PROJECT - CS361

NB: – Use the same name for functions and constructors as given in the text.
– All function must be in their curried form.

1. We provide the datatype *POINT* to define the coordinates of a point in 2D.

   ```
   datatype POINT = Coord of real * real;
   ```

   (a) Write a function *xCoordinate* that returns the x coordinate of a point.
   (b) Write a function *yCoordinate* that returns the y coordinate of a point.
   (c) Write a function *distance* that computes the distance between 2 points.
   (d) Write a function *equals* that computes test if 2 points are equal.
   (e) Write a function *calculateAngle* that computes the angle that a line (OA) (where A is a point) does with the x line (in degrees).

2. Write a JAVA class *Point* that contains 2 constructors:

   ```
   public Point() and public Point(Point aPoint)
   ```

   and the methods *xCoordinate*, *yCoordinate*, *distance*, *equals* and *calculateAngle*.

3. We provide the datatype *RECTANGLE* to define a rectangle.

   ```
   datatype RECTANGLE = Rectangle of POINT * POINT;
   ```

   A rectangle is formed by two diagonally opposite points. We assume that the rectangle has sides parallel to x and y-axis. The first point is the point of the lower left corned of the rectangle and the other point is at the upper right corner of the rectangle.

   (a) Write a function *rearrange* which rearranges two points of a rectangle so that one point is at the lower level and the other point is at the upper right corner (to form a rectangle).
   Example: If we rearrange the points $A(10, 6)$ and $B(5, 15)$ of the rectangle R, we return the points $C(5, 6)$ and $B(10, 15)$ to form the new rectangle.
   The type of *rearrange* is:

   ```
   val rearrange = fn : RECTANGLE -> RECTANGLE
   ```

   (b) Write a function *isValid* that checks if a given rectangle is really a rectangle (those two points form a rectangle).

(c) Write a function *containsPoint* which checks if a point lies within a rectangle.

(d) Write a function *containsRectangle* which checks if a rectangle is within a rectangle.

(e) Write a function *getDiagonal* that computer the length of the diagonals of the rectangle.

(f) Write a function *intersects* which checks if one rectangle intersects the other. You have to check if one or more corner points of one rectangle is contained in the other rectangle.

4. Write a JAVA class *Rectangle* that contains a constructor:

```
public Rectangle(Point p1, Point p2)
```

and the methods *rearrange*, *isValid*, *containsPoint*, *containsRectangle*, *getDiagonal* and *intersects*.

5. To do:

(a) Form a group of 3 students (more than 3 students is not allowed).

(b) Write the SML functions as asked.

(c) Write the 2 JAVA classes as asked.

(d) Provide a report that will contain the different parts described below. Your report must be well-organized and well-presented.

(e) Provide your different listings.

(f) Provide a demonstration of all the SML functions you defined.

(g) Provide a demonstration of all the JAVA functions you defined.

(h) Provide a list of the meeting you had to do this project.

(i) Provide a list of "what did what?"

(j) Provide a comparison of SML and JAVA with respect to this project and also more generally.

(k) Provide a description of what you liked and what you disliked...hated... in SML.

(l) Prepare a talk of 5 slides (+1 slide of presentation of the project and of your group). One slide will deal with the comparison of JAVA and SML and one slide will deal with your opinion about SML. These slides must be in the report. If we have time you will present your work. Hopefully...

(m) ...

Good work!