



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

# Laboratorio de Computación Salas A y B

*Profesor(a):* Ramírez Romero, Diego

*Asignatura:* Programación Orientada a Objetos

*Grupo:* 06

*No de Práctica(s):* 05 Abstracción y encapsulamiento

*Integrante(s):* Mendoza Camacho, Estrella de Maria

*No. de lista o  
brigada:* 23

*Semestre:* 2025 - 1

*Fecha de entrega:* 20 – Septiembre – 2024

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

**Objetivo:** Aplicar el concepto de abstracción para el diseño de clases que integran una solución, utilizando el encapsulamiento para proteger la información y ocultar la implementación.

**Actividades:**

- Obtener las características y funcionalidades principales de un objeto dado.
- Utilizar diferentes niveles de acceso a las características y funcionalidades obtenidas.

**Ejercicio 1:** Codificar el código de la página 85, crear las tres clases siguientes: Rectángulo, Trapecio, Rombo y Pentágono “crear sus atributos y métodos (iguales a los de círculo)”. Consideraciones: los atributos son ingresados por el usuario. Realizar las siguientes restricciones:

- Verificar que el radio sea diferente de cero, si es igual o menos a cero imprimir “Ingresa un valor correcto para el radio”, el programa seguirá solicitando un valor valido para el radio.
- Verificar que la base sea mayor que altura. “Mensaje a su elección”
- Verificar que la altura del trapecio sea diferente de cero. “Mensaje a su elección”.
- Se usarán todos los métodos de cada figura.

```

C:\Users\Estrella\Desktop\Practica 1\POO\Exercice 1\practicaFigurasJava - Soluion Test (IntelliJ IDE)
File Edit Selection Find View Code Tools Project Preferences Help

practicaFigurasJava
1  /*Nombre: Mendoza Camacho, Estrella de Maria
2  Programa perteneciente a la quinta práctica de Programación Orientada a Objetos:
3  "Abstracción y Encapsulamiento", programa figuras.
4  Fecha: 20 De Septiembre del 2024
5  */
6
7
8  import java.util.Scanner;
9
10 // Círculo
11 class Círculo {
12     static float PI = 3.14159f;
13     private float radio;
14     private static Scanner sc = new Scanner(System.in);
15
16     public Círculo() { // Constructor
17         System.out.println("Se creó un círculo");
18     }
19
20     public float getRadio() { // Leer el radio
21         return radio;
22     }
23
24     public void setRadio(float radio) { // Modificar el radio
25         while (radio <= 0) { // Verificación
26             System.out.println("Ingresa un valor correcto para el radio (mayor que 0:");
27             radio = sc.nextFloat();
28         }
29         this.radio = radio;
30     }
31
32     public float perimetro() { // Método perimetro
33         return 2 * PI * radio;
34     }
35
36     public float area() { // Método area
37         return PI * radio * radio;
38     }
39
40     public String toString() { // Impresión de datos
41         return "Círculo [radio = " + radio + "];";
42     }
43 }
44
45 // Rectángulo
```

```
pruebaFiguras.java
43 }
44
45 // Rectangulo
46
47 class Rectangulo {
48     private float base;
49     private float altura;
50     private static Scanner sc = new Scanner(System.in);
51
52     public Rectangulo() { // Constructor
53         System.out.println("Se creó un rectángulo");
54     }
55
56     public void setBaseAltura(float base, float altura) { // Verificación de base mayor que altura
57         while (base <= altura) {
58             System.out.println("La base debe ser mayor que la altura. Ingrese nuevamente los valores:");
59             base = sc.nextFloat();
60             altura = sc.nextFloat();
61         }
62         this.base = base;
63         this.altura = altura;
64     }
65
66     public float area() {
67         return base * altura;
68     }
69
70     public float perimetro() {
71         return 2 * (base + altura);
72     }
73
74     public String toString() {
75         return "Rectángulo [base = " + base + ", altura = " + altura + "]";
76     }
77 }
78
```

```
pruebaFiguras.java
76 }
77 }
78
79 // Trapecio
80
81 class Trapecio {
82     private float baseMayor;
83     private float baseMenor;
84     private float altura;
85     private static Scanner sc = new Scanner(System.in);
86
87     public Trapecio() {
88         System.out.println("Se creó un trapecio");
89     }
90
91     public void setBaseMayor(float baseMayor) {
92         this.baseMayor = baseMayor;
93     }
94
95     public void setBaseMenor(float baseMenor) {
96         this.baseMenor = baseMenor;
97     }
98
99     public void setAltura(float altura) { // Verificación altura mayor a 0
100         while (altura <= 0) {
101             System.out.println("Ingresa un valor correcto para la altura (mayor que 0):");
102             altura = sc.nextFloat();
103         }
104         this.altura = altura;
105     }
106
107     public float area() {
108         return ((baseMayor + baseMenor) * altura) / 2;
109     }
110
111     public float perimetro() { // Fórmula para el perimetro (teorema de Pitagoras para obtener las diagonales)
112         float diagonal = (float) Math.sqrt(Math.pow((baseMayor - baseMenor) / 2, 2) + Math.pow(altura, 2));
113         return baseMayor + baseMenor + (2 * diagonal);
114     }
115
116     public String toString() {
117         return "Trapecio [base mayor = " + baseMayor + ", base menor = " + baseMenor + ", altura = " + altura + "]";
118     }
119 }
120
```

```
121 // Rombo
122
123 class Rombo {
124     private float diagonalMayor;
125     private float diagonalMenor;
126
127     public Rombo() {
128         System.out.println("Se creó un rombo");
129     }
130
131     public void setDiagonalMayor(float diagonalMayor) {
132         this.diagonalMayor = diagonalMayor;
133     }
134
135     public void setDiagonalMenor(float diagonalMenor) {
136         this.diagonalMenor = diagonalMenor;
137     }
138
139     public float area() { // Área
140         return (diagonalMayor * diagonalMenor) / 2;
141     }
142
143     public float perimetro() { // Fórmula para obtener las diagonales (lados)
144         float lado = (float) Math.sqrt(Math.pow(diagonalMayor / 2, 2) + Math.pow(diagonalMenor / 2, 2));
145         return 4 * lado; // Suma de lados
146     }
147
148     public String toString() {
149         return "Rombo [diagonal mayor = " + diagonalMayor + ", diagonal menor = " + diagonalMenor + "]\n";
150     }
151 }
152
153 // Pentagono
154
155 class Pentagono {
156     private float lado;
157
158     public Pentagono() {
159         System.out.println("Se creó un pentágono");
160     }
161
162     public void setLado(float lado) {
163         this.lado = lado;
164     }
165 }
```

```
153 // Pentagono
154
155 class Pentagono {
156     private float lado;
157
158     public Pentagono() {
159         System.out.println("Se creó un pentágono");
160     }
161
162     public void setLado(float lado) {
163         this.lado = lado;
164     }
165
166     public float area() { // Fórmula para obtener el area de un pentagono
167         return (float) (Math.sqrt(5 * (5 + 2 * Math.sqrt(5))) / 4) * lado * lado;
168     }
169
170     public float perimetro() {
171         return 5 * lado;
172     }
173
174     public String toString() {
175         return "Pentágono [lado = " + lado + "]\n";
176     }
177 }
```

```

C:\Users\Estrella\Desktop\Práctica 5 POO\Ejercicio 1\pruebaFiguras.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

177 }
178
179 // Clase principal
180 public class pruebaFiguras {
181     public static void main(String[] args) {
182         Scanner sc = new Scanner(System.in);
183
184         // Circulo prueba
185
186         Circulo cir = new Circulo();
187         System.out.println("Ingresa el radio del círculo:");
188         cir.setRadio(sc.nextFloat());
189         System.out.println(cir);
190         System.out.println("Perímetro: " + cir.perimetro());
191         System.out.println("Área: " + cir.area());
192
193         // Rectangulo prueba
194         Rectangulo rectangulo = new Rectangulo();
195         System.out.println("Ingresa la base y la altura del rectángulo:");
196         rectangulo.setBaseAltura(sc.nextFloat(), sc.nextFloat());
197         System.out.println(rectangulo);
198         System.out.println("Perímetro: " + rectangulo.perimetro());
199         System.out.println("Área: " + rectangulo.area());
200
201         // Trapecio
202         Trapecio trapecio = new Trapecio();
203         System.out.println("Ingresa la base mayor, base menor y altura del trapecio:");
204         trapecio.setBaseMayor(sc.nextFloat());
205         trapecio.setBaseMenor(sc.nextFloat());
206         trapecio.setAltura(sc.nextFloat());
207         System.out.println(trapecio);
208         System.out.println("Perímetro: " + trapecio.perimetro());
209         System.out.println("Área: " + trapecio.area());
210
211         // Rombo
212         Rombo rombo = new Rombo();
213         System.out.println("Ingresa la diagonal mayor y la diagonal menor del rombo:");
214         rombo.setDiagonalMayor(sc.nextFloat());
215         rombo.setDiagonalMenor(sc.nextFloat());
216         System.out.println(rombo);
217         System.out.println("Perímetro: " + rombo.perimetro());
218         System.out.println("Área: " + rombo.area());
219
220         // Pentagono
221         Pentagono pentagono = new Pentagono();
222         System.out.println("Ingresa el lado del pentágono:");
223         pentagono.setLado(sc.nextFloat());
224         System.out.println(pentagono);
225         System.out.println("Perímetro: " + pentagono.perimetro());
226         System.out.println("Área: " + pentagono.area());
227
228         sc.close(); // Cerrar el scann
229     }
230 }
231

```

Corrida en terminal:

```

C:\Users\Estrella\Desktop\Práctica 5 POO\Ejercicio 1>javac pruebaFiguras.java
C:\Users\Estrella\Desktop\Práctica 5 POO\Ejercicio 1>java pruebaFiguras
Se creó un círculo
Ingresa el radio del círculo:
9
Ingresa un valor correcto para el radio (mayor que 0):
9
Círculo [radio = 9.0]
Perímetro: 56.548622
Área: 254.4688
Se creó un rectángulo
Ingresa la base y la altura del rectángulo:
9
18
La base debe ser mayor que la altura. Ingresa nuevamente los valores:
18
9
Rectángulo [base = 18.0, altura = 9.0]
Perímetro: 54.0
Área: 162.0
Se creó un trapecio
Ingresa la base mayor, base menor y altura del trapecio:
27
18
0
Ingresa un valor correcto para la altura (mayor que 0):
27
Trapecio [base mayor = 27.0, base menor = 18.0, altura = 27.0]
Perímetro: 99.74486
Área: 607.5
Se creó un rombo
Ingresa la diagonal mayor y la diagonal menor del rombo:
18
9
Rombo [diagonal mayor = 18.0, diagonal menor = 9.0]
Perímetro: 40.24922
Área: 81.0
Se creó un pentágono
Ingresa el lado del pentágono:
9
Pentágono [lado = 9.0]

```

**Ejercicio 2:** Codificar el código de la página 89. Como evidencia de esta actividad insertar una imagen de tu código y corrida en terminal aquí.:

```
C:\Users\Estrella\Desktop\PruebaPersona.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

1  /*Nombre: Mendoza Camacho, Estrella de Maria
2  Programa perteneciente a la quinta práctica de Programación Orientada a Objetos:
3  "Abstracción y Encapsulamiento", programa persona / fechas.
4  Fecha: 20 De Septiembre del 2024
5  */
6
7  class Fecha {
8      private int dia;
9      private int mes;
10     private int anio;
11
12     // Constructor por defecto
13     public Fecha() {
14         // Inicialización vacía
15     }
16
17     // Constructor con parámetros
18     public Fecha(int dia, int mes, int anio) {
19         setDia(dia);
20         setMes(mes);
21         setAnio(anio);
22     }
23
24     // Método para obtener el día
25     public int getDia() {
26         return dia;
27     }
28
29     // Método para establecer el día con verificación de rango
30     public void setDia(int dia) {
31         if (dia > 0 && dia < 32) {
32             this.dia = dia;
33         } else {
34             System.out.println("Día no válido");
35         }
36     }
37
38     // Método para obtener el mes
39     public int getMes() {
40         return mes;
41     }
42
43     // Método para establecer el mes con verificación de rango
44     public void setMes(int mes) {
45         if (mes > 0 && mes < 13) {
46             this.mes = mes;
47         } else {
48             System.out.println("Mes no válido");
49         }
50     }
51 }
```

```
C:\Users\Estrella\Desktop\PruebaPersona.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

52 // Método para obtener el año
53 public int getAnio() {
54     return anio;
55 }
56
57 // Método para establecer el año con verificación de rango
58 public void setAnio(int anio) {
59     if (anio > 0) {
60         this.anio = anio;
61     } else {
62         System.out.println("El año no puede ser negativo");
63     }
64 }
65
66 // Método para representar la fecha como una cadena
67 public String toString() {
68     return dia + "/" + mes + "/" + anio;
69 }
70 }
71
72 class Persona {
73     private String nombre;
74     private String apellido;
75     private Fecha fNacimiento;
76
77     // Constructor por defecto
78     public Persona() {
79         fNacimiento = new Fecha(); // Inicialización de la fecha de nacimiento
80     }
81
82     // Constructor con parámetros
83     public Persona(String nombre, String apellido, int fDia, int fMes, int fAnio) {
84         this.nombre = nombre;
85         this.apellido = apellido;
86         fNacimiento = new Fecha(fDia, fMes, fAnio); // Inicialización de la fecha de nacimiento con valores
87     }
88
89     // Método para obtener el nombre
90     public String getNombre() {
91         return nombre;
92     }
93
94     // Método para establecer el nombre
95     public void setNombre(String nombre) {
96         this.nombre = nombre;
97     }
98
99     // Método para obtener el apellido
100    public String getApellido() {
101        return apellido;
102    }
103 }
```

```
C:\Users\Estrella\Desktop\PruebaPersona.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

pruebaFiguras.java x PruebaPersona.java x

104 // Método para establecer el apellido
105 public void setApellido(String apellido) {
106     this.apellido = apellido;
107 }
108
109 // Método para establecer la fecha de nacimiento
110 public void setFNacimiento(int dia, int mes, int anio) {
111     fNacimiento.setDia(dia);
112     fNacimiento.setMes(mes);
113     fNacimiento.setAnio(anio);
114 }
115
116 // Método para obtener la fecha de nacimiento
117 public Fecha getFNacimiento() {
118     return fNacimiento;
119 }
120 }
121
122 public class PruebaPersona {
123     public static void main(String[] args) {
124         // Creación de una instancia de Persona
125         Persona per1 = new Persona();
126         per1.setNombre("Juan");
127         per1.setApellido("Perez");
128         per1.setFNacimiento(15, 8, 1950);
129
130         // Impresión de los datos de la persona
131         System.out.println("Nombre: " + per1.getNombre());
132         System.out.println("Apellido: " + per1.getApellido());
133         System.out.println("Fecha Nacimiento: " + per1.getFNacimiento());
134     }
135 }
136
```

Corrida en terminal:

```
Microsoft Windows [Versión 10.0.19045.4894]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Estrella>cd C:\Users\Estrella\Desktop\Práctica 5 P00\Ejercicio 2

C:\Users\Estrella\Desktop\Práctica 5 P00\Ejercicio 2>javac PruebaPersona.java

C:\Users\Estrella\Desktop\Práctica 5 P00\Ejercicio 2>java PruebaPersona
Nombre: Juan
Apellido: Perez
Fecha Nacimiento: 15/8/1950

C:\Users\Estrella\Desktop\Práctica 5 P00\Ejercicio 2>
```

**Conclusiones:** La implementación, con ayuda de los códigos de la página, fue sencilla.

Lo que quizá pudo tener su complejidad fue hacer verificaciones, pues en un inicio las empleé en el main, para después cambiarlas directamente a los constructores.

Me agradó la práctica y considero que el objetivo de aplicar el concepto de abstracción para el diseño de clases que integran una solución, utilizando el encapsulamiento para proteger la información y ocultar la implementación se cumplió, pues tenía ideas vagas, pero implementarlo hizo que comprendiera mejor el concepto.