Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа по ООП**

**«№3»**

Выполнил:

студент группы РИС-23-2б

Ившин Максим Сергеевич

Проверила:

доцент кафедры ИТАС

О.А. Полякова

2024 г.

**Разработка алгоритма**

**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).

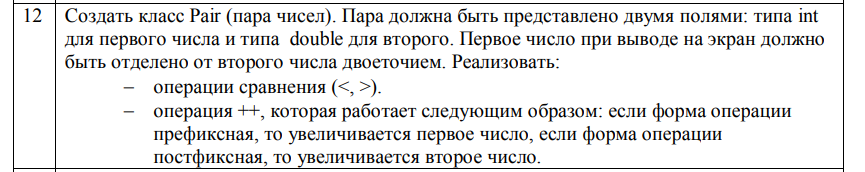
5. Перегрузить операцию присваивания.

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:



**Анализ задачи:**

1. В классе Pair определяем пару значений с типами данных integer и double.
2. Создаем конструкторы, деструктор, операторы сравнения (< и >), префиксные и постфиксные операторы приращения, операторы ввода/вывода (<<) и методы получения/установки полей класса get() set().
3. Операторы сравнения позволяют нам сравнивать два объекта Pair на основе суммы их значений.
4. Перегрузка оператора инкремента позволяют увеличивать либо первое, либо второе поле, в зависимости от того, постфиксный или префиксный оператор был применен к обьекту.
5. Перегрузка оператора вывода (<<) позволяет выводить объекты Pair в стандартный поток вывода.

Код на C++:

Pair.cpp:

#pragma once

#include <iostream>

class Pair

{

private:

    int first;

    double second;

public:

    Pair();

    Pair(int, double);

    Pair(Pair &);

    ~Pair();

    bool operator<(Pair &);

    bool operator>(Pair &);

    Pair &operator++();

    Pair &operator++(int);

    friend std::ostream &operator<<(std::ostream &out, Pair i);

    void get();

    void set(int, double);

};

Pair::Pair()

{

    this->first = 0;

    this->second = 0;

}

Pair::Pair(int fir, double sec)

{

    this->first = fir;

    this->second = sec;

}

Pair::Pair(Pair &copying)

{

    this->first = copying.first;

    this->second = copying.second;

}

Pair::~Pair()

{

    std::cout << "\nВызван деструктор";

}

bool Pair::operator<(Pair &temp)

{

    if ((this->first + this->second) < (temp.first + temp.second))

        return true;

    return false;

}

bool Pair::operator>(Pair &temp)

{

    if ((this->first + this->second) > (temp.first + temp.second))

        return true;

    return false;

}

Pair &Pair::operator++()

{

    ++this->first;

    return \*this;

}

Pair &Pair::operator++(int)

{

    ++this->second;

    return \*this;

}

void Pair::get()

{

    std::cout << "first = " << this->first << ", second = " << this->second;

}

void Pair::set(int fir, double sec)

{

    this->first = fir;

    this->second = sec;

}

std::ostream &operator<<(std::ostream &pairout, Pair temp)

{

    return (pairout << temp.first << ':' << temp.second);

}

main.cpp:

#include <iostream>

using namespace std;

#include "Pair.cpp"

int main()

{

    Pair a(1, 5.5);

    Pair b;

    a.get();

    cout << endl;

    b.get();

    cout << endl;

    b.set(6, 2.2);

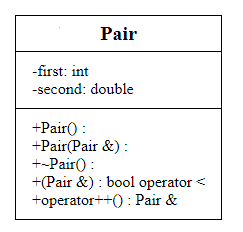
    cout << endl << a << endl << b<< endl;

    cout << endl << (a > b);

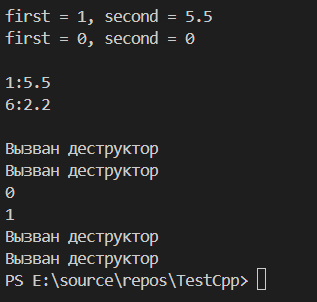
    cout << endl << (a < b);

}

Диаграмма класса:



Пример работы программы:



**GitHub:** *https://github.com/geroineee/PNRPU*