

Reporte Practica 1

Karla Romina Juárez Torres

318013712

1. Respuestas a las Preguntas

a) Principales complicaciones al realizar la práctica

Generalmente los problemas que tuve fueron menores, como la construcción de algunas formulas y con la nueva estructura de las carpetas me pude manejar mejor conforme a como estaba usando los métodos.

b) Herencia en Java y su uso en esta práctica

La herencia se utiliza para crear una nueva clase que sea un tipo de una clase existente. Ayuda a ampliar la funcionalidad de una clase sin modificarla, respetando así el Principio Abierto/Cerrado del diseño de software.

Concretamente solo usé la herencia para que la clase `PoligonoRegular` se amplie a cuadrado, triángulo y pentagono:

Se heredan los atributos: `numeroLados`, `longitudLado`, `tipoFigura`, además de los métodos para calcular perimetro, sin embargo el calculo de área no es dependiente

c) Definición formal de los algoritmos

Algoritmo `calcularPerimetro()` - `PoligonoRegular`

Algorithm 1 Calcular perímetro de polígono regular

Require: Polígono regular p con $p.numeroLados \in \mathbb{Z}^+$ y $p.longitudLado \in \mathbb{R}^+$

Ensure: $perimetro \in \mathbb{R}^+$

1: $perimetro \leftarrow p.numeroLados \times p.longitudLado$

2: **return** $perimetro$

Algoritmo `calcularArea()` - Cuadrado

Algorithm 2 Calcular área de cuadrado

Require: Cuadrado c con $c.longitudLado \in \mathbb{R}^+$

Ensure: $area \in \mathbb{R}^+$

1: $area \leftarrow c.longitudLado \times c.longitudLado$

2: **return** $area$

Algoritmo `calcularArea()` - `TrianguloEquilatero`

Algorithm 3 Calcular área de triángulo equilátero

Require: Triángulo t con $t.longitudLado \in \mathbb{R}^+$

Ensure: $area \in \mathbb{R}^+$

1: $area \leftarrow \frac{\sqrt{3}}{4} \times t.longitudLado \times t.longitudLado$

2: **return** $area$

Algoritmo calcularArea() - Pentagono

Algorithm 4 Calcular área de pentágono regular

Require: Pentágono p con $p.longitudLado \in \mathbb{R}^+$ y $p.numeroLados = 5$

Ensure: $area \in \mathbb{R}^+$

- 1: $perimetro \leftarrow p.numeroLados \times p.longitudLado$
 - 2: $apotema \leftarrow \frac{p.longitudLado}{2 \times \tan(\pi/p.numeroLados)}$
 - 3: $area \leftarrow \frac{perimetro \times apotema}{2}$
 - 4: **return** $area$
-

Algoritmo calcularArea() - Circulo

Algorithm 5 Calcular área de círculo

Require: Círculo c con $c.radio \in \mathbb{R}^+$ y $c.PI \approx 3,1416$

Ensure: $area \in \mathbb{R}^+$

- 1: $area \leftarrow c.PI \times c.radio \times c.radio$
 - 2: **return** $area$
-

Algoritmo calcularPerimetro() - Circulo

Algorithm 6 Calcular perímetro de círculo

Require: Círculo c con $c.radio \in \mathbb{R}^+$ y $c.PI \approx 3,1416$

Ensure: $perimetro \in \mathbb{R}^+$

- 1: $perimetro \leftarrow 2 \times c.PI \times c.radio$
 - 2: **return** $perimetro$
-

2. Preguntas adicionales

a) ¿Por qué sería una mala decisión de diseño que la clase `Circulo` extendiera a la clase abstracta `PoligonoRegular`?

Principalmente por que el concepto de circulo para aproximarse por poligono regular requiere una mayor cantidad de tiempo de computo para obtener una aproximación apenas cercana al usar un valor de π ya establecido, además que las aproximaciones con cambios sumamente pequeños causan errores numericos.

b) ¿Cuáles son las ventajas de que exista la interfaz `Figura`?

Principalmente por que nos permite tratar a los métodos de manera uniforme donde tambien requiere que se mantengan métodos obligatorios, esto garantiza que podamos añadir nuevos métodos para el calculo de nuevos tipos de figura es decir mantiene escalabilidad.

3. Respuestas IA

Consulté al LLM [Gemini](#) con el enlace a la conversación directo en el hipervinculo, donde suministré el codigo en su totalidad.

```
El código proporcionado muestra una jerarquía de clases para calcular el área y el perímetro de di
---
```

```
### ¿Por qué sería una mala decisión de diseño que la clase 'Circulo' extienda a la clase abstracta 'P
```

```
Sería una mala decisión de diseño que la clase 'Circulo' extienda a la clase abstracta 'PoligonoRegula
```

Además, si 'Circulo' heredara de 'PoligonoRegular', tendría que heredar sus atributos ('numeroLados' y

¿Cuáles son las ventajas de que exista la interfaz 'Figura'?

La principal ventaja de tener la interfaz 'Figura' es que proporciona una **abstracción común** y un *

Aquí hay algunas ventajas clave:

* **Polimorfismo**: La interfaz 'Figura' permite escribir código que puede trabajar con cualquier obje

* **Cohesión y desacoplamiento**: La interfaz desacopla la implementación de las figuras de su uso. El

* **Extensibilidad**: Si en el futuro se quiere añadir una nueva figura, como un hexágono, solo se nec

Concretamente siento que dio una respuesta más formal sin embargo siento que se omite el hecho que la clase circulo puede ser distinta y no usa el concepto limite ni sobre todo la existencia de métodos numéricos.

4. Análisis de Complejidad Computacional de los Algoritmos

Desglose Detallado por Clase

Polígono Regular

- `calcularPerimetro()`: $\text{numeroLados} \times \text{longitudLado}$
1 operación: 1 multiplicación

Cuadrado

- `calcularArea()`: $\text{longitudLado} \times \text{longitudLado}$
1 operación: 1 multiplicación

Triángulo Equilátero

- `calcularArea()`: $\frac{\sqrt{3}}{4} \times \text{longitudLado} \times \text{longitudLado}$
4 operaciones:
 1. 1 cálculo de $\sqrt{3}$ (función matemática)
 2. 1 división ($/4$)
 3. 2 multiplicaciones ($\times \text{longitudLado}$ dos veces)

Pentágono

- `calcularApotema()`: $\frac{\text{longitudLado}}{2 \times \tan(\frac{\pi}{5})}$
5 operaciones:
 1. 1 división ($180^\circ/5$)
 2. 1 conversión a radianes (`Math.toRadians`)
 3. 1 cálculo de tangente (`Math.tan`)
 4. 1 multiplicación ($2 \times \tan(\dots)$)
 5. 1 división ($\text{longitudLado}/\dots$)
- `calcularArea()`: $\frac{\text{perimetro} \times \text{apotema}}{2}$
8 operaciones:

1. 5 operaciones de `calcularApotema()`
2. 1 operación de `calcularPerimetro()` ($5 \times longitudLado$)
3. 1 multiplicación ($perimetro \times apotema$)
4. 1 división ($/2$)

Círculo

- `calcularArea()`: $\pi \times radio \times radio$
2 operaciones: 2 multiplicaciones
- `calcularPerimetro()`: $2 \times \pi \times radio$
2 operaciones: 2 multiplicaciones

Operaciones Elementales por Figura

Cuadro 1: Número de operaciones elementales por método

Figura	Método	Operaciones
Polígono Regular	<code>calcularPerimetro</code>	1
Cuadrado	<code>calcularArea</code>	1
Triángulo Equilátero	<code>calcularArea</code>	4
Pentágono	<code>calcularApotema</code>	5
Pentágono	<code>calcularArea</code>	8
Círculo	<code>calcularArea</code>	2
Círculo	<code>calcularPerimetro</code>	2

Todas las implementaciones tienen complejidad constante $O(1)$.