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**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Сохранение данных в файле с использованием потоков.

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Токарев Павел Аркадьевич

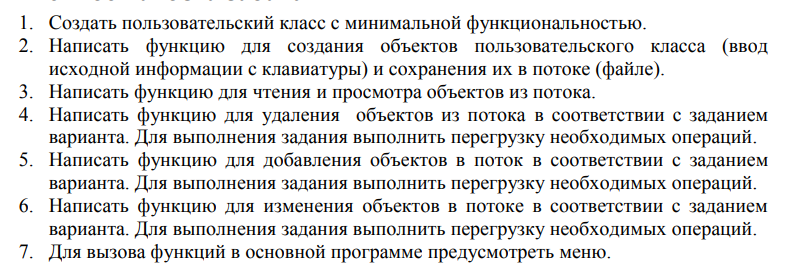
Проверил

Доцент кафедры ИТАС

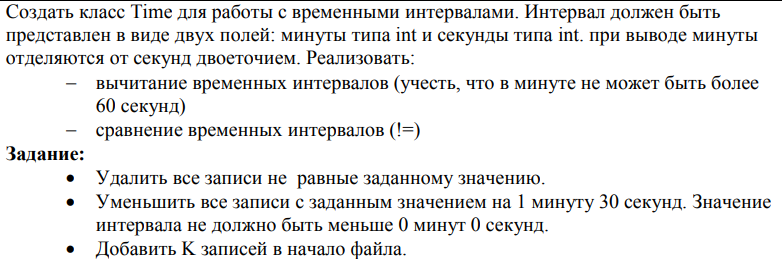
Полякова Ольга Андреевна

Г. Пермь-2023

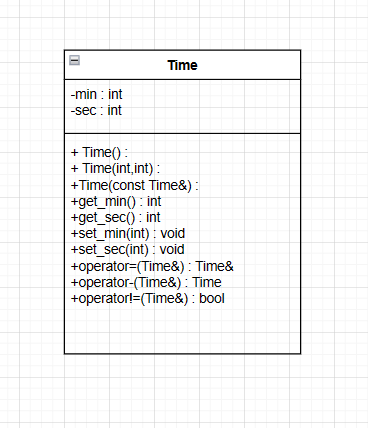
**Постановка задачи**



Вариант 2:



**Диаграмма UML**

****

**Код программы**

Time.cpp

#include "Time.h"

#include <iostream>

using namespace std;

Time& Time::operator=(const Time& t) {

if (&t == this) {

return \*this;

}

else {

min = t.min;

sec = t.sec;

return \*this;

}

}

Time Time::operator-(const Time& t) {

int tmp1 = (min \* 60) + sec;

int tmp2 = (t.min \* 60) + t.sec;

Time newtime;

newtime.min = (tmp1 - tmp2) / 60;

newtime.sec = (tmp1 - tmp2) % 60;

return newtime;

}

bool Time::operator!=(const Time& t) {

if ((min != t.min) && (sec != t.sec)) {

return true;

}

else {

return false;

}

};

istream& operator>>(istream& in, Time& t) {

cout << "Минута:";

in >> t.min;

cout << "Секунда:";

in >> t.sec;

return in;

}

ostream& operator<<(ostream& out, const Time& t) {

return (out << t.min << ":" << t.sec);

}

fstream& operator>>(fstream& fin, Time& t) {

fin >> t.min;

fin >> t.sec;

return fin;

}

fstream& operator<<(fstream& fout, const Time& t) {

fout << t.min << endl << t.sec << endl;

return fout;

}

Time.h

#pragma once

#include <iostream>

#include<fstream>

using namespace std;

class Time {

int min, sec;

public:

Time() {

min = 0;

sec = 0;

}

Time(int tmp1, int tmp2) {

min = tmp1;

sec = tmp2;

}

Time(const Time& t) {

min = t.min;

sec = t.sec;

}

~Time() {};

int getminute() {

return min;

}

int getsecond() {

return sec;

}

void setminute(int tmp) {

min = tmp;

}

void setsecond(int tmp) {

sec = tmp;

}

Time& operator=(const Time&);

Time operator-(const Time&);

bool operator!=(const Time&);

friend istream& operator>>(istream& in, Time& t);

friend ostream& operator<<(ostream& out, const Time& t);

friend fstream& operator>>(fstream& fin, Time& t);

friend fstream& operator<<(fstream& fout, const Time& t);

};

Filestream.h

#pragma once

#include <iostream>

#include"Time.h"

using namespace std;

void CreateFile(const char\* filename) {

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

if (!fout) {

cout << "Не удалось открыть файл" << endl;

}

else {

Time tmp;

int size;

cout << "Введите число объектов для записи: ";

cin >> size;

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

cin >> tmp;

fout << tmp;

}

fout.close();

}

}

void PrintFile(const char\* filename) {

fstream fin(filename, ios::in);

if (!fin) {

cout << "Не удалось открыть файл" << endl;

}

else {

Time tmp;

while (fin >> tmp) {

cout << tmp << endl;

}

}

fin.close();

}

void DeleteFromFile(const char\* filename, Time&t) {

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

fstream fin(filename, ios::in);

Time tm;

while (fin >> tm) {

if (fin.eof()) {

break;

}

if ((t.getminute() != tm.getminute()) || (t.getsecond() != tm.getsecond())) {

temp << tm;

}

}

temp.close();

fin.close();

remove(filename);

rename("temp", filename);

}

void AddKElelement(const char\* filename, int k) {

Time tm;

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

fstream fout(filename, ios::in);

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

cin >> tm;

temp << tm;

}

while (fout >> tm) {

if (fout.eof()) {

break;

}

temp << tm;

}

fout.close();

temp.close();

remove(filename);

rename("temp", filename);

}

void ReduceTime(const char\* filename, Time& t) {

Time tm;

Time rq;

rq.setminute(1);

rq.setsecond(30);

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

fstream fout(filename, ios::in);

int min30 = t.getminute() \* 60 + t.getsecond();

while (fout >> tm) {

int newtime = tm.getminute() \* 60 + tm.getsecond();

if (tm.getminute() == t.getminute() && tm.getsecond() == t.getsecond() && tm.getminute() \* 60 + tm.getsecond() >= 90) {

tm = tm - rq;

temp << tm;

}

else {

temp << tm;

}

}

fout.close();

temp.close();

remove(filename);

rename("temp", filename);

}

main.cpp

#include <iostream>

#include"Filestream.h"

#include"Time.h"

using namespace std;

void menu() {

int sw=-1;

int k;

Time sample;

char fname[30];

while (sw != 0) {

cout << "1.Создать файл" << endl;

cout << "2.Вывести элементы файла на экран" << endl;

cout << "3.Удалить элементы с заданным значением" << endl;

cout << "4.Добавить К элементов в начало файла" << endl;

cout << "5.Уменьшить время элементов с заданным значение на 1 минуту 30 секунд" << endl;

cout << "6.Выход из прграммы" << endl;

cin >> sw;

switch (sw) {

case 1:

cout << "Введите имя файла:";

cin >> fname;

CreateFile(fname);

break;

case 2:

PrintFile(fname);

break;

case 3:

cout << " Введите значения элементов для удаления"<<endl;

cin >> sample;

DeleteFromFile(fname, sample);

break;

case 4:

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

cin >> k;

AddKElelement(fname, k);

break;

case 5:

cin >> sample;

ReduceTime(fname, sample);

break;

default:

sw = 0;

break;

}

}

}

int main()

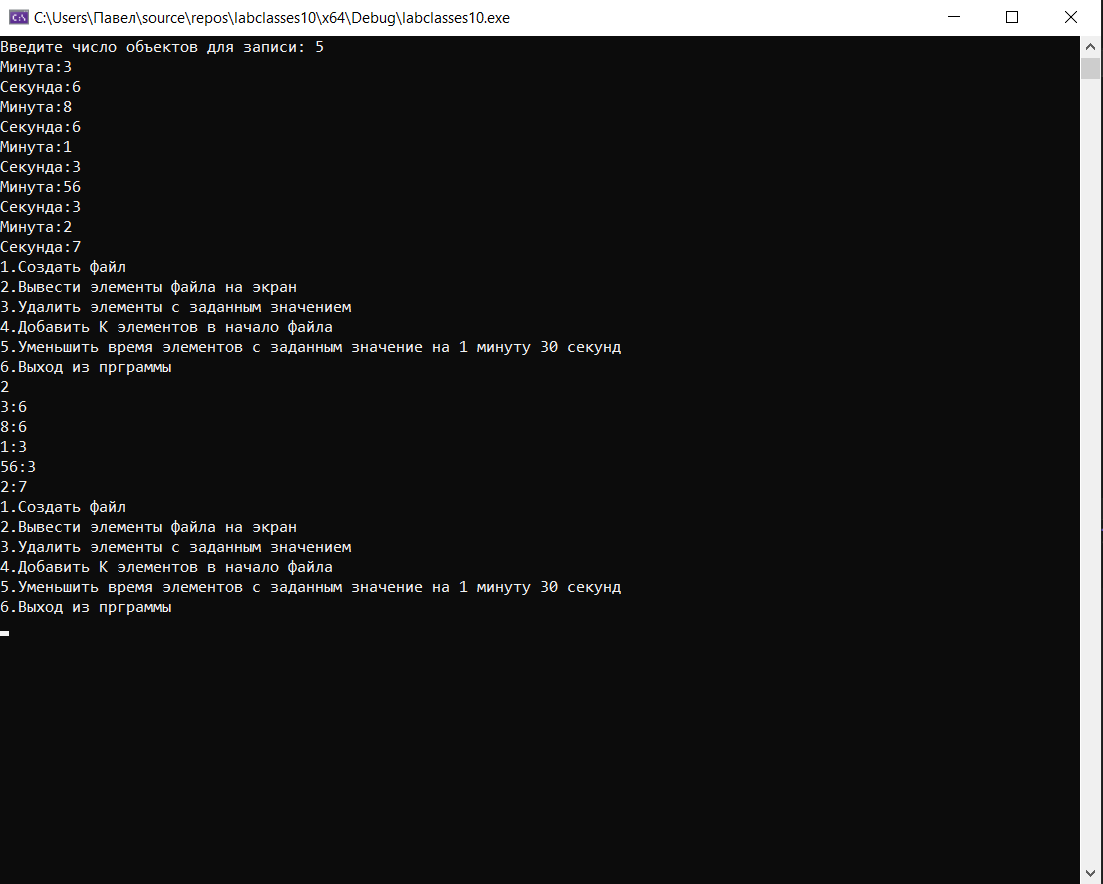
{

setlocale(LC\_ALL, "rus");

menu();

}

**Результат работы программы**

****

**Анализ результатов**

Удалось успешно запись объектов класса в текстовый файл, используя файловые потоки.

**Контрольные вопросы**

1.Что такое поток?

Поток - определяется как последовательность байтов и не зависит от конкретного устройства, с которым производится обмен (оперативная память, файл на диске, клавиатура или принтер)

2.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.

3.Какую библиотеку надо подключить при использовании стандартных потоков?

#include <iostream>

4.Какую библиотеку надо подключить при использовании файловых потоков?

#include <fstream>

5.Какую библиотеку надо подключить при использовании строковых потоков?

#include <string>

6.Какая операция используется при выводе в форматированный поток?

Operator<<

7.Какая операция используется при вводе из форматированных потоков?

Operator>>

8.Какие методы используются при выводе в форматированный поток?

ostream

9.Какие методы используется при вводе из форматированного потока?

istream

10.Какие режимы для открытия файловых потоков существуют?

in открыть поток для чтения (по умолчанию для ifstream) out (по умолчанию для оfstream) trunk удалить старое содержимое файла (по умолчанию для оfstream) app открыть поток для записи в конец файла ate открыть поток для чтения и/или записи и встать в конец файла binary открыть поток в двоичном режиме

11.Какой режим используется для добавления записей в файл?

out|app дозапись в файл, если файла нет, то он создается

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

in

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

app

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

out

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

файл открывается для дозаписывания. Старые данные не удаляются.

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

стирание и запись, если файла нет, то он создается

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

стирание, чтение и запись, если файла нет, то он создается

18.Каким образом можно открыть файл для чтения?

ios::in

19.Каким образом можно открыть файл для записи?

ios::out

20.Привести примеры открытия файловых потоков в различных режимах.

ios::app: файл открывается для дозаписи. Старые данные не удаляются.

ios::ate: после открытия файла перемещает указатель в конец файла

ios::trunc: файл усекается при открытии. Может быть установлен, если также установлен режим out

ios::binary: файл открывается в бинарном режиме

21.Привести примеры чтения объектов из потока.

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

22.Привести примеры записи объектов в поток.

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

23.Сформулировать алгоритм удаления записей из файла.

Открываем поток и файл, в цикле идем до конца файла, и удаляем каждую строку закрываем поток и файл.

24.Сформулировать алгоритм добавления записей в файл.

Открываем потом и файл, вставляем запись, закрываем поток и файл.

25.Сформулировать алгоритм изменения записей в файле.

Открываем файл, идем в цикле пока не найдем нужную нас строку, нашли строку изменили в ней данные. Выходим из цикла, закрываем поток.