

*Додаток 1*

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 2 з дисципліни  
«Основи програмування - 2. Модульне програмування»

«Класи та об'єкти»

Варіант 4

Виконав студент Бутов Даниїл Романович  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна  
(прізвище, ім'я, по батькові)

## Лабораторна робота 2

### Класи та об'єкти

**Мета** – вивчити механізми створення і використання класів та об'єктів.

### Варіант 4

4. Розробити клас "тривимірний вектор", який заданий своїми координатами. Створити масив об'єктів даного класу. Визначити вектор, який має найбільшу довжину.

### Особливості завдання.

Першим кроком було створення класу, який має назву "Three Dimensional Vector" , який отримує конструктором 3 змінних: значення x, y, z. Значення довжини буде знаходити метод set\_length() який встроєний до конструктора. Значення length, x, y, z є приватними у нашому класі, користувач може використати лише get\_values() - для повернення string у вигляді інформації про значення стартових значень та get\_length() - для повернення значення length.

Наступним кроком стане створення функцій для формування нашого масиву з об'єктами класу та його обробки. Функції розділені на -> 1. Створення масиву; 2. Виведення масиву; 3. Створення масиву з довжин векторів на основі першого масиву; 4. Пошук найбільшого за довжиною вектора.

Для зберігання об'єктів у формі масиву ми використовуємо динамічну структуру - вектор.

### Програма.

C++

main.cpp

```
#include "classes.h"

int main() {

    // Створення змінних
    vector<ThreeDimensionalVector> content;
    vector<double> vector_length;

    // Створення масиву з об'єктами класу
    content = create_base_with_vectors();
    output_vector_base( content_with_vectors: content);

    // Визначення довжини
    vector_length = set_vector_length( content_with_vectors: content);
    output_vector_length( content_with_length: vector_length);

    //Отримання найбільшої довжини
    find_the_biggest_length( content_with_length: vector_length);

}
```

### *classes.hpp*

```
#include <iostream>
#include <vector>
#include <cmath>

using namespace std;

// Ініціалізація класа тривимірного вектора

class ThreeDimensionalVector {
    int x, y, z;

    void set_length();

    double length();
public:
    ThreeDimensionalVector(int, int, int);

    void get_values() const;

    double get_length() const;
};

// Прототипи функцій для створення, обробки масиву об'єктів

vector<ThreeDimensionalVector> create_base_with_vectors();

void output_vector_base(const vector<ThreeDimensionalVector> &content_with_vectors);

void output_vector_length(const vector<double> &content_with_length);

vector<double> set_vector_length(const vector<ThreeDimensionalVector> &content_with_vectors);

void find_the_biggest_length(vector<double> content_with_length);
```

### *classes.cpp*

```
//
// Created by Danya Butov on 17.04.2022.
//

#include "classes.h"

// Методи класа тривимірного вектора

// Конструктор класа
ThreeDimensionalVector::ThreeDimensionalVector(int x_value, int y_value, int z_value) {
    x = x_value;
    y = y_value;
    z = z_value;
    set_length();
}

// Отримати координати вектора
void ThreeDimensionalVector::get_values() const {
    cout << "x= " << x << " "; y= " << y << " "; z= " << z << " "; << endl;
}

// Встановити значення довжини вектора
void ThreeDimensionalVector::set_length() {
    length = sqrt(pow( |cpp_x| x, |cpp_y| 2) + pow( |cpp_x| y, |cpp_y| 2) + pow( |cpp_x| z, |cpp_y| 2));
}

// Отримати значення довжини
double ThreeDimensionalVector::get_length() const {
    return length;
}
```

## functions.cpp

```
#include "classes.h"

// Функції для створення, обробки масиву об'єктів

// Створення вектора з об'єктами класу тривимірного вектора
vector<ThreeDimensionalVector> create_base_with_vectors() {
    vector<ThreeDimensionalVector> content_with_vectors;
    int count, x_values, y_values, z_values;

    cout << "How many vectors do u wanna create? ";
    cin >> count;
    for (int i = 0; i < count; ++i) {
        cout << to_string(val: i + 1) + "th Vector: " << endl;

        cout << "Enter x -> ";
        cin >> x_values;
        cout << "Enter y -> ";
        cin >> y_values;
        cout << "Enter z -> ";
        cin >> z_values;

        ThreeDimensionalVector Vector(x_values, y_values, z_values);
        content_with_vectors.push_back(Vector);
    }
    cout << endl;

    return content_with_vectors;
}
```

```
// Виведення вектора зі значеннями об'єктів класу
void output_vector_base(const vector<ThreeDimensionalVector> &content_with_vectors) {

    for (int i = 0; i < content_with_vectors.size(); ++i) {
        cout << to_string(val: i + 1) + "th Vector: ";
        content_with_vectors[i].get_values();
    }
}

// Створення вектора з довжин кожних об'єктів класу тривимірного вектора
vector<double> set_vector_length(const vector<ThreeDimensionalVector> &content_with_vectors) {
    vector<double> length_vector;

    length_vector.reserve(n: content_with_vectors.size());
    for (auto &content_with_vector : const ThreeDimensionalVector &: content_with_vectors) {
        length_vector.push_back(content_with_vector.get_length());
    }

    return length_vector;
}

// Визначення найбільшого значення довжини
void find_the_biggest_length(vector<double> content_with_length) {
    double max = content_with_length[0];
    int index;

    for (int i = 0; i < content_with_length.size(); ++i) {
        if (content_with_length[i] > max) {
            index = i;
            max = content_with_length[i];
        }
    }

    cout << "\nThe biggest length has " << to_string(val: index+1) << "th Vector with value: " << max << endl;
}
```

```
// Виведення значень довжини
void output_vector_length(const vector<double> &content_with_length) {

    cout << "\nVectors length: " << endl;
    for (int i = 0; i < content_with_length.size(); ++i) {
        cout << to_string(val: i + 1) + "th Vector: ";
        cout << content_with_length[i] << endl;
    }
}
```

### **Висновок.**

Я вивчив механізми створення і використання класів та об'єктів. Зрозумів основні принципи формування класу та його особливостей. На практиці зрозумів принципи абстракції та інкапсуляції при створенні класу, де були виділені деякий значення та методи. Створив програму у якій використовується клас для формування об'єкту за основою тривимірного вектора.