

Вспоминалки

Русификация

```
#include <iostream>
#include <fstream>
#include <locale>

using namespace std;

int main(){
    setlocale(LC_ALL,"Russian");
    fstream FILE;

    FILE.open("001in.txt",ios_base::in);
    char str[50];
    FILE.getline(str,50);
    cout<<"Первая строка"<<str<<endl;
    FILE.getline(str,50);
    cout<<"Вторая строка"<<str<<endl;
    FILE.getline(str,50);
    cout<<"Третья строка"<<str<<endl;
    FILE.close();
```

```
FILE.open("001out.txt",ios_base::out);
FILE<<"Первая строка"<<endl;
FILE<<"Вторая строка"<<endl;
FILE<<"Третья строка"<<endl;
FILE.close();

FILE.open("001in.txt",ios_base::in);
cout<<endl<<FILE.rdbuf()<<endl;
FILE.close();

FILE.open("001in.txt",ios_base::in);
while(!FILE.eof()){
    FILE.getline(str,50);
    cout<<str<<endl;
}

return 0;
}
```

Русфикация

- * `#include <windows.h>`
- * `SetConsoleCP(1251);`
- * `SetConsoleOutputCP(1251);`

```
system("chcp 1251");  
fstream FILE;
```

```
#include <locale>  
//#include <windows.h>  
  
using namespace std;  
  
int main(){  
    setlocale(LC_ALL, "Russian");  
    //SetConsoleCP(1251);  
}
```

```
#include <locale>  
//#include <windows.h>  
  
using namespace std;  
  
int main(){  
    setlocale(LC_ALL, "");  
    //SetConsoleCP(1251);  
}
```

Сравнение двух времен в лоб

```
struct Time{
    int h,m,s,err;
    Time& read(istream &cin){ //метод возвращает исходный объект
        cin>>h>>m>>s;
        err=!iscorrect();
        correct();
        return *this;
    }
    Time& write(ostream &cout){
        cout<<h/10<<h%10<<" ";
        <<m/10<<m%10<<" ";
        <<s/10<<s%10<<endl;
        return *this;
    }
    Time& writeerror(ostream &cout){
        if (err==1)cout<<"incorrect input \n";
        return *this;
    }
    void correct(){
        m+=s/60;
        s%=60;
        h+=m/60;
        m%=60;
        h%=24;
    }
    bool iscorrect(){
        return h>=0&&h<24&&m>=0&&m<60&&s>=0&&s<60;
    }
    bool ismorethen(Time B){
        return 3600*h+60*m+s>3600*B.h+60*B.m+B.s;
    }
};
```

```
int main(int argc, char const *argv[])
{
    ifstream cin("003in.txt");
    ofstream cout("005out.txt");
    Time t1,t2;
    t1.read(cin);
    t2.read(cin);
    if(t1.ismorethen(t2)) cout<<"t1>t2";|
        else cout<<"t2>t1";
    return 0;
}
```

```
1 80 1111 12
2 17 111 11|
```

```
1 |t2>t1
```

Сравнение двух времен через перегрузку оператора

```
bool operator > (Time A,Time B){  
    return A.ismorethen(B);  
}  
  
int main(int argc, char const *argv[])  
{  
    ifstream cin("003in.txt");  
    ofstream cout("005out.txt");  
    Time t1,t2;  
    t1.read(cin);  
    t2.read(cin);  
    if(t1>t2) cout<<"t1>t2";  
    else cout<<"t2>t1";  
    return 0;  
}
```

Перегрузка ввода и вывода

```
istream& operator>>(istream &cin, Time &A){
    A.read(cin);
    return cin;
}
ostream& operator<<(ostream &cout, Time &A){
    A.write(cout);
    return cout;
}

int main(int argc, char const *argv[])
{
    ifstream cin("003in.txt");
    ofstream cout("005out.txt");
    Time t1,t2;
    cin>>t1>>t2;
    cout<<t1<<t2;
    if(t1>t2) cout<<"t1>t2";
    else cout<<"t2>t1";
    return 0;
}
```

```
000 01 01|
222 22 22
```

```
1 00:01:01
2 06:22:22
3 t2>t1
```

Массив из структур

```
struct TElement{
    int h,m,s;
    TElement& read (istream &cin){
        cin>>h>>m>>s;
        return *this;
    }
    TElement& write(ostream &cout){
        cout<<h<<" "<<m<<" "<<s<<endl;
        return *this;
    }
};

istream& operator >>(istream &cin, TElement &a){
    a.read(cin);
    return cin;
}
ostream& operator << (ostream &cout, TElement &a){
    a.write(cout);
    return cout;
}
```

```
1 5
2 1 2 3
3 4 5 6
4 7 8 9
5 10 11 12
6 13 14 15
```

```
1 1 2 3
2 4 5 6
3 7 8 9
4 10 11 12
5 13 14 15
6
7 13 14 15
8 10 11 12
9 7 8 9
10 4 5 6
11 1 2 3
12
```

Поиск по более сложной структуре

```
struct TElement{
    int h,m,s;
    TElement& read (istream &cin){
        cin>>h>>m>>s;
        return *this;
    }
    TElement& write(ostream &cout){
        cout<<h/10<<h%10<<":"<<
            m/10<<m%10<<":"<<
            s/10<<s%10<<endl;
        return *this;
    }
    bool operator == (TElement b){
        return h==b.h && m==b.m && s==b.s;
    }

    bool operator != (TElement b){
        return !(*this==b);
    }
};
```

```
5
1 2 3
4 5 6
7 8 9
10 11 12
13 14 15
4 5 6
```

```
1 1
```


Поиск максимума. Перегрузка логических операторов

```
bool operator == (TElement b){  
    return a==b.a;  
}  
bool operator != (TElement b){  
    return !(*this==b);  
    //return a!=b.a;  
}  
bool operator > (TElement b){  
    return a>b.a;  
}  
bool operator >=(TElement b){  
    return *this>b||*this==b;  
}  
bool operator <(TElement b){  
    return !(*this>=b);  
}  
bool operator <=(TElement b){  
    return !(*this>b);  
}
```

Поиск минимума

```
int main(){  
    ifstream cin("001in.txt");  
    ofstream cout("001out.txt");  
    setlocale(LC_ALL,"Russian");  
  
    TElement A[100];  
    int N,res=0;  
    cin >>N;  
    for (int i=0;i<N;i++){  
        cin>>A[i];  
    }  
    for (int i=0;i<N;i++){  
        if(A[i]<A[res]){  
            res=i;  
        }  
    }  
    cout <<A[res]<<" " <<res;  
  
    return 0;  
}
```

```
1 10  
2 1 2 3 4 0 6 7 8 9 10
```

```
1 | 0 4
```

Поиск максимума

```
TElement A[100];
int N, res=0;
cin >>N;
for (int i=0; i<N; i++){
    cin>>A[i];
}
for (int i=0; i<N; i++){
    if(A[i]>A[res]){
        res=i;
    }
}
cout <<A[res]<<" " <<res;
```

```
1 10
2 1 2 3 4 0 6 7 8 9 10
```

```
1 10 9
```

Предоставление массива в виде структуры

```
struct TArray{
    TElement A[100];
    int N;

    TArray& read (istream &cin){
        cin>>N;
        for (int i=0;i<N;i++){
            cin >>A[i];
        }
        return *this;
    }
    TArray& write(ostream &cout){
        for (int i=0;i<N;i++){
            cout<<A[i];
        }
        return *this;
    }
};

istream& operator >> (istream &cin, TArray &a){
    a.read(cin);
    return cin;
}
ostream& operator <<(ostream &cout,TArray &a){
    a.write(cout);
    return cout;
}
```

```
int main(){  
  
    ifstream cin("001in.txt");  
    ofstream cout("001out.txt");  
    setlocale(LC_ALL, "Russian");  
  
    TArray A;  
    cin>>A;  
    cout<<A;  
  
    return 0;  
}
```

```
1 10  
2 1 2 3 4 5 6 7 8 9 10
```

```
1 |1 2 3 4 5 6 7 8 9 10
```

Функции внутри структур

```
#include <iostream>
#include <fstream>
#include <cmath>
#include <iomanip>

using namespace std;
struct Time{
    int h,m,s;
    void read (istream &cin){
        cin>>h>>m>>s;
    }
    void write(ostream &cout){
        cout<<h/10<<h%10<<":"<<
            <<m/10<<m%10<<":"<<
            <<s/10<<s%10<<endl;
    }
};

int main(int argc, char const *argv[])
{
    ifstream cin("003in.txt");
    ofstream cout("005out.txt");
    Time t;
    t.read(cin);
    t.write(cout);
    return 0;
}
```

```
1 8 1 12
```

```
1 08:01:12
2
```

```

struct Time{
    int h,m,s;
    void read (istream &cin){
        cin>>h>>m>>s;
    }
    void write(ostream &cout){
        cout<<h/10<<h%10<<":"<<
            <<m/10<<m%10<<":"<<
            <<s/10<<s%10<<endl;
    }
    void correct(){
        m+=s/60;
        s%=60;
        h+=m/60;
        m%=60;
        h%=24;
    }
    bool iscorrect(){
        return h>=0&&h<24&&m>=0&&m<60&&s>=0&&s<60;
    }
};

```

```

int main(int argc, char const *argv[])
{
    ifstream cin("003in.txt");
    ofstream cout("005out.txt");
    Time t;
    t.read(cin);
    if (!t.iscorrect()) cout<<"incorrect input \n";
    t.correct();
    t.write(cout);
    return 0;
}

```

```

struct Time{
    int h,m,s,err;
    void read (istream &cin){
        cin>>h>>m>>s;
        err=!incorrect();
        correct();
    }
    void write(ostream &cout){
        cout<<h/10<<h%10<<":"<<
            <<m/10<<m%10<<":"<<
            <<s/10<<s%10<<endl;
    }
    void writeerror(ostream &cout){
        if (err==1)cout<<"incorrect input \n";
    }
    void correct(){
        m+=s/60;
        s%=60;
        h+=m/60;
        m%=60;
        h%=24;
    }
    bool incorrect(){
        return h>=0&&h<24&&m>=0&&m<60&&s>=0&&s<60;
    }
};

```

```

int main(int argc, char const *argv[])
{
    ifstream cin("003in.txt");
    ofstream cout("005out.txt");
    Time t;
    t.read(cin);
    t.writeerror(cout);
    t.write(cout);
    return 0;
}

```

```

1 |incorrect input
2 | 02:31:12
3 |

```



```

using namespace std;
struct Time{
    int h,m,s,err;
    Time& read(istream &cin){
        cin>>h>>m>>s;
        err=!incorrect();
        correct();
        return *this;
    }
    Time& write(ostream &cout){
        cout<<h/10<<h%10<<":"<<
            <<m/10<<m%10<<":"<<
            <<s/10<<s%10<<endl;
        return *this;
    }
    Time& writeerror(ostream &cout){
        if (err==1)cout<<"incorrect input \n";
        return *this;
    }
    void correct(){
        m+=s/60;
        s%=60;
        h+=m/60;
        m%=60;
        h%=24;
    }
    bool incorrect(){
        return h>=0&&h<24&&m>=0&&m<60&&s>=0&&s<60;
    }
};

```

```

int main(int argc, char const *argv[])
{
    ifstream cin("003in.txt");
    ofstream cout("005out.txt");
    Time t;
    t.read(cin)
    .writeerror(cout)
    .write(cout);
    return 0;
}

```

```

1 |incorrect input
2 | 02:31:12
3 |

```

Предоставление массива в виде структуры

```
struct TArray{
    TElement A[100];
    int N;

    TArray& read (istream &cin){
        cin>>N;
        for (int i=0;i<N;i++){
            cin >>A[i];
        }
        return *this;
    }
    TArray& write(ostream &cout){
        for (int i=0;i<N;i++){
            cout<<A[i];
        }
        return *this;
    }
};

istream& operator >> (istream &cin, TArray &a){
    a.read(cin);
    return cin;
}
ostream& operator <<(ostream &cout,TArray &a){
    a.write(cout);
    return cout;
}
```

```
int main(){  
  
    ifstream cin("001in.txt");  
    ofstream cout("001out.txt");  
    setlocale(LC_ALL,"Russian");  
  
    TArray A;  
    cin>>A;  
    cout<<A;  
  
    return 0;  
}
```

```
1 10  
2 1 2 3 4 5 6 7 8 9 10
```

```
1 |1 2 3 4 5 6 7 8 9 10
```

Поиск экстремумов

```
struct TArray{
    TElement A[100];
    int N,iMax,iMin;

    TArray& read (istream &cin){
        cin>>N;
        for (int i=0;i<N;i++){
            cin >>A[i];
        }
        findE();
        return *this;
    }
,
void findE(){
    iMin=0;
    iMax=0;
    for (int i=0;i<N;i++){
        if (A[i]<A[iMin]) iMin=i;
        if (A[i]>A[iMax]) iMax=i;
    }
}
```

```
int main(){  
  
    ifstream cin("001in.txt");  
    ofstream cout("001out.txt");  
    setlocale(LC_ALL,"Russian");  
  
    TArray A;  
    TElement K;  
    cin>>A;  
    cin>>K;  
    cout<<A<<endl;;  
    cout<<A.A[A.iMax]<<" "<<A.iMax<<endl;  
    cout<<A.A[A.iMin]<<" "<<A.iMin<<endl;  
  
    return 0;  
}
```

```
1 2 3 4 5 6 7 8 9 10  
10 9  
1 0
```

Заготовка под структуру

```
#include <iostream>
#include <fstream>

using namespace std;

struct TElement{
    int data,k;
    TElement& read(istream&); //метод чтения
    TElement& write(ostream&);
    bool operator == (TElement); // оператор сравнения
    bool operator != (TElement);
    bool operator > (TElement);
    bool operator >= (TElement);
    bool operator < (TElement);
    bool operator <= (TElement);
};

istream& operator >> (istream&, TElement); //оператор чтения
ostream& operator << (ostream&, TElement); //вне ибо нужен поток ввода, а не объект класса

int main(int argc, char const *argv[])
{
    ifstream cin("input.txt");
    ofstream cout("output.txt");

    int N;

    return 0;
}
```

Заготовка под описание методов

```
TElement& TElement::read (istream& cin){
}
TElement& TElement::write (ostream& cin){
}
bool TElement::operator == (TElement b){
}
bool TElement::operator != (TElement b){
}
bool TElement::operator > (TElement b){
}
bool TElement::operator >= (TElement b){
}
bool TElement::operator < (TElement b){
}
bool TElement::operator <= (TElement b){
}
istream& operator >> (istream &cin, TElement a){
}
ostream& operator << (ostream &cout, TElement a){
}
|
```

Метод вычисления ключа

```
struct TElement{
    int data,k;
    void calcK();// вычисление ключа
    TElement& read(istream&); //метод чтения
    TElement& write(ostream&);
    bool operator == (TElement);// оператор сравнения
    bool operator != (TElement);
    bool operator > (TElement);
    bool operator >= (TElement);
    bool operator < (TElement);
    bool operator <= (TElement);
};
```

```
void TElement::calcK(){
    k=0;
    for (int t=abs(data);t;t/=10){
        k+=t%10;
    }
}
```


Заполненные методы ввода и вывода

```
TElement& TElement::read (istream& cin){ //TElement – имя класса cin имя переменной (метода)
    cin >> data;
    calcK();
}
TElement& TElement::write (ostream& cout){
    cout <<data<<" ";
}
istream& operator >> (istream &cin, TElement a){
    a.read(cin);
}
ostream& operator << (ostream &cout, TElement a){
    a.write(cout);
}
```

Перегрузка условных операторов

```
bool TElement::operator == (TElement b){
    return k==b.k;
}
bool TElement::operator != (TElement b){
    return !(*this==b);
}
bool TElement::operator > (TElement b){
    return k>b.k;
}
bool TElement::operator >= (TElement b){
    return *this>b || *this==b;
}
bool TElement::operator < (TElement b){
    return b>*this;
}
bool TElement::operator <= (TElement b){
    return b>=*this;
}
```

```
TElement& TElement::read (istream& cin){ //TElement - имя класса cin имя переменной (метода)
    cin >> data;
    calck();
    return *this;
}
TElement& TElement::write (ostream& cin){
    cout <<data<<" ";
    return *this;
}
istream& operator >> (istream &cin, TElement a){ //операторы вызывают метод класса
    a.read(cin);
    return cin;
}
ostream& operator << (ostream &cout, TElement a){
    a.write(cout);
    return cout;
}
```

Если нашли ошибку, то молодцы

```
TElement& TElement::read (istream& cin){ //TElement - имя класса cin имя переменной (метода)
    cin >> data;
    calcK();
    return *this;
}
TElement& TElement::write (ostream& cout){
    cout <<data<<" ";
    cout <<k;
    return *this;
}
```

```
1 10
2 1 2 4 17 19 2 6 7 11 12 22 43
```

```
1 10 1
```

```
istream& operator >> (istream&, TElement&); //оператор чтения  
ostream& operator << (ostream&, TElement&); //вне ибо нужен поток ввода, а не объект класса
```

```
}  
istream& operator >> (istream &cin, TElement &a){ //операторы вызывают метод класса  
    a.read(cin);  
    return cin;  
}  
ostream& operator << (ostream &cout, TElement &a){  
    a.write(cout);  
    return cout;  
}
```

Массив структур

```
int main(int argc, char const *argv[])
{
    ifstream cin("input_s.txt");
    ofstream cout("output.txt");

    int N;
    TElement arr[100];
    cin>>N;
    for (int i=0;i<N;i++){
        cin>>arr[i];
    }
    for (int i=0;i<N;i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
    for (int i=0;i<N-1;i++){
        if (arr[i]>arr[i+1]){
            swap(arr[i],arr[i+1]);
            i=i-2*(i>0);
        }
    }
    for (int i=0;i<N;i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;

    return 0;
}
```

Расширение условий >

```
bool TElement::operator == (TElement b){ //определен явно
    return k==b.k;
}
bool TElement::operator != (TElement b){
    return !(*this==b);
}
bool TElement::operator > (TElement b){ //определен явно
    return k>b.k && data > b.data || k>b.k;
}
```

Заготовка для массива структур

```
struct TArray{
    TElement arr[100];
    int N;
    TArray& read(istream&);
    TArray& write(ostream&);
    TArray& sort();
};

istream& operator >> (istream&, TArray&); //оператор чтения
ostream& operator << (ostream&, TArray&);
```


Описание

```
TArray& TArray::read(istream& cin){
    cin>>N;
    for (int i=0;i<N;i++){
        cin>>arr[i];
    }
    return *this;
}

TArray& TArray::write(ostream& cout){
    for (int i=0;i<N;i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
    return *this;
}

TArray& TArray::sort(){
    for (int i=0;i<N-1;i++){
        if (arr[i]>arr[i+1]){
            swap(arr[i],arr[i+1]);
            i=i-2*(i>0);
        }
    }
    return *this;
}

istream& operator >> (istream &cin, TArray &a){
    a.read(cin);
    return cin;
}

ostream& operator << (ostream &cout, TArray &a){
    a.write(cout);
    return cout;
}
```

Использование

```
int main(int argc, char const *argv[])
{
    ifstream cin("input_s.txt");
    ofstream cout("output.txt");

    TArray A,B;
    A.read(cin).sort().write(cout);
    cin>>B;
    B.sort();
    cout<<B;

    return 0;
}
```

Доступ к свойствам и методам

- * **Public** — доступ открыт всем, кто видит определение данного класса.
- * **Private** — доступ открыт самому классу (т.е. функциям-членам данного класса) и друзьям (friend) данного класса, как функциям, так и классам.
- * **Protected** — доступ открыт классам, производным от данного.
- *

Исходный модификатор доступа

	public	private	protected
public-наследование	public	private	protected
private-наследование	private	private	private
protected-наследование	protected	private	protected

```
struct TElement{  
    private:  
        int data,k;  
        void calcK();// вычисление ключа  
    public:  
        TElement& read(istream&); //метод чтения  
        TElement& write(ostream&);  
        bool operator == (TElement); // оператор сравнения  
        bool operator != (TElement);  
        bool operator > (TElement);  
        bool operator >= (TElement);  
        bool operator < (TElement);  
        bool operator <= (TElement);  
};
```

```
int main(int argc, char const *argv[])  
{  
    ifstream cin("input_s.txt");  
    ofstream cout("output.txt");  
  
    TElement a;  
    cin >>a;  
  
    return 0;  
}
```

```
struct TElement{
    private:
        int data,k;
        void calcK();// вычисление ключа
    public:
        TElement& read(istream&); //метод чтения
        TElement& write(ostream&);
        TElement& setData(int); //методы получения и задания параметров
        int getK();
        bool operator == (TElement);// оператор сравнения
        bool operator != (TElement);
        bool operator > (TElement);
        bool operator >= (TElement);
        bool operator < (TElement);
        bool operator <= (TElement);
};
```

```
int main(int argc, char const *argv[])
{
    ifstream cin("input_s.txt");
    ofstream cout("output.txt");

    TElement a;
    a.setData(1111);
    cout<<a<<" "<<a.getK();

    return 0;
}
```

```
TElement& TElement::setData(int _data){
    data=_data; //изменение данных с перерасчетом ключа
    calcK();
    return *this;
}
int TElement::getK(){
    return K;
}
```

Class

```
class TElement{  
    int data,k;  
    void calcK();// вычисление ключа  
public:  
    TElement& read(istream&); //метод чтения  
    TElement& write(ostream&);  
    TElement& setData(int); //методы получения и задания параметров  
    int getK();  
    bool operator == (TElement); // оператор сравнения  
    bool operator != (TElement);  
    bool operator > (TElement);  
    bool operator >= (TElement);  
    bool operator < (TElement);  
    bool operator <= (TElement);  
};
```


Использование <vector>

```
#include <iostream>
#include <vector>
#include <fstream>

using namespace std;

int main(int argc, char const *argv[])
{
    ifstream cin("input_s.txt");
    ofstream cout("output.txt");
    vector <int> myVector (10);
    for (int i=0;i<myVector.size();i++){
        cout<<myVector[i]<<' ';
    }
    return 0;
}
```



1 0 0 0 0 0 0 0 0 0

Простое выделение памяти

```
#include <iostream>
#include <vector>
#include <fstream>

using namespace std;

int main(int argc, char const *argv[])
{
    ifstream cin("input_s.txt");
    ofstream cout("output.txt");
    vector<int> myVector;
    myVector.reserve(10);
    for (int i=0;i<myVector.size();i++){
        cout<<myVector[i]<<' ';
    }
    return 0;
}
```

1

Копирование вектора

```
#include <iostream>
#include <vector>
#include <fstream>

using namespace std;

int main(int argc, char const *argv[])
{
    ifstream cin("input.txt");
    ofstream cout("output.txt");
    vector<int> myVector1 (10);

    for (int i=0; i<myVector1.size(); i++){
        cin>>myVector1[i];
    }
    for (int i=0; i<myVector1.size(); i++){
        cout<<myVector1[i]<<' ';
    }
    cout<<endl;
    vector<int> myVector2(myVector1);
    for (int i=0; i<myVector2.size(); i++){
        cout<<myVector2[i]<<' ';
    }
    return 0;
}
```

```
1 1 2 4 17 19 2 6 7 11 12 22 43
```

```
1 1 2 4 17 19 2 6 7 11 12
2 1 2 4 17 19 2 6 7 11 12
```