Московский Авиационный Институт

(Национальный исследовательский Университет)

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

**Лабораторная работа**

**по курсу «ООП»**

**Тема:**

**Простые классы.**

Студент:

Кузьмичев А. Н.

Группа:

М80-206Б-18

Преподаватель:

Журавлев А.А.

Вариант:

12

Оценка:

Дата:

Москва

2019**1. Код программы на языке C++:**

**Rectangle.h:**

#ifndef **LABA1\_HOME\_RECTANGLE\_H**

#define **LABA1\_HOME\_RECTANGLE\_H**

#include **<iostream>**

**class** Rectangle {

**private**:

**double** left\_lower\_x;

**double** left\_lower\_y;

**double** right\_upper\_x;

**double** right\_upper\_y;

**double** a;

**double** b;

**double** move\_x;

**double** move\_y;

**double** x\_size;

**public**:

Rectangle(**double** llx, **double** lly, **double** rrx, **double** rry) : left\_lower\_x(llx), left\_lower\_y(lly),

right\_upper\_x(rrx), right\_upper\_y(rry) {}

Rectangle() = **default**;

**double** square;

**double** perimetr;

**void** Input(); *// ввод с клавиатуры*

**void** Output(); *// вывод на экран*

**void** Square(); *// подсчет площади*

**void** Perimeter(); *// подсчет периметра*

**void** Move(); *// перемещение*

**void** Size(); *// изменение размеров*

**double** Compare(); *// сравнение по площади и периметру*

**void** Cross(); *// Пересечение двух прямоугольников*

**void** Min(); *// минимальный прямоугольник, включающий оба прямугольника*

};

#endif **LABA1\_HOME\_RECTANGLE\_H**

**Rectangle.cpp:**

#include **<iostream>**

#include **"Rectangle.h"**

**void** Rectangle::Input() {

std::cout << **"Введите координаты левой нижней точки"** << std::endl;

std::cin >> left\_lower\_x >> left\_lower\_y;

std::cout << **"Введите координаты правой верхней точки"** << std::endl;

std::cin >> right\_upper\_x >> right\_upper\_y;

}

**void** Rectangle::Output() {

std::cout << **"A - "** << **"("** << left\_lower\_x << **","** << left\_lower\_y << **")"** << std::endl;

std::cout << **"B - "** << **"("** << left\_lower\_x << **","** << right\_upper\_y << **")"** << std::endl;

std::cout << **"C - "** << **"("** << right\_upper\_x << **","** << right\_upper\_y << **")"** << std::endl;

std::cout << **"D - "** << **"("** << right\_upper\_x << **","** << left\_lower\_y << **")"** << std::endl;

}

**void** Rectangle::Square() {

std::cout << **"Введите координаты левой нижней точки"** << std::endl;

std::cin >> left\_lower\_x >> left\_lower\_y;

std::cout << **"Введите координаты правой верхней точки"** << std::endl;

std::cin >> right\_upper\_x >> right\_upper\_y;

**double** a = right\_upper\_y - left\_lower\_y;

**double** b = right\_upper\_x - left\_lower\_x;

**double** square = a \* b;

std::cout << **"Площадь прямоугольника равна: "** << square << std::endl;

}

**void** Rectangle::Perimeter() {

std::cout << **"Введите координаты левой нижней точки"** << std::endl;

std::cin >> left\_lower\_x >> left\_lower\_y;

std::cout << **"Введите координаты правой верхней точки"** << std::endl;

std::cin >> right\_upper\_x >> right\_upper\_y;

**double** a = right\_upper\_y - left\_lower\_y;

**double** b = right\_upper\_x - left\_lower\_x;

**double** perimeter = 2 \* (a + b);

std::cout << **"Периметр прямоугольника равен: "** << perimeter << std::endl;

}

**void** Rectangle::Move() {

std::cout << **"На сколько единиц вы хотите переместить прямоугольник по оси x?"** << std::endl;

std::cin >> move\_x;

std::cout << **"На сколько единиц вы хотите переместить прямоугольник по оси y?"** << std::endl;

std::cin >> move\_y;

std::cout << **"Координаты нового прямоугольника:"** << std::endl;

std::cout << **"A:"** << **"("** << left\_lower\_x + move\_x << **", "** << left\_lower\_y + move\_y << **")"** << std::endl;

std::cout << **"B:"** << **"("** << left\_lower\_x + move\_x << **", "** << right\_upper\_y + move\_y << **")"** << std::endl;

std::cout << **"C:"** << **"("** << right\_upper\_x + move\_x << **", "** << right\_upper\_y + move\_y << **")"** << std::endl;

std::cout << **"D:"** << **"("** << right\_upper\_x + move\_x << **", "** << left\_lower\_y + move\_y << **")"** << std::endl;

}

**void** Rectangle::Size() {

std::cout << **"Во сколько раз вы хотите изменить размеры прямоугольника?"** << std::endl;

std::cin >> x\_size;

std::cout << **"Координаты нового прямоугольника:"** << std::endl;

std::cout << **"A - "** << **"("** << left\_lower\_x \* x\_size << **","** << left\_lower\_y \* x\_size << **")"** << std::endl;

std::cout << **"B - "** << **"("** << left\_lower\_x \* x\_size << **","** << right\_upper\_y \* x\_size << **")"** << std::endl;

std::cout << **"C - "** << **"("** << right\_upper\_x \* x\_size << **","** << right\_upper\_y \*x\_size << **")"** << std::endl;

std::cout << **"D - "** << **"("** << right\_upper\_x \* x\_size << **","** << left\_lower\_y \* x\_size << **")"** << std::endl;

}

**double** Rectangle::Compare() {

**double** left\_lower\_x\_2, left\_lower\_y\_2, right\_upper\_x\_2, right\_upper\_y\_2, a\_2, b\_2;

**double** square1 = square;

**double** perimeter1 = perimetr;

std::cout << **"Введите координаты левой нижней точки второго прямоугольника"** << std::endl;

std::cin >> left\_lower\_x\_2 >> left\_lower\_y\_2;

std::cout << **"Введите координаты правой верхней точки второго прямоугольника"** << std::endl;

std::cin >> right\_upper\_x\_2 >> right\_upper\_y\_2;

a\_2 = right\_upper\_y\_2 - left\_lower\_y\_2;

b\_2 = right\_upper\_x\_2 - left\_lower\_x\_2;

**double** square2 = a\_2 \* b\_2;

**double** perimeter2 = (a\_2 + b\_2) \* 2;

**if** (square1 > square2) {

std::cout << **"Площадь первого прямоугольника больше площади второго"** << std::endl;

} **else if** (square1 < square2) {

std::cout << **"Площадь второго прямоугольника больше площади первого"** << std::endl;

} **else if** (square1 == square2) {

std::cout << **"Площадь первого и второго прямоугольника равны"** << std::endl;

}

**if** (perimeter1 > perimeter2) {

std::cout << **"Периметр первого прямоугольника больше периметра второго"** << std::endl;

} **else if** (perimeter1 < perimeter2) {

std::cout << **"Периметр второго прямоугольника больше периметра первого"** << std::endl;

} **else if** (perimeter1 == perimeter2) {

std::cout << **"Периметры первого и второго прямоугольника равны"** << std::endl;

}

}

**void** Rectangle::Min() {

**double** left\_lower\_x\_2, left\_lower\_y\_2, right\_upper\_x\_2, right\_upper\_y\_2, x\_max = left\_lower\_x, y\_max = left\_lower\_y, x\_min = left\_lower\_x , y\_min = left\_lower\_y;

std::cout << **"Введите координаты левой нижней точки второго прямоугольника"** << std::endl;

std::cin >> left\_lower\_x\_2 >> left\_lower\_y\_2;

std::cout << **"Введите координаты правой верхней точки второго прямоугольника"** << std::endl;

std::cin >> right\_upper\_x\_2 >> right\_upper\_y\_2;

std::cout << **"Координаты первого прямоугольника: "** << std::endl;

std::cout << **"A - "** << **"("** << left\_lower\_x << **","** << left\_lower\_y << **")"** << std::endl;

std::cout << **"B - "** << **"("** << left\_lower\_x << **","** << right\_upper\_y << **")"** << std::endl;

std::cout << **"C - "** << **"("** << right\_upper\_x << **","** << right\_upper\_y << **")"** << std::endl;

std::cout << **"D - "** << **"("** << right\_upper\_x << **","** << left\_lower\_y << **")"** << std::endl;

std::cout << **"Координаты второго прямоугольника: "** << std::endl;

std::cout << **"A - "** << **"("** << left\_lower\_x\_2 << **","** << left\_lower\_y\_2 << **")"** << std::endl;

std::cout << **"B - "** << **"("** << left\_lower\_x\_2 << **","** << right\_upper\_y\_2 << **")"** << std::endl;

std::cout << **"C - "** << **"("** << right\_upper\_x\_2 << **","** << right\_upper\_y\_2 << **")"** << std::endl;

std::cout << **"D - "** << **"("** << right\_upper\_x\_2 << **","** << left\_lower\_y\_2 << **")"** << std::endl;

**double** A[4] = {left\_lower\_x, right\_upper\_x, left\_lower\_x\_2, right\_upper\_x\_2}, B[4] = {left\_lower\_y, right\_upper\_y, left\_lower\_y\_2, right\_upper\_y\_2};

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

**if** (x\_min > A[i]) {

x\_min = A[i];

}

**if** (y\_min > A[i]) {

y\_min = A[i];

}

**if** (x\_max < A[i]) {

x\_max = A[i];

}

**if** (y\_max < A[i]) {

y\_max = A[i];

}

}

std::cout << **"Координаты наименьшего прямоугольника, содержащего два введенных прямоугольника:"** << std::endl;

std::cout << **"A - "** << **"("** << x\_min << **","** << y\_min << **")"** << std::endl;

std::cout << **"B - "** << **"("** << x\_min << **","** << y\_max << **")"** << std::endl;

std::cout << **"C - "** << **"("** << x\_max << **","** << y\_max << **")"** << std::endl;

std::cout << **"D - "** << **"("** << x\_max << **","** << y\_min << **")"** << std::endl;

}

**void** Rectangle::Cross() {

**double** left\_lower\_x\_2, left\_lower\_y\_2, right\_upper\_x\_2, right\_upper\_y\_2, x\_max = left\_lower\_x, y\_max = left\_lower\_y, x\_min = left\_lower\_x , y\_min = left\_lower\_y, x\_1\_middle, x\_2\_middle, y\_1\_middle, y\_2\_middle;

std::cout << **"Введите координаты левой нижней точки второго прямоугольника"** << std::endl;

std::cin >> left\_lower\_x\_2 >> left\_lower\_y\_2;

std::cout << **"Введите координаты правой верхней точки второго прямоугольника"** << std::endl;

std::cin >> right\_upper\_x\_2 >> right\_upper\_y\_2;

std::cout << **"Координаты первого прямоугольника: "** << std::endl;

std::cout << **"A - "** << **"("** << left\_lower\_x << **","** << left\_lower\_y << **")"** << std::endl;

std::cout << **"B - "** << **"("** << left\_lower\_x << **","** << right\_upper\_y << **")"** << std::endl;

std::cout << **"C - "** << **"("** << right\_upper\_x << **","** << right\_upper\_y << **")"** << std::endl;

std::cout << **"D - "** << **"("** << right\_upper\_x << **","** << left\_lower\_y << **")"** << std::endl;

std::cout << **"Координаты второго прямоугольника: "** << std::endl;

std::cout << **"A - "** << **"("** << left\_lower\_x\_2 << **","** << left\_lower\_y\_2 << **")"** << std::endl;

std::cout << **"B - "** << **"("** << left\_lower\_x\_2 << **","** << right\_upper\_y\_2 << **")"** << std::endl;

std::cout << **"C - "** << **"("** << right\_upper\_x\_2 << **","** << right\_upper\_y\_2 << **")"** << std::endl;

std::cout << **"D - "** << **"("** << right\_upper\_x\_2 << **","** << left\_lower\_y\_2 << **")"** << std::endl;

**double** A[4] = {left\_lower\_x, right\_upper\_x, left\_lower\_x\_2, right\_upper\_x\_2}, B[4] = {left\_lower\_y, right\_upper\_y, left\_lower\_y\_2, right\_upper\_y\_2};

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

**if** (x\_min > A[i]) {

x\_min = A[i];

}

**if** (y\_min > A[i]) {

y\_min = A[i];

}

**if** (x\_max < A[i]) {

x\_max = A[i];

}

**if** (y\_max < A[i]) {

y\_max = A[i];

}

}

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

**if** ((A[i] != x\_max) && (A[i] != x\_min)) {

x\_1\_middle = A[i];

**break**;

; }

}

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

**if** ((A[i] != x\_max) && (A[i] != x\_min) && (A[i] != x\_1\_middle)) {

x\_2\_middle = A[i];

**break**;

}

}

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

**if** ((A[i] != y\_max) && (A[i] != y\_min)) {

y\_1\_middle = A[i];

**break**;

}

}

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

**if** ((A[i] != y\_max) && (A[i] != y\_min) && (A[i] != y\_1\_middle)) {

y\_2\_middle = A[i];

**break**;

}

}

std::cout << **"Прямоугольник, представляющий общую часть двух прямоугольников, имеет координаты: "** << std::endl;

std::cout << **"A - "** << **"("** << x\_1\_middle << **","** << y\_2\_middle << **")"** << std::endl;

std::cout << **"B - "** << **"("** << x\_2\_middle << **","** << y\_2\_middle << **")"** << std::endl;

std::cout << **"C - "** << **"("** << x\_2\_middle << **","** << y\_1\_middle << **")"** << std::endl;

std::cout << **"D - "** << **"("** << x\_1\_middle << **","** << y\_1\_middle << **")"** << std::endl;

}

**Main.cpp:**

#include **<iostream>**

#include **"Rectangle.h"**

**int** main() {

std::cout << **"Меню: \n 1 - Ввод данных \n 2 - Вывод данных \n 3 - Подсчет площади \n 4 - Подсчет периметра \n 5 - Перемещение прямоугольника \n 6 - Изменение размеров \n 7 - Сравнение двух прмоугольников \n 8 - Пересечение двух прямоугольников \n 9 - Минимальный прямоугольник, влючающий оба прямоугольника \n 0 - выход\n"** << std::endl;

Rectangle Object1;

**bool** control=**true**;

**int** myCase=-1;

std::cout << **">"** << std::endl;

**while** (control) {

std::cin>>myCase;

**switch** (myCase) {

**case** 1: {

Object1.Input();

std::cout << **">"** << std::endl;

**break**;

}

**case** 2: {

Object1.Output();

std::cout << **">"** << std::endl;

**break**;

}

**case** 3: {

Object1.Square();

std::cout << **">"** << std::endl;

**break**;

}

**case** 4: {

Object1.Perimeter();

std::cout << **">"** << std::endl;

**break**;

}

**case** 5: {

Object1.Move();

std::cout << **">"** << std::endl;

**break**;

}

**case** 6: {

Object1.Size();

std::cout << **">"** << std::endl;

**break**;

}

**case** 7: {

Object1.Compare();

std::cout << **">"** << std::endl;

**break**;

}

**case** 8: {

Object1.Cross();

std::cout << **">"** << std::endl;

**break**;

}

**case** 9: {

Object1.Min();

std::cout << **">"** << std::endl;

**break**;

}

**case** 0:

{

control = **false**;

**break**;

}

}

}

**return** 0;

}

**CmakeLists.txt:**

cmake\_minimum\_required(**VERSION 3.12.3**)

project(**laba1\_home**)

add\_executable(**laba1\_home main.cpp Rectangle.cpp**)

set\_property(**TARGET laba1\_home PROPERTY CXX\_STANDART 14**)

set(**CMAKE\_CXX\_FLAGS** "**${CMAKE\_CXX\_FLAGS} -Wall -Wextra -v**")

**test.sh:**

executable=$1

for file in test\_??.test

do

$executable < $file > tmp

if cmp tmp ${file%%.test}.result

then

echo Test "$file": SUCCESS

else

echo Test "$file": FAIL

fi

rm tmp

done

**2. Ссылка на репозиторий на GitHub.**

https://github.com/poisoned-monkey/oop\_exercise\_01.git

**3. Набор testcases.**

Test 1

A - (2,2)

B - (2,4)

C - (4,4)

D - (4,2)

Test 2

A - (1,1)

B - (1,10)

C - (10,10)

D - (10,1)

**4. Результаты выполнения тестов.**

Test 1

Площадь - 4

Периметр - 8

Перемещение на 1 по x и 1 по y:

Координаты нового прямоугольника:

A:(3, 3)

B:(3, 5)

C:(5, 5)

D:(5, 3)

Увеличение в 2 раза:

Координаты нового прямоугольника:

A - (4,4)

B - (4,8)

C - (8,8)

D - (8,4)

Сравнение:

Второй прямоугольник:

A - (0, 0)

B - (0, 10)

C - (10, 10)

D - (10, 0)

Площадь второго прямоугольника больше площади первого

Периметр второго прямоугольника больше периметра первого

Пересечение:

Координаты второго прямоугольника:

A - (3,3)

B - (3,6)

C - (6,6)

D - (6,3)

Прямоугольник, представляющий общую часть двух прямоугольников, имеет координаты:

A - (4,3)

B - (3,3)

C - (3,4)

D - (4,4)

Минимальный, содержащий 2 прямоугольника:

Координаты второго прямоугольника:

A - (4,4)

B - (4,5)

C - (5,5)

D - (5,4)

Координаты наименьшего прямоугольника, содержащего два введенных прямоугольника:

A - (2,2)

B - (2,5)

C - (5,5)

D - (5,2)

Test 2:

Площадь - 81

Периметр - 36

Перемещение на 2 по x и 2 по y:

A:(3, 3)

B:(3, 12)

C:(12, 12)

D:(12, 3)

Увеличение в 3 раза:

A - (3,3)

B - (3,30)

C - (30,30)

D - (30,3)

Сравнение:

Второй прямоугольник

A - (0,0)

B - (0,15)

C - (15,15)

D - (15,0)

Площадь второго прямоугольника больше площади первого

Периметр второго прямоугольника больше периметра первого

Пересечение:

Координаты первого прямоугольника:

A - (0,0)

B - (0,15)

C - (15,15)

D - (15,0)

Координаты второго прямоугольника:

A - (1,1)

B - (1,3)

C - (3,3)

D - (3,1)

Прямоугольник, представляющий общую часть двух прямоугольников, имеет координаты:

A - (1,3)

B - (3,3)

C - (3,1)

D - (1,1)

Минимальный, содержащий 2 прямоугольника:

Координаты первого прямоугольника:

A - (1,1)

B - (1,10)

C - (10,10)

D - (10,1)

Координаты второго прямоугольника:

A - (1,1)

B - (1,12)

C - (12,12)

D - (12,1)

Координаты наименьшего прямоугольника, содержащего два введенных прямоугольника:

A - (1,1)

B - (1,12)

C - (12,12)

D - (12,1)

**Вывод:**

Выполняя данную лабораторную работу, я познакомился с основными понятиями, реализовал свой первый класс, написал код в ООП стиле. Я поработал с модификаторами доступа private и public, а так же реализовал меню через switch. Помимо этого я научился работать с утилитой CMake, что пригодится мне в дальнейших лабораторных.