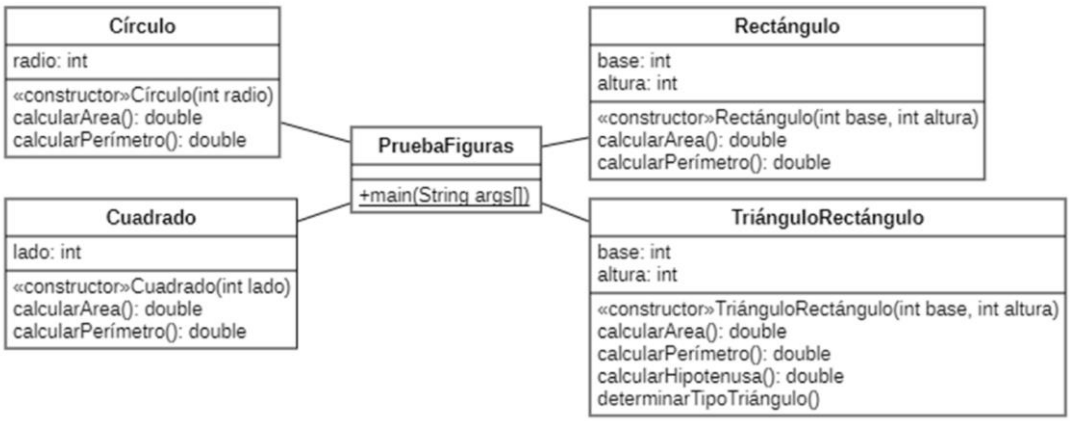
EJERCICIO 2.

Se requiere un programa que modele varias figuras geométricas: el círculo, el rectángulo, el cuadrado y el triángulo rectángulo.

- El círculo tiene como atributo su radio en centímetros.
- El rectángulo, su base y altura en centímetros.
- El cuadrado, la longitud de sus lados en centímetros.
- El triángulo, su base y altura en centímetros.

Se requieren métodos para determinar el área y el perímetro de cada figura geométrica.



CLASE CIRCULO

```
public class Circulo {
   int radio;
   public Circulo(int radio) {
      this.radio = radio;
   }
   double calcularArea(){
      return Math.PI*Math.pow(radio,2);
   }
   double calcularPeri(){
      return 2*Math.PI*radio;
   }
   public int getRadio() {
      return radio;
   }
   public void setRadio(int radio) {
      this.radio = radio;
   }
   public Circulo() {
   }
}
```

CLASE CUADRADO

```
public class Cuadrado {
   int lado;

   public Cuadrado(int lado) {
      this.lado = lado;
   }
   double calcularArea(){
      return lado*lado;
   }
   double calcularPeri(){
      return(4*lado);
   }
   public int getLado() {
      return lado;
   }
   public void setLado(int lado) {
      this.lado = lado;
   }
   public Cuadrado() {
   }
}
```

CLASE RECTANGULO

```
public class Rectangulo {
    int base;
    int altura;

   public Rectangulo(int base, int altura) {
      this.base = base;
      this.altura = altura;
   }

   double calcularArea(){
      return base*altura;
   }
   double calcularPeri(){
      return(2*base)+(2*altura);
   }
}
```

CLASE TRIANGULO RECTANGULO

```
public class TrianguloRectangulo {
   int base;
   int altura;

   public TrianguloRectangulo(int base, int altura) {
```

```
      this.base = base;
      this.altura = altura;
   }
   double calcularArea(){
      return(base*altura/2);
   }
   double calcularPeri(){
      return(base+altura+calcularHipotenusa());
   }
   double calcularHipotenusa(){
      return Math.pow(base*base+ altura*altura,0.5);
   }
}
```

CLASE FIGURAS CON MAIN

```
public class Figuras {

   public static void main(String[] args) {
      Circulo figura1=new Circulo();
      Rectangulo figura2=new Rectangulo(1,2);
      Cuadrado figura3=new Cuadrado();
      TrianguloRectangulo figura4=new TrianguloRectangulo(3,5);
      double r;

      figura3.setLado(5);
      figura1.setRadio(4);
      System.out.println("-----Circulo-----");
      System.out.println("El área del circulo es="+figura1.calcularArea());
      System.out.println("El perímetro del circulo es="+figura1.calcularPeri());
      System.out.println("-----Rectángulo-----");
      System.out.println("El área del Rectángulo es="+figura2.calcularArea());
      System.out.println("El perímetro del Rectángulo es="+figura2.calcularPeri());
      System.out.println("-----Cuadrado-----");
      System.out.println("El área del Cuadrado es="+figura3.calcularArea());
      System.out.println("El perímetro del Cuadrado es="+figura3.calcularPeri());
      System.out.println("-----Triángulo Rectángulo-----");
      System.out.println("El área del Triángulo Rectángulo es="+figura4.calcularArea());
      System.out.println("El perímetro del Triángulo Rectángulo
es="+figura4.calcularPeri());
   }
}
```

EJERCICIO 3.

Se requiere un programa que modele una cuenta bancaria que posee los siguientes atributos:

- Nombres del titular.
- Apellidos del titular.
- Número de la cuenta bancaria.
- Tipo de cuenta: puede ser una cuenta de ahorros o una cuenta corriente.
- Saldo de la cuenta.

Se debe definir un constructor que inicialice los atributos de la clase. Cuando se crea una cuenta bancaria, su saldo inicial tiene un valor de cero. En una determinada cuenta bancaria se puede:

- Imprimir por pantalla los valores de los atributos de una cuenta bancaria.
- Consultar el saldo de una cuenta bancaria.
- Consignar un determinado valor en la cuenta bancaria, actualizando el saldo correspondiente.
- Retirar un determinado valor de la cuenta bancaria, actualizando el saldo correspondiente. Es necesario tener en cuenta que no se puede realizar el retiro si el valor solicitado supera el saldo actual de la cuenta.

Enunciado: clase CuentaBancaria

Se requiere modificar el programa de la cuenta bancaria (del anterior ejercicio) para que realice las siguientes actividades:

Comparar saldos entre cuentas bancarias.

- La cuenta para comparar es un objeto que se envía como parámetro del método. El método devuelve un valor booleano de verdadero si la cuenta actual es mayor o igual a la cuenta que se pasó como parámetro.
- Transferir dinero de una cuenta bancaria a otra. El método debe recibir como parámetro la cuenta de destino y el valor a transferir.
- El saldo de la cuenta actual debe disminuir el valor a transferir y el saldo de la cuenta destino debe aumentar. El método debe reutilizar el método retirar para evaluar si la cantidad a transferir se encuentra en la cuenta de origen.

```
package com.mycompany.cuentabancaria;
public class CuentaBancaria {
   // Atributo que define los nombres del titular de la cuenta bancaria
   String nombresTitular;
   // Atributo que define los apellidos del titular de la cuenta bancaria
   String apellidosTitular;
   // Atributo que define el número de la cuenta bancaria
   int númeroCuenta;
     // Tipo de cuenta como un valor enumerado
   enum tipo {AHORROS, CORRIENTE}
```

```
   // Atributo que define el tipo de cuenta bancaria
   tipo tipoCuenta;
   /* Atributo que define el saldo de la cuenta bancaria con valor inicial cero */
   float saldo = 0;

   public CuentaBancaria(String nombresTitular, String apellidosTitular, int
númeroCuenta, tipo tipoCuenta) {
      this.nombresTitular = nombresTitular;
      this.apellidosTitular = apellidosTitular;
      this.númeroCuenta = númeroCuenta;
      this.tipoCuenta = tipoCuenta;
   }

    void imprimir() {
   System.out.println("Nombres del titular = " + nombresTitular);
   System.out.println("Apellidos del titular = " + apellidosTitular);
   System.out.println("Número de cuenta = " + númeroCuenta);
   System.out.println("Tipo de cuenta = " + tipoCuenta);
   System.out.println("Saldo = " + saldo);
}

    void consultarSaldo() {
System.out.println("El saldo actual es = " + saldo);
}
    boolean consignar(int valor) {
// El valor a consignar debe ser mayor que cero
if (valor > 0) {
saldo = saldo + valor; /* Se actualiza el saldo de la cuenta con el valor consignado */
System.out.println("Se ha consignado $" + valor + " en la cuenta. El nuevo saldo es $" +
saldo);
return true;
} else {
System.out.println("El valor a consignar debe ser mayor que cero.");
return false;
}
}
    boolean retirar(int valor) {
/* El valor debe ser mayor que cero y no debe superar el saldo
actual */
if ((valor > 0) && (valor <= saldo)) {
saldo = saldo - valor; /* Se actualiza el saldo de la cuenta con
el valor retirado */
System.out.println("Se ha retirado $" + valor + " en la cuenta. El nuevo saldo es $" +
saldo);
return true;
```

```
} else {
System.out.println("El valor a retirar debe ser menor que el saldo actual.");
return false;
}
}


/** * Método que compara los saldos de dos cuentas bancarias y * muestra el
resultado en pantalla * @param cuenta Parámetro que define otra cuenta bancaria
con la * cual se va a comparar la cuenta bancaria actual */
void compararCuentas(CuentaBancaria cuenta) {
/* Determina si el saldo de la cuenta actual es mayor o igual que el saldo de la otra
cuenta */
 if (saldo >= cuenta.saldo) {
System.out.println(“El saldo de la cuenta actual es mayor o igual al saldo de la cuenta
pasada como parámetro.”); }
else { System.out.println(“El saldo de la cuenta actual es menor al saldo de la cuenta
pasada como parámetro.”); }
}

public static void main(String[] args) {
        CuentaBancaria cuenta = new
        CuentaBancaria("Pedro","Pérez",123456789,tipo.AHORROS);
        cuenta.imprimir();
        cuenta.consignar(50000);
        cuenta.consignar(60000);
        cuenta.retirar(70000);
   }
```