

Programación orientada a objetos

EJERCICIOS PRÁCTICOS III

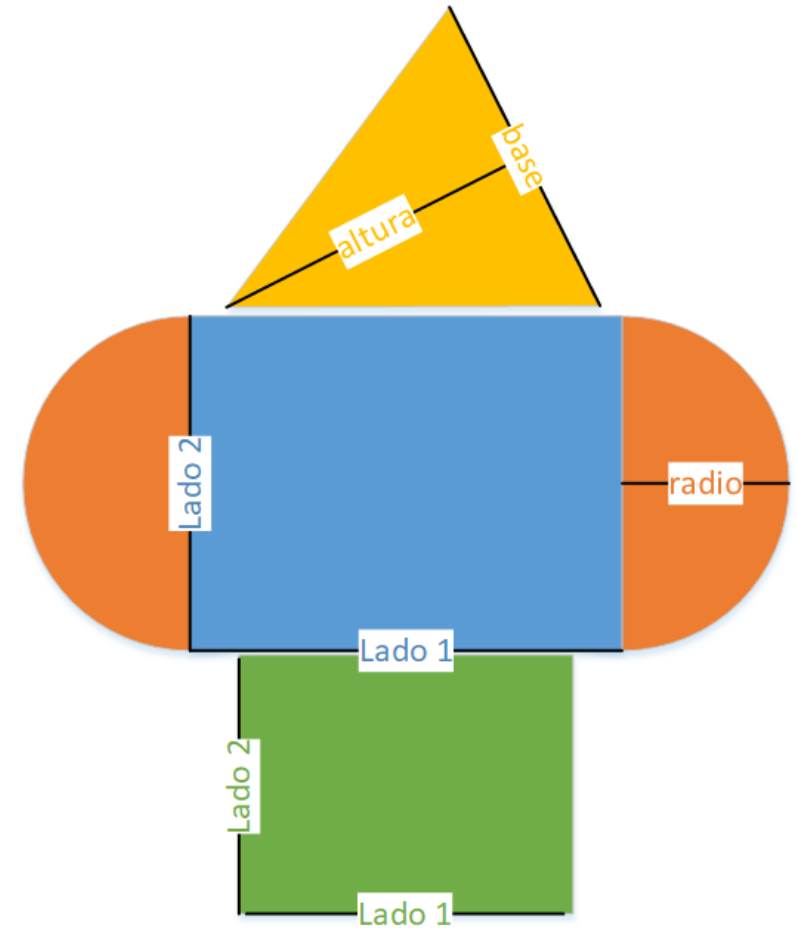
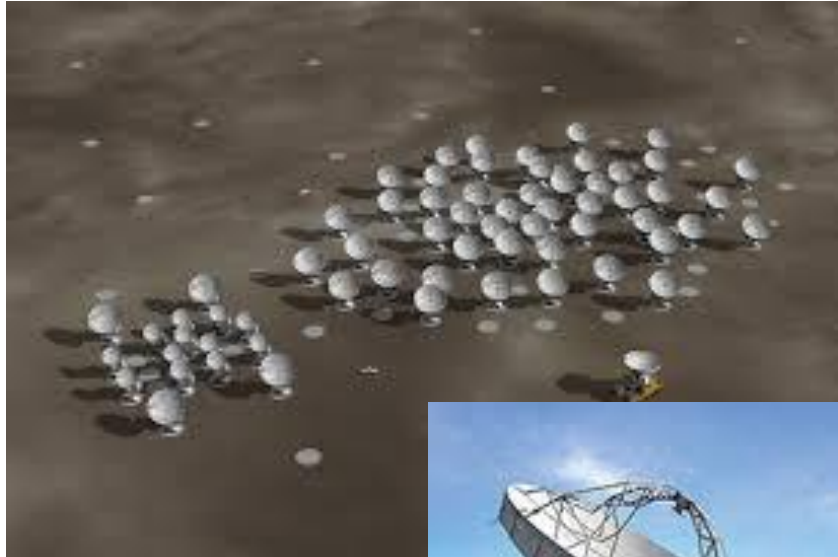
Clase fracción

- Reto:
 - Construir la clase Fraccion.
 - Una fracción se define por su numerador y su denominador (enteros).
 - Se puede inicializar con todas las combinaciones de numerador, denominador y a partir de otra fracción.
 - Hay que implementar los métodos que realizan la suma, resta y multiplicación de fracciones; devolviendo el resultado de la operación.
 - Se podrá imprimir la fracción.
 - Se podrá reducir a un número entero y a un número real.
 - Se podrá simplificar la fracción y comprobar si es reducible o no.
 - Se podrá comprobar si la función es equivalente a otra.

Problema: el nuevo observatorio (I)

- Existen unas instalaciones de atletismo abandonadas que se quieren reconvertir en un observatorio astronómico.
- Es necesario saber cuántas antenas se pueden instalar en la extensión del antiguo complejo deportivo.
- Cada antena ocupa un área equivalente al de un círculo.
- El complejo deportivo está formado por una pista de atletismo, un campo de fútbol y una zona triangular adicional donde se realizaban lanzamientos.
- Resolver el problema empleando POO: calcular el área total disponible y el número de antenas que se podrían instalar.

Problema: el nuevo observatorio (II)



Ejercicios extra...

Clase círculo

- Reto:
 - Construir la clase Círculo.
 - Un círculo se define por su centro y su radio.
 - Se puede inicializar con todas las combinaciones de radio y centro.
 - Se puede desplazar y cambiar de tamaño.
 - Se puede consultar su centro y su radio.
 - Se puede solicitar su área y la longitud de la circunferencia.
 - Se puede comparar con otro círculo para ver cuál es mayor.

Clase rectángulo

- Reto:
 - Construir la clase Rectángulo.
 - Un rectángulo se define por sus esquinas inferior izquierda y superior derecha, se puede instanciar mediante:
 - Indicando estos puntos (no valen todas las combinaciones de puntos).
 - Indicando su base y altura.
 - Copiando las propiedades de otro objeto Rectángulo.
 - Se puede desplazar y cambiar de tamaño.
 - Se puede consultar su base, altura y posición.
 - Se puede solicitar su área y su perímetro.
 - Se puede desplazar.
 - Se pueden dar los datos del rectángulo por teclado.

Interacción con el teclado

```
import java.util.Scanner;

public class Escaner {
    Scanner entradaDelTeclado;
    public Escaner(){
        this.entradaDelTeclado = new Scanner(System.in);
    }

    public String leeCadena(){
        System.out.println("Introduzca cadena de texto: ");
        String cadena = this.entradaDelTeclado.nextLine();
        return cadena;
    }

    public int leeEntero(){
        System.out.println("Introduzca número entero: ");
        int entero = this.entradaDelTeclado.nextInt();
        return entero;
    }

    public double leeReal(){
        System.out.println("Introduzca número real: ");
        double real = this.entradaDelTeclado.nextDouble();
        return real;
    }
}
```


Repaso clase punto

Clase punto

- Reto:
 - Construir la clase Punto.
 - Un punto posee unas coordenadas en un plano bidimensional (x e y).
 - Un punto se puede inicializar de dos modos:
 - Sin argumentos → el punto estará en el origen de coordenadas.
 - Con dos argumentos → coordenadas del punto.
 - Un punto tendrá el siguiente comportamiento:
 - Se podrá trasladar:
 - Indicando el desplazamiento en ambas coordenadas.
 - Indicando el desplazamiento en x , quedando y inalterada.
 - El punto devolverá el valor de su coordenada x .
 - El punto devolverá una copia de sí mismo.
 - El punto podrá informar de su posición mediante un mensaje de consola

Clase punto - resolución (I)

```
//Clase Punto
public class Punto {
    //Atributos de la clase
    private int x, y;

    //Constructor por defecto
    public Punto() {
        x = 0;
        y = 0;
    }

    //Constructor con dos argumentos
    public Punto(int coordenadaX, int coordenadaY) {
        x = coordenadaX;
        y = coordenadaY;
    }
}
```

Clase punto - resolución (II)

//Método que desplaza un punto, modificando los atributos

```
public void desplazar(int desplazamientoX, int desplazamientoY) {  
    x += desplazamientoX; //x = x+desplazamientoX;  
    y += desplazamientoY; //y = y+desplazamientoY;  
}
```

//Método que desplaza un punto, modificando uno de sus atributos

```
public void desplazar(int desplazamientoX) {  
    x += desplazamientoX; //x = x+desplazamientoX;  
}
```

```
/*  
 * Método que imprime por pantalla la posición del punto.  
 * Nótese que no tiene argumentos, trabaja con el propio punto, el argumento implícito.  
 */  
public void posicion() {  
    System.out.println("El valor de la coordenada x es: " + x + "\nEl valor de la coordenada y es: " +  
y);  
}
```

Clase punto - resolución (III)

```
//Método que devuelve la coordenada X del punto.
public int coordenadaX() {
    return x;
}

/*
 * Método que devuelve una copia del punto.
 * El código comentado es equivalente al programado
 */
public Punto copia() {

    Punto c = new Punto(x, y);
    return c;
    //return new Punto (x,y);
}
```

Clase punto - resolución (IV)

```
public static void main(String[] args) {

    Punto puntoUno = new Punto();
    puntoUno.posicion();
    Punto puntoDos = new Punto(3,7);
    puntoDos.posicion();

    System.out.println("\n");

    puntoUno.desplazar(7);
    puntoDos.desplazar(2,4);
    puntoUno.posicion();
    puntoDos.posicion();

    int coordenadaX = puntoUno.coordenadaX();
    System.out.println("\nLa coordenada x del punto uno es: " + coordenadaX);

    System.out.println("\n");

    Punto copia = puntoUno.copia();
    copia.posicion();
    puntoUno.posicion();
}
```