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

Асташин Д. А.

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.

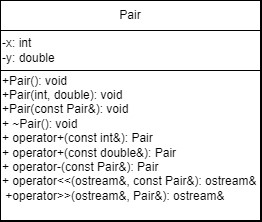
Реализовать:

* 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 программы**

#pragma once

#include <iostream>

#include <fstream>

using namespace std;

class Pair

{

private:

int first;

double second;

public:

Pair() {

first = 0;

second = 0;

}

Pair(int first, double second) {

this->first = first;

this->second = second;

}

Pair(const Pair& p) {

first = p.first;

second = p.second;

}

Pair operator = (const Pair& p) {

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

first = p.first;

second = p.second;

return \*this;

}

~Pair() {}

friend ostream& operator<<(ostream& stream, const Pair& p) {

stream << '(' << p.first << ":" << p.second << ')' << endl;

return stream;

}

friend istream& operator>>(istream& stream, Pair& p) {

cout << "Введите первое число: ";

stream >> p.first;

cout << "Введите второе число: ";

stream >> p.second;

return stream;

}

Pair operator+(const int& x) {

return Pair(this->first + x, this->second);

}

Pair operator+(const double& y) {

return Pair(this->first, this->second + y);

}

Pair operator-(const Pair& p) {

return Pair(this->first - p.first, this->second - p.second);

}

//Функции для сохранения файлов в классе

friend fstream& operator>>(fstream& fin, Pair& p)

{

fin >> p.first;

fin >> p.second;

return fin;

}

friend fstream& operator<<(fstream& fout, const Pair& p)

{

fout << p.first << "\n" << p.second << "\n";

return fout;

}

};

// Работа файлов

int make\_file(const char\* f\_name)

{

fstream stream(f\_name, ios::out | ios::trunc);

if (!stream) return - 1;

int n;

Pair p;

cout << "Введите кол-во элементов для добавления в файл: ";

cin >> n;

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

cin >> p;

stream << p << "\n";

}

stream.close();

return n;

}

// Чтение объектов из потока

int print\_file(const char\* f\_name)

{

fstream stream(f\_name, ios::in);

if (!stream) return -1;

Pair p; int i = 0;

while (stream >> p)

{

cout << p << "\n";

i++;

}

stream.close();

return i;

}

//Удаление объектов из потока

int del\_file(const char\* f\_name, int k)

{

fstream temp("temp", ios::out);

fstream stream(f\_name, ios::in);

if (!stream) return -1;

int i = 0; Pair p;

while (stream >> p)

{

if (stream.eof()) break;

i++;

if (i != k) temp << p;

}

stream.close(); temp.close();

remove(f\_name);

rename("temp", f\_name);

return i;

}

int add\_file(const char\* f\_name, int k, Pair pp)

{

fstream temp("temp", ios::out);

fstream stream(f\_name, ios::in);

if (!stream)return -1;

Pair p; int i = 0, l = 0;

while (stream >> p)

{

if (stream.eof()) break;

i++;

if (i == k)

{

temp << pp;

l++;

}

temp << p;

}

stream.close(); temp.close();

remove(f\_name);

rename("temp", f\_name);

return l; // количество добавлений

}

int add\_end(const char\* f\_name, Pair p)

{

fstream stream(f\_name, ios::app);

if (!stream) return -1;

stream << p;

return 1;

}

int change\_file(const char\* f\_name, int k, Pair pp)

{

fstream temp("temp", ios::out);

fstream stream(f\_name, ios::in);

if (!stream)return -1;

Pair p; int i = 0, l = 0;

char x;

while (stream >> p)

{

if (stream.eof()) break;

i++;

if (i == k)

{

cout << p << " - is changing... Continue[y/n]?\n";

cin >> x;

if (x == 'n' || x == 'N') break;

temp << pp;

l++;

}

else temp << p;

}

stream.close(); temp.close();

remove(f\_name);

rename("temp", f\_name);

return l;

}

int add\_L(const char\* f\_name, int L)

{

fstream temp("temp", ios::out);

fstream stream(f\_name, ios::in);

if (!stream)return -1;

Pair p;

while (stream >> p)

{

p = p + L;

temp << p;

}

stream.close(); temp.close();

remove(f\_name);

rename("temp", f\_name);

return 1;

}

int main() {

system("chcp 1251 > Null");

Pair p, p1, p2;

int k, c, L;

char file\_name[30];

do

{

cout << "\n1. Make file";

cout << "\n2. Print file";

cout << "\n3. Delete record from file";

cout << "\n4. Add record to file";

cout << "\n5. Change record in file";

cout << "\n6. Add L to pairs";

cout << "\n0. Exit\n";

cin >> c;

switch (c)

{

case 1: cout << "file name?"; cin >> file\_name;

k = make\_file(file\_name);

if (k < 0) cout << "Can`t make file;";

break;

case 2: cout << "file name?"; cin >> file\_name;

k = print\_file(file\_name);

if (k == 0)cout << "Empty file\n";

if (k < 0)cout << "Can`t read file\n";

break;

case 3: cout << "file name?"; cin >> file\_name;

int nom; cout << "nom?"; cin >> nom;

k = del\_file(file\_name, nom);

if (k < 0) cout << "Can`t read file";

break;

case 4: cout << "file name?"; cin >> file\_name;

cout << "nom?"; cin >> nom;

cout << "New Pair: ";

cin >> p1;

k = add\_file(file\_name, nom, p1);

if (k < 0) cout << "Can`t read file";

if (k == 0) k = add\_end(file\_name, p1);

break;

case 5: cout << "file name?"; cin >> file\_name;

cout << "nom?"; cin >> nom;

cout << "New pair: ";

cin >> p2;

k = change\_file(file\_name, nom, p2);

if (k < 0)cout << "\nCan`t read file";

if (k == 0) cout << "\nNot such record";

break;

case 6: cout << "file name?"; cin >> file\_name;

cout << "L?"; cin >> L;

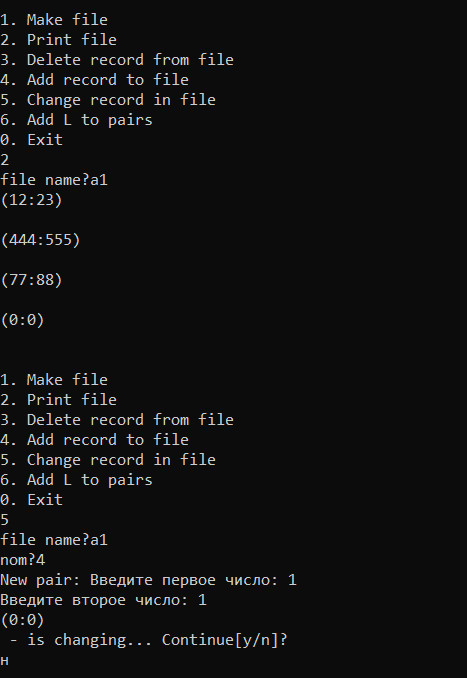
k = add\_L(file\_name, L);

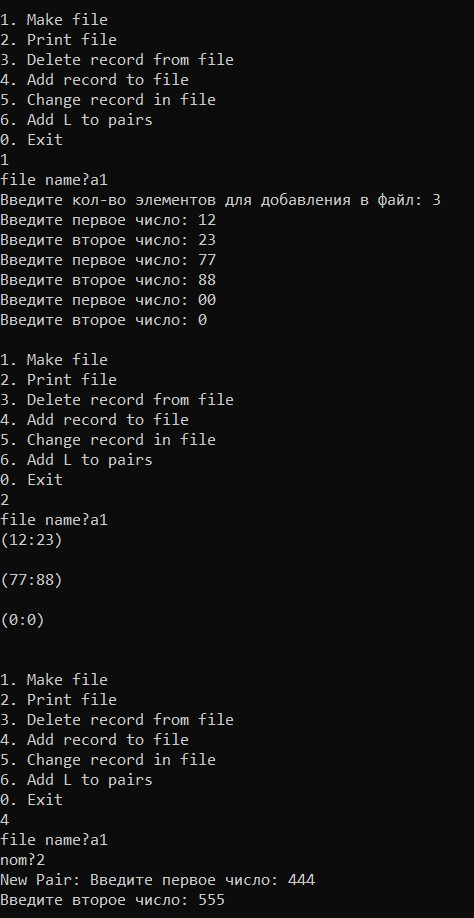
break;

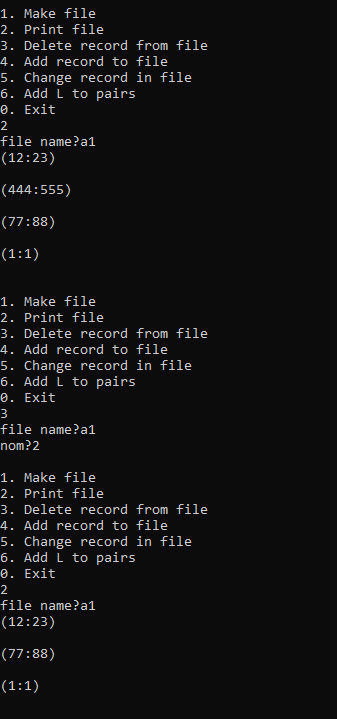
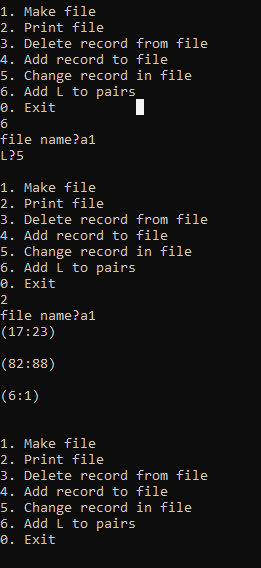
}

} while (c != 0);

}

****  **Результаты работы**

****

**** ****

**Выводы**

Программа работает успешно

GitHub: <https://github.com/DimetriusAsti/OOP10.git>

**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