**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «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: Ивенкова Любовь Васильевна

Группа: 80-208

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

1. Постановка задачи

Вариант 12

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

* Создаём класс Rentagle c полями x1, y1, x2, y2;
* Реализуем у него методы вычисления площади и периметра, перемещения вдоль осей, изменение размеров, сравнение по площади и по периметру, объединения и пересечения прямоугольников. Также дополнительно создаём функции изменения координат прямоугольника, вывода координат, вывода меню.
* Программа принимает на вход целые числа, и выводит также целые числа.
* Функция main:

Создание двух указателей на класс Rentagle.

Заполнение первого введенными координатами.

Вывод меню и дальнейший вывод результатов вычислений в зависимости от выбора пользователя

1. Набор тестов

Таблица 1.

|  |  |
| --- | --- |
| 0 0 8 8  1  2  4  9  7  0 0 1 1  8  0 0 1 1 | -координаты углов первого прямоугольника;  -нахождение площади прямоугольника;  -нахождения периметра прямоугольника;  -уменьшение в 2 раза;  -вывод новых координат прямоугольника;  -нахождение пересечения прямоугольников;  -нахождение объединения прямоугольников; |

Таблица 2.

|  |  |
| --- | --- |
| 0 0 1 1  4  9  6  4 4 8 8  8  -1 -1 0 0  10  0 0 -1 -1 | -координаты углов первого прямоугольника;  увеличение в 2 раза;  -вывод новых координат прямоугольника;  -сравнение периметров прямоугольников;  -нахождение объединения прямоугольников;  -ввод новых координат прямоугольника; |

1. Результаты выполнения тестов

Введите координаты углов прямоугольника

0 0 8 8

Меню:

1 - Найти площадь прямоугольника;

2 - Найти периметр прямоугольника;

3 - Переместить прямоугольник вдоль осей;

4 - Изменить размер прямоугольника;

5 - Сравнить с др. прямоугольником по площади

6 - Сравнить с др. прямоугольником по периметру

7 - Найти пересечение с др. прямоугольником

8 - Найти объединение с др. прямоугольником;

9 - Вывести координаты прямоугольника;

10 - Ввести/изменить координаты прямоугольника;

11 - Вывести меню;

0 - Выход

>>>>> 1

> 64

>>>>> 2

> 32

>>>>> 4

> Вы хотите уменьшить (1) или увеличить (2) размер?1

Укажите, во сколько раз вы хотите изменить размер: 2

>>>>> 9

> 0 0 4 4

>>>>> 7

> Введите координаты углов второго прямоугольника: 0 0 1 1

0 0 1 1

>>>>> 8

> Введите координаты углов второго прямоугольника: 0 0 1 1

0 0 4 4

>>>>>

Введите координаты углов прямоугольника

0 0 1 1

Меню:

1 - Найти площадь прямоугольника;

2 - Найти периметр прямоугольника;

3 - Переместить прямоугольник вдоль осей;

4 - Изменить размер прямоугольника;

5 - Сравнить с др. прямоугольником по площади

6 - Сравнить с др. прямоугольником по переметру

7 - Найти пересечение с др. прямоугольником

8 - Найти объединение с др. прямоугольником;

9 - Вывести координаты прямоугольника;

10 - Ввести/изменить координаты прямоугольника;

11 - Вывести меню;

0 - Выход

>>>>> 4

> Вы хотите уменьшить (1) или увеличить (2) размер?2

Укажите, во сколько раз вы зотите изменить размер: 2

>>>>> 9

> 0 0 2 2

>>>>> 6

> Введите координаты углов второго прямоугольника: 4 4 8 8

Периметр второго больше периметра первого

>>>>> 8

> Введите координаты углов второго прямоугольника: -1 -1 0 0

-1 -1 2 2

>>>>> 10

> Введите координаты прямоугольника: 0 0 -1 -1

Неверный формат входных данных!

>>>>>

1. **Листинг программы**

/\* Ивенкова Любовь Васильевна, М8О-208Б-19

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.\*/

#include <iostream>

#include <algorithm>

using namespace std;

class Rectangle {

public:

void set\_value(int, int, int, int);

int check\_(int, int, int, int);

int get\_value\_x1();

int get\_value\_y1();

int get\_value\_x2();

int get\_value\_y2();

int area();

int perimetr();

int movie(int, int);

int size();

int comparision\_area(int);

int comparision\_perimetr(int);

int intersection(int, int, int, int);

int unification(int, int, int, int);

private:

int x1, y1, x2, y2;

};

int Rectangle::check\_(int x1, int y1, int x2, int y2) {

if (x1 == x2 || y1 == y2) return '1';

else if (x1 > x2 || y1 > y2) return '2';

return 0;

}

void Rectangle::set\_value(int x1, int y1, int x2, int y2) {

Rectangle::x1 = x1;

Rectangle::y1 = y1;

Rectangle::x2 = x2;

Rectangle::y2 = y2;

}

int Rectangle::get\_value\_x1() {

int a = Rectangle::x1;

return a;

}

int Rectangle::get\_value\_y1() {

int a = Rectangle::y1;

return a;

}

int Rectangle::get\_value\_x2() {

int a = Rectangle::x2;

return a;

}

int Rectangle::get\_value\_y2() {

int a = Rectangle::y2;

return a;

}

int Rectangle::area() {

int s = (Rectangle::x2 - Rectangle::x1) \* (Rectangle::y2 - Rectangle::y1);

return s;

}

int Rectangle::perimetr() {

int p = 2 \* (Rectangle::x2 - Rectangle::x1) + \

+ 2 \* (Rectangle::y2 - Rectangle::y1);

return p;

}

int Rectangle::movie(int x, int y) {

Rectangle::set\_value(Rectangle::x1 + x, Rectangle::y1 + y, \

Rectangle::x2 + x, Rectangle::y2 + y);

return 0;

}

int Rectangle::size() {

cout << "Вы хотите уменьшить (1) или увеличить (2) размер?";

char s;

cin >> s;

std::cout << "Укажите, во сколько раз вы зотите изменить размер: ";

int d;

std::cin >> d;

if (s == '1') {

Rectangle::set\_value(Rectangle::x1, Rectangle::y1, \

(Rectangle::x2) / d, Rectangle::y2 / d);

}

if (s == '2') {

Rectangle::set\_value(Rectangle::x1, Rectangle::y1, \

(Rectangle::x2) \* d, Rectangle::y2 \* d);

}

return 0;

}

int Rectangle::comparision\_area(int n) {

int m = Rectangle::area();

if (m > n) std::cout << "Площадь первого больше площади второго";

if (m < n) std::cout << "Площадь второго больше площади первого";

if (m == n) std::cout << "Площади равны";

return 0;

}

int Rectangle::comparision\_perimetr(int n) {

int m = Rectangle::perimetr();

if (m > n) std::cout << "Периметр первого больше периметра второго";;

if (m < n) std::cout << "Периметр второго больше периметра первого";

if (m == n) std::cout << "Периметры равны";

return 0;

}

int Rectangle::intersection(int x3, int y3, int x4, int y4) {

if (Rectangle::y1 >= y4 || y3 >= Rectangle::y2 || x4 <= Rectangle::x1 || Rectangle::x2 <= x3)

std::cout << "Прямоугольники не пересекаются";

else {

int d1 = std::max(Rectangle::x1, x3);

int h1 = std::max(Rectangle::y1, y3);

int d2 = std::min(Rectangle::x2, x4);

int h2 = std::min(Rectangle::y2, y4);

std::cout << d1 << " " << h1 << " " << d2 << " " << h2 << "\n";

}

return 0;

}

int Rectangle::unification(int x3, int y3, int x4, int y4) {

int d1 = std::min(Rectangle::x1, x3);

int h1 = std::min(Rectangle::y1, y3);

int d2 = std::max(Rectangle::x2, x4);

int h2 = std::max(Rectangle::y2, y4);

std::cout << d1 << " " << h1 << " " << d2 << " " << h2 << "\n";

return 0;

}

void menu() {

std::cout << "Меню:" << std::endl;

std::cout << "1 - Найти площадь прямоугольника;" << std::endl;

std::cout << "2 - Найти периметр прямоугольника;" << std::endl;

std::cout << "3 - Переместить прямоугольник вдоль осей;" << std::endl;

std::cout << "4 - Изменить размер прямоугольника;" << std::endl;

std::cout << "5 - Сравнить с др. прямоугольником по площади" << std::endl;

std::cout << "6 - Сравнить с др. прямоугольником по переметру" << std::endl;

std::cout << "7 - Найти пересечение с др. прямоугольником" << std::endl;

std::cout << "8 - Найти объединение с др. прямоугольником;" << std::endl;

std::cout << "9 - Вывести координаты прямоугольника;" << std::endl;

std::cout << "10 - Ввести/изменить координаты прямоугольника;" << std::endl;

std::cout << "11 - Вывести меню;" << std::endl;

std::cout << "0 - Выход" << std::endl;

}

int main() {

setlocale(LC\_ALL, "RUS");

int x1, y1, x2, y2, x3 = 0, y3 = 0, x4 = 0, y4 = 0;

Rectangle\* Rectangle1 = new Rectangle;

Rectangle\* Rectangle2 = new Rectangle;

std::cout << "Введите координаты углов прямоугольника" << std::endl;

std::cin >> x1 >> y1 >> x2 >> y2;

Rectangle1->check\_(x1, y1, x2, y2);

if (Rectangle1->check\_(x1, y1, x2, y2) == '1') std::cout << "Это не прямоугольник!";

else if (Rectangle1->check\_(x1, y1, x2, y2) == '2') std::cout << "Неверный формат входных данных!";

else Rectangle1->set\_value(x1, y1, x2, y2);

int a;

menu();

std::cout << ">>>>> ";

std::cin >> a;

while (a != 0) {

std::cout << "> ";

if (a == 1)

std::cout << Rectangle1->area();

if (a == 2)

std::cout << Rectangle1->perimetr();

if (a == 3) {

std::cout << "Укажите, на сколько вы хотите переместить прям. по осям x, y: ";

int x, y;

std::cin >> x >> y;

Rectangle1->movie(x, y);

}

if (a == 4) {

Rectangle1->size();

}

if (a == 5) {

std::cout << "Введите координаты углов второго прямоугольника: ";

std::cin >> x3 >> y3 >> x4 >> y4;

if (Rectangle2->check\_(x3, y3, x4, y4) == '1') std::cout << "Это не прямоугольник!";

else if (Rectangle2->check\_(x3, y3, x4, y4) == '2') std::cout << "Неверный формат входных данных!";

else {

Rectangle2->set\_value(x3, y3, x4, y4);

int n = Rectangle2->area();

Rectangle1->comparision\_area(n);

}

}

if (a == 6) {

std::cout << "Введите координаты углов второго прямоугольника: ";

std::cin >> x3 >> y3 >> x4 >> y4;

if (Rectangle2->check\_(x3, y3, x4, y4) == '1') std::cout << "Это не прямоугольник!";

else if (Rectangle2->check\_(x3, y3, x4, y4) == '2') std::cout << "Неверный формат входных данных!";

else {

Rectangle2->set\_value(x3, y3, x4, y4);

int n = Rectangle2->perimetr();

Rectangle1->comparision\_perimetr(n);

}

}

if (a == 7) {

std::cout << "Введите координаты углов второго прямоугольника: ";

std::cin >> x3 >> y3 >> x4 >> y4;

if (Rectangle2->check\_(x3, y3, x4, y4) == '1') std::cout << "Это не прямоугольник!";

else if (Rectangle2->check\_(x3, y3, x4, y4) == '2') std::cout << "Неверный формат входных данных!";

else Rectangle1->intersection(x3, y3, x4, y4);

}

if (a == 8) {

std::cout << "Введите координаты углов второго прямоугольника: ";

std::cin >> x3 >> y3 >> x4 >> y4;

if (Rectangle2->check\_(x3, y3, x4, y4) == '1') std::cout << "Это не прямоугольник!";

else if (Rectangle2->check\_(x3, y3, x4, y4) == '2') std::cout << "Неверный формат входных данных!";

else Rectangle1->unification(x3, y3, x4, y4);

}

if (a == 9) {

cout << Rectangle1->get\_value\_x1() << " " << Rectangle1->get\_value\_y1() << " " << \

Rectangle1->get\_value\_x2() << " " << Rectangle1->get\_value\_y2();

}

if (a == 10) {

std::cout << "Введите координаты прямоугольника: ";

std::cin >> x1 >> y1 >> x2 >> y2;

if (Rectangle1->check\_(x1, y1, x2, y2) == '1') std::cout << "Это не прямоугольник!";

else if (Rectangle1->check\_(x1, y1, x2, y2) == '2') std::cout << "Неверный формат входных данных!";

else Rectangle1->set\_value(x1, y1, x2, y2);

}

if (a == 11) {

menu();

}

std::cout << " " << std::endl;

std::cout << ">>>>> ";

std::cin >> a;

}

std::cout << "Программа завершена!" << std::endl;

return 0;

}

1. Вывод

Я научилась работать с простыми классами в C++, производить оперцации над ними и выводить данные.