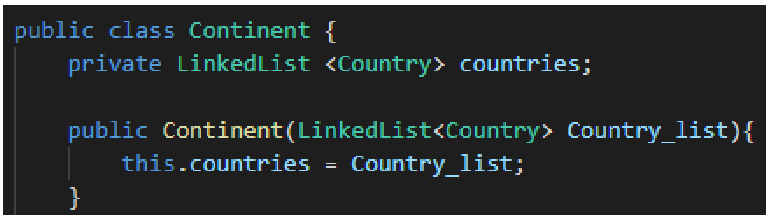
Armand Sandiu
Eric Navarro

Report Lab 3 POO

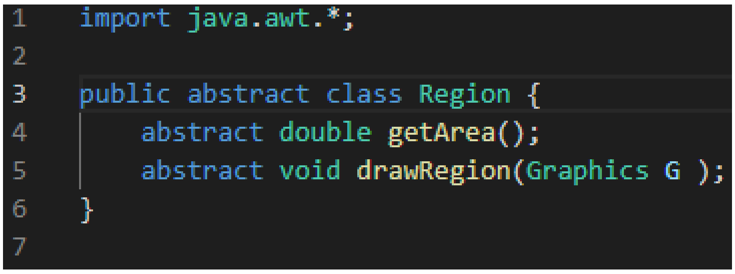
Introduction and Countries:

The aim of this lab is the reproduction of lab 2 but with some changes.The first is to change the class continent. Now this class has to refer to the countries which at the same time is a new class differing of the past lab.



```java
public class Continent {
    private LinkedList <Country> countries;

    public Continent(LinkedList<Country> Country_list){
        this.countries = Country_list;
    }
```

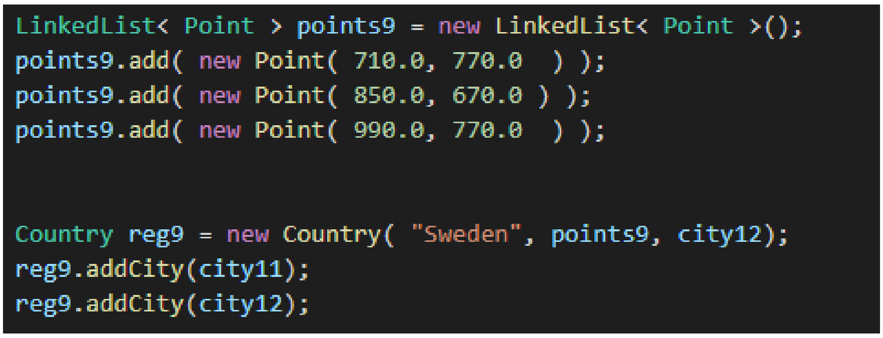
We are also asked to rewrite the code to reflect that countries inherit from polygonal regions, which in turn inherit from regions. For this, we create a new class called Region which is abstract, then we recycle Polygonal Region and we inherit the class countries with the option extend of Java.



```java
import java.awt.*;

public abstract class Region {
    abstract double getArea();
    abstract void drawRegion(Graphics G );
}

```

Here is our abstract class Region.

We also used the classes my window and myMap, but on this occasion, we had to rewrite the code of mymap in order to add more points which are directly passed to the country class in the form of a linked list. Here an example of a set of points implemented on one of the countries:



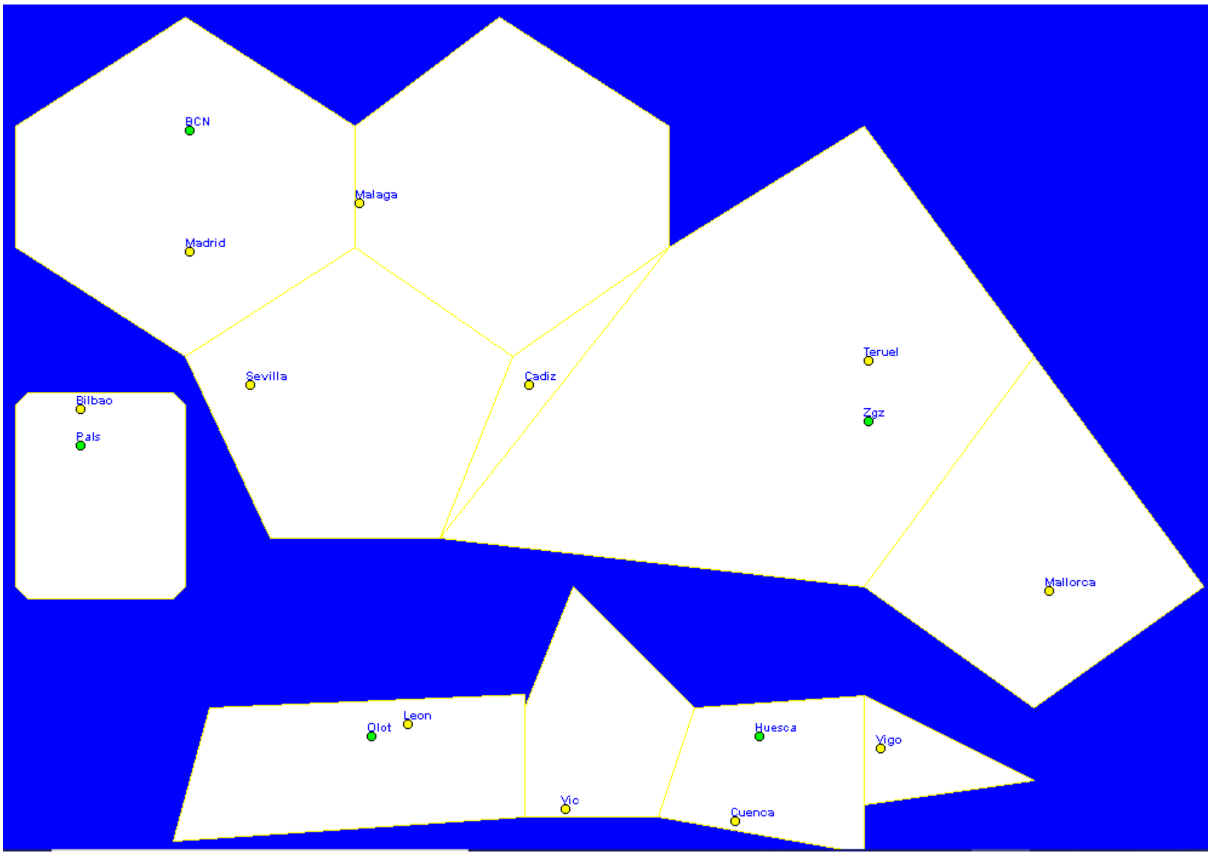
```java
LinkedList< Point > points9 = new LinkedList< Point >();
points9.add( new Point( 710.0, 770.0  ) );
points9.add( new Point( 850.0, 670.0 ) );
points9.add( new Point( 990.0, 770.0  ) );


Country reg9 = new Country( "Sweden", points9, city12);
reg9.addCity(city11);
reg9.addCity(city12);
```

Cities:

Armand Sandiu
Eric Navarro

At this point, we implemented 3 new classes, which are: Point class, GeoPoint and City class. For the first 2 we reuse the same classes done for the previous labs. For City class this inherits directly from Geopoint so with and extend and the function super we recover the same arguments of the inherited class. Also, for the draw function, we implemented a code which will print a city with one color, depending if it is capital or not. To do it, we used the two methods that we are encouraged to use. These are fillOval and drawString, the last one permitted us to draw the name of any city.

Finally, by adding the corresponding cities in our previous on the MyMap class, we reach the objective of creating a world with different countries which the respective cities where just one is the capital:



Color red refers to the capital of each country