

HERENCIA

Creamos la clase Persona

```
public class Persona {  
    private String nombre;  
    private String apellido;  
    private String dni;  
  
    public Persona(String nombre, String apellido, String dni) {  
        super();  
        this.nombre = nombre;  
        this.apellido = apellido;  
        this.dni = dni;  
    }  
  
    @Override  
    public String toString() {  
        return this.nombre + " - " + this.apellido + " - " + this.dni;  
    }  
}
```

Ahora creamos las clases derivada Jefe y Empleado

```
public class Jefe extends Persona {  
    private String cargo ;  
    public Jefe(String nombre, String apellido, String dni,String cargo) {  
        super(nombre, apellido, dni);  
        this.cargo = cargo;  
    }  
  
    @Override  
    public String toString() {  
        return "Jefe" + super.toString() +  
            " Cargo " + this.cargo;  
    }  
}
```

```
public class Empleado extends Persona {  
    private float sueldo;  
    public Empleado(String nombre, String apellido, String dni, float sueldo) {  
        super(nombre, apellido, dni);  
        this.sueldo = sueldo;  
    }  
    @Override  
    public String toString() {  
        return "Empleado" + super.toString() +  
            "Sueldo " + this.sueldo + "];"  
    }  
}
```

Java no admite la herencia múltiple pero en su lugar podemos utilizar las Interfaces.

SOBRECARGA Y SOBRESCRITURA

Para el mismo método realizamos dos implementaciones distintas para el método paga

```
public void paga(int cuanto) {
    this.sueldo = this.sueldo + cuanto;
}

public void paga() {
    this.sueldo = this.sueldo - 50;
}

public class Principal {

    public static void main(String[] args) {
        Empleado miEmpleado = new Empleado("Juan", "Perez", "499998635W", 500);
        System.out.println(miEmpleado);
        miEmpleado.paga(); //Resta 50 euros al salario
        miEmpleado.paga(500); //Abona 500 euros en el salario
        System.out.println(miEmpleado);
    }
}
```

Resultado :

```
Empleado: Juan - Perez - 499998635W, Sueldo: 500.0€
Empleado: Juan - Perez - 499998635W, Sueldo: 950.0€
```

Sobreescritura

Implementamos el método trabajar en la clase Persona y en la clase empleado

Clase Persona :

```
public void trabajar() {
    System.out.println("Esta persona trabaja");
}
```

Clase Empleado:

```
@Override
public void trabajar() {
    System.out.println("El empleado trabaja");
}
```

Es el mismo método pero cambia el mensaje.

COLECCIONES

ArrayList:

```
import java.util.ArrayList;
public class Principal {

    public static void main(String[] args) {
        ArrayList<String> color=new ArrayList<String>();
        color.add("Verde");
        color.add("Azul");
        color.add("Rojo");
        System.out.println("ArrayList de colores: "+color);

    }
}
```

Obtenemos lo siguiente :

```
<terminated> Principal (26) [Java Application] C:\Program Files\Java\jdk-14.0.1\bin
ArrayList de colores: [Verde, Azul, Rojo]
```

Cola :

```
import java.util.LinkedList;
public class fifo {
    public static void main(String[] args) {
        Queue<String> queue = new LinkedList<String>();
        queue.add("rojo");
        queue.add("amarillo");
        queue.add("azul");
        System.out.println("Colores" + queue);
        String value = (String) queue.remove();
        System.out.println("El elemento eliminado es: " + value);

    }
}
```

```
<terminated> fifo (Application) C:\Program Files\Java\jdk-14.0.1\bin
Colores[rojo, amarillo, azul]
El elemento eliminado es: rojo
```

Pila:

```
import java.util.Stack;
public class pila {

    public static void main(String[] args) {
        Stack<String> stack = new Stack<String>();
        stack.push("gris");
        stack.push("negro");
        stack.push("blanco");
        System.out.println("Pila de colores:"+stack);
        System.out.println("El elemento eliminado es "+stack.pop());

    }
}
```

```
Pila de colores:[gris, negro, blanco]  
El elemento eliminado es blanco
```

Una cola es una estructura lineal, que sigue un orden en el que se ejecutan las operaciones. Para ver la diferencia entre pila y cola , vemos que el elemento eliminado de la pila ha sido el último , mientras que en la cola el elemento eliminado ha sido el primero.