**Лабораторна робота №1**

**Розробка та реалізація програми з лінійною структурою**

**Мета роботи:** закріплення знань алфавіту мови программування C++; набуття навичок запису його констант, змінних, виразів, операторів присвоєння; оволодіння навичками складання програми з лінійною структурою.

**Завдання**

**Завдання 1.1.** Записати мовою C++ математичні вирази.

а) 10-4 e-2 f + | z3 | б) x + z < 0 ∪ 0 < f < 2

*Розв’язання*

а) 1.0e-4 \* exp(-2 \* f) + abs(pow(z, 3))

б) (x + z < 0) | | (f > 0 && f < 2)

**Завдання 1.2.** Представити математичний запис виразу і показати порядок дій.

(x+4)/3./x+sqrt(abs(x))/2.\*x+1.0e−6\*pow(x,1./3.)

*Розв’язання*

1 2 3 8 5 4 6 7 11 10 9

(x + 4) / 3. / x + sqrt(abs(x)) / 2. \* x + 1.0e−6 \* pow(x, 1./3.);

**Завдання 1.3.** Скласти програму знаходження модуля вектора **−2a+4b**, якщо a=(3; 2) и b=(0; −1).

*Розв’язання*

1. Постановка задачі

Скласти програму знаходження модуля вектора **−2a+4b**, якщо a=(3; 2) и b=(0; −1) на мові С++.

1. Методика розв’язання задачі

Для знаходженя модуля заданного вектора необхідно виконати дії:

* Множеня вектора на число (множуться всі координати вектора на це число)
* Додавання векторів (додаються відповідні координати)
* Знаходження модуля вектора (квадратний корінь з суми квадратів координат)

1. Алгоритм розв’язання задачі

Алгоритм розв’язання задачі можна представити у вигляді такої

послідовності:

Дія 1. Введення коондинат векторів.

Дія 2. Множення векторів a та b на -2 та 4 відповідно.

Дія 3. Знаходження сумми отриманих векторів.

Дія 4. Знаходження модуля суми векторів.

1. **Текст програми**

***main.cpp***

//Найти модуль вектора −2a+4b, если a=(3; 2) и b=(0; −1)

#include <iostream>

#include <Windows.h>

#include <cmath>

#include "Vector.h"

using namespace std;

using namespace vectors;

void setCharset();

void printHello();

void printResult(const double res);

void enterCoords(Vector &a, char mes[]);

double getCoords();

bool askRepeat();

int main(int argc, char\* argv[]) {

setCharset();

printHello();

Vector a, b;

while (true) {

enterCoords(a, "Введите координаты вектора а: ");

enterCoords(b, "Введите координаты вектора b: ");

a.mul(-2);

b.mul(4);

printResult( sumVectors(a, b, '+').abs() );

if (!askRepeat()) break;

}

return 0;

}

void setCharset() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

}

void printHello() {

cout << "\* \* \* |-2а + 4b| \* \* \* \n\n";

}

void printResult(const double res) {

cout << "|-2a + 4b| = " << res << endl << endl;

}

void enterCoords(Vector &a, char mes[]) {

cout << mes << endl;

cout << "x: ";

a.setX(getCoords());

cout << "y: ";

a.setY(getCoords());

}

double getCoords() {

while (true) {

double val;

cin >> val;

if (cin.fail()) {

cin.clear();

cin.ignore(32767, '\n');

cout << "Произошла ошибка. Введите еще раз: ";

} else {

cin.ignore(32767, '\n');

return val;

}

}

}

bool askRepeat() {

cout << "Еще раз (1 - да)? ";

char answer;

cin >> answer;

cin.ignore(32767, '\n');

return answer == '1';

}

**Vector.h**

#pragma once

namespace vectors {

class Vector {

private:

double x, y, z;

public:

Vector();

Vector(Vector &a, double k);

Vector(const double x, const double y);

Vector(const double x, const double y, const double z);

double getX();

double getY();

double getZ();

void setX(double x);

void setY(double y);

void setZ(double z);

void setCoords(Vector &a);

void setCoords(const double x, const double y);

void setCoords(const double x, const double y, const double z);

void mul(double k);

double abs();

~Vector();

};

//общие методы

Vector sumVectors(Vector &a, Vector &b, char type);

}

**Vector.cpp**

#include "Vector.h"

#include <cmath>

namespace vectors {

//Без параметров - все координаты - 0

Vector::Vector() {

setCoords(0, 0, 0);

}

//копирование с умножением на число

Vector::Vector(Vector &a, double k) {

setCoords(a.getX(), a.getY(), a.getZ());

mul(k);

}

Vector::Vector(const double x, const double y) {

setCoords(x, y);

}

Vector::Vector(const double x, const double y, const double z) {

setCoords(x, y, z);

}

//Возвращают cоответствующие координаты

double Vector::getX() {

return this->x;

}

double Vector::getY() {

return this->y;

}

double Vector::getZ() {

return this->z;

}

//Устанавливают соответствующие координаты

void Vector::setX(double x) {

this->x = x;

}

void Vector::setY(double y) {

this->y = y;

}

void Vector::setZ(double z) {

this->z = z;

}

//Устанавливает координаты равными переданном вектору

void Vector::setCoords(Vector &a) {

setX(a.getX());

setY(a.getY());

setZ(a.getZ());

}

void Vector::setCoords(const double x, const double y) {

setX(x);

setY(y);

}

void Vector::setCoords(const double x, const double y, const double z) {

setX(x);

setY(y);

setZ(z);

}

//Умножение вектора на число

void Vector::mul(const double k) {

setCoords(getX() \* k, getY() \* k, getZ() \* k);

}

double Vector::abs() {

return sqrt(pow(getX(), 2) + pow(getY(), 2) + pow(getZ(), 2));

}

Vector::~Vector() {

}

//Общие методы

//Сумирование векторов (вовращает результирующий вектор)

//Если k <= 0 - вычитание векторов

Vector sumVectors(Vector &a, Vector &b, char type) {

int k = (type == '+') ? 1 : -1;

Vector res(

a.getX() + k \* b.getX(),

a.getY() + k \* b.getY(),

a.getZ() + k \* a.getZ()

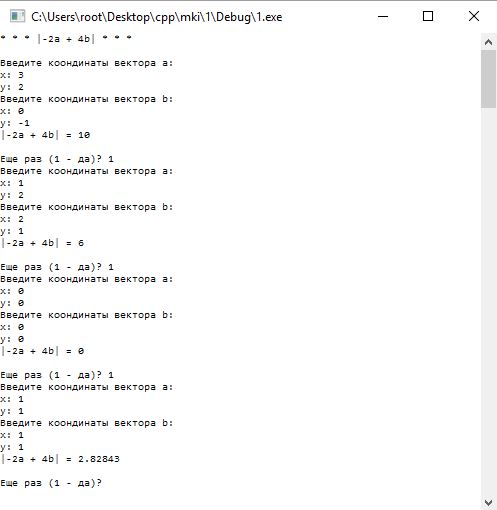
);

return res;

}

}

1. Результати роботи програми



**Висновок:** на цій лабораторній роботі я закріпив знання алфавіту мови програмування С++, набув навички запису його констант, змінних, виразів, оволодів навичками складання програми з лінійною структурою.