Universidad Mariano Gálvez De Guatemala

Facultad de ingeniería, Matemática y ciencias Físicas

Campus Villa Nueva Guatemala

Ingeniería en Sistemas de Información y ciencias de la computación

Curso: programación 1

Código de curso: 012

Código de la carrera: 5090

Estudiante: COOSEMANS Spiegeler Max Eduardo

Sección: A

Numero de carné: 5090-23-3282

Fecha: 08/03/2024

La introducción proporciona una visión general de las clases que serán implementadas para gestionar diferentes aspectos, como libros en una biblioteca, estudiantes, una calculadora simple, cuentas bancarias y películas.

Gestión de libros en una biblioteca:

Se crea una clase Libro con miembros privados para almacenar información como título, autor, año de publicación, etc. Se implementan métodos públicos para establecer y obtener estos datos, junto con un método para mostrar la información del libro.

Registro de estudiantes:

Se diseña una clase Estudiante con miembros privados para almacenar información como nombre, apellido, edad, curso, etc. Se incluyen métodos públicos para establecer y obtener estos datos, además de un método para mostrar la información del estudiante.

Calculadora simple:

Se crea una clase Calculadora que contiene métodos públicos para realizar operaciones básicas de suma, resta, multiplicación y división. Los números de entrada son miembros privados de la clase.

Gestión de cuentas bancarias:

Se diseña una clase CuentaBancaria que contiene información privada como el nombre del titular, número de cuenta, saldo, etc. Se implementan métodos públicos para depositar, retirar y consultar saldo.

Registro de películas:

Se crea una clase Pelicula que almacena información privada como título, director, año de lanzamiento, etc. Se incluyen métodos públicos para establecer y obtener estos datos, así como un método para mostrar la información de la película

#include <iostream>

#include <string>

class Libro {

private:

std::string titulo;

std::string autor;

int anioPublicacion;

public:

// Constructor

Libro(const std::string& t, const std::string& a, int anio)

: titulo(t), autor(a), anioPublicacion(anio) {}

// Métodos para establecer los datos

void setTitulo(const std::string& t) {

titulo = t;

}

void setAutor(const std::string& a) {

autor = a;

}

void setAnioPublicacion(int anio) {

anioPublicacion = anio;

}

// Métodos para obtener los datos

std::string getTitulo() const {

return titulo;

}

std::string getAutor() const {

return autor;

}

int getAnioPublicacion() const {

return anioPublicacion;

}

// Método para mostrar la información del libro

void mostrarInformacion() const {

std::cout << "Título: " << titulo << std::endl;

std::cout << "Autor: " << autor << std::endl;

std::cout << "Año de publicación: " << anioPublicacion << std::endl;

}

}

;

int main() {

// Crear un objeto de tipo Libro

Libro miLibro("El Gran Gatsby", "F. Scott Fitzgerald", 1925);

// Mostrar la información del libro

miLibro.mostrarInformacion();

// Cambiar el título y mostrar la información actualizada

miLibro.setTitulo("1984");

std::cout << "\nInformación actualizada:\n";

miLibro.mostrarInformacion();

return 0;

}

Texto

Descripción generada automáticamente

#include <iostream>

#include <string>

class Estudiante {

private:

std::string nombre;

std::string apellido;

int edad;

std::string curso;

public:

// Constructor

Estudiante(std::string nombre, std::string apellido, int edad, std::string curso) {

this->nombre = nombre;

this->apellido = apellido;

this->edad = edad;

this->curso = curso;

}

// Métodos públicos para establecer y obtener datos

void setNombre(std::string nombre) {

this->nombre = nombre;

}

std::string getNombre() {

return nombre;

}

void setApellido(std::string apellido) {

this->apellido = apellido;

}

std::string getApellido() {

return apellido;

}

void setEdad(int edad) {

this->edad = edad;

}

int getEdad() {

return edad;

}

void setCurso(std::string curso) {

this->curso = curso;

}

std::string getCurso() {

return curso;

}

// Método para mostrar información del estudiante

void mostrarInformacion() {

std::cout << "Nombre: " << nombre << std::endl;

std::cout << "Apellido: " << apellido << std::endl;

std::cout << "Edad: " << edad << std::endl;

std::cout << "Curso: " << curso << std::endl;

}

};

int main() {

// Crear un objeto Estudiante

Estudiante estudiante("Juan", "Perez", 20, "Informática");

// Mostrar información del estudiante

estudiante.mostrarInformacion();

// Cambiar la edad del estudiante

estudiante.setEdad(21);

// Mostrar información actualizada del estudiante

estudiante.mostrarInformacion();

return 0;

}

Captura de pantalla de computadora

Descripción generada automáticamente

#include <iostream>

class Calculadora {

private:

double numero1;

double numero2;

public:

// Constructor

Calculadora(double num1, double num2) {

numero1 = num1;

numero2 = num2;

}

// Método para sumar

double sumar() {

return numero1 + numero2;

}

// Método para restar

double restar() {

return numero1 - numero2;

}

// Método para multiplicar

double multiplicar() {

return numero1 \* numero2;

}

// Método para dividir

double dividir() {

if (numero2 != 0) {

return numero1 / numero2;

} else {

std::cerr << "Error: No se puede dividir por cero." << std::endl;

return 0; // Se devuelve 0 en caso de división por cero

}

}

};

int main() {

// Crear un objeto Calculadora

Calculadora calc(10, 5);

// Realizar operaciones

std::cout << "Suma: " << calc.sumar() << std::endl;

std::cout << "Resta: " << calc.restar() << std::endl;

std::cout << "Multiplicación: " << calc.multiplicar() << std::endl;

std::cout << "División: " << calc.dividir() << std::endl;

return 0;

}

Captura de pantalla de computadora

Descripción generada automáticamente

#include <iostream>

#include <string>

class CuentaBancaria {

private:

std::string nombreTitular;

std::string numeroCuenta;

double saldo;

public:

// Constructor

CuentaBancaria(std::string nombre, std::string cuenta, double saldoInicial) {

nombreTitular = nombre;

numeroCuenta = cuenta;

saldo = saldoInicial;

}

// Método para depositar dinero

void depositar(double cantidad) {

saldo += cantidad;

std::cout << "Se han depositado $" << cantidad << ". Saldo actual: $" << saldo << std::endl;

}

// Método para retirar dinero

void retirar(double cantidad) {

if (cantidad <= saldo) {

saldo -= cantidad;

std::cout << "Se han retirado $" << cantidad << ". Saldo actual: $" << saldo << std::endl;

} else {

std::cout << "No hay suficiente saldo para realizar el retiro." << std::endl;

}

}

// Método para consultar saldo

double consultarSaldo() {

return saldo;

}

};

int main() {

// Crear una cuenta bancaria

CuentaBancaria cuenta("Juan Perez", "123456789", 1000);

// Realizar operaciones

cuenta.depositar(500);

cuenta.retirar(200);

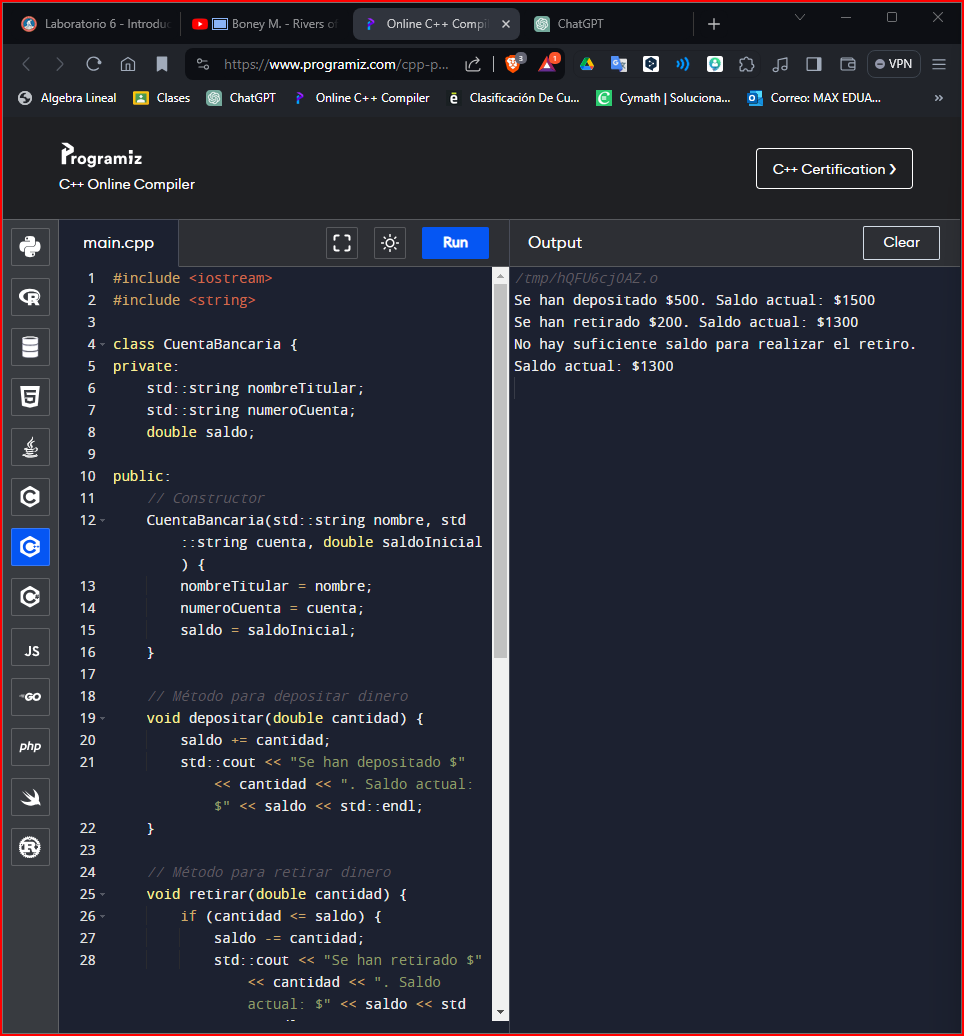
cuenta.retirar(2000); // Intentar retirar más dinero del disponible

// Consultar saldo

std::cout << "Saldo actual: $" << cuenta.consultarSaldo() << std::endl;

return 0;

}



#include <iostream>

#include <string>

class Pelicula {

private:

std::string titulo;

std::string director;

int anioLanzamiento;

public:

// Constructor

Pelicula(std::string titulo, std::string director, int anioLanzamiento) {

this->titulo = titulo;

this->director = director;

this->anioLanzamiento = anioLanzamiento;

}

// Métodos para establecer datos

void setTitulo(std::string titulo) {

this->titulo = titulo;

}

void setDirector(std::string director) {

this->director = director;

}

void setAnioLanzamiento(int anioLanzamiento) {

this->anioLanzamiento = anioLanzamiento;

}

// Métodos para obtener datos

std::string getTitulo() {

return titulo;

}

std::string getDirector() {

return director;

}

int getAnioLanzamiento() {

return anioLanzamiento;

}

// Método para mostrar información de la película

void mostrarInformacion() {

std::cout << "Título: " << titulo << std::endl;

std::cout << "Director: " << director << std::endl;

std::cout << "Año de lanzamiento: " << anioLanzamiento << std::endl;

}

};

int main() {

// Crear una película

Pelicula pelicula("El padrino", "Francis Ford Coppola", 1972);

// Mostrar información de la película

pelicula.mostrarInformacion();

// Cambiar el director de la película

pelicula.setDirector("Martin Scorsese");

// Mostrar información actualizada de la película

pelicula.mostrarInformacion();

return 0;

}

Texto

Descripción generada automáticamente

Este código contiene clases para la gestión de libros en una biblioteca, registro de estudiantes, una calculadora simple, gestión de cuentas bancarias y registro de películas. Cada clase tiene métodos para establecer y obtener datos, así como para mostrar la información relevante.