PROGRAMACIÓN II

Trabajo Práctico 7: Herencia y Polimorfismo en Java

Alumno: Gonzalo Barrios.

GitHub: https://github.com/Goonza88/Programacion-II

1. Vehículos y herencia básica:

Clase Vehículo:

```java
/**...4 lines */
public class Vehiculo {
    private String marca;
    private String modelo;

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

    public void mostrarInfo() {
        System.out.println("Vehiculo: \nMarca: " + marca + ".\nModelo: " + modelo + '.');
    }
}
```

Clase Auto:

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

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

    @Override
    public void mostrarInfo() {
        super.mostrarInfo();
        System.out.println("Cantidad de puertas: " + cantidadPuertas + ".");
    }
}
```

Main y Resultados:

```java
public class HerenciaYPolimorfismo {
    public static void main(String[] args) {
        Vehiculo a1 = new Vehiculo("Ford", "Fiesta");
        Vehiculo a2 = new Auto(4,"Ford", "Fiesta");

        a1.mostrarInfo();
        a2.mostrarInfo();
    }
}
```

```
Output ×
Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Prog
run:
Vehiculo:
Marca: Ford.
Modelo: Fiesta.

Vehiculo:
Marca: Ford.
Modelo: Fiesta.
Cantidad de puertas: 4.
```

2. Figuras geométricas y métodos abstractos:

Clase Figura:

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

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

    public abstract double calcularArea();
}
```

Clase Circulo:

```java
public class Circulo extends Figura {
    private double radio;

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

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

Clase Rectángulo:

```java
public class Rectangulo extends Figura {
    private double base;
    private double altura;

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

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

Main y Resultados:

```
public class HerenciaYPolimorfismo2 {
    public static void main(String[] args) {
        ArrayList <Figura> figuras = new ArrayList<>();

        figuras.add(new Circulo(2, "Primer Circulo"));
        figuras.add(new Rectangulo(3, 4, "Primer Rectangulo"));
        figuras.add(new Circulo(5, "Segundo Circulo"));
        figuras.add(new Rectangulo(6, 7, "Segundo Rectangulo"));

        for (Figura f : figuras) {
            System.out.println(f.calcularArea());
        }
    }
```

```
Output ×
Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II ×
run:
12.566370614359172
12.0
78.53981633974483
42.0
```

3. Empleados y polimorfismo:

Clase Empleado:

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

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

    public String getNombre() {
        return nombre;
    }

    public abstract double calcularSueldo();
}
```

Clase EmpleadoPlanta:

```java
public class EmpleadoPlanta extends Empleado {
    private final double sueldo = 250.00;

    public EmpleadoPlanta(String nombre) {
        super(nombre);
    }

    @Override
    public double calcularSueldo() {
        return sueldo;
    }
}
```

Clase EmpleadoTemporal:

```java
public class EmpleadoTemporal extends Empleado {
    private final double sueldo = 150.00;

    public EmpleadoTemporal(String nombre) {
        super(nombre);
    }

    @Override
    public double calcularSueldo() {
        return sueldo;
    }
}
```

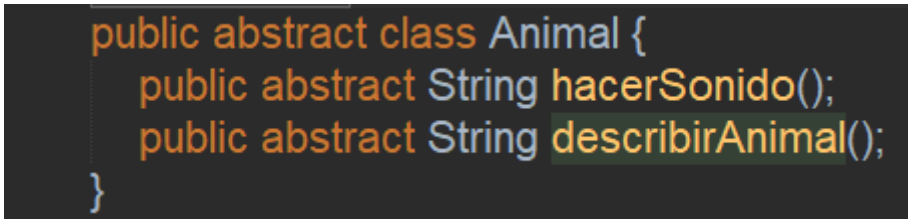
Main y Resultados:

```java
public class HerenciaYPolimorfismo3 {
    public static void main(String[] args) {
        ArrayList <Empleado> E = new ArrayList<>();

        E.add(new EmpleadoPlanta("Juan"));
        E.add(new EmpleadoPlanta("Lucia"));
        E.add(new EmpleadoTemporal("Jose"));
        E.add(new EmpleadoTemporal("Maria"));

        for (Empleado e : E) {
            if (e instanceof EmpleadoPlanta) {
                System.out.println("Empleado de Planta\nNombre: "  + e.getNombre() +
                        ".\nSueldo: " + e.calcularSueldo() + ".\n");
            } else {
                System.out.println("Empleado Temporal\nNombre: "  + e.getNombre() +
                        ".\nSueldo: " + e.calcularSueldo() + ".\n");
            }
```

```
tput ×
Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II ×   HerenciaYPolimorfismo (run) ×
run:
Empleado de Planta
Nombre: Juan.
Sueldo: 250.0.

Empleado de Planta
Nombre: Lucia.
Sueldo: 250.0.

Empleado Temporal
Nombre: Jose.
Sueldo: 150.0.

Empleado Temporal
Nombre: Maria.
Sueldo: 150.0.
```

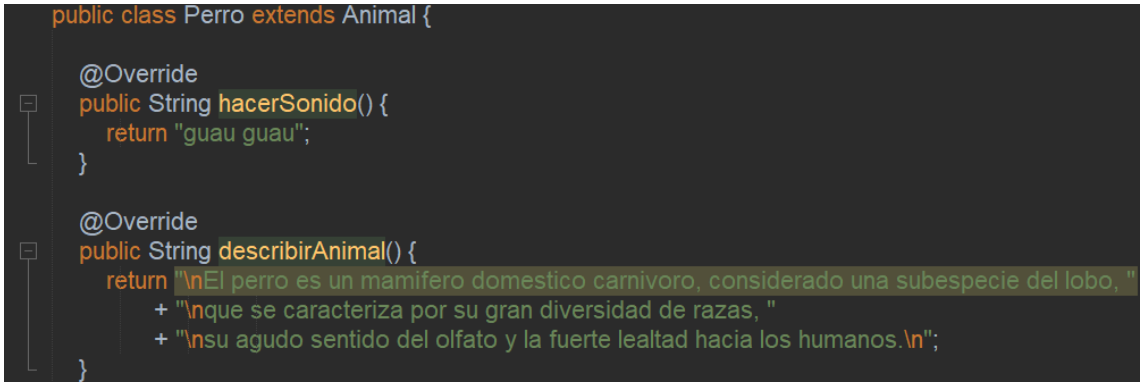
4. Animales y comportamiento sobrescrito:

Clase Animal:



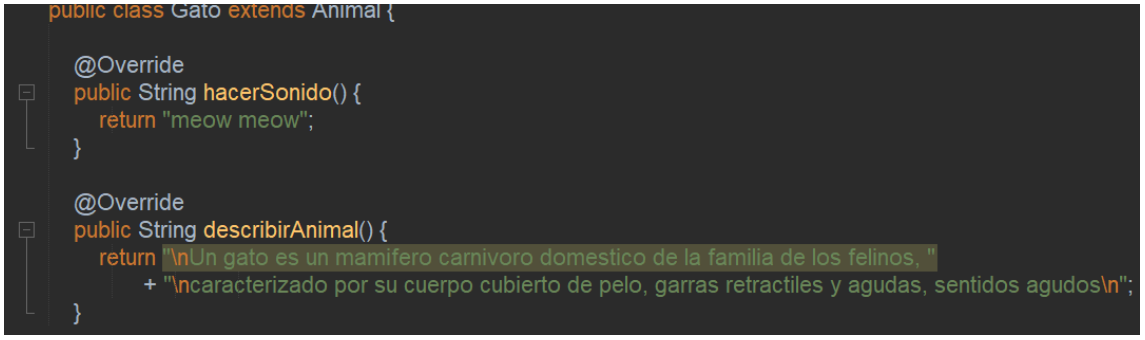
```
public abstract class Animal {
    public abstract String hacerSonido();
    public abstract String describirAnimal();
}
```

Clase Perro:



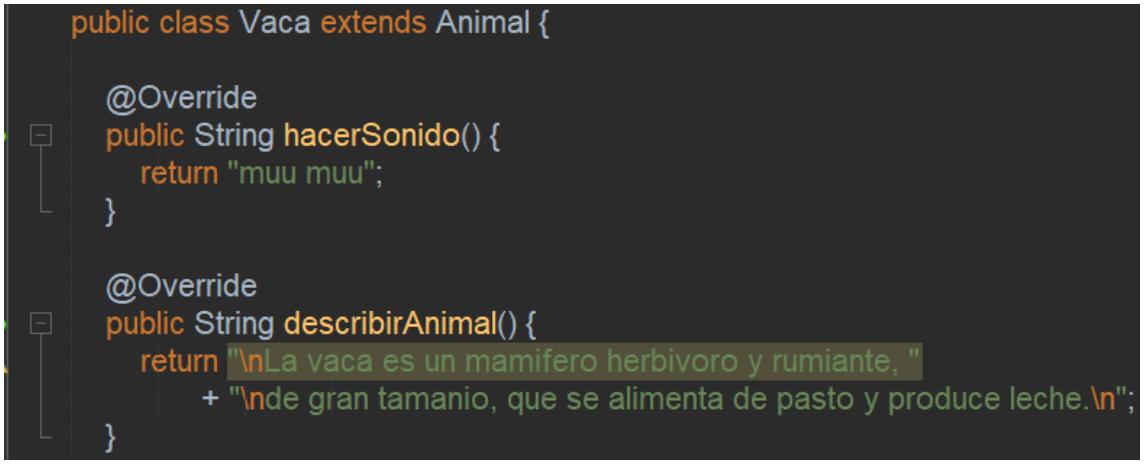
```
public class Perro extends Animal {

    @Override
    public String hacerSonido() {
        return "guau guau";
    }

    @Override
    public String describirAnimal() {
        return "\nEl perro es un mamifero domestico carnivoro, considerado una subespecie del lobo, "
                + "\nque se caracteriza por su gran diversidad de razas, "
                + "\nsu agudo sentido del olfato y la fuerte lealtad hacia los humanos.\n";
    }
```

Clase Gato:



```
public class Gato extends Animal {

    @Override
    public String hacerSonido() {
        return "meow meow";
    }

    @Override
    public String describirAnimal() {
        return "\nUn gato es un mamifero carnivoro domestico de la familia de los felinos, "
                + "\ncaracterizado por su cuerpo cubierto de pelo, garras retractiles y agudas, sentidos agudos\n";
    }
```

Clase Vaca:



```
public class Vaca extends Animal {

    @Override
    public String hacerSonido() {
        return "muu muu";
    }

    @Override
    public String describirAnimal() {
        return "\nLa vaca es un mamifero herbivoro y rumiante, "
                + "\nde gran tamanio, que se alimenta de pasto y produce leche.\n";
    }
```

Main y Resultados:

```java
public class HerenciaYPolimorfismo4 {
    public static void main(String[] args) {
        ArrayList <Animal> A = new ArrayList<>();

        Animal  P = new Perro();
        Animal  G = new Gato();
        Animal  V = new Vaca();

        A.add(P);
        A.add(G);
        A.add(V);

        for (Animal a : A) {
            if (a instanceof Perro) {
                System.out.println("El perro hace:\n" + a.hacerSonido() + "\n" + a.describirAnimal());
            } else if (a instanceof Gato) {
                System.out.println("El gato hace:\n" + a.hacerSonido() + "\n" + a.describirAnimal());
            } else {
                System.out.println("La vaca hace:\n" + a.hacerSonido() + "\n" + a.describirAnimal());
```

```
tput ×
Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II ×   HerenciaYPolimorfismo (run) ×
run:
El perro hace:
guau guau

El perro es un mamifero domestico carnivoro, considerado una subespecie del lobo,
que se caracteriza por su gran diversidad de razas,
su agudo sentido del olfato y la fuerte lealtad hacia los humanos.

El gato hace:
meow meow

Un gato es un mamifero carnivoro domestico de la familia de los felinos,
caracterizado por su cuerpo cubierto de pelo, garras retractiles y agudas, sentidos agudos

La vaca hace:
muu muu

La vaca es un mamifero herbivoro y rumiante,
de gran tamanio, que se alimenta de pasto y produce leche.
```

5. Sistema de pagos con polimorfismo y genéricos:

Interfaz Pagable:

```java
public interface Pagable {
    void pagar();
}
```

Clase Transferencia:

```java
public class Transferencia implements Pagable {
    @Override
    public void pagar() {
        System.out.println("Pagando con Transferencia bancaria...");
    }
}
```

Clase TarjetaCredito:

```java
public class TarjetaCredito implements Pagable {
    @Override
    public void pagar() {
        System.out.println("Pagando con Tarjeta de Credito...");
    }
}
```

Clase Efectivo:

```java
public class Efectivo implements Pagable {
    @Override
    public void pagar() {
        System.out.println("Pagando en Efectivo...");
    }
}
```

Main y Resultados:

```java
public class HerenciaYPolimorfismo5 {
    public static void main(String[] args) {
        Pagable Tarjeta = new TarjetaCredito();
        Pagable Transferencia = new Transferencia();
        Pagable Efectivo = new Efectivo();

        procesarPago(Tarjeta);
        procesarPago(Transferencia);
        procesarPago(Efectivo);
    }

    public static void procesarPago(Pagable medio) {
        medio.pagar();
        System.out.println("Pago procesado con exito.\n");
    }
```

```
Output ×
Programacion II - C:\Users\Gonza\Documents\Estudio\2 - Segundo Cuatrimestre\Repositorios\Programacion II
run:
Pagando con Tarjeta de Credito...
Pago procesado con exito.

Pagando con Transferencia bancaria...
Pago procesado con exito.

Pagando en Efectivo...
Pago procesado con exito.
```