ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа №10

Сохранение данных в файле с использованием потоков

Вариант №15

Выполнил студент группы РИС-23-3Б

Блинов А. Е.

Проверила доцент кафедры ИТАС

О. А. Полякова

2024 г.

**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 добавления объектов в поток в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.

5. Написать функцию для изменения объектов в потоке в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.

6. Для вызова функций в основной программе предусмотреть меню.

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.

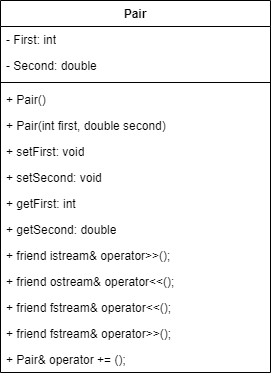
Реализовать:

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

Задание:

* Увеличить все записи с заданным значением на число L.
* Добавить K записей после элемента с заданным номером.

**UML диаграмма**



**Код на языке C++**

#include <iostream>

#include <fstream>

using namespace std;

class Pair {

private:

int First;

double Second;

public:

Pair() {}

Pair(int first, double second) {

setFirst(first);

setSecond(second);

}

void setFirst(int first) { First = first; }

void setSecond(double second) { Second = second; }

int getFirst() { return First; }

double getSecond() { return Second; }

friend istream& operator>>(istream& stream, Pair& pair);

friend ostream& operator<<(ostream& stream, Pair& pair);

friend fstream& operator<<(fstream& stream, Pair& pair);

friend fstream& operator>>(fstream& stream, Pair& pair);

Pair& operator += (double variable) {

int fullPart = variable;

if (variable - fullPart == 0)

First += variable;

else

Second += variable;

return \*this;

}

};

istream& operator>>(istream& stream, Pair& pair) {

cout << "Введите первое число пары: "; stream >> pair.First;

cout << "Введите второе число пары: "; stream >> pair.Second;

return stream;

}

ostream& operator<<(ostream& stream, Pair& pair) { return { stream << pair.First << " : " << pair.Second }; }

fstream& operator<<(fstream& stream, Pair& pair) {

stream << pair.First << endl << pair.Second << endl;

return stream;

}

fstream& operator>>(fstream& stream, Pair& pair) {

stream >> pair.First;

stream >> pair.Second;

return stream;

}

void WriteToFile(int length) {

fstream F1("F1.txt", ios::out | ios::trunc);

if (!F1) {

cout << "Не удалось открыть файл.";

return;

}

for (int i = 0; i < length; i++) {

Pair pair;

cin >> pair;

F1 << pair;

cout << endl;

}

F1.close();

}

void printAFile() {

fstream F1("F1.txt", ios::in);

if (!F1) {

cout << "Не удалось открыть файл.";

return;

}

Pair pair;

while (F1 >> pair) { cout << pair << endl; }

}

void addNumToPair(double num) {

fstream F1("F1.txt", ios::in);

fstream Temp("Temp.txt", ios::out | ios::trunc);

Pair buffer;

while (F1 >> buffer) {

buffer += num;

Temp << buffer << endl;

}

F1.close();

Temp.close();

remove("F1.txt");

rename("Temp.txt", "F1.txt");

}

void addKElems(int position, int k) {

if (position <= 0 || k <= 0) {

cout << "Неправильная позиция: она должна быть >= 1" << endl;

return;

}

fstream F1("F1.txt", ios::in);

if (!F1) {

cout << "Не удалось открыть файл.";

return;

}

fstream Temp("Temp.txt", ios::out | ios::trunc);

Pair buffer;

int counter = 0;

while (F1 >> buffer) {

counter++;

Temp << buffer << endl;

if (counter == position) {

for (int i = 0; i < k; i++) {

Pair newPair;

cin >> newPair;

Temp << newPair << endl;

}

}

}

if (counter < position) { cout << "Введена позиция, которой нет в файле." << endl; }

F1.close();

Temp.close();

remove("F1.txt");

rename("Temp.txt", "F1.txt");

}

int main() {

system("chcp 1251 > Null");

int lenght;

do { cout << "Введите кол-во пар в файле: "; cin >> lenght; } while (lenght <= 0);

WriteToFile(lenght);

bool shouldStop = false;

while (!shouldStop) {

cout << endl << endl;

cout << "1 - Пересоздать файл\n2 - Добавить L ко всем элементам\n3 - Добавить К пар\n4 - Распечатать файл\n5 - Закончить работу программы";

int choice;

cout << "\n\nВаш выбор: "; cin >> choice;

cout << endl;

Pair comparePair;

switch (choice) {

case 1:

do {

cout << "Введите кол-во пар: "; cin >> lenght;

} while (lenght <= 0);

WriteToFile(lenght);

break;

case 2:

double L;

cout << "Введите число, на которое увеличиваем: "; cin >> L;

cout << endl;

addNumToPair(L);

break;

case 3:

int k, position;

cout << "Введите, на какую позицию добавляем: "; cin >> position;

cout << "Введите, сколько пар добавляем: "; cin >> k;

cout << endl;

addKElems(position, k);

break;

case 4:

printAFile();

break;

case 5:

shouldStop = true;

break;

default:

cout << "\nТакой команды нет." << endl;

break;

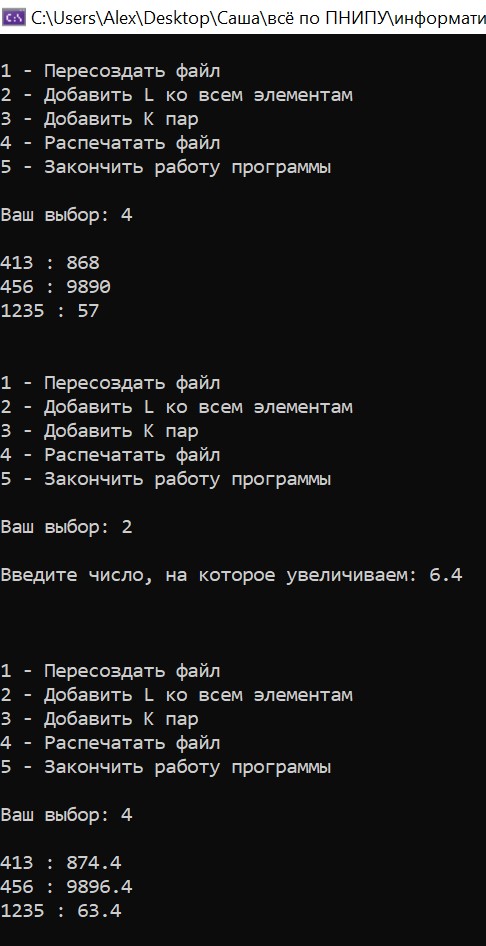
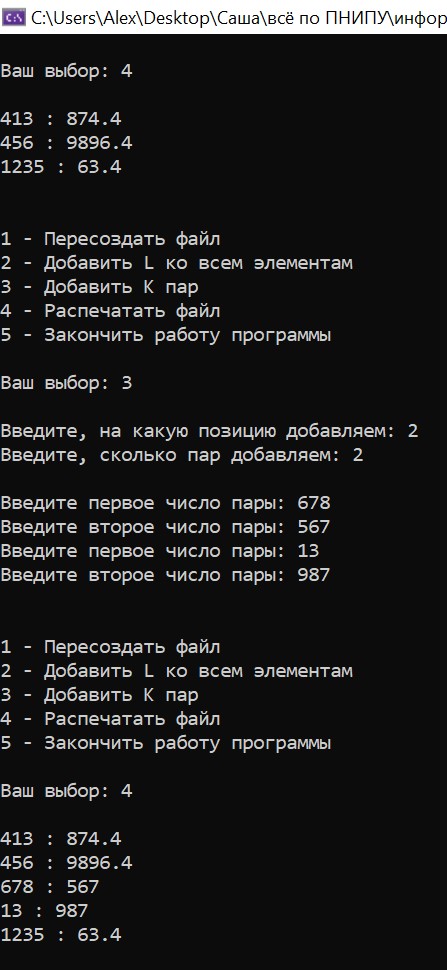
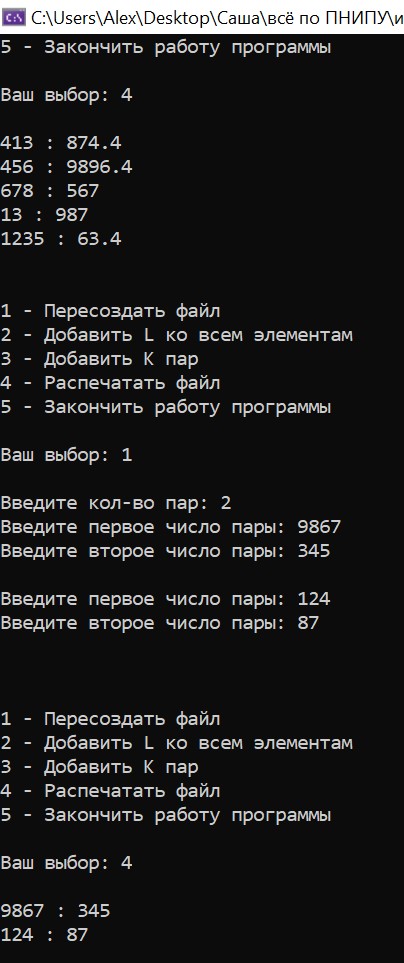
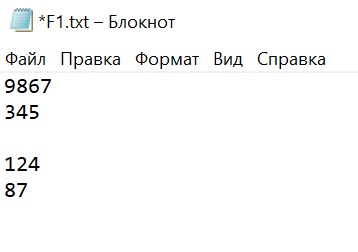
}

}

return 0;

}

**Результаты работы программы**

**Выводы**

Программа работает успешно, записывает пары чисел в файл без ошибок.

GitHub:

**Контрольные вопросы**

1. Что такое поток?

Поток – это абстракция, которая обеспечивает доступ к последовательности данных

1. Какие типы потоков существуют?

Три основных: потоки ввода, потоки вывода, потоки ошибок

1. Какую библиотеку надо подключить при использовании стандартных потоков?

**istream**, **ostream** (библиотека **iostream**) – общие потоковые классы, которые могут быть связаны с любым буферным объектом

1. Какую библиотеку надо подключить при использовании файловых потоков?

**ifstream**, **ofstream** (библиотека **fstream**) – предназначены для работы  файлами

1. Какую библиотеку надо подключить при использовании строковых потоков?

**istringstream**, **ostringstream** – потоковые классы для объектов строк.

1. Какая операция используется при выводе в форматированный поток?

Операция «<<»

1. Какая операция используется при вводе из форматированных потоков?

Операция «>>»

1. Какие методы используются при выводе в форматированный поток?

Обычно используются cout для вывода в консоль и ofstream для вывода в файл

1. Какие методы используются при вводе из форматированного потока?

Обычно используют cin для ввода из консоли и ifstream для чтения из файла

1. Какие режимы для открытия файловых потоков существуют?

* ios::in – открытие файла для чтения
* ios::out – открытие файла для записи
* ios::trunc – обрезка файла до нулевой длины, если он существует

1. Какой режим используется для добавления записей в файл?

Используется режим ios::app

1. Какой режим (комбинация режимов) используется в конструкторе ifstream file(“f.txt”)?

Используется комбинация режимов std::ios::in, то есть файл открывается для чтения

1. Какой режим (комбинация режимов) используется в конструкторе fstream file(“f.txt”)?

Так как явно не указан ни один режим, по умолчанию файл открывается с комбинацией режим std::ios::in или std:: ios::out

1. Какой режим (комбинация режимов) используется в конструкторе ofstream file(“f.txt”)?

Используется комбинация режимов std::ios::out, что означает открытие файла только для записи