

Universidad Nacional de San Agustín

Facultad de Ingeniería de Producción y Servicios
Escuela Profesional de Ciencia de la Computación



ASIGNATURA:

Ciencia de la Computacion (Laboratorio)

DOCENTE:

Enzo Edir Velásquez Lobatón

PERIODO ACADÉMICO:

SEMESTRE 2022-A

GRUPO:

B

ALUMNO:

Apaza Andaluz, Diego Francisco

AREQUIPA - PERÚ

2022

Ejercicio 1:

Defina una clase Forma que tenga los siguientes miembros de datos:

- Color
- Coordenada del centro de la forma (objeto Punto)
- Nombre de la forma (char *)

Y, al menos, las siguientes funciones miembro:

- Imprimir
- Obtener y cambiar el color
- Mover la forma (o sea, su centro)

Defina una clase derivada Rectángulo que tenga los siguientes miembros como datos:

- Lado menor.
- Lado mayor.

Y, al menos, las siguientes funciones miembro:

- Imprimir. Debe imprimir qué se trata de un rectángulo mostrando su nombre, color, centro y lado. Debería usarse la función Imprimir de la clase base para realizar parte de este trabajo.
- Calcular el área (lado menor * lado mayor).
- Calcular el perímetro. ($2 * \text{lado menor} + 2 * \text{lado mayor}$).
- Cambiar el tamaño del rectángulo. Recibe como parámetro un factor de escala. Así, por ejemplo, si el factor vale 2, el rectángulo duplicará su tamaño y si es 0,5 se reducirá a la mitad.

Realice un programa que pruebe el funcionamiento de estas clases. Debe crear objetos y comprobar el correcto funcionamiento de las funciones miembro.

2. Defina una clase Elipse derivada de forma. Recordatorio: una elipse queda definida por su radio mayor (R) y su radio menor (r), tal que el área de una elipse es igual a $\pi * (R * r)$.

3. Defina una clase Cuadrado derivada de la clase Rectángulo.

4. Defina una clase Circulo derivada de la clase Elipse.

5. Realice un programa que defina varias formas diferentes, cree un vector de punteros de la clase Forma que apunten a los objetos creados. El programa debe

realizar un bucle que recorra todas las formas, las ponga todas del mismo color y las mueva a una determinada posición.

6. Analice qué ocurre en el ejercicio anterior si se intenta imprimir la información de cada forma y qué sucede si se intenta obtener en ese bucle el área de todas las formas del vector.

7. Utilice la técnica de las funciones virtuales para arreglar los comportamientos anómalos detectados en el ejercicio anterior.

```
int main()
{

    Forma a("Rojo","Rectangulo",5,9);
    a.imprimir();
    cout<<"Color desde el main: "<<a.getcolor()<<endl;
    a.cambiarCentro(8,9);
    cout<<"\n";
    a.cambiarColor("Cafe");
    a.imprimir();
    cout<<"\n";
    Rectangulo b("Verde","Rectangulo",5,9,10,20);
    b.imprimir();
    cout<<"\n";
    cout<<"El area es: "<<b.area()<<endl;
    cout<<"El perimetro es: "<<b.perimetro();
    b.escala(2);
    cout<<"\n";
    b.imprimir();
    cout<<"\n";
    cout<<"\n";
    Elipse n("Azul","Elipse",3,9,32,8);
    n.imprimir();
```

```

        cout<<"El area es: "<<n.area()<<endl;

        cout<<"\n";

        cout<<"\n";

        circulo m("Azul","Circulo",3,9,8,8);

        m.imprimir();

        cout<<"\n";

        cout<<"\n";

        Cuadrado k("Rojo","Cuadrado",3,9,10,10);

        k.imprimir();

        cout<<"\n";

        cout<<"\n";

        vector<Forma*> lista;

        Rectangulo* r1=new Rectangulo("Verde","Rectangulo",5,9,10,20);

        Cuadrado* r2=new Cuadrado("Rojo","Cuadrado",3,9,10,10);

        lista.push_back(r1);

        lista.push_back(r2);

        lista[0]->imprimir();

        cout<<"\n";

        cout<<"\n";

        lista[1]->imprimir();

        delete r1;

        delete r2;

        return 0;

    }

//Polimorfismo.h

#include <iostream>

#include <string>

using namespace std;

class Forma{

    private:

        string color;

```

```

        string nombre;

        pair<int,int> centro;

    public:

        Forma(string,string,int,int);

        void imprimir();

        string getcolor();

        void cambiarCentro(int,int);

        void cambiarColor(string);

        ~Forma();

};

Forma::Forma(string a,string c,int _x,int _y){

    color=a;

    nombre=c;

    centro.first=_x;

    centro.second=_y;

}

void Forma::imprimir(){

    cout<<"Color: "<<color<<endl;

    cout<<"Nombre: "<<nombre<<endl;

    cout<<"Centro: ("<<centro.first<<","<<centro.second<<)"<<endl;

}

void Forma::cambiarCentro(int x,int y){

    centro.first=x;

    centro.second=y;

    imprimir();

}

string Forma::getcolor(){

    return color;

}

void Forma::cambiarColor(string p){

    color=p;

```

```

}

Forma::~Forma(){}

//Rectangulo.h

#include <iostream>

#include <string>

using namespace std;

class Rectangulo:public Forma{
    private:
        float ladomenor;
        float ladomayor;
    public:
        Rectangulo(string,string,int,int,float,float);
        virtual void imprimir();
        int area();
        int perimetro();
        void escala(float);
        ~Rectangulo();
};

Rectangulo::Rectangulo(string v1,string v2,int v3,int v4,float L,float l):Forma(v1,v2,v3,v4){
    ladomenor=L;
    ladomayor=l;
}

void Rectangulo::imprimir(){
    Forma::imprimir();
    cout<<"Lado mayor: "<<ladomayor<<endl;
    cout<<"Lado menor: "<<ladomenor<<endl;
}

int Rectangulo::area(){
    return ladomenor*ladomayor;
}

```

```
int Rectangulo::perimetro(){  
    return (2*ladomenor)+(2*ladomayor);  
}
```

```
void Rectangulo::escala(float f){  
    ladomenor=ladomenor*f;  
    ladomayor=ladomayor*f;  
}
```

```
Rectangulo::~~Rectangulo(){}  
//ELIPSE.h
```

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
class Elipse:public Forma{  
    private:  
        float r,R;  
    public:  
        Elipse(string,string,int,int,float,float);  
        float area();  
        virtual void imprimir();  
        ~Elipse();  
};
```

```
Elipse::Elipse(string n1,string n2,int n3,int n4,float n5,float n6):Forma(n1,n2,n3,n4){  
    R=n5;  
    r=n6;  
}
```

```
float Elipse::area(){  
    return 3.14*R*r;  
}
```

```
void Elipse::imprimir(){  
    Forma::imprimir();  
}
```

```

        cout<<"Radio mayor: "<<R<<endl;

        cout<<"radio menor: "<<r<<endl;

    }

    Elipse::~Elipse(){}

//CIRCULO.h

#include <iostream>

#include <string>

using namespace std;

class circulo:public Elipse{

    private:

        float Radio;

    public:

        circulo(string,string,int,int,float,float);

        void imprimir();

        ~circulo();

};

circulo::circulo(string m1,string m2,int m3,int m4,float m5,float
m6):Elipse(m1,m2,m3,m4,m5,m6){

    Radio=m6;

}

void circulo::imprimir(){

    Forma::imprimir();

    cout<<"Radio: "<<Radio<<endl;

}

circulo::~~circulo(){}

//CUADRADO.h

#include <iostream>

#include <string>

```



```

using namespace std;

class Cuadrado:public Rectangulo{
    private:
        float L;
    public:
        Cuadrado(string,string,int,int,float,float);
        void imprimir();
        ~Cuadrado();
};

Cuadrado::Cuadrado(string h1,string h2,int h3,int h4,float h5,float
h6):Rectangulo(h1,h2,h3,h4,h5,h6){
    L=h5;
}

void Cuadrado::imprimir(){
    Forma::imprimir();
    cout<<"Lado: "<<L<<endl;
}

Cuadrado::~~Cuadrado

```

```

Color: Azul
Nombre: Elipse
Centro: (3,9)
Radio mayor: 32
radio menor: 8
El area es: 803.84

Color: Azul
Nombre: Circulo
Centro: (3,9)
Radio: 8

Color: Rojo
Nombre: Cuadrado
Centro: (3,9)
Lado: 10

```

8. Desarrolle un programa que, dado un conjunto de formas, calcule cuál tiene el área máxima e imprima la información de dicha forma.

```

void area_mayor(vector<Forma*> b){

```

```

float a1,a2;

Rectangulo* rectangulo=(Rectangulo*)b[0];

Elipse* elipse=(Elipse*)b[1];

a1=rectangulo->area();

a2=elipse->area();

cout<<"Elipse: "<<a2<<endl;

cout<<"Rectangulo: "<<a1<<endl;

if(a1>a2){

    cout<<"El mayor area es del rectangulo \n";

}else{

    cout<<"El mayor area es del elipse \n";

}

}

int main()

{

    vector<Forma*> vect;

    Rectangulo* p1=new Rectangulo("Verde","Rectangulo",5,9,10,20);

    Elipse* e1=new Elipse("Azul","Elipse",3,9,32,8);

    Cuadrado* p2=new Cuadrado("Rojo","Cuadrado",3,9,10,10);

    vect.push_back(p1);

    vect.push_back(e1);

    vect.push_back(p2);

    cout<<"\n";

    cout<<"\n";

    area_mayor(vect);


    delete p1;

    delete p2;

    delete e1;

    return 0;

}

```

```
Elipse: 803.84  
Rectangulo: 200  
El mayor area es del elipse
```

```
-----  
Process exited after 0.02563 seconds with return value 0  
Presione una tecla para continuar . . .
```