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

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 area();
}

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

public class Rectangle extends Figure {
    private double base, height;
    public Rectangle(double x, double y, double b, double h) {
        super(x, y); base = b; height = h;
    }
    public double area() { return base * height; }
}

public class Square extends Rectangle {
    public Square(double x, double y, double l) { super(x, y, l, l); }
}
```

Pregunta 1 (3.00 puntos)

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

- La clase **Parallelogram** se usará para modelizar figuras que sean paralelogramos.
- La clase tendrá un atributo correspondiente al valor de un ángulo interior (medido en grados sexagesimales), 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 **Parallelogram**.
- La clase tendrá un método abstracto, llamado **isRotatable**, que devolverá un valor lógico indicando si, cuando se gira 90 grados, el paralelogramo cambia (visualmente) o no.

Escribe el código de la clase **Parallelogram** en el siguiente recuadro:

```
public abstract class Parallelogram extends Figure {  
    private double angle;  
    public Parallelogram(double x, double y, double a) {  
        super(x,y); angle = a;  
    }  
    public abstract boolean isRotatable();  
}
```

Pregunta 2 (3.00 puntos)

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

- Los círculos no son paralelogramos.
- Los rectángulos y los cuadrados son paralelogramos con ángulos interiores de 90°.
- Si se giran 90°, los rectángulos cambian, pero los cuadrados no cambian.

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

```
public class Rectangle extends Parallelogram {
    private double base, height;
    public Rectangle(double x, double y, double b, double h) {
        super(x, y, 90); base = b; height = h;
    }
    public double area() { return base * height; }
    public boolean isRotatable() { return true; }
}

public class Square extends Rectangle {
    public Square(double x, double y, double l) { super(x, y, l, l); }
    public boolean isRotatable() { return false; }
}
```

Pregunta 3 (4.00 puntos)

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

- La clase `FigureList` heredar  de la clase `ArrayList<T>`, pero la genericidad deber  restringirse a la clase `Figure`.
- La clase `FigureList` dispondr  de los atributos y m todos que, por herencia de `ArrayList<T>`, le corresponda.
- La clase tendr  un m todo, llamado `howMany`, que devolver  el n mero de objetos `Parallelogram`, almacenados en `this`, de  rea mayor a un valor recibido como argumento.

Escribe el c digo de la clase `FigureList` en el siguiente recuadro:

```
public class FigureList <T extends Figure> extends ArrayList<T> {  
    public int howMany(double x) {  
        int res = 0;  
        for (int i = 0; i < size(); i++) {  
            Figure f = get(i);  
            if (f instanceof Parallelogram && f.area() > x) res++;  
        }  
        return res;  
    }  
}
```