

# Polimorfismo

11

Programación II y Laboratorio de Computación II

Edición 2018

# Polimorfismo

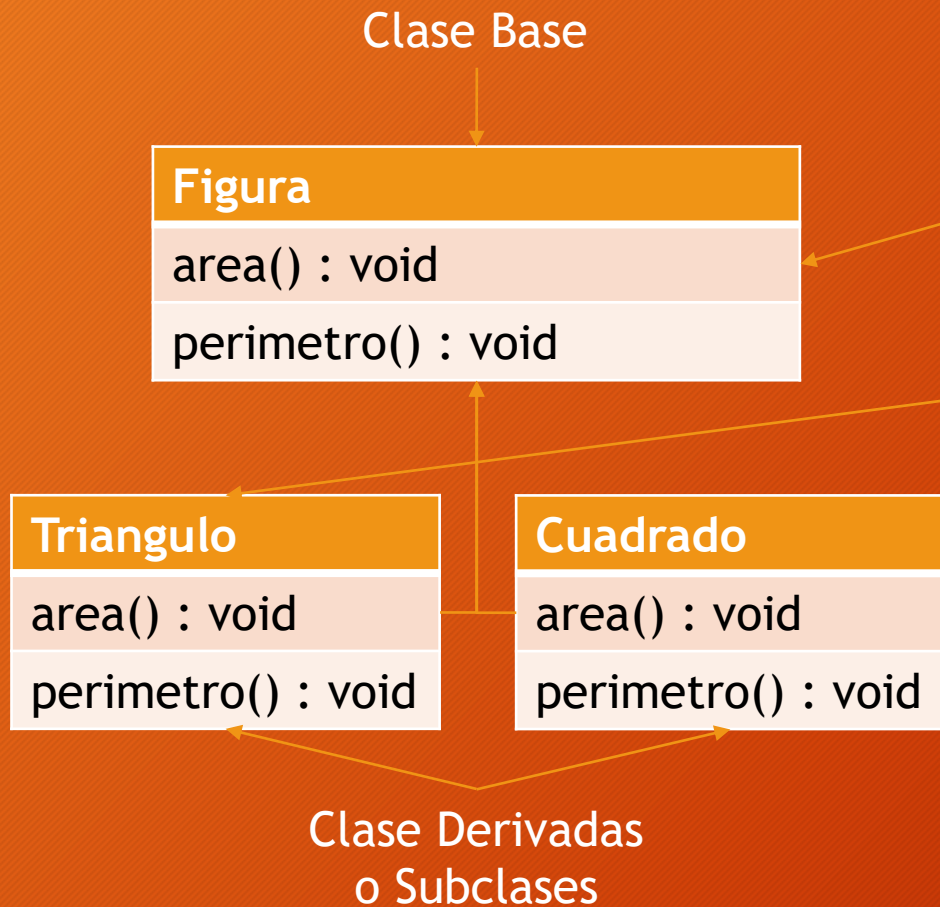
- Es la propiedad que tienen los objetos de permitir invocar genéricamente un comportamiento (método) cuya implementación será delegada al objeto correspondiente recién en tiempo de ejecución.
- En otras palabras, es la capacidad de tratar objetos diferentes de la misma forma.
- El polimorfismo tiende a existir en las relaciones de herencia.
- El polimorfismo basado en herencia implica la definición de métodos en una clase base y sobrescribirlos con nuevas implementaciones en clases derivadas.

# Polimorfismo

- La definición del método reside en la clase base.
  - Se utiliza *virtual*.
- La implementación del método reside en la clase derivada.
  - Se utiliza *override*.
- La invocación es resuelta en tiempo de ejecución.



# Polimorfismo



```
Figura f = new Figura();  
f.area();
```

```
Triangulo t = new Triangulo();  
t.area();
```

```
Cuadrado c = new Cuadrado();  
c.area();
```

```
Figura h = new Cuadrado();  
h.area();
```

# Virtual y Override

- El polimorfismo implica la definición de métodos y/o propiedades en una clase base, mediante el uso de la palabra reservada **virtual** y sobrescribirlos con nuevas implementaciones en clases derivadas con la palabra reservada **override**.

```
[modificadores] virtual retorno Metodo( [Args])  
{  
    // Implementación del método en clase Base  
}
```

```
[modificadores] override retorno Metodo( [Args])  
{  
    // Implementación del método en clase Derivada  
}
```

# Métodos: Virtual y Override

```
class ClaseBase
{
    public virtual void NombreMétodo()
    {
        // Implementación del método
    }
}
class ClaseDerivada : ClaseBase
{
    public override void NombreMétodo()
    {
        base.NombreMétodo(); // Llamada al método virtual
        // Implementación específica
    }
}
```

# Propiedades: Virtual y Override

```
class ClaseBase {  
    public virtual string NombreProp  
    {  
        get { // Implementación }  
        set { // Implementación }  
    }  
}  
  
class ClaseDerivada : ClaseBase {  
    public override string NombreProp  
    {  
        get { // Implementación }  
        set { // Implementación }  
    }  
}
```