

# computinglegan.com.

## OOP con Java.

### Introducción a la herencia con Java.

### Actualizado.

Martes 23 de diciembre de 2025.

### Descripción.

La Programación Orientada a Objetos (POO), u Oriented Programation Oriented Object (OOP) es una fase que todavía existió hoy en día y que se hace tanto para el desarrollo del software e incluso para cualquier otro ámbito de la vida que no tiene una solución clara, ese método se hace en forma de fases, en orden, y cada fase en el momento adecuado y hay que hacer cada fase en el momento adecuado y con el máximo número de personas, no únicamente con personas de conocimientos en el área de Ingeniería Informática, sino, con el mayor número de personas posibles en diferentes áreas de conocimientos y no únicamente de niveles de conocimiento de máxima cualificación, sino, en la medida que sea posible, con el máximo número de personas de diferentes niveles de conocimiento, es decir, tanto, con personas de altísima cualificación, alta cualificación, media cualificación, baja cualificación y bajísima cualificación para que tenga el máximo éxito posible lo que se intentó hacer no sólo en el desarrollo del software, sino en cualquier otro ámbito de la vida.

A continuación, voy a publicar una pequeña introducción en el área del desarrollo del software de la herencia de OOP, en el apartado de referencias, hay información mucho más detallada de la que yo no puedo proporcionar.

Si poder puedo, refinaré algo más esto, pero en primer lugar intentando aportando algo a los demás, pero eso sí, si veo que más o menos ya he aportado algo a los demás, intentaré hacer algo proporcional hacia mí, eso se llama ALTRUISMO.

### Ejemplo.

#### SerVivoConNombre.java

```
public class SerVivoConNombre {
    private String nombre;
    protected int fechaNacimiento;

    public int getEdad() {
        //Para obtener el año se pueden utilizar formas más complejas como por ejemplo obtener la fecha actual del sistema.
        int edad = 2025 - this.fechaNacimiento;
        return(edad);
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getNombre() {
        return(this.nombre);
    }
}
```

#### Persona.java

```
public class Persona extends SerVivoConNombre {
    private String nombreDePila;

    public void setNombreDePila(String nombreDePila) {
        this.nombreDePila = nombreDePila;
    }

    public String getNombreDePila() {
        return(this.nombreDePila);
    }
}
```

```
    public int getNumeroExtremidadesInferiores() {  
        return(2);  
    }  
}
```

## Mascota.java

```
public class Mascota extends SerVivoConNombre {  
    private int chipIdentificacion;  
  
    public void setChipIdentificacion(int chipIdentificacion) {  
        this.chipIdentificacion = chipIdentificacion;  
    }  
  
    public int getChipIdentificacion() {  
        return(chipIdentificacion);  
    }  
}
```

## Gato.java

```
public class Gato extends Mascota {  
  
    public int getExtremidadesInferiores() {  
        return(4);  
    }  
}
```

## Pajaro.java

```
public class Pajaro {  
  
    public int getExtremidadesInferiores() {  
        return(2);  
    }  
}
```

## Main.java

```
public class Main {  
  
    public static void main(String[] args) {  
        //estó lo dejo libré para qué quien pueda lo rellene.  
    }  
}
```

## Material de apoyo.

Descarga del código fuente en el archivo. [code.zip](#).

Descarga de este articulo en pdf. [article.pdf](#).

## Referencias.

[Utilizar otra clase.](#)

[Página Web que explica algo sobre la herencia en OOP.](#)

[Definición de la Programación Orientada a Objetos \(POO\) u Oriented Programation Objects \(OOP\) según la wikipedia, la enciclopedia libre.](#)

## Contacto.

legan76@hotmail.com.