UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

FACULTAD DE INGENIERIA DE PRODUCCION Y SERVICIOS

ESCUELA PROFESIONAL DE CIENCIAS DE LA COMPUTACIÓN

LABORATORIO 08 – Polimorfismo.

DOCENTE:

Enzo Edir Velásquez Lobatón

ALUMNO:

Owen Haziel Roque Sosa.

FECHA:

22/05/2022

Arequipa – Perú

```
#ifndef FORMA_H
#define FORMA_H
#include "Punto.h"
#include <iostream>
using namespace std;
//class Punto;
class Forma {
public:
    Forma(char*,Punto*,string);
    ~Forma();
    virtual void imprimir();
    virtual void cambiarColor();
    virtual void cambiarColor(string);
    virtual void moverForma();
    virtual void moverForma(float,float);
    virtual float area() = 0;
    virtual float perimetro() = 0;
    virtual void resize(float) = 0;
protected:
    char* nombre;
    Punto *coord_centro;
    string color;
};
#endif
```

1. *Forma.h*

```cpp
#include "Forma.h"
#include <iostream>
using namespace std;

Forma::Forma(char* _nombre, Punto *_centro, string _color)
    : nombre(_nombre), coord_centro(_centro), color(_color){

}
//solo constructores pueden usar member initializer list

Forma::~Forma(){
    delete coord_centro;    // Eliminar la asignación de memoria asignada a punteros Punto/nombre
    delete nombre;          // mas sigue almacenada en la clase principal Forma
}

void Forma::imprimir(){
    cout<<"Color: "<<color<<endl;
    cout<<"Coordenadas Centro: ";
    cout<<"("<<coord_centro->getX()<<"; "<<coord_centro->getY()<<")"<<endl;
    cout<<"Nombre: "<<nombre<<endl;
    cout<<endl;
}

void Forma::cambiarColor(){
    string ncolor;
    cout<<"Nuevo color: ";
    cin.ignore();
    getline(cin,ncolor);
    color = ncolor;
    cout<<"Hecho!\n";
}
```

```cpp
void Forma::cambiarColor(string _inputcolor){
    color = _inputcolor;
}

void Forma::moverForma(){
    float a, b;
    cout<<"Nuevo valor x,y: ";
    cin>>a>>b;
    coord_centro->x = a;
    coord_centro->y = b;
    cout<<"Hecho!\n";
}

void Forma::moverForma(float _x, float _y){
    coord_centro->x = _x;
    coord_centro->y = _y;
}
```

1.1. Forma.cpp

```
#ifndef RECTANGULO_H
#define RECTANGULO_H

#include "Forma.h"
#include "Punto.h"
#include "Cuadrado.h"
#include <iostream>
using namespace std;

class Rectangulo : public Forma {
public:
    Rectangulo(char*, Punto*, string, float, float);
    ~Rectangulo();
    void convertirCuadrado(Cuadrado c);
    void imprimir();
    float area();
    float perimetro();
    void resize(float);
private:
    float ladoMenor;
    float ladoMayor;
};

#endif
```

2. Rectangulo.h

```cpp
#include "Forma.h"
#include "Punto.h"
#include "Rectangulo.h"
#include "Cuadrado.h"
#include <iostream>
using namespace std;

Rectangulo::Rectangulo(char* _nombre, Punto *_centro,string _color, float _menor, float _mayor)
    : Forma(_nombre,_centro,_color){
    ladoMenor = _menor;
    ladoMayor = _mayor;
}

Rectangulo::~Rectangulo() {}

void Rectangulo::convertirCuadrado(Cuadrado c) {
    ladoMenor = c.lado;
    ladoMayor = c.lado;
}

void Rectangulo::imprimir(){
    Forma::imprimir();
    if (ladoMayor == ladoMenor)
        cout<<"Lado: "<<ladoMayor<<endl;
    else {
    cout<<"Lado Menor: "<<ladoMenor<<endl;
    cout<<"Lado Mayor: "<<ladoMayor<<endl;
    }
}
```

```cpp
float Rectangulo::area(){
    float area = ladoMayor*ladoMenor;
    return area;
}

float Rectangulo::perimetro(){
    float per = 2*(ladoMayor+ladoMenor);
    return per;
}

void Rectangulo::resize(float factor){
    ladoMayor *= factor;
    ladoMenor *= factor;
    if (ladoMayor == ladoMenor)
        cout<<"Nuevo Lado: "<<ladoMayor<<endl;
    else {
    cout<<"Nuevo Lado Menor: "<<ladoMenor<<endl;
    cout<<"Nuevo Lado Mayor: "<<ladoMayor<<endl;
    }
}
```

2.1. Rectangulo.cpp

Clase Adicional Punto:

```h
#ifndef PUNTO_H
#define PUNTO_H
#include <iostream>
using namespace std;

class Punto {
friend class Forma;
public:
    Punto(float,float);
    ~Punto();
    float getX();
    float getY();
protected:
    float x;
    float y;
};

#endif
```

3. Punto.h

```cpp
#include "Punto.h"
#include <iostream>
using namespace std;

Punto::Punto(float _x, float _y) {
    this->x = _x;
    this->y = _y;
}

Punto::~Punto(){}

float Punto::getX(){
    return x;
}

float Punto::getY(){
    return y;
}
```

3.1. Punto.cpp

```h
#ifndef ELIPSE_H
#define ELIPSE_H

#include "Forma.h"
#include "Punto.h"
#include "Circulo.h"
#include <iostream>
using namespace std;

class Elipse : public Forma {
public:
    Elipse(char*, Punto*, string, float, float);
    ~Elipse();
    void convertirCirculo(Circulo c);
    void imprimir();
    float area();
    float perimetro();
    void resize(float);
private:
    float radioMenor;
    float radioMayor;
};
```

4. Elipse.h

```cpp
#include "Forma.h"
#include "Punto.h"
#include "Elipse.h"
#include "Circulo.h"
#define PI 3.14159
#include <iostream>
using namespace std;

Elipse::Elipse(char* _nombre, Punto *_centro,string _color, float _rmenor, float _rmayor)
    : Forma(_nombre,_centro,_color) {
    radioMenor = _rmenor;
    radioMayor = _rmayor;
}

Elipse::~Elipse() {}

void Elipse::convertirCirculo(Circulo c) {
    radioMenor = c.radio;
    radioMayor = c.radio;
}

void Elipse::imprimir(){
    Forma::imprimir();
    if (radioMayor == radioMenor)
        cout<<"Radio: "<<radioMayor<<endl;
    else {
        cout<<"Radio Menor: "<<radioMenor<<endl;
        cout<<"Radio Mayor: "<<radioMayor<<endl;
    }
}
```

```cpp
float Elipse::area(){
    float area = PI*(radioMayor*radioMenor);
    return area;
}

float Elipse::perimetro(){
    if (radioMayor == radioMenor){
        float per = 2*PI*radioMayor;
        return per;
    }
    else
        return 0;
}

void Elipse::resize(float factor){
    radioMayor *= factor;
    radioMenor *= factor;
    if (radioMayor == radioMenor)
        cout<<"Nuevo Radio: "<<radioMayor<<endl;
    else {
        cout<<"Nuevo Radio Menor: "<<radioMenor<<endl;
        cout<<"Nuevo Radio Mayor: "<<radioMayor<<endl;
    }
}
```

4.1. Elipse.cpp

```
// Guia adquirida de: https://m.cplusplus.com/doc/tutorial/inheritance/
#ifndef CUADRADO_H
#define CUADRADO_H

#include "Forma.h"
#include "Punto.h"
#include "Rectangulo.h"
#include <iostream>
using namespace std;

class Cuadrado{
friend class Rectangulo;
public:
    Cuadrado(float);
    ~Cuadrado();
private:
    float lado;
};

#endif
```

5. Cuadrado.h

```
#include "Forma.h"
#include "Punto.h"
#include "Rectangulo.h"
#include "Cuadrado.h"
#include <iostream>
using namespace std;

Cuadrado::Cuadrado(float _lado){
    lado = _lado;
}

Cuadrado::~Cuadrado(){}
```

5.1. Cuadrado.cpp

```
#ifndef CIRCULO_H
#define CIRCULO_H

#include "Forma.h"
#include "Punto.h"
#include "Elipse.h"
#include <iostream>
using namespace std;

class Circulo {
friend class Elipse;
public:
    Circulo(float);
    ~Circulo();
private:
    float radio;
};

#endif
```

6. Circulo.h

```
#include "Forma.h"
#include "Punto.h"
#include "Elipse.h"
#include "Circulo.h"

Circulo::Circulo(float _radio) {
    radio = _radio;
}

Circulo::~Circulo(){}
```

6.1. Circulo.cpp

```cpp
#include "Punto.h"
#include "Forma.h"
#include "Rectangulo.h"
#include "Elipse.h"
#include "Cuadrado.h"
#include "Circulo.h"
#include <iostream>
using namespace std;

//globales
const int nFIG = 20;
int idx = 0;     //Numero de figuras disponibles a crear = 20
Forma *Formas[nFIG];
```

7. *main.cpp - Variables globales y clases usadas*

```
void crearForma(){
    char *nombreForma = new char[20];
    char figOpt;
    string colorForma;
    float a,b;

    cout<<"Nombre de la figura: ";
    cin>>nombreForma;   cin.ignore();
    cout<<"Color Forma: ";
    cin>>colorForma;    cin.ignore();
    cout<<"Coordenadas del centro de la figura: ";
    cin>>a>>b;
    Punto *center=new Punto(a,b);
    //Definicion del objeto
    cout<<"Rectangulo o Elipse: R|E\t";
    cin>>figOpt;
    while ((figOpt != 'R') && (figOpt != 'E')){
        cout<<"Opcion no Valida. Ingrese nuevamente: ";
        cin>>figOpt;
    }
    switch (figOpt){
    case 'R':
        float x, y;
        cout<<"Ingrese valores de los lados: ";
        cin>>x>>y;
        if (x > y)
            Formas[idx] = new Rectangulo(nombreForma,center,colorForma,y,x);
        else if (x == y)
            Formas[idx] = new Rectangulo(nombreForma,center,colorForma,x,x);
        else
            Formas[idx] = new Rectangulo(nombreForma,center,colorForma,x,y);
        Formas[idx]->imprimir();
        idx++;
        break;
    case 'E':
        float r1, r2;
        cout<<"Ingrese valores de los radios: ";
        cin>>r1>>r2;
        if (r1 > r2)
            Formas[idx] = new Elipse(nombreForma,center,colorForma,r2,r1);
        else if (r1 == r2)
            Formas[idx] = new Elipse(nombreForma,center,colorForma,r1,r1);
        else
            Formas[idx] = new Elipse(nombreForma,center,colorForma,r1,r2);
        Formas[idx]->imprimir();
        idx++;
        break;
    }
}
```

7.1. main.cpp - Método CrearFigura

```
void funcionesForma(){
    int i;
    char FunctOpt;
    cout<<"Indice de Forma a acceder: ";
    cin>>i;
    cout<<"\tI| Imprimir Datos Figura.\n";
    cout<<"\tC| Cambiar Color Figura.\n";
    cout<<"\tM| Mover Figura (centro).\n";
    cout<<"\tA| Calcular Area.\n";
    cout<<"\tP| Calcular Perimetro\n";
    cout<<"\tR| Re-escalar.\n";
    cout<<"\tX| Salir\n";
    cout<<"Ingrese Opcion: ";
    cin>>FunctOpt;
    while (FunctOpt != 'X'){
        switch (FunctOpt){
        case 'I':
            Formas[i]->imprimir();
            break;
        case 'C':
            Formas[i]->cambiarColor();
            break;
        case 'M':
            Formas[i]->moverForma();
            break;
        case 'A':
            cout<<"Area: "<<Formas[i]->area();
            break;
        case 'P':
            cout<<"Perimetro: "<<Formas[i]->perimetro();
            break;
        case 'R':
            float factor;
            cout<<"Factor de escala: ";
            cin>>factor;
            Formas[i]->resize(factor);
            break;
        default:
            cout<<"Opcion no valida.";
        }
        cin.ignore();
        cout<<"\nIngrese Opcion: ";
        cin>>FunctOpt;
    }
}
```

7.2. main.cpp - Método funcionesForma

```
void changeColorPosition4All(){
    float a, b;
    string inputcolor;
    cout<<"Color a asignar a todos: ";
    cin.ignore();
    getline(cin,inputcolor);
    cout<<"Coordenadas de posicion central de todos: ";
    cin>>a>>b;
    for(int i=0;i<idx;i++){
        Formas[i]->cambiarColor(inputcolor);
        Formas[i]->moverForma(a,b);
    }
    cout<<"Hecho!\n";
}

void maxAreaofAll(){
    int indx;
    float max = 0;
    for(int i=0;i<idx;i++){
        float area = Formas[i]->area();
        if (area > max){
            max = area;
            indx = i;
        }
    }
    cout<<"Area maxima: "<<Formas[indx]->area();
    cout<<"\nInformacion de la figura:\n";
    Formas[indx]->imprimir();
}

void ShowInfo_AreaofAll(){
    for(int i=0;i<idx;i++){
        Formas[i]->imprimir();
        cout<<"Area de Figura "<<i+1<<": "<<Formas[i]->area();
    }
}
```

7.3. main.cpp - Métodos cambiarPosicionColor, areaMaxima, ImprimirAreaeInfo

```
void menu(char menuOpt) {
    switch (menuOpt) {
    case 'a':   //crear figura
        crearForma();
        break;
    case 'b':   //Acceder a figura, con idx, acceder a las funciones del objeto especifico
        funcionesForma();
        break;
    case 'c':   //Ejercicio 5
        changeColorPosition4All();
        break;
    case 'd':   //Ejercicio 8
        maxAreaofAll();
        break;
    case 'e':   // Ejercicio 6
        ShowInfo_AreaofAll();
        break;
    case 'f':   //Salir
        break;
    default:
        cout<<"Opción no válida.";
    }
}

int main (int argc, char *argv[]) {
    char menuOpt;
    while (menuOpt != 'f'){
        cout<<"********************MENU DE NAVEGACION********************\n";
        cout<<"\ta| Crear nueva Figura.\n";
        cout<<"\tb| Acceder a metodos de una Figura.\n";
        cout<<"\tc| Ejercicio 5: Cambiar color y mover todas las formas.\n";
        cout<<"\td| Ejercicio 8: Calcular el area maxima de todas las formas.\n";
        cout<<"\te| Ejercicio 6: Imprimir info y area de todas las formas.\n";
        cout<<"\tf| Salir\n";
        cin>>menuOpt;
        menu(menuOpt);
    }
    cout<<"Programa Terminado\n";
    delete [] Formas;
    return 0;
}
```

7.4. main.cpp - Menú principal

Resultados:

```
********************MENU DE NAVEGACION********************
        a| Crear nueva Figura.
        b| Acceder a metodos de una Figura.
        c| Ejercicio 5: Cambiar color y mover todas las formas.
        d| Ejercicio 8: Calcular el area maxima de todas las formas.
        e| Ejercicio 6: Imprimir info y area de todas las formas.
        f| Salir
a
Nombre de la figura: MiRectangulo
Color Forma: Blue
Coordenadas del centro de la figura: 2
3
Rectangulo o Elipse: R|E        R
Ingrese valores de los lados: 4
5
Color: Blue
Coordenadas Centro: (2; 3)
Nombre: MiRectangulo

Lado Menor: 4
Lado Mayor: 5
********************MENU DE NAVEGACION********************
        a| Crear nueva Figura.
        b| Acceder a metodos de una Figura.
        c| Ejercicio 5: Cambiar color y mover todas las formas.
        d| Ejercicio 8: Calcular el area maxima de todas las formas.
        e| Ejercicio 6: Imprimir info y area de todas las formas.
        f| Salir
```

```
a
Nombre de la figura: MiCirculo
Color Forma: Negro
Coordenadas del centro de la figura: 2
1
Rectangulo o Elipse: R|E        E
Ingrese valores de los radios: 5
5
Color: Negro
Coordenadas Centro: (2; 1)
Nombre: MiCirculo

Radio: 5
********************MENU DE NAVEGACION********************
        a| Crear nueva Figura.
        b| Acceder a metodos de una Figura.
        c| Ejercicio 5: Cambiar color y mover todas las formas.
        d| Ejercicio 8: Calcular el area maxima de todas las formas.
        e| Ejercicio 6: Imprimir info y area de todas las formas.
        f| Salir
a
Nombre de la figura: MiElipse
Color Forma: Blanco
Coordenadas del centro de la figura: 5
6
Rectangulo o Elipse: R|E        E
Ingrese valores de los radios: 8
9
Color: Blanco
Coordenadas Centro: (5; 6)
Nombre: MiElipse

Radio Menor: 8
Radio Mayor: 9
```

```
********************MENU DE NAVEGACION********************
        a| Crear nueva Figura.
        b| Acceder a metodos de una Figura.
        c| Ejercicio 5: Cambiar color y mover todas las formas.
        d| Ejercicio 8: Calcular el area maxima de todas las formas.
        e| Ejercicio 6: Imprimir info y area de todas las formas.
        f| Salir
a
Nombre de la figura: MiCuadrado
Color Forma: Marron
Coordenadas del centro de la figura: 0
0
Rectangulo o Elipse: R|E        R
Ingrese valores de los lados: 6
6
Color: Marron
Coordenadas Centro: (0; 0)
Nombre: MiCuadrado

Lado: 6
********************MENU DE NAVEGACION********************
        a| Crear nueva Figura.
        b| Acceder a metodos de una Figura.
        c| Ejercicio 5: Cambiar color y mover todas las formas.
        d| Ejercicio 8: Calcular el area maxima de todas las formas.
        e| Ejercicio 6: Imprimir info y area de todas las formas.
        f| Salir
c
Color a asignar a todos: Celeste
Coordenadas de posicion central de todos: 1
1
Hecho!
```

```
**********MENU DE NAVEGACION**********
        a| Crear nueva Figura.
        b| Acceder a metodos de una Figura.
        c| Ejercicio 5: Cambiar color y mover todas las formas.
        d| Ejercicio 8: Calcular el area maxima de todas las formas.
        e| Ejercicio 6: Imprimir info y area de todas las formas.
        f| Salir
e
Color: Celeste
Coordenadas Centro: (1; 1)
Nombre: MiRectangulo

Lado Menor: 4
Lado Mayor: 5
Area de Figura 1: 20Color: Celeste
Coordenadas Centro: (1; 1)
Nombre: MiCirculo

Radio: 5
Area de Figura 2: 78.5397Color: Celeste
Coordenadas Centro: (1; 1)
Nombre: MiElipse

Radio Menor: 8
Radio Mayor: 9
Area de Figura 3: 226.194Color: Celeste
Coordenadas Centro: (1; 1)
Nombre: MiCuadrado

Lado: 6
Area de Figura 4: 36**********MENU DE NAVEGACION**********
```

```
        a| Crear nueva Figura.
        b| Acceder a metodos de una Figura.
        c| Ejercicio 5: Cambiar color y mover todas las formas.
        d| Ejercicio 8: Calcular el area maxima de todas las formas.
        e| Ejercicio 6: Imprimir info y area de todas las formas.
        f| Salir
d
Area maxima: 226.194
Informacion de la figura:
Color: Celeste
Coordenadas Centro: (1; 1)
Nombre: MiElipse

Radio Menor: 8
Radio Mayor: 9
```

```
********************MENU DE NAVEGACION********************
        a| Crear nueva Figura.
        b| Acceder a metodos de una Figura.
        c| Ejercicio 5: Cambiar color y mover todas las formas.
        d| Ejercicio 8: Calcular el area maxima de todas las formas.
        e| Ejercicio 6: Imprimir info y area de todas las formas.
        f| Salir
f
Programa Terminado


<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>
```