МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра информационные технологии и автоматизированные системы

**Дисциплина Информатика**

**Потоки**

Выполнил студент ИВТ-22-2б:

Мифтахов Марат Ринатович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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.

**Задание**

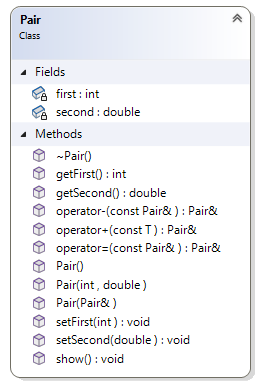
Создать класс Pair (пара чисел). Пара должна быть представлено двумя полями: типа int для первого числа и типа double для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием.

Реализовать:

1. Вычитание пар чисел
2. добавление константы к паре (увеличивается первое число, если константа целая, второе, если константа вещественная).
3. Задание:
4. Удалить все записи меньшие заданного значения.
5. Увеличить все записи с заданным значением на число L.
6. Добавить K записей после элемента с заданным номером.

**Диаграмма классов**

Ниже представлена UML-диаграмма классов.



**Программный код**

Файл Pair.h:

#pragma **once**

#include <iostream>

#include <fstream>

class Pair {

private:

int first;

double second;

public:

Pair();

Pair(int, double);

Pair(Pair&);

~Pair() {};

int getFirst();

void setFirst(int);

double getSecond();

void setSecond(double);

void show();

template <typename T>

Pair& operator+(const T);

Pair& operator-(const Pair&);

Pair& operator=(const Pair&);

friend std::istream& operator>>(std::istream&, Pair&);

friend std::ostream& operator<<(std::ostream&,const Pair&);

friend std::fstream& operator<<(std::fstream&, const Pair&);

friend std::fstream& operator>>(std::fstream&, Pair&);

};

template <typename T>

Pair& Pair::operator+(const T p) {

if (p != (int)p) {

second += p;

}

else {

first += p;

}

return \*this;

}

Файл fileWork.h:

#pragma **once**

#include "Pair.h"

using namespace std;

int file\_makeFile\_userInput(const char\* filename) {

fstream outStream(filename, ios::out | ios::trunc);

if (!outStream) return -1;

int numberOfPairs;

cout << "number of pairs?" << endl;

cin >> numberOfPairs;

Pair pair;

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

cout << "--------" << endl;

cin >> pair;

cout << "--------" << endl;

outStream << pair;

}

outStream.close();

return numberOfPairs;

}

void file\_print(const char\* filename) {

fstream inStream(filename, ios::in);

if (!inStream) {

cout << "ошибка чтения" << endl;

return;

}

Pair pair;

while (inStream >> pair) {

cout << pair << '\n';

}

inStream.close();

}

int file\_deleteRecordsUnderValue(const char\* filename, int limitValue) {

fstream fromStream(filename, ios::in);

fstream inStream("tmpFile", ios::out | ios::trunc);

int numberOfPairs = 0;

Pair p;

while (fromStream >> p) {

if (p.getFirst() > limitValue && p.getSecond() > limitValue && !fromStream.eof()) {

inStream << p;

numberOfPairs++;

}

}

fromStream.close();

inStream.close();

remove(filename);

rename("tmpFile", filename);

return numberOfPairs;

}

template <typename type, typename type1>

void file\_addRecordsWithValue(const char\* filename, type currentValue, type1 additionalValue) {

fstream fromStream(filename, ios::in);

fstream inStream("tmpFile", ios::out | ios::trunc);

Pair p;

while (fromStream >> p) {

if ((p.getFirst() == currentValue || p.getSecond() == currentValue) && !fromStream.eof()) {

p + additionalValue;

}

inStream << p;

}

fromStream.close();

inStream.close();

remove(filename);

rename("tmpFile", filename);

}

int file\_addNumberOfRecordsAfterPair(const char\* filename, int pairPosition, int numberOfElements) {

fstream fromStream(filename, ios::in);

fstream inStream("tmpFile", ios::out | ios::trunc);

int numberOfPairs = 0;

Pair p;

for (int i = 0; i < pairPosition - 1 && !fromStream.eof(); i++) {

fromStream >> p;

if (!fromStream.eof()) inStream << p;

numberOfPairs++;

}

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

cout << "--------" << endl;

cin >> p;

cout << "--------" << endl;

inStream << p;

numberOfPairs++;

}

while (!fromStream.eof() && fromStream >> p) {

if (!fromStream.eof()) {

inStream << p;

}

numberOfPairs++;

}

fromStream.close();

inStream.close();

remove(filename);

rename("tmpFile", filename);

return numberOfPairs;

**}**

Файл Pair.cpp:

#include **"Pair.h"**

#include <typeinfo>

Pair::Pair() {

first = 0;

second = 0.0;

}

Pair::Pair(int a, double b) {

first = a;

second = b;

}

Pair::Pair(Pair& t) {

first = t.first;

second = t.second;

}

int Pair::getFirst() {

return first;

}

void Pair::setFirst(int tf) {

first = tf;

}

double Pair::getSecond() {

return second;

}

void Pair::setSecond(double tf) {

second = tf;

}

Pair& Pair::operator=(const Pair& ptr) {

if (&ptr == this) return \*this;

first = ptr.first;

second = ptr.second;

return \*this;

}

void Pair::show() {

std::cout << first << ":" << second << std::endl;

}

Pair& Pair::operator-(const Pair& a) {

first -= a.first;

second -= a.second;

return \*this;

}

std::istream& operator>>(std::istream& in, Pair& p) {

std::cout << "First? " << std::endl; in >> p.first;

std::cout << "Second?" << std::endl; in >> p.second;

return in;

}

std::ostream& operator<<(std::ostream& out, const Pair& p) {

return out << p.first << ":" << p.second;

}

std::fstream& operator>>(std::fstream& fin, Pair& p) {

fin >> p.first;

fin >> p.second;

return fin;

}

std::fstream& operator<<(std::fstream& fout, const Pair& p) {

fout << p.first << '\n';

fout << p.second << '\n';

return fout;

**}**

Файл main.cpp:

#include **"Pair.h"**

#include "fileWork.h"

using namespace std;

int main() {

const char filename[] = "name";

int numberOfPairs = 0;

int c;

while (true) {

system("cls");

cout << "1. create file" << endl;

cout << "2. print file" << endl;

cout << "3. delete records from file under limit value" << endl;

cout << "4. increase records with current value" << endl;

cout << "5. add k elements on the current position" << endl;

cout << "enter c: "; cin >> c;

switch (c) {

case 1:

numberOfPairs = file\_makeFile\_userInput(filename);

break;

case 2:

if (numberOfPairs > 0) file\_print(filename);

else { cout << "empty file" << endl; }

break;

case 3:

double limitValue;

cout << "limit value?: "; cin >> limitValue;

numberOfPairs = file\_deleteRecordsUnderValue(filename, limitValue);

break;

case 4:

double currentValue;

double additionalValue;

cout << "current value?: "; cin >> currentValue;

cout << "current value?: "; cin >> additionalValue;

file\_addRecordsWithValue(filename, currentValue, additionalValue);

break;

case 5:

int k, pos;

cout << "k? "; cin >> k;

cout << "pos? "; cin >> pos;

numberOfPairs = file\_addNumberOfRecordsAfterPair(filename, pos, k);

break;

default:

cout << numberOfPairs;

}

system("pause");

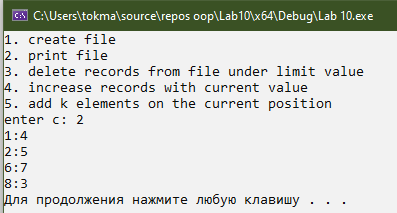
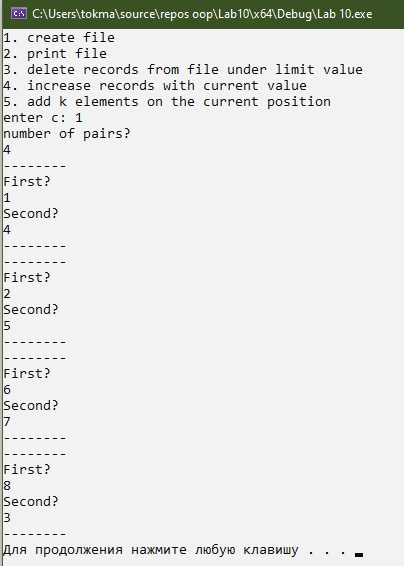
}

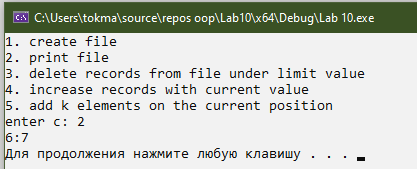
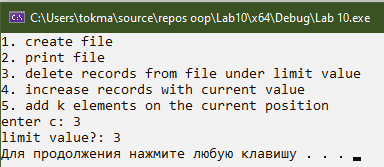
return 0;

**}**

**Вывод программы**

Ниже представлен вывод программы на консоль.

****

****

**Контрольные вопросы**

1. Что такое поток?

Ответ: Поток - последовательность байтов, не зависящаяот конкретного устройства, с которым производится обмен

1. Какие типы потоков существуют?

Ответ: Стандартные, строковые и файловые. Все они делятся на однонаправленные и двунаправленные.

1. Какую библиотеку надо подключить при использовании стандартных потоков?

Ответ: <iostream>

1. Какую библиотеку надо подключить при использовании файловых потоков?

Ответ: <fstream>

1. Какую библиотеку надо подключить при использовании строковых потоков?

Ответ: <sstream>

1. Какая операция используется при выводе в форматированный поток?

Ответ: operator<<

1. Какая операция используется при вводе из форматированных потоков?

Ответ: operator>>

1. Какие методы используются при выводе в форматированный поток?

Ответ:



1. Какие методы используется при вводе из форматированного потока?

Ответ:





1. Какие режимы для открытия файловых потоков существуют?

Ответ:



1. Какой режим используется для добавления записей в файл?

Ответ: режим app

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

Ответ: in

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

Ответ: in | out

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

Ответ: out

1. Каким образом открывается поток в режиме ios::out|ios::app?

Ответ: поток открывается на запись в конец файла.

1. Каким образом открывается поток в режиме ios::out [ios::trunc?

Ответ: поток открывается для записи, предварительно очистив файл

1. Каким образом открывается поток в режиме ios::out ios::in/ios::trunk?

Ответ: на ввод/ вывод данных с предварительной очисткой файла

1. Каким образом можно открыть файл для чтения?

Ответ: явно методом open и неявно при вызове конструктора

1. Каким образом можно открыть файл для записи?

Ответ: явно методом open и неявно при вызове конструктора

1. Привести примеры открытия файловых потоков в различных режимах.

Ответ: ofstream output(“file”); ifstream input(“file”); ofstream base(“file”, ios::trunc);

1. Привести примеры чтения объектов из потока.

Ответ: считывание строки getline(), считывание чисел, символов

«поток» » something,

1. Привести примеры записи объектов в поток.

Ответ: запись строки, чисел, символов

«поток» << something,

1. Сформулировать алгоритм удаления записей из файла.

Ответ:

1. Открыть 2 файла.
2. Проверить каждый элемент в файле 1 на соответствие.
3. Соответствующие элементы записать в файл 2.
4. Закрыть файлы.
5. Переименовать файл 2 в файл 1.
6. Сформулировать алгоритм добавления записей в файл.

Ответ:

1. Открыть 2 файла.
2. Проверить каждый элемент в файле 1 на соответствие.
3. Соответствующие элементы записать в файл 2.
4. За ними записать необходимые элементы
5. Закрыть файлы.
6. Переименовать файл 2 в файл 1.
7. Сформулировать алгоритм изменения записей в файле.

Ответ:

1. Открыть 2 файла.
2. Проверить каждый элемент в файле 1 на соответствие.
3. Соответствующие элементы записать в файл 2 c измененными данными.
4. Закрыть файлы.
5. Переименовать файл 2 в файл 1.