



ALUMNO:

EPS

Asignatura: IN0P14 Técnicas de Programación Avanzada

Curso: 2020/2021

Examen: Final

Fecha: 21-01-21

Semestre: 1

Convocatoria: Ordinaria

1. Dada la librería Polinomio.

```
public class Polinomio {
    int grado;
    double [] coeficientes;

    public Polinomio(double [] coeficientes) {
        this.grado = coeficientes.length - 1;
        this.coeficientes = new double[coeficientes.length];

        for (int i=0; i<coeficientes.length; i++)
            this.coeficientes[i] = coeficientes[i];
    }

    public Polinomio cuadrado(Polinomio p) {
        // eleva al cuadrado un polinomio
    }
}
```

[2,0 puntos] Desarrolle el método cuadrado(Polinomio p).

Por ejemplo, el cuadrado del polinomio $-5x^2 + 3x + 2$ es $25x^4 - 30x^3 - 11x^2 + 12x + 4$.

2. [3,0 puntos] Desarrolle la clase Cola con tipos genéricos basada en un vector. Defina los siguientes métodos: inserta(T e), elimina(), vacia(), primero(), ultimo(), toString().

```
import java.util.Vector;

public class Cola<T> {

    public Cola() {
        // método constructor
    }

    public void inserta(T e) {
        // inserta el elemento e en la cola
    }

    public T elimina() {
        // devuelve el primer elemento de la cola y lo elimina
    }

    public boolean vacia() {
        // devuelve true si la cola está vacía y false en caso contrario
    }

    public T primero() {
        // devuelve el primer elemento de la cola
    }

    public T ultimo() {
        // devuelve el último elemento de la cola
    }

    public String toString() {
        // devuelve un String con el contenido de la cola
    }
}
```

Por ejemplo, el siguiente programa muestra una cola con los valores {2, 4, 8, 16}.

```
public class Programa {
    public static void main(String[] args) {
        Cola<Integer> cola = new Cola<Integer>();

        cola.inserta(1);
        cola.inserta(2);
        cola.inserta(4);
        cola.inserta(8);
        cola.inserta(16);

        System.out.println("Cola    " + cola.toString());
        System.out.println("Primero " + cola.primer());
        System.out.println("Ultimo  " + cola.ultimo());

        cola.elimina();

        System.out.println("Cola    " + cola.toString());
        System.out.println("Primero " + cola.primer());
        System.out.println("Ultimo  " + cola.ultimo());
    }
}
```

3. [3,0 puntos] Desarrolle la clase AdaptadorPolarRectangular que implementa la interfaz IPuntoRectangular para adaptar objetos de la clase PuntoPolar a objetos de la clase PuntoRectangular.

```
public interface IPuntoPolar {
    public double getRadio();
    public double getAngulo();
}

public interface IPuntoRectangular {
    public double getX();
    public double getY();
}

public class PuntoPolar implements IPuntoPolar {
    // la clase PuntoPolar implementa la interfaz IPuntoPolar
}

public class PuntoRectangular implements IPuntoRectangular {
    // la clase PuntoRectangular implementa la interfaz IPuntoRectangular
}

public class AdaptadorPolarRectangular implements IPuntoRectangular {
    // adaptador de un objeto de la clase PuntoPolar a un objeto PuntoRectangular
}
```

Un objeto de la clase PuntoPolar se define por el radio y un ángulo en grados. Para convertir un punto en coordenadas polares a un punto en coordenadas rectangulares, las coordenadas x e y se calculan:

$$x = \text{radio} \cos(\text{angulo})$$
$$y = \text{radio} \sin(\text{angulo})$$

Utilice las funciones Math.cos, Math.sin, Math.toRadians para desarrollar la clase AdaptadorPolarRectangular. Las funciones Math.cos y Math.sin reciben como argumento un ángulo en radianes. La función Math.toRadians convierte un ángulo en grados a radianes.

La clase TestAdaptadorPolarRectangular muestra un ejemplo de uso del adaptador.

```
public class TestAdaptadorObjetos {

    public static void main(String[] args) {

        PuntoPolar p1 = new PuntoPolar(5, 40);
        PuntoPolar p2 = new PuntoPolar(15, 120);

        AdaptadorPolarRectangular r1 = new AdaptadorPolarRectangular(p1);
        AdaptadorPolarRectangular r2 = new AdaptadorPolarRectangular(p2);

        System.out.println("Polar      " + p1.getRadio() + " " + p1.getAngulo() +
                           " grados");
        System.out.println("Rectangular " + r1.getX() + " " + r1.getY() + "\n");
        System.out.println("Polar      " + p2.getRadio() + " " + p2.getAngulo() +
                           " grados");
        System.out.println("Rectangular " + r2.getX() + " " + r2.getY() + "\n");

    }

}
```

La salida por la consola:

```
Polar      5.0 40.0 grados
Rectangular 3.83022221559489 3.2139380484326963

Polar      15.0 120.0 grados
Rectangular -7.4999999999999964 12.99038105676658
```

4. [2,0 puntos] Implemente la interfaz `IModaEstadistica` con la clase `ModaArray`. El método `moda()` calcula la moda de un array de números enteros. La moda es el valor más repetido. Por ejemplo, para el array {8, 4, 9, 11, 4, 3, 7, 11, 12, 3, 10, 2, 4 }, la moda es el número 4.

```
public interface IModaEstadistica {
    public double moda();
}

public class ModaArray implements IModaEstadistica {
    // la clase ModaArray implementa la interfaz IModaEstadistica
}

public class TestModaArray {

    public static void main(String[] args) {

        int [] datos = {8, 4, 9, 11, 4, 3, 7, 11, 12, 3, 10, 2, 4 };

        ModaArray m = new ModaArray(datos);
        System.out.println("La moda es " + m.moda());

    }

}
```

La salida por la consola:

```
La moda es 4
```