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 №10.**

**«Сохранение данных в файле с использованием потоков»**

Выполнил студент гр. РИС-24-2б

Молочко Артём Анатольевич

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

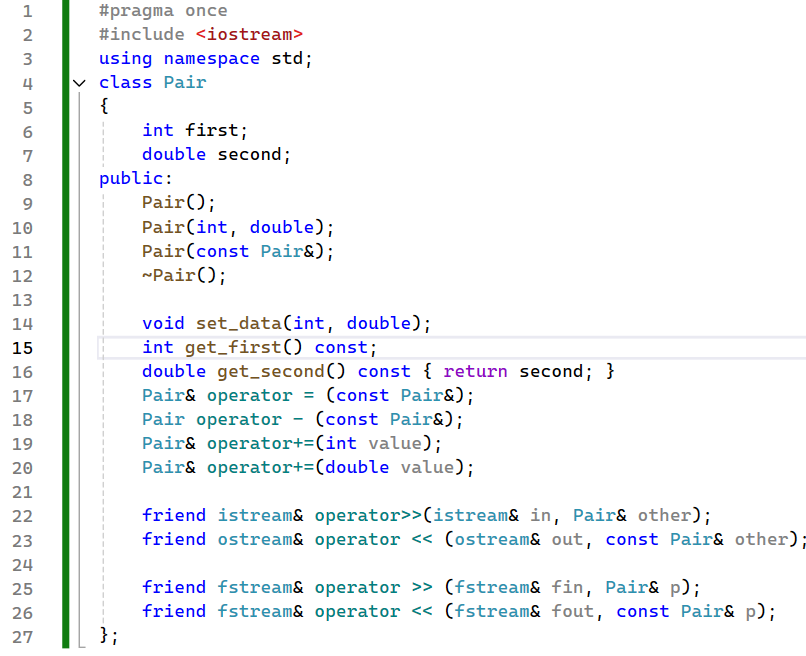
**Постановка задачи**

1. Создать класс Pair (пара чисел). Пара должна быть представлена 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).
2. Написать функции для работы с файлами:
   * Функция для создания объектов класса Pair (ввод с клавиатуры) и сохранения их в файле
   * Функция для чтения и просмотра объектов из файла
   * Функция для удаления всех записей меньших заданного значения
   * Функция для увеличения всех записей с заданным значением на число L
   * Функция для добавления K записей после элемента с заданным номером
3. Для вызова функций в основной программе предусмотреть меню с возможностью:
   * Создания нового файла
   * Просмотра содержимого файла
   * Удаления записей
   * Увеличения записей
   * Добавления новых записей
   * Выхода из программы

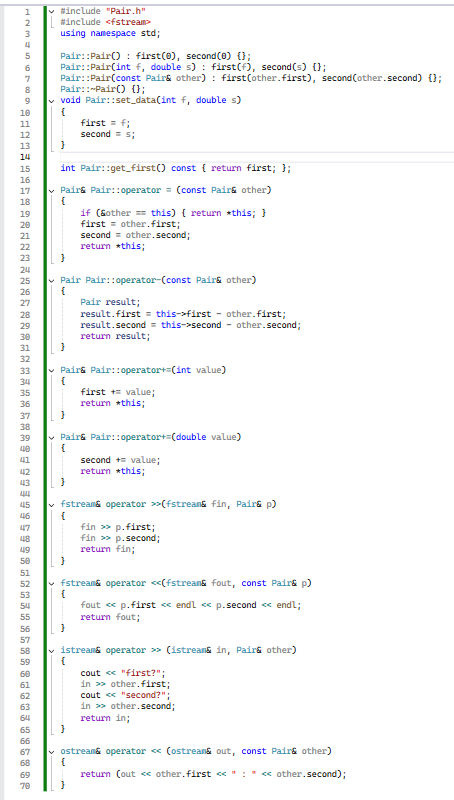
**Анализ задачи**

* Класс инкапсулирует два поля: int first и double second.
* Реализованы конструкторы (по умолчанию, параметризированный, копирующий) и деструктор.
* Перегружены операторы:
  + operator- для вычитания пар
  + operator+= для добавления константы (целой и вещественной)
  + operator>> и operator<< для ввода/вывода (как в консоль, так и в файл)
* Добавлены методы доступа: get\_first(), get\_second().
* make\_file: Создание файла и запись пар (ввод с клавиатуры).
* print\_file: Чтение и вывод содержимого файла.
* del: Удаление пар, меньших заданного значения.
* increase: Увеличение пар с заданным значением на число L.
* insert\_after\_position: Вставка K записей после указанной позиции.

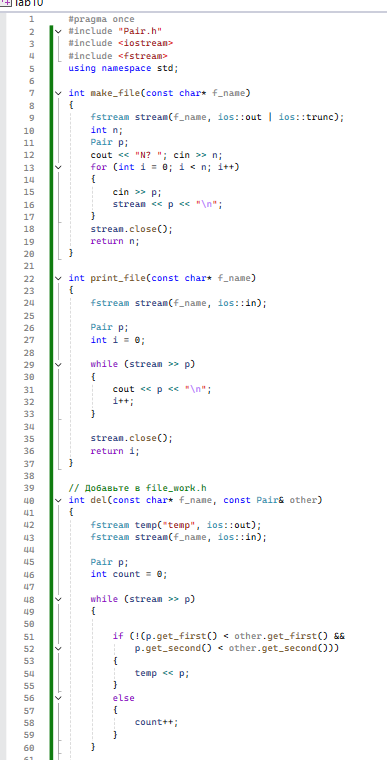
Код программы

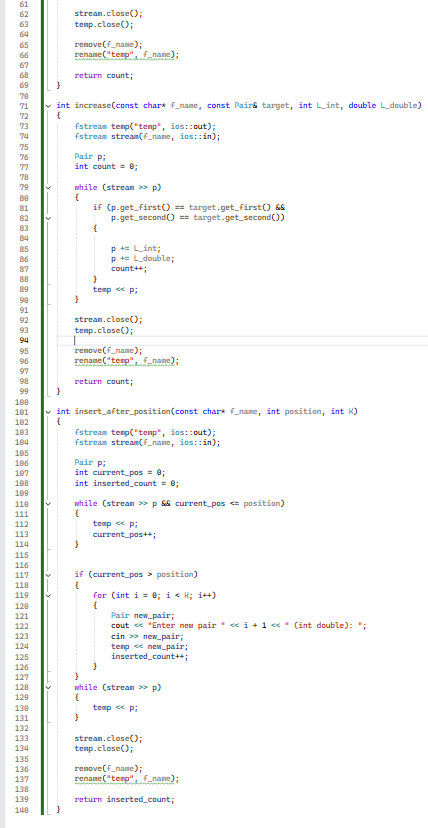
Pair.h

Pair.cpp

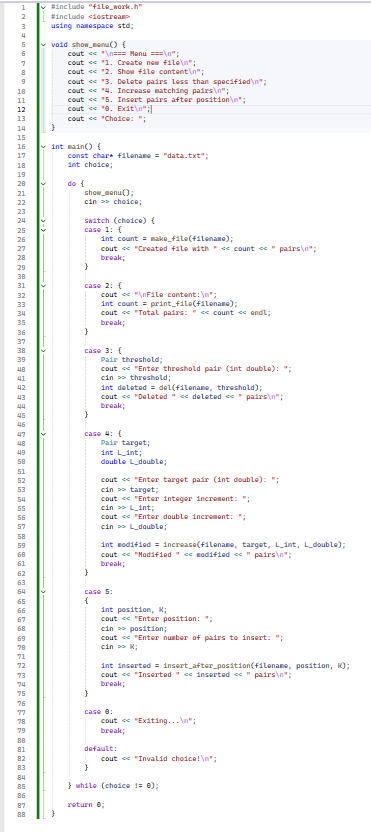


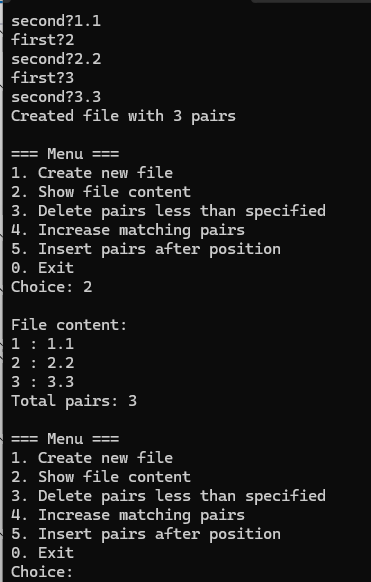
File\_work.h

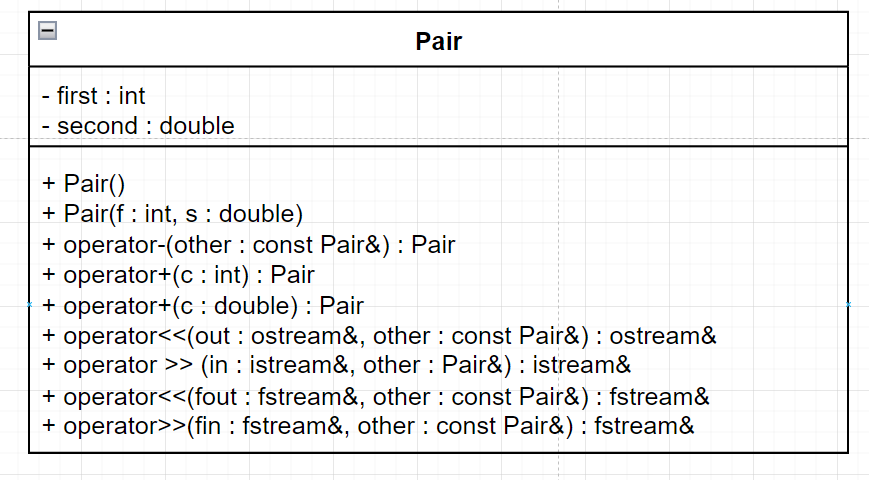




Main.cpp







**Контрольные вопросы**

**1. Что такое поток?**  
**Ответ:** Поток (stream) — это абстракция для работы с последовательностью данных (ввод/вывод).

**2. Какие типы потоков существуют?**  
**Ответ:**

* Консольные (cin, cout, cerr, clog)
* Файловые (ifstream, ofstream, fstream)
* Строковые (istringstream, ostringstream, stringstream)

**3. Какую библиотеку надо подключить при использовании стандартных потоков?**  
**Ответ:** <iostream>

**4. Какую библиотеку надо подключить при использовании файловых потоков?**  
**Ответ:** <fstream>

**5. Какую библиотеку надо подключить при использовании строковых потоков?**  
**Ответ:** <sstream>

**6. Какая операция используется при выводе в форматированный поток?**  
**Ответ:** Оператор << (например, cout << "Hello";).

**7. Какая операция используется при вводе из форматированных потоков?**  
**Ответ:** Оператор >> (например, cin >> x;).

**8. Какие методы используются при выводе в форматированный поток?**  
**Ответ:**

* put() — вывод одного символа
* write() — вывод блока данных
* flush() — сброс буфера

**9. Какие методы используются при вводе из форматированного потока?**  
**Ответ:**

* get() — чтение одного символа
* getline() — чтение строки
* read() — чтение блока данных

**10. Какие режимы для открытия файловых потоков существуют?**  
**Ответ:**

* ios::in — чтение
* ios::out — запись
* ios::app — добавление в конец
* ios::trunc — очистка файла
* ios::binary — бинарный режим

**11. Какой режим используется для добавления записей в файл?**  
**Ответ**: ios::app (например, ofstream file("log.txt", ios::app);).

**12. Какой режим используется в конструкторе**ifstream file("f.txt");?  
**Ответ**: ios::in (только чтение).

**13. Какой режим используется в** конструкторе fstream file("f.txt");?  
**Ответ:** По умолчанию — ios::in | ios::out (чтение и запись).

**14. Какой режим используется в** конструкторе ofstream file("f.txt");?  
**Ответ:** ios::out (запись, с очисткой файла, если он существует).

**15. Как открыть поток в** режиме ios::out | ios::app?

**ofstream file("data.txt", ios::out | ios::app);**

**16. Как открыть поток в режиме**ios::out | ios::trunc**?**

**ofstream file("data.txt", ios::out | ios::trunc);**

**17. Как открыть поток в режиме**ios::out | ios::in | ios::trunc**?**

**fstream file("data.txt", ios::out | ios::in | ios::trunc);**

**18. Как открыть файл для чтения?**

**ifstream file("input.txt");**

**19. Как открыть файл для записи?**

**ofstream file("output.txt");**

**20. Примеры открытия файловых потоков:**

**ifstream in("input.txt"); // Чтение**

**ofstream out("output.txt"); // Запись (очистка файла)**

**fstream io("data.txt", ios::out | ios::app); // Добавление в конец**

**fstream bin("data.bin", ios::binary); // Бинарный режим**

**21. Пример чтения объектов из потока:**

**ifstream file("data.txt");**

**int x;**

**file >> x; // Чтение числа**

**22. Пример записи объектов в поток:**

**ofstream file("output.txt");**

**file << "Hello, world!"; // Запись строки**

**23. Алгоритм удаления записей из файла:**  
**Ответ:**

1. Открыть исходный файл для чтения.
2. Создать временный файл для записи.
3. Копировать записи, кроме удаляемых.
4. Удалить исходный файл, переименовать временный.

**24. Алгоритм добавления записей в файл:**  
**Ответ:**

1. Открыть файл в режиме ios::app.
2. Записать новые данные.
3. Закрыть файл.

**25. Алгоритм изменения записей в файле:**  
**Ответ:**

1. Открыть исходный файл для чтения.
2. Создать временный файл для записи.
3. Копировать записи, изменяя нужные.
4. Удалить исходный файл, переименовать временный.