

# CONSTRUCTORES - INTERFAZ PÚBLICA

---

Blautzik - Gasior - Videla

Julio - Diciembre de 2021

**Definición:**

Un constructor es una subrutina cuyo propósito es inicializar el estado de un objeto

# Declaración y definición

```
class <nombre de la clase> {  
    <nombre de la clase>(<parametros>) {  
        sentencia 1;  
        sentencia 2;  
        ...  
        sentencia n;  
    }  
}
```

# CONSTRUCTORES VS. MÉTODOS

Los constructores, a diferencia de los métodos:

- Se ejecutan automáticamente al momento de instanciar un objeto.
- Son rutinas de inicialización, por lo tanto no devuelven valores ni pueden ser invocados.

# MODIFICADORES DE ACCESO

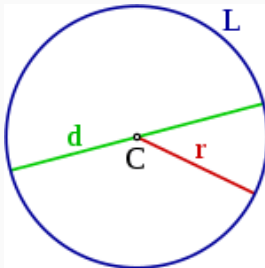
Permiten limitar el acceso a los miembros de una clase Los modificadores de acceso preceden a la declaración de un miembro de la clase

Modificador	Visibilidad
<b>public</b>	desde cualquier otra clase
(no se indica nada)	solo dentro del paquete
<b>protected</b>	dentro de la misma clase o de sus clases derivadas
<b>private</b>	solo dentro de la misma clase

## **Lo que otras clases ven de una clase:**

- Declaración de los constructores públicos.
- Declaración de los métodos públicos.

## Veamos un ejemplo completo: class Circulo



**Figure 1:** Círculo

Seleccionamos las características que sean relevantes para nuestro problema: radio, diámetro, perímetro, area.

# Pensamos en objetos y elegimos qué será público y qué será privado

## Diseñamos Atributos y Constructores

```
class Circulo {  
  
    private double radio; // no accesible  
  
    public Circulo(double radioInicial) { //accesible  
        setRadio(radioInicial);  
    }  
  
    public Circulo() { //accesible  
        setRadio(1.0);  
    }  
}
```



## La interfaz pública - Los Setters

```
public void setRadio(double r) {  
    if(r < 0 )  
        throw new Error("No se aceptan radios negativos");  
    radio = r;  
}  
  
public void setDiametro(double diametro) {  
    setRadio(diametro / 2);  
}
```

## La interfaz pública - Los Getters

```
public double getRadio() {  
    return radio;  
}  
  
public double getDiametro(){  
    return getRadio() * 2;  
}  
  
public double getPerimetro() {  
    return getDiametro() * Math.PI;  
}  
  
public double getArea() {  
    return Math.PI * Math.pow(getRadio(),2);  
}  
}
```