

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 прямоугольника
};

#ifdef 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 прямоугольников \n 8 - Пересечение двух прямоугольников \n 9 -
    Минимальный прямоугольник, включающий 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_STANDARD 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.