

Actividad:

1. Adicionar la siguiente línea al inicio del método figuraConMayorPerimetro():

```
if ( figuras.isEmpty()) return "No hay figuras";
```

2. Probar con el main propuesto si el programa efectivamente funciona. El main propuesto es:

```
CajonFiguras cajon = new CajonFiguras();
System.out.println(cajon.figuraConMayorPerimetro());
Figura figura1 = new Figura();
System.out.println(figura1);
cajon.adicionarFigura(figura1);
figura1 = new Figura(3, 0,0, 0, 5, 5,0);
System.out.println(figura1);
cajon.adicionarFigura(figura1);
figura1 = new Figura(3, 0,3,6,-3,-2,-5);
System.out.println(figura1);
cajon.adicionarFigura(figura1);
System.out.println("Hay " + cajon.calcularNumFiguras() + " figuras.");
System.out.println(cajon.figuraConMayorPerimetro());
```

El resultado esperado es:

No hay figuras

La figura es un triángulo ESCALENO de perímetro 12.0.

La figura es un triángulo ISOSCELES de perímetro 17.071067811865476.

La figura es un triángulo ESCALENO de perímetro 24.977703876709214.

Hay 3 figuras.

La figura con mayor perímetro tiene 3 lados, perimetro de 24.977703876709214 y está en la posicion 2

3. Crear un método llamado `cargarTriangulos` que no reciba nada ni retorne nada. El método se encargará de subir los datos almacenados en el archivo `triangulos.csv`, generar los triángulos correspondientes y adicionarlos al `ArrayList` `figuras`. Luego, usar el método para que, al crear un objeto de la clase `CajonFiguras`, se carguen los datos. En la parte del `catch`, retornar un mensaje que indique que no se pudo cargar los datos. ¿El método necesita ser público?
4. Crear un método llamado `figuraConMenorPerimetro()` que no reciba nada por parámetro y que retorne un mensaje que indique cuántos lados tiene la figura, cuál es el perímetro de la figura y en qué posición del `ArrayList` está. Si no hay figuras, el método debe retornar “No hay figuras a evaluar”. ¿El método necesita ser público?
5. Crear un método llamado `figurasConMayorPerimetro()` que no reciba nada por parámetro y que retorne (como `String`) el perímetro de las 3 figuras con mayor perímetro del `ArrayList` `figuras`. Si hay menos de 3 figuras, el método debe retornar “No se pudo determinar las figuras con mayor perímetro”. ¿El método necesita ser público?
6. Crear un método llamado `contarFigurasConPerimetroMayorA(double)` que reciba por parámetro el perímetro mínimo que quiere evaluar y que retorne un entero que cuente el número de figuras que tienen un perímetro mayor al indicado por parámetro. ¿El método necesita ser público?

Hints:

- Para el método `cargarTriangulos()` recuerde usar el mecanismo `try-catch`.
- Recuerde usar el método `split(",")` de la clase `String` para separar cada línea del archivo en los datos que necesita.
- Para convertir datos que son de la clase `String` al tipo `int`, recuerde usar `Integer.parseInt(String s)`.
- Como los datos están para generar triángulos 2D, puede usar el constructor

```
Figura ( int pNumLados, int x1, int y1, int x2, int y2, int x3, int y3)
```

BONUS: Crear un método llamado `cargarFiguras` que no reciba nada ni retorne nada. El método se encargará de subir los datos almacenados en el archivo `figuras.csv`, generar las figuras correspondientes y adicionarlas al `ArrayList` `figuras`. Luego, use el método para que, al crear un objeto de la clase `CajonFiguras`, se carguen los datos. En la parte del `catch`, retorne un mensaje que indique que no se pudo cargar los datos. ¿El método necesita ser público? **Nota:** Note que en este caso, no se puede usar el constructor de triángulos. Así, puede usar el constructor `Figura(int pNumLados,`

`Punto[] pPuntos)` que recibe por parámetro un array de puntos. Siga estas instrucciones como sugerencia para este método:

- a. Escriba el mecanismo try-catch que retorne en el catch si hubo algún error en la carga de datos.
- b. Luego de leer el archivo como lo hizo en `cargarTriangulos()`, escriba un ciclo while que haga lo siguiente:
 - I. Hacer un split de la línea leída por el programa y guardarlos en una variable “datos”.
 - II. Extraer (de la coordenada 0 de la variable “datos”) el número de lados que tendrá la figura.
 - III. Crear un array primitivo de la clase Punto con el tamaño dado por el número de lados.
 - IV. Generar un for con inicio desde 1 hasta la longitud del array “datos” y que aumente de dos en dos y que haga lo siguiente
 - a. Extraer la información de la coordenada x y y que están en la posición i e $i + 1$.
 - b. Crear un objeto de la clase Punto con las coordenadas obtenidas en el punto anterior.
 - c. Almacenar este objeto Punto en el array primitivo de la clase Punto.
 - V. Luego, cree la figura usando el constructor `Figura(int pNumLados, Punto[] pPuntos)`.
 - VI. Finalmente, adicione la figura al ArrayList figuras.