

Polimorfismo

Tecnología de Programación



Adolfo Muñoz – Juan Magallón
Grado en Ingeniería Informática



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza

Polimorfismo



Se denomina **polimorfismo** a la característica de un lenguaje de programación que permite que un mismo elemento del código pueda representar distintos elementos del lenguaje durante la ejecución del programa.



Ejemplos en los que podríamos encontrar **polimorfismo**:

- Identificador de variable / atributo.
- Identificador de tipo / clase.
- Identificador de función / método.
- Fragmento de código

No hay que confundir el polimorfismo con la ausencia de tipado fuerte en un lenguaje (Python).



El **polimorfismo** se suele aplicar en lenguajes tipados permitiendo representar **diferentes tipos de datos** mediante los mismos elementos sintácticos del lenguaje:

- Estructuras de datos que con el mismo código puedan estar asociadas a diferentes tipos de datos.
- Funciones y/o métodos que con el mismo código trabajen con distintos tipos de datos.

Objetivo: Reutilización de código y representación única.



Pregunta



¿ Cómo se consigue representar **diferentes tipos de datos** con los mismos elementos del lenguaje ?



Pregunta



¿ Cómo se consigue representar **diferentes tipos de datos** con los mismos elementos del lenguaje ?

Respuesta



Permitiendo definir **identificadores genéricos** de tipos que puedan representar a diferentes tipos de datos.

Dichos identificadores genéricos están sin concretar en el momento de escribir el código polimórfico.

Distintos lenguajes implementan o simulan el polimorfismo de diversas formas.



Si el tipo de un dato puede no estar concretado, surgen algunas preguntas:

- ¿ Cómo defino variables o parámetros sin fijar su tipo ?
- ¿ Cómo los almaceno ?
- ¿ Qué puedo hacer con algo que no sé qué es ?

Dadas estas restricciones, los algoritmos que se pueden implementar de forma polimórfica son relativamente limitados, pero aún así son extremadamente útiles
(contenedores, gestión de datos, ...)

Tipos de Polimorfismo:

- **Polimorfismo *ad-hoc* o *a medida***

El código ejecutado (función, método) es distinto según los tipos de datos involucrados.

Este mecanismo también se denomina **sobrecarga**.



Tipos de Polimorfismo:

- **Polimorfismo de inclusión**

Si existe una relación de subtipado entre diferentes tipos de datos, se puede definir código que trabaje con el tipo base y sus subtipos relacionados.

- 1 -- ADA
- 2 **subtype** DiaDelMes **is** integer **range** 1..31;
- 3 **subtype** Spanish **is** Person(country => "Spain");

En POO el mecanismo de subtipado se denomina **herencia**.

Tipos de Polimorfismo:

- **Polimorfismo paramétrico**

El mismo código es capaz de operar sobre distintos tipos de datos, sin que exista ninguna relación entre dichos tipos.

Este mecanismo también se denomina **programación genérica**.



Sobrecarga



Polimorfismo *ad-hoc* o sobrecarga

Se denomina **polimorfismo *ad-hoc*** (*para esto*, a medida) a la característica de un lenguaje de programación que permite escribir código que utilice el **mismo identificador** para denominar **diferentes elementos** del lenguaje, normalmente funciones.

Decimos que ese identificador está **sobrecargado**.

El lenguaje debe ser capaz de diferenciar entre cada uno de dichos elementos en función del contexto.



```
1 int max(int a, int b) {  
2     if (a>b) return a; else return b;  
3 }  
4 float max(float a, float b) {  
5     if (a>b) return a; else return b;  
6 }  
7 int main() {  
8     cout << max(3,5) << " - " << max(3.3f,5.5f) << endl;  
9 }
```



```
1 int max(int a, int b) {  
2     if (a>b) return a; else return b;  
3 }  
4 float max(float a, float b) {  
5     if (a>b) return a; else return b;  
6 }  
7 int main() {  
8     cout << max(3,5.5f) << endl;  
9 }
```



Pregunta:

¿ Qué hace este programa ?

```
1 class Sobrecarga {
2     static int suma(int a, int b) { return a+b; }
3     static double suma(double a, double b) { return a+b; }
4     static double suma(double a, double b, double c) {
5         return a+b+c;
6     }
7     public static void main(String args[]) {
8         System.out.println(suma(1,2));
9         System.out.println(suma(1.1, 2.3));
10        System.out.println(suma(1.1, 2.3, 3.4));
11    }
12 }
```



```
1 class Sobrecarga {  
2     static int suma(int a, int b) { return a+b; }  
3     static double suma(double a, double b) { return a+b; }  
4     static double suma(double a, double b, double c) {  
5         return a+b+c;  
6     }  
7     public static void main(String args[]) {  
8         System.out.println(suma(1,2.0));  
9     }  
10 }
```



Pregunta:

¿ Qué hace este programa ?

Polimorfismo

Tecnología de Programación



Adolfo Muñoz – Juan Magallón
Grado en Ingeniería Informática



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza