Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

**Лабораторна робота №2**

**Тема:** «Конструктори і деструктори»

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Гончаренко В.В |
| Перевірила:  Казірова Н.Л. |
|  |

Кропивницький

2024

**ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ**

**Лабораторна робота №2**

**Конструктори і деструктори**

**Мета:** ознайомитись з основними поняттями конструктор і деструктор в ООП та навчитись їх програмно реалізовувати мовою С++.

**Завдання**

**Завдання 1 1.**

Реалізуйте конструктор за замовчуванням, конструктор з параметрами та копіюючий конструктор для вашого класу з лабораторної роботи

1. • Конструктор за замовчуванням має встановлювати значення полів за замовчуванням.

• Конструктор з параметрами має приймати значення для кожного поля. • Копіюючий конструктор має копіювати значення полів з іншого об'єкта класу. 2. Реалізуйте деструктор для класу. Деструктор має виводити повідомлення про знищення об'єкта.

3. У функції main створіть об'єкт за допомогою конструктора за замовчуванням та виведіть значення його полів.

4. Створіть новий об'єкт за допомогою конструктора з параметрами та встановіть значення для полів. Виведіть значення полів цього об'єкта.

5. Створіть ще один об'єкт і скопіюйте значення полів з першого об'єкта за допомогою копіюючого конструктора. Виведіть значення полів цього об'єкта.

6. Завершіть функцію main, що призведе до виходу з області видимості створених об'єктів і виклику їх деструкторів. Переконайтесь, що повідомлення про знищення об'єктів виводяться.

**Завдання 2**

**Варіант 2**

Розробіть клас «Вектор» – Vector розмірності 𝑛. Клас повинен містити конструктори. Реалізуйте методи для обчислення модуля вектора, скалярного добутку, додавання, віднімання, множення на константу. Створіть масив об'єктів. Напишіть функцію, яка для заданої пари векторів буде визначати, чи є вони колінеарними або ортогональними.

**-- Лістинг --**

**Vector.h**

#ifndef VECTOR\_H

#define VECTOR\_H

#include <iostream>

#include <cmath>

#include <vector>

class Vector {

private:

std::vector<double> components;

int size;

public:

Vector(int n);

Vector(const std::vector<double>& components);

Vector(const Vector& other);

double magnitude() const;

double dotProduct(const Vector& other) const;

Vector add(const Vector& other) const;

Vector subtract(const Vector& other) const;

Vector multiplyByConstant(double scalar) const;

bool isCollinear(const Vector& other) const;

bool isOrthogonal(const Vector& other) const;

void display() const;

Vector& operator=(const Vector& other);

};

#endif // VECTOR\_H

**Vector.cpp**

#include "Vector.h"

Vector::Vector(int n) : size(n) {

components.resize(n, 0.0);

}

Vector::Vector(const std::vector<double>& components) : components(components), size(components.size()) {}

Vector::Vector(const Vector& other) : components(other.components), size(other.size) {}

double Vector::magnitude() const {

double sum = 0;

for (double comp : components) {

sum += comp \* comp;

}

return sqrt(sum);

}

double Vector::dotProduct(const Vector& other) const {

double result = 0;

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

result += components[i] \* other.components[i];

}

return result;

}

Vector Vector::add(const Vector& other) const {

std::vector<double> result(size);

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

result[i] = components[i] + other.components[i];

}

return Vector(result);

}

Vector Vector::subtract(const Vector& other) const {

std::vector<double> result(size);

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

result[i] = components[i] - other.components[i];

}

return Vector(result);

}

Vector Vector::multiplyByConstant(double scalar) const {

std::vector<double> result(size);

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

result[i] = components[i] \* scalar;

}

return Vector(result);

}

// Перевірка на колінеарність (якщо вектори пропорційні)

bool Vector::isCollinear(const Vector& other) const {

double ratio = components[0] / other.components[0];

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

if (components[i] / other.components[i] != ratio) {

return false;

}

}

return true;

}

// Перевірка на ортогональність (скалярний добуток дорівнює 0)

bool Vector::isOrthogonal(const Vector& other) const {

return dotProduct(other) == 0;

}

void Vector::display() const {

std::cout << "(";

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

std::cout << components[i];

if (i < size - 1) {

std::cout << ", ";

}

}

std::cout << ")\n";

}

Vector& Vector::operator=(const Vector& other) {

if (this != &other) {

components = other.components;

size = other.size;

}

return \*this;

}

**Main.cpp**

#include "Vector.h"

#include <iostream>

int main() {

Vector v1({1.0, 2.0, 3.0});

Vector v2({2.0, 4.0, 6.0});

std::cout << "Вектор v1: ";

v1.display();

std::cout << "Вектор v2: ";

v2.display();

std::cout << "Модуль вектора v1: " << v1.magnitude() << "\n";

std::cout << "Скалярний добуток v1 і v2: " << v1.dotProduct(v2) << "\n";

Vector v3 = v1.add(v2);

std::cout << "v1 + v2 = ";

v3.display();

Vector v4 = v1.subtract(v2);

std::cout << "v1 - v2 = ";

v4.display();

Vector v5 = v1.multiplyByConstant(3.0);

std::cout << "v1 \* 3 = ";

v5.display();

if (v1.isCollinear(v2)) {

std::cout << "Вектори v1 і v2 є колінеарними\n";

} else {

std::cout << "Вектори v1 і v2 не є колінеарними\n";

}

if (v1.isOrthogonal(v2)) {

std::cout << "Вектори v1 і v2 є ортогональними\n";

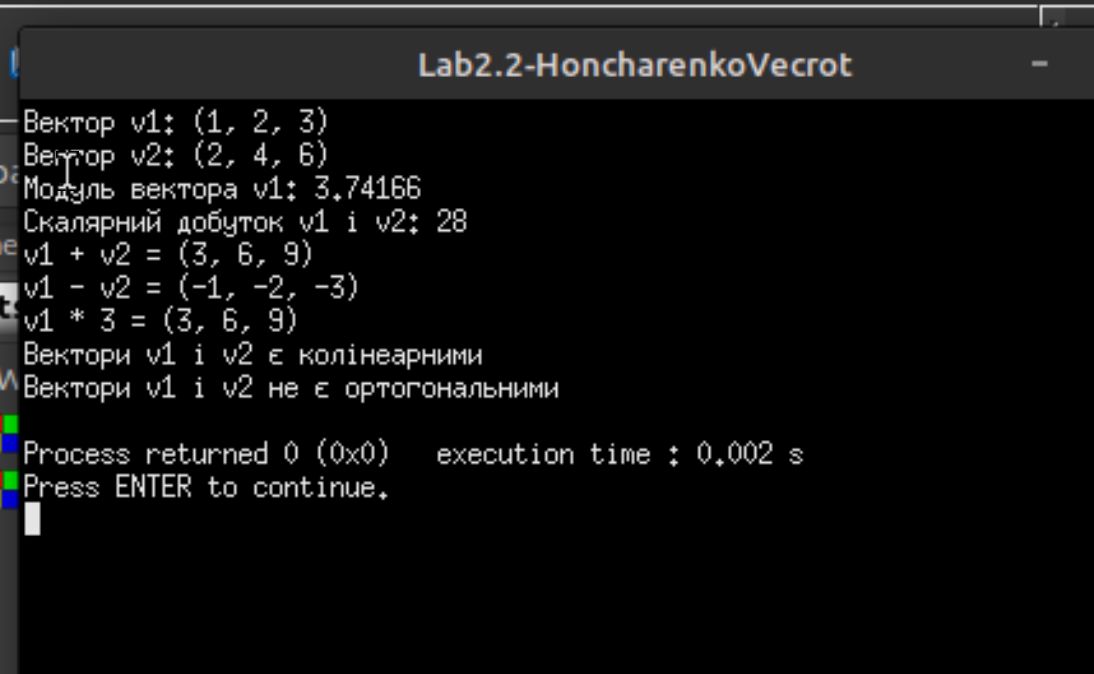
} else {

std::cout << "Вектори v1 і v2 не є ортогональними\n";

}

return 0;

}



**Висновок:**

У цій роботі було розроблено клас «Вектор», який містить необхідні методи для виконання базових операцій над векторами, таких як обчислення модуля вектора, скалярного добутку, додавання, віднімання та множення на константу. Також були реалізовані методи для перевірки колінеарності та ортогональності векторів. Клас дозволяє наочно працювати з векторами та здійснювати математичні операції, важливі для розв'язання різноманітних задач лінійної алгебри.