Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

Лабораторная работа №3.

" Перегрузка операций. "

15 вариант

Выполнил студент гр. РИС-24-2б

Иванова Елена Олеговна

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

Цель:

1. Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.
2. Использование перегруженных операций в классах

Задание:

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).

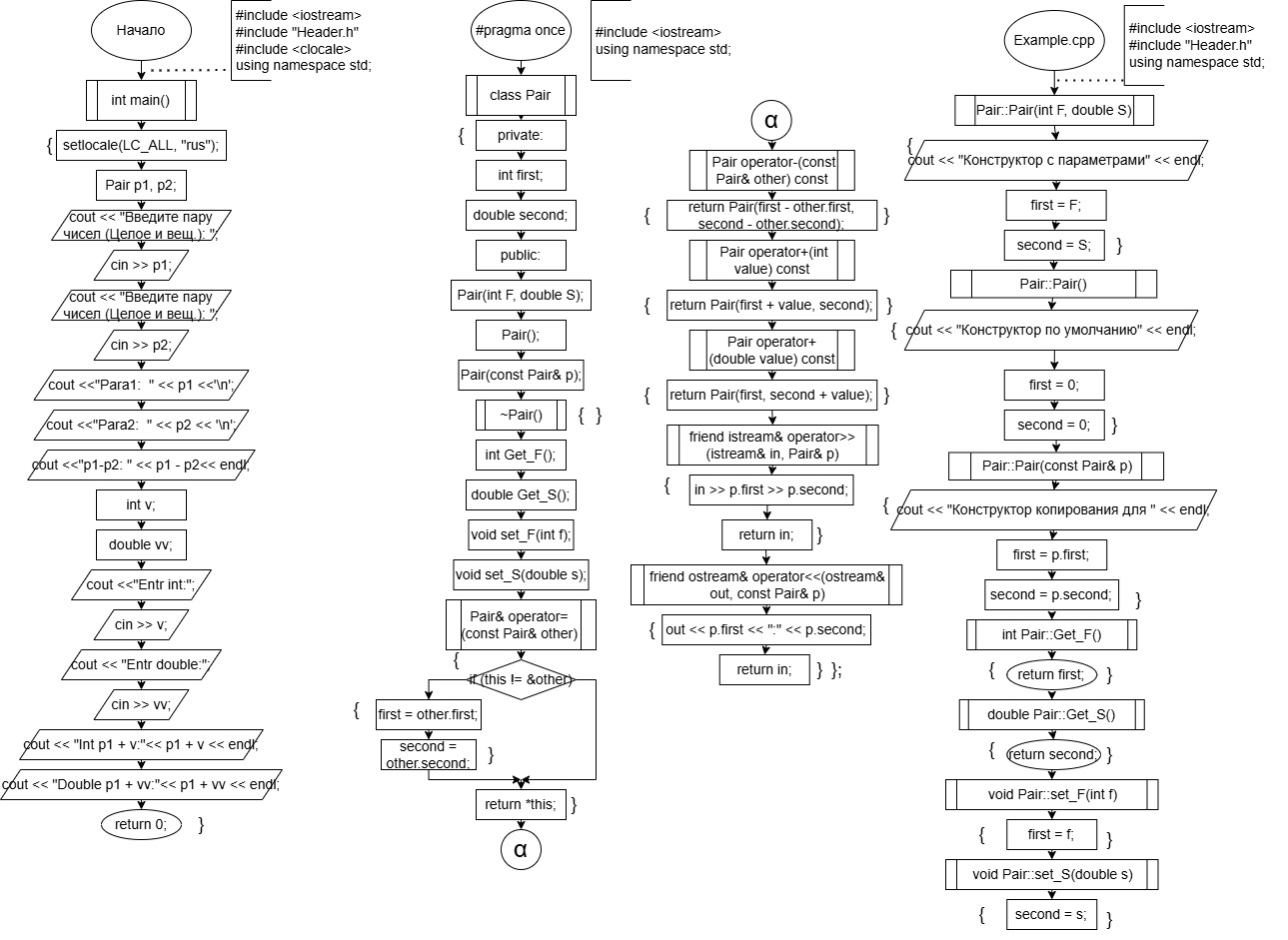
Постановка задачи

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.

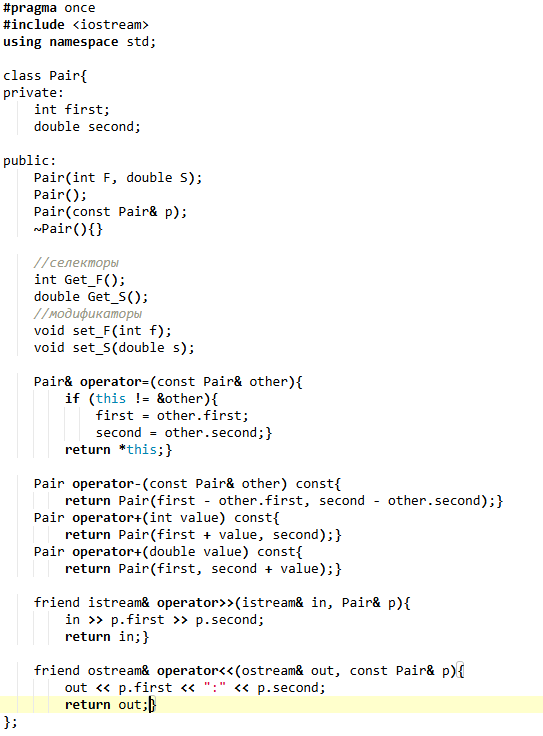
Анализ задачи.

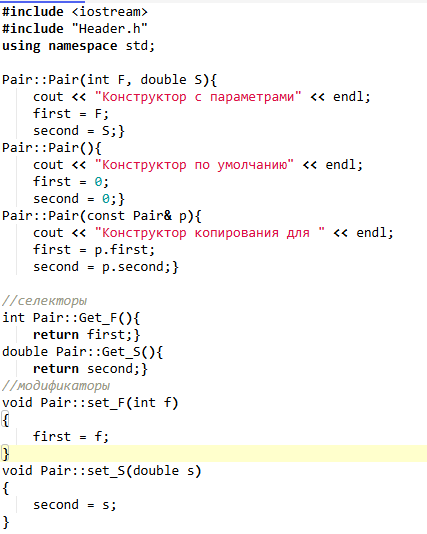
1. Определить конструкторы: без параметров, с параметрами, копирования и деструктор.
2. Определить селекторы и модификаторы
3. Перегрузить операцию присваивания.
4. Перегрузить операции ввода и вывода.
5. Перегрузить операции вычитания пар чисел и добавления константы к паре.

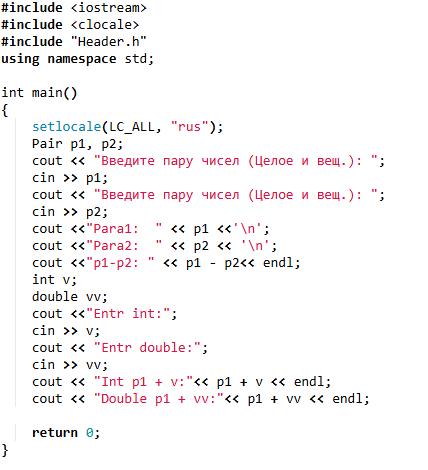
Блок схема.



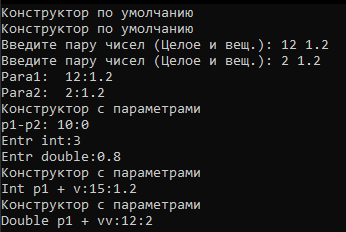
Код.







Вывод.



Контрольные вопросы

1. Для чего используются дружественные функции и классы?

Дружественные функции и классы используются для предоставления доступа к private и protected членам класса другим функциям или классам, которые не являются его членами.

1. Сформулировать правила описания и особенности дружественных функций.

Правила описания дружественных функций:

1. Дружественная функция должна быть объявлена внутри класса с ключевым словом friend.
2. Она не является методом класса, поэтому её определение должно быть вне класса.
3. Если класс наследуется, дружественные функции не передаются производным классам.
4. У неё нет указателя this.
5. Дружественные функции могут быть шаблонными.

Особенности дружественных функций:

1. Дружественная функция может обращаться к private и protected полям и методам класса.
2. Не может использовать this.
3. Одна функция может быть дружественной для нескольких классов.
4. Часто применяются для перегрузки операторов ввода/вывода.
5. Каким образом можно перегрузить унарные операции?
6. Сколько операндов должна иметь унарная функция-операция, определяемая внутри класса?
7. Сколько операндов должна иметь унарная функция-операция, определяемая вне класса?
8. Сколько операндов должна иметь бинарная функция-операция, определяемая внутри класса?
9. Сколько операндов должна иметь бинарная функция-операция, определяемая вне класса?
10. Чем отличается перегрузка префиксных и постфиксных унарных операций?
11. Каким образом можно перегрузить операцию присваивания?
12. Что должна возвращать операция присваивания?
13. Каким образом можно перегрузить операции ввода-вывода?
14. В программе описан класс

class Student {… Student& operator++(); …. };

и определен объект этого класса Student s;

Выполняется операция ++s; Каким образом, компилятор будет воспринимать вызов функции-операции?

1. В программе описан класс class Student {… friend Student& operator ++( Student&); …. }; и определен объект этого класса Student s; Выполняется операция ++s; Каким образом, компилятор будет воспринимать вызов функции-операции?
2. В программе описан класс class Student {… bool operator<(Student &P); …. }; и определены объекты этого класса Student a,b; Выполняется операция cout<<a<b; Каким образом, компилятор будет воспринимать вызов функции-операции?
3. В программе описан класс class Student {… friend bool operator >(const Person&, Person&) …. }; и определены объекты этого класса Student a,b; Выполняется операция cout<b; Каким образом, компилятор будет воспринимать вызов функции-операции?