# Đa hình 2

Mô tả Bài nộp Thảo luân Đề bài: Mức đô: Khó 7 Tổng số bài làm đúng: <<abstract>>Shape # color : String # filled : boolean Tổng lượt nộp bài: 80 + Shape() + Shape(color : String, filled : boolean) + getColor(): String + setColor(color: String): void + isFilled(): boolean + setFilled(filled: boolean): void + getArea() : double + getPerimeter : double + toString(): String Extends Extends Circle Rectangle # center : Point # topLeft : Point # radius : double # width : double # length : double + Circle() + Circle(radius : double) + Rectangle() + Rectangle(width : double, length : double) + Circle(radius : double, color : String, filled : boolean) + Rectangle(width : double, length:double, color:String,filled:boolean) + Circle(center : Point, radius : double, color : String, filled : boolean) + Rectangle(topLeft:Point, width:double, length: double,color:String,filled:boolean) + getCenter() : Point + getTopLeft() : Point + setCenter(center : Point) : void + setTopLeft(topLeft:Point) : void + getRadius() : double + setRadius(radius : double): void + getArea() : double + getWidth(): double + setWidth(width:double) : void + getLength() : double + setLength(length:double) : void + getPerimeter() : double + toString(): String + equals(o: Object): boolean + hashCode(): int + getArea() : double + getPerimeter() : double + equals(o : Object) : boolean + hashCode(): int + toString() : String Extends Square + Square() + Square(side : double) + Square(side : double, color:String,filled:boolean) + Square(topLeft:Point, side:double,color:String,filled:boolean) + getSide() : double + setSide(side:double) : void + setWidth(width:double) : void + setLength(length:double): void + equals(o : Object) : boolean + hashCode(): int + toString() : String

#### **Point**

pointX : doublepointY : double

+ Point(pointX:double, pointY:double)

+ getPointX(): double

+ setPointX(pointX:double): void

+ getPointY() : double

+ setPointY(pointY:double) : void + distance(newPoint : Point) : double

+ equals(o : Object) : boolean

+ hashCode() : int + toString() : String

#### Layer

- shapes : ArrayList<Shape>

+ add>nape(snape : Snape) : void

+ removeCircles() : void + getInfo() : String

+ removeDuplicates(): void

Tiếp tục từ bài buổi trước, quan sát biểu đồ lớp trên đây, hãy:

• Định nghĩa thêm lớp **Point** như biểu đồ lớp. Phương thức *distance()* dùng để tính khoảng cách giữa 2 đối tượng **Point**. Phương thức *toString()* trả về thông tin có định dạng:

## (1,2)

- Thêm các thuộc tính center cho lớp Circle, topLeft cho lớp Rectangle
- Thêm các constructor có bổ sung tham số mới cho các lớp Circle, Rectangle, Square
- Cập nhật giá trị trả về của phương thức to String() cho các lớp Circle, Rectangle, Square như sau:

```
// Giá tri double in ra làm tròn đến chữ số thập phân thứ nhất

// Circle
Circle[center=(10.0,5.0),radius=8.0,color=RED,filled=true]

// Rectangle
Rectangle[topLeft=(10.0,5.0),width=8.0,length=6.0,color=RED,filled=true]

// Square
Square[topLeft=(10.0,5.0),side=8.0,color=RED,filled=true]
```

- Cài đặt các phương thức equals() cho các lớp Circle, Rectangle, Square. Sai số cho phép khi so sánh các giá trị double là 0.001
- Định nghĩa lớp Layer chứa danh sách các đối tượng Shape
  - o Cài đặt phương thức addShape() để thêm 1 hình vào layer
  - Cài đặt phương thức *getInfo()* trả về thông tin các hình trong 1 layer theo định dạng (hình nào thêm trước thì hiện lên trước):

```
Layer of crazy shapes:

Circle[center=(10.0,5.0),radius=8.0,color=RED,filled=true]

Rectangle[topLeft=(10.0,5.0),width=8.0,length=6.0,color=RED,filled=true]

Circle[center=(10.0,5.0),radius=8.0,color=RED,filled=true]

Square[topLeft=(10.0,5.0),side=8.0,color=RED,filled=true]
```

- o Cài đặt phương thước removeCircles() để xóa tất cả hình tròn trong layer
- Cài phương thức removeDuplicates() để xóa các hình trùng nhau trong một Layer (ví dụ: với Circle, 2 hình trùng nhau sẽ tọa độ tâm và độ lớn bán kính như nhau). Chú ý không so sánh dựa trên thuộc tính color và filled, lúc xóa thì xóa những hình được thêm vào sau. Gợi ý: Bạn có thể tự cài đặt thuật toán hoặc dùng Lambda Expression hoặc Set. Đối với cách thứ 2 bạn cần cài đặt phương thức hashCode() cho các đối tượng cần so sánh.

### Định dạng đầu vào:

- 1. Các file .java nộp lên **không định danh package** trong đó (bỏ tất cả dòng pakage)
- 2. Tất cả file .java đặt cùng trong một folder và được nén lại dưới đuôi .zip
- 3. **Tên folder** chứa các **file .java** không được chứa ký tự đặc biệt hoặc ký tự khoảng trắng.

Bạn còn 0 lần nộp bài