PROGRAMACIÓN II

TRABAJO PRÁCTICO Nº7: HERENCIA Y POLIMORFISMO

CHRISTIAN EMMANUEL OLIVERO

https://github.com/CHRISTIAN2320/UTN-TUPAD-Programacion2

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()

```java
public class Vehiculo {
    private String marca;
    private String modelo;

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

    public String getMarca() {
        return marca;
    }

    public String getModelo() {
        return modelo;
    }

    public void mostrarInfo(){
        System.out.println("La marca del auto es: " + getMarca()
        + "\nel modelo del auto es: " + getModelo());
    }
```

- Subclase: Auto con atributo adicional cantidadPuertas, sobrescribe mostrarInfo()

```java
public class Auto extends Vehiculo{
    private int cantidadPuertas ;

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

    public int getCantidadPuertas() {
        return cantidadPuertas;
    }

    @Override
    public void mostrarInfo(){
        super.mostrarInfo();
        System.out.println("El auto Tiene "+ getCantidadPuertas() + " puertas");
    }
```

- Tarea: Instanciar un auto y mostrar su información completa.

```java
public static void main(String[] args) {

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

```
La marca del auto es: Toyota
el modelo del auto es: Corolla
El auto Tiene 4 puertas
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Figuras geométricas y métodos abstractos

- Clase abstracta: Figura con método calcularArea() y atributo nombre

```java
public abstract class Figura {

    private String nombre;

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

    public abstract void calcularArea();
```

- Subclases: Círculo y Rectángulo implementan el cálculo del área

```java
public class Circulo extends Figura {

    private double radio;

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

    @Override
    public void calcularArea() {
        System.out.println("El area del circulo es: "
                + Math.PI * radio * radio);
    }
```

```java
public class Rectangulo extends Figura {

    private int base;
    private int altura;

    public Rectangulo(int base, int altura, String nombre) {
        super(nombre);
        this.base = base;
        this.altura = altura;
    }

    @Override
    public void calcularArea() {
        System.out.println("El area del rectangulo es: "
                + base*altura);
    }
```

- Tarea: Crear un array de figuras y mostrar el área de cada una usando polimorfismo.

```java
public static void main(String[] args) {
    ArrayList<Figura> figuras = new ArrayList <>();

    figuras.add(new Circulo(5, "circulo1"));
    figuras.add(new Rectangulo(2, 5, "rectangulo"));

    for (Figura figura : figuras) {
        figura.calcularArea();
    }
```

```
run:
El area del circulo es: 78.53981633974483
El area del rectangulo es: 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Empleados y polimorfismo

- Clase abstracta: Empleado con método calcularSueldo()

```java
public abstract class Empleado {
    private String nombre;

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

    public String getNombre() {
        return nombre;
    }

    public abstract void calcularSueldo ();
```

- Subclases: EmpleadoPlanta, EmpleadoTemporal

```java
public class EmpleadoPlanta extends Empleado{

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

    public double getSueldoFijo() {
        return sueldoFijo;
    }

    @Override
    public void calcularSueldo() {
        System.out.println("El sueldo de "+ getNombre() + " es de: $"
        +getSueldoFijo() );
        }
```

```java
public class EmpleadoTemporal extends Empleado{
    private double horasTrabajadas ;
    private double valorHora;

    public EmpleadoTemporal(String nombre, double horasTrabajadas, double valorHora) {
        super(nombre);
        this.horasTrabajadas= horasTrabajadas;
        this.valorHora=valorHora;
    }

    public double getHorasTrabajadas() {
        return horasTrabajadas;
    }

    public double getValorHora() {
        return valorHora;
    }

    @Override
    public void calcularSueldo() {
        System.out.println("El sueldo de "+ getNombre() + " es de: $"
        + getValorHora()*getHorasTrabajadas());
     }
```

- Tarea: Crear lista de empleados, invocar calcularSueldo() polimórficamente, usar instanceof para clasificar

```java
public class Ejercicio3 {

    public static void main(String[] args) {
        ArrayList<Empleado> empleados = new ArrayList<>();

        empleados.add(new EmpleadoPlanta("Carlos", 250000));
        empleados.add(new EmpleadoTemporal("Juan", 8, 25000));

        for (Empleado empleado : empleados) {
            empleado.calcularSueldo();
            if (empleado instanceof EmpleadoPlanta) {
                System.out.println("Tipo: Planta");
            } else if (empleado instanceof EmpleadoTemporal) {
                System.out.println("Tipo: Temporal");
            }
```

```
run:
El sueldo de Carlos es de: $250000.0
Tipo: Planta
El sueldo de Juan es de: $200000.0
Tipo: Temporal
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Animales y comportamiento sobrescrito

- Clase: Animal con método hacerSonido() y describirAnimal()

```java
public abstract class Animal {

    private String nombre;
    private int edad;

    public Animal(String nombre, int edad) {
        this.nombre = nombre;
        this.edad = edad;
    }

    public String getNombre() {
        return nombre;
    }

    public int getEdad() {
        return edad;
    }

    public abstract void hacerSonido();

    public void describirAnimal() {
        System.out.println("el nombre del animal es "
                + getNombre() + " y tiene " + getEdad() + " anios");
    }
```

- Subclases: Perro, Gato, Vaca sobrescriben hacerSonido() con @Override

```java
public class Perro extends Animal{

    public Perro(String nombre, int edad) {
        super(nombre, edad);
    }

    @Override
    public void hacerSonido() {
        System.out.println("El perro hace guau guau");

    }


}
```

```java
public class Gato extends Animal{

    public Gato(String nombre, int edad) {
        super(nombre, edad);
    }

    @Override
    public void hacerSonido() {
        System.out.println("El gato hace miau miau");
        }
```

```java
public class Vaca extends Animal{

    public Vaca(String nombre, int edad) {
        super(nombre, edad);
    }

    @Override
    public void hacerSonido() {
        System.out.println("La vaca hace muu muu");
        }
}
```

- Tarea: Crear lista de animales y mostrar sus sonidos con polimorfismo

```java
public class Ejercicio4 {

    public static void main(String[] args) {
        ArrayList<Animal> animales = new ArrayList<>();
        animales.add(new Perro("Firulays", 3));
        animales.add(new Gato("Michi", 1));
        animales.add(new Vaca("Lola", 2));

        for (Animal animal : animales) {
            animal.describirAnimal();
            animal.hacerSonido();
            System.out.println("----------------------");
        }

    }

}
```

```
run:
el nombre del animal es Firulays y tiene 3 anios
El perro hace guau guau
-----------------------
el nombre del animal es Michi y tiene 1 anios
El gato hace miau miau
-----------------------
el nombre del animal es Lola y tiene 2 anios
La vaca hace muu muu
-----------------------
BUILD SUCCESSFUL (total time: 0 seconds)
```