TRABAJO PRACTICO N°14  
1. Crea una clase Biblioteca que tenga una colección de objetos Libro. Implementa métodos para agregar, eliminar y buscar libros. Agrega un método para calcular el número total de páginas de todos los libros.

1. #include <iostream>

#include <vector>

using namespace std;

class Libro {

public:

string titulo;

int paginas;

Libro(string t, int p) : titulo(t), paginas(p) {}

void mostrar() {

cout << "Titulo: " << titulo << ", Paginas: " << paginas << endl;

}

};

class Biblioteca {

private:

vector<Libro> libros;

public:

void agregarLibro(Libro libro) {

libros.push\_back(libro);

}

void eliminarLibro(string titulo) {

for (vector<Libro>::iterator it = libros.begin(); it != libros.end(); ++it) {

if (it->titulo == titulo) {

libros.erase(it);

cout << "Libro '" << titulo << "' eliminado." << endl;

return;

}

}

cout << "Libro no encontrado." << endl;

}

void buscarLibro(string titulo) {

for (int i = 0; i < libros.size(); ++i) {

if (libros[i].titulo == titulo) {

libros[i].mostrar();

return;

}

}

cout << "Libro no encontrado." << endl;

}

int calcularTotalPaginas() {

int total = 0;

for (int i = 0; i < libros.size(); ++i) {

total += libros[i].paginas;

}

return total;

}

};

int main() {

Biblioteca biblioteca;

biblioteca.agregarLibro(Libro("C++ para principiantes", 300));

biblioteca.agregarLibro(Libro("JavaScript avanzado", 250));

biblioteca.buscarLibro("C++ para principiantes");

cout << "Total de paginas: " << biblioteca.calcularTotalPaginas() << endl;

biblioteca.eliminarLibro("JavaScript avanzado");

biblioteca.buscarLibro("JavaScript avanzado");

return 0;

}

2. Diseña una clase Restaurante con una colección de objetos Mesa. Cada Mesa tiene un número de comensales y un estado (ocupada, libre).Implementa métodos para reservar una mesa, liberar una mesa y obtener el número de mesas libres.

#include <iostream>

#include <vector>

using namespace std;

class Mesa {

public:

int comensales;

bool ocupada;

Mesa(int c) : comensales(c), ocupada(false) {}

void reservar() {

if (!ocupada) {

ocupada = true;

cout << "Mesa reservada." << endl;

} else {

cout << "Mesa ya ocupada." << endl;

}

}

void liberar() {

if (ocupada) {

ocupada = false;

cout << "Mesa liberada." << endl;

} else {

cout << "Mesa ya libre." << endl;

}

}

};

class Restaurante {

private:

vector<Mesa> mesas;

public:

void agregarMesa(int comensales) {

mesas.push\_back(Mesa(comensales));

}

int contarMesasLibres() {

int libres = 0;

for (int i = 0; i < mesas.size(); ++i) {

if (!mesas[i].ocupada) {

libres++;

}

}

return libres;

}

void reservarMesa(int index) {

if (index >= 0 && index < mesas.size()) {

mesas[index].reservar();

} else {

cout << "índice no valido." << endl;

}

}

void liberarMesa(int index) {

if (index >= 0 && index < mesas.size()) {

mesas[index].liberar();

} else {

cout << "índice no valido." << endl;

}

}

};

int main() {

Restaurante restaurante;

restaurante.agregarMesa(4);

restaurante.agregarMesa(2);

restaurante.agregarMesa(6);

cout << "Mesas libres: " << restaurante.contarMesasLibres() << endl;

restaurante.reservarMesa(0);

cout << "Mesas libres: " << restaurante.contarMesasLibres() << endl;

restaurante.liberarMesa(0);

cout << "Mesas libres: " << restaurante.contarMesasLibres() << endl;

return 0;

}

3. Crea una clase Empresa que tenga una colección de objetos Empleado. Cada Empleado tiene un nombre, salario y departamento. Implementa métodos para calcular la nómina total, encontrar al empleado con mayor salario y listar a los empleados de un departamento.

#include <iostream>

#include <vector>

using namespace std;

class Empleado {

public:

string nombre;

float salario;

string departamento;

Empleado(string n, float s, string d) : nombre(n), salario(s), departamento(d) {}

void mostrar() {

cout << "Empleado: " << nombre << ", Salario: " << salario << ", Departamento: " << departamento << endl;

}

};

class Empresa {

private:

vector<Empleado> empleados;

public:

void agregarEmpleado(Empleado emp) {

empleados.push\_back(emp);

}

float calcularNomina() {

float total = 0;

for (int i = 0; i < empleados.size(); ++i) {

total += empleados[i].salario;

}

return total;

}

void empleadoConMayorSalario() {

if (empleados.empty()) {

cout << "No hay empleados." << endl;

return;

}

Empleado mayor = empleados[0];

for (int i = 1; i < empleados.size(); ++i) {

if (empleados[i].salario > mayor.salario) {

mayor = empleados[i];

}

}

cout << "Empleado con mayor salario: " << mayor.nombre << " - " << mayor.salario << endl;

}

void listarEmpleadosPorDepartamento(string departamento) {

bool encontrado = false;

for (int i = 0; i < empleados.size(); ++i) {

if (empleados[i].departamento == departamento) {

empleados[i].mostrar();

encontrado = true;

}

}

if (!encontrado) {

cout << "No hay empleados en el departamento " << departamento << endl;

}

}

};

int main() {

Empresa empresa;

empresa.agregarEmpleado(Empleado("Juan", 1500, "Ventas"));

empresa.agregarEmpleado(Empleado("Ana", 2000, "Marketing"));

cout << "Nomina total: " << empresa.calcularNomina() << endl;

empresa.empleadoConMayorSalario();

empresa.listarEmpleadosPorDepartamento("Ventas");

return 0;

}

4. Crea una clase base Figura con atributos como color y posición. Deriva clases como Circulo, Rectangulo y Triangulo, cada una con sus propios atributos y métodos para calcular el área y el perímetro. Sobrecarga el método calcularArea() para que pueda calcular el área de diferentes tipos de figuras.

#include <iostream>

#include <cmath>

using namespace std;

class Figura {

public:

string color;

int x, y;

Figura(string c, int x\_pos, int y\_pos) : color(c), x(x\_pos), y(y\_pos) {}

virtual void calcularArea() = 0;

virtual void calcularPerimetro() = 0;

};

class Circulo : public Figura {

public:

int radio;

Circulo(string c, int x\_pos, int y\_pos, int r) : Figura(c, x\_pos, y\_pos), radio(r) {}

void calcularArea() {

cout << "area del circulo: " << M\_PI \* radio \* radio << endl;

}

void calcularPerimetro() {

cout << "Perimetro del circulo: " << 2 \* M\_PI \* radio << endl;

}

};

class Rectangulo : public Figura {

public:

int ancho, alto;

Rectangulo(string c, int x\_pos, int y\_pos, int a, int h) : Figura(c, x\_pos, y\_pos), ancho(a), alto(h) {}

void calcularArea() {

cout << "area del rectangulo: " << ancho \* alto << endl;

}

void calcularPerimetro() {

cout << "Perimetro del rectangulo: " << 2 \* (ancho + alto) << endl;

}

};

class Triangulo : public Figura {

public:

int base, altura;

Triangulo(string c, int x\_pos, int y\_pos, int b, int h) : Figura(c, x\_pos, y\_pos), base(b), altura(h) {}

void calcularArea() {

cout << "area del triangulo: " << 0.5 \* base \* altura << endl;

}

void calcularPerimetro() {

cout << "Perimetro del tringulo: " << "No implementado" << endl;

}

};

int main() {

Circulo circulo("Rojo", 0, 0, 5);

circulo.calcularArea();

circulo.calcularPerimetro();

Rectangulo rectangulo("Azul", 0, 0, 4, 6);

rectangulo.calcularArea();

rectangulo.calcularPerimetro();

Triangulo triangulo("Verde", 0, 0, 3, 4);

triangulo.calcularArea();

triangulo.calcularPerimetro();

return 0;

}  
5. Implementa una clase Calculadora con métodos para realizar operaciones aritméticas básicas (+, -, \*, /). Sobrecarga estos métodos para que puedan operar con diferentes tipos de datos (enteros, flotantes).  
Crea una clase Comparador con un método comparar() que reciba dos objetos de cualquier tipo y determine si son iguales. Sobrecarga este método para comparar diferentes tipos de datos (enteros, cadenas, objetos personalizados). Implementa una clase Conversor con métodos para convertir entre diferentes unidades de medida (longitud, masa, temperatura).Sobrecarga estos métodos para permitir diferentes tipos de conversiones.

#include <iostream>

using namespace std;

class Calculadora {

public:

int sumar(int a, int b) { return a + b; }

float sumar(float a, float b) { return a + b; }

int restar(int a, int b) { return a - b; }

float restar(float a, float b) { return a - b; }

int multiplicar(int a, int b) { return a \* b; }

float multiplicar(float a, float b) { return a \* b; }

int dividir(int a, int b) { return a / b; }

float dividir(float a, float b) { return a / b; }

};

int main() {

Calculadora calc;

cout << "Suma (int): " << calc.sumar(10, 5) << endl;

cout << "Suma (float): " << calc.sumar(10.5f, 5.2f) << endl;

cout << "Resta (int): " << calc.restar(10, 5) << endl;

cout << "Resta (float): " << calc.restar(10.5f, 5.2f) << endl;

cout << "Multiplicacion (int): " << calc.multiplicar(10, 5) << endl;

cout << "Multiplicacion (float): " << calc.multiplicar(10.5f, 5.2f) << endl;

cout << "Division (int): " << calc.dividir(10, 5) << endl;

cout << "Division (float): " << calc.dividir(10.5f, 5.2f) << endl;

return 0;

}