

Programacion II [PRACTICA]

1
0
0
1
0
1
0
1
0
1
0
1
0
1
0
0
0

1
0
1
0
0
1
0
1
0
1
0
0
1
0
1
0



CLASE

04



1
1
0
0
1
0
1
0
1
0
1
0
1
1
1
1
0



Biblioteca String



C++ ofrece una biblioteca estándar llamada `string` que proporciona una forma más moderna y segura de trabajar con cadenas. Esta biblioteca ofrece una amplia gama de funciones para manipular cadenas, como:

- ❑ `size()`: Obtiene la longitud de la cadena.
- ❑ `append(cadena)`: Concatena una cadena al final de la otra.
- ❑ `compare(cadena)`: Compara dos cadenas y devuelve un valor que indica si son iguales, menores o mayores.
- ❑ `find(subcadena)`: Busca la primera aparición de una subcadena dentro de otra.



LINK



LINK

Separar la Interfaz de la Implementación ●●

En el desarrollo de software, uno de los principios fundamentales para fomentar la reutilización y el mantenimiento del código es la separación de la interfaz de la implementación. Este enfoque es crucial para lograr una buena ingeniería de software.

La interfaz de una clase

Las interfaces definen y estandarizan las formas en las que las personas y los sistemas interactúan entre sí.

La interfaz de una clase describe qué servicios pueden usar los clientes de la clase y cómo solicitar esos servicios, pero no cómo lleva a cabo la clase esos servicios. La interfaz pública de una clase consiste en las funciones miembro pública de la clase (también conocidas como servicios públicos).



class.h



class.cpp



main.cpp

class.h: Definición de la interfaz de una clase mediante prototipos de funciones.

Un prototipo de función es una declaración de una función que indica al compilador el nombre de la función, su tipo de valor de retorno y los tipos de sus parámetros.

class.cpp: definir las funciones miembro en un archivo de código fuente separado .

clase.h



```
1  #include <string>
2  // definición de la clase LibroCalificaciones
3  using namespace std;
4  class LibroCalificaciones
5  {
6  public:
7      explicit LibroCalificaciones( std::string ); // constructor que inicializa a nombreCurso
8      void establecerNombreCurso( std::string ); // establece el nombre del curso
9      string obtenerNombreCurso() const; // obtiene el nombre del curso
10     void mostrarMensaje() const; // muestra un mensaje de bienvenida
11 private:
12     string nombreCurso; // nombre del curso para este LibroCalificaciones
13 }; // fin de la clase LibroCalificaciones
14
```

clase.cpp



```
1  #include <iostream>
2  #include "LibroCalificaciones.h" // incluye la definición de la clase LibroCalificaciones
3  using namespace std;
4  // el constructor inicializa nombreCurso con el objeto string suministrado como argumento
5  LibroCalificaciones::LibroCalificaciones( string nombre )
6      : nombreCurso( nombre ) // inicializador de miembro para inicializar nombreCurso
7  {
8      // cuerpo vacío
9  } // fin del constructor de LibroCalificaciones
10 // función para establecer el nombre del curso
11 void LibroCalificaciones::establecerNombreCurso( string nombre )
12 {
13     nombreCurso = nombre; // almacena el nombre del curso en el objeto
14 } // fin de la función establecerNombreCurso
15 // función para obtener el nombre del curso
16 string LibroCalificaciones::obtenerNombreCurso() const
17 {
18     return nombreCurso; // devuelve el nombreCurso del objeto
19 } // fin de la función obtenerNombreCurso
20 // muestra un mensaje de bienvenida al usuario de LibroCalificaciones
21 void LibroCalificaciones::mostrarMensaje() const
22 {
23     // llama a obtenerNombreCurso para obtener el nombreCurso
24     cout << "Bienvenido al libro de calificaciones para\n" << obtenerNombreCurso() << "!" << endl;
25 } // fin de la función mostrarMensaje
```

main.cpp



```
1  #include <iostream>
2  #include "LibroCalificaciones.h" // incluye la definición de la clase LibroCalificaciones GradeBook
3  using namespace std;
4  // la función main empieza la ejecución del programa
5  int main()
6  {
7  // crea dos objetos LibroCalificaciones
8  LibroCalificaciones libroCalificaciones1( "CS101 Introduccion a la programacion en C++" );
9  LibroCalificaciones libroCalificaciones2( "CS102 Estructuras de datos en C++" );
10 // muestra el valor inicial de nombreCurso para cada LibroCalificaciones
11 cout << "libroCalificaciones1 creado para el curso: " << libroCalificaciones1.obtenerNombreCurso() << endl;
12 cout << "\nlibroCalificaciones2 creado para el curso: " << libroCalificaciones2.obtenerNombreCurso() << endl;
13 } // fin de main
```

10010101001010



01 Crear Clases :



A. Desarrolla una clase **Cuenta Bancaria** que represente una cuenta bancaria. La clase debe incluir métodos para depositar dinero, retirar dinero, y consultar el saldo. Agrega un atributo para almacenar el número de cuenta y otro para el nombre del titular.

1
0
0
1
0
1
0
1
0
0
1
0
0

B. Crea una clase **Estudiante** que contenga los atributos nombre, apellido, edad, y una lista de calificaciones. Implementa métodos para calcular el promedio de calificaciones, y un método para determinar si el estudiante está aprobado (promedio ≥ 60).

1
0
1
0
0
1
0
0
1
1
0
1
1

C. Desarrolla una clase **Empleado** con atributos como nombre, ID, salario básico y horas trabajadas. La clase debe tener un método para calcular el salario total en función de las horas trabajadas y otro método para mostrar toda la información del empleado.

D. Desarrolla una clase **Aeropuerto** que contenga atributos como nombre, ciudad, y una lista de vuelos disponibles (cada vuelo tiene número de vuelo, destino, y hora de salida). Implementa métodos para agregar un vuelo, mostrar todos los vuelos disponibles, y buscar vuelos por destino

02 Clase Evento



Define una clase base Evento con atributos como nombre, fecha y lugar. Deriva clases como Concierto, Conferencia y Exposición, añadiendo atributos específicos (por ejemplo, artista, ponente, obras expuestas). Implementa métodos para registrar asistentes y generar reportes.

Clase Base Evento

Atributos:

- nombre: Cadena de caracteres que identifica el evento.
- fecha: Objeto de tipo Date o una estructura personalizada para representar la fecha y hora del evento.
- lugar: Cadena de caracteres que indica la ubicación del evento.

Método

- registrarAsistente(Asistente asistente): Agrega un objeto de tipo Asistente a una lista interna de asistentes.
- generarReporte(): Genera un reporte en formato texto o HTML que contenga información sobre el evento, como nombre, fecha, lugar y lista de asistentes.

03 Encuentra los errores ...



El siguiente programa tiene fallos. Encontrarlos, explicar cual se la causa, e indicar cómo se eliminaría el error.

```
1  #include <iostream.h>
2
3  class Circulo
4  {
5      int c_x, c_y;
6      float radio;
7  public:
8      void Circulo (int x, int y, float r)
9      {
10         c_x=x;
11         c_y=y;
12         radio=r;
13     }
14     void Visualizar(void)
15     {
16         cout << c_x << " " << c_y << " " << radio << "\n";
17     }
18
19     float Longitud(void)
20     {
21         return 3.14159*2*radio;
22     }
23 };
```

```
24 void main (void)
25 {
26     Circulo c1(5, 4, 4);
27     Circulo c2;
28     Circulo c3(5);
29
30     c1.Visualizar();
31     c2.Visualizar();
32     c3.Visualizar();
33
34     cout << c1.Longitud << "\n";
35 }
```