

Polimorfismo

Dr. Mario Marcelo Berón

Polimorfismo



Polimorfismo

Polimorfismo: Muchas formas.

Concepto

Hace referencia a que un nombre puede denotar instancias de muchas clases diferentes siempre y cuando estén relacionadas por alguna superclase común. Cualquier objeto denotado por este nombre es capaz de responder a un conjunto de operaciones de diversas formas.

Polimorfismo

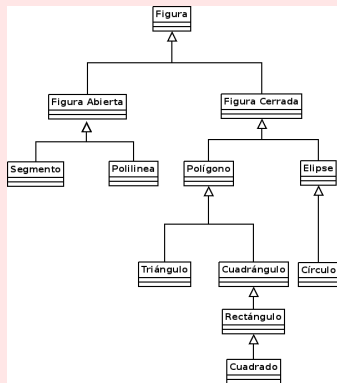


Propiedades

- Provee otra dimensión de separación de interfaz de implementación.
- Permite desacoplar el Qué del Cómo.
- Permite organizar mejor el código y la legibilidad, como así también la creación de programas extensibles.
- El desacople del polimorfismo es en términos de tipo.
- También se conoce como: *Ligadura Dinámica*, *Ligadura Tardía*, *Ligadura de Tiempo de Ejecución*.

Polimorfismo

Asignaciones Polimórficas



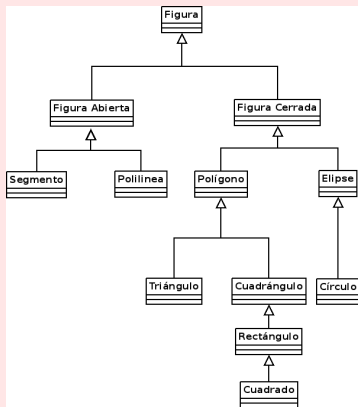
```
Polígono p;  
Rectángulo r = new Rectángulo();  
Triángulo t = new Triángulo();  
    p:=r;  
    ....  
    p:=t;
```

Asignación Polimórfica

El tipo de la fuente es distinto del tipo del destino. Un objeto como *p* es una *Objeto Polimórfico*.

Polimorfismo

Función Polimórficas



```
void f(Polígono p) {
    ....
}

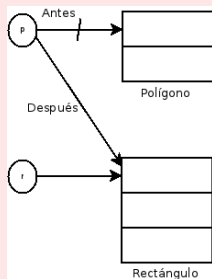
....
f(r);
....
f(t);
```

Comentarios

p es una *Entidad Polimórfica*. f es una función polimórfica.

Polimorfismo

Asignaciones Polimórficas: ¿Qué sucede?



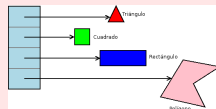
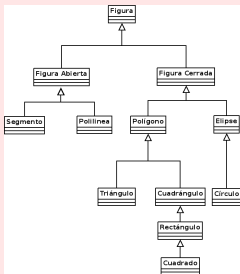
Todas las entidades previamente mencionadas son tipos referenciados. Los valores para p y r no son objetos sino *Referencias a Objetos*.

Importante

- Los objetos una vez creados no cambian su tipo sólo cambian sus referencias.
- El polimorfismo no tiene una gran penalidad de eficiencia.
- El polimorfismo se permite para *Tipos Referenciados* no *Tipos Extendidos*.

Polimorfismo

Estructuras de Datos Polimórficas

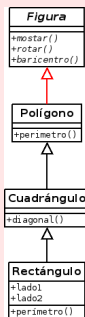


Ejemplo

```
Polígono poliArr=new Polígono[10];
Polígono p= new Polígono ();
Rectángulo r= new Rectángulo();
Cuadrado c= new Cuadrado();
Triángulo t= new Triángulo();
poliArr[1]=p;
poliArr[2]=r;
poliArr[3]=c;
poliArr[4]=t;
```

Polimorfismo

Estructuras de Datos Polimórficas



```
Polígono p= new Polígono();
Rectángulo r= new Rectángulo();
float l1,l2;

    p.perimetro();
    p.mostrar();
    p.rotar();
    r.diagonal();
    l1=r.lado1;
    l2=r.lado2;
    r.mover();
    r.rotar();
    r.perímetro();
    l1=p.lado1;
    l2=p.lado2;
```


Polimorfismo

Tipo Estático

Tipo usado para declarar una entidad.

```
Polígono p;
```

El tipo estático de p es Polígono.

Tipo Dinámico

Tipo al cual una referencia se vincula en tiempo de ejecución.

```
Polígono p;  
Rectángulo r;  
p:=r;
```

El tipo estático de p es Polígono. El tipo estático de r es Rectángulo. Después de la asignación el tipo dinámico de p es rectángulo.

Polimorfismo

Sustitución

- 1 Siempre que el valor de un cierto tipo es esperado, un subtipo puede ser provisto.
- 2 En el contexto de herencia, todas las subclases pueden ser tratadas como una clase raíz común.
- 3 Simplifica el código y facilita la reutilización.

Ligadura Dinámica

Los tipos de las variables se conocen en tiempo de ejecución.

Regla de la Ligadura Dinámica

La forma dinámica del objeto determina que operación aplicar.

Polimorfismo

Ligadura Dinámica

```
Polígono p;  
....  
if op=0 then p:= new Rectángulo();  
elseif opción=1 then p:= new Triángulo();  
elseif  
  ...  
end  
....  
x=p.perímetro();
```

Pregunta

¿Qué *perímetro()* se invoca?

Polimorfismo

Upcast-Downcast

- El *upcast* se utiliza para moverse hacia arriba en la jerarquía de herencia.
- El *downcast* se utiliza para moverse hacia abajo en la jerarquía de herencia.

Comentarios

Cuando se realiza un *upcast* siempre se está seguro de que no habrá problemas de tipos debido a que la interfaz de la clase base no es más grande que la interfaz de la clase derivada. No sucede lo mismo con el *downcast* porque no es posible conocer el tipo del objeto que se desea tratar.