

PROGRAMACIÓN II

Trabajo Práctico N.º 3: Introducción a POO

Estudiante: Emilia Gómez Juárez

Objetivo:

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.

Caso Práctico:

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

Ejercicio 1: Registro de estudiantes

- Código de creación de clase:

PROGRAMACIÓN II

```
11 public class Estudiante {
12
13     private String nombre;
14     private String apellido;
15     private String curso;
16     private double calificacion;
17
18     public Estudiante(String nombre, String apellido,
19         String curso, double calificacion) {
20
21         this.nombre = nombre;
22         this.apellido = apellido;
23         this.curso = curso;
24         this.calificacion = calificacion;
25     }
26
27     public String getNombre() {
28         return nombre;
29     }
30
31     public void setNombre(String nombre) {
32         if (apellido != null) {
33             this.nombre = nombre;
34         }
35     }
36
37     public String getApellido() {
38         return apellido;
39     }
```

PROGRAMACIÓN II

```
40
41 public void setApellido(String apellido) {
42     if (apellido != null) {
43         this.apellido = apellido;
44     }
45 }
46
47 public String getCurso() {
48     return curso;
49 }
50
51 public void setCurso(String curso) {
52     if (curso != null) {
53         this.curso = curso;
54     }
55 }
56
57 public void setCalificacion(double calificacion) {
58     if (calificacion < 0 && calificacion <= 10) {
59         this.calificacion = calificacion;
60     }
61 }
62
63 public double getCalificacion() {
64     return calificacion;
65 }
```

PROGRAMACIÓN II

```
66
67 public void subirCalificacion(double puntos) {
68     if (calificacion <= 10 && (calificacion + puntos) <= 10) {
69         calificacion = calificacion + puntos;
70     } else {
71         System.out.println("El estudiante ya tiene la nota máxima (10)");
72     }
73 }
74
75 public void bajarCalificacion(double puntos) {
76     if (calificacion >= 1 && (calificacion - puntos) >= 1) {
77         calificacion = calificacion - puntos;
78     } else {
79         System.out.println("El estudiante ya tiene la calificación "
80             + " mínima (1)");
81     }
82 }
83
84
85 public void mostrarInfo() {
86     System.out.println("Nombre : " + this.nombre +
87         "\nApellido: " + this.apellido + "\nCurso: " + this.curso +
88     );
89 }
90 }
```

- Código en Main:

```
case 1:
    Estudiante estudiante = new Estudiante ("Pedro", "Argerich",
        "Matemática", 10);
    estudiante.mostrarInfo();
    estudiante.bajarCalificacion(3);
    estudiante.mostrarInfo();
    estudiante.subirCalificacion(2);
    estudiante.mostrarInfo();
    break;
```

- Resultado en consola:

PROGRAMACIÓN II

```
Ingrese el ejercicio que desea comprobar (1 - 5). Si desea salir ingrese 0
1
Nombre :Pedro
Apellido: Argerich
Curso: Matemática
Calificación: 10.0
Nombre :Pedro
Apellido: Argerich
Curso: Matemática
Calificación: 7.0
Nombre :Pedro
Apellido: Argerich
Curso: Matemática
Calificación: 9.0
Ingrese el ejercicio que desea comprobar (1 - 5). Si desea salir ingrese 0
0
Gracias por probar mi código. Saludos!
BUILD SUCCESSFUL (total time: 10 seconds)
```

Ejercicio 2: Registro de mascotas

- Código de creación de clase:

PROGRAMACIÓN II

```
11 public class Mascota {
12     private String nombre;
13     private String especie;
14     private int edad;
15
16     public Mascota(String nombre, String especie, int edad) {
17         this.nombre = nombre;
18         this.especie = especie;
19         this.edad = edad;
20     }
21
22     public String getNombre() {
23         return nombre;
24     }
25
26     public void setNombre(String nombre) {
27         if (nombre != null){
28             this.nombre = nombre;
29         }
30     }
31
32     public String getEspecie() {
33         return especie;
34     }
35
36     public void setEspecie(String especie) {
37         this.especie = especie;
38     }
39 }
```

```
40     public int getEdad() {
41         return edad;
42     }
43
44     public void setEdad(int edad) {
45         if (edad < 0){
46             this.edad = edad;
47         }
48     }
49
50     public void mostrarInfo() {
51         System.out.println("\nNombre : " + this.nombre + "\nEspecie: "
52             + this.especie + "\nEdad: " + this.edad);
53     }
54
55     public void cumplirAños() {
56         this.edad += 1;
57     }
58 }
```

PROGRAMACIÓN II

- Código en Main:

```
34         case 2:
35             Mascota perro = new Mascota ("Inti", "Labrador", 9);
36             perro.mostrarInfo();
37             perro.cumplirAños();
38             perro.mostrarInfo();
39             break;
```

- Resultado en consola:

```
Ingrese el ejercicio que desea comprobar (1 - 5). Si desea salir ingrese 0
2

Nombre :Inti
Especie: Labrador
Edad: 9

Nombre :Inti
Especie: Labrador
Edad: 10
Ingrese el ejercicio que desea comprobar (1 - 5). Si desea salir ingrese 0
0
Gracias por probar mi código. Saludos!
BUILD SUCCESSFUL (total time: 9 seconds)
```

Ejercicio 3: Encapsulamiento con clase Libro

- Código de creación de clase:

PROGRAMACIÓN II

```
11 public class Libro {
12     private String titulo;
13     private String autor;
14     private int anioPublicacion;
15
16     public String getTitulo() {
17         return titulo;
18     }
19
20     public void setTitulo(String titulo) {
21         this.titulo = titulo;
22     }
23
24     public String getAutor() {
25         return autor;
26     }
27
28     public void setAutor(String autor) {
29         this.autor = autor;
30     }
31
32     public int getAnioPublicacion() {
33         return anioPublicacion;
34     }
35
36     public void setAnioPublicacion(int anioPublicacion) {
37         if (anioPublicacion > 0 && anioPublicacion <= 2025){
38             this.anioPublicacion = anioPublicacion;
39         }
40     }
```

- Código en Main:

PROGRAMACIÓN II

```
41     case 3:
42         Libro mobyDick = new Libro ();
43
44         mobyDick.setAutor("Herman Melville");
45         String autor = mobyDick.getAutor();
46
47         mobyDick.setTitulo("Moby Dick");
48         String titulo = mobyDick.getTitulo();
49
50         mobyDick.setAnioPublicacion(1851);
51         int anioPubli = mobyDick.getAnioPublicacion();
52
53         System.out.println("El libro " + titulo + " escrito por "
54             + autor + " fue publicado el año " + anioPubli);
55
56         mobyDick.setAnioPublicacion(2030);
57         //anioPubli = mobyDick.getAnioPublicacion();
58
59         System.out.println("El libro " + titulo + " escrito por "
60             + autor + " fue publicado el año " + anioPubli);
61
62         break;
```

- Resultado en consola:

```
run:
Ingrese el ejercicio que desea comprobar (1 - 5). Si desea salir ingrese 0
3
El libro Moby Dick escrito por Herman Melville fue publicado el año 1851
El libro Moby Dick escrito por Herman Melville fue publicado el año 1851
Ingrese el ejercicio que desea comprobar (1 - 5). Si desea salir ingrese 0
0
Gracias por probar mi código. Saludos!
BUILD SUCCESSFUL (total time: 5 seconds)
```

Ejercicio 4: Gestión de gallinas en Granja Digital

- Código de creación de clase:

PROGRAMACIÓN II

```
11 public class Gallina {
12     private int idGallina;
13     private int edad;
14     private int huevosPuestos;
15
16     public Gallina(int idGallina, int edad, int huevosPuestos) {
17         this.idGallina = idGallina;
18         this.edad = edad;
19         this.huevosPuestos = huevosPuestos;
20     }
21
22     public int getIdGallina() {
23         return idGallina;
24     }
25
26     public void setIdGallina(int idGallina) {
27         this.idGallina = idGallina;
28     }
29
30     public int getEdad() {
31         return edad;
32     }
33
34     public void setEdad(int edad) {
35         this.edad = edad;
36     }
37
38     public int getHuevosPuestos() {
39         return huevosPuestos;
40     }
}
```

PROGRAMACIÓN II

```
42 public void setHuevosPuestos(int huevosPuestos) {  
43     this.huevosPuestos = huevosPuestos;  
44 }  
45  
46 public void envejecer(int cantAnios) {  
47     this.edad += cantAnios;  
48 }  
49  
50 public void ponerHuevo(int cantHuevos) {  
51     this.huevosPuestos += cantHuevos;  
52 }  
53  
54 public void mostrarEstado() {  
55     System.out.println("ID Gallina: " + this.idGallina + "\nEdad act  
56  
57 }  
58  
59  
60 }  
61
```

- Código en Main:

```
62 case 4:  
63     Gallina gallinal = new Gallina (100, 5, 58);  
64     Gallina gallina2 = new Gallina (1002, 3, 35);  
65  
66     gallinal.ponerHuevo(10);  
67     gallinal.envejecer(2);  
68  
69     gallina2.envejecer(2);  
70     gallina2.ponerHuevo(30);  
71  
72     gallinal.mostrarEstado();  
73     gallina2.mostrarEstado();  
74  
75     break;
```

- Resultado en consola:

```
Ingrese el ejercicio que desea comprobar (1 - 5). Si desea salir ingrese 0  
4  
ID Gallina: 100  
Edad actual: 7  
Huevos puestos: 68  
ID Gallina: 1002  
Edad actual: 5  
Huevos puestos: 65
```

PROGRAMACIÓN II

Ejercicio 5: Simulación de Nave Espacial

- Código de creación de clase:

```
11 public class NaveEspacial {
12     private String nombre;
13     private int combustible;
14
15     public NaveEspacial(String nombre, int combustible) {
16         this.nombre = nombre;
17         this.combustible = combustible;
18     }
19
20     public String getNombre() {
21         return nombre;
22     }
23
24     public void setNombre(String nombre) {
25         this.nombre = nombre;
26     }
27
28     public double getCombustible() {
29         return combustible;
30     }
31
32     public void setCombustible(int combustible) {
33         this.combustible = combustible;
34     }
35
36     public void despegar(){
37         if (this.combustible >= 300){
38             System.out.println("En 3, 2, 1... Despegamos!!!");
39         }
40     }
41 }
```

PROGRAMACIÓN II

```
42 public void avanzar(int distancia){
43     if (this.combustible >= distancia){
44         System.out.println("Avanzamos " + distancia + " kilometros");
45         this.combustible -= distancia;
46         System.out.println("Te quedan " + this.combustible +
47             " litros de combustible disponibles");
48     } else {
49         System.out.println("La nave necesita recarga de combustible"
50             + " para avanzar. \nEl minimo son 500 litros");
51     }
52 }
53
54 public void recargarCombustible (int cantidad){
55     if ((this.combustible + cantidad) <= 1500){
56         this.combustible += cantidad;
57     } else {
58         int cantPosible = 1500 - this.combustible;
59         System.out.println("Solo pudimos cargar " + cantPosible +
60             " litros. Tanque lleno!");
61
62         this.combustible += cantPosible;
63     }
64 }
65
66 public void mostrarEstado(){
67     System.out.println("Nombre de la nave: " + this.nombre
68         + "\nCombustible: " + this.combustible);
69 }
70
```

- Código en Main:

```
case 5:
    NaveEspacial navel = new NaveEspacial ("XL-15", 50);
    navel.despegar();
    navel.avanzar(1000);
    navel.recargarCombustible(1555);
    navel.avanzar(1000);
    navel.mostrarEstado();

    break;
```

- Resultado en consola:

PROGRAMACIÓN II

```
run:
Ingrese el ejercicio que desea comprobar (1 - 5). Si desea salir ingrese 0
5
La nave necesita recarga de combustible para avanzar.
El minimo son 500 litros
Solo pudimos cargar 1450 litros. Tanque lleno!
Avanzamos 1000 kilometros
Te quedan 500 litros de combustible disponibles
Nombre de la nave: XL-15
Combustible: 500
Ingrese el ejercicio que desea comprobar (1 - 5). Si desea salir ingrese 0
0
Gracias por probar mi código. Saludos!
BUILD SUCCESSFUL (total time: 7 seconds)
|
```

REPOSITORIO **REMOTO:** <https://github.com/GomezJEmilia/UTN-Programacion2-TPs-EmiliaGJ.git>