

Laboratorio 07

Herencia

Apellidos y Nombres: Quispe Morocco Paul Jesús

Cui: 20190741

Github: [https://github.com/pquispemor/EPCC CCII LABORATORIO](https://github.com/pquispemor/EPCC_CCII_LABORATORIO)

1. Crear una clase Persona del cual tendrá métodos asignar una edad y nombre. Una segunda clase, alumno, tendrá que heredar este contenido y a través de esta clase poder asignar los datos de edad y nombre de los estudiantes.

Persona.h:

```
#include <iostream>
using namespace std;

class Persona
{
private:
    int edad;
    string nombre;
public:
    Persona();
    ~Persona();
    void setEdad(int);
    void setNombre(string);
    int getEdad();
    string getNombre();
    void mostrarDatos();
};

Persona::Persona()
{
}

Persona::~~Persona()
{
}

void Persona::setEdad(int edad){
    this->edad = edad;
}

void Persona::setNombre(string nombre){
    this->nombre = nombre;
}
```

```

int Persona::getEdad(){
    return edad;
}

string Persona::getNombre(){
    return nombre;
}

```

Alumno.h

```

#include <iostream>
#include "Persona.h"
using namespace std;

class Alumno : public Persona
{
private:
    /* data */
public:
    Alumno();
    ~Alumno();
    void mostrarDatos();
};

Alumno::Alumno(){}

Alumno::~Alumno(){}

void Alumno::mostrarDatos(){
    cout<<"Nombre: "<<Persona::getNombre()<<endl;
    cout<<"Edad: "<<Persona::getEdad()<<endl;
}

```

Main.cpp

```

#include <iostream>
#include "Alumno.h"
using namespace std;

int main(){
    int nroAlumnos;
    int edad;
    string nombre;
    cout<<"Indique la Cantidad de Alumnos: ";
    cin>>nroAlumnos;
    Alumno* alumnos;
    alumnos = new Alumno[nroAlumnos];
    for (int i = 0; i < nroAlumnos; i++)

```

```

{
    cout<<"Alumno Nro: "<<i+1<<endl;
    cin.ignore();
    cout<<"Nombre: ";
    getline(cin,nombre);
    cout<<"Edad: ";
    cin>>edad;
    alumnos[i].setNombre(nombre);
    alumnos[i].setEdad(edad);
}

cout << "\n\tDatos de los Alumnos :\n";
for (int i = 0; i < nroAlumnos; i++)
{
    cout<<"Alumno Nro: "<<i+1<<endl;
    alumnos[i].mostrarDatos();
}

delete[] alumnos;
return 0;
}

```

```

PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio1> g++ main.cpp -o main.exe
PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio1> ./main
Indique la Cantidad de Alumnos: 2
Alumno Nro: 1
Nombre: paul
Edad: 23
Alumno Nro: 2
Nombre: diego
Edad: 21

      Datos de los Alumnos :
Alumno Nro: 1
Nombre: paul
Edad: 23
Alumno Nro: 2
Nombre: diego
Edad: 21
PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio1> 

```

2. Crear una clase Color que mantenga 3 valores (RGB). Una segunda clase Material, tendrá como información una variable de texto que describa algún material (Ejemplo: madera, vidrio, plastico, etc.) Una tercera clase, Objetos, deberá de heredar contenido de ambas clases con la finalidad de describir diferentes objetos en cuanto a color y el material. (Ejemplo: mesa de color café y material de plástico)

Color.h:

```

#include <iostream>
using namespace std;

```

```

class Color
{
private:
    int red;
    int green;
    int blue;
public:
    Color();
    Color(int,int,int);
    int getRed();
    int getGreen();
    int getBlue();
    ~Color();
};

Color::Color(){}

Color::Color(int red, int green, int blue){
    this->red = red;
    this->green = green;
    this->blue = blue;
}

Color::~~Color(){}

int Color::getRed(){
    return red;
}

int Color::getGreen(){
    return green;
}

int Color::getBlue(){
    return blue;
}

```

Material.h:

```

#include <iostream>
using namespace std;

class Material
{
private:
    string mat;
public:
    Material();
    Material(string);
    ~Material();
}

```

```

        string getMaterial();
};

Material::Material(/* args */){}

Material::Material(string mat){
    this->mat = mat;
}

Material::~~Material(){}

string Material::getMaterial(){
    return mat;
}

```

Objeto.h:

```

#include <iostream>
#include "Color.h"
#include "Material.h"
using namespace std;

class Objeto : public Color, Material
{
private:
    string nombre;
public:
    Objeto();
    Objeto(int,int,int,string,string);
    ~Objeto();
    void mostrarDatos();
};

Objeto::Objeto(){}
Objeto::Objeto(int red, int green, int blue, string mat, string nombre)
: Color(red,green,blue), Material(mat)
{
    this->nombre = nombre;
}

Objeto::~~Objeto(){}

void Objeto::mostrarDatos(){
    cout<<"Objeto: "<<nombre<<endl;
    cout<<"Color RGB: "<<endl;
    cout<<"Rojo: "<<getRed()<<endl;
    cout<<"Green: "<<getGreen()<<endl;
    cout<<"Blue: "<<getBlue()<<endl;
    cout<<"Material: "<<getMaterial()<<endl;
}

```

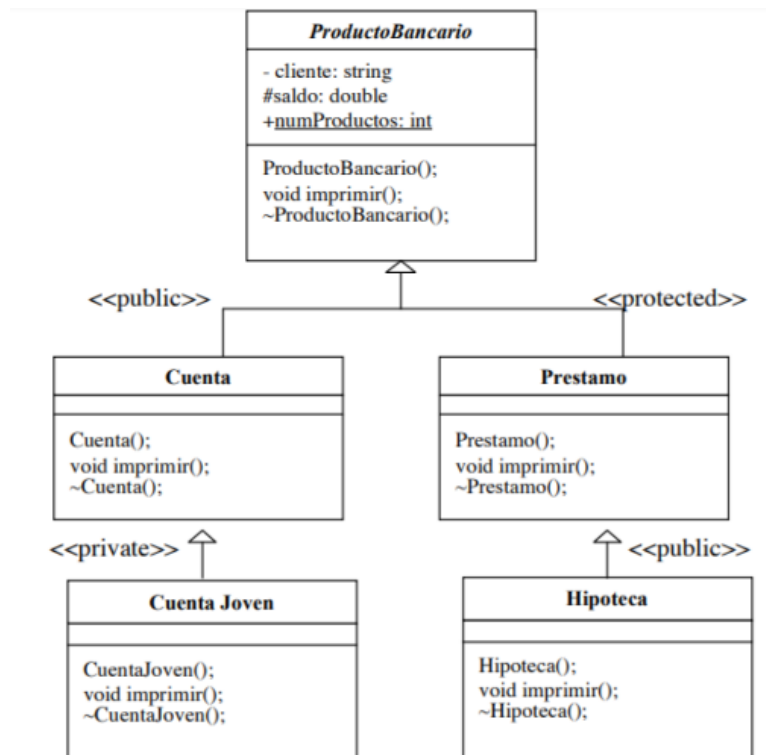
Main.cpp:

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

int main(){
    Objeto* ob1 = new Objeto(128,64,0,"Madera","Mesa");
    ob1->mostrarDatos();
    delete ob1;
    return 0;
}
```

```
PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio2> g++ main.cpp -o main.exe
PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio2> ./main
Objeto: Mesa
Color RGB:
Rojo: 128
Green: 64
Blue: 0
Material: Madera
PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio2> █
```

3. Dada la siguiente jerarquía de herencia, indica la visibilidad de los atributos de la clase ProductoBancario en las clases CuentaJoven e Hipoteca.



ProductoBancario.h

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

class ProductoBancario
{
private:
    string cliente;
protected:
    double saldo;
public:
    ProductoBancario();
    ProductoBancario(string, double);
    ~ProductoBancario();
    void imprimir();
    static int numProductos;
};

int ProductoBancario::numProductos = 0;
ProductoBancario::ProductoBancario(){}

ProductoBancario::ProductoBancario(string cliente, double saldo){
    this->cliente = cliente;
    this->saldo = saldo;
    numProductos++;
}

ProductoBancario::~~ProductoBancario(){}

void ProductoBancario::imprimir(){
    cout<<"Cliente: "<<cliente<<endl;
    cout<<"Saldo: "<<saldo<<endl;
    cout<<"Numero de Productos: "<<numProductos<<endl;
}

#endif
```

Cuenta.h

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

class Cuenta : public ProductoBancario
{
private:

public:
```

```

    Cuenta();
    Cuenta(string,double);
    ~Cuenta();
    void imprimir();
};

Cuenta::Cuenta(/* args */){}

Cuenta::Cuenta(string cliente, double saldo):
ProductoBancario(cliente,saldo){}
Cuenta::~~Cuenta(){}

void Cuenta::imprimir(){
    cout<<"Cuenta: \n";
    ProductoBancario::imprimir();
}

```

CuentaJoven.h

```

#include <iostream>
#include "Cuenta.h"
using namespace std;

class CuentaJoven : private Cuenta
{
private:
    /* data */
public:
    CuentaJoven(/* args */);
    CuentaJoven(string,double);
    ~CuentaJoven();
    void imprimir();
};

CuentaJoven::CuentaJoven(/* args */){}

CuentaJoven::CuentaJoven(string cliente, double saldo):
Cuenta(cliente,saldo){}

CuentaJoven::~~CuentaJoven(){}

void CuentaJoven::imprimir(){
    cout<<"Cuenta Joven: \n";
    ProductoBancario::imprimir();
}

```


Prestamo.h

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

class Prestamo : protected ProductoBancario
{
private:
    /* data */
public:
    Prestamo(/* args */);
    Prestamo(string,double);
    ~Prestamo();
    void imprimir();
};

Prestamo::Prestamo(/* args */)
{
}

Prestamo::Prestamo(string cliente, double saldo):
ProductoBancario(cliente,saldo){}

Prestamo::~~Prestamo(){}

void Prestamo::imprimir(){
    cout<<"Prestamo: \n";
    ProductoBancario::imprimir();
}
```

Hipoteca.h

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

class Hipoteca : public Prestamo
{
private:
    /* data */
public:
    Hipoteca(/* args */);
    Hipoteca(string,double);
    ~Hipoteca();
    void imprimir();
};
```

```

Hipoteca::Hipoteca(/* args */)
{
}

Hipoteca::Hipoteca(string cliente, double saldo):
Prestamo(cliente,saldo){}

Hipoteca::~Hipoteca(){}

void Hipoteca::imprimir(){
    cout<<"Hipoteca: \n";
    ProductoBancario::imprimir();
}

```

Main.cpp

```

#include <iostream>
#include "CuentaJoven.h"
#include "Hipoteca.h"
using namespace std;

int main(){
    CuentaJoven* cj= new CuentaJoven("Paul",124.50);
    cj->imprimir();
    Hipoteca* hp = new Hipoteca("Rodrigo",500.65);
    hp->imprimir();

    delete cj;
    delete hp;
    return 0;
}

```

```

PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio3> g++ main.cpp -o main.exe
PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio3> ./main
Cuenta Joven:
Cliente: Paul
Saldo: 124.5
Numero de Productos: 1
Hipoteca:
Cliente: Rodrigo
Saldo: 500.65
Numero de Productos: 2
PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio3>

```

4. Escribe una clase de nombre ClaseDisco, que herede de la clase ClaseMultimedia los atributos y métodos definidos por usted. La clase ClaseDisco tiene, aparte de los elementos heredados, un atributo más también definido por usted. Al momento de imprimir la información debe mostrarse por pantalla toda la información.

ClaseMultimedia.h

```
#include <iostream>
using namespace std;

class ClaseMultimedia
{
private:
    string autor, cancion;
    int anio;
public:
    ClaseMultimedia(string, string, int);
    ~ClaseMultimedia();
    string getAutor(){
        return autor;
    }
    string getCancion(){
        return cancion;
    }
    int getAnio(){
        return anio;
    }
};

ClaseMultimedia::ClaseMultimedia(string autor, string cancion, int anio)
{
    this->autor = autor;
    this->cancion = cancion;
    this->anio = anio;
}

ClaseMultimedia::~~ClaseMultimedia()
{
}
```

ClaseDisco.h

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

class ClaseDisco : public ClaseMultimedia
{
private:
    string tam;
public:
    ClaseDisco(string, string, int, string);
    ~ClaseDisco();
    void mostrarDatos();
}
```

```
};

ClaseDisco::ClaseDisco(string autor,string cancion, int anio, string tam)
: ClaseMultimedia(autor,cancion,anio)
{
    this->tam = tam;
}

ClaseDisco::~~ClaseDisco()
{
}

void ClaseDisco::mostrarDatos(){
    cout<<"Autor: "<<getAutor()<<endl;
    cout<<"Cancion: "<<getCancion()<<endl;
    cout<<"Año: "<<getAnio()<<endl;
    cout<<"Tamaño del Disco: "<<tam<<endl;
}
```

Main.cpp

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

int main(){
    ClaseDisco* disc = new ClaseDisco("Illeium","Fallen
Embers",2021,"700MB");

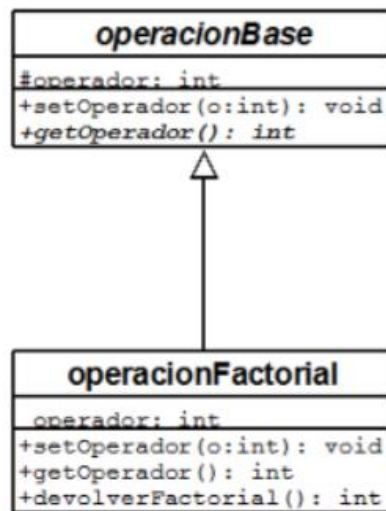
    disc->mostrarDatos();

    delete disc;

    return 0;
}
```

```
PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio4> g++ main.cpp -o main.exe
PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio4> ./main
Autor: Illeium
Canción: Fallen Embers
Año: 2021
Tamaño del Disco: 700MB
PS B:\Universidad\2do Año\Ciencias de la Computacion 2\Laboratorio\Laboratorio07\Ejercicio4> █
```

5. Escribe un programa que implemente la siguiente jerarquía de clases



6. Escribe un programa que implemente la siguiente jerarquía de clases, Debe implementar aquellos atributos y métodos necesarios para que se pueda ejecutar el siguiente programa:

