

Aranchet Carina – TP 3

EJ1)

The screenshot shows an IDE interface with two code editors and a terminal window.

Code Editor 1 (Left): Contains the `Estudiante.java` file. The code defines a class `Estudiante` with methods `mostrarInfo()`, `subirCalificacion(double puntos)`, and `bajarCalificacion(double puntos)`. It also has a constructor and a main method.

```
public class Estudiante {
    String nombre;
    String apellido;
    String curso;
    double calificacion;
}

public void mostrarInfo() {
    System.out.println("Estudiante: " + nombre + " " + apellido);
    System.out.println("Curso: " + curso);
    System.out.println("Calificación: " + calificacion);
    System.out.println("-----");
}

public void subirCalificacion(double puntos) {
    calificacion += puntos;
    if (calificacion > 10) calificacion = 10;
}

public void bajarCalificacion(double puntos) {
    calificacion -= puntos;
    if (calificacion < 0) calificacion = 0;
}
}
```

Code Editor 2 (Right): Contains the `MainEJ1.java` file. It creates an `Estudiante` object, calls its methods, and prints the results.

```
package EJ1;
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
/**
 *
 * @author Carina Aranchet
 */
public class MainEJ1 {
    public static void main(String[] args) {
        Estudiante el = new Estudiante();
        el.nombre = "Carina";
        el.apellido = "Aranchet";
        el.curso = "Programación 1";
        el.calificacion = 7.5;

        el.mostrarInfo();
        el.subirCalificacion(puntos: 1.0);
        el.mostrarInfo();

        el.bajarCalificacion(puntos: 3.0);
        el.mostrarInfo();
    }
}
```

Terminal Output:

```
UTN-TUPd-P2 - C:\Users\Carina Aranchet\FACU\2do Semestre\Programación II\UTN-TUPd-P2 x Tp3_Introducción_a_la_POO (run) x

RUN:
Estudiante: Carina A.
Curso: Programación 1
Calificación: 7.5
-----
Estudiante: Carina A.
Curso: Programación 1
Calificación: 8.0
-----
Estudiante: Carina A.
Curso: Programación 1
Calificación: 5.5
-----
BUILD SUCCESSFUL (total time: 0 seconds)
```

EJ2)

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * template file and wish to re-enable it.
 */
package EJ2;

/**
 * Tarea: Crear un libro, intentar modificar el año con un valor inválido y luego con
 * uno válido, mostrar la información final.
 */
public class Mascota {

    String nombre;
    String especie;
    int edad;

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

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

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * template file and wish to re-enable it.
 */
package EJ2;

public class MainEj3 {

    public static void main(String[] args) {

        Mascota ml = new Mascota();
        ml.nombre = "Luna";
        ml.especie = "Gato";
        ml.edad = 2;

        ml.mostrarInfo();

        ml.cumplirAnios();
        ml.cumplirAnios();

        ml.mostrarInfo();
    }
}
```

EJ3)

The screenshot shows an IDE interface with two code editors and a terminal window.

Left Editor (Libro.java):

```
1 package Ej3;
2
3 public class Libro {
4     private String titulo;
5     private String autor;
6     private int anioPublicacion;
7
8     public Libro(String titulo, String autor, int anioPublicacion) {
9         this.titulo = titulo;
10        this.autor = autor;
11        setAnioPublicacion(anioPublicacion);
12    }
13
14    public String getTitulo() {
15        return titulo;
16    }
17
18    public String getAutor() {
19        return autor;
20    }
21
22    public void setAnioPublicacion(int anio) {
23        if (anio > 0 && anio <= 2025) {
24            this.anioPublicacion = anio;
25        } else {
26            System.out.println("Anio de publicacion invalido: " + anio);
27        }
28    }
29
30 }
```

Right Editor (MainEj3.java):

```
1 package Ej3;
2
3 /**
4 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/licenses-default.txt to change this
5 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
6 */
7
8 /*
9 * @author Carina Aranchet
10 */
11
12 public class MainEj3 {
13     public static void main(String[] args) {
14         Libro libro1 = new Libro("Libro 1", "Carina Aranchet", 1967);
15
16         libro1.mostrarInfo();
17
18         libro1.setAnioPublicacion(1500);
19
20         libro1.setAnioPublicacion(1985);
21
22         libro1.mostrarInfo();
23     }
24
25 }
```

Terminal Output:

```
UTN-TUPaD-P2 - C:\Users\Carina Aranchet\FACU\2do Semestre\Programación II\UTN-TUPaD-P2 > Tp3_Introducción_a_la_POO (run) >
EjLibro > mostrarInfo >
Titular: Libro 1
Autor: Carina Aranchet
Anio de publicacion: 1967
-----
Anio de publicacion invalido: -1500
Titular: Libro 1
Autor: Carina Aranchet
Anio de publicacion: 1985
-----
BUILD SUCCESSFUL (total time: 0 seconds)
```

EJ4)

The screenshot shows an IDE interface with two code editors and a terminal window.

Left Editor (Gallina.java):

```
1 package Ej4;
2
3 public class Gallina {
4     int idGallina;
5     int edad;
6     int huevosPuestos;
7
8     public void ponerHuevo(){
9         this.huevosPuestos++;
10    }
11
12    public void envejecer(){
13        this.edad++;
14    }
15
16    public void mostrarEstado() {
17        System.out.println("Gallina ID: " + idGallina);
18        System.out.println("Edad: " + edad + " años");
19        System.out.println("Huevos puestos: " + huevosPuestos);
20    }
21
22 }
```

Right Editor (MainEj4.java):

```
1 package Ej4;
2
3 public class MainEj4 {
4     public static void main(String[] args) {
5
6         Gallina g1 = new Gallina();
7         g1.idGallina = 1;
8         g1.edad = 2;
9         g1.huevosPuestos = 0;
10        g1.mostrarEstado();
11
12
13         Gallina g2 = new Gallina();
14         g2.idGallina = 2;
15         g2.edad = 1;
16         g2.huevosPuestos = 0;
17         g2.mostrarEstado();
18
19
20         g1.ponerHuevo();
21         g1.ponerHuevo();
22         g1.envejecer();
23
24         g2.ponerHuevo();
25         g2.envejecer();
26         g2.envejecer();
27
28         g1.mostrarEstado();
29         g2.mostrarEstado();
30
31     }
32
33 }
```

Terminal Output:

```
UTN-TUPaD-P2 - C:\Users\Carina Aranchet\FACU\2do Semestre\Programación II\UTN-TUPaD-P2 > Tp3_Introducción_a_la_POO (run) >
EjGallina > mostrarEstado >
Huevos puestos: 0
-----
Gallina ID: 1
Edad: 2 años
Huevos puestos: 0
-----
Gallina ID: 2
Edad: 3 años
Huevos puestos: 2
-----
Gallina ID: 2
Edad: 3 años
Huevos puestos: 1
-----
BUILD SUCCESSFUL (total time: 0 seconds)
```

EJ5)

```
Source History EJS NaveEspecial.java [-/A] x Gelfina.java [-/A] x MainEJava [-/A] x
1 package EJS;
2
3 /**
4  * Tareas: Crear una nave con 50 unidades de combustible, intentar avanzar sin
5  * recargar, luego recargar y avanzar correctamente. Mostrar el estado al final.
6  */
7
8 public class MainEJava {
9     public static void main(String[] args) {
10         NaveEspecial nave = new NaveEspecial(nombre: "Free Roal", combustibleInicial: 50);
11         nave.avanzar(distancia: 30);
12         nave.recargarCombustible(cantidad: 40);
13         nave.despegar();
14         nave.avanzar(distancia: 20);
15         nave.mostrarEstado();
16     }
17 }
18
19 package EJS;
20
21 /**
22  * Reglas: Valida que haya suficiente combustible antes de avanzar y evitar que
23  * se supere el límite al recargar
24  */
25
26 public class NaveEspecial {
27     private String nombre;
28     private int combustible;
29     private final int MAX_COMBUSTIBLE = 100; // límite máximo
30
31     public NaveEspecial(String nombre, int combustibleInicial) {
32         this.nombre = nombre;
33         this.combustible = combustibleInicial;
34     }
35
36     public void despegar() {
37         if (combustible >= 10) {
38             combustible -= 10 // gastar 10 en el despegue
39             System.out.println(nombre + " despegó (consumió 10 de combustible)");
40         } else {
41             System.out.println("No hay suficiente combustible para despegar.");
42         }
43     }
44
45     public void avanzar(int distancia) {
46         int consumo = distancia * 3; // unidades por cada distancia
47         if (combustible >= consumo) {
48             combustible -= consumo;
49             System.out.println(nombre + " avanza " + distancia + " km (consumió " + consumo + " de combustible)");
50         } else {
51             System.out.println("Combustible insuficiente para avanzar " + distancia + " km.");
52             System.out.println("Se recargaron 40 unidades de combustible.");
53             nave.despegar();
54             nave.avanzar(distancia);
55             nave.recargarCombustible(cantidad: 40);
56             nave.despegar();
57             nave.avanzar(distancia);
58         }
59     }
60
61     public void recargarCombustible(int cantidad) {
62         if (cantidad < 0) {
63             System.out.println("Cantidad de combustible no puede ser negativa.");
64         } else if (combustible + cantidad > MAX_COMBUSTIBLE) {
65             System.out.println("No se puede cargar más combustible. Límite máximo: " + MAX_COMBUSTIBLE);
66         } else {
67             combustible += cantidad;
68             System.out.println("Nave: " + nombre);
69             System.out.println("Combustible: " + combustible + "/" + MAX_COMBUSTIBLE);
70         }
71     }
72
73     public void mostrarEstado() {
74         System.out.println("Nave: " + nombre);
75         System.out.println("Combustible: " + combustible + "/" + MAX_COMBUSTIBLE);
76     }
77 }
```

Output X

```
UTN-TUPaD-P2 - C:\Users\Carina Aranchet\FACTU\2do Semestre\Programación I\UTN-TUPaD-P2 X | Tp3_Introducción_a_la_POO (run) X
run:
Nave: Free Roal
Combustible: 50/100
-----
Combustible insuficiente para avanzar 30 km.
Se recargaron 40 unidades de combustible.
Free Roal despegó (consumió 10 de combustible)
Free Roal avanza 20 km (consumió 40 de combustible).
Nave: Free Roal
Combustible: 40/100
-----
BUILD SUCCESSFUL (total time: 0 seconds)
```

LINK AL REPOSITORIO: https://github.com/carinaAranchet/UTN-TUPaD-P2/tree/main/Pr%C3%A1ctica/Tp3_ Introducción a la POO