

**UNIVERSIDAD NACIONAL DE SAN
AGUSTÍN DE AREQUIPA**

**FACULTAD DE INGENIERIA DE PRODUCCION Y
SERVICIOS**

**ESCUELA PROFESIONAL DE CIENCIAS DE LA
COMPUTACIÓN**



LABORATORIO 07 – Herencia.

DOCENTE:

Enzo Edir Velásquez Lobatón

ALUMNO:

Owen Haziél Roque Sosa.

FECHA:

22/05/2022

Arequipa – Perú

```

#ifndef PERSONA_H
#define PERSONA_H
#include <iostream>
using namespace std;

class Persona {
public:
    Persona(int, string);
    ~Persona();
    void setNombre();
    void setEdad();
    void mostrar();
protected:
    int edad;
    string nombre;
};

#endif

```

1. Persona.h

```

#include "Persona.h"
using namespace std;

Persona::Persona(int _edad, string _nombre) {
    edad = _edad;
    nombre = _nombre;
}

Persona::~~Persona() {}

void Persona::setNombre() {
    string _nombre;
    cout<<"Asignar Nombre: ";
    cin.ignore();
    getline(cin, _nombre);
    nombre = _nombre;
}

void Persona::setEdad() {
    int _edad;
    cout<<"Asignar Edad: ";
    cin>>_edad;
    edad = _edad;
}

void Persona::mostrar() {
    cout<<"Nombre: "<<nombre<<endl;
    cout<<"Edad: "<<edad<<endl;
}

```

1.1. Persona.cpp

```

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

class Alumno : public Persona {
public:
    Alumno(int, string);
    ~Alumno();
};

#endif

```

2. Alumno.h

```

#include "Persona.h"
#include "Alumno.h"
using namespace std;

Alumno::Alumno(int _edad, string _nombre) : Persona(_edad, _nombre){}

Alumno::~Alumno() {}

```

2.1. Alumno.cpp

```

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

int main (int argc, char *argv[]) {
    Alumno alu = Alumno(14, "Eduardo");
    alu.mostrar();
    alu.setEdad();
    alu.setNombre();
    alu.mostrar();
    return 0;
}

```

3. main.cpp

```

Nombre: Eduardo
Edad: 14
Asignar Edad: 15
Asignar Nombre: David
Nombre: David
Edad: 15

```

Ejercicio 1: Resultado

```

#ifndef COLOR_H
#define COLOR_H
#include <iostream>
using namespace std;

class Color {
public:
    Color();
    ~Color();
protected:
    int R;
    int G;
    int B;
};

#endif

```

```

#include "Color.h"

Color::Color() {}

Color::~~Color() {}

```

4. Color.h / Color.cpp

```

#ifndef MATERIAL_H
#define MATERIAL_H
#include <iostream>
using namespace std;

class Material {
public:
    Material();
    ~Material();
protected:
    string material;
};

#endif

```

```

#include "Material.h"

Material::Material() {}

Material::~~Material() {}

```

5. Material.h / Material.cpp

```

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

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

#endif

```

```

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

Objeto::Objeto (string _obj, int _r, int _g, int _b, string _mat) {
    obj = _obj;
    R = _r;
    G = _g;
    B = _b;
    material = _mat;
}

Objeto::~Objeto() {
}

void Objeto::describirObj(){
    cout<<obj<<" de color "<<(" "<<R<<"", "<<G<<"", "<<B<<""), y material de "<<material<<"".\\n";
}

```

6. Objeto.h / Objeto.cpp

```

#include "Color.h"
#include "Material.h"
#include "Objeto.h"
#include <iostream>
using namespace std;
//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)
int main (int argc, char *argv[]) {
    Objeto Nuevo = Objeto("Silla",250,122,255,"vidrio");
    Nuevo.describirObj();
    return 0;
}

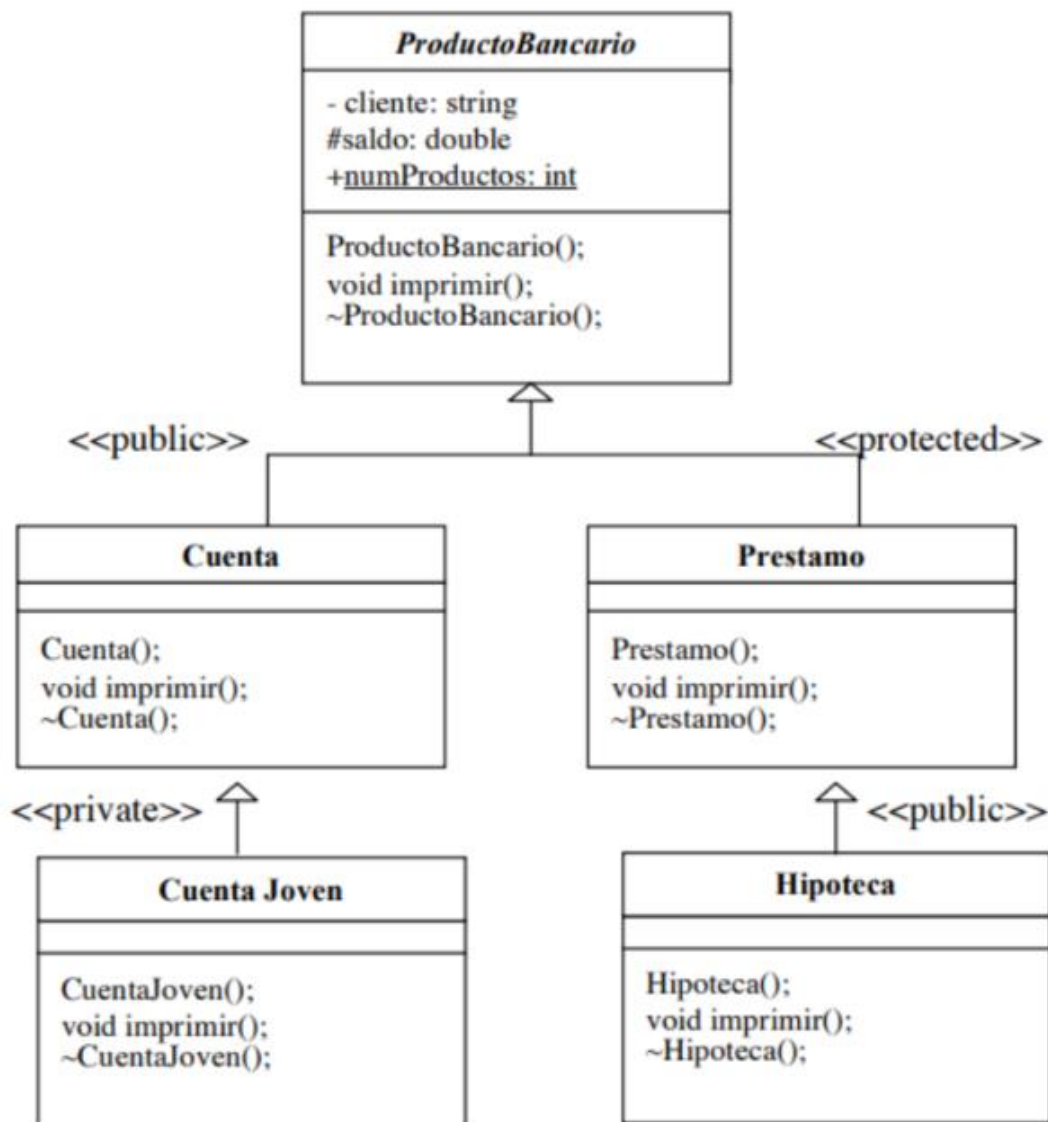
```

7. main.cpp

Silla de color (250, 122, 255), y material de vidrio.

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

Resultado - Ejercicio 2



Ejercicio 3:

- Visibilidad de atributos de clase **ProductoBancario** en **CuentaJoven**:
 - o Si es visible, gracias al comando reservado `public` establecido en el padre **Cuenta**.
- Visibilidad de atributos de clase **ProductoBancario** en **Hipoteca**:
 - o No es visible, solo sería visible a los atributos del padre **Prestamo**, mas no de **ProductoBancario**.

```

#ifndef CLASEMULTIMEDIA_H
#define CLASEMULTIMEDIA_H
#include <iostream>
using namespace std;

class claseMultimedia {
public:
    claseMultimedia();
    ~claseMultimedia();
    void setTipo(string);
protected:
    string tipo;    //físico o digital
};

#endif

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

claseMultimedia::claseMultimedia() {}

claseMultimedia::~~claseMultimedia(){}

void claseMultimedia::setTipo(string _tipo){
    tipo = _tipo;
    cout<<"Hecho\n";
}

```

```

#ifndef CLASEDISCO_H
#define CLASEDISCO_H
#include "claseMultimedia.h"
#include <iostream>
using namespace std;

class claseDisco : public claseMultimedia {
public:
    claseDisco(string, float);
    ~claseDisco();
    void imprimir();
private:
    float capacity;
};

#endif

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

claseDisco::claseDisco(string _tipo, float _capacity) {
    tipo = _tipo;
    capacity = _capacity;
}

claseDisco::~~claseDisco() {}

void claseDisco::imprimir() {
    cout<<"Tipo de multimedia: "<<tipo<<endl;
    cout<<"Capacidad: "<<capacity<<" GB"<<endl;
}

```



```
#include "claseMultimedia.h"
#include "claseDisco.h"
#include<iostream>
using namespace std;
//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.
int main (int argc, char *argv[]) {
    claseDisco DVD = claseDisco("Fisico",4.7);
    DVD.imprimir();
    return 0;
}
```

10. main.cpp

```
Tipo de multimedia: Fisico
Capacidad: 4.7 GB

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

Resultado - Ejercicio 4

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

class operacionBase {
public:
    operacionBase();
    ~operacionBase();
    void setOperador(int);
    int getOperador();
protected:
    int operador;
};

#endif
```

```
#include "operacionBase.h"

operacionBase::operacionBase() {}

operacionBase::~~operacionBase() {}

void operacionBase::setOperador(int _op) {
    operador = _op;
}

int operacionBase::getOperador() {
    return operador;
}
```

11. operacionBase.h / operacionBase.cpp

```

#ifndef OPERACIONFACTORIAL_H
#define OPERACIONFACTORIAL_H
#include "operacionBase.h"
#include <iostream>
using namespace std;

class operacionFactorial : public operacionBase {
public:
    operacionFactorial();
    ~operacionFactorial();
    int devolverFactorial();
private:
};

#endif

```

```

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

operacionFactorial::operacionFactorial() {}

operacionFactorial::~~operacionFactorial() {}

int operacionFactorial::devolverFactorial() {
    int res=1;
    for (int i = 1; i <= operador; i++)
        res*=i;
    return res;
}

```

```

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

int main (int argc, char *argv[])    {
    operacionFactorial op;
    op.setOperador(7);
    cout<<"VALOR OPERADOR: "<<op.getOperador()<<endl;
    cout<<"Factorial: "<<op.devolverFactorial()<<endl;
    return 0;
}

```

13. main.cpp

```

VALOR OPERADOR: 7
Factorial: 5040

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

```

Resultado - Ejercicio 5

```

#ifndef AVE_H
#define AVE_H
#include <iostream>
using namespace std;

class Ave {
public:
    Ave(string,int);
    ~Ave();
    void mostrar();
private:
    string tipo; //terrestre o acuatica
    int mudaPlumajePorAnio;
};

#endif

```

```

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

Ave::Ave(string _tipo,int _mudaplumaje) {
    tipo = _tipo;
    mudaPlumajePorAnio = _mudaplumaje;
}

Ave::~~Ave() {}

void Ave::mostrar(){
    cout<<"Clasificacion del ave: "<<tipo<<endl;
    cout<<"Muda de plumaje por anio: "<<mudaPlumajePorAnio<<endl;
}

```

```

#ifndef GANSO_H
#define GANSO_H
#include "Ave.h"
#include <iostream>
using namespace std;

class Ganso : public Ave {
public:
    Ganso(string,int,float);
    ~Ganso();
    void mostrarGanso();
private:
    float PlumasKg;
};

#endif

```

```

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

Ganso::Ganso(string _tipo,int _mudaplumaje,float _plumaskg) : Ave(_tipo,_mudaplumaje) {
    PlumasKg = _plumaskg;
}

Ganso::~Ganso() {}

void Ganso::mostrarGanso(){
    Ave::mostrar();
    cout<<"Plumas de ganso importadas por Kg: "<<PlumasKg<<endl;
}

```

15. Ganso.h / Ganso.cpp

```

#ifndef GALLINA_H
#define GALLINA_H
#include "Ave.h"
#include <iostream>
using namespace std;

class Gallina : public Ave {
public:
    Gallina(string, int, int);
    ~Gallina();
    void mostrarGallina();
private:
    int cantAnualHuevos;
};

#endif

```

```

#include "Gallina.h"
#include "Ave.h"
using namespace std;

Gallina::Gallina(string _tipo, int _cantPlumaje, int _huevos) : Ave(_tipo, _cantPlumaje) {
    cantAnualHuevos = _huevos;
}

Gallina::~Gallina() {}

void Gallina::mostrarGallina() {
    Ave::mostrar();
    cout<<"Cantidad anual de huevos: "<<cantAnualHuevos<<endl;
}

```

```

#ifndef PATO_H
#define PATO_H
#include "Ave.h"
#include <iostream>
using namespace std;

class Pato : public Ave {
public:
    Pato(string,int,int);
    ~Pato();
    void mostrarPato();
private:
    int carnePatoExported;
};

#endif

```

```

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

Pato::Pato(string _tipo,int _mudaplumaje,int _carneExp) : Ave(_tipo,_mudaplumaje) {
    carnePatoExported = _carneExp;
}

Pato::~~Pato() {}

void Pato::mostrarPato() {
    Ave::mostrar();
    cout<<"Cantidad de carne de pato exportada (Kg): "<<carnePatoExported<<endl;
}

```

```

#include "Ave.h"
#include "Ganso.h"
#include "Gallina.h"
#include "Pato.h"
#include<iostream>
using namespace std;

int main (int argc, char *argv[]) {
    Gallina g("Terrestre",1,41200);
    Ganso a("Acuatico",1,50);
    Pato p("Acuatico",2,1500);
    g.mostrarGallina();
    a.mostrarGanso();
    p.mostrarPato();
    return 0;
}

```

18. main.cpp

```

Clasificacion del ave: Terrestre
Muda de plumaje por anio: 1
Cantidad anual de huevos: 41200
Clasificacion del ave: Acuatico
Muda de plumaje por anio: 1
Plumas de ganso importadas por Kg: 50
Clasificacion del ave: Acuatico
Muda de plumaje por anio: 2
Cantidad de carne de pato exportada (Kg): 1500

```

Resultado - Ejercicio 6