**Programación Orientada a Objetos**

**Segundo Parcial 2019**

Se pide implementar en lenguaje C++ el diagrama que se muestra a continuación:

ManejadorFiguras

*CFigura*

\*

2

CColor

CCirculo

CRectangulo

…

Adjunto a este documento se entregan los archivos de encabezado que definen la interface y los atributos de todas las clases del diagrama. Además, se entrega el archivo main.cpp con la función main implementada y se pide:

1) Completar con el código y clases necesarias para que el proyecto compile sin errores ni advertencias.

2) Agregar el manejo estructurado de errores, de modo que el programa no deje de funcionar si surge algún error en tiempo de ejecución.

3) El método “CargarFiguras” deberá cargar el vector de figuras con los datos que se muestran a continuación:

CRectangulo \*MyRec = new CRectangulo();

float base = 1.2, altura = 2.5;

int color = 0x0A0BFE42;

MyRec->vSetBase(base);

MyRec->vSetAltura(altura);

MyRec->vSetForeColor(color);

color = 0x0BFE0A42;

MyRec->vSetBackColor(color);

Figuras.push\_back(MyRec);

CCirculo \*MyCir = new CCirculo();

float radio = 1.0;

color = 0xFFFF0010;

MyCir->vSetRadio(radio);

MyCir->vSetForeColor(color);

color = 0xFF00FF10;

MyCir->vSetBackColor(color);

Figuras.push\_back(MyCir);

MyRec = new CRectangulo();

base = 0.7; altura = 0.3;

color = 0x03FAAF00;

MyRec->vSetBase(base);

MyRec->vSetAltura(altura);

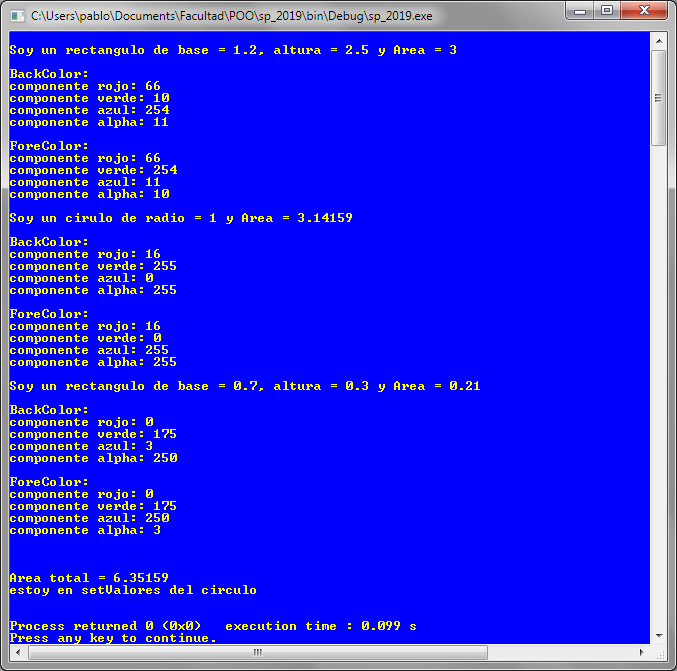
MyRec->vSetForeColor(color);

color = 0xFA03AF00;

MyRec->vSetBackColor(color);

Figuras.push\_back(MyRec);

4) Y con los datos del punto anterior la salida estandar deberá coincidir con la que se muestra a continuación:



5) El archivo “salida.txt” deberá guardar la información con el mismo formato de la salida estandar, salvo que dicha información deberá estar ordenada por área y de forma ascendente.

6) Agregue un nuevo tipo de figura, por ejemplo, un triángulo, con los datos que uds quiera y comente los cambios necesarios para que el programa sea capaz de manejar el nuevo tipo.