CLASE EMPLEADO

Atributos:

- int id: Identificador único del empleado.
- String nombre: Nombre completo.
- String puesto: Cargo que desempeña.
- double salario: Salario actual.
- static int totalEmpleados: Contador global de empleados creados.

```
private int id;
private String nombre;
private String puesto;
private double salario;
private static int totalEmpleados = 0;
```

REQUERIMIENTOS

1. Uso de this:
    - Utilizar this en los constructores para distinguir parámetros de atributos.
2. Constructores sobrecargados:
    - Uno que reciba todos los atributos como parámetros.
    - Otro que reciba solo nombre y puesto, asignando un id automático y un salario por defecto.
    - Ambos deben incrementar totalEmpleados.

```java
public Empleado(int id, String nombre, String puesto, double salario) {
    this.id = id;
    this.nombre = nombre;
    this.puesto = puesto;
    this.salario = salario;
    totalEmpleados++;
}

public Empleado(String nombre, String puesto) {
    this.id = (int) (Math.random() * (50 - 10 + 1)) + 10;
    this.nombre = nombre;
    this.puesto = puesto;
    this.salario = 1000000;
    totalEmpleados++;
}
```

3. Métodos sobrecargados actualizarSalario:

    ○ Uno que reciba un porcentaje de aumento.

    ○ Otro que reciba una cantidad fija a aumentar.

```java
public void actualizarSalario(double porcentajeAumento){
    this.salario = this.salario + (this.salario * porcentajeAumento / 1000);
}

public void actualizarSalario(){
   actualizarSalario(20);
}
```

4. Método toString():

    ○ Mostrar id, nombre, puesto y salario de forma legible.

```java
@Override
public String toString() {
    return "Empleado{" + "id=" + id + ", nombre=" + nombre + ", puesto=" + puesto + ", salario=" + salario + '}';
}
```

5. Método estático mostrarTotalEmpleados():

    ○ Retornar el total de empleados creados hasta el momento.

```java
public static int mostrarTotalEmpleados(){
    return Empleado.totalEmpleados;
}
```

6. Encapsulamiento en los atributos:

    ○ Restringir el acceso directo a los atributos de la clase.

    ○ Crear los métodos Getters y Setters correspondientes.

```java
// Getters y Setters

public int getId() {
    return id;
}

public String getNombre() {
    return nombre;
}

public String getPuesto() {
    return puesto;
}

public double getSalario() {
    return salario;
}

public void setId(int id) {
    this.id = id;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public void setPuesto(String puesto) {
    this.puesto = puesto;
}

public void setSalario(double salario) {
    this.salario = salario;
}
```

TAREAS A REALIZAR

1. Implementar la clase Empleado aplicando todos los puntos anteriores.

2. Crear una clase de prueba con método main que:

○ Instancie varios objetos usando ambos constructores.

```
Empleado empleadoUno = new Empleado(1, "JUAN", "PROGRAMADOR", 2000.50);
Empleado empleadoDos = new Empleado(2, "LUCAS", "EMPRESARIO", 15000.50);
Empleado empleadoTres = new Empleado("FERNADO", "PELUCQUERO");
Empleado empleadoCuatro = new Empleado( "FACUNDO", "ABOGADO");
```

○ Aplique los métodos actualizarSalario() sobre distintos empleados.

```
empleadoUno.actualizarSalario(10);
empleadoDos.actualizarSalario(65);
empleadoTres.actualizarSalario();
empleadoCuatro.actualizarSalario();
```

○ Imprima la información de cada empleado con toString().

```
Empleado{id=1, nombre=JUAN, puesto=PROGRAMADOR, salario=2000.5}
Empleado{id=2, nombre=LUCAS, puesto=EMPRESARIO, salario=15000.5}
Empleado{id=17, nombre=FERNADO, puesto=PELUCQUERO, salario=1000000.0}
Empleado{id=36, nombre=FACUNDO, puesto=ABOGADO, salario=1000000.0}

Empleado{id=1, nombre=JUAN, puesto=PROGRAMADOR, salario=2020.505}
Empleado{id=2, nombre=LUCAS, puesto=EMPRESARIO, salario=15975.5325}
Empleado{id=17, nombre=FERNADO, puesto=PELUCQUERO, salario=1020000.0}
Empleado{id=36, nombre=FACUNDO, puesto=ABOGADO, salario=1020000.0}
```

○ Muestre el total de empleados creados con mostrarTotalEmpleados().

```
la cantidad de empleados es: 4
------------------------------------------------
```

LINK AL REPOSITORIO:

https://github.com/BarreraMariano/UTN-TUPaD-P2