

Врсте датотека

Као извор података може се користити у/и уређај (тастатура, штампач) као и датотека која се налази на неком од уређаја спољне меморије.

У зависности како се складиште датотеке, оне могу бити текстуалне и бинарне.

Текстуалне датотеке се састоје од низа знакова који у себи садрже и специјалне знакове које се не штампају (ескејп карактери). За ове датотеке се мора користити конверзија када се подаци смештају у меморију или се читају из ње.

Бинарне датотеке се састоје из низа бајтова и садржај им је исти када се смештају у меморију. Оне су практичне за смештај бројчаних података јер заузимају мање меморије.

Према организацији датотеке се деле на :

1. Секвенцијалне – под. у њима се може прићи само по редоследу којим су подаци и уписани у датотеку (секвенцијалан приступ)
2. Релативне – под. у њима се може приступити на основу редног броја записа у датотеци (директан приступ)
3. Индексне – омогућавају произвољан приступ и то преко дела записа који се третира као кључ тога записа (приступ помоћу кључа)

У С++ постоје само секвенцијалне датотеке где се датотеке сматрају као дугачки низови знакова. Никада се аутоматски не умеће знак за нови ред, нити функције за улаз њега региструју. То значи да је могуће написати један ред у датотеци у више редова или више редова као један ред.

Рад са улазима и излазима датотеке се регулише кроз стандардне библиотеке и није део стандардног С++ језика.

Коришћењем `iostream` хедера могу се користити велики број класа помоћу којих се обављају у/и операције.

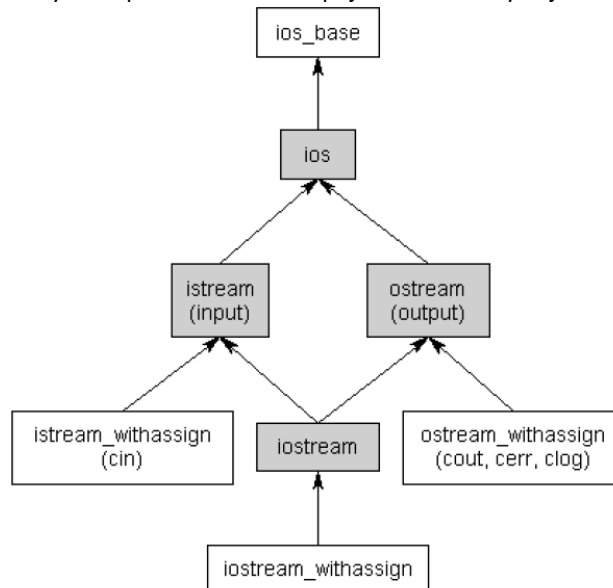


График који показује зависност функција у библиотекама

Секвенцијални приступ датотекама значи да постоје два различита типа приступа: улазне и излазне секвенце.

Улазне секвенце постоје у класи `istream` и за приступ таквим подацима се користи оператор `>>` (extraction operator).

Излазне секвенце постоје у класи `ostream` и за приступ таквим подацима се користи оператор `<<` (insertion operator).

Класа `iostream` омогућује рад и са улазним и са излазним секвенцама.

У раду са датотекама се користе посебне класе:

- `ofstream` – изведена из класе `ostream`, која омогућава упис у датотеке
- `ifstream` – изведена из класе `istream`, која омогућава читање из датотеке
- `fstream` – изведена из класе `iostream`, која омогућава упис и читање у/из датотеке

Радње при раду са датотекама:

- отварање датотеке – успостављање везе програма са датотеком
- приступ датотеци – уписивање или читање података
- испитивање стања датотеке – утврђивање да ли она уопште постоји на одређеном месту у меморији
- затварање датотеке – прекидање везе између датотеке и програма

Све потребне рутине за рад са датотекама с еналазе у функцији `fstream`.

Отварање датотека

Да би се програм повезао са постојећом датотеком, прво је потребно отворити датотеку и теме омогућити приступ програма датотеци: `open(filename, mode);`

где је **filename** назив реалне датотеке а **mode** опциони параметар отварања датотеке.

Као опциони параметар, могу се користити:

ios::in Улазне операције
ios::out Излазне операције
ios::binary Рад са бинарним датотекама
ios::ate Отвара датотеку и поставља почетну позицију на крај датотеке
ios::app Све операције се додају на крају и при томе се чувају сви претходни подаци из датотеке
ios::trunc Ако је датотека отворена за излазне операције њен садржај се брише пре уписа нових података и при томе се може користити bitwise оператор или („|“).

Пример : Ако треба отворити бинарну датотеку за додавање садржаја, може се користити опциони параметар:

```
ios::out | ios::app | ios::binary
```

Свака функција `open` класа `ofstream`, `ifstream`, `fstream` има подразумевани режим који се користи ако се датотека отвори без навођења другог аргумента: `ofstream (ios::out)`, `ifstream (ios::in)`, `fstream (ios::in | ios::out)`.

Затварање датотека

Када се обаве све потребне операције над датотекама, датотека се мора затворити тако да оперативни систем буде обавештен да су употребљени ресурси сада слободни : `close()`

После коришћења ове наредбе, објекат који је користио датотеку се може поново користити.

Ако датотека није затворена, или ће се аутоматски затворити или ће пријавити грешку.

Пример: У датотеку `primer.txt` уписати стринг.

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    ofstream mojFajl ("primer.txt");
    if (mojFajl.is_open())
    {
        mojFajl << "recenica";
        mojFajl.close();
    }
    else cout << "Datoteka se ne moze otvoriti." << endl;
    return 0;
}
```

Пример: Из датотеке `primer.txt` прочитати стринг.

```
#include<iostream>
#include<fstream>
#include<string>
using namespace std;
int main()
{
    string S;
    ifstream mojFajl ("primer.txt");
    if (mojFajl.is_open())
    {
        mojFajl >> S;
        mojFajl.close();
    }
    else cout << "Datoteka se ne moze otvoriti." << endl;
    cout << S << endl;
    return 0;}
}
```

Функције за рад са датотекама

1. open() – отвара датотеку чије је име наведено као аргумент за читање или унос података

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    ofstream fajl;
    fajl.open("test.txt");
    fajl << "recenica";
    fajl.close();
    return 0;
}
```

2. is_open() – проверава да ли је датотека отворена или није отворена

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    ofstream fajl ("test.txt");
    if (fajl.is_open())
    {
        fajl << "tekst" << endl;
        fajl.close();
        cout << "Podaci su upisani u datoteku." << endl;
    }
    else cout << "Datoteka se ne moze otvoriti." << endl;
    return 0;
}
```

3. get() – чита из датотеке знакове

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    char a[10], b;
    ifstream fajl("test.txt");
    if (fajl.is_open())
    {
        b = fajl.get(); // cita jedan znak
        fajl.get(a, 10); // cita 10 znakova
        fajl.close();
        cout << b << endl;
        cout << a << endl;
    }
    else cout << "Datoteka se ne moze otvoriti." << endl;
    return 0;
}
```

4. getline() – чита задати број знакова из датотеке

```
// primer citanja zadatog broja znakova iz datoteke
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    char a[100];
```

```

ifstream fajl("test.txt");
if (fajl.is_open())
{
    fajl.getline(a, 100); // cita 100 znakova
    fajl.close();
    cout << a << endl;
}
else cout << "Datoteka se ne moze otvoriti." << endl;
return 0;
}

```

Текстуалне датотеке

5. eof() – враћа вредност true ако се читањем дошло до краја датотеке

```

// primer rada sa eof
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    char x;
    ifstream fajl("test.txt");
    if (fajl.is_open())
    {
        while (!fajl.eof())
        {
            x = fajl.get();
            cout << x;
        }
        fajl.close();
    }
    else cout << "Datoteka se ne moze otvoriti." << endl;
    return 0;
}

```

На излазу се исписује садржај датотеке ако је она присутна у директоријуму са .cpp кодом.

6. precision() – поставља број цифара са приказ реалних бројева

```

// primer rada sa precision
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    char x;
    ofstream fajl("primer.txt");
    if (fajl.is_open())
    {
        fajl.precision(3);
        fajl << 3.14259 << endl;
        fajl.close();
    }
    else cout << "Datoteka se ne moze otvoriti." << endl;
    return 0;
}

```

Излази фајл садржи 3.14.

7. width() – поставља најамњу ширину поља за упис података

```

// primer rada sa width
#include<iostream>
#include<fstream>

```

```

using namespace std;
int main()
{
    char x;
    ofstream fajl("primer.txt");
    if (fajl.is_open())
    {
        fajl.width(10);
        fajl << left << "Pi = ";
        fajl.width(8);
        fajl << right << 3.14259;
        fajl.close();
    }
    else cout << "Datoteka se ne moze otvoriti." << endl;
    return 0;
}
Pi = 3.14259

```

Текстуалне датотеке

Пример: У датотеку programiranje.txt уписати презимена, имена и оцене из предмета програмирање ученика једног одељења.

// u datoteku "programiranje.txt" upisati prezimena, imena i ocene ucenika
// jednog odeljenja iz predmeta programiranje

```

#include<iostream>
#include<fstream>
#include<string>
using namespace std;
int main()
{
    struct ucenik
    {
        string ime, prezime;
        int ocena;
    };
    ucenik x[40];
    int n = 0, i;
    cout << "Broj ucenika u odeljenju: ";
    cin >> n;
    for (i = 0; i < n; i++)
    {
        cout << "Prezime ucenika: ";
        cin >> x[i].prezime;
        cout << "Ime ucenika: ";
        cin >> x[i].ime;
        cout << "Ocena ucenika iz predmeta programiranje: ";
        cin >> x[i].ocena;
    }
    ofstream fajl("programiranje.txt");
    if (fajl.is_open())
    {
        for (i = 0; i < n; i++)
        {
            fajl.width(15);
            fajl << left << x[i].prezime;
            fajl.width(10);
            fajl << x[i].ime;
            fajl.width(2);

```

```

Broj ucenika u odeljenju: 3
Prezime ucenika: Micic
Ime ucenika: Mica
Ocena ucenika iz predmeta programiranje: 3
Prezime ucenika: Peric
Ime ucenika: Pera
Ocena ucenika iz predmeta programiranje: 2
Prezime ucenika: Dragovic
Ime ucenika: Drago
Ocena ucenika iz predmeta programiranje: 5
Podaci su upisani u datoteku.
Press any key to continue . . .

```

Micic	Mica	3
Peric	Pera	2
Dragovic	Drago	5

```

        fajl << right << x[i].ocena << endl;
    }
    fajl.close();
    cout << "Podaci su upisani u datoteku." << endl;
}
else cout << "Datoteka se ne moze otvoriti." << endl;
return 0;}

```

Пример: Из датотеке programiranje.txt учитати презимена, имена и оцене из предмета програмирање ученика једног одељења.

```

// iz datoteke "programiranje.txt" ucitati prezimena, imena i ocene ucenika
// jednog odeljenja iz predmeta programiranje
#include<iostream>
#include<fstream>
#include<string>
using namespace std;
int main()
{
    struct ucenik
    {
        string ime, prezime;
        int ocena;
    };
    ucenik x[40];
    int i = 0;
    ifstream fajl("programiranje.txt");
    if (fajl.is_open())
    {
        while (!fajl.eof())
        {
            fajl >> x[i].prezime >> x[i].ime >> x[i].ocena;
            i++;}

            i--;
            for (int j = 0; j < i; j++)
            {
                cout.width(15);
                cout << left << x[j].prezime;
                cout.width(10);
                cout << x[j].ime;
                cout.width(2);
                cout << right << x[j].ocena << endl;
            }
            fajl.close();
        }
        else cout << "Datoteka se ne moze otvoriti." << endl;
        return 0;}

```

Директан приступ датотекама

Пример: Из датотеке programiranje.txt учитати презимена, имена и оцене из предмета програмирање ученика једног одељења и одредити колико има недовољних оцена и приказати ученике са недовољним оценама.

```

// iz datoteke "programiranje.txt" ucitati prezimena, imena i ocene ucenika
// jednog odeljenja iz predmeta programiranje i odrediti koliko ima
// nedovoljnihocena i prikazati ucenike sa nedovoljnim ocenama
#include<iostream>
#include<fstream>
#include<string>
using namespace std;
int main()

```

```

{
    struct ucenik
    {
        string ime, prezime;
        int ocena;
    };
    ucenik x[40];
    int i = 0;
    ifstream fajl("programiranje.txt");
    if (fajl.is_open())
    {
        while (!fajl.eof())
        {
            fajl >> x[i].prezime >> x[i].ime >> x[i].ocena;
            i++;
        }
        i--;
        int n = 0;
        cout << "Ucenici sa nedovoljnim ocenama:" << endl;
        for (int j = 0; j < i; j++)
        {
            if (x[j].ocena == 1)
            {
                cout.width(15);
                cout << left << x[j].prezime;
                cout.width(10);
                cout << x[j].ime;
                cout.width(2);
                cout << right << x[j].ocena << endl;
                n++;
            }
        }
        cout << endl;
        cout << "Broj ucenika sa nedovoljnim ocenama: " << n << endl;
        fajl.close();
    }
    else cout << "Datoteka se ne moze otvoriti." << endl;
    return 0;
}

```

```

Micic      Mica      3
Peric      Pera      2
Dragovic   Drago     5
Jovicic    Jovan     1
Dejanovic  Dejan     4
Anovic     Ana       1
Gejts     Bil       1

```

```

Ucenici sa nedovoljnim ocenama:
Jovicic    Jovan     1
Anovic     Ana       1
Gejts     Bil       1

```

```

Broj ucenika sa nedovoljnim ocenama: 3
Press any key to continue . . .

```

Датотеке

Пример: Унети два броја и сабрати их, затим њихове вредности и резултат уписати у датотеку чије име задаје корисник.

```

// uneti dva broja, sabrati ih i te vrednosti upisati u datoteku
// cije ime zadaje korisnik
#include<iostream>
#include<fstream>
#include<string>
using namespace std;

```

```

int main()
{
    string f;
    int a, b, c;
    cout << "Unesi dva cela broja: ";
    cin >> a >> b;
    c = a + b;
    cout << "Uneti naziv datoteke u koju ce se upisati podaci: " << endl;
    cin >> f;
    ofstream fajl(f.c_str());
    // ofstream prihvata samo nisku znakova pa se naziv datoteke mora
    // konvertovati iz stringa u nisku znakova
    // to se radi sa funkcijom c_str()
    if (fajl.is_open())
    {
        fajl << "a = " << a << endl;
        fajl << "b = " << b << endl;
        fajl << "c = " << c << endl;
        fajl.close();
        cout << "Podaci su upisani u datoteku." << endl;
    }
    else cout << "Datoteka se ne moze otvoriti." << endl;
    return 0;
}

```

```

Unesi dva cela broja: 5 10
Uneti naziv datoteke u koju ce se upisati podaci:
Zbir
Podaci su upisani u datoteku.
Press any key to continue . . . █
U datoteci se ispisiuje:
a = 5
b = 10
c = 15

```

Пример: Из датотеке celibrojevi.txt преписати у датотеку vecioddeset.txt само оне вредности које су веће од 10.

```

// iz datoteke celibrojevi.txt prepisati u datoteku vecioddeset.txt
// samo one vrednosti koje su vece od 10
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    ifstream ulaz("celibrojevi.txt");
    ofstream izlaz("vecioddeset.txt");
    int a, b;
    if (ulaz.is_open())
    {
        while (!ulaz.eof())
        {
            ulaz >> a;
            if (a > 10)
            {
                izlaz << a << " ";
            }
        }
        izlaz.close();
        ulaz.close();
    }
}

```



```
    else cout << "Datoteka se ne moze otvoriti." << endl;  
    return 0;  
}  
У датотеци celibrojevi.txt је уписано: 1 5 6 11 12 5  
У датотеци vecioddeset.txt ће бити уписано: 11 12
```