

Universidad Nacional de San Agustín
Escuela Profesional de Ingeniería de Sistemas
Fundamentos de Programación II
Tema N° 15:
POO – Herencia, Polimorfismo y Tópicos Avanzados

Nombre: Jhonatan Benjamin Mamani Céspedes

CUI: 20232188

Link de GitHub: <https://github.com/JBenjamin01/fp2-24b>

Ejercicio 1:

1. Crear diagrama de clases UML y programa
2. En un software graficador tenemos 3 tipos de figuras: Triángulo, Rectángulo y Círculo.
3. Aplicar polimorfismo usando una sola variable de referencia para que almacene un objeto de cada tipo de figura y llamando al método `toString` (llamada polimórfica) para que muestre los respectivos datos miembro (atributos) y área de cada objeto.
4. Sin crear una superclase para dichas figuras.

Clase Triangulo.java:

```
1  public class Triangulo {
2      private double base;
3      private double altura;
4
5      public Triangulo(double base, double altura) {
6          this.base = base;
7          this.altura = altura;
8      }
9
10     public double calcularArea() {
11         return (base * altura) / 2;
12     }
13
14     @Override
15     public String toString() {
16         return "Triangulo --> [Base: " + base + " - Altura: " + altura + " | Area = " + calcularArea() + "];"
17     }
18 }
```

Clase Rectangulo.java:

```
1 public class Rectangulo {
2     private double largo;
3     private double ancho;
4
5     public Rectangulo(double largo, double ancho) {
6         this.largo = largo;
7         this.ancho = ancho;
8     }
9
10    public double calcularArea() {
11        return largo * ancho;
12    }
13
14    @Override
15    public String toString() {
16        return "Rectangulo --> [Largo: " + largo + " - Ancho: " + ancho + " | Area = " + calcularArea() + "];"
17    }
18 }
```

Clase Circulo.java

```
1 public class Circulo {
2     private double radio;
3
4     public Circulo(double radio) {
5         this.radio = radio;
6     }
7
8     public double calcularArea() {
9         return Math.PI * radio * radio;
10    }
11
12    @Override
13    public String toString() {
14        return "Circulo --> [Radio: " + radio + " | Area = " + calcularArea() + "];"
15    }
16 }
```

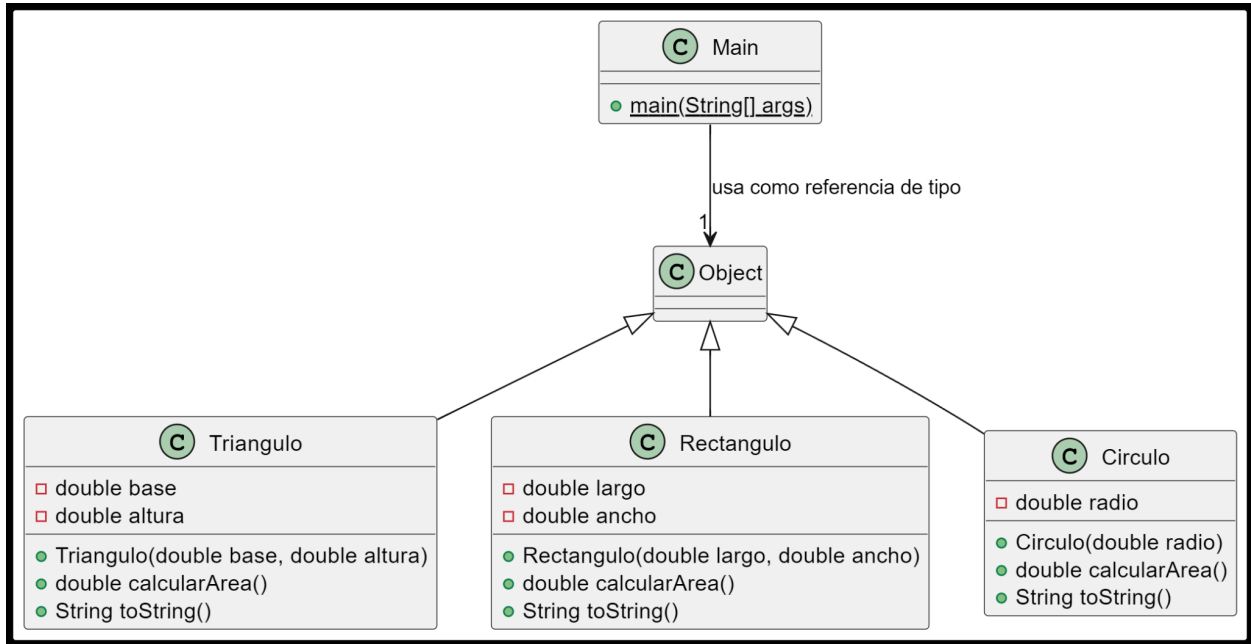
Clase Main.java:

```
1 public class Main {
2     public static void main(String[] args) {
3         Object[] figuras = new Object[3];
4
5         figuras[0] = new Triangulo(7, 9);
6         figuras[1] = new Rectangulo(12, 10.5);
7         figuras[2] = new Circulo(9);
8
9         for (Object f : figuras) {
10             System.out.println(f.toString());
11         }
12     }
13 }
```

Consola:

```
Triangulo --> [Base: 7.0 - Altura: 9.0 | Area = 31.5]  
Rectangulo --> [Largo: 12.0 - Ancho: 10.5 | Area = 126.0]  
Circulo --> [Radio: 9.0 | Area = 254.46900494077323]
```

Diagrama de clases UML (hecho con la extensión de VS Code; Plant UML):



Ejercicio 2:

1. Crear diagrama de clases UML y programa
2. En un software graficador tenemos 3 tipos de figuras: Triángulo, Rectángulo y Círculo.
3. Crear una superclase para dichas figuras.
4. Aplicar polimorfismo creando una sola variable de referencia de la clase padre y usando los métodos polimórficos `toString` (muestra datos miembro y área) y `getArea` (que muestre su respectiva área).

Clase Figura.java:

```
1 public class Figura {
2     public double calcularArea() {
3         return 0;
4     }
5
6     @Override
7     public String toString() {
8         return "Figura sin ninguna propiedad";
9     }
10 }
```

Clase Triangulo.java:

```
1 public class Triangulo extends Figura {
2     private double base;
3     private double altura;
4
5     public Triangulo(double base, double altura) {
6         this.base = base;
7         this.altura = altura;
8     }
9
10    @Override
11    public double calcularArea() {
12        return (base * altura) / 2;
13    }
14
15    @Override
16    public String toString() {
17        return "Triangulo --> [Base: " + base + " - Altura: " + altura + " | Area = " + calcularArea() + "];";
18    }
19 }
```

Clase Rectangulo.java:

```
1 public class Rectangulo extends Figura {
2     private double largo;
3     private double ancho;
4
5     public Rectangulo(double largo, double ancho) {
6         this.largo = largo;
7         this.ancho = ancho;
8     }
9
10    @Override
11    public double calcularArea() {
12        return largo * ancho;
13    }
14
15    @Override
16    public String toString() {
17        return "Rectangulo --> [Largo: " + largo + " - Ancho: " + ancho + " | Area = " + calcularArea() + "];";
18    }
19 }
```

Clase Circulo.java

```
1 public class Circulo extends Figura {
2     private double radio;
3
4     public Circulo(double radio) {
5         this.radio = radio;
6     }
7
8     @Override
9     public double calcularArea() {
10         return Math.PI * radio * radio;
11     }
12
13     @Override
14     public String toString() {
15         return "Circulo --> [Radio: " + radio + " | Area = " + calcularArea() + "];";
16     }
17 }
```

Clase Main.java:

```
1 public class Main {
2     public static void main(String[] args) {
3         Figura f;
4
5         f = new Triangulo(3, 4);
6         System.out.println(f);
7
8         f = new Rectangulo(5, 6);
9         System.out.println(f);
10
11         f = new Circulo(7);
12         System.out.println(f);
13     }
14 }
```

Consola:

```
U:\jdc_ws\Ejercicio 2_c2974974\bin -> Main
Triangulo --> [Base:3.0 - Altura: 4.0 | Area = 6.0]
Rectangulo --> [Largo: 5.0 - Ancho: 6.0 | Area = 30.0]
Circulo --> [Radio: 7.0 | Area = 153.93804002589985]
```

Diagrama de clases UML (hecho con la extensión de VS Code; Plant UML):

