

## PROGRAMACIÓN II

### TRABAJO PRÁCTICO 3: INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS

CHRISTIAN EMMANUEL OLIVERO

<https://github.com/CHRISTIAN2320/UTN-TUPAD-Programacion2>

#### 1. Registro de Estudiantes

a. 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.

```
public class Ejercicio1 {  
  
    public static void main(String[] args) {  
        Estudiante e1= new Estudiante ();  
  
        e1.mostrarInfo();  
        int calificacion_incrementada = e1.subirCalificacion(100);  
        e1.mostrarInfo();  
        int calificacion_decrementada = e1.bajarCalificacion(105);  
        e1.mostrarInfo();  
    }  
}
```

```
public class Estudiante {  
    String nombre = "Christian";  
    String apellido = "Olivero";  
    int curso = 123;  
    int calificacion = 100;  
  
    void mostrarInfo() {  
        System.out.println(nombre + " " + apellido + " " + curso  
+" "+calificacion );  
    }  
  
    int subirCalificacion (int puntos) {  
        return calificacion += puntos;  
    }  
  
    int bajarCalificacion(int puntos) {  
        return calificacion -= puntos;  
    }  
}
```

```
Christian Olivero 123 100  
Christian Olivero 123 200  
Christian Olivero 123 95  
BUILD SUCCESSFUL (total time: 0 seconds)
```

## 2. Registro de Mascotas

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

**Métodos requeridos:** mostrarInfo(), cumplirAños().

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

```
public class Ejercicio2 {  
  
    public static void main(String[] args) {  
        Mascota m1 = new Mascota();  
  
        m1.mostrarInfo();  
        m1.cumplirAños(5);  
    }  
}
```

```
public class Mascota {  
    String nombre = "Firulays";  
    String especie = "Dogo Argentino";  
    int edad = 5 ;  
  
    void mostrarInfo(){  
        System.out.println(nombre + " es un " + especie + " y tiene " + edad + " años" );  
    }  
  
    void cumplirAños(int años) {  
        edad+= años;  
        System.out.println("Luego de " + años + "años, " + nombre + " tiene ahora" + edad + " años" );  
    }  
}
```

```
Firulays es un Dogo Argentino y tiene 5 años  
Luego de 5años, Firulays tiene ahora10 años  
BUILD SUCCESSFUL (total time: 0 seconds)
```

### 3. Encapsulamiento con la Clase Libro

a. 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.

```
public class Ejercicio3 {  
  
    public static void main(String[] args) {  
        Libro l1 = new Libro();  
  
        l1.setAnioPublicacion(-1932); // Carga de año invalido  
  
        System.out.println(" Titulo: " +l1.getTitulo()+ "\n Autor:"  
+l1.getAutor() + "\n Anio:" + l1.getAnioPublicacion() );  
  
        l1.setAnioPublicacion(1308);  
        System.out.println(" Titulo: " +l1.getTitulo()+ "\n Autor:"  
+l1.getAutor() + "\n Anio:" + l1.getAnioPublicacion() );  
    }  
  
}
```

```
public class Libro {  
    private String titulo = "La Divina Comedia ";  
    private String autor = "Dante Alighieri" ;  
    private int anioPublicacion;  
  
    public String getTitulo() {  
        return titulo;  
    }  
  
    public String getAutor() {  
        return autor;  
    }  
  
    public int getAnioPublicacion() {  
        return anioPublicacion;  
    }  
  
    public void setAnioPublicacion(int anioPublicacion) {  
        if (anioPublicacion > 0 ) {  
            this.anioPublicacion = anioPublicacion;  
        }  
    }  
}
```

```
Titulo: La Divina Comedia
Autor:Dante Alighieri
Anio:0
Titulo: La Divina Comedia
Autor:Dante Alighieri
Anio:1308
```

#### 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.

```
public static void main(String[] args) {
    //Creamos las dos gallinas
    Gallina g1 = new Gallina();
    Gallina g2 = new Gallina();

    // Estado inicial:
    g1.idGallina = 0001;
    g1.edad= 1;
    g1.huevosPuestos=20;

    g2.idGallina = 0002;
    g2.edad= 2;
    g2.huevosPuestos=30;

    g1.mostrarEstado();
    g2.mostrarEstado();

    // Con el tiempo
    g1.ponerHuevo(50);
    g1.envejecer(3);

    g2.ponerHuevo(10);
    g2.envejecer(3);

    g1.mostrarEstado();
    g2.mostrarEstado();
}
```

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

    void ponerHuevo (int cantidad){
        huevosPuestos += cantidad;
    }

    void envejecer (int años){
        edad+=años;
    }

    void mostrarEstado(){
        System.out.println("El id de la gallina es: " +
            idGallina + ", la edad de la gallina es " + edad + " años, y
            ha puesto " + huevosPuestos + " huevos");
    }
}
```

```
El id de la gallina es: 1, la edad de la gallina es 1 años, y ha puesto 20 huevos
El id de la gallina es: 2, la edad de la gallina es 2 años, y ha puesto 30 huevos
El id de la gallina es: 1, la edad de la gallina es 4 años, y ha puesto 70 huevos
El id de la gallina es: 2, la edad de la gallina es 5 años, y ha puesto 40 huevos
```

## 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.

```
public class Ejercicio5 {

    public static void main(String[] args) {
        NaveEspacial n1 = new NaveEspacial();
        n1.nombre="Nostradamus";
        n1.combustible=50;
        // DESPEGUE
        n1.despegar();
        // Avanzar
        n1.avanzar(60);
        // Recargar
        n1.recargarCombustible(30);
        // Avanzar
        n1.avanzar(60);
        // Estado Final
        n1.mostrarEstado();
    }
}
```

Puede despegar

No puede Avanzar porque no hay suficiente combustible

Puede Avanzar

El estado de la nave es:

Nombre: Nostradamus

Combustible: 20

```
public class NaveEspacial {
    String nombre;
    int combustible;

    void despegar(){
        if (combustible != 0) {
            System.out.println("Puede despegar");
        }
    }

    void avanzar(int distancia){
        if (combustible >= distancia) {
            System.out.println("Puede Avanzar");
            combustible -= distancia;
        } else {
            System.out.println("No puede Avanzar porque no hay suficiente combustible");
        }
    }

    void recargarCombustible(int cantidad){
        combustible+=cantidad;
    }

    void mostrarEstado(){
        System.out.println("El estado de la nave es: \n"
            + "Nombre: "+nombre+ "\n Combustible: " + combustible );
    }
}
```