Пермский национальный исследовательский политехнический университет

Кафедра «Информационные технологии и автоматизированные системы»

Лабораторная работа 3

**Классы и объекты. Использование конструкторов.**

Выполнил

Студент гр. РИС-24-3б Рунтов Иван Андреевич

Проверил

Доцент кафедры ИТАС Ольга Андреевна Полякова

г. Пермь, 2025

**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.

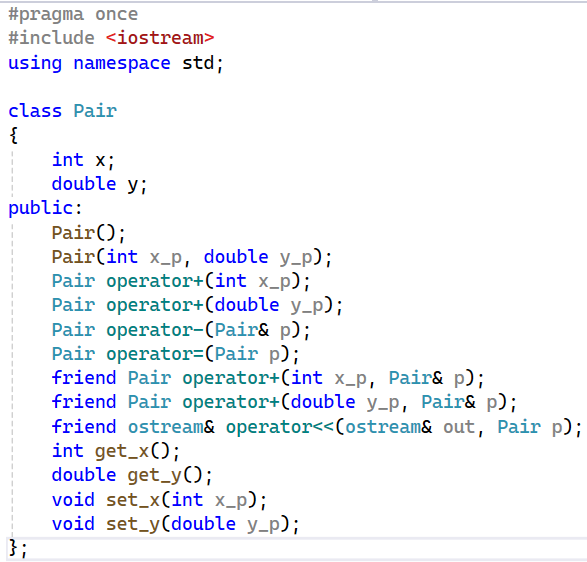
8. Написать программу, в которой продемонстрировать создание объектов и работу всех перегруженных операций.

**Вариант 15**

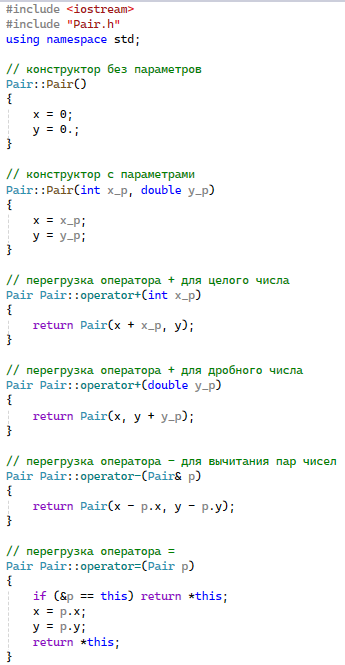
Создать класс Pair (пара чисел). Пара должна быть представлено двумя полями: типа int для первого числа и типа double для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием. Реализовать:

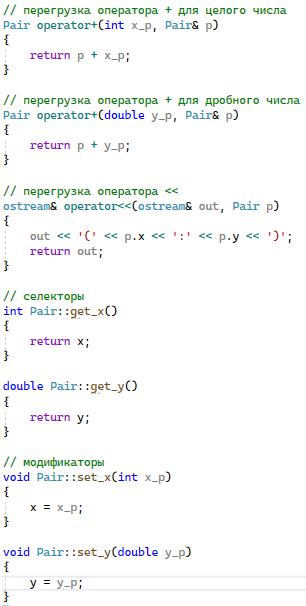
* вычитание пар чисел
* добавление константы к паре (увеличивается первое число, если константа целая, второе, если константа вещественная).

**Описание класса Pair**

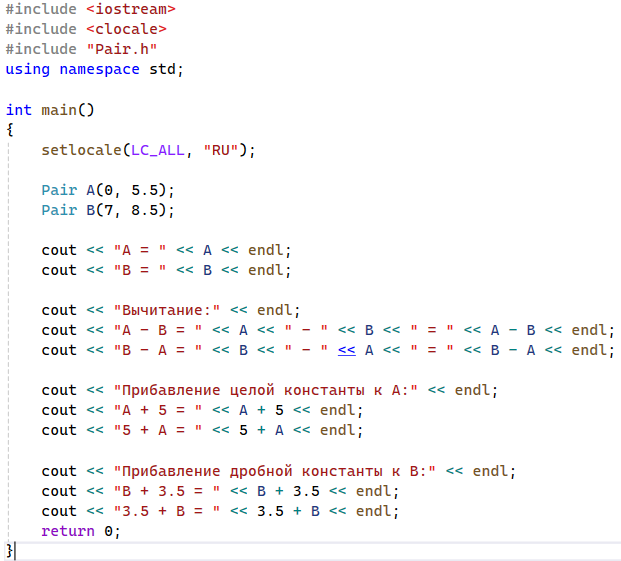


**Описание методов класса и дружественных функций**





**Основной файл**



**Ответы на контрольные вопросы**



Дружественные функции и классы используются для доступа к защищенным членам класса в любой части программы. Они не участвуют в реализации класса.



Дружественный функции объявляются внутри класса в любом месте с ключевым словом friend. Описываются вне класса.

В качестве параметра должны принимать ссылку на объект этого класса, так как не принимают по умолчанию указатель this.

Дружественная функция может быть дружественна сразу нескольким классам.



Перегрузить унарные указатели можно как методы класса или как глобальные дружественные функции этого класса.



Унарная функция-операция, определяемая внутри класса, должна не принимать параметров, поскольку операндом будет объект, который ее вызвал.



Унарная функция-операция, определяемая вне класса, должна принимать один параметр (объект класса), который будет являться операндом.



Бинарная функция-операция, определяемая внутри класса, должна иметь один параметр (объект класса). В этом случае объект, который вызвал операцию, будет левым операндом, а объект из параметров будет правым операндом.



Бинарная функция-операция, определяемая вне класса, должна иметь два параметр (объекты класса). В этом случае объект из первого параметра будет левым операндом, а объект из правого параметра будет правым операндом.



Перегрузка префиксных и постфиксных унарных операций отличается следующим: при перегрузке префиксных операций в параметры ничего не передается, а при перегрузке постфиксных операций в параметрах указывает int.



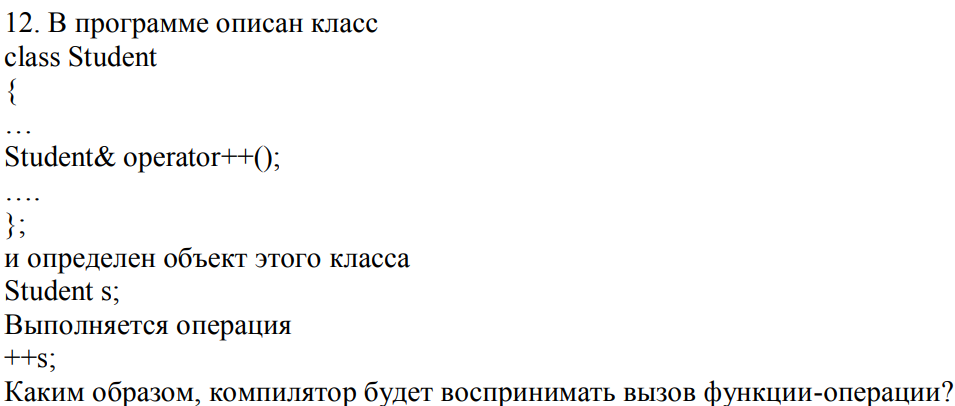
Перегрузить операцию присваивания можно только как нестатическую компоненту класса.



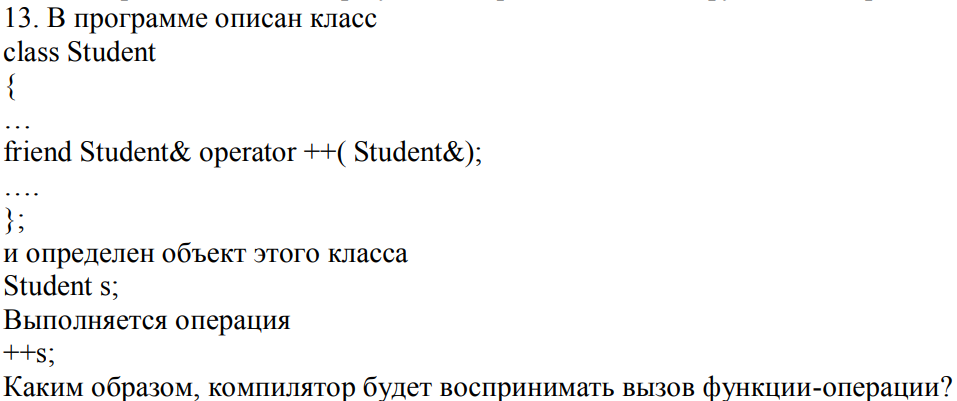
Операция присваивания должна возвращать ссылку на объект, в который происходило копирование.



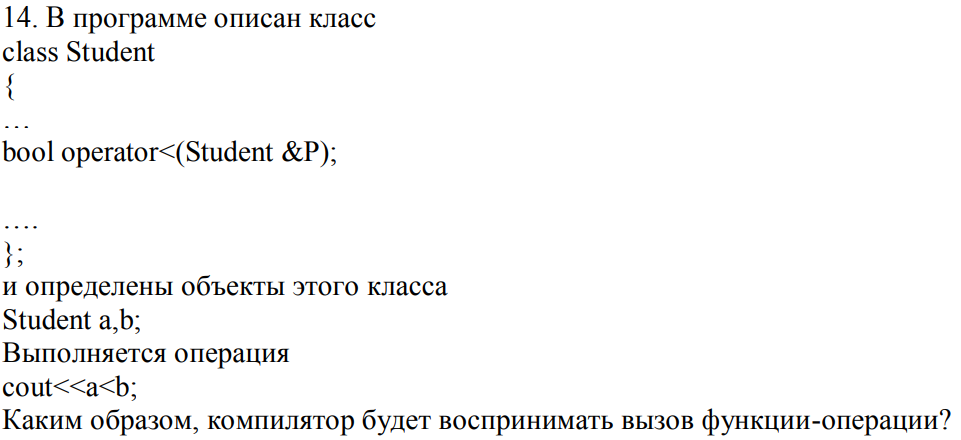
Операции ввода-вывода можно перегрузить только как дружественные классу функции. Это можно реализовать только таким образом, потому что левым операндом в этих операциях является поток.



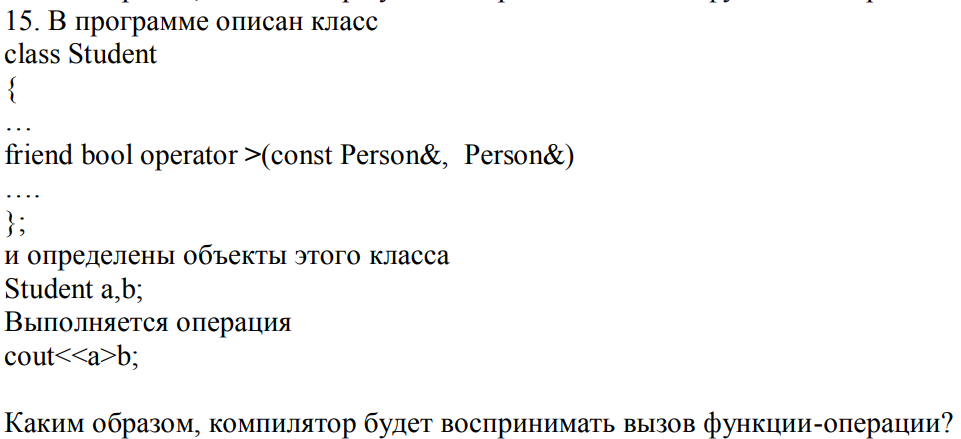
Компилятор воспримет вызов как вызов метода класса.



Компилятор воспримет вызов как вызов глобальной функции.



Так как приоритет операции << выше приоритета операции <, то программа не сможет скомпилироваться. Если бы операция сравнения была прописана в (), то компилятор воспринял бы ее вызов как вызов метода класса.



Так как приоритет операции << выше приоритета операции >, то программа не сможет скомпилироваться. Если бы операция сравнения была прописана в (), то компилятор воспринял бы ее вызов как вызов глобальной функции.

Ссылка на github: <https://github.com/RuntovIvan/Informatika>