

TAREA N° 2

PROGRAMACIÓN ORIENTADA A OBJETOS

DOCENTES:
Jazna Meza Hidalgo / Alberto Zenteno Acevedo

OBJETIVO

Aplicar lo aprendido en Java. El estudiante deberá aplicar los contenidos revisados en clases relacionados con los conceptos de la orientación a objetos y la implementación de clases en Java. Incluye los conceptos de herencia, manejo de archivos, sobrecarga y sobreescritura de métodos.

ENUNCIADO

Una aplicación geográfica maneja elementos de varios tipos y uno de ellos son los polígonos. Un polígono es un conjunto de líneas conectadas y por su parte cada línea está formada por 2 puntos (en realidad son infinitos puntos, pero para efectos de nuestra realidad sólo serán 2). Finalmente, el elemento básico es el punto que está formado por 2 coordenadas (eje X y eje Y).

De lo anterior surgen todos los conceptos vinculados con la geometría que dicen relación con:

- Distancia euclidiana entre 2 puntos.
- Punto medio de una línea recta.
- Pendiente de la recta.
- Rectas paralelas a uno de los ejes.
- Polígono regular, aquel en el que todos sus lados son iguales y todos los ángulos interiores son iguales.
- Polígono irregular, aquel que no es regular.

CREACIÓN DE RECTAS

Una recta está formada por 2 puntos (inicio y final). Los puntos pueden estar en cualquier cuadrante del plano cartesiano e incluso sobre uno de los ejes). La única forma válida de crear una recta es entregando los 2 puntos que la forman.

La longitud de una recta viene dada por la distancia euclidiana entre los puntos que la forman.

El punto medio de la recta es el punto P que es equidistante del punto de inicio y final de la recta. Dada una recta de puntos (x_1, y_1) y (x_2, y_2) el punto medio viene dado por $(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$

La pendiente de la recta es $m = \frac{y_2-y_1}{x_2-x_1}$ con (x_1, y_1) y (x_2, y_2) los puntos que forman la recta.

Dos rectas se consideran iguales si sus puntos de inicio y fin son iguales.

El ángulo entre 2 rectas se calcula de acuerdo a $\tan^{-1}(\frac{m_2-m_1}{1+m_1*m_2})$ con m_1 y m_2 las pendientes de cada recta. Recuerde que el valor del argumento de la inversa de la tangente es un valor dentro del intervalo $[-\pi; \pi]$

CREACIÓN DE POLÍGONOS

Los polígonos se crean uniendo rectas; para efectos del contexto del problema las rectas de los polígonos que darán en una lista donde la primera recta es la única que no lleva restricción y a partir de la segunda en adelante se debe cumplir que ella se pueda conectar con la

anterior, esto quiere decir que tengan al menos un punto en común. Por ejemplo, se tiene la recta formada por los puntos $(1,1)$ y $(4,3)$; la siguiente recta que sería válida para armar un polígono sería una recta que contuviera uno de los 2 puntos de la línea anterior. Lo anterior se debe dar para cada recta que se pretenda agregar.

Un polígono es válido si es cerrado, es decir, que todas sus rectas están conectadas. Por lo tanto, al momento de intentar crear un polígono debe verificar que sea válido.

Dos polígonos se consideran iguales si todas sus rectas son iguales.

CONCEPTO DE SUB-POLÍGONO

Un sub-polígono es aquel polígono que se forma uniendo todos los puntos medios de los lados del polígono original.

MOSTRAR UN POLÍGONO

Se propone un formato para mostrar la información de un polígono. Este formato incluye indicar la cantidad de lados que tiene, el detalle de las rectas que lo conforman tal y su perímetro tal y como se muestra en la figura:

Polígono :

lados : 4

Formado por :

Recta 1 $[(5,7);(5,10)]$

Recta 2 $[(5,10);(10,10)]$

Recta 3 $[(10,10);(10,7)]$

Recta 4 $[(10,7);(5,7)]$

Perímetro : 16.0

DESPLAZAMIENTO DE POLÍGONOS

Consiste en mover el polígono sobre uno de los ejes, esto es, desplazar las coordenadas de sus puntos en un valor delta (positivo o negativo) sobre uno de los ejes. El desplazamiento se realiza de forma independiente en cada uno de los ejes.

POLÍGONOS COLATERALES

Dos polígonos se llaman colaterales si comparten una de sus rectas.

SU DESAFÍO

Como una parte inicial del proyecto ya ha sido diseñado el diagrama de clases y como se sabe que usted está aprendiendo Java y tiene el desafío de mejorar su conocimiento es que ha sido seleccionado para implementar el diagrama y construir una aplicación gráfica en Java que permita cumplir con los siguientes requerimientos:

1. Cargar desde un archivo de texto llamado PUNTOS.TXT un conjunto de puntos del plano cartesiano. Los puntos deben cargarse en la clase **ListaPuntos** disponible en el diagrama de clases. No se aceptarán puntos repetidos, si se detecta un punto repetido simplemente se descarta. El formato del archivo debe ser líneas formadas por 2 números enteros (positivos o negativos) separados por un punto y coma (;), donde el primer valor se considera la coordenada en X y el segundo la coordenada en Y del punto.

2. Manejar una colección de rectas. Las rectas se deben crear a partir de los puntos disponibles.
3. Manejar una colección de polígonos. Se debe permitir crear polígonos a partir del listado de rectas disponibles. Los polígonos deben ser únicos dentro de la colección y deben ser válidos, por lo tanto, su aplicación debe verificar esto antes de incorporar el polígono a la colección. En caso de que ocurra un error al intentar agregar el polígono debe informar al usuario con un mensaje que le indique la situación.
4. Obtener, a partir de un polígono seleccionado desde la colección, el sub-polígono que se forma.
5. Obtener la lista de todos los sub-polígonos que se pueden armar a partir de la lista de polígonos existentes en la colección.
6. Obtener y mostrar la lista de todos los polígonos con menor perímetro.
7. Obtener y mostrar la lista de todos los polígonos con mayor perímetro.
8. Mostrar el total de polígonos existentes en la colección.
9. Mostrar el polígono ubicado en una determinada ubicación de la colección; el usuario debe seleccionar la ubicación del polígono que desea ver.
10. Mostrar la lista de polígonos cuyo perímetro esté dentro de un rango especificado por el usuario.
11. Mostrar toda la información de un polígono seleccionado por el usuario. Se debe mostrar el número de lados, las líneas que lo forman, su perímetro y saber si es regular o no.
12. Dados 2 polígonos seleccionados por el usuario indicar con un mensaje si son colaterales o no.

REQUERIMIENTOS DE LA APLICACIÓN GRÁFICA

1. Manejar una interfaz gráfica de usuario que cuente con un menú principal y ventanas para cada una de los requerimientos del enunciado. Cada ventana debe abrirse de forma única evitando que si se vuelve a hacer clic en la opción se replique la ventana.
2. Usted debe decidir el diseño de su interfaz gráfica incorporando la mayor cantidad de componentes gráficos que se puedan incluir, pensando siempre en la facilidad de uso de la aplicación de parte de los usuarios.

REQUERIMIENTOS DE USABILIDAD

1. El usuario en cualquier momento puede cancelar el proceso de creación del polígono; esto porque, eventualmente, no podrá crear un polígono con las rectas que se pueden crear con los puntos que se tienen.
2. Usted debe agrupar las opciones del menú para facilitar su uso de parte del usuario.

DIAGRAMA DE CLASES

Ver Anexo - DiagramaClases_T2.png

RESTRICCIONES

- El programa **DEBE** considerar la implementación de código de acuerdo a lo indicado en el diagrama de clases.
 - La tarea **DEBE** compilar al momento de ser entregada.
 - La documentación debe responder al esquema revisado en clases.
-

CONDICIONES

- **Fecha de entrega.** Viernes 14 de Diciembre hasta las 22:00
- **Forma de entrega**
 - Subir los archivos a la plataforma <http://moodleubb.ubiobio.cl>. El período para subir los archivos estará disponible el día de inicio de entrega de la tarea. SE DEBE INDICAR CLARAMENTE los integrantes del grupo y la sección.
- **Formato de entrega**
 - Proyecto NetBeans.
 - Archivo readme.txt indicando los integrantes del equipo
- **Equipo de desarrollo**
 - Dos (2) personas

FORMA DE EVALUACIÓN

- Orden y claridad en el código
- Documentación
- Compilación y ejecución correcto: Funcionamiento correcto.

FORMA DE IDENTIFICACIÓN DE INTEGRANTES DEL EQUIPO

Se debe incluir, en la sección de **@author** de la documentación, el nombre de los integrantes e incluir un archivo README.txt que indique el nombre completo de los integrantes y sección a la cual pertenecen.

OBSERVACIONES IMPORTANTES

- Se acepta el hecho de que los grupos de trabajo puedan compartir el conocimiento, pero lo que **NO SE ACEPTA** es el hecho de que los códigos se encuentren EXACTAMENTE iguales; dicha situación será evaluada con la nota mínima SIN DERECHO A APELACIÓN.
- Las dudas que NO SE RESPONDEN son el *¿cómo se hace esto?* Las dudas que SI SE RESPONDEN son *¿traté de hacer esto, pero no me ha resultado?*
- Las tareas que sean entregadas fuera del plazo establecido serán CASTIGADAS con puntos de descuento. Estos puntos de descuento vienen dados por la siguiente forma:

1,5 puntos por día de atraso

Con respecto a esto, se amerita una importante observación: por ejemplo, si la tarea se entrega entre los siguientes rangos de tiempo, se aplican los descuentos que se muestran a continuación:

Fecha de entrega	Descuento aplicado
14 de Diciembre a las 22:01 horas hasta el 15 de Diciembre a las 21:59 horas	Descuento = 1.5 puntos
15 de Diciembre a las 22:00 horas hasta el 16 de Diciembre a las 21:59 horas	Descuento = 3 puntos
16 de Diciembre a las 22:00 horas hasta el 17 de Diciembre a las 21:59 horas	Descuento = 4.5 puntos
17 de Diciembre días después de las 22:00	Calificación = 1