Universidad del Istmo de Guatemala
Facultad de Ingenieria
Ing. en Sistemas
Informatica II
Prof. Ernesto Rodriguez - erodriguez@unis.edu.gt

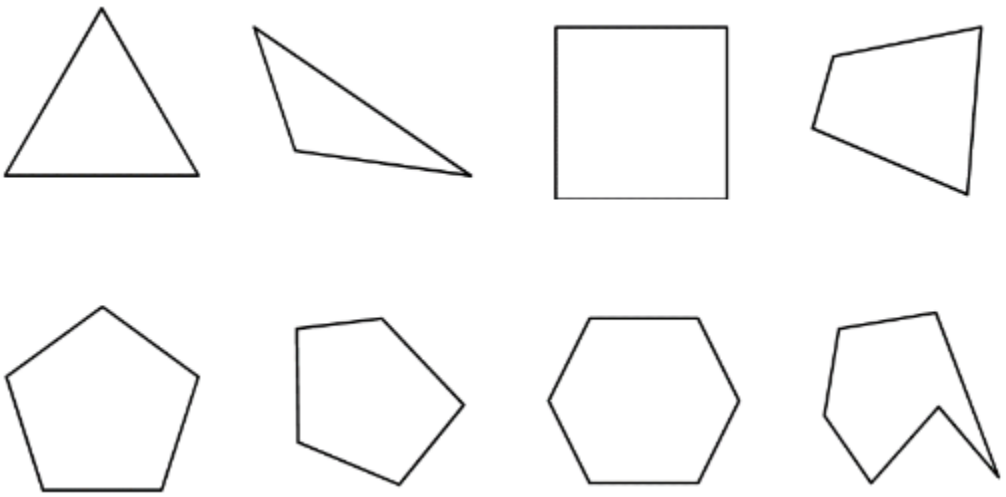
Laboratorio #6

Fecha de entrega: 7 de Marzo, 2019 - 11:59pm

Instrucciones: Resolver cada uno de los ejercicios siguiendo sus respectivas instrucciones. El trabajo debe ser entregado a traves de Github, en su repositorio del curso, colocado en una carpeta llamada "Laboratorio #6". Al menos que la pregunta indique diferente, todas las respuestas a preguntas escritas deben presentarse en un documento formato pdf, el cual haya sido generado mediante Latex. Este laboratorio debe ser elaborado en parejas.

Tarea #1 (20%)

Un poligono es una figura que esta compuesto de una serie de *vertices*. Todos esos vertices estan conectados de tal forma que la figura sea cerrada. Por ejemplo, un triangulo es un poligono con tres puntos y un rectangulo es un poigono de cuatro puntos. A continuación se muestran ejemplos de poligonos.



Su tarea es definir una clase para representar poligonos en espacios de *dos dimensiones*. Tomar en cuenta que un poligono puede tener una *cantidad arbitraria* de vertices. Recuerde hacer uso de la *composiciónn* para separar la complejidad del problema en varias clases incluyendo una clase llamada "Vertice" que se utilize para representar cada vertice del poligono.

La calse *Poligono* debe tener las siguientes caracteristicas:

- Un metodo constructor que acepte una cantidad arbitraria de *vectices* como parametros.

- Debe tener una propiedad que indica el numero de vertices que tiene el poligono.

Tarea #2 (20%)

Existen ocasiones en que un metodo puede no retornar un valor. Un ejemplo es si se desea saber cual es el quinto vertice de un triangulo, dado que un triangulo solamente tiene 3 vertices. Para expresar este comportamiento, en C++ se pueden utilizar *parametros de salida*. Un *parametro de salida* es un parametro pasado ya sea como referencia o puntero a un metodo. Al finalizar la ejecucion, el metodo puede actualizar el valor de dicho parametro mediante la asignación con el operador igual. El metodo también retorna `true` o `false` dependiendo si el metodo retorno un valor o no.

Su tarea es implementar el metodo "*obtenerVertice* : `const int` → `Vertice&` → `bool`". la cual acepta un numero indicando el vertice que se desea obtener y una referencia a una instancia de la clase *Vertice*. Si el numero indicado por el parametro corresponde a un vertice del poligono, este metodo debe asignarle dicho vertice al segundo parametro y retornar `true`. De lo contrario, debe retornar `false`.

Tarea #3 (20%)

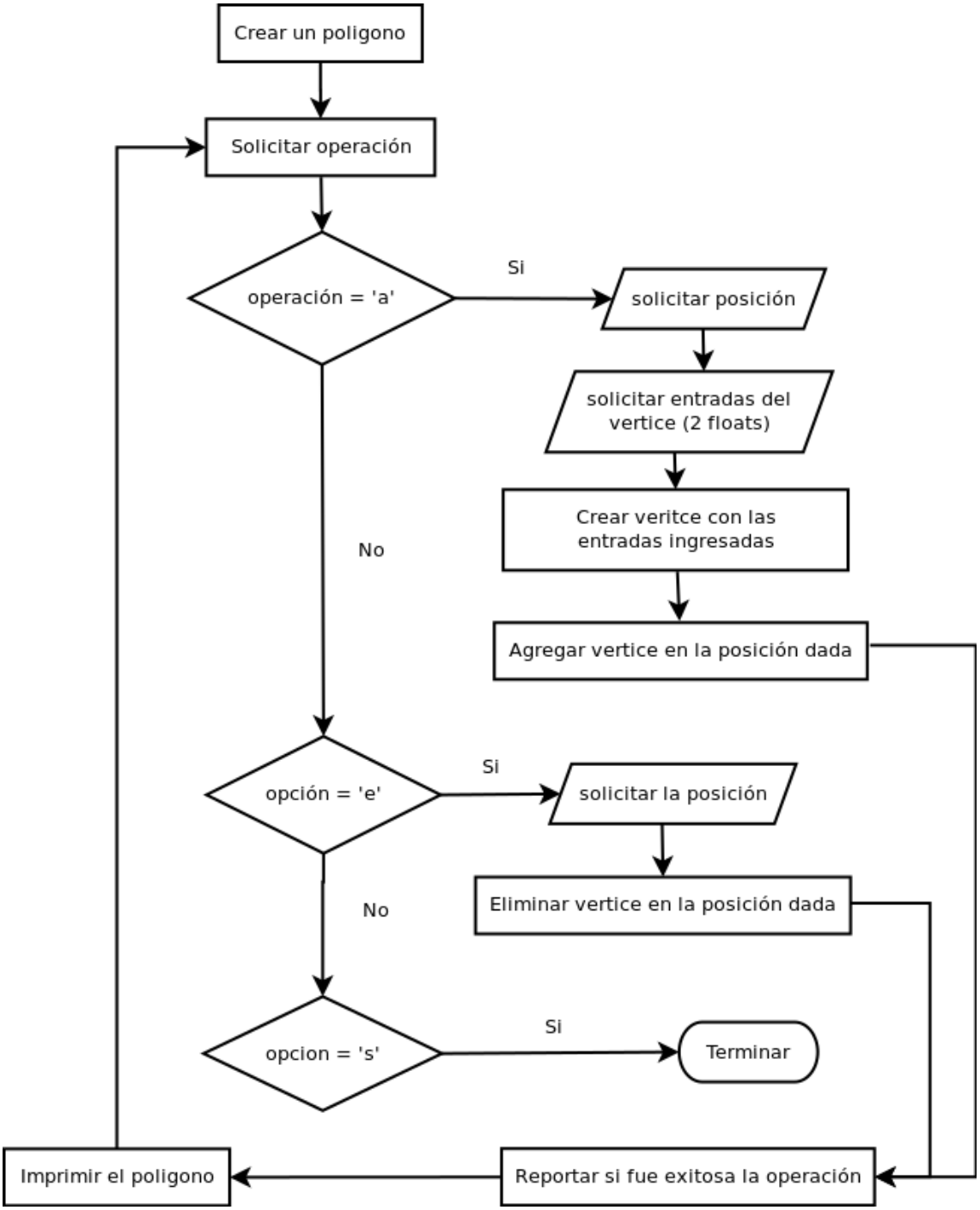
Definir el metodo "*agregarVertice* : `const int` → `const Vertice&` → `const bool`" en la clase *Poligono*. Este metodo acepta un entero como parametro y un *Vertice*. Este vertice debe ser agregado a la lista interna de vertices del poligono. El primer parametro debe estar entre 0 y el numero de vertices actual de vertices en el poligono de lo contrario el metodo no debe hacer nada y retornar `false`. De lo contrario, el vertice debe ser *agregado* en la posición indicada por el primer parametro. Los vertices que se encuentren despues de este parametro deben ser desplazados hacia "atras" para acomodar el nuevo vertice. Recuerde tambien actualizar el numero de vertices que tiene el poligono.

Tarea #4 (20%)

Definir el metodo "*eliminarVertice* : `const int` → `bool`". Este metodo acepta un numero entre 0 y un numero menos que el numero de vertices en este poligono, de lo contrario, debe retornar `false` y no hacer nada. Si el numero esta dentro del rango permitido, debe eliminar el vertice que se encuentra en esa posición del poligono. Todos los vertices posteriores al vertice eliminado deben ser movidos para llenar el espacio que ha quedado. Recuerde actualizar el numeor de vertices en el poligono.

Tarea #5 (20%)

Definir un metodo **main** el cual debe implementar el flujo definido por el diagrama a continuación. Extender el programa y el diagrama con el caso faltante, el cual corresponde a una opción diferente de 'a','e' o 's':


Crear un poligono
Solicitar operación
operación = 'a'
Si
solicitar posición
solicitar entradas del vertice (2 floats)
Crear veritce con las entradas ingresadas
Agregar vertice en la posición dada
No
opción = 'e'
Si
solicitar la posición
Eliminar vertice en la posición dada
No
opcion = 's'
Si
Terminar
Imprimir el poligono
Reportar si fue exitosa la operación