МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

|  |
| --- |
| КАФЕДРА компьютерных технологий и программной инженерии |

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| старший преподаватель |  |  |  | Шумова Е.О. |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| «Перегрузка операторов» |
| по дисциплине: Объектно-ориентированное программирование |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | Z9431 |  |  |  | Андреев Д.И. |
|  | номер группы |  | подпись, дата |  | инициалы, фамилия |
| Студенческий билет № | 2019/3781 | |  |  |  |

Санкт-Петербург 2020

Оглавление

[1. Условие 3](#_Toc70886478)

[2. Листинг программы 3](#_Toc70886479)

[3. Результаты работы программы. 5](#_Toc70886480)

[4. Вывод 7](#_Toc70886481)

1. Условие

Вариант 1. Разработать класс «Прямоугольник». Определить в нем конструкторы и деструктор, перегрузить операцию пересечения прямоугольников (операция “\*”), операцию вычисления площади прямоугольника, операции сравнения (по площади).

1. Листинг программы

Файл rectangle.h

class Rectangle

{

public:

Rectangle();

Rectangle(int x, int y, int width, int height);

~Rectangle();

int width() const;

int height() const;

int x() const;

int y() const;

void setWidth(int w);

void setHeight(int h);

void setX(int x);

void setY(int y);

Rectangle operator\*(const Rectangle& other) const;

bool operator==(const Rectangle& other) const;

bool operator!=(const Rectangle& other) const;

bool operator<(const Rectangle& other) const;

bool operator>(const Rectangle& other) const;

int operator()() const;

private:

//Левый верхний угол

int m\_x;

int m\_y;

//Размеры

int m\_width;

int m\_height;

};

Файл rectangle.cpp

#include "Rectangle.h"

#include <algorithm>

#include <iostream>

Rectangle Rectangle::operator\*(const Rectangle& other) const

{

int leftX = std::max(m\_x, other.m\_x);

int topY = std::max(m\_y, other.m\_y);

int bottomY = std::max(m\_y - m\_height, other.m\_y - other.m\_height);

int rightX = std::min(m\_x + m\_width, other.m\_x + other.m\_width);

if (rightX < leftX || bottomY > topY)

return Rectangle();

int intersectionWidth = rightX - leftX;

int intersectionHeight = topY - bottomY;

return Rectangle(leftX, topY, intersectionWidth, intersectionHeight);

}

bool Rectangle::operator==(const Rectangle& other) const

{

return this->operator()() == other();

}

bool Rectangle::operator!=(const Rectangle& other) const

{

return !this->operator==(other);

}

bool Rectangle::operator<(const Rectangle& other) const

{

return !(this->operator>(other)) && (this->operator!=(other));

}

bool Rectangle::operator>(const Rectangle& other) const

{

return this->operator()() > other();

}

int Rectangle::operator()() const

{

return m\_height \* m\_width;

}

Rectangle::Rectangle()

: m\_x(0)

, m\_y(0)

, m\_width(0)

, m\_height(0)

{

}

Rectangle::Rectangle(int x, int y, int width, int height)

: m\_x(x)

, m\_y(y)

, m\_width(0)

, m\_height(0)

{

setHeight(height);

setWidth(width);

}

Rectangle::~Rectangle()

{

}

int Rectangle::width() const

{

return m\_width;

}

int Rectangle::height() const

{

return m\_height;

}

int Rectangle::x() const

{

return m\_x;

}

int Rectangle::y() const

{

return m\_y;

}

void Rectangle::setWidth(int w)

{

if (w < 0) {

std::cerr << "width cannot be less than nothing\n";

return;

}

m\_width = w;

}

void Rectangle::setHeight(int h)

{

if (h < 0) {

std::cerr << "height cannot be less than nothing\n";

return;

}

m\_height = h;

}

void Rectangle::setX(int x)

{

m\_x = x;

}

void Rectangle::setY(int y)

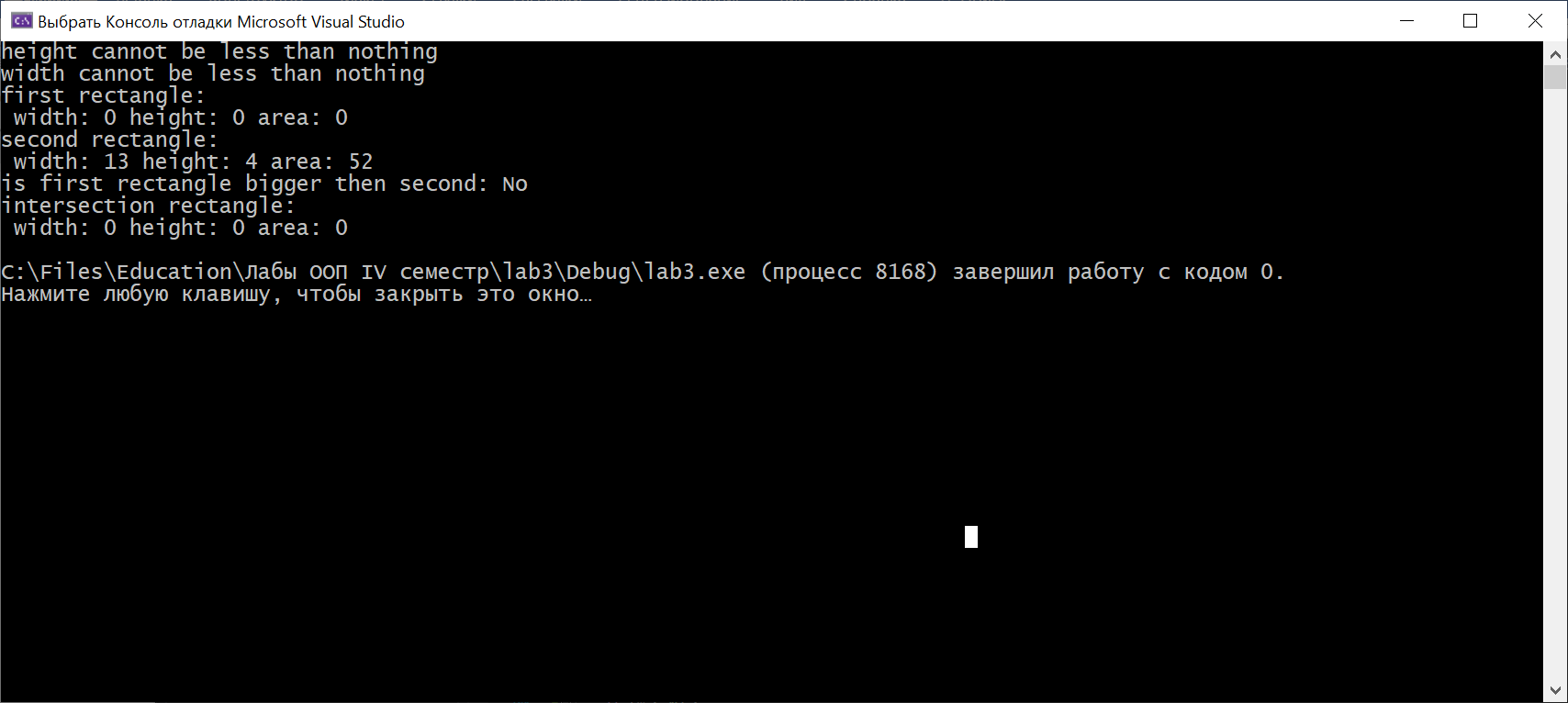
{

m\_y = y;

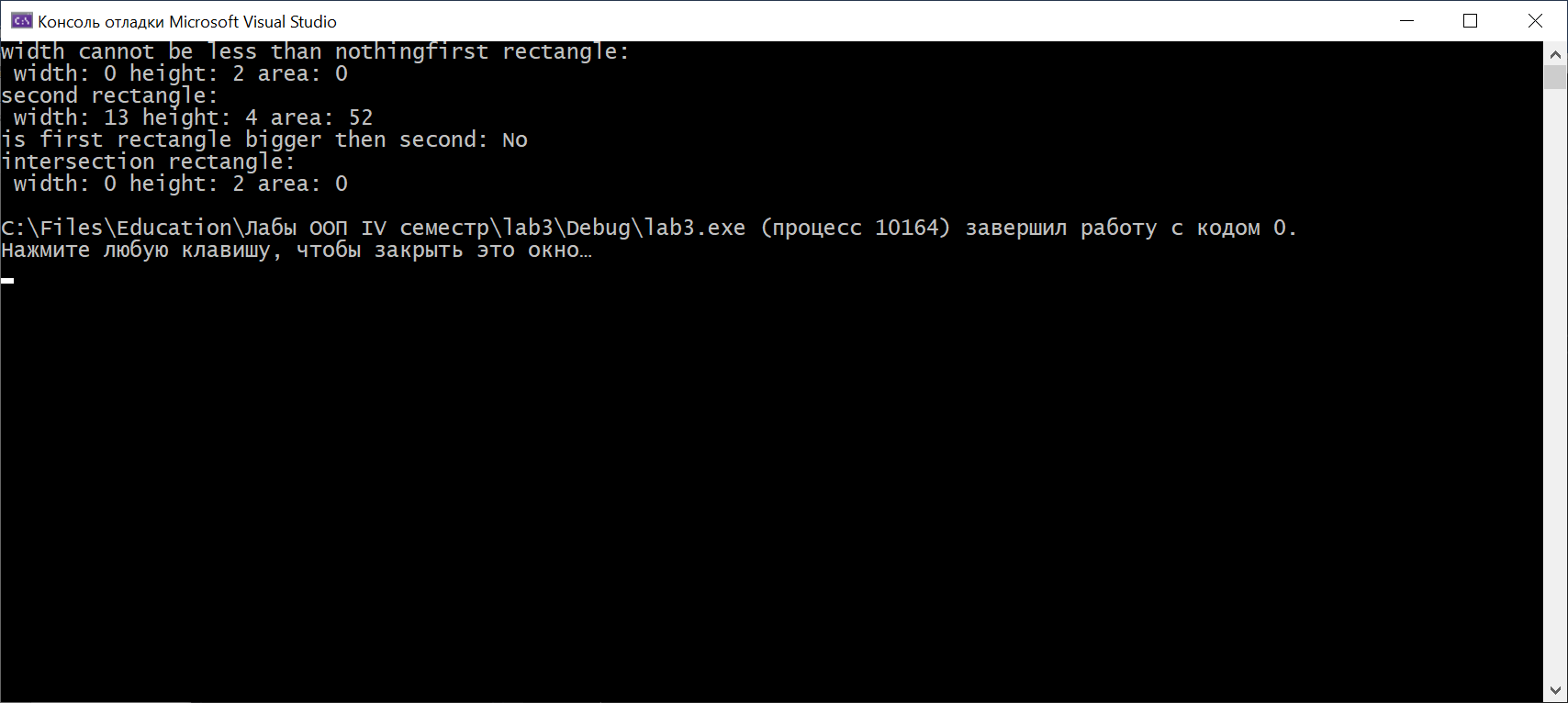
}

1. Результаты работы программы.

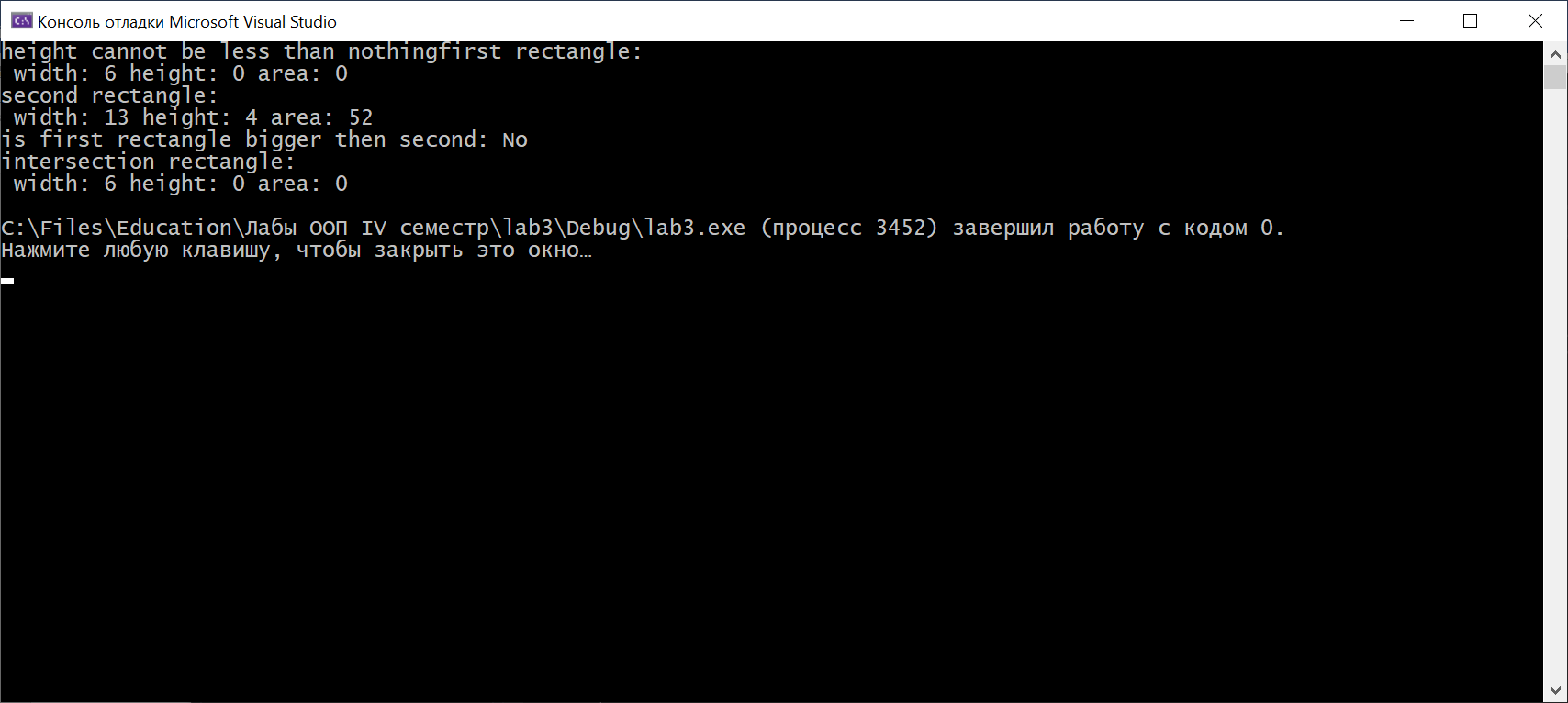
Результаты работы при width <= 0 и height <= 0:



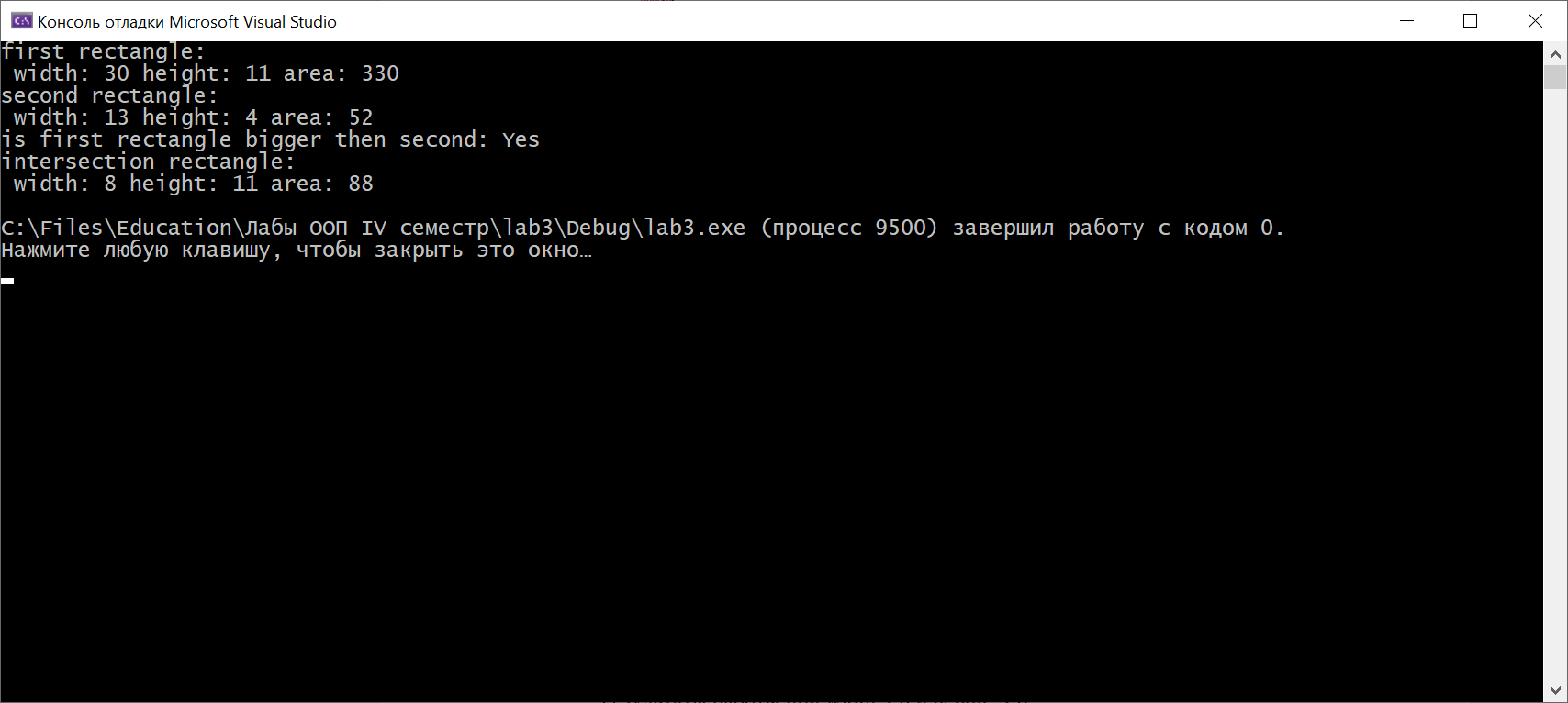
Результаты работы при width <= 0 и height > 0:



Результаты работы при width > 0 и height <= 0:



Результаты работы при width > 0 и height > 0:



1. Вывод

В ходе лабораторной работы был разработан класс, описывающий прямоугольник. Данный класс обладает возможностью нахождения собственной площади, площади пересечения с другим прямоугольником, а также сравнивать прямоугольники за счет перегрузки операторов. При этом реализация класса скрыта от пользователя.