

Enunciado.

A lo largo de esta unidad has ido aprendiendo a crear tus propias clases, así como sus distintos miembros (atributos y métodos). Has experimentado con la encapsulación y accesibilidad (modificadores de acceso a miembros), has creado miembros estáticos (de clase), y de instancia (de objeto), has escrito constructores para tus clases, has sobrecargado métodos y los has utilizado en pequeñas aplicaciones. También has tenido tu primer encuentro el concepto de herencia, que ya desarrollarás en unidades más avanzadas junto con otros conceptos avanzados de la Programación Orientada a Objetos.

Una vez finalizada la unidad se puede decir que tienes un dominio adecuado del lenguaje Java como para desarrollar tus propias clases y utilizarlas en una aplicación final que sea capaz de manipular un conjunto de datos simple. Dada esa premisa, esta tarea tendrá que objetivo escribir una pequeña aplicación en Java ampliando algunos de los elementos que has aprendido a utilizar.

Se trata desarrollar una aplicación Java en consola que permita representar varias figuras geométricas. Al lanzarse la aplicación, el programa tendrá que solicitar el usuario los siguientes datos:

1. Tipo de figura. Sólo válidas “circunferencias”, “triángulos” y “rectángulo”.
2. Coordenadas., Toda coordenada tiene 2 valores, X e Y, en número enteros. La aplicación limitará las coordenadas a los puntos +5 tanto en el eje X como en el eje Y. En función de la figura escogida, el programa dará distintas opciones:
 - Para circunferencia, dos coordenadas: centro y punto exterior.
 - Para triángulo, tres coordenadas: los tres vértices.
 - Para rectángulo, dos coordenadas, vértice inferior izquierdo y vértice superior derecho.

Una vez diseñada la figura, mediante un menú se podrán realizar determinadas operaciones:

1. Ver el tipo de figura. El programa devolverá circunferencia, triángulo, rectángulo o cuadrado en función de las coordenadas. Un cuadrado es un rectángulo cuyos lados miden todos igual.
2. Calcular área.
3. Calcular perímetro.
4. Mover coordenada. La aplicación preguntará qué coordenadas se puede mover.
5. Convertir figura. Solo se podrán convertir figuras que compartan el número de coordenadas.
6. Ver lista de coordenadas.
7. Ver figuras en el plano. No se pueden ver en el plano las circunferencias ni triángulos. Esta operación mostrará el rectángulo con O, el centro con X y

espacios en el resto de plano. Un rectángulo cuyos vértices estén en (-1, 1) y (0, 2) se verá así.

```

.....
.....
.....
....OO....
....OO....
....X....
.....
.....
.....
.....
.....
.....

```

8. Salir de la aplicación.

El programa deberá asegurar que las coordenadas son válidas en función al tipo de figura:

- Para los triángulos, al menos dos coordenadas deben de no compartir un punto (x y Y), porque si no se definiría una recta.
- Para los rectángulos, ambas coordenadas deben de no compartir ningún punto, porque si no se definiría una recta.

Además del programa principal de la aplicación (clase con una función **main**), habrá que escribir una clase **FiguraGeo** que proporcione todas las herramientas necesarias para trabajar con este tipo de información

- Constructor (o constructores) adecuados.
- Almacenamiento del tipo de figura (atributos).
- Almacenamiento de las coordenadas (atributos).
- Movimiento de coordenadas (método de interfaz pública).
- Obtención de coordenadas (métodos de interfaz pública).
- Aquellas herramientas auxiliares para poder trabajar cómodamente con el objeto. Algunas de esas herramientas podrán ser públicas. Algunas podrán ser específicas de clase y otras podrán ser de objeto (método de objeto privados, métodos estáticos públicos, etc.).

En general, deberías incluir excepciones para controlar aquellos casos en los que el uso de un método no sea posible (tipo de figura no existente, coordenadas no válidas, conversiones imposibles, puntos muy grandes, etc.).

El código fuente Java de esta clase deberá incluir comentarios en cada atributo o cada conjunto de atributos, y en cada método o cada conjunto de métodos del mismo tipo, indicando su utilidad. El programa principal también deberá incluir

comentarios explicativos sobre su funcionamiento y la utilización de objetos de la clase **FiguraGeo**.

Además del programa deberás escribir también un informe con todas las consideraciones oportunas que se necesiten para entender cómo has realizado la tarea.

El proyecto deberá contener al menos dos archivos fuente Java:

- Programa principal (clase con método **main**: AplicacionFiguraGeo.java).
- La clase FiguraGeo (FiguraGeo.Java).

El documento que contendrá el informe lo elaborarás con un procesador de texto. Debe ser de tipo “.pdf”, y lo podrás generar con cualquier editor de texto con formato, como Microsoft Word o LibreOffice.

Criterios de puntuación: Total 10 puntos.

Para poder empezar a aplicar estos criterios es necesario que la aplicación compile y se ejecute correctamente en un emulador. En caso contrario la puntuación será directamente de 0,00.

CRITERIOS DE PUNTACIÓN.

La clase **FiguraGeo** dispone de todos los atributos necesarios (1,00).

La clase **FiguraGeo** dispone de al menos un constructor y funciona correctamente (1,00).

La clase **FiguraGeo** dispone de los métodos públicos de interfaz necesarios y funcionan correctamente (4,00).

La clase **FiguraGeo** es capaz de validar el tipo de figura y las coordenadas (2,00).

Los métodos de la clase **FiguraGeo** son capaces de lanzar excepciones si se produce alguna situación anómala (1,00).

La clase **FiguraGeo** dispone de métodos estáticos públicos para proporcionar herramientas de gestión útiles al código de fuera de la clase (1,00).

No se han incluido comentarios en la clase **FiguraGeo** (-1,00).

No se han incluido comentarios apropiados en el programa principal (-1,00).

No se han entregado el informe explicativo o se trata de un informe explicativo insuficiente (-3,00).

El programa principal no es capaz de crear un objeto de la clase **FiguraGeo** (-5,00).

Alguna de las opciones de menú del programa principal no funciona correctamente (-1,00 por cada opción).

Total:

Dado que algunos criterios de puntuación son negativos, podría suceder que el balance final fuera negativo. En tal caso la puntuación final será simplemente de 0,00.

Recursos necesarios para realizar la Tarea:

- Ordenador personal.
- JDK y JRE de Java SE.
- Entorno de desarrollo NetBeans con las funcionalidades necesarias para desarrollar y emular midlets.

Consejos y Recomendaciones:

Para realizar la aplicación te realizamos la siguiente serie de recomendaciones:

- Básate en los diferentes ejemplos que has tenido que probar durante el estudio de la unidad. Algunos de ellos te podrán servir de mucha ayuda, así que aprovéchalos.
- El ejercicio resuelto de la clase DNI, en el cual se hacen comprobaciones de entrada, puede servirte de base para la comprobación de la validez.

Indicaciones de Entrega:

Una vez que tengas terminados el programa (carpeta de proyecto NetBeans, incluyendo todos los archivos fuente y todos los recursos) y el documento explicativo, comprime ambos en un único archivo comprimido. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas.