

Ejercicios 3y 4

Garcia Cintia

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

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

    empleados.add(new EmpleadoPlanta (5800.00 , "Juan"));
    empleados.add(new EmpleadoTemporal (4,4500, "Rocio"));

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

        if (e instanceof EmpleadoPlanta){
            System.out.println("Tipo: Planta Permanente");

        }else if(!(e instanceof EmpleadoTemporal)){
        } else {
            System.out.println("tipo: temporal");
        }
    }
}

Empleado: Juan
Sueldo: 5800.0
Tipo: Planta Permanente
Empleado: Rocio
Sueldo: 18000.0
```

```

public abstract class Empleado {
    protected String nombre;

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

    public String getNombre() {
        return nombre;
    }

    public abstract double calcularSueldo();
}

```

```

public class EmpleadoTemporal extends Empleado{
    private final int horasTrabajadas;
    private final int valorHora;

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

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

```

```

public class EmpleadoPlanta extends Empleado{
    private double sueldoBase;

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

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

```

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

The screenshot shows an IDE with several tabs: CategoriaAnimales.java, Animal.java (active), Perro.java, and Vaca.java. The Animal.java file contains the following code:

```
public class Animal {  
    protected String nombre;  
  
    public Animal(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public String hacerSonido(){  
        return "Sonido especifico";  
    }  
  
    public String describirAnimal(){  
        return "Ese animal se llama: " + nombre ;  
    }  
}
```

Below the code editor, there is an output window titled "HerenciaPoliformismo - C:\Users\HP\Desktop\HerenciaPoliformismo" and "CategoriaAnimales (run)". The output shows the results of running the program:

```
Ese animal se llama: Pumba  
Sonido Miau Maiou  
  
Ese animal se llama: Milka  
Sonido Muuuu
```

The screenshot shows the same IDE with an additional tab, Gato.java. The active file is CategoriaAnimales.java, which contains the following code:

```
public static void main(String[] args) {  
    ArrayList<Animal>animales= new ArrayList<>();  
  
    animales.add(new Perro ("Pipo"));  
    animales.add(new Gato("Pumba"));  
    animales.add(new Vaca ("Milka"));  
  
    for (Animal a :animales){  
        System.out.println(a.describirAnimal());  
        System.out.println("Sonido " + a.hacerSonido());  
        System.out.println();  
    }  
}
```

```

    */
    public class Vaca extends Animal{

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

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

```

CategoriaAnimales.java [-/A] x Animal.java [-/A] x Perro.java [-/A] x

```

7  /**
8   *
9   * @author HP
10  */
11  public class Perro extends Animal{
12
13      public Perro(String nombre) {
14          super(nombre);
15      }
16
17      @Override
18      public String hacerSonido(){
19          return "Guau Guau";
20      }
21  }

```

```

    * @author HP
    */
    public class Gato extends Animal{

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

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

```