

Trabajo Práctico 3: Introducción a la POO

Alumno: Facundo Gaston Vazquez

Comision: 16

Repositorio: https://github.com/facuvazquez1/UTN-TUPAD-PROGRAMACION_2

1. Registro de Estudiantes

Archivo: *Estudiante.java*

```
public class Estudiante {
    // Atributos de la clase Estudiante
    private String nombre;
    private String apellido;
    private String curso;
    private double calificacion;

    // Constructor para inicializar un nuevo objeto Estudiante
    public Estudiante(String nombre, String apellido, String curso, double calificacion) {
        this.nombre = nombre;
        this.apellido = apellido;
        this.curso = curso;
        this.calificacion = calificacion;
    }

    // Método para mostrar la información del estudiante
    public void mostrarInfo() {
        System.out.println("--- Información del Estudiante ---");
        System.out.println("Nombre: " + this.nombre + " " + this.apellido);
        System.out.println("Curso: " + this.curso);
        System.out.println("Calificación: " + this.calificacion);
    }

    // Método para aumentar la calificación
    public void subirCalificacion(double puntos) {
        this.calificacion += puntos;
        System.out.println("Calificación aumentada en " + puntos + " puntos.");
    }
}
```

```

// Método para disminuir la calificación
public void bajarCalificacion(double puntos) {
    this.calificacion -= puntos;
    System.out.println("Calificación disminuida en " + puntos + "
puntos.");
}
}

```

2. Registro de Mascotas

Archivo: *Mascota.java*

```

public class Mascota {
    // Atributos de la clase Mascota
    private String nombre;
    private String especie;
    private int edad;

    // Constructor para inicializar un nuevo objeto Mascota
    public Mascota(String nombre, String especie, int edad) {
        this.nombre = nombre;
        this.especie = especie;
        this.edad = edad;
    }

    // Método para mostrar la información de la mascota
    public void mostrarInfo() {
        System.out.println("--- Información de la Mascota ---");
        System.out.println("Nombre: " + this.nombre);
        System.out.println("Especie: " + this.especie);
        System.out.println("Edad: " + this.edad + " años");
    }

    // Método para simular que la mascota cumple años
    public void cumplirAños() {
        this.edad++;
        System.out.println("¡Feliz cumpleaños " + this.nombre + "! Ahora
tiene " + this.edad + " años.");
    }
}

```

3. Encapsulamiento con la Clase Libro

Archivo: *Libro.java*

```
public class Libro {
    // Atributos privados para aplicar encapsulamiento
    private String titulo;
    private String autor;
    private int anoPublicacion;

    // Constructor
    public Libro(String titulo, String autor, int anoPublicacion) {
        this.titulo = titulo;
        this.autor = autor;
        this.anoPublicacion = anoPublicacion;
    }

    // Getters para acceder a los atributos privados
    public String getTitulo() {
        return titulo;
    }

    public String getAutor() {
        return autor;
    }

    public int getAnoPublicacion() {
        return anoPublicacion;
    }

    // Setter con validación para el año de publicación
    public void setAnoPublicacion(int anoPublicacion) {
        if (anoPublicacion > 0) {
            this.anoPublicacion = anoPublicacion;
            System.out.println("Año de publicación actualizado correctamente.");
        } else {
            System.out.println("Error: El año de publicación no puede ser un valor negativo o cero.");
        }
    }
}
```

```
}
```

4. Gestión de Gallinas en Granja Digital

Archivo: *Gallina.java*

```
public class Gallina {
    // Atributos de la clase Gallina
    private int idGallina;
    private int edad;
    private int huevosPuestos;

    // Constructor
    public Gallina(int idGallina, int edad) {
        this.idGallina = idGallina;
        this.edad = edad;
        this.huevosPuestos = 0; // Se inicializa en 0
    }

    // Método para que la gallina ponga un huevo
    public void ponerHuevo() {
        this.huevosPuestos++;
        System.out.println("La gallina " + this.idGallina + " ha puesto un
huevo. Total: " + this.huevosPuestos);
    }

    // Método para envejecer la gallina
    public void envejecer() {
        this.edad++;
        System.out.println("La gallina " + this.idGallina + " ha
envejecido. Nueva edad: " + this.edad);
    }

    // Método para mostrar el estado actual de la gallina
    public void mostrarEstado() {
        System.out.println("--- Estado de La Gallina " + this.idGallina + "
---");
        System.out.println("Edad: " + this.edad + " años");
        System.out.println("Huevos puestos: " + this.huevosPuestos);
    }
}
```

5. Simulación de Nave Espacial

Archivo: *NaveEspacial.java*

```
public class NaveEspacial {
    // Atributos de la clase NaveEspacial
    private String nombre;
    private double combustible;
    private final double COMBUSTIBLE_MAX = 100.0;

    // Constructor
    public NaveEspacial(String nombre, double combustible) {
        this.nombre = nombre;
        this.combustible = combustible;
    }

    // Método para simular el despegue
    public void despegar() {
        System.out.println("La nave " + this.nombre + " ha despegado.");
    }

    // Método para avanzar con validación de combustible
    public void avanzar(double distancia) {
        double consumo = distancia * 0.5; // Supongamos que se consumen 0.5
        unidades por km
        if (this.combustible >= consumo) {
            this.combustible -= consumo;
            System.out.println("La nave " + this.nombre + " ha avanzado " +
            distancia + " km.");
        } else {
            System.out.println("Advertencia: No hay suficiente combustible
            para avanzar " + distancia + " km.");
        }
    }

    // Método para recargar combustible con validación de límite
    public void recargarCombustible(double cantidad) {
        if (this.combustible + cantidad <= COMBUSTIBLE_MAX) {
```

```

        this.combustible += cantidad;
        System.out.println("Se han recargado " + cantidad + " unidades
de combustible.");
    } else {
        System.out.println("Advertencia: No se puede recargar. Se
superaría el límite de combustible.");
    }
}

// Método para mostrar el estado actual de la nave
public void mostrarEstado() {
    System.out.println("--- Estado de la Nave " + this.nombre + "
---");
    System.out.println("Combustible restante: " + this.combustible + "
unidades.");
}
}

```

5. Simulación de Nave Espacial

Archivo: *Principal.java*

```

public class Principal {
    public static void main(String[] args) {
        // --- 1. Registro de Estudiantes ---
        System.out.println("\n--- Tarea: Registro de Estudiantes ---");
        Estudiante estudiante1 = new Estudiante("Juan", "Perez", "Programación II", 85.5);
        estudiante1.mostrarInfo();
        estudiante1.subirCalificacion(5.0);
        estudiante1.bajarCalificacion(2.5);
        estudiante1.mostrarInfo();

        // --- 2. Registro de Mascotas ---
        System.out.println("\n--- Tarea: Registro de Mascotas ---");
        Mascota mascota1 = new Mascota("Max", "Perro", 3);
        mascota1.mostrarInfo();
        mascota1.cumplirAnios();
        mascota1.mostrarInfo();

        // --- 3. Encapsulamiento con la Clase Libro ---
        System.out.println("\n--- Tarea: Encapsulamiento con la Clase Libro ---");
        Libro libro1 = new Libro("El Señor de los Anillos", "J.R.R. Tolkien", 1954);
        System.out.println("Título: " + libro1.getTitulo() + ", Autor: " + libro1.getAutor())
    }
}

```

```

+ ", Año: " + libro1.getAnoPublicacion());

    System.out.println("Intentando modificar el año con un valor inválido (-10):");
    libro1.setAnoPublicacion(-10);

    System.out.println("Intentando modificar el año con un valor válido (1965):");
    libro1.setAnoPublicacion(1965);
    System.out.println("Año de publicación final: " + libro1.getAnoPublicacion());

    // --- 4. Gestión de Gallinas en Granja Digital ---
    System.out.println("\n--- Tarea: Gestión de Gallinas ---");
    Gallina gallina1 = new Gallina(1, 1);
    Gallina gallina2 = new Gallina(2, 2);

    gallina1.ponerHuevo();
    gallina2.ponerHuevo();
    gallina2.ponerHuevo();
    gallina1.envejecer();
    gallina1.mostrarEstado();
    gallina2.mostrarEstado();

    // --- 5. Simulación de Nave Espacial ---
    System.out.println("\n--- Tarea: Simulación de Nave Espacial ---");
    NaveEspacial nave1 = new NaveEspacial("Apolo 13", 50);
    nave1.mostrarEstado();

    System.out.println("Intentando avanzar 200 km sin recargar:");
    nave1.avanzar(200);
    nave1.mostrarEstado();

    System.out.println("Recargando combustible:");
    nave1.recargarCombustible(60);
    nave1.mostrarEstado();

    System.out.println("Intentando avanzar 200 km después de recargar:");
    nave1.avanzar(200);
    nave1.mostrarEstado();

    System.out.println("Intentando recargar más allá del límite:");
    nave1.recargarCombustible(100);
}
}

```