

# Polimorfisme

El polimorfisme és la característica de la programació orientada a objectes que permet modificar la instància d'un objecte en temps d'execució basat en una jerarquia de herència.

Polimorfisme és la capacitat d'un objecte d'adquirir varies formes segons el context.

## Principi de substitució:

Es pot fer servir un objecte de la subclasse sempre que el programa esperi un objecte de la superclasse.

Un mateix objecte pot tenir diferent comportament en funció del context en el que es faci servir. Es diu que les variables objecte són polimòrfiques.

## Exemple 1:

Classe empleado amb una classe que hereda jefe.

Tenim dos mètodes que es diuen igual però el de la classe Jefe ens retorna el sou més un incentiu i el de la classe empleat no té incentiu, retorna només el sou.

ClsEmpleado

ClsJefe

Principal

```
class ClsEmpleado
```

```
{
```

```
    private double _Sueldo;
```

```
    private string _Nombre;
```

```
    private string _Departamento;
```

```
4 referencias
```

```
public ClsEmpleado(string d, double s, string n ) {
```

```
    this.Sueldo = s;
```

```
    this.Nombre = n;
```

```
    this.Departamento = d;
```

```
}
```

```
3 referencias
```

```
protected double Sueldo { get => _Sueldo; set => _Sueldo = value; }
```

```
1 referencia
```

```
protected string Nombre { get => _Nombre; set => _Nombre = value; }
```

```
1 referencia
```

```
protected string Departamento { get => _Departamento; set => _Departamento = value; }
```

```
2 referencias
```

```
public virtual double DameSueldo()
```

```
{
```

```
    return this.Sueldo;
```

```
}
```

```
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace ProyectoPolimorfismo
```

```
{
```

4 referencias

```
class ClsJefe:ClsEmpleado
```

```
{
```

```
    private double incentivo;
```

2 referencias

```
    public ClsJefe(string d, double s, string n, double i) : base(d,s,n) {
```

```
        this.incentivo = i;
```

```
    }
```

2 referencias

```
    public override double DameSueldo()
```

```
    {
```

```
        return this.Sueldo + this.incentivo;
```

```
    }
```

```
}
```

```
}
```

```
namespace ProyectoPolimorfismo
```

```
{
```

```
    0 referencias
```

```
    class Program
```

```
    {
```

```
        0 referencias
```

```
        static void Main(string[] args)
```

```
        {
```

```
            ClsEmpleado[] Listaempleado = new ClsEmpleado[5];
```

```
            Listaempleado[0] = new ClsEmpleado("marqueting", 1000.80, "Pedro");
```

```
            Listaempleado[1] = new ClsEmpleado("marqueting", 1000.80, "Pedro");
```

```
            Listaempleado[2] = new ClsEmpleado("marqueting", 1000.80, "Pedro");
```

```
            ClsJefe jefe1 = new ClsJefe("administración", 1000.60, "Laura", 200.75);
```

```
            Listaempleado[3] = jefe1;
```

```
            Listaempleado[4] = new ClsJefe("administración", 1000.60, "Leo", 210.75);
```

```
            int i = 1;
```

```
            foreach (ClsEmpleado empleado in Listaempleado)
```

```
            {
```

```
                Console.WriteLine("{0} Tiene el sueldo:{1}", i, empleado.DameSueldo());
```

```
                i++;
```

```
            }
```

```
            Console.ReadKey();
```

```
        }
```

El polimorfisme ens permet poder fer lo següent: polimorfisme

```
namespace ProyectoPolimorfismo
```

```
{
```

```
0 referencias
```

```
class Program
```

```
{
```

```
0 referencias
```

```
static void Main(string[] args)
```

```
{
```

```
    ClsEmpleado[] Listaempleado = new ClsEmpleado[5];
```

```
    Listaempleado[0] = new ClsEmpleado("marqueting", 1000.80, "Pedro");
```

```
    Listaempleado[1] = new ClsEmpleado("marqueting", 1000.80, "Pedro");
```

```
    Listaempleado[2] = new ClsEmpleado("marqueting", 1000.80, "Pedro");
```

```
    ClsJefe jefe1 = new ClsJefe("administración", 1000.60, "Laura", 200.75);
```

```
    Listaempleado[3] = jefe1;
```

```
    Listaempleado[4] = new ClsJefe("administración", 1000.60, "Leo", 210.75);
```

```
    int i = 1;
```

```
    foreach (ClsEmpleado empleado in Listaempleado)
```

```
    {
```

```
        Console.WriteLine("{0} Tiene el sueldo:{1}", i, empleado.DameSueldo());
```

```
        i++;
```

```
    }
```

```
    Console.ReadKey();
```

```
}
```

Instàncies de la classe ClsEmpleado

ListaEmpleado és un array  
de instàncies de la classe ClsEmpleado.





```
1 Tiene el sueldo:1000,8  
2 Tiene el sueldo:1000,8  
3 Tiene el sueldo:1000,8  
4 Tiene el sueldo:1201,35  
5 Tiene el sueldo:1211,35
```

I en el cas de recorrer el array, el mètode DameSueldo() és el mètode de la classe ClsEmpleado , però al invocar-lo en la quarta i quinta instància, s'executa el de la classe ClsJefe i no el de la classe ClssEmpleado.



9 referencias

```
class ClsEmpleado
```

```
{    static int    Num=0;
    private double _Sueldo;
    private string _Nombre;
    private string _Departamento;
```

4 referencias

```
public ClsEmpleado(string d, double s, string n ) {
```

```
    this.Sueldo = s;
```

```
    this.Nombre = n;
```

```
    this.Departamento = d;
```

```
    Num++;
```

```
}
```

1 referencia

```
public static void NumEmpleados()
```

```
{
```

```
    Console.WriteLine("Num empleados"+Num);
```

```
}
```

3 referencias

Quants empleats hi ha?

```
foreach (ClsEmpleado empleado in Listaempleado)
{
    Console.WriteLine("{0} Tiene el sueldo:{1}", i, empleado.DameSueldo());
    i++;
}
ClsEmpleado.NumEmpleados();
Console.ReadKey();
```

Quin objecte és?

```
if (empleado is ClsJefe) {  
    Console.WriteLine("El empleado {0} tiene el sueldo :{1}",  
        empleado.Nombre, empleado.DameSueldo());  
    //polimorfismo  
    i++;  
}
```