



Instituto Tecnológico de Costa Rica

## Tarea 8

Programación orientada a objetos, relaciones entre objetos, variables y métodos de clase

Jesus Araya Chaves 2021106062

Fabricio Rios Montero 2021473111

Curso: Lenguajes de Programación

Número de grupo: 20

Profesor: Maria Auxiliadora Mora

Primer semestre del 2023

<b>Instanciación de objetos</b>	<b>3</b>
<b>Métodos y variables Estáticos</b>	<b>3</b>
<b>Resultados</b>	<b>4</b>

# Instanciación de objetos

```
//creando 3 puntos
MyPoint p1 = new MyPoint( x: 5, y: 5);
MyPoint p2 = new MyPoint( x: 6, y: 8);
MyPoint p3 = new MyPoint( x: 1, y: 4);
//pruebas de metodos del punto
System.out.println("p1: "+p1.toString());
System.out.println("Distancia entre p1 y un punto xy: "+ p1.distance( x: 8, y: 9));
System.out.println("Distancia entre p1 y p2: "+p1.distance(p2));
System.out.println("Distancia euclideana del p1: "+p1.distance());
//creando un triangulo
MyTriangle t1 = new MyTriangle(p1,p2,p3);
//Creando un triangulo a partir de valores x,y de tres ubicaciones
MyTriangle t2 = new MyTriangle( x1: 0, y1: 0, x2: 4, y2: 4, x3: 8, y3: 0);
MyTriangle t3 = new MyTriangle( x1: 1, y1: 2, x2: 3, y2: 4, x3: 5, y3: 6);
//pruebas de metodos del triangulo
System.out.println("t1: "+t1.toString());
System.out.println("Perimetro de t1: "+t1.getPerimeter());
System.out.println("Tipo de t1: "+t1.getType());
System.out.println("t2: "+t2.toString());
System.out.println("Perimetro de t2: "+t2.getPerimeter());
System.out.println("Tipo de t2: "+t2.getType());
System.out.println("t3: "+t3.toString());
System.out.println("Perimetro de t3: "+t3.getPerimeter());
System.out.println("Tipo de t3: "+t3.getType());
```

## Métodos y variables Estáticos

Para la implementación de crear una variable y un método estático o de clase, se creó una clase estática llamada MathUtils , la cual posee una variable estática que es pi y un método estático llamado calculate Hypotenuse que recibe 2 doubles y retorna la hipotenusa de ese triángulo.

```
no usages
public static final double PI = 3.14159;

3 usages
public static double calculateHypotenuse(double a, double b) {
    return Math.sqrt(Math.pow(a, 2) + Math.pow(b, 2));
}
```

Por lo cual este método se utiliza en el método getType de MyTriangle que ahora también dice si el triángulo es rectángulo aparte de los otros 3.

```
if (MathUtils.calculateHypotenuse(d1, d2) == d3 || MathUtils.calculateHypotenuse(d2, d3) == d1 || MathUtils.calculateHypotenuse(d3, d1) == d2) {  
    return "right";  
}
```

Este método es útil porque no solo se usaría en triángulos si no en múltiples figura que se podrían llegar a implementar, por ejemplo para calcular la diagonal de un rectángulo, por lo cual es un método útil de que sea estático. La variable pi no se utiliza debido que en un triángulo no tiene utilidad pero si se creara una clase circulo tomaron gran importancia porque hasta se podría calcular el área de un círculo circunscrito a un triángulo pero no es el caso y pi no se utiliza.

Ejemplo de ejecución:

```
t2: Triangle (0,0), (4,4), (8,0)  
Perimetro de t2: 19.31370849898476  
Tipo de t2: right
```

Como no se utilizó se creó una variable estática count en MyTriangle que cuenta cuántos triángulos se han creado.

```
3 usages  
private static int count = 0;
```

Ejemplo en ejecución

```
Cantidad de triangulos creados: 3
```

## Resultados

Resultado final de todo el código:

```
p1: (5,5)  
Distancia entre p1 y un punto xy: 5.0  
Distancia entre p1 y p2: 3.1622776601683795  
Distancia euclidea de p1: 7.0710678118654755  
t1: Triangle (5,5), (6,8), (1,4)  
Perimetro de t1: 13.688507523218888  
Tipo de t1: scalene  
t2: Triangle (0,0), (4,4), (8,0)  
Perimetro de t2: 19.31370849898476  
Tipo de t2: right  
t3: Triangle (1,2), (3,4), (5,6)  
Perimetro de t3: 11.313708498984761  
Tipo de t3: isosceles  
Cantidad de triangulos creados: 3
```