Министерство образования и науки Российской Федерации

Федеральное государственное образовательное автономное учреждение высшего образования

"Пермский национальный исследовательский политехнический университет"

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Дисциплина: Основы алгоритмизации и программирования

Тема: Перегрузка операций

Вариант: 13

|  |  |
| --- | --- |
| Выполнил работу | |
| Студент группы РИС-22-1б | |
| Кирпичников Илья | |
|  | |
| Проверил работу | |
| Доцент кафедры ИТАС | |
| Полякова О.А. | |
|  | |

Пермь – 2023

**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.

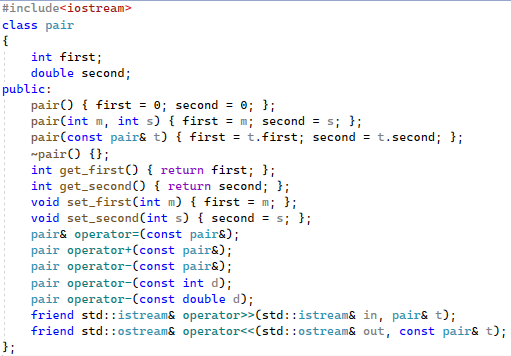
Вариант 15:

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

Реализовать:

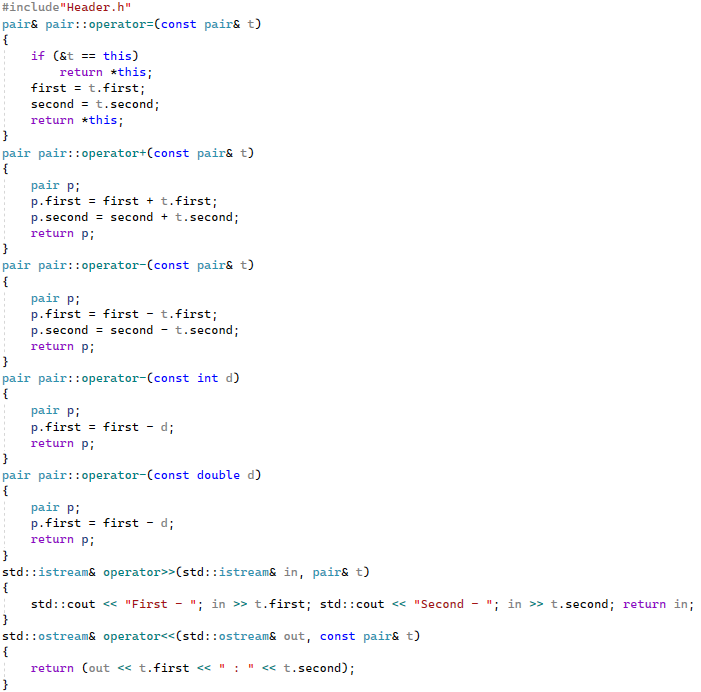
1. Вычитание пар чисел
2. Добавление константы к паре (увеличивается первое число, если константа целая, второе, если константа вещественная).

**Описание классов**

****

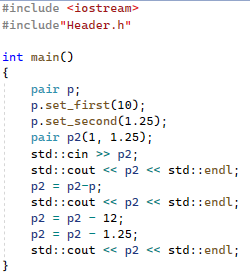
*Приложение 1 – Описание классов*

**Определение компонентных функций**

****

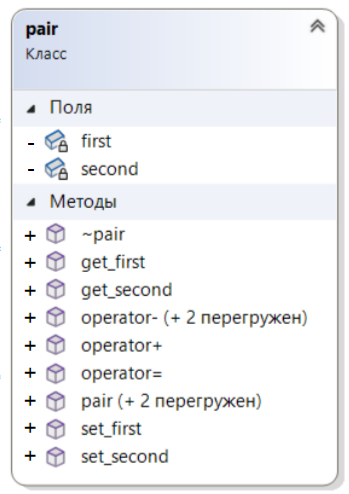
*Приложение 2 – Определение компонентных функций*

**Функция main()**

****

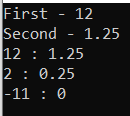
*Приложение 3 – Главная функция*

**UML-диаграмма**



*Приложение 4 – UML-диаграмма*

**Работа программы**



*Приложение 5 – Работа программы*

**Ответы на контрольные вопросы**

1. Дружественные функции и классы используются для предоставления доступа к закрытым членам класса другим функциям и классам.

2. Правила описания дружественных функций:

- Они должны быть объявлены вне класса, но определены внутри него.

- Они должны быть объявлены с помощью ключевого слова friend.

- Они могут иметь доступ к закрытым членам класса.

Особенности дружественных функций:

- Они не являются методами класса, но имеют доступ к его закрытым членам.

- Их вызов осуществляется извне класса, как обычных функций.

3. Унарные операции можно перегрузить с помощью ключевого слова operator и названия операции (например, operator+ для унарного плюса).

4. Унарная функция-операция, определяемая внутри класса, должна иметь один операнд - объект класса, для которого она вызывается.

5. Унарная функция-операция, определяемая вне класса, также должна иметь один операнд - объект класса, для которого она вызывается.

6. Бинарная функция-операция, определяемая внутри класса, должна иметь два операнда - объекты класса, между которыми производится операция.

7. Бинарная функция-операция, определяемая вне класса, также должна иметь два операнда - объекты класса, между которыми производится операция.

8. Префиксная унарная операция выполняется над объектом до его использования в выражении, а постфиксная - после использования в выражении. Например, ++a и a++.

9. Операцию присваивания можно перегрузить с помощью ключевого слова operator=.

10. Операция присваивания должна возвращать ссылку на объект, для которого она вызывается (this).

11. Операции ввода-вывода можно перегрузить с помощью ключевых слов operator<< и operator>> соответственно.

12. Компилятор будет воспринимать вызов оператора префиксного инкремента для объекта s класса Student, который был перегружен внутри класса и возвращает ссылку на этот же объект.

13. Компилятор будет воспринимать вызов дружественной функции-оператора префиксного инкремента для объекта s класса Student, который был определен вне класса и возвращает ссылку на этот же объект.

14. Компилятор будет воспринимать вызов оператора меньше для объектов a и b класса Student, который был определен внутри класса и возвращает булево значение.

15. Компилятор будет воспринимать вызов дружественной функции-оператора больше для объектов a и b класса Student, который был определен вне класса и возвращает булево значение.