

Unidad IV Modelado Orientado a Objetos con Estructuras de Control Secuenciales y Selectivas

Tema 03 Resolución de problemas empleando la Metodología Orientada a Objeto

Introducción a la Computación

Contenido

3. Resolución de problemas empleando la Metodología Orientada a Objeto MOO, en los que se requieran estructuras básicas de control: secuenciales y selectivas

3. Resolución de Problemas empleando la Metodología Orientada a Objeto (MOO)

De un Alumno se tiene el nombre, apellido y dos notas. Se requiere que realice un programa usando la metodología OO a fin de que se calcule y muestre el promedio del alumno.

FASE 1 ANÁLISIS (AOO)

PASO 1 Subrayado
(Ver el enunciado)

PASO 3 Lógica de los métodos (sólo de los métodos que implican cálculos)

calcularPromedio

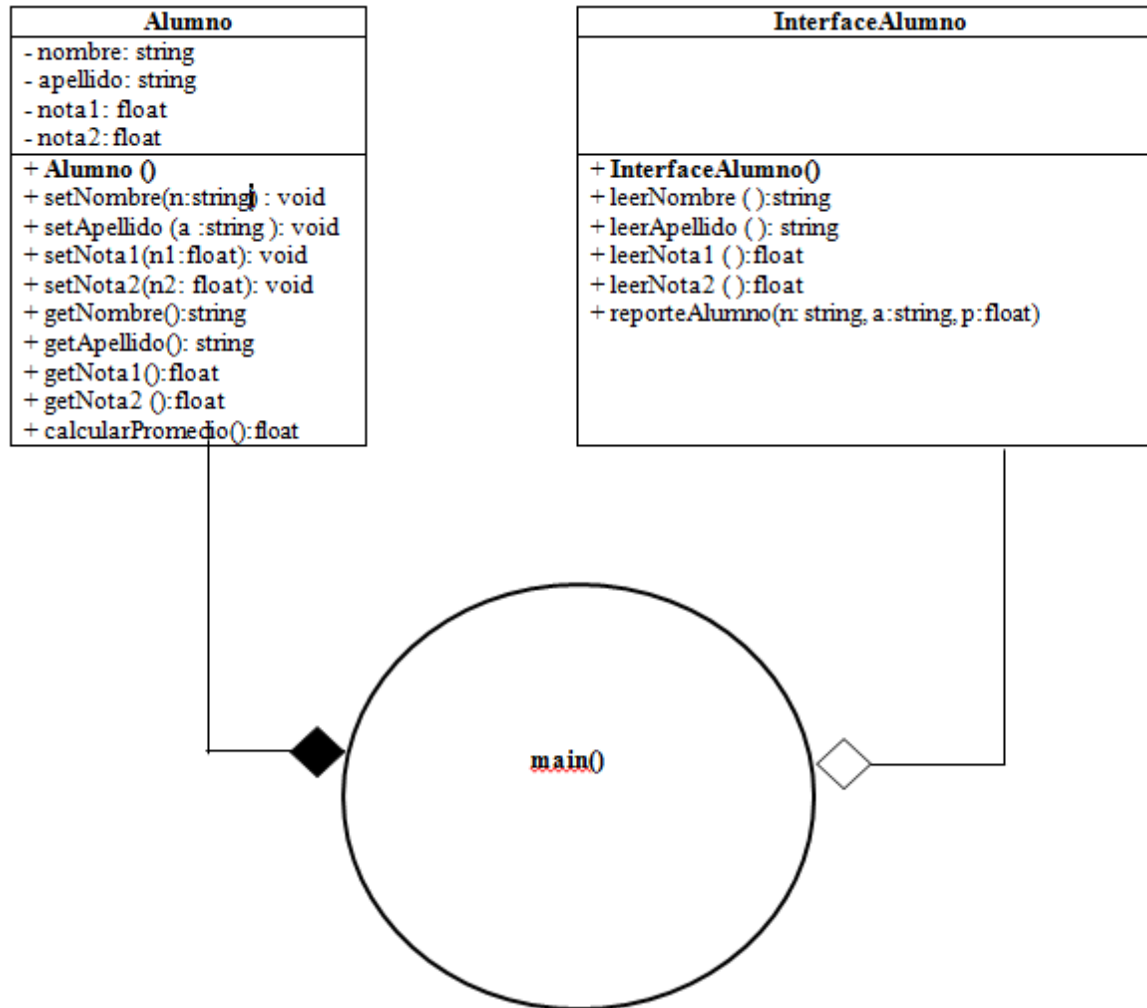
$\text{prom} = (\text{nota1} + \text{nota2}) / 2$

PASO 2 Asignar responsabilidades

Lista	Responsabilidad
Alumno	Clase
nombre apellido nota1 nota2	Atributos /Alumno
setNombre getNombre setApellido getApellido setNota1 getNota1 setNota2 getNota2 calcularPromedio	Métodos /Alumno
InterfaceAlumno	Clase
leerNombre leerApellido leerNota1 leerNota2 reporteAlumno	Métodos/InterfaceAlumno

3. Resolución de Problemas empleando la Metodología Orientada a Objeto (MOO)

FASE 2 Diseño (DOO)



Nota: Los diagramas de clases muestran las relaciones entre estas, donde las clases se representan con rectángulos.

Como el main no es una clase lo representaremos con un círculo, pero no es notación de UML.

En la programación que verá en el siguiente semestre el código del main se convertirá en una clase llamada controlador

3. Resolución de Problemas empleando la Metodología Orientada a Objeto (MOO)

FASE 3 Implementación (IOO) Usando como lenguaje formal C++

// Alumno.h

```
#ifndef ALUMNO_H_
#define ALUMNO_H_
#include <string>
using namespace std;
class Alumno {
    string nombre, apellido;
    float nota1, nota2;

public:
    Alumno();
    void setNombre (string n);
    string getNombre ();
    void setApellido (string a);
    string getApellido ();
    void setNota1 (float n1);
    float getNota1 ();
    void setNota2 (float n2);
    float getNota2 ();
    float calcularPromedio ();
};
#endif /* ALUMNO_H_ */
```

Uno de los archivo que se generan en C++ al crear una clase tiene extensión **.h** y contiene la **estructura de la Clase**

En este archivo **.h** definimos los atributos de la clase y todos los métodos que ésta contendrá

3. Resolución de Problemas empleando la Metodología Orientada a Objeto (MOO)

FASE 3 Implementación (IOO) Usando como lenguaje formal C++

//Alumno.cpp

```
#include "Alumno.h"
```

```
Alumno::Alumno() { }
```

```
void Alumno::setNombre (string n)  
{ nombre = n;}
```

```
string Alumno::getNombre ()  
{ return nombre;}
```

```
void Alumno::setApellido (string a)  
{ apellido = a;}
```

```
string Alumno::getApellido ()  
{ return apellido;}
```

```
void Alumno::setNota1 (float n1)  
{ nota1 =n1;}
```

```
float Alumno::getNota1 ()  
{ return nota1;}
```

Otro archivo que se genera en C++ al crear una clase tiene extensión **.cpp** y contendrá la implementación o programación de todos los métodos que conforman la clase y que deben estar definidos en el archivo **.h**

```
void Alumno::setNota2 (float n2)  
{  
    nota2 = n2;  
}
```

```
float Alumno::getNota2 ()  
{  
    return nota2;  
}  
float Alumno::calcularPromedio ()  
{  
    return (nota1 + nota2) /2;  
}
```

3. Resolución de Problemas empleando la Metodología Orientada a Objeto (MOO)

FASE 3 Implementación (IOO) Usando como lenguaje formal C++

//InterfaceAlumno.h

```
#ifndef INTERFACEALUMNO_H_
#define INTERFACEALUMNO_H_
#include <string>
using namespace std;

class InterfaceAlumno {
public:
    InterfaceAlumno();
    string leerNombre ();
    string leerApellido();
    float leeNota1();
    float leeNota2();
    void reporteAlumno(string n, string a,
float n1 , float n2, float p);
};

#endif /* INTERFACEALUMNO_H_ */
```

InterfaceAlumno es otra clase que se encarga de contener los métodos necesarios para la interacción con el usuario, como leer los datos por teclado y mostrar datos por pantalla (reportes)

Como toda clase en lenguaje C++ tendrá un archivo **.h** y un archivo **.cpp** para definir la estructura de la clase y la programación de los métodos respectivamente.

3. Resolución de Problemas empleando la Metodología Orientada a Objeto (MOO)

FASE 3 Implementación (IOO) Usando como lenguaje formal C++

//InterfaceAlumno.cpp

```
#include "InterfaceAlumno.h"
#include <iostream>
using namespace std;
InterfaceAlumno::InterfaceAlumno() {    }

string InterfaceAlumno::leerNombre ()
{string n;
cout << "Nombre: ";
cin >> n;
return n;}

string InterfaceAlumno::leerApellido()
{string a;
cout << "Apellido: ";
cin >> a;
return a;}
```

```
float InterfaceAlumno::leeNota1()
{ float n1;
cout<< "Introduzca la nota 1";
cin >> n1;
return n1;}
```

```
float InterfaceAlumno::leeNota2()
{ float n2;
cout<< "Introduzca la nota 2";
cin >> n2;
return n2;}
```

```
void InterfaceAlumno::reporteAlumno(string n,
string a, float p)
{
cout<< "REPORTE DEL ALUMNO\n";
cout<<"\nNombre:" << n;
cout<< "\nApellido:"<< a;
cout<< "\n\n Promedio:"<<p;
}
```


3. Resolución de Problemas empleando MOO

// Principal.cpp

FASE 3 Implementación (IOO) Usando como lenguaje formal C++

```
#include "Alumno.h"
#include "InterfaceAlumno.h"
#include <string>
using namespace std;
```

main() Es una función que se desarrolla a fin de indicar el orden en que irán ejecutando las instrucciones del proyecto.

int main ()

{ Alumno alu; **// Creación del objeto alu de la clase Alumno**

InterfaceAlumno intAlu;**// Creación del objeto IntAlu de la clase Interface Alumno**

string n,a;

float n1, n2;

// Se solicita al objeto de la clase InterfaceAlumno, nombre, apellido, nota 1 y nota 2

n = intAlu.leerNombre();

a = intAlu.leerApellido();

n1 = intAlu.leeNota1();

n2 = intAlu.leeNota2();

// Se le da estado al objeto de la clase Alumno, usando el método set de cada atributo

alu.setNombre(n);

alu.setApellido(a);

alu.setNota1(n1);

alu.setNota2(n2);

// Se pasa información al método del objeto de la clase InterfaceAlumno, para que la imprima el reporte

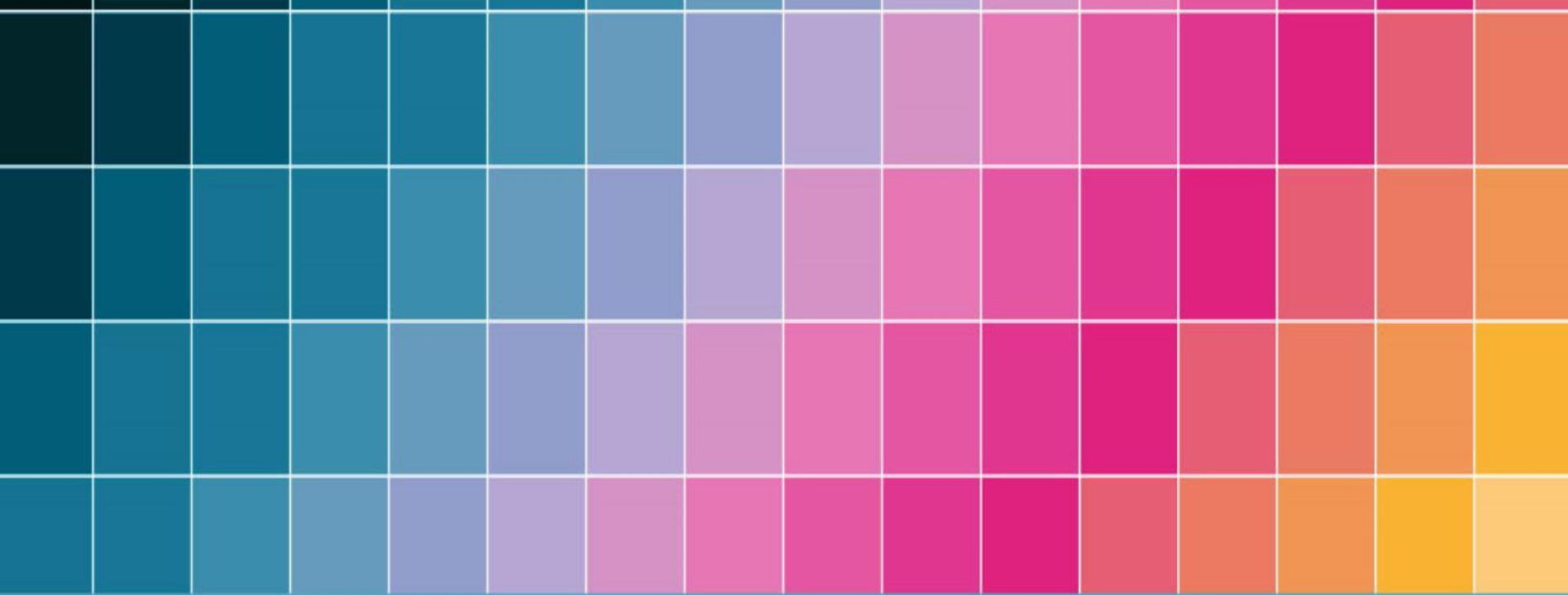
intAlu.reporteAlumno(alu.getNombre(),

alu.getApellido(),alu.calcularPromedio());

return 0; }

Ejercicios Propuestos usando la Metodología OO

1. De un vendedor se conoce el nombre, el monto de las ventas , sueldo base. Calcule la comisión que recibirá el vendedor que será del 17% del monto de las ventas. Calcule el sueldo final a percibir por el vendedor. Muestre el nombre la comisión y el sueldo final
2. Al ejercicio anterior añada los siguientes requerimientos: El vendedor recibirá un bono del 9% de su sueldo base si el monto de las ventas están por encima de 9000 \$. Calcule y muestre el monto del bono y el sueldo final a percibir por el vendedor.



Unidad IV Modelado Orientado a Objetos con Estructuras de Control Secuenciales y Selectivas

Tema 03 Resolución de problemas empleando la Metodología Orientada a Objeto

Introducción a la Computación