**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

**Лабораторная работа № 1**

Тема: Простые классы на языке С++

Студент: Купцов Илья

Группа: 80-207

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2019

1. Постановка задачи

Разработать программу на языке C++ согласно варианту задания. Программа на C++ должна собираться с помощью системы сборки CMake. Программа должна получать данные из стандартного ввода и выводить данные в стандартный вывод.

Разработать класс Rectangle, представляющий собой прямоугольник со сторонами, параллельными осям координат. Поля – координаты левого нижнего и правого верхнего угла. Требуется реализовать следующие методы: вычисление площади и периметра, перемещения вдоль осей, изменение размеров, сравнение по площади и по периметру. Реализовать метод получения прямоугольника, представляющего общую часть (пересечение) двух прямоугольников. Реализовать метод объединения двух прямоугольников: наименьший прямоугольник, включающего оба заданных прямоугольника.

1. Описание программы

В данной лабораторной работе я реализовал класс прямоугольника, использовав язык С++. Для этого я изучил словесные материалы и использовал эти знания для исполнения этой задачи на языке программирования. Далее я тщательно изучил строение прямоугольника, чтобы максимально правильно реализовать его на ЭВМ.

Создав класс прямоугольника, я ввел 5 переменных: две точки с двумя значениями в каждой: (x1, y1), (x2, y2), для воссоздания прямоугольника на координатной плоскости, а также переменную real, которая говорит, существует ли прямоугольник или нет. Далее я создал несколько методов для класса точно так, как требовалось в лабораторной работе. Их работа - простой математический счет. Затем в «main» я написал последовательность действий, которые демонстрируют работу моих методов.

1. Набор testcases

В стандартный ввод подается последовательность из 8 чисел, 4 по 4 из которых координаты двух прямоугольников соответственно. 4 координаты представляют из себя две точки (x1, y1) и (x2, y2), которые выстраивают прямоугольник.

1-ый тест:

-2 4 0 5, 1 8 -4 1;

2-ой тест:

1 2 0 1, -3 -1 -3 0;

3-ий тест

5 6 -9 1, 3 8 0 9;

1. Результаты выполнения тестов.

1-ый тест:

Rectangle info:

exist - 1

x1 = -2 x2 = 4

y1 = 0 y2 = 5

The square = 30

The perimeter = 22

--------------------------------------

Rectangle info:

exist - 1

x1 = 1 x2 = 8

y1 = -4 y2 = 1

The square = 35

The perimeter = 24

--------------------------------------

Sa > Sb? :

0

Pa > Pb? :

0

a && b:

Rectangle info:

exist - 1

x1 = 1 x2 = 4

y1 = 0 y2 = 1

The square = 3

The perimeter = 8

--------------------------------------

a || b:

Rectangle info:

exist - 1

x1 = -2 x2 = 8

y1 = -4 y2 = 5

The square = 90

The perimeter = 38

--------------------------------------

Rectangle c (a || b):

the rectangle change x plays on 3

Rectangle info:

exist - 1

x1 = 1 x2 = 11

y1 = -4 y2 = 5

The square = 90

The perimeter = 38

--------------------------------------

the rectangle change (right) x size on 2

Rectangle info:

exist - 1

x1 = 1 x2 = 13

y1 = -4 y2 = 5

The square = 108

The perimeter = 42

--------------------------------------

the rectangle change (down) y size on 3

Rectangle info:

exist - 1

x1 = 1 x2 = 13

y1 = -7 y2 = 5

The square = 144

The perimeter = 48

--------------------------------------

2-ой тест:

Rectangle info:

exist - 1

x1 = 1 x2 = 2

y1 = 0 y2 = 1

The square = 1

The perimeter = 4

--------------------------------------

Rectangle info:

exist - 1

x1 = -3 x2 = -1

y1 = -3 y2 = 0

The square = 6

The perimeter = 10

--------------------------------------

Sa > Sb? :

0

Pa > Pb? :

0

a && b:

Rectangle info:

exist - 0

The rectangle is doesn’t exist.

--------------------------------------

a || b:

Rectangle info:

exist - 1

x1 = -3 x2 = 2

y1 = -3 y2 = 1

The square = 20

The perimeter = 18

--------------------------------------

the rectangle change x plays on 3

Rectangle info:

exist - 1

x1 = 0 x2 = 5

y1 = -3 y2 = 1

The square = 20

The perimeter = 18

--------------------------------------

the rectangle change (right) x size on 2

Rectangle info:

exist - 1

x1 = 0 x2 = 7

y1 = -3 y2 = 1

The square = 28

The perimeter = 22

--------------------------------------

the rectangle change (down) y size on 3

Rectangle info:

exist - 1

x1 = 0 x2 = 7

y1 = -6 y2 = 1

The square = 49

The perimeter = 28

--------------------------------------

3-ий тест:

Rectangle info:

exist - 1

x1 = 5 x2 = 6

y1 = -9 y2 = 1

The square = 10

The perimeter = 22

--------------------------------------

Rectangle info:

exist - 1

x1 = 3 x2 = 8

y1 = 0 y2 = 9

The square = 45

The perimeter = 28

--------------------------------------

Sa > Sb? :

0

Pa > Pb? :

0

a && b:

Rectangle info:

exist - 1

x1 = 5 x2 = 6

y1 = 0 y2 = 1

The square = 1

The perimeter = 4

--------------------------------------

a || b:

Rectangle info:

exist - 1

x1 = 3 x2 = 8

y1 = -9 y2 = 9

The square = 90

The perimeter = 46

--------------------------------------

the rectangle change x plays on 3

Rectangle info:

exist - 1

x1 = 6 x2 = 11

y1 = -9 y2 = 9

The square = 90

The perimeter = 46

--------------------------------------

the rectangle change (right) x size on 2

Rectangle info:

exist - 1

x1 = 6 x2 = 13

y1 = -9 y2 = 9

The square = 126

The perimeter = 50

--------------------------------------

the rectangle change (down) y size on 3

Rectangle info:

exist - 1

x1 = 6 x2 = 13

y1 = -12 y2 = 9

The square = 147

The perimeter = 56

--------------------------------------

1. Листинг программы

#include "pch.h"

#include <iostream>

using namespace std;

void change\_values(int &a, int &b) {

a = a + b;

b = a - b;

a = a - b;

}

class Rectangle {

private:

int x1, y1;

int x2, y2;

bool real;

public:

Rectangle() {

real = 0;

}

int get\_square() {

if (!real) {

doesnt\_exist();

return 0;

}

return abs(x2 - x1) \* abs(y2 - y1);

}

int get\_perimetr() {

if (!real) {

doesnt\_exist();

return 0;

}

return 2 \* (abs(x2 - x1) + abs(y2 - y1));

}

void get\_coordinat() {

cout << "input the first x coordinate\n"; cin >> x1;

cout << "input the second x coordinate\n"; cin >> x2;

if (x2 < x1) {

change\_values(x1, x2);

}

cout << "input the first y coordinate\n"; cin >> y1;

cout << "input the second y coordinate\n"; cin >> y2;

if (y2 < y1) {

change\_values(y1, y2);

}

if (!(x1 == x2 || y1 == y2)) {

real = 1;

}

}

bool move\_on\_x(int dx) {

cout << "the rectangle change x plays on " << dx << "\n";

if (!real) {

doesnt\_exist();

return 0;

}

x1 += dx;

x2 += dx;

return 1;

}

bool move\_on\_y(int dy) {

cout << "the rectangle change y plays on " << dy << "\n";

if (!real) {

doesnt\_exist();

return 0;

}

y1 += dy;

y2 += dy;

return 1;

}

bool change\_size\_x\_right(int dx) {

cout << "the rectangle change (right) x size on " << dx << "\n";

if (!real) {

doesnt\_exist();

return 0;

}

if (dx <= -(x2 - x1)) {

real = 0;

return 1;

}

x2 += dx;

return 1;

}

bool change\_size\_x\_left(int dx) {

cout << "the rectangle change (left) x size on " << dx << "\n";

dx = -dx;

if (!real) {

doesnt\_exist();

return 0;

}

if (dx <= -(x2 - x1)) {

real = 0;

return 1;

}

x1 += dx;

return 1;

}

bool change\_size\_y\_up(int dy) {

cout << "the rectangle change (up) y size on " << dy << "\n";

if (!real) {

doesnt\_exist();

return 0;

}

if (dy <= -(y2 - y1)) {

real = 0;

return 1;

}

y2 += dy;

return 1;

}

bool change\_size\_y\_down(int dy) {

cout << "the rectangle change (down) y size on " << dy << "\n";

dy = -dy;

if (!real) {

doesnt\_exist();

return 0;

}

if (dy <= -(y2 - y1)) {

real = 0;

return 1;

}

y1 += dy;

return 1;

}

void doesnt\_exist() {

cout << "The regtangle is doesnt exist.\n";

}

void check\_info() {

cout << "Rectangle info:\n";

cout << "exist - " << real << "\n";

if (real) {

cout << "x1 = " << x1 << " x2 = " << x2 << "\n";

cout << "y1 = " << y1 << " y2 = " << y2 << "\n";

cout << "The square = " << get\_square() << "\n";

cout << "The perimetr = " << get\_perimetr() << "\n";

}

else {

doesnt\_exist();

}

cout << "--------------------------------------\n";

}

Rectangle operator && (Rectangle b) {

cout << "a && b:\n";

Rectangle c;

if (real == 0 || b.real == 0) {

c.real = 0;

return c;

}

if (b.x2 <= x1 || b.x1 >= x2 || b.y1 >= y2 || b.y2 <= y1) {

c.real = 0;

return c;

}

c.real = 1;

if (b.x1 < x1) {

c.x1 = x1;

}

else {

c.x1 = b.x1;

}

if (x2 < b.x2) {

c.x2 = x2;

}

else {

c.x2 = b.x2;

}

if (b.y1 < y1) {

c.y1 = y1;

}

else {

c.y1 = b.y1;

}

if (y2 < b.y2) {

c.y2 = y2;

}

else {

c.y2 = b.y2;

}

return c;

}

Rectangle operator || (Rectangle b) {

cout << "a || b:\n";

Rectangle r;

if (real + b.real == 0) {

r.real = 0;

return r;

}

if (real == 0) {

return b;

}

if (b.real == 0) {

return \*this;

}

r.real = 1;

int c[4];

c[0] = y1;

c[1] = y2;

c[2] = b.y1;

c[3] = b.y2;

int max\_y = c[0];

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

if (c[i] > max\_y) {

max\_y = c[i];

}

}

int min\_y = c[0];

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

if (c[i] < min\_y) {

min\_y = c[i];

}

}

c[0] = x1;

c[1] = x2;

c[2] = b.x1;

c[3] = b.x2;

int max\_x = c[0];

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

if (c[i] > max\_x) {

max\_x = c[i];

}

}

int min\_x = c[0];

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

if (c[i] < min\_x) {

min\_x = c[i];

}

}

r.x1 = min\_x;

r.x2 = max\_x;

r.y1 = min\_y;

r.y2 = max\_y;

return r;

}

bool square\_bigger (Rectangle b) {

cout << "Sa > Sb? :\n";

if (get\_square() > b.get\_square()) {

return 1;

}

else {

return 0;

}

}

bool perimetr\_bigger (Rectangle b) {

cout << "Pa > Pb? :\n";

if (get\_perimetr() > b.get\_perimetr()) {

return 1;

}

else {

return 0;

}

}

};

int main()

{

Rectangle a;

Rectangle b;

Rectangle c;

cout << "the first rectangle:\n";

a.get\_coordinat();

cout << "the second rectangle:\n";

b.get\_coordinat();

a.check\_info();

b.check\_info();

cout << a.square\_bigger(b) << '\n';

cout << a.perimetr\_bigger(b) << '\n';

c = a && b;

c.check\_info();

c = a || b;

c.check\_info();

c.move\_on\_x(3);

c.check\_info();

c.change\_size\_x\_right(2);

c.check\_info();

c.change\_size\_y\_down(3);

c.check\_info();

return 0;

}