

Programación Orientada a Objetos

Taller 1: Intro to POO

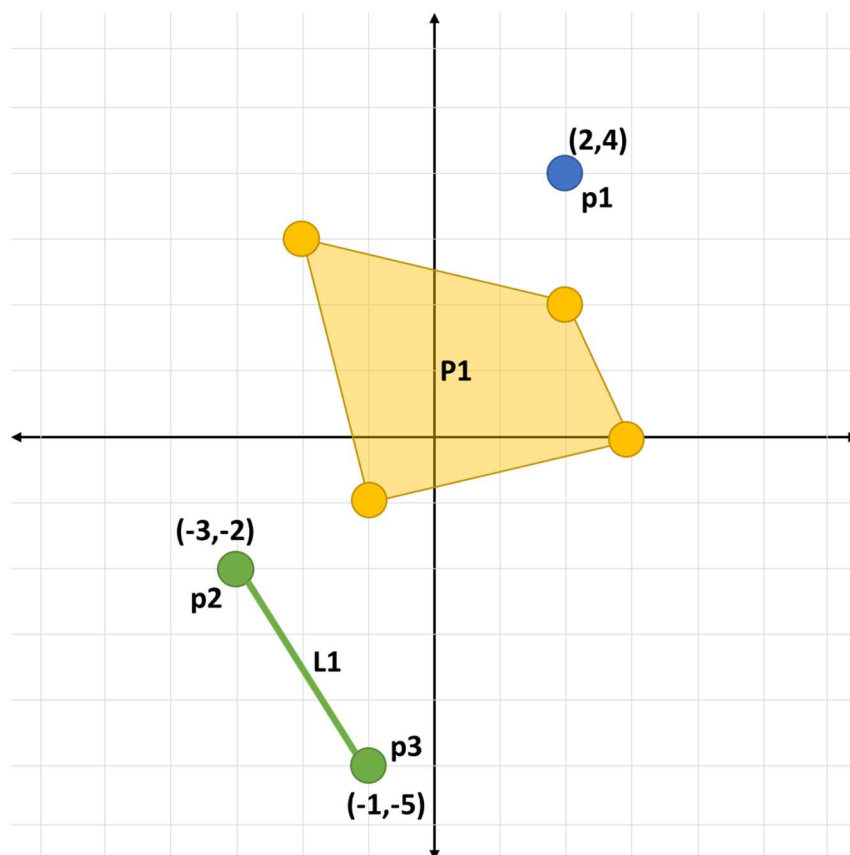
Objetivo

En este taller, usted hará un primer acercamiento práctico a Java y algunos de los conceptos de la programación orientada a objetos. Esto significa que creará clases y utilizará operadores y estructuras de control para lograr el comportamiento de un programa. En este taller, ese programa estará orientado al manejo de algunas entidades matemáticas.

Su implementación debe ser entregada al final de la clase como un **archivo .zip** conteniendo el **proyecto de Netbeans** con su código.

Introducción

La figura mostrada a continuación ilustra tres tipos de elementos matemáticos: puntos, líneas, y polígonos.



Un punto está definido por una posición en el plano cartesiano. Esta posición se expresa mediante una coordenada en el eje **x** y una coordenada en el eje **y**.

Una línea se define mediante dos puntos (pInicio y pFin)

Y un polígono, se define mediante una colección de puntos. En Java, esto puede abstraerse como una clase que contiene un arreglo de puntos.

En esta práctica usted debe:

1. Crear tres clases de Java (en tres archivos distintos) para abstraer cada uno de los conceptos detallados arriba (Punto, Línea, Polígono). Sus clases deben llamarse con el nombre de la entidad que está representando y el nombre de cada clase debe comenzar con una letra mayúscula. Asegúrese de proteger los atributos de sus clases usando el modificador `private`.
2. Implemente métodos de acceso para cada una de las clases definidas en el punto anterior. Los métodos de acceso permiten conocer (get) y modificar (set) los atributos de una clase. Por ejemplo, la clase `Punto` debe tener, entre otros, los métodos `getX` y `setX`. Asegúrese de que los métodos *setters* reciban los argumentos correspondientes.
3. Implemente métodos constructores en cada una de las clases definidas en el punto 1. Un constructor es un método que no tiene tipo de dato de retorno y se llama igual que la clase. Este tipo de método permite crear instancias de una clase mediante el uso de la palabra reservada `new` seguida del nombre de la clase. Por ejemplo, para crear un nuevo punto en un programa principal, usted podría ejecutar la línea:

```
new Punto ( /*argumentos aquí*/ )
```

Sus constructores deben recibir los parámetros correspondientes para que la creación de instancias tenga sentido. Por ejemplo, para crear un nuevo punto, es necesario contar con la información tanto de la coordenada x como de la coordenada y. Esto significa que el constructor de la clase `Punto` debe recibir dos parámetros.

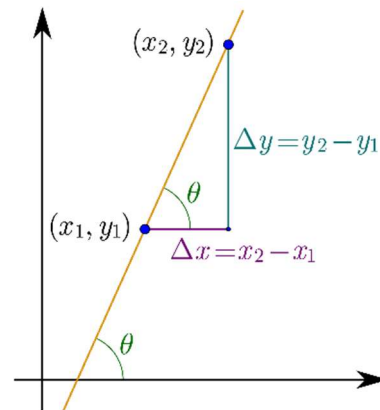
4. Implemente los siguientes métodos dentro de la clase `Punto`:
 - a. `desplazar (int deltaX, int deltaY)`: mueve el punto en `deltaX` unidades en el eje horizontal y `deltaY` unidades en el eje vertical.
 - b. `distanciaA (Punto otroPunto)`: retorna la distancia entre el punto que invoca al método y otro punto que se recibe como parámetro. Para este cálculo, utilice la fórmula de la distancia Euclidiana entre dos puntos (x_1, y_1) y (x_2, y_2), que se define mediante la siguiente fórmula:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

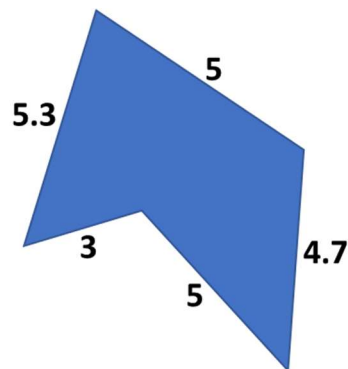
Para calcular la raíz cuadrada y la potencia en Java utilice, respectivamente, los métodos `Math.sqrt(x)` y `Math.pow(a,b)`.

5. Implemente los siguientes métodos dentro de la clase Línea:
- `getPendiente ()`: retorna la inclinación de la línea respecto al eje horizontal. Este cálculo se debe hacer mediante la fórmula:

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$



- `getLongitud ()`: retorna el largo de la línea.
6. Implemente los siguientes métodos dentro de la clase Polígono:
- `getNumLados ()`: retorna un número entero indicando cuántos lados tiene el polígono que invoca el método.
 - `getPerimetro ()`: retorna la suma de las longitudes de los lados del perímetro. Considere el ejemplo mostrado a continuación:



$$\text{Perímetro} = 5.3 + 5 + 4.7 + 5 + 3$$

$$\text{Perímetro} = 23$$

7. Pruebe sus implementaciones en un programa principal, en una clase llamada Main. Recuerde que el método principal debe tener el siguiente prototipo:

```
public static void main(String[] args)
```

- Cree en su programa principal el punto, la línea y el polígono mostrados en la primera figura de este documento.
- Imprima en pantalla las coordenadas del punto creado. Para esto, recuerde utilizar los métodos de acceso de la clase Punto y el método de Java `System.out.println`
- Utilizando el método `desplazar`, de la clase Punto, desplace el punto creado 5 unidades horizontales y 10 unidades verticales. Imprima en pantalla las nuevas coordenadas del punto.
- Imprima la pendiente y longitud de la línea creada.
- Imprima el perímetro del polígono creado.