

## PROGRAMACIÓN II

### Trabajo Práctico 7 – Herencia y Polimorfismo

**Alumno:** Chiavón, Facundo

**Github:** <https://github.com/Farvon/UTN-TUPaD-P2.git>

#### Caso Práctico 1

Implementación de la herencia y polimorfismo en clases de Java.

##### Clase main

```
1 package tp7.herencia.y.polimorfismo;
2
3 import java.util.ArrayList;
4
5 import vehiculos.*;
6 import figurasGeometricas.*;
7 import empleados.*;
8 import Animales.*;
9
10 public class TP7HerenciaYPolimorfismo {
11
12     public static void main(String[] args) {
13
14         System.out.println("----- Vehiculos y herencia -----");
15         Vehiculo al = new Auto("Dodge", "1500", 4);
16         al.mostrarInfo();
17
18         System.out.println("----- Figuras Geométricas -----");
19         ArrayList<Figura> figuras = new ArrayList<>();
20         figuras.add(new Rectangulo("Rectangulo 1", 2, 5));
21         figuras.add(new Circulo("Circulo 1", 5));
22         figuras.add(new Rectangulo("Rectangulo 2", 3, 9));
23         figuras.add(new Circulo("Circulo 2", 8));
24
25         for (Figura figura : figuras) {
26             figura.mostrarInfo();
27             figura.calcularArea();
28         }
29
30         System.out.println("----- Empleados -----");
31         ArrayList<Empleado> empleados = new ArrayList<>();
32
33         empleados.add(new EmpleadoTemporal());
34         empleados.add(new EmpleadoTemporal());
35         empleados.add(new EmpleadoPlanta());
36         empleados.add(new EmpleadoTemporal());
37         empleados.add(new EmpleadoPlanta());
38
39         for (Empleado empleado : empleados) {
40             if (empleado instanceof EmpleadoTemporal) {
41                 empleado.calcularSueldo(300);
42             } else {
43                 empleado.calcularSueldo(500);
44             }
45         }
46
47         System.out.println("----- Animales -----");
48         ArrayList<Animal> animales = new ArrayList<>();
49
50         animales.add(new Perro());
51         animales.add(new Gato());
52         animales.add(new Vaca());
53
54         for (Animal animal : animales) {
55             animal.describirAnimal();
56             animal.hacerSonido();
57         }
58
59     }
60
61 }
```

## 1. Vehículos y herencia básica

### Clase Vehículo

```
package vehiculos;

public class Vehiculo {

    String marca;
    String modelo;

    public Vehiculo(String marca, String modelo) {
        this.marca = marca;
        this.modelo = modelo;
    }

    public void mostrarInfo() {
    }

    @Override
    public String toString() {
        return "Vehiculo{" + "marca=" + marca + ", modelo=" + modelo + '}';
    }

}
```

### Clase Auto

```
package vehiculos;

public class Auto extends Vehiculo {

    int cantidadPuertas;

    public Auto(String marca, String modelo, int cantidadPuertas) {
        super(marca, modelo);
        this.cantidadPuertas = cantidadPuertas;
    }

    @Override
    public void mostrarInfo() {
        super.mostrarInfo();
        System.out.println(this);
    }

    @Override
    public String toString() {
        return super.toString() + ", Cantidad de Puertas: " + cantidadPuertas + "]";
    }

}
```

### Resultado

```
Vehiculo{marca=Dodge, modelo=1500}, Cantidad de Puertas: 4]
```

## 2. Figuras geométricas y métodos abstractos

### Clase Figura

```
package figurasGeometricas;

public abstract class Figura {

    private String nombre;

    public Figura(String nombre) {
        this.nombre = nombre;
    }

    public abstract void calcularArea();

    public void mostrarInfo() {
    }

    @Override
    public String toString() {
        return "Figura{" + "nombre=" + nombre + ", ";
    }

}
```

### Clase Rectangulo

```
package figurasGeometricas;

public class Rectangulo extends Figura {

    private double lado1;
    private double lado2;

    public Rectangulo(String nombre, double lado1, double lado2) {
        super(nombre);
        this.lado1 = lado1;
        this.lado2 = lado2;
    }

    @Override
    public void calcularArea() {
        double area = 2 * lado1 + 2 * lado2;
        System.out.println("El área es: " + area);
    }

    @Override
    public void mostrarInfo() {
        System.out.println(this);
    }

    @Override
    public String toString() {
        return super.toString() + "lado1=" + lado1 + ", lado2=" + lado2 + '}';
    }

}
```

## Clase Circulo

```
package figurasGeometricas;

public class Circulo extends Figura {

    private double radio;

    public Circulo(String nombre, double radio) {
        super(nombre);
        this.radio = radio;
    }

    @Override
    public void calcularArea() {
        double area = 2 * Math.PI * radio;
        System.out.println("El área es: " + area);
    }

    @Override
    public void mostrarInfo() {
        System.out.println(this);
    }

    @Override
    public String toString() {
        return super.toString() + "radio=" + radio + ' ';
    }

}
```

## Resultado:

```
Figura{nombre=Rectangulo 1, lado1=2.0, lado2=5.0}
El área es: 14.0
Figura{nombre=Circulo 1, radio=5.0}
El área es: 157.07963267948966
Figura{nombre=Rectangulo 2, lado1=3.0, lado2=9.0}
El área es: 27.0
Figura{nombre=Circulo 2, radio=8.0}
El área es: 402.1238596599194
```

### 3. Empleados y polimorfismo

#### Clase Empleado

```
package empleados;

public abstract class Empleado {

    public abstract void calcularSueldo(double valorHora);
}
```

#### Clase EmpleadoPlanta

```
package empleados;

public class EmpleadoPlanta extends Empleado {

    @Override
    public void calcularSueldo(double valorHora) {
        double sueldo = valorHora * 48;
        System.out.println("El sueldo es: " + sueldo);
    }
}
```

#### Clase EmpleadoTemporal

```
package empleados;

public class EmpleadoTemporal extends Empleado {

    @Override
    public void calcularSueldo(double valorHora) {
        double sueldo = valorHora * 48;
        System.out.println("El sueldo es: " + sueldo);
    }
}
```

#### Resultado

```
El sueldo es: 14400.0
El sueldo es: 14400.0
El sueldo es: 24000.0
El sueldo es: 14400.0
El sueldo es: 24000.0
```

#### 4. Animales y comportamiento sobrescrito

##### Clase Animal

```
package Animales;

public abstract class Animal {

    public abstract void hacerSonido();

    public abstract void describirAnimal();

}
```

##### Clase Perro

```
package Animales;

public class Perro extends Animal {

    @Override
    public void describirAnimal() {
        System.out.println("Soy un perro!");
    }

    @Override
    public void hacerSonido() {
        System.out.println("Guau Guau!!...");
    }

}
```

##### Clase Gato

```
package Animales;

public class Gato extends Animal {

    @Override
    public void describirAnimal() {
        System.out.println("Soy un gato!");
    }

    @Override
    public void hacerSonido() {
        System.out.println("Miau Miau!!...");
    }

}
```

### Clase Vaca

```
package Animales;

public class Vaca extends Animal {

    @Override
    public void describirAnimal() {
        System.out.println("Soy una vaca!");
    }

    @Override
    public void hacerSonido() {
        System.out.println("Muuuuu!!...");
    }

}
```

### Resultado

```
Soy un perro!
Guau Guau!!...
Soy un gato!
Miau Miau!!...
Soy una vaca!
Muuuuu!!...
```