TP 7 – Herencia y Polimorfismo

Objetivo:

Comprender y aplicar los conceptos de herencia y polimorfismo en la Programación Orientada a Objetos, reconociendo su importancia para la reutilización de código, la creación de jerarquías de clases y el diseño flexible de soluciones en Java.

Caso práctico:

Desarrollar las siguientes Katas en Java aplicando herencia y polimorfismo. Se recomienda repetir cada kata para afianzar el concepto:

1. Vehículos y herencia básica
    - Clase base: Vehículo con atributos marca, modelo y método mostrarInfo()
    - Subclase: Auto con atributo adicional cantidadPuertas, sobrescribe mostrarInfo()
    - Tarea: Instanciar un auto y mostrar su información completa.
2. Figuras geométricas y métodos abstractos
    - Clase abstracta: Figura con método calcularArea() y atributo nombre
    - Subclases: Círculo y Rectángulo implementan el cálculo del área
    - Tarea: Crear un array de figuras y mostrar el área de cada una usando polimorfismo.
3. Empleados y polimorfismo
    - Clase abstracta: Empleado con método calcularSueldo()
    - Subclases: EmpleadoPlanta, EmpleadoTemporal
    - Tarea: Crear lista de empleados, invocar calcularSueldo() polimórficamente, usar instanceof para clasificar
4. Animales y comportamiento sobrescrito
    - Clase: Animal con método hacerSonido() y describirAnimal()
    - Subclases: Perro, Gato, Vaca sobrescriben hacerSonido() con @Override
    - Tarea: Crear lista de animales y mostrar sus sonidos con polimorfismo

CONCLUSIONES ESPERADAS

- Comprender el mecanismo de herencia y sus beneficios para la reutilización de código.
- Aplicar polimorfismo para lograr flexibilidad en el diseño de programas.

- Inicializar objetos correctamente usando super en constructores.
- Controlar el acceso a atributos y métodos con modificadores adecuados.
- Identificar y aplicar upcasting, downcasting y instanceof correctamente.
- Utilizar clases y métodos abstractos como base de jerarquías lógicas.
- Aplicar principios de diseño orientado a objetos en la implementación en Java.

1)

```java
package tp7;

public class TP7 {

    public static void main(String[] args) {
        abstract class Vehiculo {
            protected String marca;
            protected String modelo;

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

            public void mostrarInfo() {
                System.out.println("Marca: " + marca + ", Modelo: " + modelo);
            }
        }

        class Auto extends Vehiculo {
            private final 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("Puertas: " + cantidadPuertas);
            }
        }

        Auto auto = new Auto("Toyota", "Corolla", 4);
        auto.mostrarInfo();
    }
}
```

```
out - TP7 (run) ×
run:
Marca: Toyota, Modelo: Corolla
Puertas: 4
BUILD SUCCESSFUL (total time: 0 seconds)
```

2)

```java
public class Ejer2 {
    public static void main(String[] args) {
        System.setOut(new PrintStream(System.out, true, StandardCharsets.UTF_8));
        abstract class Figura {
            protected String nombre;

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

            public abstract double calcularArea();

            public void mostrarArea() {
                System.out.println(nombre + " - Área: " + calcularArea());
            }
        }

        class Circulo extends Figura {
            private double radio;

            public Circulo(double radio) {
                super("Círculo");
                this.radio = radio;
            }

            @Override
            public double calcularArea() {
                return Math.PI * Math.pow(radio, 2);
            }
        }

        class Rectangulo extends Figura {
            private double base;
            private double altura;

            public Rectangulo(double base, double altura) {
                super("Rectángulo");
                this.base = base;
                this.altura = altura;
            }

            @Override
            public double calcularArea() {
                return base * altura;
            }
        }

        Figura[] figuras = {
            new Circulo(3),
            new Rectangulo(4, 5)
        };

        for (Figura f : figuras) {
            f.mostrarArea();
        }
    }
}
```

```
tp7.Ejer2 > main >
ut - TP7 (run) ×
run:
Círculo - Área: 28.274333882308138
Rectángulo - Área: 20.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

3)

```java
public class Ejer3 {
    public static void main(String[] args) {
        System.setOut(new PrintStream(System.out, true, StandardCharsets.UTF_8));
        abstract class Empleado {
            protected String nombre;

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

            public abstract double calcularSueldo();
        }

        class EmpleadoPlanta extends Empleado {
            private final double salarioBase;

            public EmpleadoPlanta(String nombre, double salarioBase) {
                super(nombre);
                this.salarioBase = salarioBase;
            }

            @Override
            public double calcularSueldo() {
                return salarioBase;
            }
        }

        class EmpleadoTemporal extends Empleado {
            private final int diasTrabajados;
            private final double pagoPorDia;

            public EmpleadoTemporal(String nombre, int diasTrabajados, double pagoPorDia) {
                super(nombre);
                this.diasTrabajados = diasTrabajados;
                this.pagoPorDia = pagoPorDia;
            }

            @Override
            public double calcularSueldo() {
                return diasTrabajados * pagoPorDia;
            }
        }

        Empleado[] empleados = {
            new EmpleadoPlanta("Carlos", 150000),
            new EmpleadoTemporal("Ana", 20, 5000)
        };

        for (Empleado e : empleados) {
            System.out.println(e.nombre + " - Sueldo: $" + e.calcularSueldo());

            if (e instanceof EmpleadoPlanta) {
                System.out.println("Es un empleado de planta \n");
            } else if (e instanceof EmpleadoTemporal) {
                System.out.println("Es un empleado temporal \n");
            }
        }
```

```
7.Ejer3 >
- TP7 (run) ×
Carlos - Sueldo: $150000.0
Es un empleado de planta

Ana - Sueldo: $100000.0
Es un empleado temporal

BUILD SUCCESSFUL (total time: 0 seconds)
```

4)

```java
package tp7;

import java.io.PrintStream;
import java.nio.charset.StandardCharsets;

public class Ejer4 {
    public static void main(String[] args) {
        System.setOut(new PrintStream(System.out, true, StandardCharsets.UTF_8));
        class Animal {
            public void hacerSonido() {
                System.out.println("El animal hace un sonido genérico.");
            }

            public void describirAnimal() {
                System.out.println("Soy un animal.");
            }
        }

        class Perro extends Animal {
            @Override
            public void hacerSonido() {
                System.out.println("Guau guau!");
            }
        }

        class Gato extends Animal {
            @Override
            public void hacerSonido() {
                System.out.println("Miau miau!");
            }
        }

        class Vaca extends Animal {
            @Override
            public void hacerSonido() {
                System.out.println("Muuu!");
            }
        }

        Animal[] animales = {
            new Perro(),
            new Gato(),
            new Vaca()
        };

        for (Animal a : animales) {
            a.describirAnimal();
            a.hacerSonido();
            System.out.println();
        }
    }
}
```

```
- TP7 (run) ×
run:
Soy un animal.
Guau guau!

Soy un animal.
Miau miau!

Soy un animal.
Muuu!

BUILD SUCCESSFUL (total time: 0 seconds)
```