TRABAJO PRACTICO N ° 7

Kata 1)

Vehículos y herencia básica

```
public 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);
    }
}
```

```
public class MainVehiculo {
    public static void main(String[] args) {
        Auto a = new Auto("Toyota", "Corolla", 4);
        a.mostrarInfo();
        Vehiculo v = a;
        v.mostrarInfo();
    }
}
```

Kata 2)

Figuras geometricas y metodos abstractos

```
public abstract class Figura {
    protected String nombre;

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

    public abstract double calcularArea();

    public String getNombre() {
        return nombre;
    }
}
```

---

public class Circulo extends Figura { private double radio;

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

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

}
```

---

public class Rectangulo extends Figura {

```
    private double ancho;
    private double alto;

    public Rectangulo(double ancho, double alto) {
        super("Rectangulo");
        this.ancho = ancho;
        this.alto = alto;
    }

    @Override
    public double calcularArea() {
        return ancho * alto;
    }
}
```

```
public class MainFiguras {
    public static void main(String[] args) {
        Figura[] fig = new Figura[3];
        fig[0] = new Circulo(3.0);
        fig[1] = new Rectangulo(4.0, 5.0);
        fig[2] = new Circulo(1.5);

        for (Figura f : fig) {
            System.out.println(f.getNombre() + " Area: " + f.calcularArea());
        }
```

```
    }
}
```

Kata 3)

Empleados y polimorfismo

```
public abstract class Empleado {
    protected String id;
    protected String nombre;

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

    public abstract double calcularSueldo();

    public String getNombre() {
        return nombre;
    }

    public String getId() {
        return id;
    }
}
```

```
public class EmpleadoPlanta extends Empleado {
    private double salarioBase;
```

```
    private double plus;

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

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

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

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

    @Override
    public double calcularSueldo() {
        return valorHora * horasTrabajadas;
    }
```

```
}
```

```
import java.util.ArrayList;
import java.util.List;


public class MainEmpleados {
    public static void main(String[] args) {
        List<Empleado> lista = new ArrayList<>();
        lista.add(new EmpleadoPlanta("E001", "Ana", 80000, 5000));
        lista.add(new EmpleadoTemporal("E002", "Carlos", 1000, 40));
        lista.add(new EmpleadoPlanta("E003", "Laura", 70000, 3000));
        lista.add(new EmpleadoTemporal("E004", "David", 1200, 30));


        for (Empleado e : lista) {
            System.out.println("Empleado: " + e.getNombre() + " Sueldo: " +
e.calcularSueldo());
            if (e instanceof EmpleadoPlanta) {
                System.out.println("  Tipo: Planta");
            } else if (e instanceof EmpleadoTemporal) {
                System.out.println("  Tipo: Temporal");
            }
        }
    }
}
```

Kata 4)

Animales y comportamiento sobrescrito

```
public class Animal {
    protected String nombre;
```

```
    public Animal(String nombre) {
        this.nombre = nombre;
    }

    public void hacerSonido() {
        System.out.println("Sonido generico de animal");
    }

    public void describirAnimal() {
        System.out.println("Animal: " + nombre);
    }
}
```

```
public class Perro extends Animal {
    public Perro(String nombre) {
        super(nombre);
    }

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

```
public class Gato extends Animal {
    public Gato(String nombre) {
        super(nombre);
    }
```

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

```
public class Vaca extends Animal {
    public Vaca(String nombre) {
        super(nombre);
    }

    @Override
    public void hacerSonido() {
        System.out.println("Muu");
    }
}
```

```
import java.util.ArrayList;
import java.util.List;

public class MainAnimales {
    public static void main(String[] args) {
        List<Animal> granja = new ArrayList<>();
        granja.add(new Perro("Rex"));
        granja.add(new Gato("Michi"));
        granja.add(new Vaca("Lola"));
        granja.add(new Perro("Bobby"));
```

```
        for (Animal a : granja) {
            a.describirAnimal();
            a.hacerSonido();
        }
    }
}
```

Anotación:
Todos los ejemplos incluyen ***main*** *de prueba para demostrar los conceptos pedidos en la consigna*