Examen de Prácticas - 5 de febrero de 2021
LTP - 1º Parcial (Java) - Tipo B

ALUMNO: _____ GRUPO: _____

Instrucciones

- El alumno dispone de 45 minutos para resolver el examen.
- El examen consta de 3 preguntas que deberán responderse en el mismo enunciado, en los recuadros incluidos en cada pregunta.
- IMPORTANTE. Se valorará, cuando sean aplicables, el uso adecuado de los mecanismos de herencia, sobrecarga, polimorfismo o genericidad.

Para todas las preguntas, considera un proyecto BlueJ con la siguiente jerarquía de 4 clases Java:

```
public abstract class Figure {
   private double x, y;
   public Figure(double x, double y) { this.x = x; this.y = y; }
   public boolean equals(Object o) {
      if (!(o instanceof Figure)) { return false; }
      return x == ((Figure) o).x && y == ((Figure) o).y;
   }
   public abstract double perimeter();
}

public class Circle extends Figure {
   private double radius;
   public Circle(double x, double y, double r) {
      super(x, y); radius = r;
   }
   public double perimeter() { return 2 * Math.PI * radius; }
}

public class Triangle extends Figure {
   private double s1, s2, s3; // longitudes de los 3 lados del triángulo
   public Triangle(double x, double y, double s1, double s2, double s3) {
      super(x, y); this.s1 = s1; this.s2 = s2; this.s3 = s3;
   }
   public double perimeter() { return s1 + s2 + s3; }
}

public class Equilateral extends Triangle { // triángulos equiláteros
   public Equilateral(double x, double y, double s) { super(x, y, s, s, s); }
}
```

Pregunta 1 (3.00 puntos)

Se quiere añadir al proyecto BlueJ una nueva clase, llamada `Polygon`, teniendo en cuenta que:

- La clase `Polygon` se usará para modelizar figuras que sean polígonos.
- La clase tendrá un atributo correspondiente al número de lados del polígono, además de los atributos `x` e `y` que indican su posición.
- La clase tendrá un constructor que permitirá dar valores a los atributos indicados.
- La clase no será instanciable. No se podrán crear objetos `Polygon`.
- La clase tendrá un método abstracto, llamado `isRegular`, que devolverá un valor lógico indicando si el polígono es regular o no.

Escribe el código de la clase `Polygon` en el siguiente recuadro:

```
public abstract class Polygon extends Figure {
   private int sides;
   public Polygon(double x, double y, int s) {
      super(x,y); sides = s;
   }
   public abstract boolean isRegular();
}
```

Pregunta 2 (3.00 puntos)

Una vez añadida al proyecto BlueJ la clase `Polygon`, se deben modificar otras clases del proyecto teniendo en cuenta que:

- Los círculos no son polígonos.
- Los triángulos son polígonos de 3 lados, pero no son regulares.
- Los triángulos equiláteros son polígonos regulares de 3 lados.

Escribe el código de la(s) clase(s) a modificar en el siguiente recuadro:

```
public class Triangle extends Polygon {
   private double s1, s2, s3;
   public Triangle(double x, double y, double s1, double s2, double s3) {
      super(x, y, 3); this.s1 = s1; this.s2 = s2; this.s3 = s3;
   }
   public double perimeter() { return s1 + s2 + s3; }
   public boolean isRegular() { return false; }
}

public class Equilateral extends Triangle {
   public Equilateral(double x, double y, double s) { super(x, y, s, s, s); }
   public boolean isRegular() { return true; }
}
```

Pregunta 3 (4.00 puntos)

Se quiere añadir al proyecto BlueJ una nueva clase, llamada `PolygonList`, teniendo en cuenta que:

- La clase `PolygonList` heredará de la clase `ArrayList<T>`, pero la genericidad deberá restringirse a la clase `Polygon`.
- La clase `PolygonList` dispondrá de los atributos y métodos que, por herencia de `ArrayList<T>`, le corresponda.
- La clase tendrá un método, llamado `greatest`, que devolverá el objeto `Polygon`, almacenado en `this`, de perímetro mayor.

Escribe el código de la clase `PolygonList` en el siguiente recuadro:

```
public class PolygonList <T extends Polygon> extends ArrayList<T> {
   public Polygon greatest() {
      double max = 0;
      Polygon res = null;
      for (int i = 0; i < size(); i++) {
         Polygon p = get(i);
         double a = p.perimeter();
         if (a > max) { max = a; res = p; }
      }
      return res;
   }
}
```