Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

**по лабораторной работе № 18.10**

Дисциплина: Информатика

# Тема: «Объектно-ориентированное программирование. Сохранение данных в файле с использованием потоков»

Вариант 1

Выполнил работу:

студент группы РИС-20-1Б

Азмагулов Артём Вадимович

Проверила:

Доцент кафедры ИТАС

Полякова О.А.

Пермь

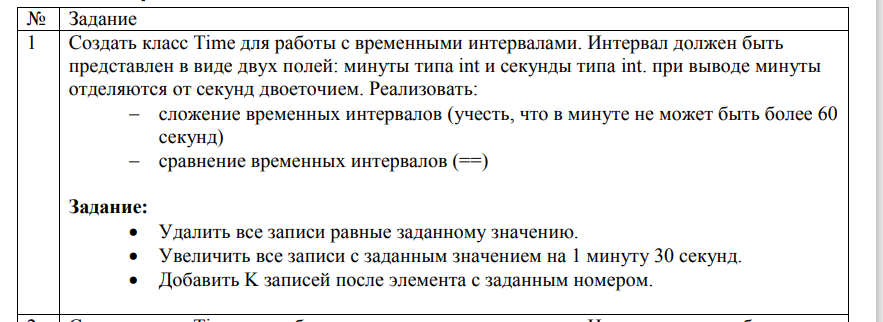
2021

**Цель работы**

* 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.

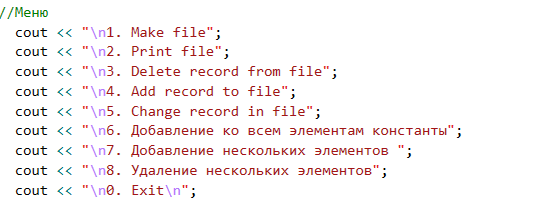
**Постановка задачи**

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.

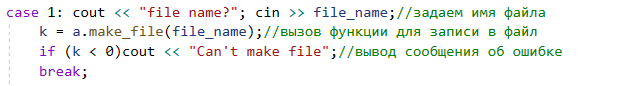


**Анализ задачи**

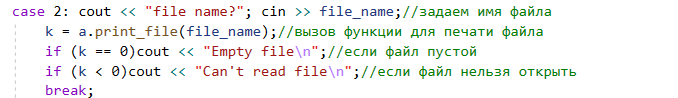
1. Необходимые действия
2. Реализовать меню с шестью опциями



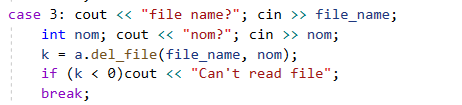
1. Реализовать функцию создания файла



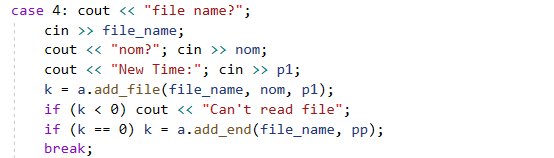
1. Реализовать вывод содержимого из файла



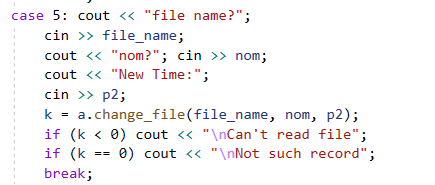
1. Добавить функцию удаления данных из файла



1. Позволить пользователю добавлять записи в файл после определенного значения



1. Увеличить все указанные записи на 1 минуту 30 секунд



1. Программный код

file\_work.h

#include "Time.h"

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class file\_work {

public:

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

{

fstream stream(f\_name, ios::out | ios::trunc);//открыть для записи

if (!stream)return -1;//ошибка открытия файла

int n;

Time p;

cout << "N?"; cin >> n;

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

{

cin >> p;//ввод атрибутов объекта из стандартного потока

stream << p << "\n";//запись объекта в файловый поток

}

stream.close();//закрыть поток

return n;//вернуть количество записанных объектов

}

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

{

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

Time p; int i = 0;

while (stream >> p)

{

cout << p << "\n";

i++;

}

stream.close();

return i;

}

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

{

fstream temp("temp", ios::out);//открыть для записи

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

int i = 0; Time p;

while (stream >> p)//пока нет конца файла выполняем чтение объекта

{

//если прочитан признак конца файла,то выход из цикла

if (stream.eof())break;

i++;

//если номер объекта не равен k, то записываем его во вспомогательый файл

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

}

//закрыть файлы

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

remove(f\_name);//удалить старый файл

rename("temp", f\_name);// переименовать temp

return i;//количество прочитанных

}

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

fstream temp("temp", ios::out);//открыть для записи

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

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

while (stream >> p)

{

if (stream.eof())break;

i++;

if (i == k)

{

temp << pp;//записать в temp новую запись

l++;

}

temp << p;

}

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

remove(f\_name);

rename("temp", f\_name);

return l;//количество добавленных

}

int add\_end(const char\* f\_name, Time pp)

{

fstream stream(f\_name, ios::app);//открыть для добавления

if (!stream)return -1;//ошибка открытия файла

stream << pp; //записать новую запись

return 1;

}

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

{

fstream temp("temp", ios::out);//открыть для записи

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

Time 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;//количество измененных элементов

}

int change\_all(const char\* f\_name){

fstream temp("temp", ios::out);//открыть для записи

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

int i=0;

int l = 0;

Time p;

Time d(1, 30);

Time c;

while (stream >> c)

{

if (stream.eof())break;

i++;

temp<<c+d;

l++;

}

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

remove(f\_name);

rename("temp", f\_name);

return l;//количество измененных элементов

}

int adds\_file(const char\* f\_name, int k, int col) {

fstream temp("temp", ios::out);//открыть для записи

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

Time pp;

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

while (stream >> p)

{

if (stream.eof())break;

i++;

if (i == k)

{

for (int z = 1; z < col; z++) {

cout << "New Time:"; cin >> pp;

temp << pp;//записать в temp новую запись

l++;

}

}

temp << p;

}

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

remove(f\_name);

rename("temp", f\_name);

return l;//количество добавленных

}

int dels\_file(const char\* f\_name, Time p1)

{

fstream temp("temp", ios::out);//открыть для записи

fstream stream(f\_name, ios::in);//открыть для чтения

if (!stream)return -1;//ошибка открытия файла

int i = 0; Time p;

while (stream >> p)//пока нет конца файла выполняем чтение объекта

{

//если прочитан признак конца файла,то выход из цикла

if (stream.eof())break;

i++;

//если номер объекта не равен k, то записываем его во вспомогательyый файл

if (p==p1!=0) temp << p;

}

//закрыть файлы

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

remove(f\_name);//удалить старый файл

rename("temp", f\_name);// переименовать temp

return i;//количество прочитанных

}

};

18.10

#include "Time.h"

#include <iostream>

#include <fstream>

#include <string>

#include "file\_work.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

file\_work a;

Time p, p1, p2,pp;

int col;

int k, c;

char file\_name[30];

do

{ //Меню

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. Добавление ко всем элементам константы";

cout << "\n7. Добавление нескольких элементов ";

cout << "\n8. Удаление нескольких элементов";

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

cin >> c;

switch (c)

{

case 1: cout << "file name?"; cin >> file\_name;//задаем имя файла

k = a.make\_file(file\_name);//вызов функции для записи в файл

if (k < 0)cout << "Can't make file";//вывод сообщения об ошибке

break;

case 2: cout << "file name?"; cin >> file\_name;//задаем имя файла

k = a.print\_file(file\_name);//вызов функции для печати файла

if (k == 0)cout << "Empty file\n";//если файл пустой

if (k < 0)cout << "Can't read file\n";//если файл нельзя открыть

break;

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

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

k = a.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 Time:"; cin >> p1;

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

if (k < 0) cout << "Can't read file";

if (k == 0) k = a.add\_end(file\_name, pp);

break;

case 5: cout << "file name?";

cin >> file\_name;

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

cout << "New Time:";

cin >> p2;

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

if (k < 0) cout << "\nCan't read file";

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

break;

case 6:

k = a.change\_all(file\_name);

if (k < 0) cout << "Can't read file";

if (k == 0) k = a.add\_end(file\_name, pp);

break;

case 7:

cout << "file name?";

cin >> file\_name;

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

cout << "количество?"; cin >> col;

cout << "New Time:"; cin >> p1;

k = a.adds\_file(file\_name, nom,col);

if (k < 0) cout << "Can't read file";

if (k == 0) k = a.add\_end(file\_name, pp);

break;

case 8:

cout << "file name?"; cin >> file\_name;

cout << "Значение, которое надо удалить?"; cin >> p1;

k = a.dels\_file(file\_name, p1);

if (k < 0)cout << "Can't read file";

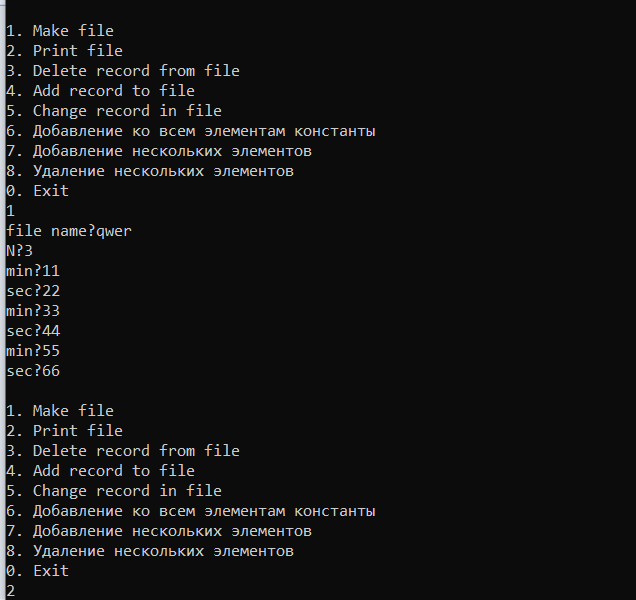
break;

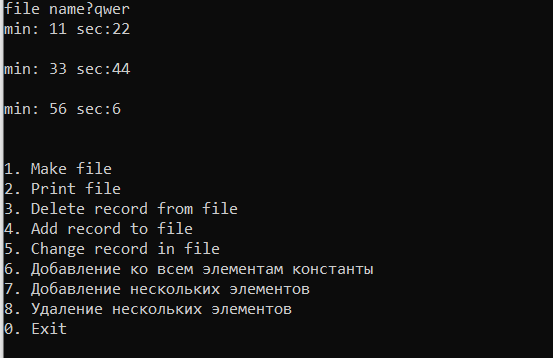
}

} while (c != 0);

}

1. Демонстрация работы программы





1. Типы переменных

Int, Money, fstream, char[]

1. Представление данных

Int – значение рублей и копеек, хранение количества элементов в файле, индекс строки в файле, выбор действий с помощью меню

Money – класс для хранения значений рублей и копеек

Fstream – файловый поток

Char[] – массив символов для хранения имени файла

1. Контрольные вопросы
   * 1. Что такое поток?

- последовательность битов

* + 1. Какие типы потоков существуют?
    - Стандартные: только однонаправленные, либо входные, либо выходные.
    - Строковые: могут быть и однонаправленными и двунаправленными
    - Файловые: могут быть и однонаправленными и двунаправленными.
    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?

operator<<

* + 1. Какая операция используется при вводе из форматированных потоков?

operator>>

* + 1. Какие методы используются при выводе в форматированный поток?

Cout, clog, cerr

* + 1. Какие методы используется при вводе из форматированного потока?

Cin, getline, get

* + 1. Какие режимы для открытия файловых потоков существуют?

In, out, trunk, app, ate, binary

* + 1. Какой режим используется для добавления записей в файл?

Out, app

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

in

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

In, out, app

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

Out, trunk

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

Если файла нет, то он создается для записи данных. Если файл существует, то запись новых данных производится в конец файла

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

Стирание старых данных и запись новых. Если файла нет, то он создается

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

Стирание, чтение и запись данных. Если файла нет, то он создается

* + 1. Каким образом можно открыть файл для чтения?

ifstream stream;

stream.open(“file.txt”);

* + 1. Каким образом можно открыть файл для записи?

ofstream stream;

stream.open(“file.txt”);

* + 1. Привести примеры открытия файловых потоков в различных режимах.

stream.open(“file.txt”, ios::in);

stream.open(“file.txt”, ios::out);

stream.open(“file.txt”, ios::in | ios::out);

stream.open(“file.txt”, ios::in | ios::out | ios::trunk);

stream.open(“file.txt”, ios::ate);

stream.open(“file.txt”, ios::binary);

* + 1. Привести примеры чтения объектов из потока.

friend fstream& operator>>(fstream& fin, Money& money)

{

fin >> money.Rubles;

fin >> money.Kopecks;

return fin;

}

istream& operator >> (istream& in, Money& money) {

cout << "Введите руб.: ";

in >> money.Rubles;

cout << "Введите коп.: ";

in >> money.Kopecks;

return in;

}

* + 1. Привести примеры записи объектов в поток.

friend fstream& operator<<(fstream& fout, const Money& money)

{

fout << money.Rubles << "\n" << money.Kopecks << "\n";

return fout;

}

ostream& operator << (ostream& out, const Money& money) {

double double\_Kopecks = money.Kopecks;

string zero\_filler = "";

if ((double\_Kopecks / 10) < 1)

zero\_filler = "0";

return (out << money.Rubles << ',' << zero\_filler << money.Kopecks << " руб.");

}

* + 1. Сформулировать алгоритм удаления записей из файла.
       1. Открыть на чтение файл, из которого нужно удалить элементы
       2. Создать для записи дополнительный файл
       3. Из файла, открытого на чтение, записать в дополнительный файл те строки, которые не нужно удалять
       4. Закрыть оба файла.
       5. Удалить файл, открытый на чтение
       6. Изменить название дополнительного файла на название удаленного файла (имя заранее нужно сохранить)
    2. Сформулировать алгоритм добавления записей в файл
       1. Открыть на чтение файл, в который нужно добавить данные
       2. Создать для записи дополнительный файл
       3. Из файла, открытого на чтение, записать в дополнительный файл те строки, после которых нужно добавить новые данные
       4. Дописать в дополнительный файл новые строки
       5. Дописать в дополнительный файл оставшиеся строки из файла, открытого на чтение
       6. Закрыть оба файла.
       7. Удалить файл, открытый на чтение
       8. Изменить название дополнительного файла на название удаленного файла (имя заранее нужно сохранить)
    3. Сформулировать алгоритм изменения записей в файле.
       1. Открыть на чтение файл, в котором нужно изменить данные
       2. Создать для записи дополнительный файл
       3. Из файла, открытого на чтение, записать в дополнительный файл те строки, которые не нужно изменять
       4. Дописать в дополнительный файл строки с изменениями из файла, открытого на чтение
       5. Дописать в дополнительный файл оставшиеся строки из файла, открытого на чтение
       6. Закрыть оба файла.
       7. Удалить файл, открытый на чтение
       8. Изменить название дополнительного файла на название удаленного файла (имя заранее нужно сохранить)