TP 10 VECTORES DE CLASES:

PUNTO:1

#include <iostream>

#include <vector>

#include <string>

class Alumno {

public:

std::string nombre;

float nota1, nota2, nota3;

void ingresarDatos() {

std::cout << "Ingrese el nombre del alumno: ";

std::cin >> nombre;

std::cout << "Ingrese nota 1: ";

std::cin >> nota1;

std::cout << "Ingrese nota 2: ";

std::cin >> nota2;

std::cout << "Ingrese nota 3: ";

std::cin >> nota3;

}

void mostrarDatos() const {

std::cout << "Nombre: " << nombre

<< ", Nota 1: " << nota1

<< ", Nota 2: " << nota2

<< ", Nota 3: " << nota3 << std::endl;

}

};

int main() {

std::vector<Alumno> alumnos;

int cantidad;

std::cout << "Ingrese la cantidad de alumnos: ";

std::cin >> cantidad;

for (int i = 0; i < cantidad; ++i) {

Alumno a;

a.ingresarDatos();

alumnos.push\_back(a);

}

std::cout << "\nDatos de los alumnos:\n";

for (const auto& alumno : alumnos) {

alumno.mostrarDatos();

}

return 0;

}

PUNTO:2

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

class Persona {

public:

std::string nombre;

int edad;

void ingresarDatos() {

std::cout << "Ingrese el nombre: ";

std::cin >> nombre;

std::cout << "Ingrese la edad: ";

std::cin >> edad;

}

void mostrarDatos() const {

std::cout << "Nombre: " << nombre << ", Edad: " << edad << std::endl;

}

};

bool compararPorEdad(const Persona &p1, const Persona &p2) {

return p1.edad < p2.edad;

}

int main() {

std::vector<Persona> personas;

int cantidad;

std::cout << "Ingrese la cantidad de personas: ";

std::cin >> cantidad;

for (int i = 0; i < cantidad; ++i) {

Persona p;

p.ingresarDatos();

personas.push\_back(p);

}

std::sort(personas.begin(), personas.end(), compararPorEdad);

std::cout << "\nDatos de las personas ordenadas por edad:\n";

for (const auto& persona : personas) {

persona.mostrarDatos();

}

return 0;

}

PUNTO:3.

#include <iostream>

#include <vector>

class Rectangulo {

public:

float base;

float altura;

void ingresarDatos() {

std::cout << "Ingrese la base: ";

std::cin >> base;

std::cout << "Ingrese la altura: ";

std::cin >> altura;

}

float area() const {

return base \* altura;

}

float perimetro() const {

return 2 \* (base + altura);

}

};

int main() {

std::vector<Rectangulo> rectangulos;

int cantidad;

std::cout << "Ingrese la cantidad de rectángulos: ";

std::cin >> cantidad;

for (int i = 0; i < cantidad; ++i) {

Rectangulo r;

r.ingresarDatos();

rectangulos.push\_back(r);

}

std::cout << "\nÁrea y perímetro de los rectángulos:\n";

for (const auto& rectangulo : rectangulos) {

std::cout << "Área: " << rectangulo.area()

<< ", Perímetro: " << rectangulo.perimetro() << std::endl;

}

return 0;

}

PUNTO:4

#include <iostream>

#include <vector>

class CuentaBancaria {

public:

std::string numeroCuenta;

std::string nombreTitular;

float saldo;

void ingresarDatos() {

std::cout << "Ingrese el número de cuenta: ";

std::cin >> numeroCuenta;

std::cout << "Ingrese el nombre del titular: ";

std::cin >> nombreTitular;

std::cout << "Ingrese el saldo inicial: ";

std::cin >> saldo;

}

void mostrarDatos() const {

std::cout << "Número de cuenta: " << numeroCuenta

<< ", Titular: " << nombreTitular

<< ", Saldo: " << saldo << std::endl;

}

void depositar(float monto) {

saldo += monto;

}

void retirar(float monto) {

if (saldo >= monto) {

saldo -= monto;

} else {

std::cout << "Fondos insuficientes." << std::endl;

}

}

};

int main() {

std::vector<CuentaBancaria> cuentas;

int cantidad;

std::cout << "Ingrese la cantidad de cuentas bancarias: ";

std::cin >> cantidad;

for (int i = 0; i < cantidad; ++i) {

CuentaBancaria c;

c.ingresarDatos();

cuentas.push\_back(c);

}

std::cout << "\nDatos de las cuentas bancarias:\n";

for (const auto& cuenta : cuentas) {

cuenta.mostrarDatos();

}

std::string numeroCuenta;

float monto;

std::cout << "Ingrese el número de cuenta para hacer un depósito: ";

std::cin >> numeroCuenta;

std::cout << "Ingrese el monto a depositar: ";

std::cin >> monto;

for (auto& cuenta : cuentas) {

if (cuenta.numeroCuenta == numeroCuenta) {

cuenta.depositar(monto);

std::cout << "Depósito realizado." << std::endl;

break;

}

}

std::cout << "Ingrese el número de cuenta para hacer un retiro: ";

std::cin >> numeroCuenta;

std::cout << "Ingrese el monto a retirar: ";

std::cin >> monto;

for (auto& cuenta : cuentas) {

if (cuenta.numeroCuenta == numeroCuenta) {

cuenta.retirar(monto);

std::cout << "Retiro realizado." << std::endl;

break;

}

}

std::cout << "\nDatos actualizados de las cuentas bancarias:\n";

for (const auto& cuenta : cuentas) {

cuenta.mostrarDatos();

}

return 0;

}

PUNTO:5

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

class Empleado {

public:

std::string nombre;

float salario;

std::string departamento;

void ingresarDatos() {

std::cout << "Ingrese el nombre: ";

std::cin >> nombre;

std::cout << "Ingrese el salario: ";

std::cin >> salario;

std::cout << "Ingrese el departamento: ";

std::cin >> departamento;

}

void mostrarDatos() const {

std::cout << "Nombre: " << nombre

<< ", Salario: " << salario

<< ", Departamento: " << departamento << std::endl;

}

};

bool compararPorSalario(const Empleado &e1, const Empleado &e2) {

return e1.salario > e2.salario;

}

int main() {

std::vector<Empleado> empleados;

int cantidad;

std::cout << "Ingrese la cantidad de empleados: ";

std::cin >> cantidad;

for (int i = 0; i < cantidad; ++i) {

Empleado e;

e.ingresarDatos();

empleados.push\_back(e);

}

std::sort(empleados.begin(), empleados.end(), compararPorSalario);

std::cout << "\nDatos de los empleados ordenados por salario (mayor a menor):\n";

for (const auto& empleado : empleados) {

empleado.mostrarDatos();

}

return 0;

}

PUNTO:6

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

class Libro {

public:

std::string titulo;

std::string autor;

int anioPublicacion;

void ingresarDatos() {

std::cout << "Ingrese el título: ";

std::cin >> titulo;

std::cout << "Ingrese el autor: ";

std::cin >> autor;

std::cout << "Ingrese el año de publicación: ";

std::cin >> anioPublicacion;

}

void mostrarDatos() const {

std::cout << "Título: " << titulo << ", Autor: " << autor<< ",