

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Пермский национальный исследовательский политехнический
университет
Электротехнический факультет
Кафедра информационных технологий и автоматизированных систем

Лабораторная работа №3

" Перегрузка операций "

Вариант: 12

Выполнил студент ИВТ-24-26:
Шишкин Максим Григорьевич

(дата, подпись)

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

(дата, подпись)

Пермь 2025

Содержание

1 Постановка задачи	3
2 Код на C++	4-5
3 Результаты работы	5
4 Блок-схема	6-8
5 Ответы на контрольные вопросы	9
6 Ссылка на github.....	9

1 Постановка задачи

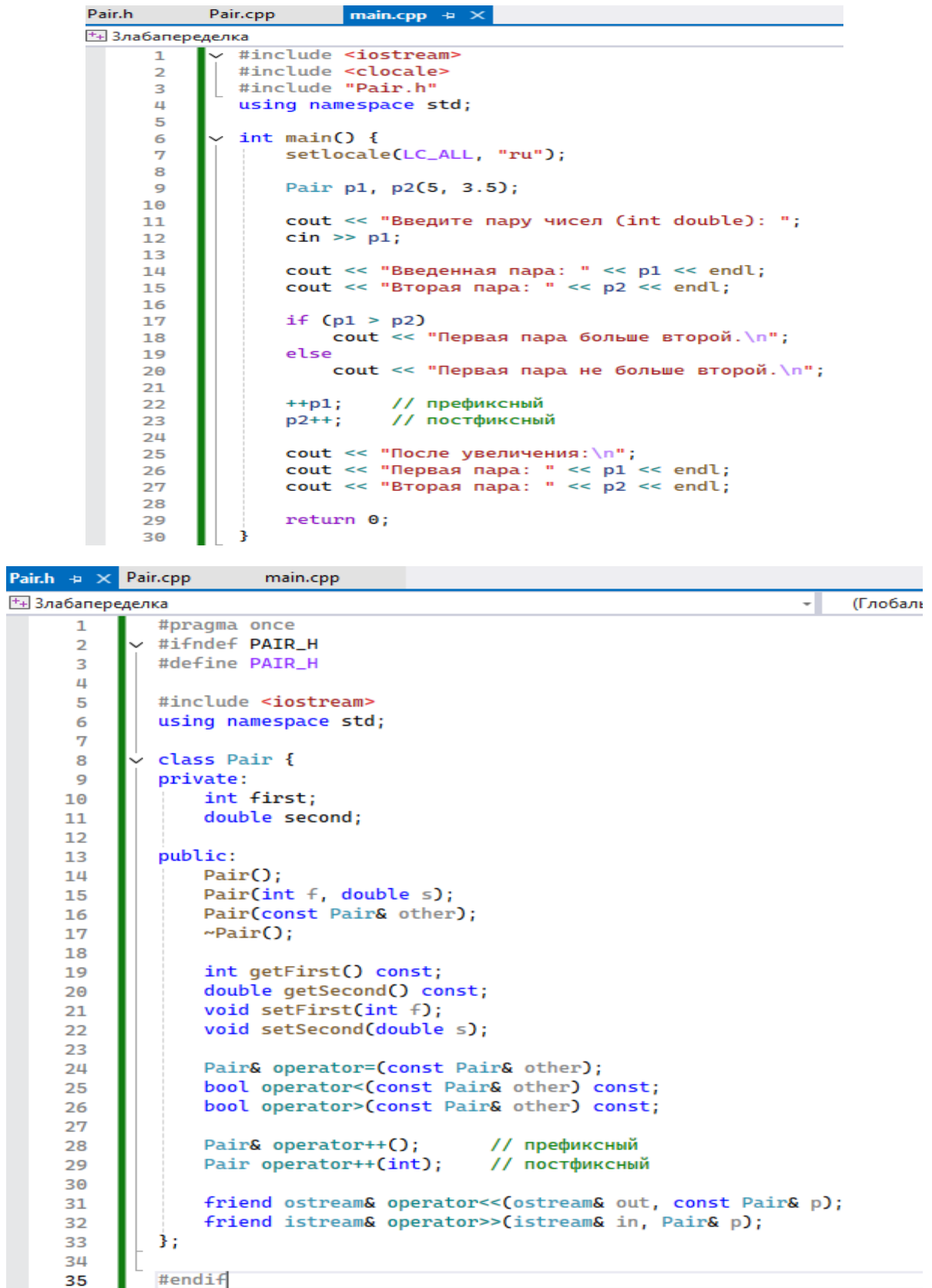
1. Определить пользовательский класс.
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
3. Определить в классе деструктор.
4. Определить в классе компоненты-функции для просмотра и установки полей данных (селекторы и модификаторы).
5. Перегрузить операцию присваивания.
6. Перегрузить операции ввода и вывода объектов с помощью потоков.
7. Перегрузить операции указанные в варианте.
8. Написать программу, в которой продемонстрировать создание объектов и работу всех перегруженных операций.

Вариант: Создать класс Pair (пара чисел). Пара должна быть представлено двумя полями: типа int для первого числа и типа double для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием. Реализовать:

- операции сравнения (<, >).

- операция ++, которая работает следующим образом: если форма операции префиксная, то увеличивается первое число, если форма операции постфиксная, то увеличивается второе число.

2 Код на C++



```
Pair.h Pair.cpp main.cpp
Злабаперделка
1 #include <iostream>
2 #include <locale>
3 #include "Pair.h"
4 using namespace std;
5
6 int main() {
7     setlocale(LC_ALL, "ru");
8
9     Pair p1, p2(5, 3.5);
10
11     cout << "Введите пару чисел (int double): ";
12     cin >> p1;
13
14     cout << "Введенная пара: " << p1 << endl;
15     cout << "Вторая пара: " << p2 << endl;
16
17     if (p1 > p2)
18         cout << "Первая пара больше второй.\n";
19     else
20         cout << "Первая пара не больше второй.\n";
21
22     ++p1; // префиксный
23     p2++; // постфиксный
24
25     cout << "После увеличения:\n";
26     cout << "Первая пара: " << p1 << endl;
27     cout << "Вторая пара: " << p2 << endl;
28
29     return 0;
30 }

Pair.h Pair.cpp main.cpp
Злабаперделка (Глобал
1 #pragma once
2 #ifndef PAIR_H
3 #define PAIR_H
4
5 #include <iostream>
6 using namespace std;
7
8 class Pair {
9 private:
10     int first;
11     double second;
12
13 public:
14     Pair();
15     Pair(int f, double s);
16     Pair(const Pair& other);
17     ~Pair();
18
19     int getFirst() const;
20     double getSecond() const;
21     void setFirst(int f);
22     void setSecond(double s);
23
24     Pair& operator=(const Pair& other);
25     bool operator<(const Pair& other) const;
26     bool operator>(const Pair& other) const;
27
28     Pair& operator++(); // префиксный
29     Pair operator++(int); // постфиксный
30
31     friend ostream& operator<<(ostream& out, const Pair& p);
32     friend istream& operator>>(istream& in, Pair& p);
33 };
34
35 #endif
```

```
Pair.h Pair.cpp main.cpp
Злабаперделка (Глобальная область)

1 #include "Pair.h"
2
3 Pair::Pair() : first(0), second(0.0) {}
4
5 Pair::Pair(int f, double s) : first(f), second(s) {}
6
7 Pair::Pair(const Pair& other) : first(other.first), second(other.second) {}
8
9 Pair::~Pair() {}
10
11 int Pair::getFirst() const { return first; }
12
13 double Pair::getSecond() const { return second; }
14
15 void Pair::setFirst(int f) { first = f; }
16
17 void Pair::setSecond(double s) { second = s; }
18
19 Pair& Pair::operator=(const Pair& other) {
20     if (this != &other) {
21         first = other.first;
22         second = other.second;
23     }
24     return *this;
25 }
26
27 bool Pair::operator<(const Pair& other) const {
28     return (first < other.first) || (first == other.first && second < other.second);
29 }
30
31 bool Pair::operator>(const Pair& other) const {
32     return (first > other.first) || (first == other.first && second > other.second);
33 }
34
35 Pair& Pair::operator++() {
36     first++;
37     return *this;
38 }
39
40 Pair Pair::operator++(int) {
41     Pair temp = *this;
42     second++;
43     return temp;
44 }
45
46 ostream& operator<<(ostream& out, const Pair& p) {
47     out << p.first << ":" << p.second;
48     return out;
49 }

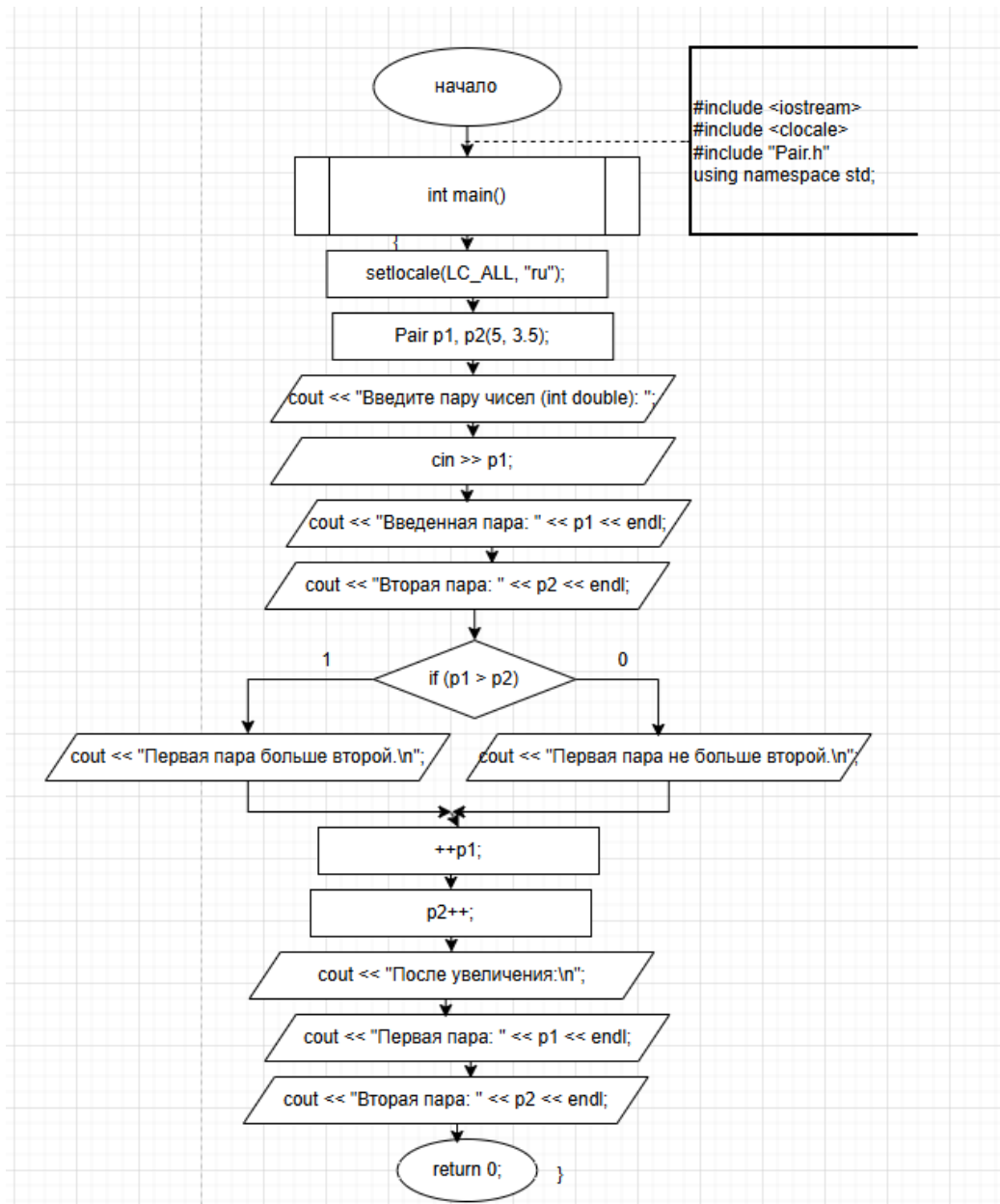
50
51 istream& operator>>(istream& in, Pair& p) {
52     in >> p.first >> p.second;
53     return in;
54 }
```

3 Результаты работы

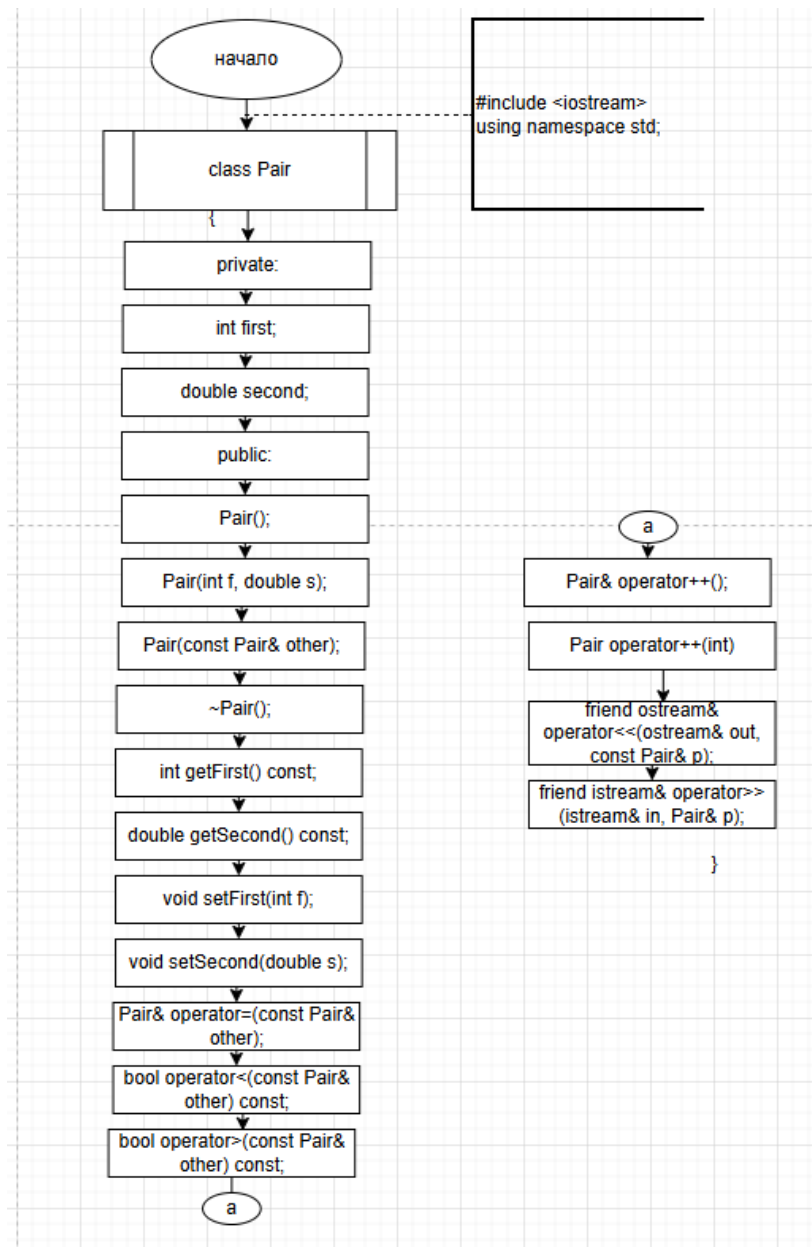
```
Введите пару чисел (int double): 2.2 4.2
Введенная пара: 2:0.2
Вторая пара: 5:3.5
Первая пара не больше второй.
После увеличения:
Первая пара: 3:0.2
Вторая пара: 5:4.5
```

4 Блок-схема

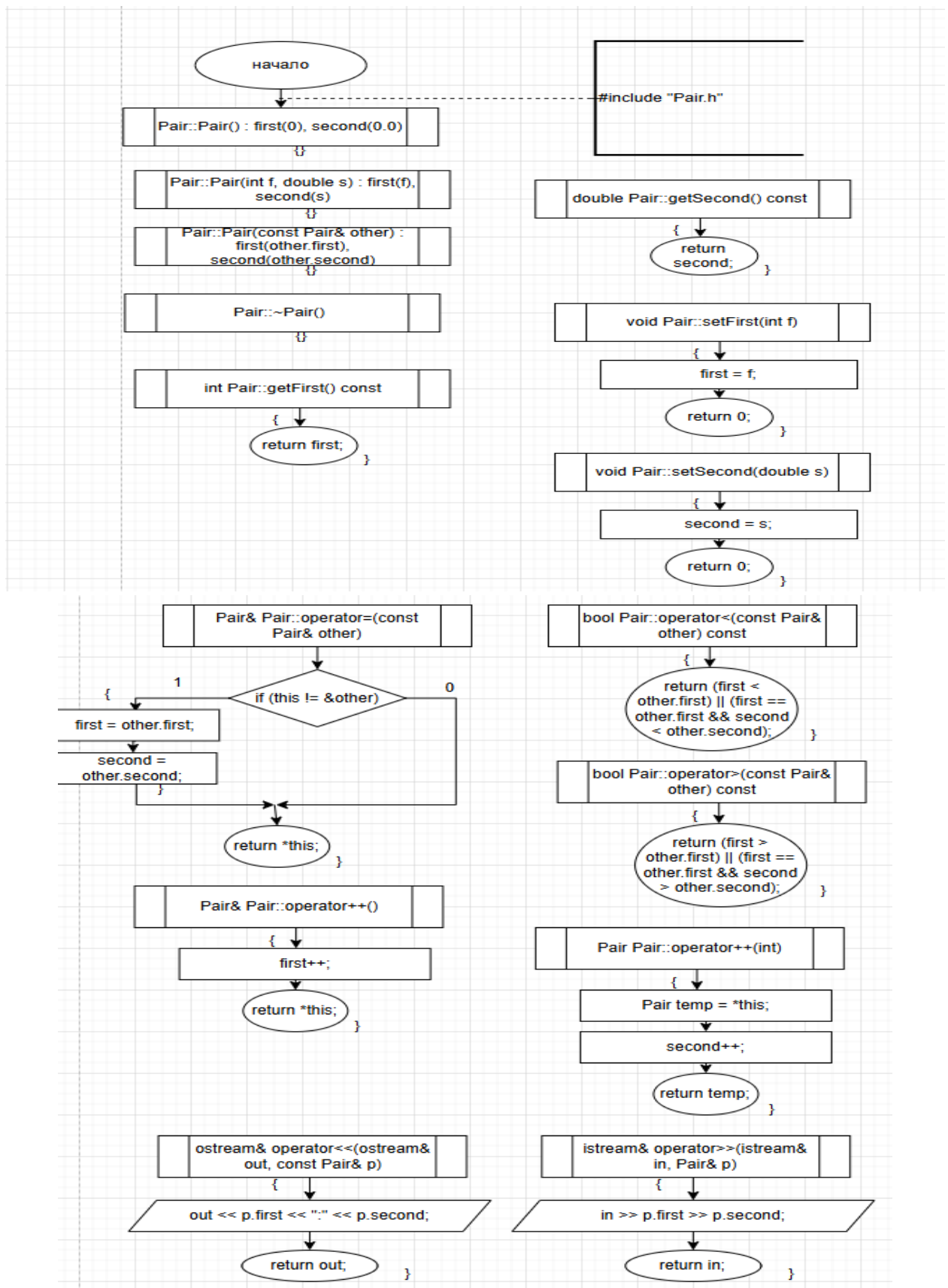
main.cpp



Pair.h



Pair.cpp



5 Ответы на контрольные вопросы

1. Для доступа к `private`-членам извне класса.
2. Описываются с ключевым словом `friend`, не являются методами класса, имеют доступ к закрытым членам.
3. Через метод внутри класса или как дружественную функцию вне класса.
4. Один операнд (неявный `this`).
5. Один параметр — объект класса.
6. Один параметр (второй — неявный `this`).
7. Два параметра — оба операнда передаются явно.
8. Постфиксная принимает дополнительный `int` параметр.
9. Через перегрузку `operator=`, обычно внутри класса.
10. Ссылку на текущий объект (`*this`) для поддержки цепочек присваивания.
11. Через дружественные функции `operator<<` и `operator>>` с потоками как первым параметром.
12. Как вызов метода-члена: `s.operator++()`;
13. Как вызов дружественной функции: `operator++(s)`;
14. Как `cout.operator<<(a<b)`;; сначала сравнение, потом вывод результата.
15. Как `operator>(a, b)`;; вызов дружественной функции с двумя параметрами.

6 Ссылка на github

ссылка на github - <https://github.com/MAKSPOWERO/mas1>