Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №18.10**

Дисциплина: «информатика»

Тема: Сохранение данных в файле с использованием потоков

Вариант 5

Выполнил работу

студент группы РИС-20-1б

Зверев А.Д.

Проверила

Доцент кафедры ИТАС

Полякова О.А.

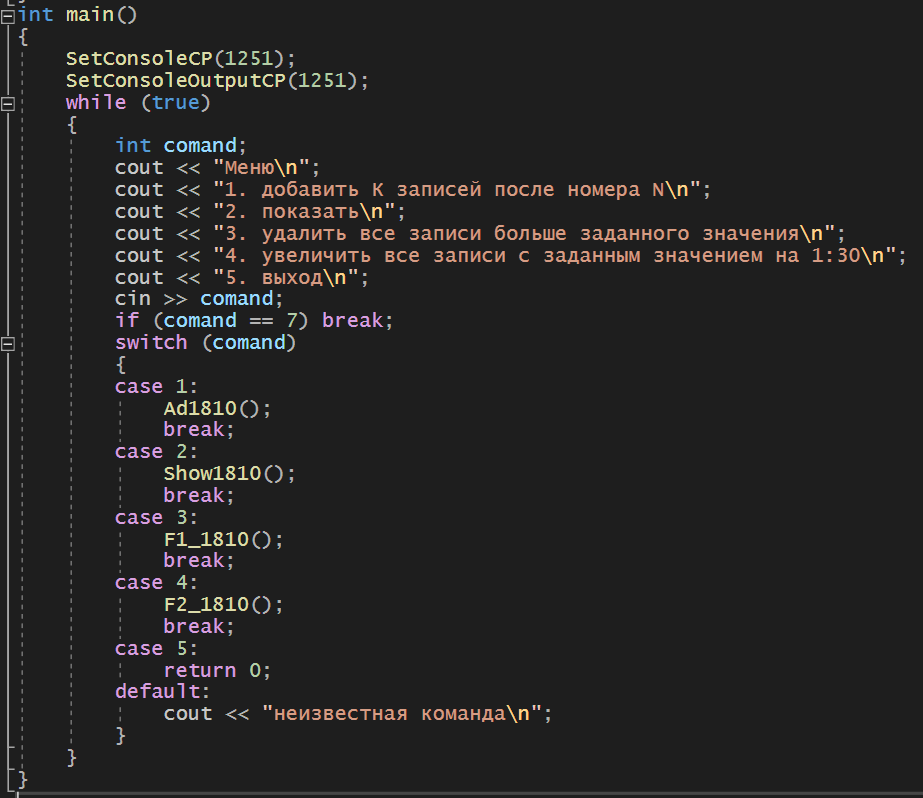
Пермь, 2021

**Постановка задачи**

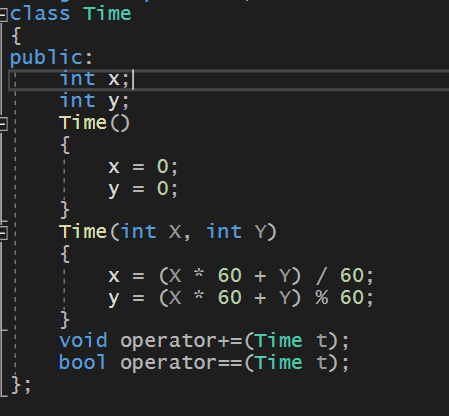
1. Создать пользовательский класс с минимальной функциональностью.
2. Написать функцию для создания объектов пользовательского класса (ввод исходной информации с клавиатуры) и сохранения их в потоке (файле).
3. Написать функцию для чтения и просмотра объектов из потока.
4. Написать функцию для удаления 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. Для решения задачи необходимо…
   1. Написать функцию int main, в которой вызывается меню программы, через которое вызываются все необходимые функции;



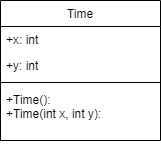
* 1. В файле lab18.h описать класс Time, необходимый для хранения различных данных и работы с ними;



1. В программе были использованы следующие типы данных:
   1. Тип данных int для хранения введенных данных;



**Блок—схема программы**



**Решение**

**18.cpp**

#include "lab18.h"

#include <string>

#include <iostream>

#include <fstream>

#include <math.h>

using namespace std;

void Time::operator+=(Time t)

{

int X = this->x + t.x;

int Y = this->y + t.y;

this->x = (X \* 60 + Y) / 60;

this->y = (X \* 60 + Y) % 60;

}

bool Time::operator==(Time t)

{

return this->x == t.x && this->y == t.y;

}

**main.cpp**

#include <iostream>

#include <fstream>

#include "lab18.h"

#include <string>

#include <windows.h>

#include <vector>

#include <list>

#include <queue>

#include <stack>

#include <set>

#include <map>

#include <algorithm>

#include <numeric>

using namespace std;

ostream& operator << (ostream& out, Time P1)

{

return (out << P1.x << ":" << P1.y << endl);

}

istream& operator >> (istream& in, Time& P1)

{

cout << "введите минуты\n";

in >> P1.x;

cout << "введите секунды\n";

in >> P1.y;

return in;

}

fstream& operator >> (fstream& fin, Time& P1)

{

fin >> P1.x;

fin >> P1.y;

return fin;

}

fstream& operator<<(fstream& fout, Time P1)

{

fout << P1.x << "\n" << P1.y << "\n";

return fout;

}

void Ad1810()

{

fstream stream("TestFile.txt", ios::in);

fstream help("TestFile2.txt", ios::out);

int K, L, c = 0;

Time p;

cout << "введите позицию нового элемента\n";

cin >> K;

cout << "введите количество новых элементов\n";

cin >> L;

while (c != K)

{

stream >> p;

help << p;

c++;

}

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

{

cin >> p;

help << p;

}

while (stream >> p) help << p;

stream.close();

help.close();

remove("TestFile.txt");

rename("TestFile2.txt", "TestFile.txt");

}

void Show1810()

{

fstream stream("TestFile.txt", ios::in);

Time p;

while (stream >> p)

{

cout << p;

}

stream.close();

}

void F1\_1810()

{

fstream stream("TestFile.txt", ios::in);

fstream help("TestFile2.txt", ios::out);

int K, L;

Time p;

cout << "введите минуты\n";

cin >> K;

cout << "введите секунды\n";

cin >> L;

while (stream >> p) if (p.x \* 60 + p.y < K \* 60 + L) help << p;

stream.close();

help.close();

remove("TestFile.txt");

rename("TestFile2.txt", "TestFile.txt");

}

void F2\_1810()

{

fstream stream("TestFile.txt", ios::in);

fstream help("TestFile2.txt", ios::out);

int K, L;

Time p;

cout << "введите минуты\n";

cin >> K;

cout << "введите секунды\n";

cin >> L;

while (stream >> p)

{

if (p.x == K && p.y == L)

{

Time t;

t.x = 1;

t.y = 30;

p += t;

help << p;

}

else help << p;

}

stream.close();

help.close();

remove("TestFile.txt");

rename("TestFile2.txt", "TestFile.txt");

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

while (true)

{

int comand;

cout << "Меню\n";

cout << "1. добавить К записей после номера N\n";

cout << "2. показать\n";

cout << "3. удалить все записи больше заданного значения\n";

cout << "4. увеличить все записи с заданным значением на 1:30\n";

cout << "5. выход\n";

cin >> comand;

if (comand == 7) break;

switch (comand)

{

case 1:

Ad1810();

break;

case 2:

Show1810();

break;

case 3:

F1\_1810();

break;

case 4:

F2\_1810();

break;

case 5:

return 0;

default:

cout << "неизвестная команда\n";

}

}

}

**Lab18.h**

#pragma once

#include <string>

#include <iostream>

#include <list>

#include <vector>

#include <fstream>

using namespace std;

class Time

{

public:

int x;

int y;

Time()

{

x = 0;

y = 0;

}

Time(int X, int Y)

{

x = (X \* 60 + Y) / 60;

y = (X \* 60 + Y) % 60;

}

void operator+=(Time t);

bool operator==(Time t);

};

**Скриншоты**

