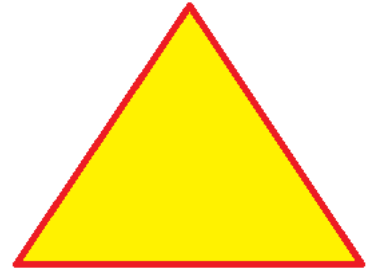


## **Trabajo Práctico N° 7 – Programación 2**

**Alumno: Gonzalo Bravo**

### **Objeto Triángulo**

A partir del análisis de este objeto en el TP 5, se podría generalizar a un concepto más abstracto como el de **Objeto Geométrico**. Para una representación más abstracta que puede especializarse en diferentes tipos de polígonos, como triángulos, rectángulos, cuadrados, etc., e identificar sus partes en términos de otros objetos como números y texto.



**Elección de contexto:** Figuras geométricas.

**Descripción del contexto:** Geometría. Se utilizan en diferentes contextos, por ejemplo en áreas como la matemática, física, ingeniería, arquitectura, etc.

Por ejemplo, en el contexto de la arquitectura, para la construcción de una casa, implica trabajar con diferentes formas geométricas para representar las estructuras y características de la casa: techo, paredes, habitaciones, puertas y ventanas. Utilizaríamos cálculos y figuras geométricas como triángulos para el techo, rectángulos para las paredes, cuadrados para las puertas y ventanas, círculos para las columnas, etc.

**Generalización:** Objeto polígono. Tomando las características comunes de varias formas geométricas que comparten ciertas propiedades.

**Especialización:** Triángulo, Rectángulo, Cuadrado, Pentágono, Hexágono, etc.

### **Instancias de clases concretas:**

#### **Triángulo**

- Equilátero (todos los lados son iguales)
- Triángulo Isósceles (dos lados son iguales)
- Triángulo Escaleno (todos los lados son diferentes)

#### **Rectángulo**

- Rectángulo Cuadrado (todos los lados son iguales)
- Rectángulo Recto (uno de los ángulos es recto)

#### **Cuadrado**

- Cuadrado Regular (todos los lados son iguales)
- Cuadrado Recto (uno de los ángulos es recto)

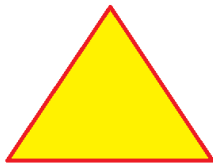
### Partes identificadas en términos de otros objetos:

-Números: Longitud de los lados, altura, coordenadas de los vértices, ángulos.

-Texto: Nombre del color de línea, nombre del color de relleno.

Por ejemplo, si tomamos los triángulos, todos los objetos tienen la misma estructura interna. El comportamiento dependerá del estado interno de cada objeto, en el siguiente caso son similares:

Triángulo (10, 10, 10, "Rojo", "Amarillo")



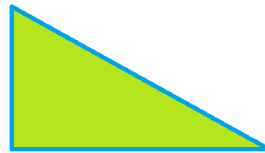
#### Comportamiento

calcular área /  
calcular perímetro /  
obtener color de línea  
obtener color de relleno /

#### Estado Interno

lado 1: 10  
lado 2: 10  
lado 3: 10  
colorDeRelleno (Amarillo)  
**colorDeLínea (Rojo)**

Triángulo (10, 20, 22.36, "Azul", "Verde")



#### Comportamiento

calcular área /  
calcular perímetro /  
obtener color de línea /  
obtener color de relleno /

#### Estado Interno

lado 1: 10  
lado 2: 20  
lado 3: 22.36  
colorDeRelleno (Verde)  
**colorDeLínea (Azul)**

## Diagramas UML

<b>Triángulo</b>
<b>Atributos</b>
<ul style="list-style-type: none"><li>-Lado1 (longitud): double</li><li>-Lado2 (longitud): double</li><li>-Lado3 (longitud): double</li><li>-Altura (longitud): double</li><li>-Angulo1 (grados): double</li><li>-Angulo2 (grados): double</li><li>-Angulo3 (grados): double</li><li>-Vértice1 (coordenadas): double x, y</li><li>-Vértice2 (coordenadas)</li><li>-Vértice3 (coordenadas)</li><li>-Color de línea (nombre del color): String</li><li>-Color de relleno (nombre del color): String</li></ul>
<b>Métodos</b>
<ul style="list-style-type: none"><li>+ double calcularArea() / <math>(base * altura) \div 2</math></li><li>+ double calcularPerímetro() / lado 1 + lado 2 + lado 3 (la suma de los tres lados del triángulo)</li><li>+ double calcularAltura() / Según el tipo de triángulo</li><li>+ double calcularAngulos() / Razones trigonométricas</li><li>+ double calcularVertice(x, y) / Según los lados y ángulos</li><li>+ String obtenerTipoDeTriángulo() / Según sus lados o ángulos</li><li>+ String obtenerColorDeLinea() / Según su nombre o código</li><li>+ String obtenerColorDeRelleno() / Según su nombre o código</li></ul>

## Objetos Geométricos

### Polígono

#### Atributos

-Cantidad de Lados: *int*  
 -Longitud de los lados: *double*  
 -Ángulos: *double*

#### Métodos

+ *double* **calcularArea()** /  $(base * altura) \div 2$   
 + *double* **calcularPerímetro()** / (la suma de los lados)  
 + *String* **obtenerTipoDePolígono()** / Según sus lados o ángulos

#### Triángulo

##### Atributos

-Lado1 (longitud): *double*  
 -Lado2 (longitud): *double*  
 -Lado3 (longitud): *double*  
 -Altura (longitud): *double*  
 -Angulo1 (grados): *double*  
 -Angulo2 (grados): *double*  
 -Angulo3 (grados): *double*  
 -Vértice1 (coordenadas): *double* x, y  
 -Vértice2 (coordenadas) *double* x, y  
 -Vértice3 (coordenadas) *double* x, y  
 -Color de línea (nombre del color): *String*  
 -Color de relleno (nombre del color): *String*

##### Métodos

+ *double* **calcularArea()** /  $(base * altura) \div 2$   
 + *double* **calcularPerímetro()** / lado 1 + lado 2 + lado 3 (la suma de los tres lados del triángulo)  
 + *double* **calcularAltura()** / Según el tipo de triángulo  
 + *double* **calcularAngulos()** / Razones trigonométricas  
 + *double* **calcularVértice(x, y)** / Según los lados y ángulos  
 + *String* **obtenerTipoDeTriángulo()** / Según sus lados o ángulos  
 + *String* **obtenerColorDeLinea()** / Según su nombre o código  
 + *String* **obtenerColorDeRelleno()** / Según su nombre o código

#### Rectángulo

##### Atributos

-Lado1 (longitud): *double*  
 -Lado2 (longitud): *double*  
 -Lado3 (longitud): *double*  
 -Lado4 (longitud): *double*  
 -Altura (longitud): *double*  
 -Angulo1 (grados): *double*  
 -Angulo2 (grados): *double*  
 -Angulo3 (grados): *double*  
 -Angulo4 (grados): *double*  
 -Vértice1 (coordenadas): *double* x, y  
 -Vértice2 (coordenadas): *double* x, y  
 -Vértice3 (coordenadas): *double* x, y  
 -Color de línea (nombre del color): *String*  
 -Color de relleno (nombre del color): *String*

##### Métodos

+ *double* **calcularArea()** /  $(base * altura)$   
 + *double* **calcularPerímetro()** / lado 1 + lado 2 + lado 3 + lado 4 (la suma de los cuatro lados del triángulo)  
 + *double* **calcularAltura()** / Según el lado  
 + *double* **calcularAngulos()** / Razones trigonométricas  
 + *double* **calcularVértice(x, y)** / Según los lados y ángulos  
 + *String* **obtenerTipoDeRectángulo()** / Según sus lados o ángulos  
 + *String* **obtenerColorDeLinea()** / Según su nombre o código  
 + *String* **obtenerColorDeRelleno()** / Según su nombre o código

#### Cuadrado

##### Atributos

-Lado1 (longitud): *double*  
 -Lado2 (longitud): *double*  
 -Lado3 (longitud): *double*  
 -Lado4 (longitud): *double*  
 -Altura (longitud): *double*  
 -Angulo1 (grados): *double*  
 -Angulo2 (grados): *double*  
 -Angulo3 (grados): *double*  
 -Angulo4 (grados): *double*  
 -Vértice1 (coordenadas): *double* x, y  
 -Vértice2 (coordenadas) *double* x, y  
 -Vértice3 (coordenadas) *double* x, y  
 -Color de línea (nombre del color): *String*  
 -Color de relleno (nombre del color): *String*

##### Métodos

+ *double* **calcularArea()** /  $lado^2$   
 + *double* **calcularPerímetro()** / lado 1 + lado 2 + lado 3 + lado 4 (la suma de los cuatro lados del cuadrado)  
 + *double* **calcularAltura()** / lado  
 + *double* **calcularAngulos()** / Razones trigonométricas  
 + *double* **calcularVértice(x, y)** / Según los lados y ángulos  
 + *String* **obtenerTipoDeCuadrado()** / Según sus lados o ángulos  
 + *String* **obtenerColorDeLinea()** / Según nombre o código  
 + *String* **obtenerColorDeRelleno()** / Según nombre o código

