

# Entrega nº 1

Isaías Cárdenas Álvarez

Métodos de programación

Profesores:

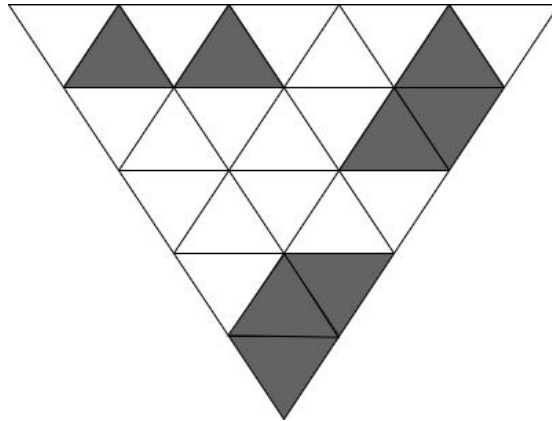
Cristian Jeldes (cátedra)

José Allende (laboratorio)



## Introducción:

El presente proyecto consiste en crear un programa capaz de calcular, en un triángulo conformado por triángulos blancos y negros (como lo muestra la figura 1), el área máxima de un subtriángulo formado por triángulos blancos. Dicho programa, además de encontrar la solución implementando Orientación a Objetos en el lenguaje de programación Java, deberá resolver el problema utilizando dos métodos: el método iterativo y el método recursivo.



El triángulo principal estará representado por dos tipos de caracteres: '#' y '-', siendo triángulos negros y blancos respectivamente. Cada fila del triángulo principal estará conformada por éstos caracteres los cuales a su vez formarán subtriángulos dentro del principal, de esta manera los subtriángulos formados por los caracteres '-' serán los triángulos blancos. Éstos triángulos blancos pueden estar orientados hacia arriba o hacia abajo.

A modo de ejemplo la figura 2 muestra el mismo triángulo de la figura 1 pero en representación de caracteres:

```
- # - # --- # -  
----- ##  
-----  
- ##  
#
```



Para una mejor comprensión de éste documento es preciso definir algunos conceptos que serán indispensables en éste proyecto:

- Programación orientada a objetos:

La programación orientada a objetos es un paradigma de programación que busca emular de manera lógica la realidad o el entorno físico. De esta manera el código de un programa orientado a objetos encapsula "elementos", que tienen ciertas "características" y "funcionalidades", utilizando clases. Una clase es la estructura lógica generalizada para objetos con las mismas características y funcionalidades, llamadas atributos y métodos respectivamente. Una instancia particular de una clase es llamada Objeto. En general los métodos son los que utilizan los atributos de una clase para interactuar con otros objetos o bien para realizar una acción en particular.

- Lenguaje de programación Java:

El lenguaje Java es un lenguaje de programación orientado a objetos, este lenguaje utiliza varias clases, predefinidas en él, que interactúan para generar una aplicación. En general los tipos de datos, las entradas y salidas y hasta el bloque principal (main) son utilizados como clases. La mayor ventaja de éste lenguaje de programación es que fue diseñado para que las aplicaciones puedan ejecutarse desde cualquier dispositivo, de manera que no necesiten ser compiladas de nuevo.

- Diagramas de flujo:

Los diagramas de flujo representan un algoritmo de procesamiento a través de bloques y flechas, así facilita la comprensión de uno o varios procesos que deben llevarse a cabo para lograr un objetivo en específico. En general la sintaxis de los diagramas de flujo consiste en dos óvalos denotados como el inicio y el fin del algoritmo, rectángulos que representan un proceso específico, rombos que indican bifurcaciones en el proceso general y flechas que relacionan las figuras mencionadas anteriormente.



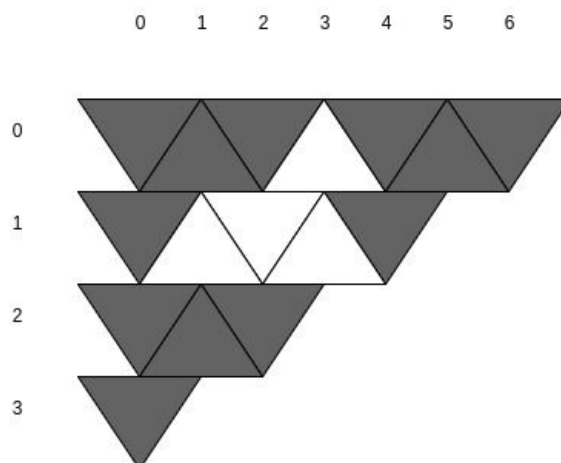
### Descripción del problema:

Como se mencionó anteriormente las filas del triángulo principal están conformadas por dos caracteres: '#' y '-' y dado que cada triángulo blanco (formado por guiones) puede estar orientado hacia arriba o hacia abajo, es necesario verificar todos los triángulos orientados hacia arriba y todos los que estén orientados hacia abajo y luego comparar cuál de todas las áreas es mayor, cabe destacar que todos los triángulos están contenidos en el triángulo principal que tiene orientación hacia abajo. Dicho esto es notable que el triángulo principal se puede recorrer desde la punta a la base o bien desde la base a la punta, lo que podría facilitar o complicar el procedimiento.

Es necesario dejar en claro que el usuario ingresará primeramente la cantidad de filas del triángulo principal y luego ingresará línea a línea las filas de dicho triángulo, desde la de mayor longitud hasta la fila de largo 1. Del mismo modo si el triángulo está representado en un archivo de texto, la primera línea de este archivo será la cantidad de filas del triángulo principal y las siguientes líneas serán las filas del triángulo una a una desde la más larga hasta la de largo 1.

También es destacable que si se alinean todas las filas del triángulo principal se encuentran dos patrones importantes:

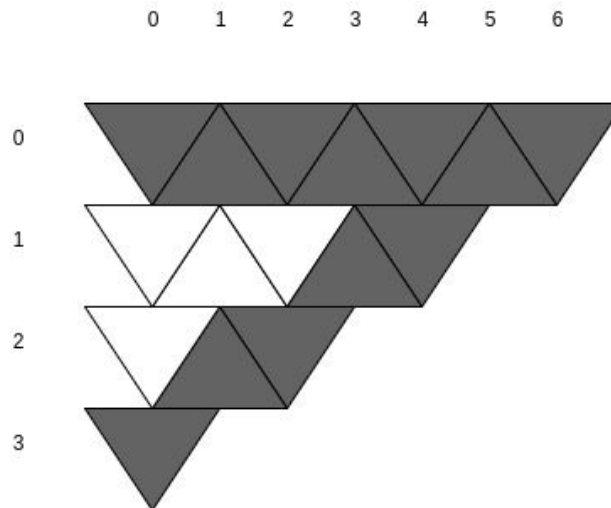
- Triángulos hacia arriba: Tal y como lo muestra la figura 3 cada fila de un triángulo orientado hacia arriba está alineada desde la posición de la punta del mismo hacia la izquierda. Dicho de otro modo, si la posición de la punta del triángulo es "x", la siguiente fila del triángulo blanco estará desde la posición "x-2" hasta "x", luego la fila que sigue estará desde la posición "x-4" hasta "x" y así sucesivamente. Además una punta de un triángulo hacia arriba sólo puede estar en una posición impar.







- Triángulos hacia abajo: De manera similar en los triángulos orientados hacia abajo las filas están alineadas pero hacia la derecha (como lo muestra la figura 4), es decir, si la punta del triángulo está en la posición "x" la fila siguiente estará desde la posición "x" hasta "x+2" y la que le sigue estará desde la posición "x" hasta "x+4". Además una punta de un triángulo hacia abajo sólo puede estar en una posición par.



Si en ambos casos se considera la fila en la que se encuentra la punta del triángulo blanco como la fila nº 0, es posible calcular el área de cada fila del triángulo de la forma:

$$\text{área} = 2 * \text{n}^{\circ}\text{fila} + 1$$

O bien si, además de considerar la posición en la que comienza la fila del triángulo blanco ("x"), consideramos la posición en la que termina como "y"; es posible calcular el área de la fila de la forma:

$$\text{área} = \text{abs}(x - y) + 1$$

El valor absoluto es debido a que dependiendo si el triángulo está hacia arriba o hacia abajo, el valor de "y" podría ser mayor al de "x" o viceversa.

De ésta manera se pueden obtener las áreas de cada fila de un triángulo blanco a partir de las posiciones verticales u horizontales de manera independiente. Luego el área total de un triángulo blanco es la suma de las áreas de las filas que lo componen.



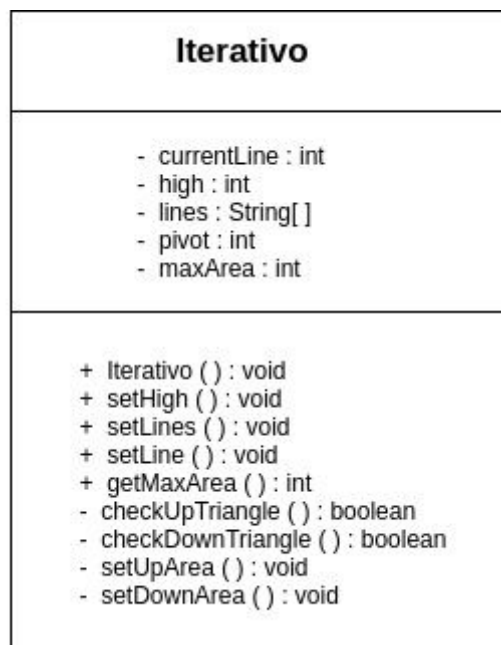
### Descripción de la solución:

Para la resolución de éste problema se utilizará un arreglo de "n" Strings que contenga en cada elemento del arreglo una fila del triángulo principal, desde la fila más larga (en el elemento 0) hasta la fila que contiene un carácter (en el elemento n-1). Luego de solicitarle la información al usuario en el bloque principal (se detalla más adelante) se obtendrá el área del triángulo blanco más grande de dos maneras, según el método a emplear.

Para cada método de resolución se generaron clases diferentes llamadas "Iterativo" y "Recurativo" las cuales son instanciadas en bloque principal según sea el caso.

- Método iterativo:

La clase "Iterativo" fué generada tal como lo muestra el diagrama de clases en la figura 5:



El método principal es "getMaxArea()"; éste método obtiene el área mayor entre los triángulos orientados hacia arriba a través de setUpArea() y los triángulos orientados hacia abajo a través de setDownArea(), ambos métodos mencionados guardan el área mayor en el atributo "maxArea" en caso de que sea mayor al valor anterior. Luego de obtener el área más grande el método retorna el valor del atributo "maxArea".

El método "setUpArea()" obtiene el área mayor entre los triángulos orientados hacia arriba formando triángulos por cada guión en una posición impar y calculando sus áreas sumando la de sus respectivas filas, recorriendo el triángulo principal desde la base a la punta. Utilizando el método "checkUpTriangle()" puede determinar cuando un triángulo hacia arriba es válido por lo que sumará las áreas de las filas de un triángulo blanco mientras sea válido. Luego de calcular un área de un triángulo hacia arriba, a través del número de sus filas,



guarda el valor de ésta en el atributo "maxArea" siempre y cuando sea mayor a su valor. El proceso está descrito en el diagrama de flujo en la figura 6.

`figura 6`

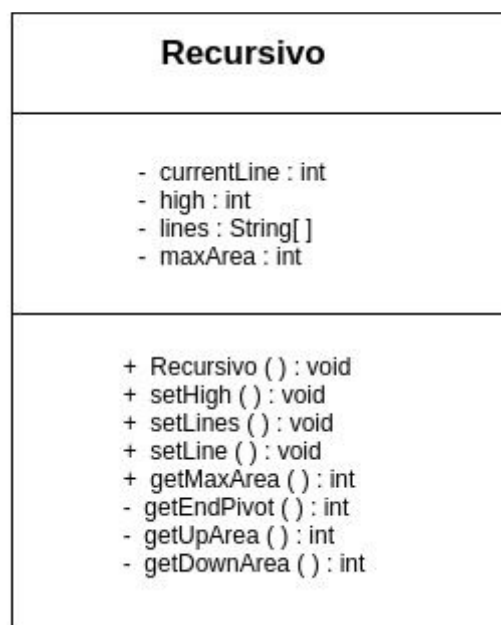
El método "setDownArea()" obtiene el área mayor entre los triángulos orientados hacia abajo formando triángulos por cada guión en una posición par y calculando sus áreas sumando las de sus respectivas filas, recorriendo el triángulo principal desde la punta a la base. De manera similar a "setUpArea()" éste método utiliza "checkDownTriangle()" que verifica si un triángulo hacia abajo es válido. Luego de calcular el área de un triángulo hacia abajo, a través del número de sus filas, compara si ésta es mayor al valor del atributo "maxArea" y en tal caso reemplaza su valor. El proceso está representado en el diagrama de flujo en la figura 7.

`figura 7`

El proceso general que realiza la clase "Iterativo" para calcular el área del triángulo blanco más grande puede observarse en el diagrama de flujo en la figura 8, que resume el proceso completo del método iterativo.

- Método recursivo:

La clase "Recursivo" fue generada como lo muestra el diagrama de clases en la figura 8:



De manera similar a la clase "Iterativo" el método principal es "getMaxArea()" que recorre el triángulo principal desde la



- Main:

Conclusiones:

