Міністерство освіти і науки України

Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформатика та програмної інженерії

Звіт

З лабораторної роботи № 3 з дисципліни “Основи програмування-2” “Класи та об’єкти”

Варіант 2

Виконав: ІП-13 Бабашев Олексій

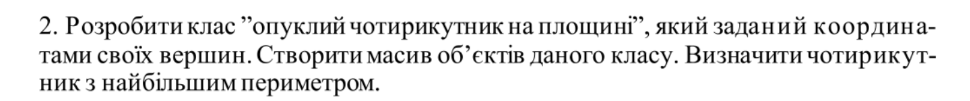
Перевірила: Вєчерковська А. С.

Київ - 2022

**Лабораторна робота 3**

**Класи та об’єкти**

**Індивідуальне завдання:**



**Виконання програми в терміналі та код на С++:**

**Вихідний код:**

**lab3.cpp**

#include "Choturukutnik.h"

int main()

{

int size;

cout << "Enter amount of chotirikutnik: "; cin >> size;

cout << "Enter clockwise or counter-clockwise coordinates!\n";

vector<Choturukutnik> MyArr = initilising\_Array(size);

ShowPerimetrOfChoturukutnik(MyArr);

int maxPerIndex = indexOfMaxPerimetr(MyArr);

cout << "The biggest perimetr is in " << maxPerIndex + 1 << " choturukutnik";

return 1;

}

**Choturukutnik.cpp**

#include "Choturukutnik.h"

Point::Point() {

x = 0;

y = 0;

}

Point::Point(int x, int y) {

this->x = x;

this->y = y;

}

int Point::get\_x() {

return x;

}

int Point::get\_y() {

return y;

}

double Choturukutnik::length(Point p1, Point p2) {

return sqrt(pow((p2.get\_x() - p1.get\_x()), 2) + pow((p2.get\_y() - p1.get\_y()), 2));

}

Choturukutnik::Choturukutnik() {}

Choturukutnik::Choturukutnik(Point p1, Point p2, Point p3, Point p4) {

point1 = p1;

point2 = p2;

point3 = p3;

point4 = p4;

}

bool Choturukutnik::isNotCorrect() {

float t1, t2, t3, t4;

t1 = (point4.get\_x() - point1.get\_x()) \* (point2.get\_y() - point1.get\_y()) -

(point4.get\_y() - point1.get\_y()) \* (point2.get\_x() - point1.get\_x());

t2 = (point4.get\_x() - point2.get\_x()) \* (point3.get\_y() - point2.get\_y()) -

(point4.get\_y() - point2.get\_y()) \* (point3.get\_x() - point2.get\_x());

t3 = (point4.get\_x() - point3.get\_x()) \* (point1.get\_y() - point3.get\_y()) -

(point4.get\_y() - point3.get\_y()) \* (point1.get\_x() - point3.get\_x());

t4 = (point1.get\_x() - point3.get\_x()) \* (point2.get\_y() - point1.get\_y()) -

(point1.get\_y() - point3.get\_y()) \* (point2.get\_x() - point1.get\_x());

if ((t1 \* t2 \* t3 \* t4) > 0) {

return true;

}

return false;

}

double Choturukutnik::perimetr() {

return length(point1, point2) + length(point2, point3) + length(point3, point4) + length(point4, point1);

}

**Functions.cpp**

#include "Choturukutnik.h"

vector<Point> SetPoints() {

int x, y;

vector<Point> res;

for (int i = 0; i < 4; i++)

{

cout << "x: "; cin >> x;

cout << "y: "; cin >> y;

res.push\_back(Point(x, y));

}

return res;

}

vector<Choturukutnik> initilising\_Array(int size) {

vector<Choturukutnik> res;

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

{

while (true)

{

vector<Point> points = SetPoints();

Choturukutnik temp(points[0], points[1], points[2], points[3]);

if (temp.isNotCorrect())

{

cout << "FAIL! TRY AGAIN!!!\n";

}

else

{

res.push\_back(temp);

cout << "======================\n";

break;

}

}

}

return res;

}

int indexOfMaxPerimetr(vector<Choturukutnik>& arr) {

int res = 0;

for (int i = 1; i < arr.size(); i++)

{

if (arr[i - 1].perimetr() < arr[i].perimetr()) {

res = i;

}

}

return res;

}

**Choturukutnik.h**

#pragma once

#define \_USE\_MATH\_DEFINES

#define TO\_DEGREES (180 / M\_PI)

#include <iostream>

#include<math.h>

#include<vector>

using namespace std;

class Point

{

private:

int x, y;

public:

Point();

Point(int x, int y);

int get\_x();

int get\_y();

};

class Choturukutnik

{

private:

Point point1, point2, point3, point4;

double length(Point p1, Point p2);

public:

Choturukutnik();

Choturukutnik(Point p1, Point p2, Point p3, Point p4);

bool isNotCorrect();

double perimetr();};

**Funtions.h**

#pragma once

#include "Choturukutnik.h"

vector<Point> SetPoints();

vector<Choturukutnik> initilising\_Array(int size);

# int indexOfMaxPerimetr(vector<Choturukutnik>& arr);

**Висновок:** Вивчив особливості створення і обробки класів та об’єктів. Застосував отриманні знання на практиці.