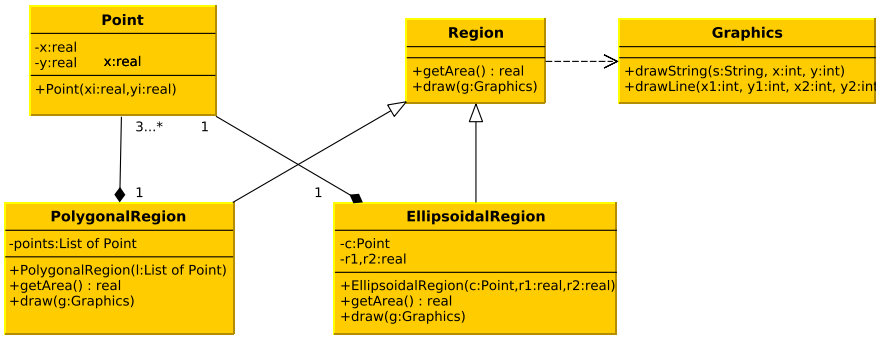
Seminar 4: Designing an application using the concept of inheritance
24292-Object Oriented Programming

1 Introduction

The objective of this seminar is to design an application making special attention to the use of the concept of inheritance. We are going to depart from the region hierarchy of previous seminar to design a drawing application. The solution to these exercises will be implemented in Java during the laboratory session 4.

2 Region Hierarchy

In Seminar 2 we defined the region hierarchy:



We will now extend this hierarchy with other region types and operations. Remember that there existed two types of regions Polygonal and Ellipsoid. A polygonal region is a polygonal shape made up of points. An an ellipsoidal region must specify its center and its two diameters.

We will now add other particular types of regions: Triangular regions, Rectangular and Circular regions.

2.1 Functionality

We are going to add operations to manipulate regions. We want now to be able to specify that a region is rotated a certain number of degrees without modifying too much our previous classes.

We will then add rotate and move/translate operations to manipulate regions. The rotation operation will be implemented in a simple way by storing the rotation to perform without modifying the region. The translate operation, on the other hand, will modify the position of the regions accordingly.

It will be useful for being able to select regions to have a function that answers if a certain point is inside the region or not.

Specify the attributes, constructors and methods of all classes.

3 The entity hierarchy

For our drawing application we are going to consider that anything that can be drawn in the screen is an entity. In addition to regions also lines and text labels can be drawn in the screen. All share the color with which they would be drawn in the screen. An entity has a color and a region has in addition a fill color. An entity can be rotated and translated in the same way as we described in previous section. Incorporate those in the design.

We are going to define what is a color which is specified by three components that can be either its Red, Green, Blue or Hue, Saturation, Value characteristics. There exist transformation operations between these characteristics just add methods to perform them.

4 Drawing application

The drawing application stores all the entities to be drawn in the screen and also adds some global operations to them. A move/translate operation exists that can be applied to all entities at once: sketch an implementation making use of polymorphism.

Entities can be selected, a function exists for this purpose: given the coordinates it returns the entities that where inside or close enough: this function can reuse previously specified functions; you can consider that a region is selected when clicked inside and a line and a text label when close enough to one of the points. The app stores at every moment all the selected entities. Think of what modifications you can do to implement this feature. Sketch how you would implement it. You can use the `instanceof` command seen in class and here follows an example of usage:

```
if ( person instanceof Student ) {
        Student student = (Student)person;  // Downcast example
        student.getNumberOfCourses();
}
```