Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

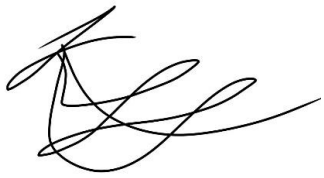
«Национальный исследовательский университет ИТМО»

Факультет информационных технологий и программирования

Лабораторная работа № 3 Вариант 12

Название работы: Перегрузка операторов

Выполнил студент группы № M3113
Крамской Вадим Вадимович
Подпись:



Санкт-Петербург
2023

Условие

№	Тип данных	Операторы
1	Комплексное число	Умножение комплексного числа на вещественное число. Сложение двух комплексных чисел. Умножение двух комплексных чисел. Длина комплексного числа (используйте для этого, например, операцию «приведение к double» – operator double (Complex&).
2	Квадрат на плоскости. Задается координатой левого верхнего угла, стороной квадрата и углом, на который квадрат повернут относительно оси OX.	Равенство площадей квадратов (перегрузите операции ==, !=, <, >) Умножение квадрата на вещественное число (увеличивает сторону квадрата). Прибавление к квадрату вектора (смещение квадрата на указанный вектор).
3	Треугольник на плоскости (самостоятельно выбирайте необходимые данные).	Равенство площадей треугольников (перегрузите операции ==, !=, <, >) – для вычисления площади можете использовать, например, формулу Герона (зависит от тех данных, что используются для хранения треугольника). Прибавление вектора (смещение треугольника на указанный вектор).
4	Матрица 3x3	Перемножение двух матриц. Умножение матрицы на вещественное число. Вычитание и сложение матриц. Сравнение матриц (==, !=, >, <);
5	Стек целых чисел глубиной не более 100.	Добавление числа в стек (operator <<).Изъятие числа из стека (operator >>). Не забудьте написать простую функцию (не оператор) для вывода стека на экран – с ней будет удобнее производить отладку.
6	Массив целых чисел (длина не более 100).	Объединение двух массивов в один (operator+) Сравнение длин массивов (==, >, < !=).
7	Подмножество множества целых чисел от нуля до девяти: { 0, 1, 2, … 9 }.	Объединение двух множеств (operator+).Сравнение (== и !=). Добавление числа в множество (operator+=). Изъятие числа из множества (operator-=).
8	FIFO (очередь) целых чисел длиной не более100.	Добавить целое число в очередь (operator<<) Взять число из очереди (operator>>)

Решение

1) Класс квадрата и точки

```
class Point {
public:
    int data;

    Point(int a) : data(a) {}

    void operator+=(const Point &a) {
        this->data = this->data + a.data;
    }
};

class Square {
public:
    Square(Point x1, float length) : s_x1(x1), s_length(length) {}

public:
    Point s_x1;
    float s_length;
};
```

2) Перегрузка операторов

```cpp
bool operator==(Square &a1, Square &a2) {
    return a1.s_length == a2.s_length;
}

bool operator!=(Square &a1, Square &a2) {
    return a1.s_length != a2.s_length;
}

bool operator>(Square &a1, Square &a2) {
    return a1.s_length > a2.s_length;
}

bool operator<(Square &a1, Square &a2) {
    return a1.s_length < a2.s_length;
}

void operator*(Square &a1, float b) {
    a1.s_length *= b;
}

void operator+(Square &a1, int array[2][1]) {
    a1.s_x1 += array[0][0];
}

std::ostream &operator<<(std::ostream &o, const Square &a) {
    o << a.s_x1.data << ' ' << a.s_length << '\n';
    return o;
}
```

3) Класс очереди

```
class FIFO{
public:
    FIFO() {
        i = 0;
        for (int & j : array){
            j = 0;
        }
    };
    int array[100];
    int i;
};
```

4) Перегрузка операторов

```
void operator<<(FIFO &a1,int a){
    a1.array[a1.i++] = a;
}
void operator>>(FIFO& a1,int &b){
    b = a1.array[0];
    for (int i = 1; i < a1.i; ++i){
        a1.array[i-1] = a1.array[i];
    }
}
```