МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа № 3:**

"Перегрузка операторов"

Выполнила: студентка гр.РИС-23-3Б

Мокроусова Ангелина Андреевна

Проверила: доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2024

**Постановка задачи:**

Общая часть:

* Определить пользовательский класс
* Определить в классе следующие конструкторы: без параметров, с параметрами, копирования
* Определить в классе деструктор
* Определить в классе компоненты-функций для просмотра и установки полей данных ( селекторы и модификаторы)
* Перегрузить операции присваивания
* Перегрузитьоперации ввода и вывода объектов с помощью потоков и указанные в варианте
* Перегрузить операции указанные в варианте
* Написать программу в которой продемонстрировать создание объектов и работу всех перегруженных операций

Задание варианта:

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 уменьшается первое число, если форма операции постфиксная, то уменьшается второе число

**Анализ задачи:**

* Необходимо правильно определить тип возвращаемого значения и типы параметров для каждой перегруженной операции.
* Операции сравнения должны возвращать логическое значение (true или false).
* Необходимо правильно определить логику сравнения для операций < и >.
* Операция -- должна уменьшать значение либо первого, либо второго поля в зависимости от формы операции (префиксная или постфиксная).
* Необходимо правильно реализовать логику уменьшения значения поля.
* При перегрузке операторов ввода (>>) и вывода (<<) необходимо определить формат ввода и вывода данных.
* Для ввода и вывода данных необходимо использовать объекты потоков (например, cin и cout).

**Код на С++ с комментариями:**

#include <iostream>

using namespace std;

class Pair {

private:

int first;

double second;

public:

Pair() { // Конструктор по умолчанию

first = 0;

second = 0.0;

}

Pair(int first, double second) { // Конструктор с параметрами

this->first = first;

this->second = second;

}

Pair(const Pair& other) { // Конструктор копирования

first = other.first;

second = other.second;

}

~Pair() { // Деструктор

}

int getFirst() const { return first; } // Селекторы

double getSecond() const { return second; }

void setFirst(int first) { this->first = first; } // Модификаторы

void setSecond(double second) { this->second = second; }

Pair& operator=(const Pair& other) { // Операция присваивания

first = other.first;

second = other.second;

return \*this;

}

bool operator<(const Pair& other) const { // Операции сравнения

return first < other.first;

}

bool operator>(const Pair& other) const {

return second > other.second;

}

Pair& operator--() { // Операция -- (префиксная)

first--;

return \*this;

}

Pair& operator--(int) { // Операция -- (постфиксная)

second--;

return \*this;

}

friend istream& operator>>(istream& in, Pair& pair) { // Операция ввода

in >> pair.first >> pair.second;

return in;

}

friend ostream& operator<<(ostream& out, const Pair& pair) { // Операция вывода

out << pair.first << ":" << pair.second;

return out;

}

};

int main() {

setlocale(LC\_ALL, "ru");

Pair pair1;

cout << "Введите первую пару чисел: " << endl;

cin >> pair1;

cout << "Введенная пара: " << pair1 << endl << endl;

Pair pair2;

cout << "Введите вторую пару чисел: " << endl;

cin >> pair2;

cout << "Введенная пара: " << pair2 << endl << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "Демонстрация операции <>: \n";

cout << "1. (<)\t" << pair1.getFirst() << "<" << pair2.getFirst() << " = " << (pair1 < pair2) << endl;

cout << "2. (>)\t" << pair1.getSecond() << ">" << pair2.getSecond() << " = " << (pair1 > pair2) << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

cout << "Демонстрация операции --: \n";

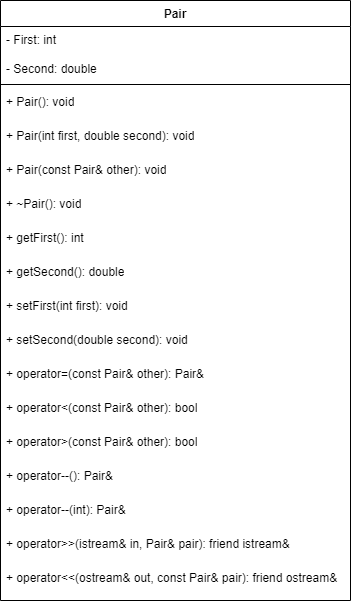
cout << "1. (0--)\t" << "\t" << pair1-- << "\t" << "Изменилось: " << pair1.getSecond() << "\n";

cout << "2. (--0)\t" << "\t" << --pair2 << "\t" << "Изменилось: " << pair2.getFirst() << "\n";

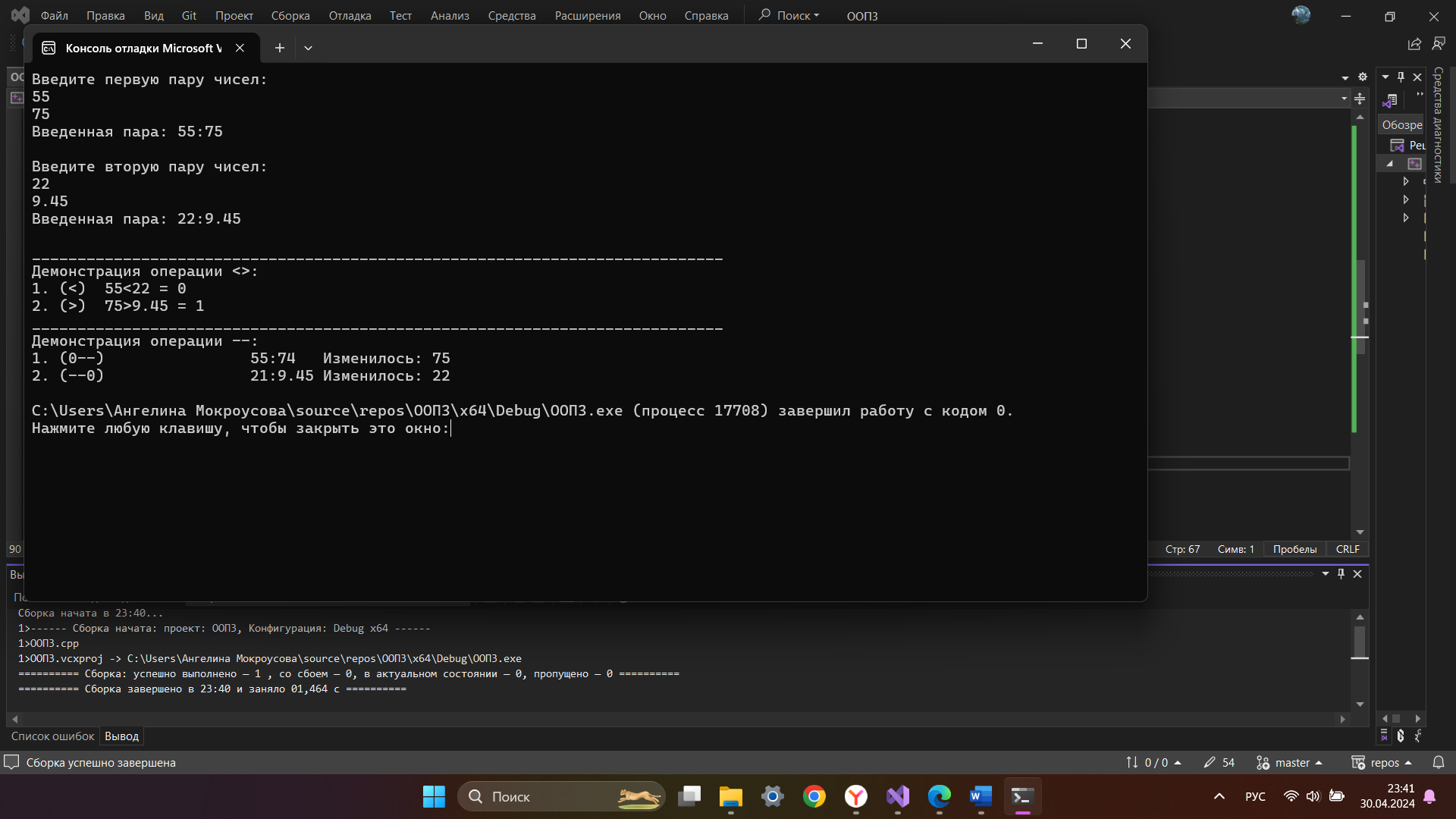
return 0;

}

**UML диаграмма:**

****

**Скриншот работы программы:**



**Вывод:**

Мною была аписана программа, которая успешно выполняет поставленную перед ней задачу. В процессе написания был повторен и закреплен на практике пройденный ранее материал.

**Ответы на вопросы:**

1. Для чего используются дружественные функции и классы? Дружественные функции и классы используются для предоставления функциям и классам неограниченный доступ к полям класса;
2. Сформулировать правила описания и особенности дружественных функций.

Дружестванная функция должна быть объявлена внутри класса с ключевым словом friend, в качестве параметра должна принимать экземпляр класса, так как указатель this ей не доступен, имеет неограниченный доступ к полям класса, может быть дружественной сразу к нескольким классам, может быть перегруженна, однако перегрузка не будет дружественной, если её отдельно не описать, как таковую;

1. Каким образом можно перегрузить унарные операции?

Унарные операции можно перегрузить как метод класса или же как отдельную функцию;

1. Сколько операндов должна иметь унарная функция-операция, определяемая внутри класса?

Один операнд – экземпляр класса, для которого перегружена функция;

1. Сколько операндов должна иметь унарная функция-операция, определяемая вне класса?

Унарная функция всегда имеет один операн;

1. Сколько операндов должна иметь бинарная функция-операция, определяемая внутри класса?

Два операнда. Первый – сам объект класса, для которого перегружена операция, второй – объект, с которым взаимодейтсвует объект класса;

1. Сколько операндов должна иметь бинарная функция-операция, определяемая вне класса?

Два. Кол-во операндов не меняется от месте определения перегрузки;

1. Чем отличается перегрузка префиксных и постфиксных унарных операций?

Префиксная функция не имеет параметров и возвращает ссылку на преобразованный объект класса. Постфиксная операция имеет один параметр типа int и возвращает объект класса до его преобразований;

1. Каким образом можно перегрузить операцию присваивания?

Операция присваивания должна принимать в себя ссылку на другой объект того же класса и возвращать ссылку на левый операнд;

1. Что должна возвращать операция присваивания?

Операция присваивания додлжна возвращать ссылку на левый операнд;

1. Каким образом можно перегрузить операции ввода-вывода?

Операции ввода-вывода можно перегрузить только как внешние функции. По необходимости могут быть дружественными;

1. Компилятор будет воспринимать вызов функции-операции как вызов перегрузки унарного префиксного оператора для класса Student;
2. Компилятор будет воспринимать вызов операции как вызов перегрузки унарного префиксного оператора для класса Student;
3. Компилятор воспримет данный вызов как ошибку. Оператор << имеет более высокий приоритет чем <, из чего следует, что оператору < предложат сравнить типы данных ostream и student, однако такой перегрузки не существует;
4. Компилятор воспримет данный вызов как ошибку. Оператор << имеет более высокий приоритет чем > из чего следует, что оператору > предложат сравнить типы данных ostream и student, однако такой перегрузки не существует;

**Скриншот с GitHab:**