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 № 4 з дисципліни

«Основи програмування 2. Модульне програмування»

Варіант 25

Виконав студент ІП-14 Радзівіло Валерія Артемівна

(шифр, прізвище, ім'я, по батькові)

Перевірив

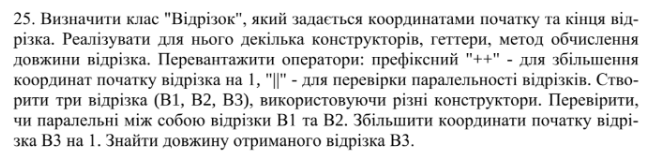
( прізвище, ім'я, по батькові)

Київ 2022

**Лабораторна робота №4**

**Тема роботи:** Перевантаження операторів

**Мета роботи:** Вивчити механізми створення класів з використанням перевантажених операторів(операцій).



**Вирішення задачі С++:**

**main:**

#include <iostream>  
#include "header.cpp"  
  
int main() {  
 int a = 1;  
 Line\*lines[3];  
 do {  
 Line line\_one(1,5, -2, -4);  
 cout << "First line"<< endl;  
 line\_one.print();  
 cout <<"Next line: "<< endl;  
 Line line\_two;  
 cout <<"Next line: "<< endl;  
 Line line\_three;  
 cout << endl << "RESULTS:" << endl;  
 // check if 2 lines are parallel  
 int check = line\_one || line\_two;  
 if (check == 1)  
 cout <<"The first line and second one are parallel!!!"<< endl;  
 // add 1 to third line's length using ++  
 cout<<"Length of third line before addding: " << line\_three.getLineLength() << endl;  
 cout<<"After adding 1: " << ++line\_three << endl;  
 // program can be re-used  
 cout <<endl << "Do you want to re-run the program?(print '0' if yes): ";  
 cin >> a;  
 }while(a==0);  
 return 0;  
  
}

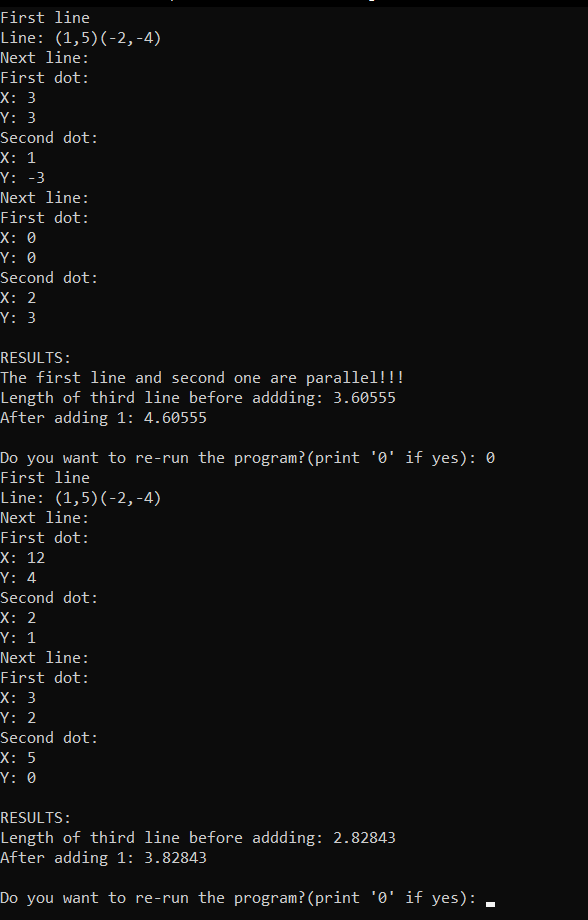
**header.h:**

#ifndef INC\_4LAB\_HEADER\_H  
#define INC\_4LAB\_HEADER\_H  
#include <iostream>  
#include <math.h>  
using namespace std;  
class Line{  
 private:  
 // dot A - start of line:  
 int startX, startY;  
 // dot B - end of line  
 int endX, endY;  
 float line\_length;  
 public:  
 Line(int startX, int endX,int startY, int endY);  
 Line();  
 // Getter  
 float getLineLength();  
 // print line as 2 coords  
 void print();  
 //overloading  
 float operator ++ ();  
 int operator ||(Line const &obj);  
};  
  
#endif //INC\_4LAB\_HEADER\_H

**header.cpp:**

#include "header.h"  
  
Line::Line(int startX, int endX,int startY, int endY){  
 this->startX = startX;  
 this->startY = startY;  
 this->endX = endX;  
 this->endY = endY;  
}  
Line::Line(){  
 cout<<"First dot:"<<endl<<"X: ";  
 cin>>startX;  
 cout<<"Y: ";  
 cin>>endX;  
 cout<<"Second dot:" <<endl<<"X: ";  
 cin>>startY;  
 cout<<"Y: ";  
 cin >>endY;  
}  
  
void Line::print() {  
 cout << "Line: "<<"("<<startX<<","<<endX<<")" <<"(" <<startY <<","<<endY<<")" << endl;  
}  
  
int Line::operator||(const Line &obj)  
{  
 float k\_one = (float)((endX - endY) /(startX - startY));  
 float k\_two = (float)((obj.endX - obj.endY) /(obj.startX - obj.startY));  
 if(k\_one == k\_two)  
 return 1;  
  
 else return 0;  
  
}  
  
float Line::operator++()  
{  
 line\_length += 1.0;  
 return line\_length;  
}  
  
float Line::getLineLength()  
{  
 line\_length =pow(pow(startY - startX, 2)+pow(endY - endX,2),0.5);  
 return line\_length;  
}

**Результат:**



**Висновок**: У цій лабораторній роботі було вивчено механізми створення класів з використанням перевантажених операторів(а саме : ++ та ||). Була постановлена задача, в якій визначалися 3 відрізки одного класу, один введений у програмі, інші введені користувачем. У алгоритмі були уточнені обмеження з використанням умов при яких задача буде виконана правильно. За допомоги цієї лабораторної роботи можливо виведення координат відрізка, рахування довжини відрізка, додавання 1 до довжини відрізка без складних операцій, два різні конструктори введення до класу, перевірка 2х відрізків на паралельність.