

Guía de prácticas de Algoritmos en Sistemas
Electrónicos
Programa de Ingeniería Electrónica
Escuela Colombiana de Ingeniería Julio Garavito

Ing. Alexander Pérez Ruiz MSc. PhD.

2020 – 1

1. Práctica 1

1.1. Objetivo

Entender y aplicar el principio de herencia de la OOP. Desarrollar la habilidad de establecer relaciones de especialización entre las clases que componen un programa a través del uso de la herencia.

1.2. Conceptos a utilizar

Se deben utilizar los conceptos de:

1. Diagramas de clases.
2. Herencia.
3. Punteros.
4. `std::vector<>`.
5. Funciones virtuales.
6. Memoria dinámica (*new* y *delete*).

1.3. Problema

A partir del análisis de las clases generadas para el problema anterior, crear una jerarquía de clases a partir de una clase padre **Geometrica** de la cual se deben heredar las clases **Circulo**, **Cuadrado**, **Triangulo** y **Pentagono**. Realizar este diseño en UML. General el código para cada una de las clases en su respectivo par de archivos (*.h y *.cpp). Cada uno de las clases debe permitir cambiar el centro de la figura y sus características básicas como el radio en el

círculo o el lado en el cuadrado sin violar el principio de encapsulamiento. Se debe implementar una función *area* y una función *perimetro* para cada una de ellas.

Se debe crear con estas clases un programa que solicite al usuario por consola qué figura geométrica desea crear y solicitarle los valores para las variables necesarias (p.e. el radio en un círculo) y repetir esta acción hasta que el usuario decida no crear más. Estas figuras deben quedar almacenadas en un *std::vector*.

Al terminar la creación de figuras, se debe mostrar en la salida estándar un listado de las figuras ingresadas al vector y mostrar sus características básicas, su perímetro y su área.

2. Práctica 2

2.1. Objetivo

Recordar el procesamiento de cadenas de caracteres e incorporar el tipo de dato *string*.

2.2. Conceptos a utilizar

Se deben utilizar los conceptos de:

1. Enumeraciones.
2. *std::string*.
3. *substring*.
4. *find*.

2.3. Problema

A partir de la jerarquía de clases del problema anterior, crear un programa que permita la lectura de un archivo de texto plano que contiene en cada línea el identificador de la figura geométrica que se debe crear y la lista de parámetros de configuración de acuerdo a lo definido para cada una de ellas. Por ejemplo para un círculo o un triángulo se tendría:

```
1 rd xc yx angulo
3 base altura xc yc angulo
```

Se debe definir una enumeración para evitar números mágicos dentro del programa. El programa leerá todas las líneas del archivo y creará las figuras geométricas de acuerdo a lo allí indicado. Posteriormente se debe mostrar en la salida estándar un listado de las figuras creadas y mostrar sus características básicas, su perímetro y su área. *Bonus:* Procesar el texto en una función *setParametros* y utilizarla para crear un constructor con parámetro *string*.

3. Práctica 3

3.1. Objetivo

Comprender y manejar la estructura de datos tipo lista (*list*) y cola (*Queue*) de la librería estándar de C++ para ser utilizada en los diseños de programas.

3.2. Conceptos a utilizar

Se deben utilizar los conceptos de:

1. *std::list*.
2. *std::queue*.
3. Iteradores.

3.3. Problema

Crear otra implementación de la clase **Polinomio** basada en la lista de la librería estándar (*std::list*) para conocer y utilizar los métodos de *push_front*, *push_back*, *pop_front* y *pop_back*. Para el cálculo del valor promedio de los datos de la lista se debe hacer uso de un iterador.

Realizar otra implementación de la clase **Polinomio** basada en una cola de la librería estándar (*std::queue*) para conocer y utilizar los métodos de *push* y *pop*.