

Polimorfismo



Referencias a objetos hijos

- En Java, es posible almacenar en una variable de una clase cualquier objeto de sus subclases

```
Object ob=new String("es una cadena");  
:  
Figura f=new Circulo(4,"Verde");
```

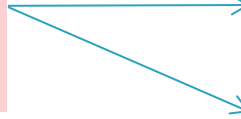
- A través de esa variable podemos llamar a métodos del objeto hijo, que estén presentes también en la superclase

```
ob.toString(); //ok  
ob.length();//error, length no está presente en Object  
:  
f.superficie(); //ok  
f.setRadio();// error, ese método es propio de Circulo, no está  
//presente en Figura
```

Polimorfismo: Definición

➤ Aplicado en la herencia, el polimorfismo consiste en poder utilizar una misma expresión para llamar a diferentes versiones de un mismo método

La misma instrucción permite llamar a dos versiones diferentes del método superficie



```
Figura f=new Circulo(4,"Verde");  
f.superficie(); //versión de Circulo  
f=new Triangulo(6,2,"Rojo");  
f.superficie(); //versión de Triangulo
```

➤ Ventajas: Reutilización y simplificación de código

Ejemplos polimorfismo

➤ Algunos ejemplos reales de polimorfismo en clases Java Estándar:

- `contains(Object ob)`. Método *contains()* de `HashSet`
- `PrintStream(OutputStream destino)`. Constructor de `PrintStream`
- `add(Component cmp)`. Método *add()* de `Container`

➤ Sin polimorfismo, habría que disponer de un método por cada tipo de objeto con el que operar.