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: «Информатика»

"Классы и объекты. Сохранение данных в файле с использованием потоков."

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Бадртдинов Т.З

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Г. Пермь-2023

# 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.

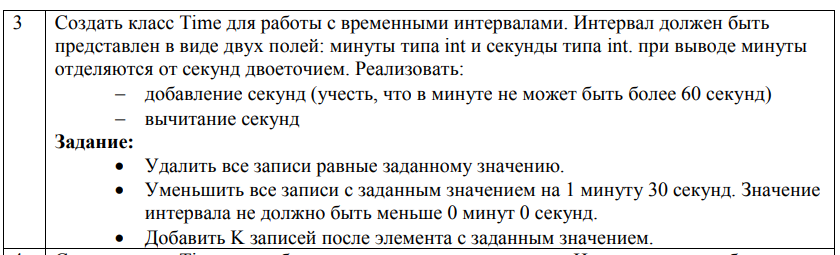
4. Написать функцию для удаления объектов из потока в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.

5. Написать функцию для добавления объектов в поток в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.

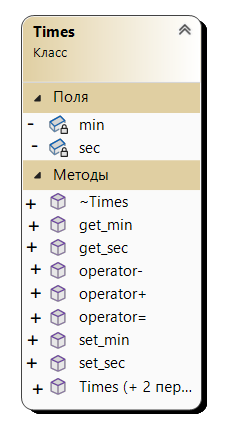
6. Написать функцию для изменения объектов в потоке в соответствии с заданием варианта. Для выполнения задания выполнить перегрузку необходимых операций.

7. Для вызова функций в основной программе предусмотреть меню.

Вариант 3



Описание класса

**

# Код программы(file\_work.h)

#include "Timess.h"

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int del\_file(const char\* f\_name, Times qwe) {

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,c,d; Times p;

c = qwe.get\_min();

d = qwe.get\_sec();

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

{

//если прочитан признак конца файла,то выход из

if (stream.eof())break;

i++;

int a, b;

a = p.get\_min();

b = p.get\_sec();

//если номер объекта не равен k, то записыва

if ((a == c) && (b == d))

i;

else 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 makes\_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;

Times 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

}

stream.close();//закрыть поток

return n;//вернуть количество записанных объектов

}

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

{

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

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

Times p;

int i = 0;

while (stream >> p)

{

cout << p << "\n";

i++;

}

stream.close();

return i;

}

int add\_file(const char\* f\_name, int k, Times 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

Times 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, Times 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) {

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

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

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

Times p; int i = 0; Times qwe;

qwe.set\_min(1);

qwe.set\_sec(30);

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

{

if (stream.eof())break;

i++;

p=p - qwe;

temp << p;

}

stream.close();

temp.close();

remove(f\_name);

rename("temp", f\_name);

return i;//количество элементов

}

# Код программы(timess.cpp)

#include "Timess.h"

#include <iostream>

using namespace std;

//перегрузка операции присваивания

Times& Times::operator=(const Times& t)

{

//проверка на самоприсваивание

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

min = t.min;

sec = t.sec;

return \*this;

};

//перегрузка бинарной операции сложения

Times Times::operator+(const Times&t)

{

int temp1=min\*60+sec;

int temp2=t.min\*60+t.sec;

Times p;

p.min=(temp1+temp2)/60;

p.sec=(temp1+temp2)%60;

return p;

}

//перегрузка бинарной операции вычитания

Times Times::operator-(const Times& t)

{

int temp1 = min \* 60 + sec;

int temp2 = t.min \* 60 + t.sec;

Times p;

p.min = (temp1 - temp2) / 60;

p.sec = (temp1 - temp2) % 60;

if (p.sec < 0) p.sec = 0 ;

if (p.min < 0) p.min = 0;

return p;

}

//перегрузка глобальной функции-операции ввода

istream&operator>>(istream&in, Times&t)

{

cout<<"min? ";in>>t.min;

cout<<"sec? ";in>>t.sec;

return in;

}

//перегрузка глобальной функции-операции вывода

ostream&operator<<(ostream&out, const Times&t)

{

out << "min:" << t.min << "sec:" << t.sec << "\n";

return out;

}

fstream& operator>>(fstream& fin, Times& p)

{

fin >> p.min;

fin >> p.sec;

return fin;

}

fstream& operator<<(fstream& fout, const Times& p)

{

fout << p.min << "\n" << p.sec << "\n";

return fout;

}

# Код программы(timess.h)

#pragma once

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class Times

{

private:

int min, sec;

public:

Times() { min = 0; sec = 0; };

Times(int m, int s) { min = m; sec = s; }

Times(const Times& t) { min = t.min; sec = t.sec; }

~Times() {};

int get\_min() { return min; }

int get\_sec() { return sec; }

void set\_min(int m) { min = m; }

void set\_sec(int s) { sec = s; }

//перегруженные операции

Times&operator=(const Times&);

Times operator+ (const Times&);

Times operator- (const Times&);

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

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

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

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

};

# Код программы(main)

#include "Timess.h"

#include "file\_work.h"

#include <iostream>

using namespace std;

int main()

{

Times p,p1,p2,qwe;

int k, c;

char file\_name[30];

do

{ //Меню

cout << "\n1. Make file";//создание файла

cout << "\n2. Print file";//печать файла

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

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

cout << "\n5. -1:30";

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

cin >> c;

switch (c)

{

case 1:

cout << "file name?";

cin >> file\_name;//задаем имя файла

k = makes\_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 = prints\_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 << "?";

cin >> qwe;

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

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

break;

case 4:

cout << "file name?";

cin >> file\_name;

cout << "nom?";

cin >> nom;

cout << "New person:";

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;

k=change\_file(file\_name);

if(k<0)

cout<<"\nCan't read file";

break;

}

}

while(c!=0);

return 0;

# }

# Работы программы

