



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Dra. Rocío Alejandra Aldeco Pérez

Asignatura: Programación orientada a objetos -1323

Grupo: 6

No de Práctica(s): 2

Integrante(s): Mendoza Hernández Carlos Emiliano

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada:

Semestre: 2023-1

Fecha de entrega: 2 de septiembre del 2022

Observaciones:

CALIFICACIÓN: _____



Práctica 2.

Fundamentos y sintaxis del lenguaje.

OBJETIVO

- Crear programas que implementen variables y constantes de diferentes tipos de datos, expresiones y estructuras de control de flujo.

ACTIVIDADES

- Crear variables y constantes de diferentes tipos de datos.
- Crear diversas expresiones (operadores, declaraciones, etc.)
- Implementar estructuras de control de flujo (if/else, switch, while, etc.)

INTRODUCCIÓN

Los lenguajes de programación tienen elementos básicos que se utilizan como bloques constructivos, así como reglas para que esos elementos se combinen. Estas reglas se denominan **sintaxis del lenguaje**. Solamente las instrucciones sintácticamente correctas pueden ser interpretadas por la computadora y los programas que contengan errores de sintaxis son rechazados por la máquina.

La **sintaxis** de un lenguaje de programación se define como el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación.



PROGRAMACIÓN ORIENTADA A OBJETOS



Realiza las siguientes actividades después de leer y revisar en clase la *Práctica de Estudio 2: Fundamentos y sintaxis del lenguaje*.

Crea un programa en java que **calcule el perímetro y el área de las figuras geométricas básicas**. Para esto deberás crear una clase por figura para posteriormente instanciarlas en tu clase principal.

Las figuras son:

- Triángulo.
- Cuadrado.
- Rectángulo.
- Rombo.
- Romboide.
- Trapecio
- Polígono Regular.
- Círculo.

1. En este espacio describe que atributos y métodos tendrá cada una de estas clases. Recuerda incluir su constructor y los métodos para calcular perímetro y área. Posteriormente genera el correspondiente código en java de estas clases.



PROGRAMACIÓN ORIENTADA A OBJETOS



Clase	Atributos	Métodos
Cuadrado	<ul style="list-style-type: none">• <i>lado</i>	<ul style="list-style-type: none">• <i>Cuadrado()</i>• <i>perimetro()</i>• <i>area()</i>
Circulo	<ul style="list-style-type: none">• <i>PI</i>• <i>radio</i>	<ul style="list-style-type: none">• <i>Circulo()</i>• <i>perimetro()</i>• <i>area()</i>
Poligono	<ul style="list-style-type: none">• <i>numLados</i>• <i>lado</i>• <i>apotema</i>	<ul style="list-style-type: none">• <i>Poligono()</i>• <i>perimetro()</i>• <i>area()</i>
Rectangulo	<ul style="list-style-type: none">• <i>base</i>• <i>altura</i>	<ul style="list-style-type: none">• <i>Rectangulo()</i>• <i>perimetro()</i>• <i>area()</i>
Rombo	<ul style="list-style-type: none">• <i>diagonalMayor</i>• <i>diagonalMenor</i>• <i>lado</i>	<ul style="list-style-type: none">• <i>Rombo()</i>• <i>perimetro()</i>• <i>area()</i>
Romboide	<ul style="list-style-type: none">• <i>base</i>• <i>altura</i>• <i>lado</i>	<ul style="list-style-type: none">• <i>Romboide()</i>• <i>perimetro()</i>• <i>area()</i>
Trapecio	<ul style="list-style-type: none">• <i>baseMayor</i>• <i>baseMenor</i>• <i>altura</i>• <i>lado</i>	<ul style="list-style-type: none">• <i>Trapecio()</i>• <i>perimetro()</i>• <i>area()</i>
Triángulo	<ul style="list-style-type: none">• <i>lado1</i>• <i>lado2</i>• <i>base</i>• <i>altura</i>	<ul style="list-style-type: none">• <i>Triangulo()</i>• <i>perimetro()</i>• <i>area()</i>



PROGRAMACIÓN ORIENTADA A OBJETOS



2. Genera tu clase principal en base a la información proporcionada por Alphagrader.

Código de la clase principal:

```
1  import java.util.Scanner;
2
3  public class Main {
4      /**
5       * @param args
6       */
7      public static void main(String[] args) {
8          int numLados;
9          double radio;
10         double apotema = 0;
11         double lado = 0;
12         double lado1 = 0, lado2 = 0, base = 0, altura = 0;
13         double diagonalMayor = 0, diagonalMenor = 0;
14         double baseMayor = 0, baseMenor = 0;
15         String letra;
```

Variables que servirán para llamar a los constructores de las clases y, *letra*, para guardar la opción del usuario.

```
17         Scanner stdin = new Scanner(System.in);
18         letra = stdin.next();
19         switch (letra) {
20             case "S":
21                 lado = stdin.nextDouble();
22                 Cuadrado unCuadrado = new Cuadrado(lado);
23                 System.out.println("PERIMETRO: " + unCuadrado.perimetro());
24                 System.out.println("AREA: " + unCuadrado.area());
25                 break;
26
27             case "T":
28                 lado1 = stdin.nextDouble();
29                 lado2 = stdin.nextDouble();
30                 base = stdin.nextDouble();
31                 altura = stdin.nextDouble();
32                 Triangulo unTriangulo = new Triangulo(lado1, lado2, base, altura);
33                 System.out.println("PERIMETRO: " + unTriangulo.perimetro());
34                 System.out.println("AREA: " + (int)unTriangulo.area());
35                 break;
36
37             case "R":
38                 base = stdin.nextDouble();
39                 altura = stdin.nextDouble();
40                 Rectangulo unRectangulo = new Rectangulo(base, altura);
41                 System.out.println("PERIMETRO: " + unRectangulo.perimetro());
42                 System.out.println("AREA: " + unRectangulo.area());
43                 break;
44         }
```

Estructura de control *switch*.

Se lee(n) la(s) variable(s) necesaria(s) para ser usada(s) como parámetro(s) del constructor.

Se instancia un objeto de la clase usando el constructor que se creó

Llamando a sus métodos para calcular el perímetro y el área

Caso *T* (triángulo)

Caso *R* (rectángulo)



PROGRAMACIÓN ORIENTADA A OBJETOS



```
Main.java > Main > main(String[])
45     case "A":
46         lado = stdin.nextDouble();
47         diagonalMayor = stdin.nextDouble();
48         diagonalMenor = stdin.nextDouble();
49         Rombo unRombo = new Rombo(lado, diagonalMayor, diagonalMenor);
50         System.out.println("PERIMETRO: " + unRombo.perimetro());
51         System.out.println("AREA: " + String.format(format "%.1f", unRombo.area()));
52         break;
53
54     case "B":
55         base = stdin.nextDouble();
56         altura = stdin.nextDouble();
57         lado = stdin.nextDouble();
58         Romboide unRomboide = new Romboide(base, altura, lado);
59         System.out.println("PERIMETRO: " + (int)unRomboide.perimetro());
60         System.out.println("AREA: " + (int)unRomboide.area());
61         break;
62
63     case "D":
64         baseMayor = stdin.nextDouble();
65         baseMenor = stdin.nextDouble();
66         altura = stdin.nextDouble();
67         lado = stdin.nextDouble();
68         Trapecio unTrapecio = new Trapecio(baseMayor, baseMenor, altura, lado);
69         System.out.println("PERIMETRO: " + unTrapecio.perimetro());
70         System.out.println("AREA: " + (int)unTrapecio.area());
71         break;
72
```

Caso A (rombo)

Caso B (romboide)

Caso D (trapecio)

```
Main.java > Main > main(String[])
73     case "P":
74         numLados = stdin.nextInt();
75         lado = stdin.nextDouble();
76         apotema = stdin.nextDouble();
77         Poligono unPoligono = new Poligono(numLados, lado, apotema);
78         System.out.println("PERIMETRO: " + (int)unPoligono.perimetro());
79         System.out.println("AREA: " + unPoligono.area());
80         break;
81
82     case "C":
83         radio = stdin.nextDouble();
84         Circulo unCirculo = new Circulo(radio);
85         System.out.println("PERIMETRO: " + String.format(format "%.1f", unCirculo.perimetro()));
86         System.out.println("AREA: " + Math.floor(unCirculo.area()*100)/100);
87         break;
88
89     default:
90         System.out.println("Letra invalida");
91         break;
92 }
93 stdin.close();
94 }
95 }
96
```

Caso P (poligono)

Caso C (círculo)

Caso por default

3. Cuando estés seguro de que tu programa es correcto, súbelo a Alphagrader.
4. En este reporte incluye tu carátula y tus conclusiones.



CONCLUSIONES

En esta práctica se desarrolló un programa que permitió el uso y comprensión de los siguientes conceptos:

- Tipos de datos; tales como *double*, *int* y *String*. Se utilizaron para guardar, principalmente, las dimensiones de las figuras.
- Operadores aritméticos y declaraciones de variables.
- Estructuras de control. Como ejemplo, el *switch* que se implementó en el menú principal.
- Abstracción y codificación de clases.
- Instanciación por medio de un constructor.
- Llamado a métodos de diferentes instancias.

El uso correcto de la sintaxis del lenguaje no solo es importante, sino necesario para que las instrucciones puedan ser interpretadas por la computadora.

Considero que se cubrieron todos los aspectos relevantes de sintaxis de Java tales como tipos de datos, variables, constantes, entrada y salida de datos por consola, operadores y estructuras de control, por lo tanto, se cumplieron los objetivos de la práctica.



REFERENCIAS

- *Martín, Antonio*
Programador Certificado Java 2.
Segunda Edición.
México
Alfaomega Grupo Editor, 2008
- *Sierra Katy, Bate Bert*
SCJP Sun Certified Programmer for Java 6 Study Guide.
McGrawHill
- *Dean John, Dean Raymond*
Introducción a la programación con Java.
Primera Edición.
México
McGrawHill, 2009

Yo, Carlos Emiliano Mendoza Hernández, hago mención que esta práctica fue de mi autoría.