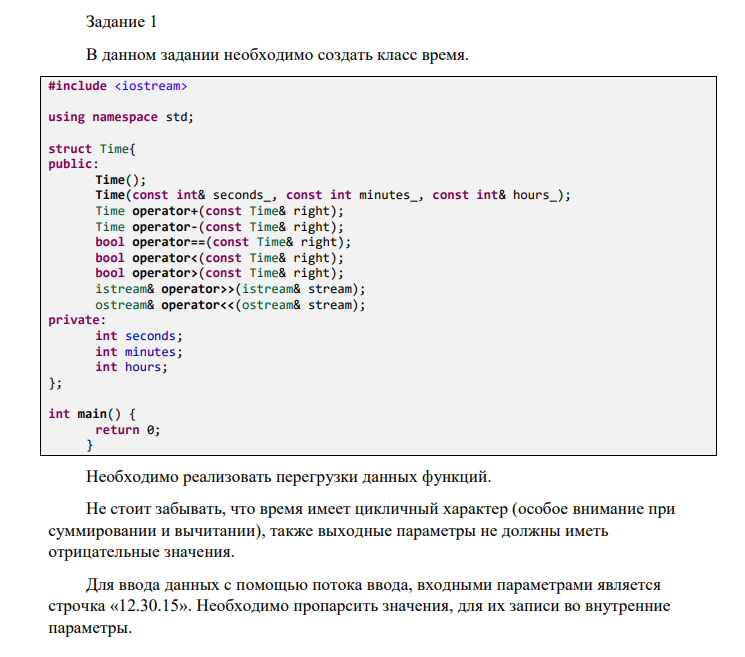
Лабораторная работа №7

Авдеев Владислав ИКТ-31



Код:

//udachi

#include <iostream>

#include <fstream>

#include <string>

#include <sstream>

using namespace std;

struct Time{

    public:

        Time(){

            seconds=0;

            minutes=0;

            hours=0;

        };

        Time(const int& seconds\_, const int minutes\_, const int& hours\_){

            seconds=seconds\_;

            minutes=minutes\_;

            hours=hours\_;

        };

        Time operator+(const Time& right){

            int second=seconds+minutes\*60+hours\*3600+right.seconds+right.minutes\*60+right.hours\*3600;

            if(second>86400) second=second-86400;

            return Time((second-int(second/60)\*60),(second-int(second/3600)\*3600)/60,second/3600);

        };

        Time operator-(const Time& right){

            int second=seconds+minutes\*60+hours\*3600-(right.seconds+right.minutes\*60+right.hours\*3600);

            if(second<0) second=86400-abs(second);

            return Time((second-int(second/60)\*60),(second-int(second/3600)\*3600)/60,second/3600);

        };

        bool operator==(const Time& right){

            if((seconds+minutes\*60+hours\*3600)==(right.seconds+right.minutes\*60+right.hours\*3600)) return true;

            else return false;

        };

        bool operator<(const Time& right){

            if((seconds+minutes\*60+hours\*3600)<(right.seconds+right.minutes\*60+right.hours\*3600)) return true;

            else return false;

        };

        bool operator>(const Time& right){

            if((seconds+minutes\*60+hours\*3600)>(right.seconds+right.minutes\*60+right.hours\*3600)) return true;

            else return false;

        };

     friend std::   istream& operator>>(istream& stream, Time& time);

     friend std::   ostream& operator<<(ostream& stream,const Time& time);

     void SetH(const int& h){this->hours=h;};

     void SetM(const int& m){this->minutes=m;};

     void SetS(const int& s){this->seconds=s;};

    private:

        int seconds;

        int minutes;

        int hours;

};

std::   istream& operator>>(istream& stream, Time& time){

            string b="";

            getline(stream,b,'.');

            time.SetH(stoi(b));

            b="";

            getline(stream,b,'.');

            time.SetM(stoi(b));

            b="";

            getline(stream,b);

            time.SetS(stoi(b));

            return stream;

        }

     std::   ostream& operator<<(ostream& stream,const Time& time){

            stream<<time.hours<<'.'<<time.minutes<<'.'<<time.seconds;

            return stream;

        }

int main() {

    Time a(30,30,1),b(31,30,1),c; // Format Time(Seconds,Minutes,Hours)

   std:: stringstream time("12.30.15");

   cout<<"Vremya a:"<<a<<" Vremya b:"<<b<<endl;

   cout<<"Summa a+b:"<<a+b<<endl;

    cout<<"Raznost' a-b:"<<a-b<<endl;

    cout<<"Sravnenie a==b(1 or 0):"<<bool(a==b)<<endl;

    cout<<"Dalshe zapishem s potoka 'time' na struct 'c' vremya. Dopustim 12.30.15"<<endl;

    time>>c;

    cout<<"Vremya 'c':"<<c<<endl;

    return 0;

}