

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Перегрузка операторов.

по курсу: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4136

подпись, дата

Бобрович Н. С.

инициалы, фамилия

Санкт-Петербург 2022

Цель работы:

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

Закрепить знания по теме:

Перегрузка операторов.

Условие:

В работе необходимо реализовать класс в соответствии с вариантом задания. В классе должны быть предусмотрены:

- конструктор для установки начальных значений полей,
- перегруженный оператор, являющийся членом класса,
- перегруженный дружественный оператор.

Вариант 2.

Разработать класс «Прямоугольник», в котором содержатся поля для хранения 4 вершин прямоугольника. Определить в нем конструкторы и деструктор, перегрузить операцию объединения прямоугольников (операция `"*"`) для случая успешного выполнения перегруженной операции проверки совпадения сторон с равной длиной), операцию вычисления периметра прямоугольника, операции сравнения двух прямоугольников (по периметру).

Листинг программы:

```

1  #include <iostream>
2  #include <math.h>
3
4  using namespace std;
5
6  class praymoyg
7  {
8  private:
9      int a;
10 public:
11     int x1, x2, x3, x4, y1, y2, y3, y4, n, P;
12     praymoyg operator*(praymoyg t);
13     praymoyg operator|(praymoyg t); //первый периметр
14     praymoyg operator==(praymoyg t); //второй периметр
15     praymoyg operator/(praymoyg t); //сравнение периметров
16     praymoyg operator+(praymoyg t);
17     friend int operator-(praymoyg t, int a);
18     void show();
19     void showP();
20     void assign();
21     praymoyg () {}
22     praymoyg (int x1, int x2, int x3, int x4, int y1, int y2, int y3,
23         this->x1 = x1;
24         this->x2 = x2;
25         this->x3 = x3;
26         this->x4 = x4;
27         this->y1 = y1;
28         this->y2 = y2;
29         this->y3 = y3;
30         this->y4 = y4;
31         this->n = n;
32         this->P = P;
33     }
34     ~praymoyg() {}
35 };
36
37 praymoyg praymoyg:: operator*(praymoyg t)
38 {
39     praymoyg yvel;
40     yvel.x1 = x1 * n;
41     yvel.x2 = x2 * n;
42     yvel.x3 = x3 * n;
43     yvel.x4 = x4 * n;
44     yvel.y1 = y1 * n;
45     yvel.y2 = y2 * n;
46     yvel.y3 = y3 * n;
47     yvel.y4 = y4 * n;
48     return yvel;

```

```

49     }
50     praimoyg praimoyg:: operator+(praimoyg t)
51     {
52         praimoyg yvel;
53         yvel.x1 = x1 + t.x1;
54         yvel.x2 = x2 + t.x2;
55         yvel.x3 = x3 + t.x3;
56         yvel.x4 = x4 + t.x4;
57         yvel.y1 = y1 + t.y1;
58         yvel.y2 = y2 + t.y2;
59         yvel.y3 = y3 + t.y3;
60         yvel.y4 = y4 + t.y4;
61         return yvel;
62     }
63     praimoyg praimoyg:: operator|(praimoyg t)
64     {
65         praimoyg vivel;
66         vivel.P = (abs(x4) - abs(x1) + abs(y2) - abs(y1)) * 2;
67         return vivel;
68     }
69     praimoyg praimoyg::operator==(praimoyg t)
70     {
71         praimoyg yvel;
72         yvel.P = (abs(t.x4) - abs(t.x1) + abs(t.y2) - abs(t.y1)) * 2;
73         return yvel;
74     }
75     praimoyg praimoyg::operator/(praimoyg t)//если равны, то выводит первое
76     {
77         praimoyg yvel, vivel;
78         vivel.P = (abs(x4) - abs(x1) + abs(y2) - abs(y1)) * 2;
79         yvel.P = (abs(t.x4) - abs(t.x1) + abs(t.y2) - abs(t.y1)) * 2;
80         if (vivel.P == yvel.P) {
81             return yvel;
82         }
83         else {
84             return vivel;
85         }
86     }
87     void praimoyg::show()
88     {
89         cout << "x1= " << x1 << endl << "y1= " << y1 << endl << "x2= " <<
90     }
91     void praimoyg::showP()
92     {
93         cout << "P = " << P << endl ;
94     }
95     void praimoyg::assign()
96     {

```

```

97         cout << "Введите координаты: " << endl;
98         cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3 >> x4 >> y4;
99         cout << "Введите коэффициент n: " << endl;
100        cin >> n;
101        cout << endl;
102    }
103    int operator-(praymoyg t, int a)
104    {
105        praymoyg yvel;
106        yvel.P += a;
107        return a;
108    }
109
110    int main()
111    {
112        system("color F0");
113        setlocale(0, "Rus");
114        praymoyg a, b, c;
115        int k = 1;
116        int P;
117        a.assign();
118        b.assign();
119        c = a;
120        c = a * b;
121        c.show();
122        c = b * a;
123        c.show();
124        c = a + b;
125        c.show();
126        c = a | b;
127        c.showP();
128        c = a == b;
129        c.showP();
130        c = a / b;
131        c.showP();
132        P = a - k;
133        cout << P << endl;
134        return 0;
135    }

```

Результат работы:

Введите координаты:

0

0

2

0

0

2

2

2

Введите коэффициент n:

3

Введите координаты:

2

2

4

2

2

4

4

4

Введите коэффициент n:

2

x1= 0

y1= 0

x2= 6

y2= 0

x3= 0

y3= 6

cs Консоль отладки Microsoft Visual Studio

x1= 0

y1= 0

x2= 6

y2= 0

x3= 0

y3= 6

x4= 6

y4= 6

x1= 4

y1= 4

x2= 8

y2= 4

x3= 4

y3= 8

x4= 8

y4= 8

x1= 2

y1= 2

x2= 6

y2= 2

x3= 2

y3= 6

x4= 6

y4= 6

P = 4

P = 4

P = 4

Вывод:

В результате выполнения работы были получены навыки обращения с перегрузкой операторов.