

Trabajo Práctico 3 – Introducción a la Programación Orientada a Objetos

Alumno: Ciro Giorgini

Materia: Programación Estructurada

Año: 2025

Ejercicio 1 – Clase Estudiante

En este ejercicio cree la clase Estudiante con atributos basicos y metodos para mostrar la informacion y modificar la calificacion. Esto me permitio comprender como un objeto puede representar a una persona con datos y acciones.

```
package tp3;
```

```
public class Estudiante {
    String nombre;
    String apellido;
    String curso;
    int calificacion;

    public void mostrarInfo() {
        System.out.println(nombre + " " + apellido + " - Curso: " + curso + " - Calificacion: "
    }

    public void subirCalificacion(int puntos) {
        calificacion += puntos;
    }

    public void bajarCalificacion(int puntos) {
        calificacion -= puntos;
    }
}
```

Ejercicio 2 – Clase Mascota

En este ejercicio represente una mascota con su nombre, especie y edad. Agregue un metodo para que cumpla años y otro para mostrar la informacion.

```
package tp3;
```

```
public class Mascota {  
    String nombre;  
    String especie;  
    int edad;
```

```
    public void mostrarInfo() {  
        System.out.println("Nombre: " + nombre + " - Especie: " + especie + " - Edad: " + edad);  
    }
```

```
    public void cumplirAnios() {  
        edad++;  
    }  
}
```

Ejercicio 3 – Clase Libro

En este ejercicio trabaje con encapsulamiento. Declare los atributos privados y use getters y setters para acceder a ellos. Además valide que el año de publicación sea válido.

```
package tp3;
```

```
public class Libro {
    private String titulo;
    private String autor;
    private int anioPublicacion;

    public Libro(String titulo, String autor, int anioPublicacion) {
        this.titulo = titulo;
        this.autor = autor;
        setAnioPublicacion(anioPublicacion);
    }

    public String getTitulo() { return titulo; }
    public String getAutor() { return autor; }
    public int getAnioPublicacion() { return anioPublicacion; }

    public void setAnioPublicacion(int anio) {
        if (anio > 0 && anio <= 2025) {
            this.anioPublicacion = anio;
        } else {
            System.out.println("Anio invalido");
        }
    }

    public void mostrarInfo() {
        System.out.println(titulo + " de " + autor + " (" + anioPublicacion + ")");
    }
}
```

Ejercicio 4 – Clase Gallina

En este ejercicio implemente una clase Gallina con metodos para envejecer y poner huevos. Me ayudo a entender como un objeto puede cambiar su estado a traves de metodos.

```
package tp3;
```

```
public class Gallina {
    int idGallina;
    int edad;
    int huevosPuestos;

    public void ponerHuevo() {
        huevosPuestos++;
    }

    public void envejecer() {
        edad++;
    }

    public void mostrarEstado() {
        System.out.println("Gallina " + idGallina + " - Edad: " + edad + " - Huevos: " + huevosPuestos);
    }
}
```

Ejercicio 5 – Clase Nave Espacial

En este ejercicio desarrolle una clase NaveEspacial que puede despegar, avanzar y recargar combustible. Esto me permitio practicar el uso de condiciones y validaciones en los metodos.

```
package tp3;
```

```
public class NaveEspacial {
    String nombre;
    int combustible;
    int capacidadMaxima = 100;

    public void despegar() {
        if (combustible >= 10) {
            combustible -= 10;
            System.out.println(nombre + " ha despegado.");
        } else {
            System.out.println("No hay suficiente combustible para despegar.");
        }
    }

    public void avanzar(int distancia) {
        int consumo = distancia * 2;
        if (combustible >= consumo) {
            combustible -= consumo;
            System.out.println(nombre + " avanzo " + distancia + " km.");
        } else {
            System.out.println("No hay suficiente combustible para avanzar.");
        }
    }

    public void recargarCombustible(int cantidad) {
        if (combustible + cantidad <= capacidadMaxima) {
            combustible += cantidad;
            System.out.println("Recargado " + cantidad + ". Combustible actual: " + combustible);
        } else {
            System.out.println("No se puede superar el limite de " + capacidadMaxima);
        }
    }

    public void mostrarEstado() {
        System.out.println(nombre + " - Combustible: " + combustible);
    }
}
```

Clase Principal – TP3

Finalmente, cree una clase principal TP3 donde probe todas las clases anteriores. De esta manera verifique que el código funcionaba correctamente.

```
package tp3;

public class TP3 {
    public static void main(String[] args) {
        // Estudiante
        Estudiante e = new Estudiante();
        e.nombre = "Ciro";
        e.apellido = "Giorgini";
        e.curso = "Programacion II";
        e.calificacion = 7;
        e.mostrarInfo();
        e.subirCalificacion(2);
        e.bajarCalificacion(1);
        e.mostrarInfo();

        // Mascota
        Mascota m = new Mascota();
        m.nombre = "Firulais";
        m.especie = "Perro";
        m.edad = 3;
        m.mostrarInfo();
        m.cumplirAnios();
        m.mostrarInfo();

        // Libro
        Libro l = new Libro("El Principito", "Saint-Exupery", 1943);
        l.mostrarInfo();
        l.setAnioPublicacion(-5);
        l.setAnioPublicacion(2000);
        l.mostrarInfo();

        // Gallina
        Gallina g1 = new Gallina();
        g1.idGallina = 1;
        g1.edad = 1;
        g1.ponerHuevo();
        g1.envejecer();
        g1.mostrarEstado();

        Gallina g2 = new Gallina();
        g2.idGallina = 2;
        g2.edad = 2;
        g2.ponerHuevo();
        g2.ponerHuevo();
        g2.mostrarEstado();

        // Nave Espacial
        NaveEspacial nave = new NaveEspacial();
        nave.nombre = "Apollo";
        nave.combustible = 50;
        nave.avanzar(10);
        nave.recargarCombustible(30);
        nave.despegar();
    }
}
```

```
        nave.avanzar(20);  
        nave.mostrarEstado();  
    }  
}
```


Conclusión

Con este trabajo practico aprendi a crear clases, objetos, atributos y metodos en Java. Tambien puse en practica el encapsulamiento y la validacion de datos. Siento que me ayudo a comprender mejor los conceptos basicos de la programacion orientada a objetos y a escribir un codigo mas organizado y modular.