
- A method changing the state of the object to its symmetric with respect to the origin.
- A method which, given as argument another point, translates the first point accordingly. E.g. if point `p1` has state `(1, 2)` and point `p2` `(3, 4)`, after `p1.translate(p2)` the state of `p1` should be `(4, 6)`.

You can add other global and member functions as you find suitable, however keep in mind principles of abstraction and encapsulation.

Symmetry, translation, distance

Write a `main` to test the class. For instance check that the distance from the origin is not affected by symmetry transformations with respect to the origin, check how the distance of a point from the origin changes after a translation.

Points and lines

Write a program which:

- Reads from the user a vector of numbers, each representing the parameter b in the line equation $y = b$.
- Reads from the user (the coordinates of) a point P_0.
- For each number b, computes the distance between the line $y = b$ and P_0 using the member function `distance` described above (hint: as for the declaration of usual variables, you can construct an object locally to the scope e.g. of a loop), and prints it on the screen.

Farthest point

Write a (global) function which takes as argument a vector of points and returns the index of the one which is farthest from the origin. Write a main to test the function.

Triangles

Define a class `Triangle`, whose attributes are the three points delimiting it.

Define a constructor which takes three points as arguments. Define a member function `perimeter` which returns the perimeter of the triangle object on which it is called. Define a member function `translate` which takes as argument a point (representing a vector) and changes the state of the triangle translating it by the vector.

Write a main to test the class, check for instance that the perimeter of a triangle is the same before and after a translation.