Министерство образования РФ

Пермский национальный исследовательский политехнический университет

Кафедра ИТАС

ЛАБОРАТОРНАЯ РАБОТА № 10

ОСНОВЫ АЛГОРИТМИЗАЦИИ ЗА II СЕМЕСТР

Выполнил студент:

Тимолянов Григорий Констнатинович

Группа РИС-21-1бз

Проверила:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

ПЕРМЬ 2024

# Лабораторная работа №10 "Сохранение данных в файле с использованием потоков"

**Вариант №15**

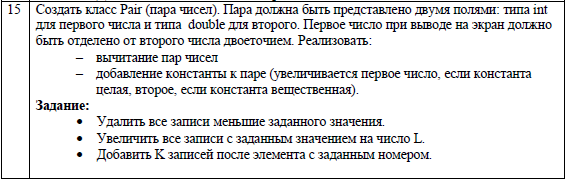
**Цель:**

1) Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.

2) Разработка программы, в которой данные сохраняются в файле, корректируются и выводятся из файла на печать. Работа с файлом осуществляется с использованием потоковых классов.

## **Задача:**

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



**Решение:**

Файл Потоки.cpp

#include "Pair.h"

#include <iostream>

#include <fstream>

#include <string>

#include "File\_work.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

Pair p, p1, p2;

int k, c, num1, num2;

char fileName[30];

do {

cout << "\n1. Создать файл"

"\n2. Вывести файл в консоль"

"\n3. Очистить файл"

"\n4. Изменить запись в файле"

"\n5. Добавить запись в файл"

"\n0. Выйти\n\n"

"\n6. Удалить все записи меньше заданного значения"

"\n7. Увеличить все записи с заданным значением на число L"

"\n8. Добавить K записей после элемента с заданным номером\n\n";

cin >> c;

switch (c) {

case 1:

cout << "\nНаименование файла (без пробелов, через .txt) ";

cin >> fileName;

k = MakeFile(fileName);

if (k < 0) {

cout << "Невозможно создать файл!";

}

break;

case 2:

cout << "\nНаименование файла (без пробелов, через .txt) ";

cin >> fileName;

PrintFile(fileName);

break;

case 3:

cout << "\nНаименование файла (без пробелов, через .txt) ";

cin >> fileName;

k = ClearFile(fileName, 0);

if (k < 0) {

cout << "Невозможно прочитать файл!";

}

break;

case 4:

cout << "\nНаименование файла (без пробелов, через .txt) ";

cin >> fileName;

cin >> p1;

k = AddFile(fileName, p1);

if (k < 0) {

cout << "\nНевозможно прочитать файл!";

}

k = AddNote(fileName, p1);

break;

case 5:

cout << "\nНаименование файла (без пробелов, через .txt) ";

cin >> fileName;

cin >> p2;

k = AddNote(fileName, p2);

if (k < 0) {

cout << "\nНевозможно прочитать файл!";

}

break;

case 6:

cout << "\nНаименование файла (без пробелов, через .txt) ";

cin >> fileName;

cout << "Укажите значение: ";

cin >> num1;

HandleFile1(fileName, "result1.txt", num1);

break;

case 7:

cout << "\nНаименование файла (без пробелов, через .txt) ";

cin >> fileName;

cout << "Укажите значение и число, на которое увеличить: ";

cin >> num1 >> num2;

HandleFile2(fileName, "result1.txt", num1, num2);

break;

case 8:

cout << "\nНаименование файла (без пробелов, через .txt) ";

cin >> fileName;

cout << "Укажите номер элемента и сколько строк добавить: ";

cin >> num1 >> num2;

HandleFile3(fileName, "result1.txt", num1, num2);

break;

}

} while (c != 0 && c <= 0);

return 0;

}

Файл Pair.cpp

#include "Pair.h"

Pair::Pair() { first = 0; second = 0; }

Pair::~Pair() {}

Pair::Pair(int F, double S) { first = F; second = S; }

Pair::Pair (const Pair& p) { first = p.first; second = p.second; }

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

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

first = p.first;

second = p.second;

return \*this;

}

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

out << "Первое число (int): " << p.first << " Второе число (double): " << p.second << "\n";

return out;

}

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

cout << "Введите первое число (int): ";

in >> p.first;

cout << "Введите второе число (double): ";

in >> p.second;

return in;

}

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

fin >> p.first;

fin >> p.second;

return fin;

}

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

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

return fout;

}

Файл Pair.h

#pragma once

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class Pair {

int first;

double second;

public:

Pair();

Pair(int, double);

Pair(const Pair&);

Pair operator = (const Pair&);

friend ostream& operator << (ostream& out, const Pair& p);

friend istream& operator >> (istream& in, Pair& p);

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

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

~Pair();

};

Файл File\_work.h

#pragma once

#include "Pair.h"

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int MakeFile(const char\* f\_first) {

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

if (!stream) return -1;

cout << "\nФайл создан!\n";

stream.close();

return 0;

}

void PrintFile(const char\* f\_first) {

string text;

ifstream f(f\_first);

if (f.is\_open()) {

getline(f, text, '\f');

}

else {

cout << "Невозможно прочитать файл!\n";

}

cout << text << endl;

f.close();

}

int ClearFile(const char\* f\_first, int k) {

string text = f\_first;

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

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

if (!stream) return -1;

stream.close();

temp.close();

remove(f\_first);

rename("temp", f\_first);

cout << "\nВсе записи из файла удалены!\n";

return 0;

}

int AddFile(const char\* f\_first, Pair pp) {

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

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

if (!stream) return -1;

Pair p;

int i = 0, l = 0;

while (stream >> p) {

if (stream.eof()) break;

i++;

}

stream.close();

temp.close();

remove(f\_first);

rename("temp", f\_first);

return l;

}

int AddNote(const char\* f\_first, Pair pp) {

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

if (!stream) return -1;

stream << pp;

cout << "\nФайл изменен!\n";

return 1;

}

int ChangeFile(const char\* f\_first, Pair pp) {

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

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

if (!stream) { return -1; }

Pair p;

int i, l;

i = 0;

l = 0;

while (stream >> p) {

if (stream.eof()) break;

i++;

temp << p;

}

stream.close();

temp.close();

remove(f\_first);

rename("temp", f\_first);

cout << "\nФайл изменен!\n";

return l;

}

void HandleFile1(const char\* f\_first, const char\* f\_result, int nMoreThan) {

ifstream streamInput(f\_first);

ofstream streamOutput(f\_result);

int a = 0;

while (streamInput >> a) {

if (a > nMoreThan) {

string s = to\_string(a);

streamOutput << s << endl;

}

}

streamInput.close();

streamOutput.close();

}

void HandleFile2(const char\* f\_first, const char\* f\_result, int nValue, int L) {

ifstream streamInput(f\_first);

ofstream streamOutput(f\_result);

int a = 0;

while (streamInput >> a) {

if (a == nValue) {

a += L;

}

string s = to\_string(a);

streamOutput << s << endl;

}

streamInput.close();

streamOutput.close();

}

void HandleFile3(const char\* f\_first, const char\* f\_result, int nNumber, int nCount) {

ifstream streamInput(f\_first);

ofstream streamOutput(f\_result);

int a = 0;

int nLine = 1;

while (streamInput >> a) {

string s = to\_string(a);

streamOutput << s << endl;

if (nLine == nNumber) {

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

streamOutput << i << "\n";

}

}

nLine++;

}

streamInput.close();

streamOutput.close();

}

