Министерство образования и науки Российской Федерации

Федеральное государственное образовательное автономное учреждение высшего образования

"Пермский национальный исследовательский политехнический университет"

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №10

Дисциплина: Основы алгоритмизации и программирования

Тема: Сохранение данных в файле с использованием потоков

Вариант 15

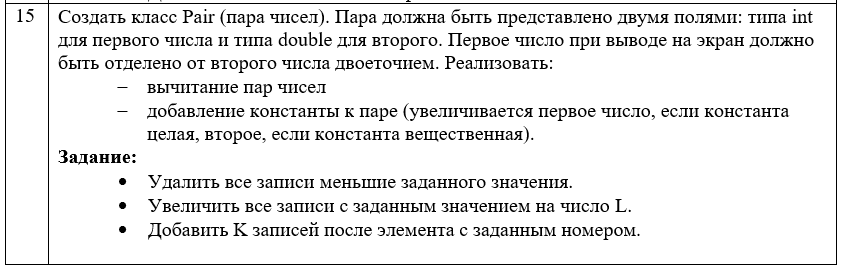
|  |  |
| --- | --- |
| Выполнил работу | |
| Студент группы РИС-22-1б | |
| Кирпичников Илья | |
|  | |
| Проверил работу | |
| Доцент кафедры ИТАС | |
| Полякова О.А. | |
|  | |

Пермь – 2023

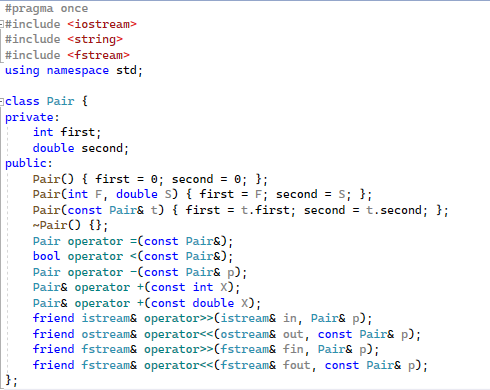
**Постановка задачи**

1. Создать пользовательский класс с минимальной функциональностью.
2. Написать функцию для создания объектов пользовательского класса (ввод исходной информации с клавиатуры) и сохранения их в потоке (файле).
3. Написать функцию для чтения и просмотра объектов из потока.
4. Написать функцию для удаления объектов из потока в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.
5. Написать функцию для добавления объектов в поток в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.
6. Написать функцию для изменения объектов в потоке в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.
7. Для вызова функций в основной программе предусмотреть меню.

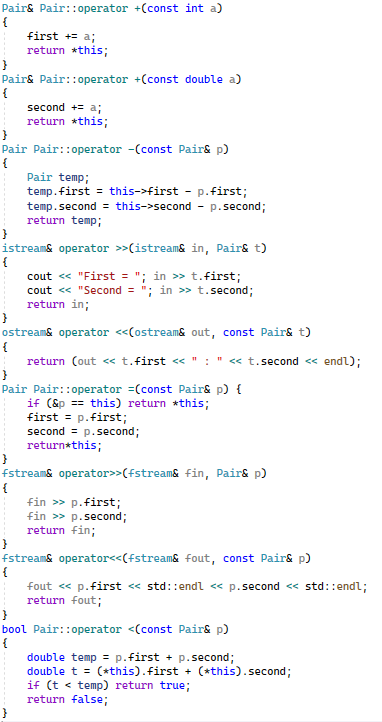
Вариант 15:



# Описание класса Pair и его методов

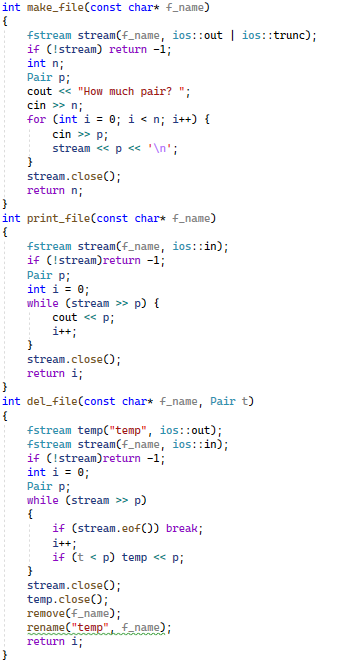


*Приложение 1 – Описание классов*

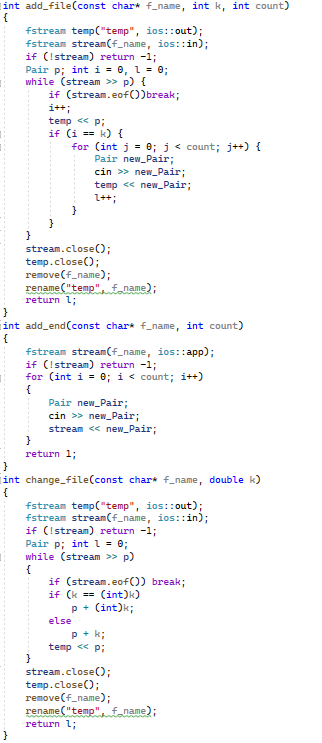
**

*Приложение 2 – Описание методов класса Pair*

# Описание функций для работы с файлом

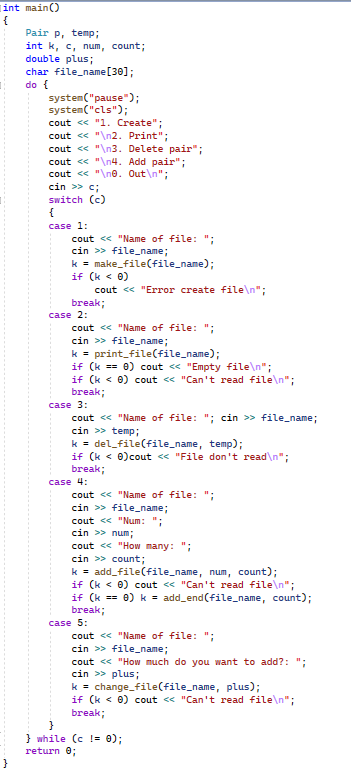
****

*Приложение 3 – Описание функций для работы с файлом*

****

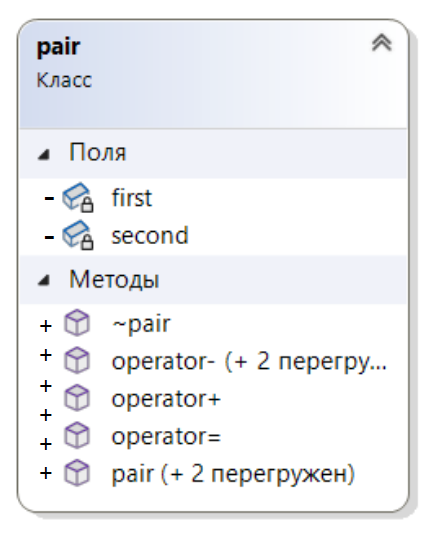
*Приложение 4 – Описание функций для работы с файлом*

**Функция main()**

****

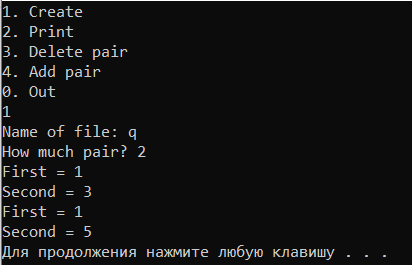
*Приложение 5 – Главная функция*

**UML-Диаграмма**

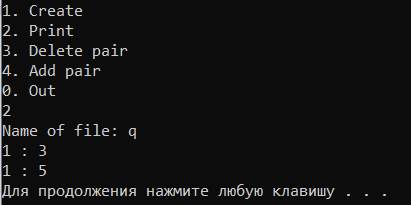
****

*Приложение 6 – UML-Диаграмма*

**Работа программы**



*Приложение 9 – Работа программы*

**

*Приложение 10 – Работа программы*

**Ответы на контрольные вопросы**

1. Поток - это последовательность данных, которые могут быть прочитаны или записаны.

2. Существуют стандартные потоки (cin, cout, cerr), файловые потоки (ifstream, ofstream, fstream) и строковые потоки (istringstream, ostringstream).

3. Для использования стандартных потоков не нужно подключать дополнительную библиотеку.

4. Для использования файловых потоков нужно подключить библиотеку <fstream>.

5. Для использования строковых потоков нужно подключить библиотеку <sstream>.

6. Операция <<.

7. Операция >>.

8. Методы setw(), setprecision(), setfill().

9. Методы getline(), ignore().

10. Режимы для открытия файловых потоков: ios::in (для чтения), ios::out (для записи), ios::app (для добавления записей в конец файла), ios::binary (для работы с бинарными файлами).

11. Режим ios::app.

12. Режим ios::in.

13. Режим ios::in | ios::out.

4. В конструкторе ofstream file("f.txt") используется режим ios::out, который открывает файл для записи.

15. Для открытия потока в режиме ios::out|ios::app нужно использовать конструктор ofstream и передать имя файла в качестве параметра. Например: ofstream file("f.txt", ios::out|ios::app);

16. Для открытия потока в режиме ios::out|ios::trunc нужно использовать конструктор ofstream и передать имя файла в качестве параметра. Например: ofstream file("f.txt", ios::out|ios::trunc);

17. Для открытия потока в режиме ios::out|ios::in|ios::trunk нужно использовать конструктор fstream и передать имя файла в качестве параметра. Например: fstream file("f.txt", ios::out|ios::in|ios::trunc);

18. Для открытия файла для чтения нужно использовать объект ifstream и передать имя файла в качестве параметра. Например: ifstream file("f.txt");

19. Для открытия файла для записи нужно использовать объект ofstream и передать имя файла в качестве параметра. Например: ofstream file("f.txt");

20. Примеры открытия файловых потоков:

- ifstream file("input.txt"); // открыть файл для чтения

- ofstream file("output.txt"); // открыть файл для записи

- fstream file("data.txt", ios::in|ios::out); // открыть файл для чтения и записи

21. Примеры чтения объектов из потока:

- int x;

cin >> x; // чтение целого числа из стандартного потока ввода

- string s;

getline(cin, s); // чтение строки из стандартного потока ввода

22. Примеры записи объектов в поток:

- int x = 10;

cout << x; // запись целого числа в стандартный поток вывода

- string s = "Hello, world!";

ofstream file("output.txt");

file << s; // запись строки в файловый поток

23. Алгоритм удаления записей из файла:

- Открыть файл в режиме ios::in|ios::out.

- Создать временный файл.

- Считать записи из исходного файла, пропуская удаляемые записи.

- Записать считанные записи во временный файл.

- Закрыть оба файла.

- Удалить исходный файл.

- Переименовать временный файл в имя исходного файла.

24. Алгоритм добавления записей в файл:

- Открыть файл в режиме ios::out|ios::app.

- Записать новые записи в конец файла.

- Закрыть файл.

25. Алгоритм изменения записей в файле:

- Открыть файл в режиме ios::in|ios::out.

- Считать записи из файла и изменить нужные записи.

- Записать измененные записи в файл.

- Закрыть файл.