ГУАП

КАФЕДРА № 43

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

ассистент Шумова Е.О.

должность, уч. степень, звание подпись, дата инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

«ПЕРЕГРУЗКА ОПЕРАТОРОВ»

по курсу: ОБЪЕКТНО ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

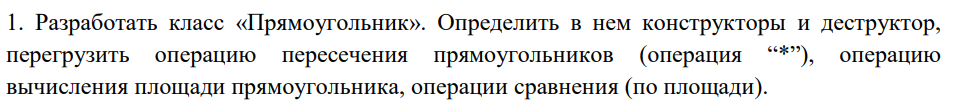
СТУДЕНТ ГР. № 4133к Иванов И.В.

подпись, дата инициалы, фамилия

Санкт-Петербург 2022

**Цель работы:** Изучить механизм перегрузки операторов для типов, определенных пользователем посредством использования методов класса и дружественных функций.

**Условие для варианта:** 1



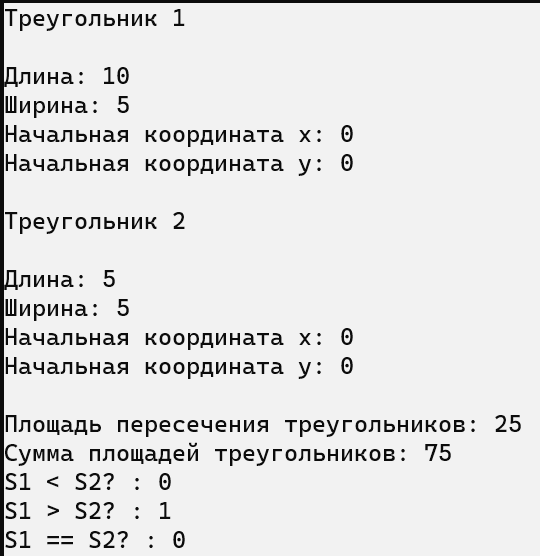
**Листинг программы:**

1. #include <iostream>
3. using namespace std;
5. class Rectangle {
7. private:
9. double start\_x\_coord;
10. double start\_y\_coord;
11. double width;
12. double lenght;
13. double area;
15. public:
17. Rectangle(double l, double w, double sxc, double syc);
19. double getArea() {
20. return area;
21. }
22. double getWidth() {
23. return width;
24. }
25. double getLenght() {
26. return lenght;
27. }
28. double getX() {
29. return start\_x\_coord;
30. }
31. double getY() {
32. return start\_y\_coord;
33. }
35. friend double operator\*(const Rectangle &r1, const Rectangle& r2);
36. bool operator<(const Rectangle& r2);
37. friend bool operator>(const Rectangle& r1, const Rectangle& r2);
38. friend bool operator<=(const Rectangle& r1, const Rectangle& r2);
39. friend bool operator>=(const Rectangle& r1, const Rectangle& r2);
40. friend bool operator==(const Rectangle& r1, const Rectangle& r2);
42. double operator+(const Rectangle &value);
44. ~Rectangle() {};
45. };
47. Rectangle::Rectangle(double l, double w, double sxc, double syc) {
48. if (l <= 0 || w <= 0) {
49. cout << "Прямоугольник с заданными сторонами не может быть создан." << endl;
50. return;
51. }
53. lenght = l;
54. width = w;
55. area = l \* w;
57. start\_x\_coord = sxc;
58. start\_y\_coord = syc;
59. }
61. // Overloading operator by method
62. double Rectangle::operator+(const Rectangle& value) {
63. return area + value.area;
64. }
66. // Overloading operators by friendly functions
67. double operator\*(const Rectangle& r1, const Rectangle& r2) {
69. double left, right, top, bottom;
70. double out\_w, out\_l;
72. left = max(r1.start\_x\_coord, r2.start\_x\_coord);
73. right = min((r1.start\_x\_coord + r1.lenght), (r2.start\_x\_coord + r2.lenght));
74. bottom = max(r1.start\_y\_coord, r2.start\_y\_coord);
75. top = min((r1.start\_y\_coord + r1.width), (r2.start\_y\_coord + r2.width));
77. out\_w = top - bottom;
78. out\_l = right - left;
80. if (out\_w < 0 || out\_l < 0) {
81. return 0;
82. }
84. return out\_w \* out\_l;
85. }
87. bool Rectangle::operator<(const Rectangle& r2) {
88. return area < r2.area;
89. }
90. bool operator>(const Rectangle& r1, const Rectangle& r2) {
91. return r1.area > r2.area;
92. }
93. bool operator<=(const Rectangle& r1, const Rectangle& r2) {
94. return r1.area <= r2.area;
95. }
96. bool operator>=(const Rectangle& r1, const Rectangle& r2) {
97. return r1.area >= r2.area;
98. }
99. bool operator==(const Rectangle& r1, const Rectangle& r2) {
100. return r1.area == r2.area;
101. }


105. int main() {
106. system("color F0");
108. setlocale(LC\_ALL, "Rus");
110. Rectangle r1(10, 5, 0, 0), r2(5, 5, 0, 0);
112. cout << "Треугольник 1" << endl << endl;
113. cout << "Длина: " << r1.getLenght() << endl;
114. cout << "Ширина: " << r1.getWidth() << endl;
115. cout << "Начальная координата x: " << r1.getX() << endl;
116. cout << "Начальная координата y: " << r1.getY() << endl << endl;
118. cout << "Треугольник 2" << endl << endl;
119. cout << "Длина: " << r2.getLenght() << endl;
120. cout << "Ширина: " << r2.getWidth() << endl;
121. cout << "Начальная координата x: " << r2.getX() << endl;
122. cout << "Начальная координата y: " << r2.getY() << endl << endl;

125. cout << "Площадь пересечения треугольников: " << r1 \* r2 << endl;
126. cout << "Сумма площадей треугольников: " << r1 + r2 << endl;
127. cout << "S1 < S2? : " << (r1 < r2) << endl;
128. cout << "S1 > S2? : " << (r1 > r2) << endl;
129. cout << "S1 == S2? : " << (r1 == r2) << endl;
130. }

**Результаты работы программы:**



**Вывод:**

В ходе лабораторной работы я изучил механизм перегрузки операторов для типов, определенных пользователем посредством использования методов класса и дружественных функций.