Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе 18.3**

Тема: Классы и объекты. Перегрузка операций.

Выполнил работу

студент группы РИС-20-1б

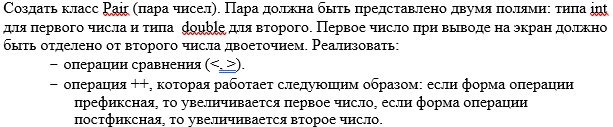
Томилов В.А.

Проверила

Доцент кафедры ИТАС

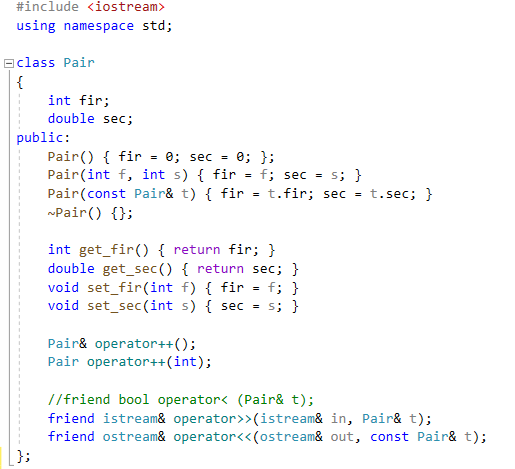
к.т.н ПоляковаО.А.

Пермь, 2021

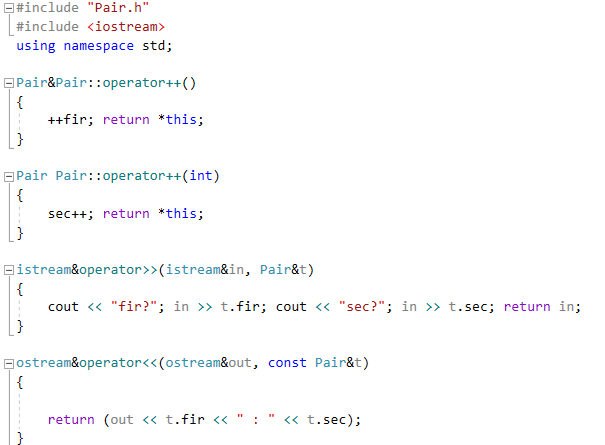
Постановка задачи

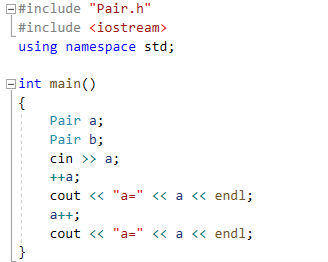
Анализ задачи

* 1. Для решения задачи необходимо
  2. Добавить в проект файл Pair.h, содержащий описание класса.



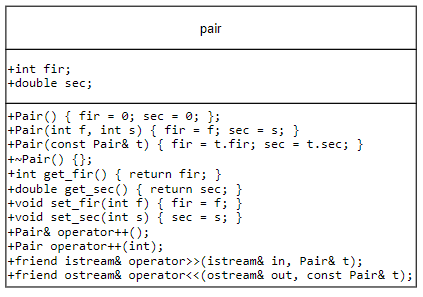
* 1. Добавим в проект файл Pair.cpp, содержащий описание методов класса Pair.



* 1. Добавить в проект файл Lab3\_main.cpp, содержащий основную программу.
  2. В ходе работы были использованы следующие типы данных:

2.1 (int) fir – первое число

2.2 (double) sec – второе число

UML – диаграмма

Код программы

Main.cpp

#include "Pair.h"

#include <iostream>

using namespace std;

int main()

{

Pair a;

Pair b;

cin >> a;

++a;

cout << "a=" << a << endl;

a++;

cout << "a=" << a << endl;

}

Pair.h

#include <iostream>

using namespace std;

class Pair

{

int fir;

double sec;

public:

Pair() { fir = 0; sec = 0; };

Pair(int f, int s) { fir = f; sec = s; }

Pair(const Pair& t) { fir = t.fir; sec = t.sec; }

~Pair() {};

int get\_fir() { return fir; }

double get\_sec() { return sec; }

void set\_fir(int f) { fir = f; }

void set\_sec(int s) { sec = s; }

Pair& operator++();

Pair operator++(int);

//friend bool operator< (Pair& t);

friend istream& operator>>(istream& in, Pair& t);

friend ostream& operator<<(ostream& out, const Pair& t);

};

Pair.cpp

#include "Pair.h"

#include <iostream>

using namespace std;

Pair&Pair::operator++()

{

++fir; return \*this;

}

Pair Pair::operator++(int)

{

sec++; return \*this;

}

istream&operator>>(istream&in, Pair&t)

{

cout << "fir?"; in >> t.fir; cout << "sec?"; in >> t.sec; return in;

}

ostream&operator<<(ostream&out, const Pair&t)

{

return (out << t.fir << " : " << t.sec);

}

Ответы на вопросы

1. Дружественная функция — это функция, которая имеет доступ к закрытым членам класса, как если бы она сама была членом этого класса. Во всех других отношениях дружественная функция является обычной функцией. Ею может быть, как обычная функция, так и метод другого класса.
2. Для объявления дружественной функции используется ключевое слово friend перед прототипом функции, которую вы хотите сделать дружественной классу.
3. 4. 5. Любая унарная операция может быть определена двумя способами: либо как компонентная функция без параметров, либо как глобальная (возможно дружественная) функция с одним параметром.

6. 1

7. произвольное

8. operator ++();

operator ++(int);

Указание параметра int для постфиксной формы не специфицирует второй операнд, а используется только для отличия от префиксной формы.

9. Операция может перегружаться только в области определения класса. Это гарантирует, что первым операндом всегда будет леводопустимое выражение.

10. Оператор присваивания возвращает ссылку на свой левый аргумент

11. friend istream& operator>>(istream& in, Pair& t);

friend ostream& operator<<(ostream& out, const Pair& t);