

# UNIVERSIDAD TECMILENIO

Mauricio Estrada De la Garza  
AL02976904  
12/10/2023  
Programación Orientada a Objetos  
Reto 7

Reto 7: Implementación de los conceptos fundamentales de Programación Orientada a Objetos, **incluyendo Herencia y excepciones**

## Objetivo de la actividad:

El alumno desarrollará un programa en el cual se implemente los conceptos:

1. Creación de clases. (x)
2. Implementación de Clases usando objetos. (x)
3. Creación y definición de clases abstractas. (x)
4. Implementación de clases abstractas usando herencia. (x)

## Requerimientos para la actividad:

Uso Java jdk a través de un IDE, preferentemente Netbeans. ☕

**Instrucciones para el alumno:**

**Creación de una aplicación donde apliques todos los conceptos**

**Descripción:** Desarrollar una aplicación en Java que permita gestionar el cálculo de al menos cinco figuras geométricas

**Instrucciones:**

1. Define y crea una clase abstracta que contenga la siguiente definición:
  1. **Dos atributos privados.**
  2. **Dos métodos getter.**
  3. **Dos métodos setter.**
  4. **Dos métodos abstractos.**
2. Define Cinco clases diferentes que implementen las clases abstractas definidas en el punto anterior. (X)
3. **Genera un menú** donde se indique cual es la figura que se pretende escoger. (x)

## Conceptos a calificar:

1. **Creación y definición de clases abstractas. (X)**
2. **Definición de métodos abstractos. (X)**
3. **Implementación de clases abstractas mediante herencia (X)**
4. **Implementación de constructores. (X)**
5. **Implementación de métodos abstractos. (X)**
6. **Funcionalidad del programa. (X)**

  Brownie points por correcta definición de los casos de prueba    

6. **Reflexionar:** Discutir y reflexionar con diversos ejemplos de uso de objetos y posibles implementaciones de clases abstractas y la su relación con el concepto de herencia.

# Entregables:

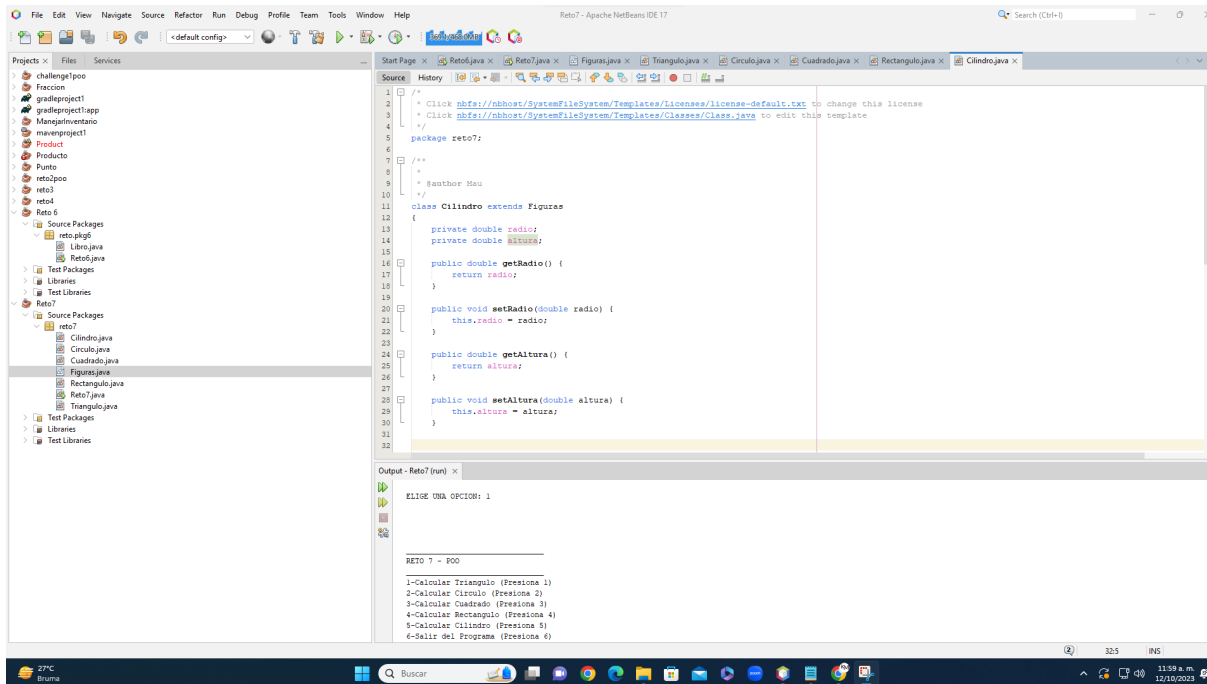
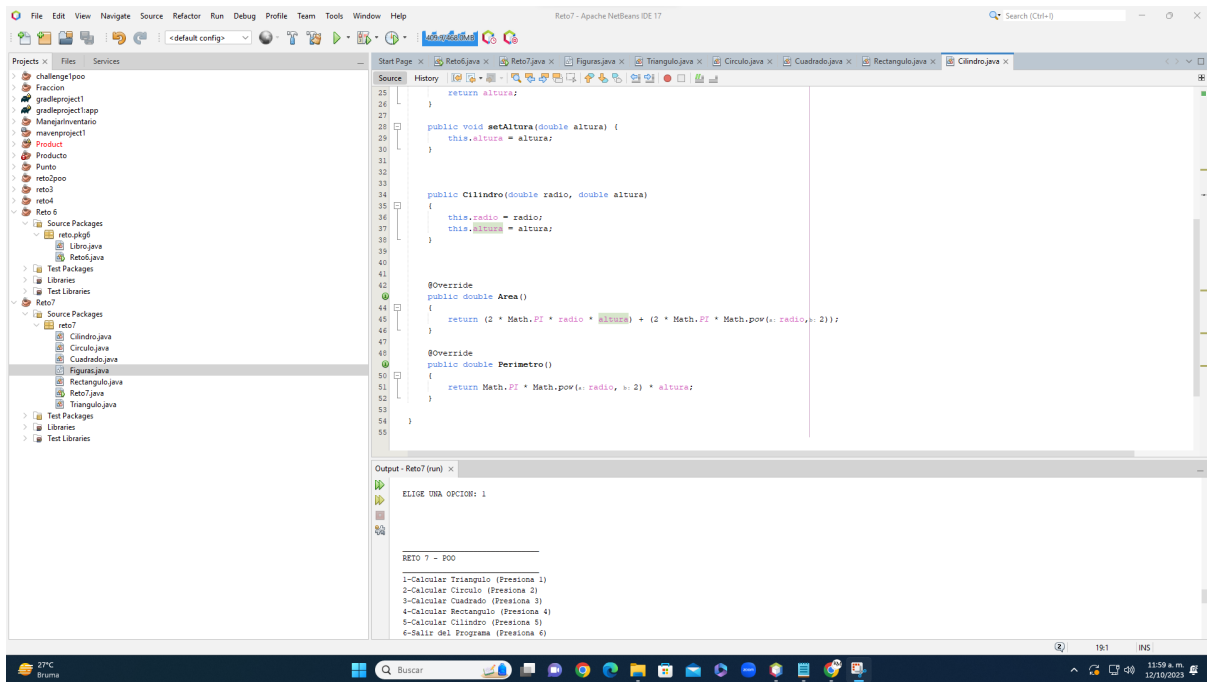
- Código Fuente, archivos con extensión .java (Recuerda, cualquier otro tipo de archivo no es permitido, i.e. .class, .zip, .jar, etc)
- Compilación sin errores.
- Cumplimiento de Especificaciones.
- 🏗️ Reflexión individual por integrante del equipo. 🏗️

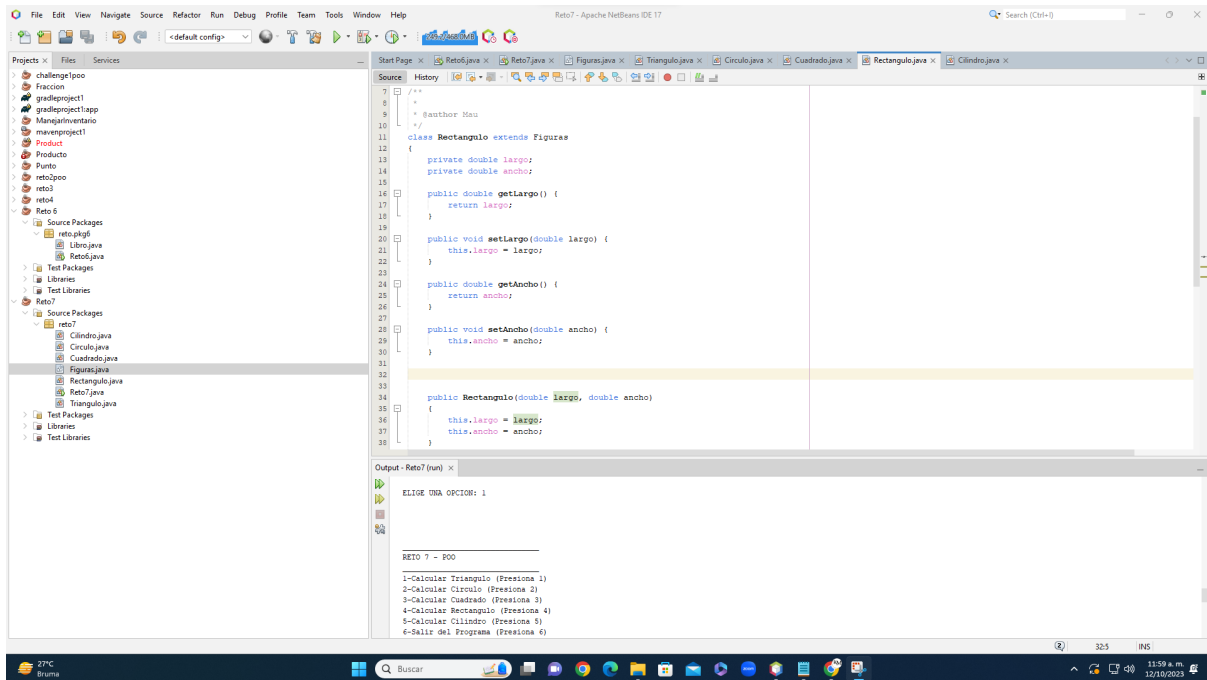
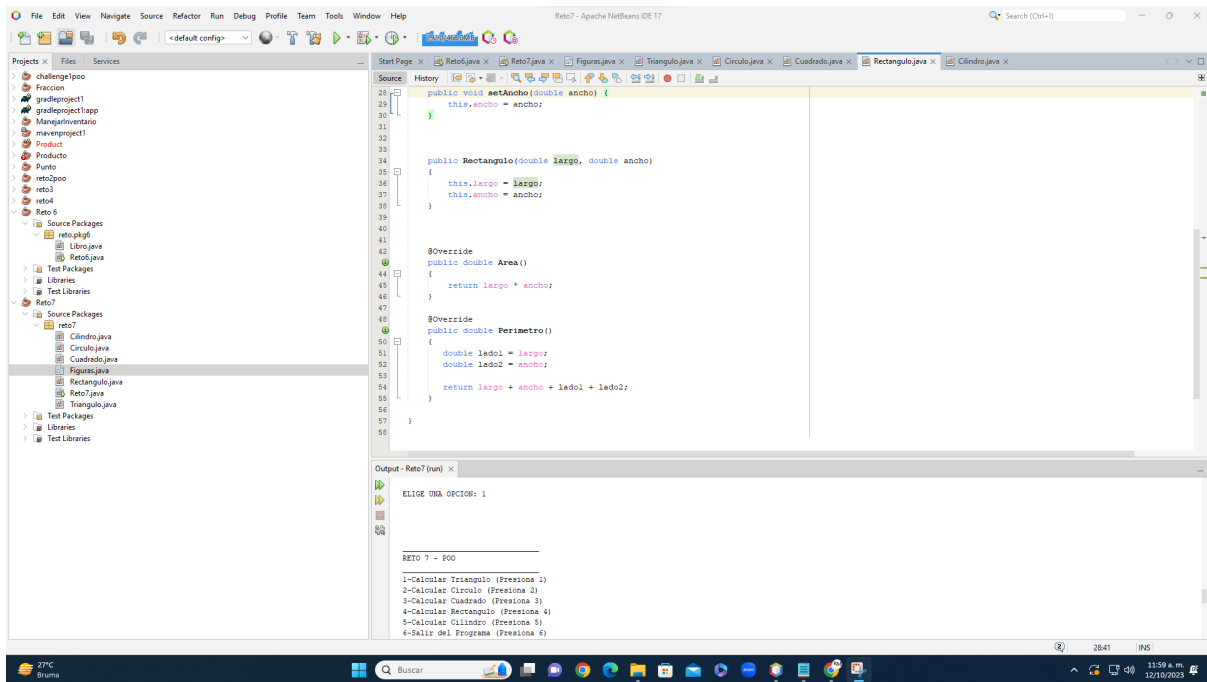
**Importante:** Solo se revisa la última entrega

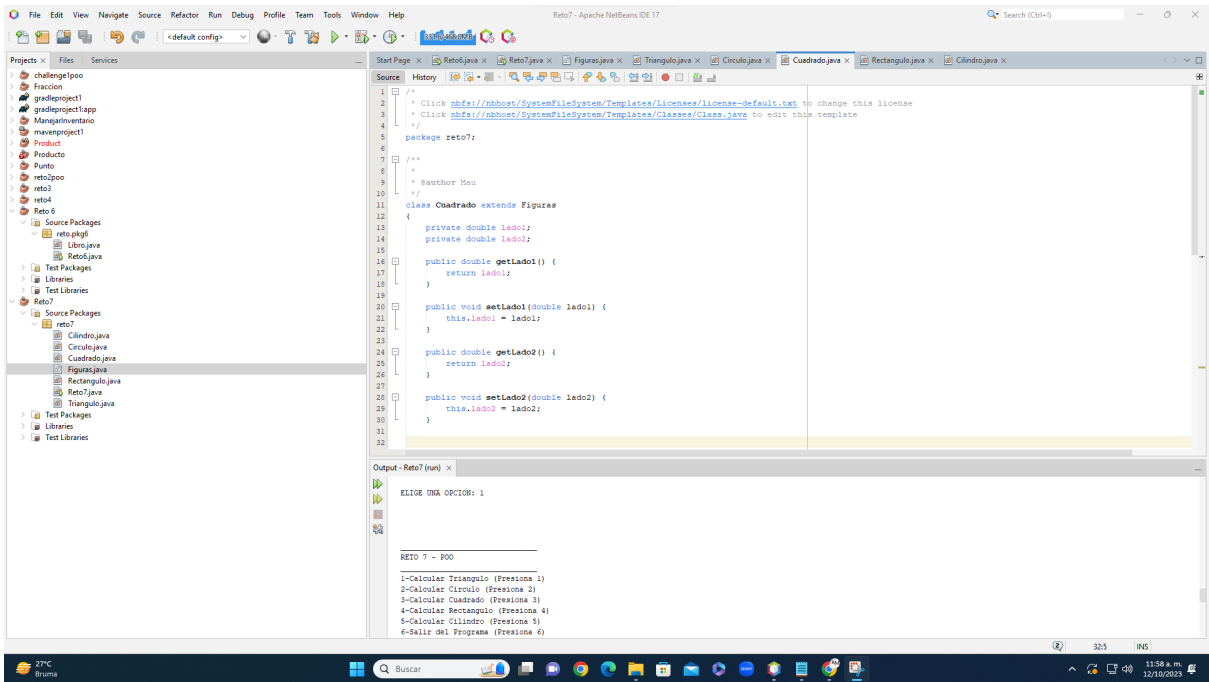
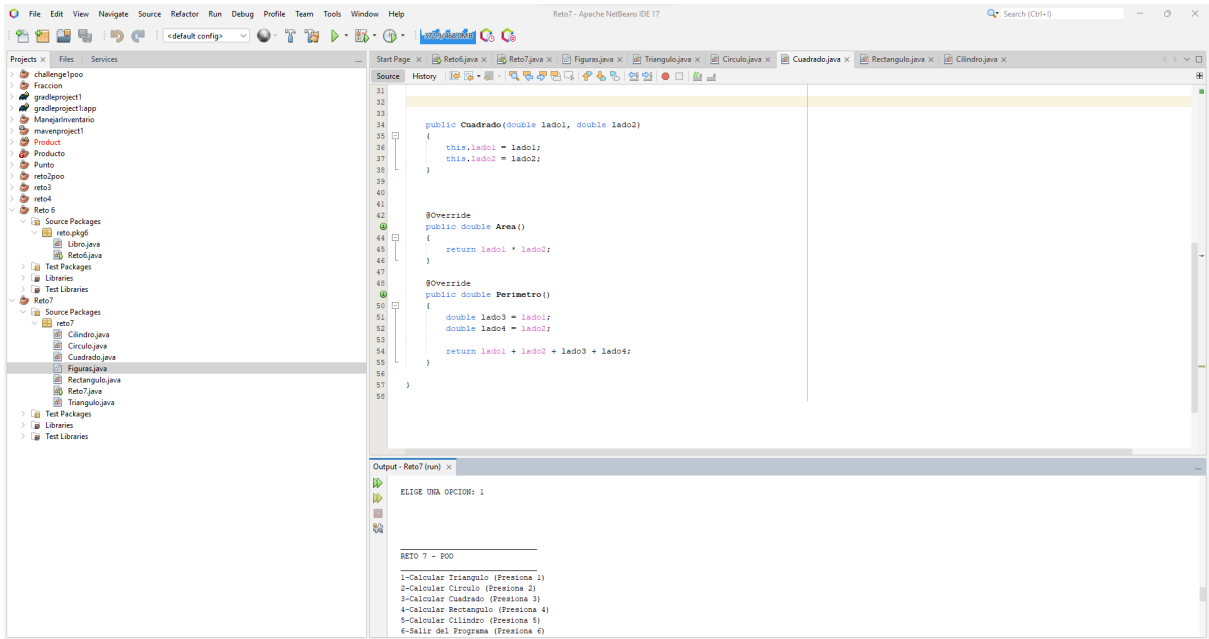
\* Sin reporte y/o código fuente, ningún porcentaje es considerado ⇒ Calificación NE

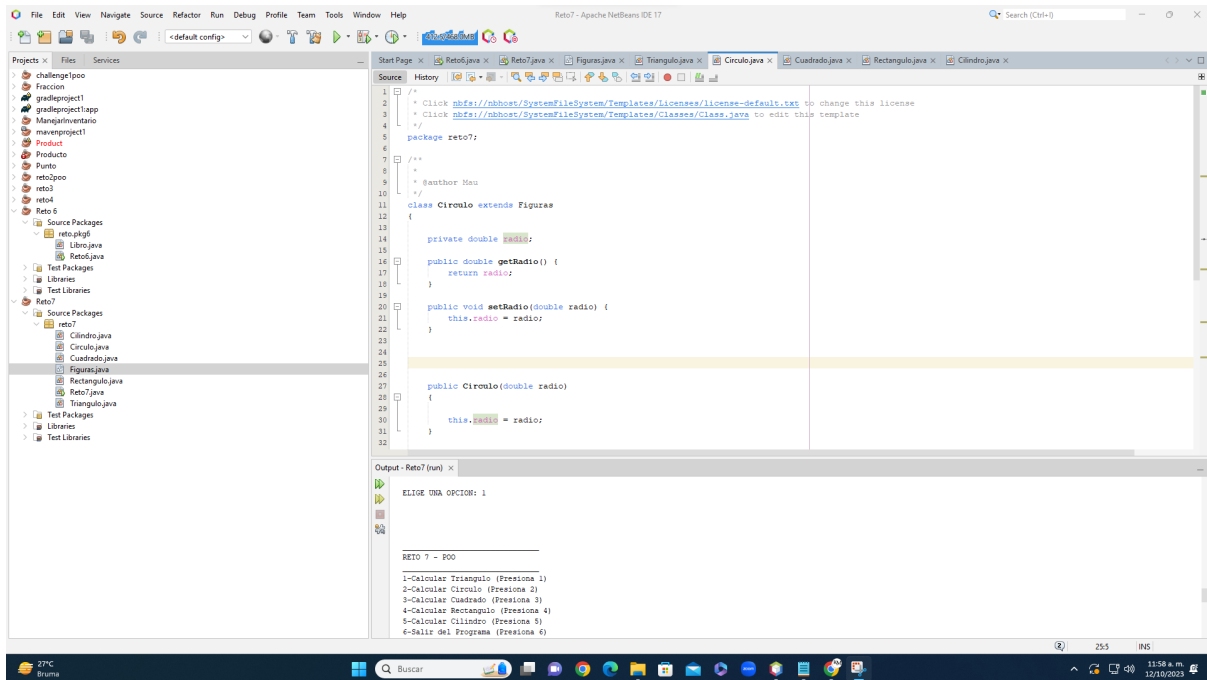
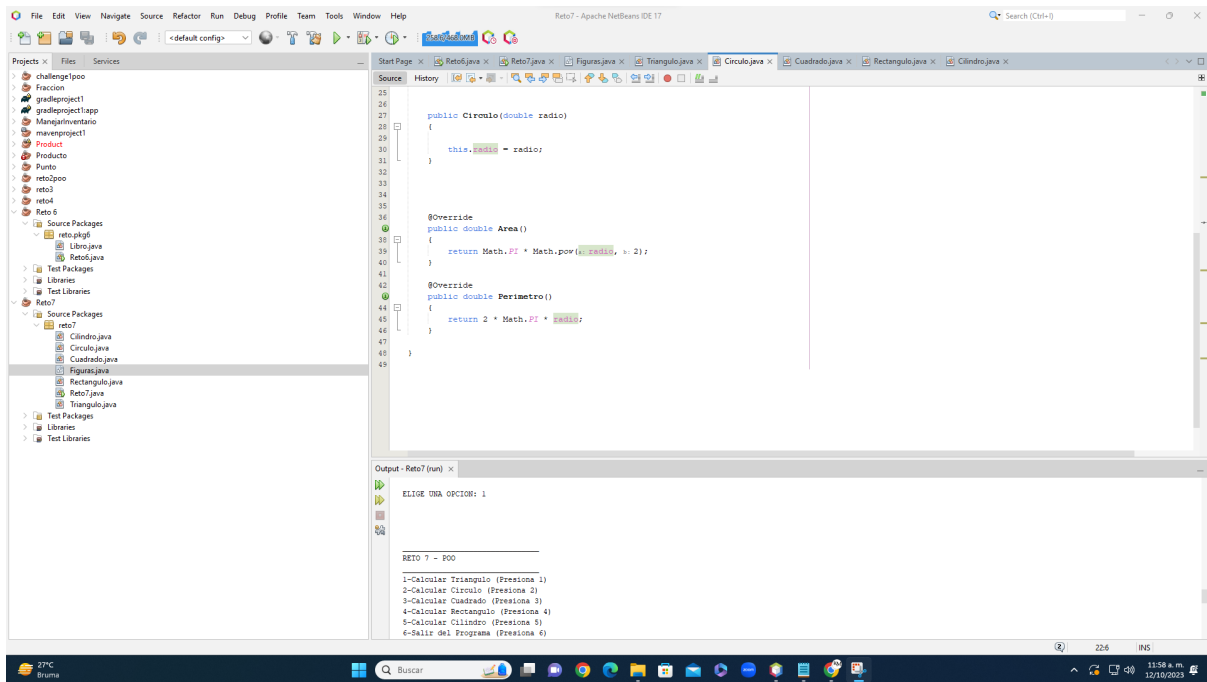
**Restricción:** Equipos de no más de tres personas, no menos de dos. 😊

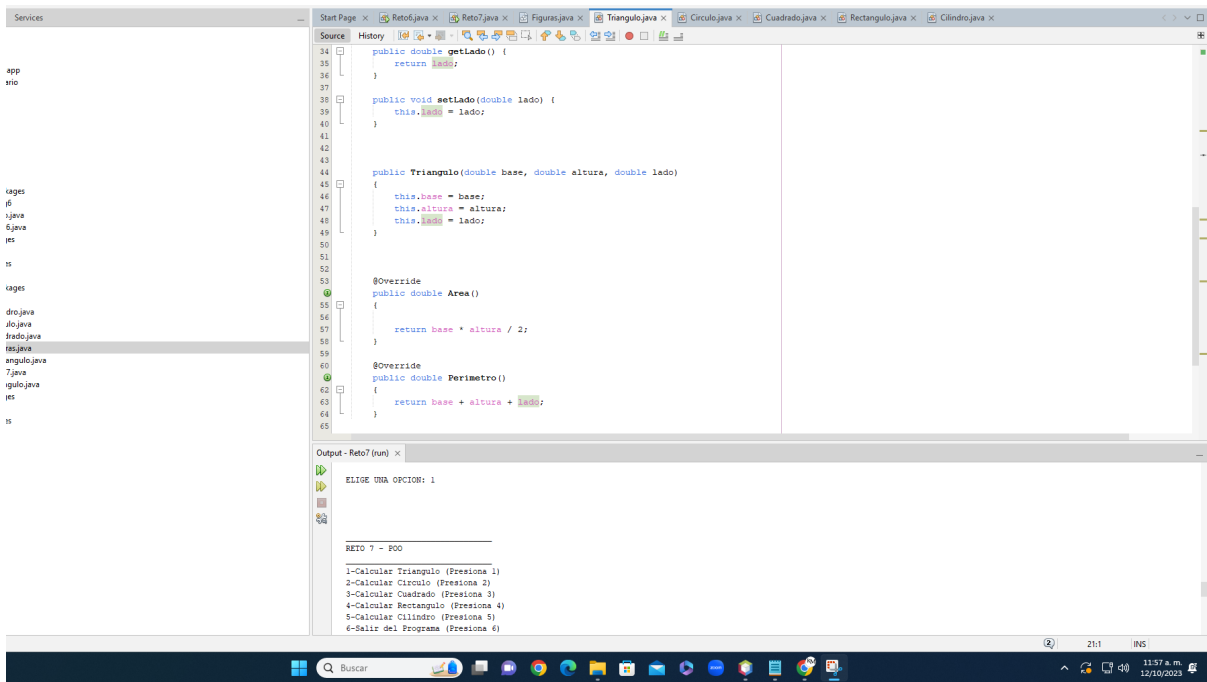
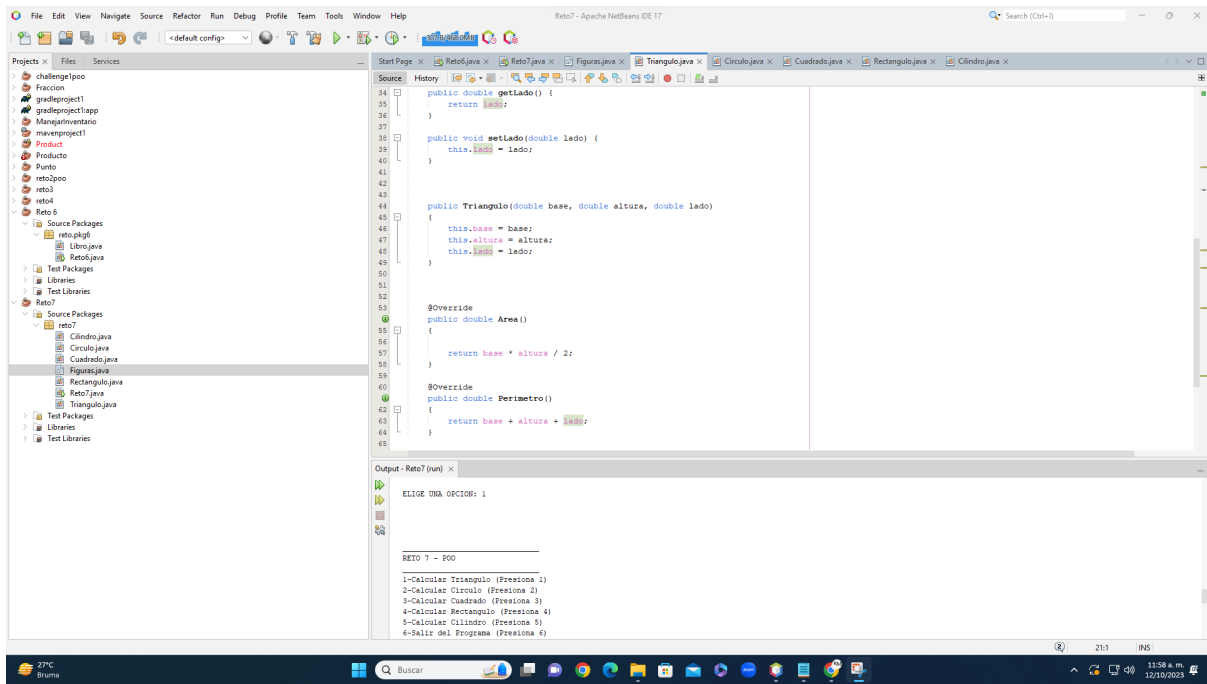
**CÓDIGO:**



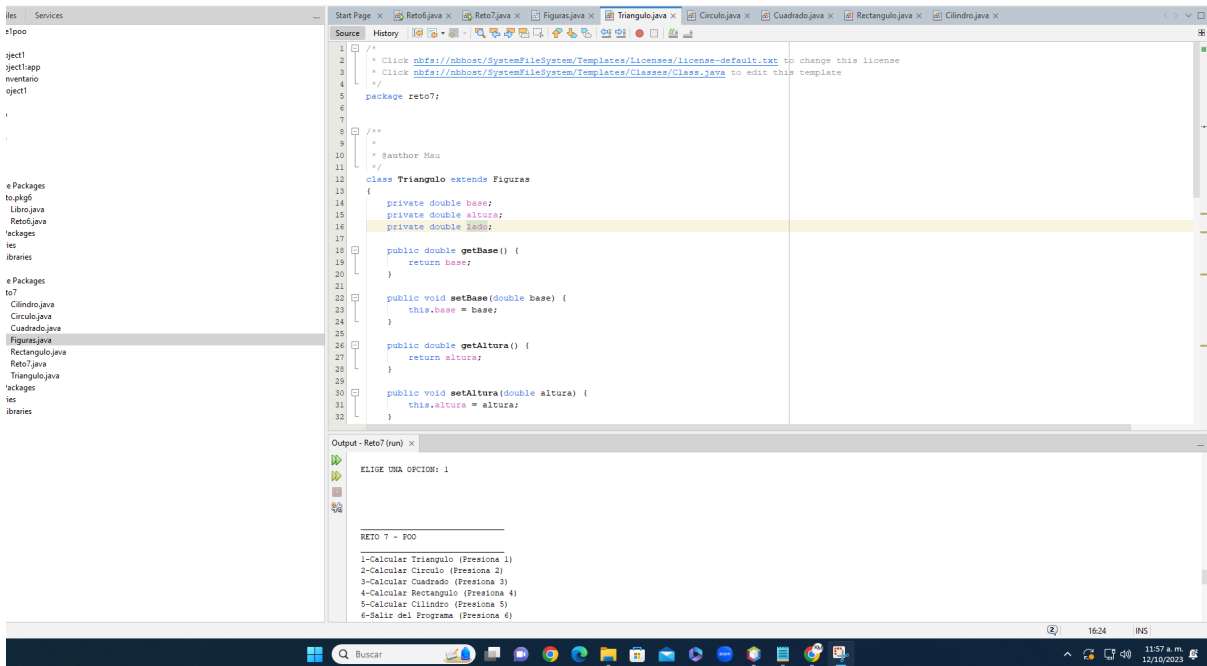
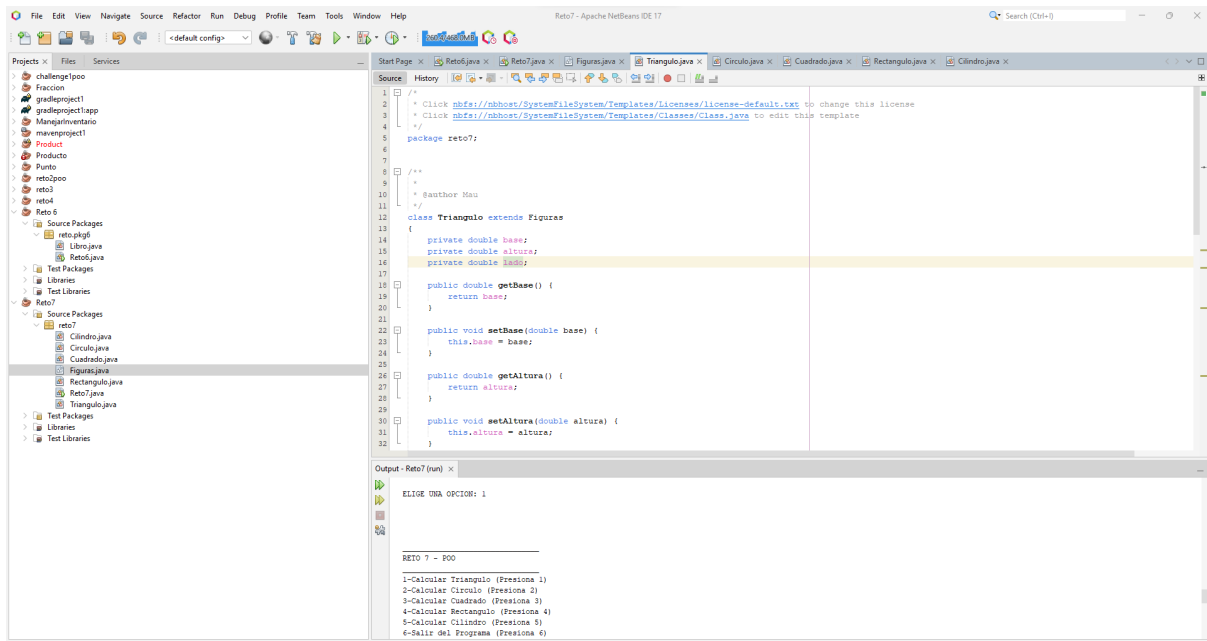


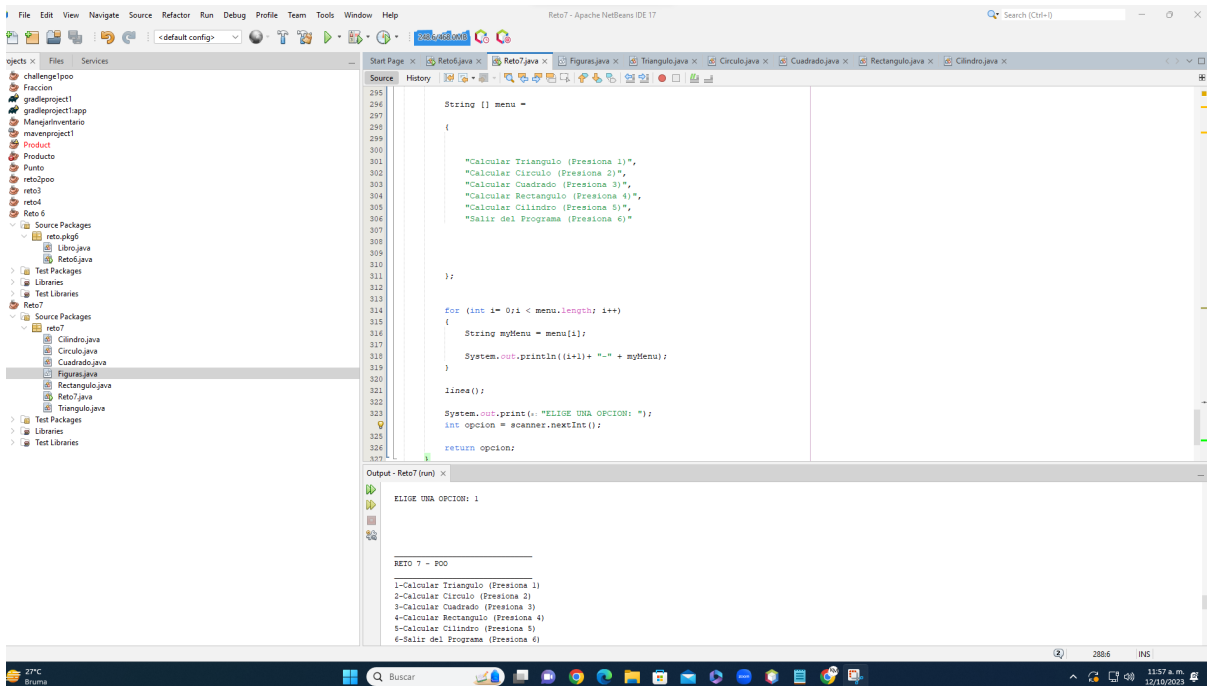
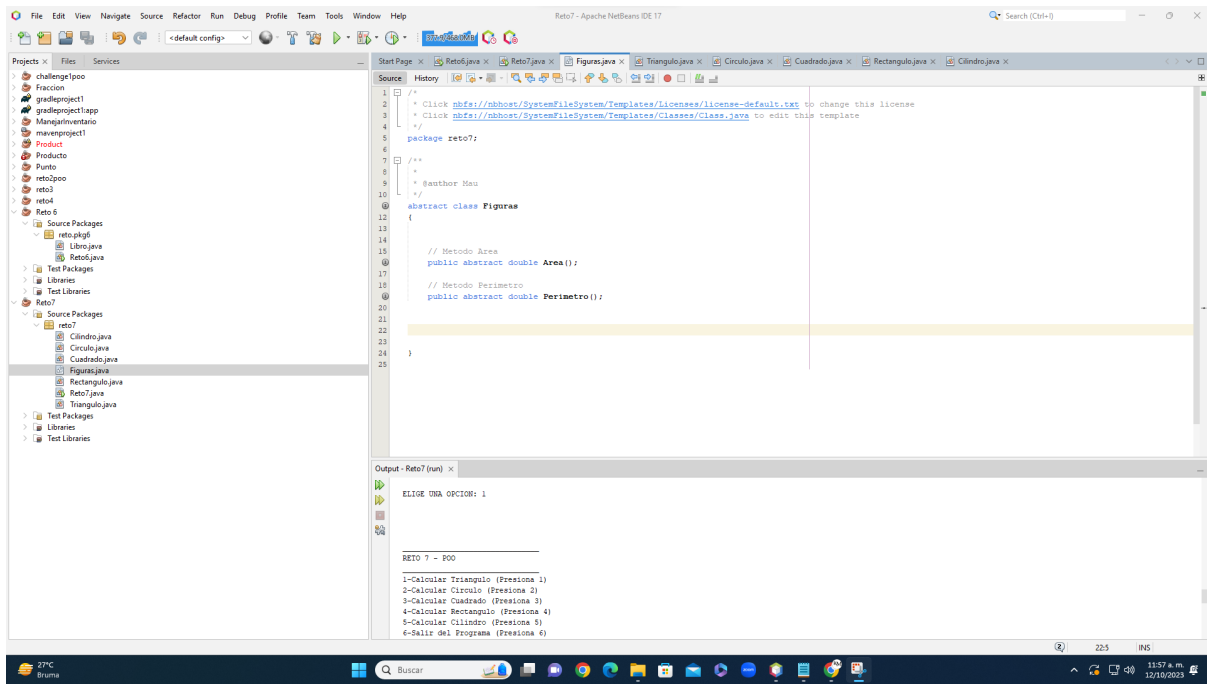


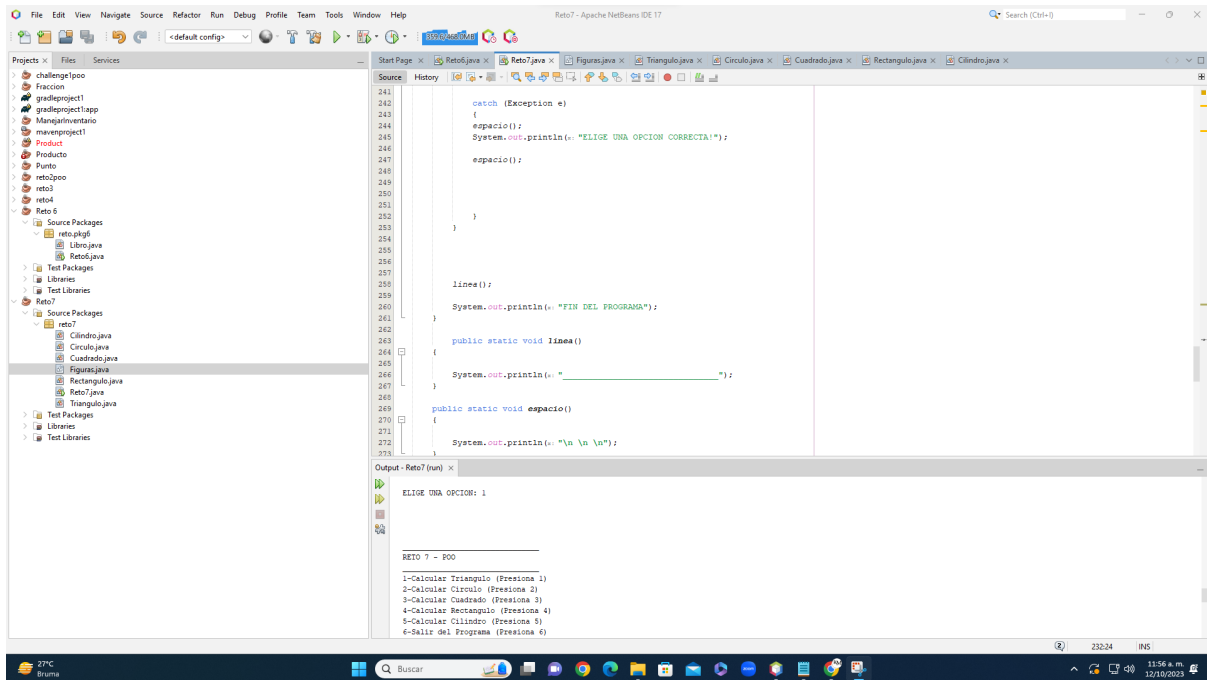
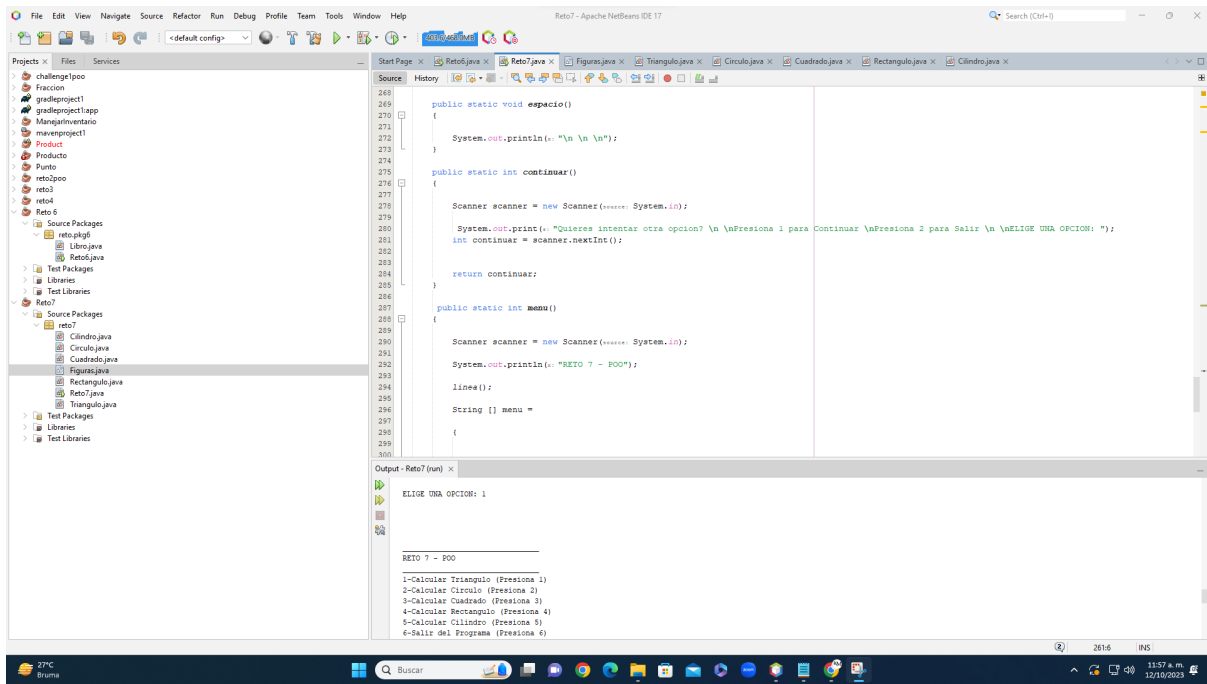


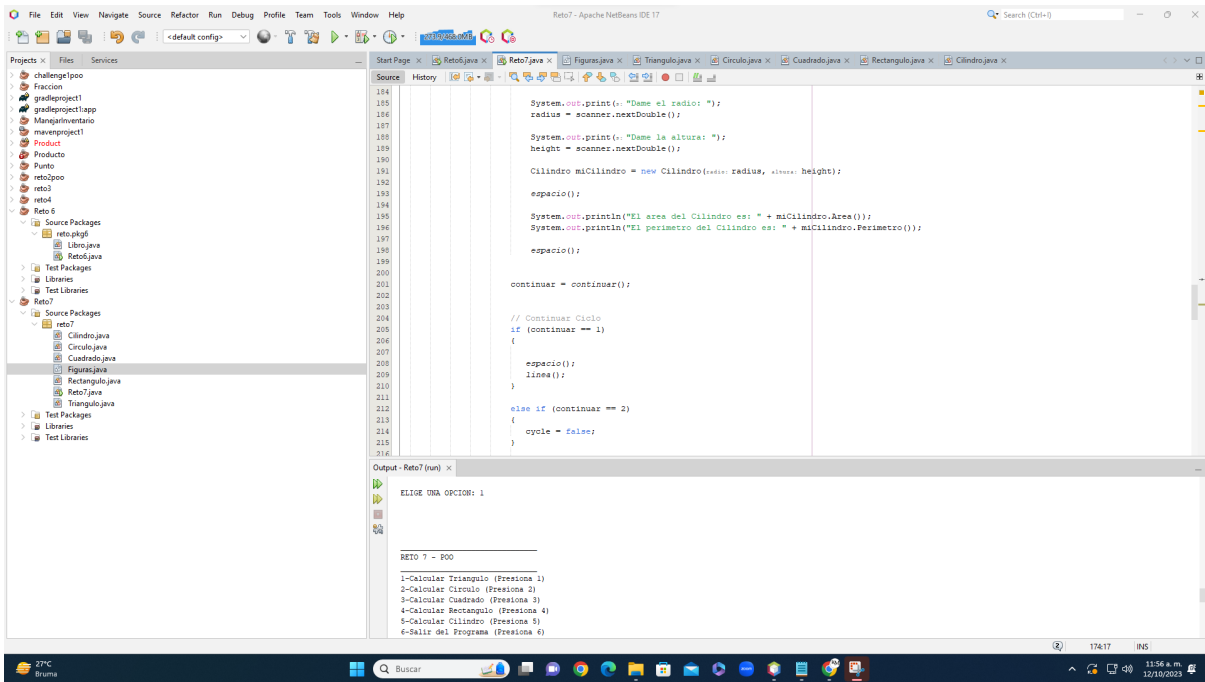
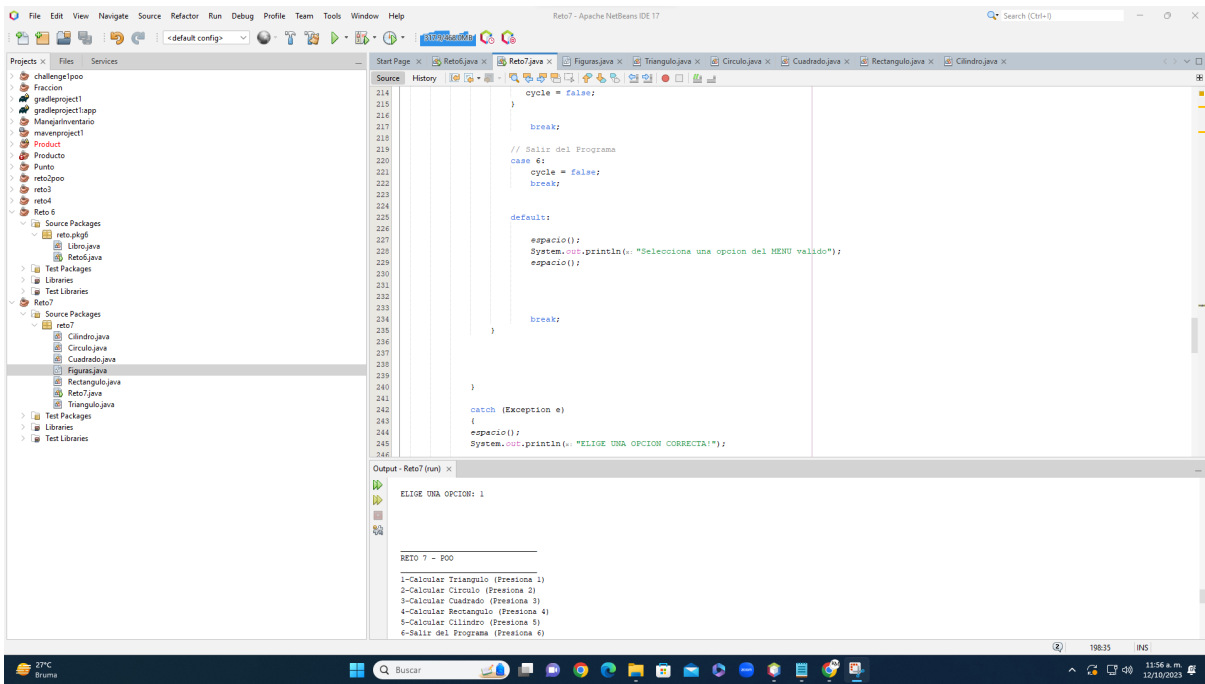


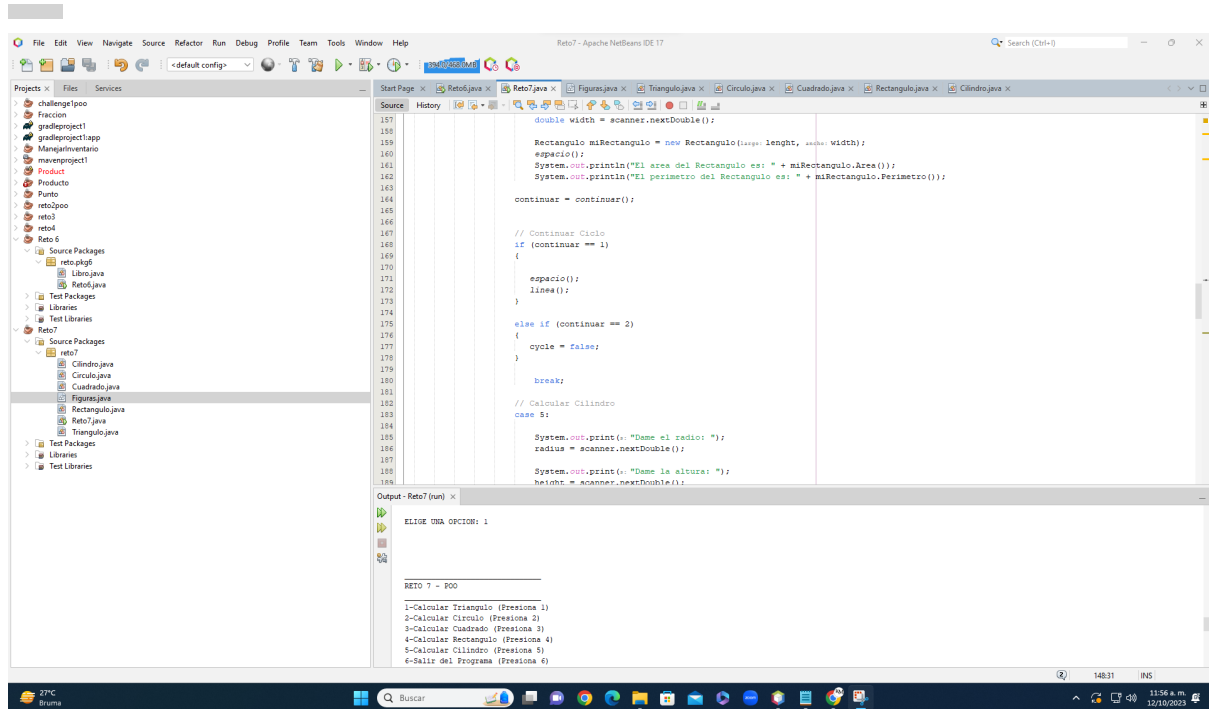




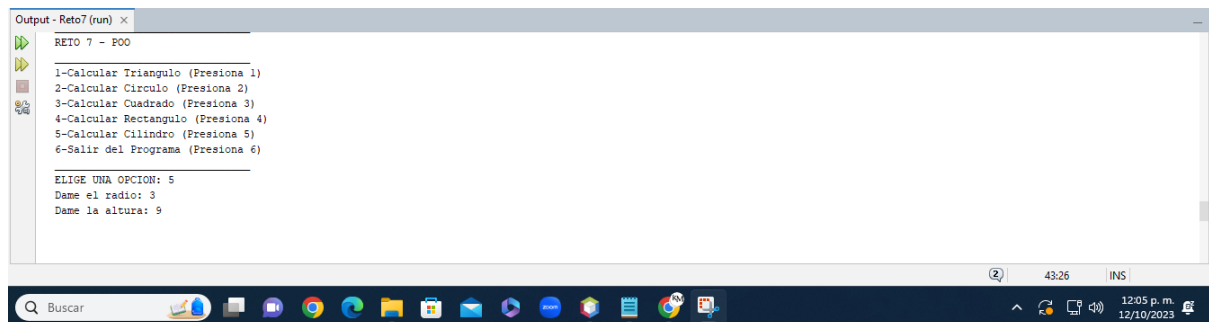
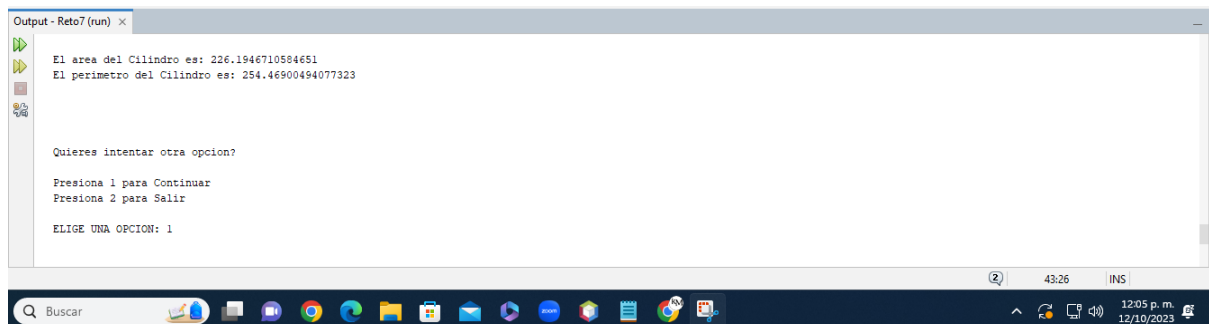
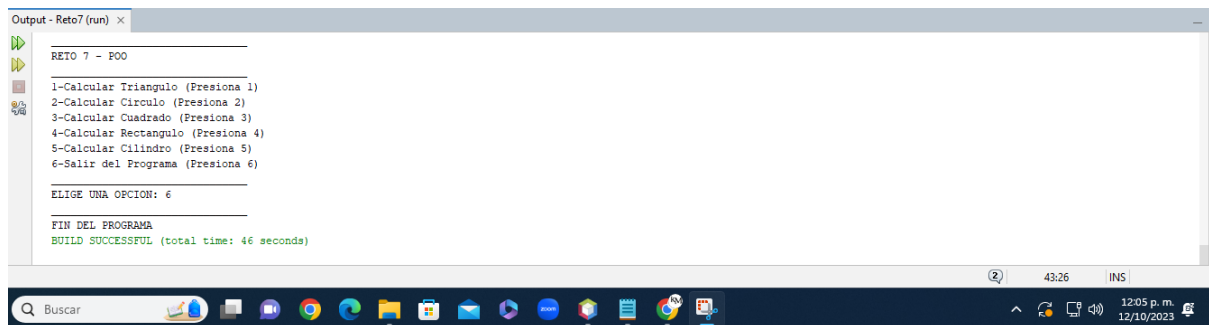








## PRUEBAS:



```
Output - Reto7 (run) x
El area del Rectangulo es: 30.0
El perimetro del Rectangulo es: 22.0
Quieres intentar otra opcion?

Presiona 1 para Continuar
Presiona 2 para Salir

ELIGE UNA OPCION: 1
```

```
Output - Reto7 (run) x
RETO 7 - POO

1-Calcular Triangulo (Presiona 1)
2-Calcular Circulo (Presiona 2)
3-Calcular Cuadrado (Presiona 3)
4-Calcular Rectangulo (Presiona 4)
5-Calcular Cilindro (Presiona 5)
6-Salir del Programa (Presiona 6)

ELIGE UNA OPCION: 4
Dame el largo: 5
Dame el ancho: 6
```

```
Output - Reto7 (run) x
El area del cuadrado es: 20.0
El perimetro del cuadrado es: 18.0

Quieres intentar otra opcion?

Presiona 1 para Continuar
Presiona 2 para Salir

ELIGE UNA OPCION: 1
```

```
Output - Reto7 (run) x
RETO 7 - POO

1-Calcular Triangulo (Presiona 1)
2-Calcular Circulo (Presiona 2)
3-Calcular Cuadrado (Presiona 3)
4-Calcular Rectangulo (Presiona 4)
5-Calcular Cilindro (Presiona 5)
6-Salir del Programa (Presiona 6)

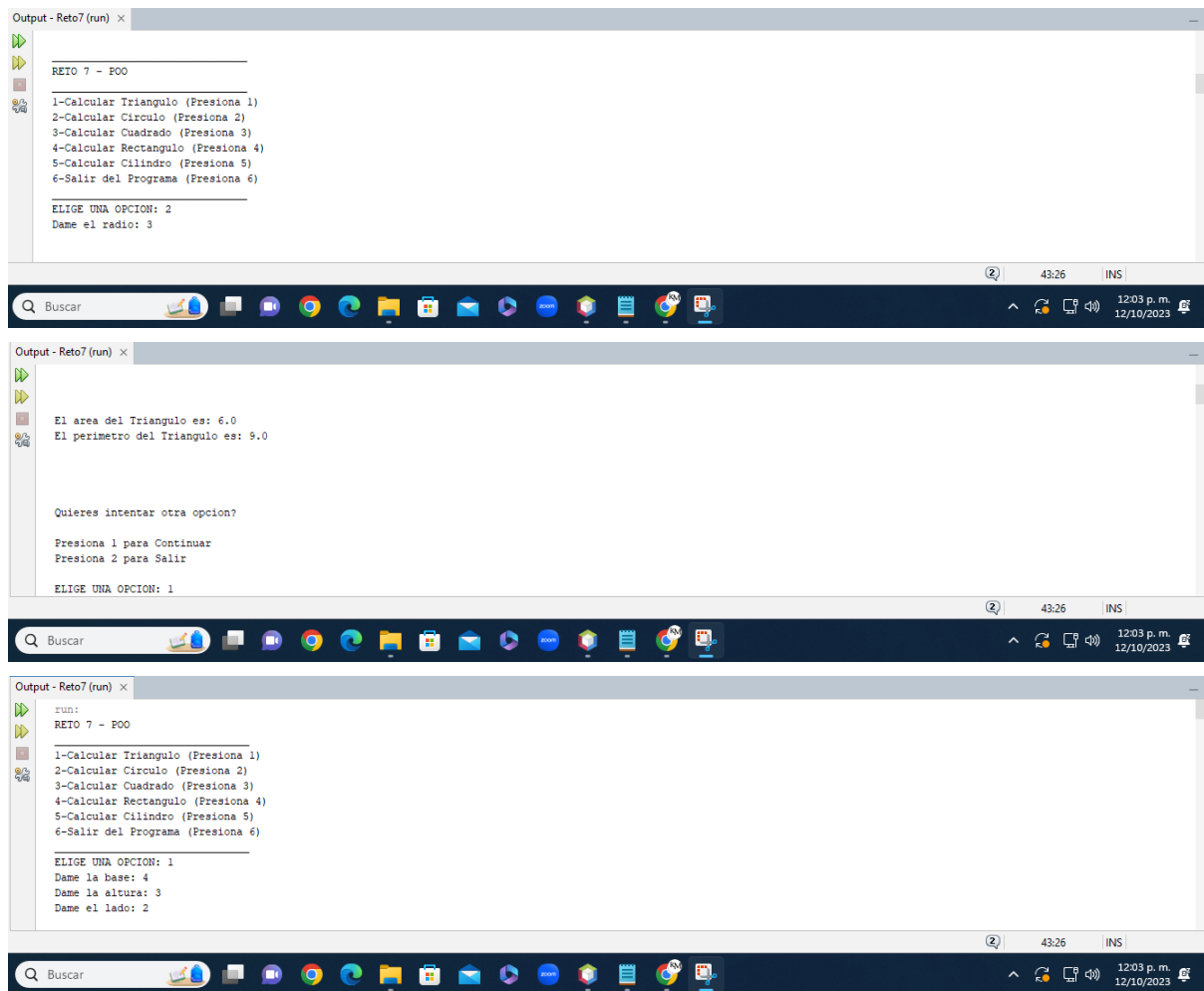
ELIGE UNA OPCION: 3
Dame el primer lado: 4
Dame el segundo lado: 5
```

```
Output - Reto7 (run) x
El area del Circulo es: 28.274333882308138
El perimetro del Circulo es: 18.84955592153876

Quieres intentar otra opcion?

Presiona 1 para Continuar
Presiona 2 para Salir

ELIGE UNA OPCION: 1
```



## **REPORTE:**

Empezaré por explicar las clases abstractas del Programa, la clase superior es Figuras, dentro de la clase únicamente implemente 2 métodos por default para calcular el area y perimetro de cada figura, ya que cada figura geométrica tiene sus propias fórmulas y atributos, si implementaba las características de las figuras dentro de la clase superior luego al hacer herencia, como cada figura tiene sus características únicas al proporcionar parámetros luego iba marcar error porque unas figuras tienen más variables que otras, entonces para simplificarlo solamente implemente atributos en las subclases. Las subclases son de Triángulo, Círculo, Rectángulo, Cuadrado y Cilindro.

En la subclase de Triángulo implemente los atributos de base, altura, y lado, luego a esas características las encapsule como privado, implemente los getters y setters, después agregue el constructor y en los Overrides, en el del Area únicamente codifique que me regresara la formula del area de un triángulo ( $A = b \cdot h / 2$ ) y en el del Perímetro lo mismo pero con su fórmula ( $P = \text{base} + \text{altura} + \text{lado}$ )

En la subclase de Círculo implemente solamente el atributo de radio ya que Java al igual que otros lenguajes tienen reservada la función de PI y exponentes. Luego implemente los

getters y setters, al final en los overrides, en el método del Área únicamente escribir que me regresara el resultado de la fórmula respectiva ( $A = \pi * r^2$ ) y lo mismo con el perímetro del círculo ( $P = 2 * \pi * r$ )

En la subclase del Cuadrado implemente como atributos el lado 1 y lado 2, después implemente los getters, setters y el constructor. Y en los overrides, en el del Área codifique que me retornara  $\text{lado 1} * \text{lado 2}$ , y en el del Perímetro declare 2 variables de tipo double, el lado 3 que fuera igual a lado 1 y lado 4 igual a lado 2, y al final que me regresa la suma de las variables y atributos.

En la subclase del Rectángulo lo hice de la misma forma que la del Cuadrado, únicamente que los atributos les cambie de nombre a Largo y Ancho.

En la subclase de Cilindro declare 2 atributos, el de radio y altura, implemente los getters, setters y constructores, y los Overrides los hice de la siguiente manera: El método de Área implemente su fórmula respectiva ( $A = 2(\pi)(r)(h) + 2(\pi)(r)^2$ ) y en el Método de Perímetro también hice lo mismo con la fórmula que corresponde ( $P = \pi * r^2 * h$ )

Ya finalmente en el Main, structure el programa de manera similar a como lo he estado haciendo con los trabajos de este semestre, implementando funciones básicas que siempre coloqué en los retos, el ciclo While, las excepciones, condicionales Switch, etc. Ya solamente en cada procedimiento instanciaba los objetos desde las subclases y les pasaba los parámetros con los datos de entrada, y en el dato de salida mandaba a llamar los métodos de cada objeto.

## **CONCLUSIÓN:**

Solamente puedo comentar que ya aprendí la diferencia entre instanciar un objeto con una clase normal y una clase abstracta, también comprendo mejor el tema de la herencia y el encapsulamiento, y por lo que veo definitivamente estos conceptos si son útiles para desarrollar código más limpio en vez de estar escribiendo a veces las mismas líneas de código como en la Programación Estructurada sin usar funciones, pero a diferencia de las funciones, los objetos tienen la ventaja de guardar más información de diferentes tipos de datos y pueden realizar diferentes acciones.