МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа № 10:**

"Сохранение данных в файле с использованием потоков"

Выполнила: студентка гр.РИС-23-3Б

Мокроусова Ангелина Андреевна

Проверила: доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2024

**Постановка задачи:**

Общая часть:

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 просморта 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 из потока в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций
6. Написать функцию для изменения объектов из потока в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций
7. Для вызова функции в основной программе предусмотреть меню

Задание варианта:

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:

* Операции сравнения (<, >)
* Операция --, которая работает следующим образом: если форма операции префиксная, то уменьшается первое число, если форма операции постфиксная, то уменьшается второе число

Задание:

* Удалить все записи из интервала от k1 до k2, где k1 и k2 переменные типа Pair.
* Увеличть все записи с заданным значением в два раза.
* Добавить К записей в начало файла.

**Анализ задачи:**

Входные данные:

* Пара чисел, вводимая пользователем.
* Текстовый файл, содержащий пары чисел.

Выходные данные:

* Текстовый файл, содержащий обновленные пары чисел.
* Сообщения, информирующие пользователя о ходе программы.

Алгоритм:

* Создать класс Pair для представления пары чисел.
* Написать функции для сохранения и чтения пар из текстового файла.
* Написать функции для удаления и добавления пар в текстовый файл.
* Написать функции для изменения значений пар в текстовом файле.
* Написать программу с меню, позволяющую пользователю выбирать операции для выполнения.

**Код на С++ с комментариями:**

#include <iostream> // Подключаем необходимые заголовочные файлы

#include <fstream>

#include <vector>

using namespace std;

// Класс Pair (пара)

class Pair {

private:

int num1;

double num2;

public:

// Конструктор с параметрами по умолчанию

Pair(int n1 = 0, double n2 = 0.0) : num1(n1), num2(n2) {}

// Оператор сравнения для сортировки

bool operator<(const Pair& other) const {

return (num1 < other.num1) || (num1 == other.num1 && num2 < other.num2);

}

// Оператор сравнения для сортировки

bool operator>(const Pair& other) const {

return (num1 > other.num1) || (num1 == other.num1 && num2 > other.num2);

}

// Уменьшение (постфикс)

Pair operator--(int) {

Pair temp = \*this;

num2--;

return temp;

}

// Уменьшение (префикс)

Pair& operator--() {

num1--;

return \*this;

}

// Перегрузка оператора вывода

friend ostream& operator<<(ostream& out, const Pair& pair) {

out << pair.num1 << " : " << pair.num2;

return out;

}

// Перегрузка оператора ввода

friend istream& operator>>(istream& in, Pair& pair) {

in >> pair.num1 >> pair.num2;

return in;

}

// Увеличить на два

void increaseByTwo() {

num1 \*= 2;

num2 \*= 2;

}

// Ввод пары с клавиатуры

static Pair inputPairFromUser() {

Pair pair;

cout << "Введите пару (число вещественное число): ";

cin >> pair;

return pair;

}

};

// Сохранение пар в файл

void savePairsToFile(const string& filename, const vector<Pair>& pairs) {

ofstream file(filename, ios::out);

if (file.is\_open()) {

for (const Pair& p : pairs) {

file << p << endl;

}

file.close();

}

else {

cout << "Не удалось открыть файл для записи." << endl;

}

}

// Чтение пар из файла

vector<Pair> readPairsFromFile(const string& filename) {

vector<Pair> pairs;

Pair pair;

ifstream file(filename, ios::in);

if (file.is\_open()) {

while (file >> pair) {

pairs.push\_back(pair);

}

file.close();

}

else {

cout << "Не удалось открыть файл для чтения." << endl;

}

return pairs;

}

// Удаление пар из диапазона

void removePairsInRange(vector<Pair>& pairs, const Pair& k1, const Pair& k2) {

pairs.erase(remove\_if(pairs.begin(), pairs.end(),

[k1, k2](const Pair& p) { return p > k1 && p < k2; }), pairs.end());

}

// Добавление K пар в начало файла

void addKPairsToBeginningOfFile(const string& filename, int k) {

vector<Pair> pairs; // Инициализирован пустым вектором

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

Pair pair = Pair::inputPairFromUser();

pairs.insert(pairs.begin(), pair);

}

savePairsToFile(filename, pairs);

}

int main() {

setlocale(LC\_ALL, "ru");

vector<Pair> pairs;

string filename = "pairs.txt";

string option;

Pair k1, k2, value;

int k;

do {

cout << "Меню:" << std::endl;

cout << "1. Сохранить пары в файл" << std::endl;

cout << "2. Прочитать пары из файла" << std::endl;

cout << "3. Удалить пары в диапазоне" << std::endl;

cout << "4. Удвоить пары со значением" << std::endl;

cout << "5. Добавить K пар в начало файла" << std::endl;

cout << "6. Выход" << std::endl;

cout << "Выберите пункт меню: ";

cin >> option;

if (option == "1") {

pairs.push\_back(Pair::inputPairFromUser());

savePairsToFile(filename, pairs);

}

else if (option == "2") {

pairs = readPairsFromFile(filename);

for (const Pair& p : pairs) {

cout << p << endl;

}

}

else if (option == "3") {

cout << "Введите пару k1: ";

cin >> k1;

cout << "Введите пару k2: ";

cin >> k2;

removePairsInRange(pairs, k1, k2);

savePairsToFile(filename, pairs);

}

/\*else if (option == "4") {

cout << "Введите пару value: ";

cin >> value;

addKPairsToBeginningOfFile(pairs, value);

savePairsToFile(filename, pairs);

}\*/

else if (option == "5") {

cout << "Введите количество пар для добавления: ";

cin >> k;

addKPairsToBeginningOfFile(filename, k);

}

else if (option == "6") {

cout << "Выход из программы." << endl;

break;

}

else {

cout << "Неверный пункт меню. Повторите попытку." << endl;

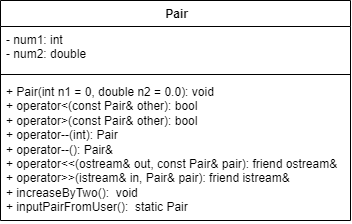
}

} while (true);

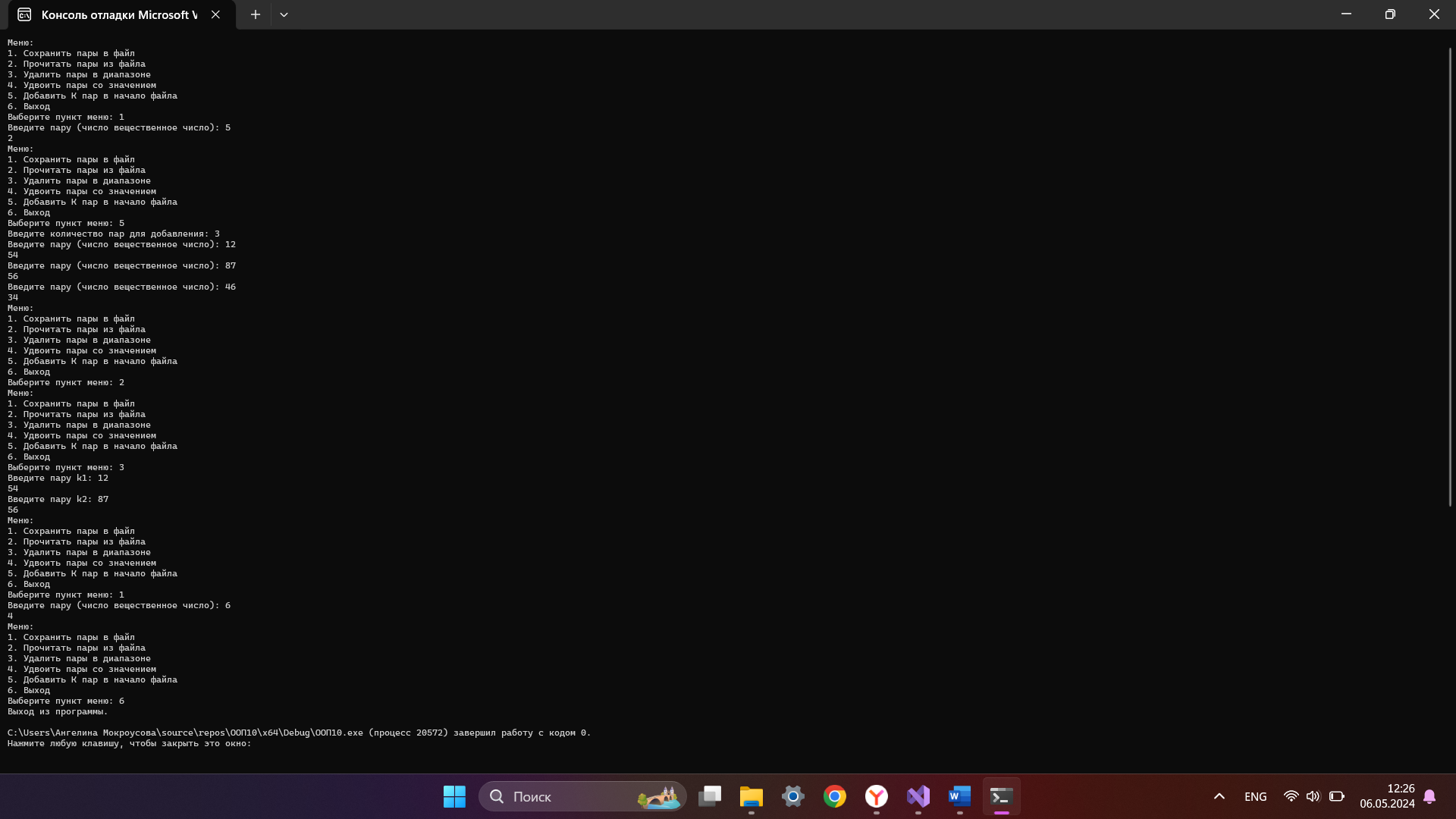
return 0;

}

**UML диаграмма:**

****

**Скриншот работы программы:**



**Вывод:**

Мною была аписана программа, которая успешно выполняет поставленную перед ней задачу. В процессе написания был повторен и закреплен на практике пройденный ранее материал.

**Ответы на вопросы:**

1. Что такое поток?

Поток - это абстрактный канал связи, используемый для ввода или вывода данных.

1. Какие типы потоков существуют?

Стандартные потоки (ввод, вывод, ошибка)

Файловые потоки

Строковые потоки

1. Какую библиотеку надо подключить при использовании стандартных потоков?

Iostream

1. Какую библиотеку надо подключить при использовании файловых потоков?

fstream

1. Какую библиотеку надо подключить при использовании строковых потоков?

sstream

1. Какая операция используется при выводе в форматированный поток?

<<

1. Какая операция используется при вводе из форматированных потоков?

>>

1. Какие методы используются при выводе в форматированный поток?

Методы форматирования, такие как setw(), setprecision(), setfill() и др.

1. Какие методы используется при вводе из форматированного потока?

ignore(), getline(), peek() и др.

1. Какие режимы для открытия файловых потоков существуют?

ios::in - открытие для чтения

ios::out - открытие для записи

ios::app - открытие для добавления

ios::trunc - открытие с отсечением содержимого

ios::ate - открытие с размещением указателя ввода-вывода в конце файла

1. Какой режим используется для добавления записей в файл?

ios::app

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

ios::in

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

ios::in | ios::out

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

ios::out

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

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

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

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

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

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

1. Каким образом можно открыть файл для чтения?

ifstream file("f.txt");

1. Каким образом можно открыть файл для записи?

ofstream file("f.txt");

1. Привести примеры открытия файловых потоков в различных режимах.

ifstream file("f.txt", ios::in); - открыть файл для чтения

ofstream file("f.txt", ios::out | ios::app); - открыть файл для добавления

fstream file("f.txt", ios::out | ios::in | ios::trunc); - открыть файл для перезаписи

1. Привести примеры чтения объектов из потока.

file >> variable;

1. Привести примеры записи объектов в поток.

file << variable;

1. Сформулировать алгоритм удаления записей из файла.

Прочитать все записи из файла во временный список.

Удалить нужные записи из временного списка.

Записать временный список обратно в файл, переписав его.

1. Сформулировать алгоритм добавления записей в файл.

Открыть файл в режиме добавления.

Записать новые записи в файл.

Закрыть файл.

1. Сформулировать алгоритм изменения записей в файле.

Прочитать все записи из файла во временный список.

Найти нужную запись во временном списке.

Изменить запись.

Записать временный список обратно в файл, переписав его.

**Скриншот с GitHab:**

