

PROGRAMACIÓN II

Trabajo Práctico 3: Introducción a la Programación Orientada a Objetos

Alumno: Farfán Matías Gastón

Github:

<https://github.com/matfarfan/Programacion-II-UTN.git>

OBJETIVO GENERAL

Comprender los fundamentos de la Programación Orientada a Objetos, incluyendo clases, objetos, atributos y métodos, para estructurar programas de manera modular y reutilizable en Java.

MARCO TEÓRICO

Concepto	Aplicación en el proyecto
Clases y Objetos	Modelado de entidades como Estudiante, Mascota, Libro, Gallina y NaveEspacial
Atributos y Métodos	Definición de propiedades y comportamientos para cada clase
Estado e Identidad	Cada objeto conserva su propio estado (edad, calificación, combustible, etc.)
Encapsulamiento	Uso de modificadores de acceso y getters/setters para proteger datos
Modificadores de acceso	Uso de private, public y protected para controlar visibilidad
Getters y Setters	Acceso controlado a atributos privados mediante métodos

Reutilización de código

Definición de clases reutilizables en
múltiples contextos

Caso Práctico

Desarrollar en Java los siguientes ejercicios aplicando los conceptos de programación orientada a objetos:

1. Registro de Estudiantes

- Crear una clase Estudiante con los atributos: nombre, apellido, curso, calificación.

Métodos requeridos: `mostrarInfo()`, `subirCalificacion(puntos)`, `bajarCalificacion(puntos)`.

Tarea: Instanciar a un estudiante, mostrar su información, aumentar y disminuir calificaciones.

2. Registro de Mascotas

- Crear una clase Mascota con los atributos: nombre, especie, edad.

Métodos requeridos: `mostrarInfo()`, `cumplirAnios()`.

Tarea: Crear una mascota, mostrar su información, simular el paso del tiempo y verificar los cambios.

3. Encapsulamiento con la Clase Libro

- Crear una clase Libro con atributos privados: titulo, autor, añoPublicacion.

Métodos requeridos: Getters para todos los atributos. Setter con validación para añoPublicacion.

Tarea: Crear un libro, intentar modificar el año con un valor inválido y luego con uno válido, mostrar la información final.

4. Gestión de Gallinas en Granja Digital

- a. Crear una clase Gallina con los atributos: idGallina, edad, huevosPuestos.

Métodos requeridos: `ponerHuevo()`, `envejecer()`, `mostrarEstado()`.

Tarea: Crear dos gallinas, simular sus acciones (envejecer y poner huevos), y mostrar su estado.

5. Simulación de Nave Espacial

Crear una clase NaveEspacial con los atributos: nombre, combustible.

Métodos requeridos: `despegar()`, `avanzar(distancia)`, `recargarCombustible(cantidad)`, `mostrarEstado()`.

Reglas: Validar que haya suficiente combustible antes de avanzar y evitar que se supere el límite al recargar.

Tarea: Crear una nave con 50 unidades de combustible, intentar avanzar sin recargar, luego recargar y avanzar correctamente. Mostrar el estado al final.

CONCLUSIONES ESPERADAS

- Comprender la diferencia entre clases y objetos.
- Aplicar principios de encapsulamiento para proteger los datos.
- Usar getters y setters para gestionar atributos privados.
- Implementar métodos que definen comportamientos de los objetos.
- Manejar el estado y la identidad de los objetos correctamente.
- Aplicar buenas prácticas en la estructuración del código orientado a objetos.
- Reforzar el pensamiento modular y la reutilización del código en Java.

Resolución

- Principal

```
package tp3_poo;

import java.util.Scanner;

/**
 *
 * @author mfarf
 */
public class Principal {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
```

```
Scanner input = new Scanner(System.in);

//Estudiante
System.out.println("1- Estudiante");
//Instanciar estudiante
Estudiante e = new Estudiante();
e.nombre = "Matias";
e.apellido = "Farfan";
e.curso = "Programacion II";
e.calificacion = 7;
//mostrar su info
e.mostrarInfo();
//aumentar y disminuir calificaciones
System.out.println(" ");
e.subirCalificacion(8);
System.out.println("Calificación: "+e.calificacion);
System.out.println(" ");
e.bajarCalificacion(10);
System.out.println("Calificación: "+e.calificacion);

//Mascota
System.out.println(" ");
System.out.println("2- Mascota");
//Crear mascota
Mascota m = new Mascota();
m.nombre = "Ciro";
m.especie = "Boxer";
m.edad = 8;
//Mostrar info
m.mostrarInfo();
//Paso del tiempo
m.cumplirAnios();
m.cumplirAnios();
System.out.println(" ");
System.out.println("Dos años después");
System.out.println("Nueva edad: " + m.edad);

//Libro
System.out.println(" ");
System.out.println("3- Libro (encapsulamiento)");
//Crear libro
Libro l = new Libro("Cien años de soledad", "Gabriel García Marquez", 1967);
//Mostramos
System.out.println("Título: " + l.getTitulo());
System.out.println("Autor: " + l.getAutor());
System.out.println("Año: " + l.getAnioPublicacion());
//Intentamos modificar
System.out.println(" ");
l.setAnioPublicacion(2030);
System.out.println(" ");
l.setAnioPublicacion(1980);
//VOLvemos a mostrar
System.out.println(" ");
System.out.println("Título: " + l.getTitulo());
System.out.println("Autor: " + l.getAutor());
System.out.println("Año: " + l.getAnioPublicacion());
```

```
//Gallina
System.out.println(" ");
System.out.println("4- Gallina");
//Crear dos gallinas
Gallina g1 = new Gallina();
g1.idGallina = 0001;
g1.edad = 3;
g1.huevosPuestos = 15;
g1.mostrarEstado();
System.out.println(" ");

Gallina g2 = new Gallina();
g2.idGallina = 0002;
g2.edad = 5;
g2.huevosPuestos = 28;
g2.mostrarEstado();

//Simulacion y nuevos datos
System.out.println(" ");
System.out.println("Nuevos estados: ");
g1.ponerHuevos(); g1.ponerHuevos(); g1.ponerHuevos();
g1.envejecer();
g1.mostrarEstado();
System.out.println(" ");

g2.ponerHuevos();
g2.envejecer(); g2.envejecer(); g2.envejecer();
g2.mostrarEstado();

//Nave espacial
System.out.println(" ");
System.out.println("5- Nave Espacial");
NaveEspacial n = new NaveEspacial();
n.nombre = "Halcón Galáctico";
n.combustible = 50;
n.mostrarEstado();
//despegue
n.despegar();
//avanzar
n.avanzar(15);
n.avanzar(12);
n.avanzar(100);
//Recarga combustible
System.out.println(" ");
n.recargarCombustible(100);
n.recargarCombustible(20);
//Mostrar estado
System.out.println(" ");
n.mostrarEstado();

    }
}
```

- Estudiante
public class Estudiante {
 String nombre;
 String apellido;

```
String curso;
int calificacion;

public void mostrarInfo() {
    System.out.println(nombre + " " + apellido + "\nCurso: " + curso + "\nCalificación: " + calificacion);
}

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

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

- **Mascota**

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

    public void mostrarInfo(){
        System.out.println("Nombre: " + nombre + "\nEspecie: " + especie + "\nEdad: " + edad);
    }

    public void cumplirAnios(){
        edad++;
    }

}
```

- **Libro**

```
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;
        this.anioPublicacion = anioPublicacion;
    }

    public String getTitulo() {
        return titulo;
    }

    public String getAutor() {
        return autor;
    }

    public int getAnioPublicacion() {
        return anioPublicacion;
    }
}
```

```
}  
  
public void setAnioPublicacion(int anioPublicacion) {  
    if (anioPublicacion>0 && anioPublicacion <= 2025) {  
        this.anioPublicacion = anioPublicacion;  
        System.out.println("Nuevo año de publicación: "+anioPublicacion);  
    } else{  
        System.out.println(anioPublicacion + " es un año invalido");  
    }  
}  
  
}
```

- **Gallina**

```
public class Gallina {  
    int idGallina;  
    int edad;  
    int huevosPuestos;  
  
    public void ponerHuevos(){  
        huevosPuestos++;  
    }  
  
    public void envejecer() {  
        edad++;  
    }  
  
    public void mostrarEstado() {  
        System.out.println("Gallina: " + idGallina + "\nEdad: " + edad + "\nHuevos: " +  
huevosPuestos);  
    }  
}
```

- **Nave Espacial**

```
public class NaveEspacial {  
    String nombre;  
    int combustible;  
  
    public void despegar() {  
        if (combustible>0) {  
            System.out.println("Despegue exitoso!!!");  
        }else{  
            System.out.println("Combustible insuficiente para despegar");  
        }  
    }  
  
    public void avanzar(int distancia) {  
        if (combustible>=distancia) {  
            combustible -= distancia;  
            System.out.println("Avanzamos "+distancia+" de distancia.");  
        }else{  
            System.out.println("Combustible insuficiente para esa distancia: "+distancia);  
        }  
    }  
}
```

```
public void recargarCombustible(int cantidad) {  
    if (cantidad>0 && (combustible+cantidad)<=50) {  
        combustible += cantidad;  
        System.out.println("Regarca de "+cantidad+" exitosa!");  
    } else{  
        System.out.println("Cantidad de combustible inválido para recargar:  
"+cantidad);  
    }  
}  
  
public void mostrarEstado() {  
    System.out.println("Nave: "+nombre+"\nCombustible: "+combustible);  
}  
}
```