МІНІСТЕРСВО ОСВІТИ І НАУКИ УКРАЇНИ

Технічний коледж

НУ «Львівська політехніка»

Відділення

Інформаційних технологій

та комп`ютерної техніки

Звіт про виконання завдань

З предмету навчальна практика з основ програмування

та алгоритмічних мов.

Підготував

Студент групи 21-ПЗ

Гула Володимир

Львів 2021

Лабораторна робота № 2. Перевантаження операторів С++.

**Мета роботи:** Ознайомитись з поняттям та призначенням дружніх функцій та конструктора копіювання. Навчитися реалізовувати перевантаження операторів.

**Завдання до лабораторної роботи №2**

Описати клас, що реалізовує вказаний нижче тип даних. Клас повинен містити множину конструкторів для створення об'єктів певного типу (конструктор по замочуванню та з параметрами, конструктор копії) та подані у таблиці операції над об’єктами класу (плюс обов’язково операції присвоювання та порівняння) з використанням механізму перевантаження операцій:

Варіант №6 „вектор у просторі” - віднімання та складання векторів, порівняння векторів, скалярний добуток векторів

Головна програма:  
#include <iostream>

#include "vector.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int size;

cout << "Скiльки векторiв ви хочете ввести?\n";

cin >> size;

vector\* line = new vector[size];

for (int i = 0; i < size; i++) { line[i].inputVector(); }

cout << "Усi введенi вектори\n";

for (int i = 0; i < size; i++) { line[i].showVector(); }

cout << "Робота над векторами!\n";

cout << "Виберіть напрямок\n1 - додавання/віднімання\n2 - Порівняння/скалярний добуток\n";

int ans;

cin >> ans;

if (ans == 1) { line->DoVector(line, size); }

else if (ans == 2) { line->DoVector(line, size, true); }

else { cout << "Некоректна відповідь!\n"; }

}

Оголошення класу:  
#pragma once

class vector

{

private:

double x, y, z;

int countVector;

void input();

void show();

void WorkOnVector(vector\* Data, int size);

void WorkOnVector(vector\* Data, int size, int ans);

public:

vector();

vector(double xA, double yA, double zA, double xB, double yB, double zB, int count);

vector(vector& other);

~vector();

void setAll(double xA, double yA, double zA, double xB, double yB, double zB, int count);

int getCount();

void inputVector(bool ans = false);

void showVector(bool ans = false);

void DoVector(vector\* data, int size, bool ans = false);

};

Опис методів класу:  
#include <iostream>

#include "vector.h"

using namespace std;

void vector::input() {

double xBegin, yBegin, zBegin;

double xEnd, yEnd, zEnd;

cout << "Введiть номер вектора:\n";

cin >> countVector;

cout << "Введiть початковi точки:\n";

cout << "Початковий X: ";

cin >> xBegin;

cout << "Початковий Y: ";

cin >> yBegin;

cout << "Початковий Z: ";

cin >> zBegin;

cout << "Введiть кiнцевi точки\n";

cout << "Кiнцеве X: ";

cin >> xEnd;

cout << "Кiнцеве Y: ";

cin >> yEnd;

cout << "Кiнцеве Z: ";

cin >> zEnd;

setAll(xBegin, yBegin, zBegin, xEnd, yEnd, zEnd, countVector);

cout << endl;

}

void vector::show() {

cout << "Номер вектора: " << countVector << endl;

cout << "Координати вектора: " << "(" << x << ";" << y << ";" << z << ")\n";

cout << endl;

}

void vector::WorkOnVector(vector\* data, int size) {

int v1, v2, ans;

cout << "Виберiть функцiю\n1 - додавання векторiв\n2 - вiднiмання векторiв\n";

cin >> ans;

if (ans == 1) {

cout << "Введiть номер 1 доданку\n";

cin >> v1;

cout << "Введiть номер 2 доданку\n";

cin >> v2;

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

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

if (data[i].countVector == v1 && data[j].countVector == v2) {

cout << "Результат додавання: ";

cout << "(" << data[i].x + data[j].x << ";" << data[i].y + data[j].y << ";" << data[i].z + data[j].z << ")\n";

}

}

}

}

else if (ans == 2) {

cout << "Введiть номер зменшуваного\n";

cin >> v1;

cout << "Введiть номер вiд'ємника\n";

cin >> v2;

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

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

if (data[i].countVector == v1 && data[j].countVector == v2) {

cout << "Результат вiднiмання: ";

cout << "(" << data[i].x - data[j].x << ";" << data[i].y - data[j].y << ";" << data[i].z - data[j].z << ")\n";

}

}

}

}

else { cout << "Не коректна вiдповiдь!\n"; }

}

void vector::WorkOnVector(vector\* data, int size, int ans) {

int v1, v2;

if (ans == 1) {

cout << "Введiть номер першого вектора\n";

cin >> v1;

cout << "Введiть номер другого вектора\n";

cin >> v2;

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

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

if (data[i].countVector == v1 && data[j].countVector == v2) {

if (data[i].x == data[j].x && data[i].y == data[j].y && data[i].z == data[j].z) {

cout << "Вектори " << v1 << " i " << v2 << " рiвнi";

}

else if (data[i].x == (-data[j].x) && data[i].y == (-data[j].y) && data[i].z == (-data[j].z)) {

cout << "Вектори " << v1 << " i " << v2 << "протилежнi";

}

else { cout << "Вектори НЕ рiвнi"; }

}

else { cout << "Не коректне введення векторiв\n"; }

}

}

}

else if (ans == 2) {

cout << "Введiть номер першого вектора\n";

cin >> v1;

cout << "Введiть номер другого вектора\n";

cin >> v2;

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

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

if (data[i].countVector == v1 && data[j].countVector == v2) {

cout << "Результат скалярного добутку: ";

cout << "(" << data[i].x \* data[j].x << ";" << data[i].y \* data[j].y << ";" << data[i].z \* data[j].z << ")\n" ;

}

else { cout << "Не коректне введення векторiв\n"; }

}

}

}

else { cout << "Не коректна вiдповiдь!\n"; }

}

// Public mathods:

vector::vector() { setAll(0, 0, 0, 0, 0, 0, 0); }

vector::vector(double xA, double yA, double zA, double xB, double yB, double zB, int count) { setAll(xA, yA, zA, xB, yB, zB, count); }

vector::vector(vector& other) {

x = other.x;

y = other.y;

z = other.z;

countVector = other.countVector;

}

vector::~vector(){}

void vector::setAll(double xA, double yA, double zA, double xB, double yB, double zB, int count) {

x = xB - xA;

y = yB - yA;

z = zB - zA;

this->countVector = count;

}

int vector::getCount() {

return countVector + 1;

}

void vector::inputVector(bool ans) { input(); }

void vector::showVector(bool ans) { show(); }

void vector::DoVector(vector\* data, int size, bool ans) {

if (ans){

int answer;

cout << "Оберіть функцію\n1 - Порівняня векторів\n2 - Скалярний добуток векторів:\n";

cin >> answer;

WorkOnVector(data, size, answer);

}

else { WorkOnVector(data, size); }

}

Scrin:  
