

**Obyektyönlü
programlaşdırma**

C++



Dərs №10

C++ dili ilə
obyektyönlü
proqramlaşdırma

Mündəricat

Praktik nümunə.

Obyekt strukturların faylda saxlanması.....	4
Fayllarla işləmək üçün funksiyalar dəsti	6
<i>stdio.h</i> — faylın adının dəyişdirilməsi	
və faylın silinməsi	6
<i>io.h</i> — faylın axtarılması	7
Fayllarla işləmək üçün proqram nümunəsi	9
Kataloqlarla işləmək üçün əməliyyatlar.....	13
Funksiyalar dəsti.....	13
Kataloqlarla işləmək üçün nümunə	14
Praktik nümunə.	
Kataloqun məzmununun göstərilməsi.....	16
Əmrlər sətrinin argumentlərinin istifadəsi.....	23

Birinci üsul.

Visual Studio vasitəsi ilə ötürmə 23

İkinci üsul.

“Start” menyusunun “Run” əmri

vasitəsi ilə ötürmə..... 24

Üçüncü üsul.

Konsol vasitəsi ilə ötürmə (cmd.exe)..... 25

Praktik nümunə 26

Ev tapşırığı..... 28

Praktik nümunə. Obyekt strukturların faylda saxlanması

Başlanğıcda biz sizinlə keçilmiş materialı məhkəmləndirəcəyik və obyekt strukturun faylda saxlanılmasına baxacağıq. Başlayaq.

```
#include <iostream>
#include <string.h>
#include <stdio.h>
using namespace std;
//insan haqqında informasiya ehtiva edən struktur

struct Man{
    //Ad
    char str[255];
    //Yaş
    int age;
};

void main()
{
    //Struktur obyektlərin yaradılması
    Man A,B;
    //Klaviaturadan daxil edilmiş informasiyanın
    //A obyektinə yazılması
    cout<<"\nEnter name:\n";
    cin>>A.str;
    cout<<"\nEnter age:\n";
    cin>>A.age;
```

```
// faylın yazmaq üçün açılması
FILE*f=fopen("Test.txt","w+");
if(!f) exit(0);
//A obyektinin fayla yazılması
fwrite(&A,sizeof(Man),1,f);
c
fclose(f);

//Faylın oxumaq üçün açılması
f=fopen("Test.txt","r+");
if(!f) exit(0);

//Faylın məzmununun B obyektinə oxunması
fread(&B,sizeof(Man),1,f);
//Faylın yazmaq üçün açılması
fclose(f);

//nəticənin ekranda göstərilməsi
cout<<"\nName - "<<B.str<<"\n\nAge - "<<B.age<<"\n\n";
}
```

Fayllarla işləmək üçün əməliyyatlar

Biz sizinlə artıq faylın məzmununu analiz etməyi öyrənmişik. İndi isə fayllar üzərində əməliyyatlar aparmağı öyrənməyin vaxtı çatmışdır. Yəni, indi biz, faylın adının dəyişdirilməsi, yerinin dəyişdirilməsi, silinməsi və axtarılması kimi əməliyyatlara baxacağıq.

Fayllarla işləmək üçün funksiyalar dəsti

stdio.h — faylların adının dəyişdirilməsi və silinməsi

```
rename (char * oldname, char * newname)
```

Faylın adını dəyişdirir.

oldname — faylın yolu və cari adı.

newname — faylın yolu və yeni adı.

Əgər faylın adı müvəffəqiyyətlə dəyişdirilmişsə, 0 qaytarır, əgər dəyişmə baş vermədisə, sıfır olmayan qiymət qaytarır.

Qeyd: Yeri gəlmişkən!!! Əgər faylın yalnız yeni adı deyil, həmçinin, yeni yolu da verilərsə, faylın adı nəinki dəyişdiriləcək o həmçinin uyğun yerə yerini dəyişəcək.

```
int remove(const char * filename)
```

Faylı silir.

filename — faylın yolu və adı.

Fayl müvəffəqiyyətlə silinəcəksə, funksiya 0 qaytaracaq, silmə baş verməyəcəksə, sıfır olmayan qiymət qaytarılacaq.

Qeyd: Yadda saxlayın!!! Silmə zamanı fayl bağlanmış olmalıdır. Bundan başqa, yaddan çıxarmayın ki, silinmə qaçılmazdır.

io.h — faylların axtarılması

```
_findfirst(char * path, _finddata_t * fileinfo)
```

Göstərilən yola, uyğun maskaya görə faylları tapır.

Qeyd: Maska — * (ixtiyari simvolların ixtiyari ardıcılığı) və ya ? (ixtiyari bir simvol) simvolları ehtiva edən axtarış kriteriyaları sətridir.

path — yolların və maskaların kombinasiyasını ehtiva edən sətir.

fileinfo — tapılmış fayl haqqında informasiyanın yazılacağı obyekt strukturun göstəricisi. Struktur növbəti sahələri ehtiva edir:

1. **unsigned attrib** — faylın atributları haqqında informasiya ehtiva edir.
 - ■ **_A_NORMAL** — Yazma və ya oxuma qadağası olmayan adi fayl.
 - ■ **_A_RDONLY** — yalnız oxumaq üçün fayl.
 - ■ **_A_HIDDEN** — Gizli fayl.
 - ■ **_A_SYSTEM** — Sistem faylı.
 - ■ **_A_ARCH** — Arxiv faylı.
2. **time_t time_create** — faylın yaradılma zamanı/tarixi (FAT sistemi üçün 1-ə bərabərdir).

3. **time_t time_access** — faylın sonuncu açılışı zamanı/tarixi (FAT sistemi üçün 1-ə bərabərdir).
4. **time_t time_write** — faylın sonuncu redaktəsi zamanı/tarixi.
5. **_fsize_t size** — faylın ölçüsü.
6. **char name[260]** — faylın adı.

Əgər hər şey müvəffəqiyyətlə yerinə yetirilərsə, ilk tapılmış fayl haqqında informasiya `_finddata_t` strukturunun obyektinə yazılacaq. Bu zaman başlanğıcda ilk tapılmış faylın daxili göstəricisini əldə edən siyahı formalaşdırılacaq. Bu halda funksiya alınan fayllar qrupu ilə bağlı olan unikal nömrə qaytaracaq.

Əgər axtarış uğursuzluqla nəticələnərsə, funksiya -1 qaytaracaq.

```
_findnext(long done, _finddata_t * fileinfo)
```

Funksiya qrupda növbəti tapılmış fayla keçidi həyata keçirir.

done — Fayllar qrupunun yaddaşda unikal nömrəsi.

fileinfo — növbəti tapılmış fayl haqqında informasiyanın yazılacağı obyekt strukturun göstəricisi.

Fayllar siyahısının sonuna gəlinərsə, funksiya -1 qaytaracaq.

```
_findclose(long done)
```

Funksiya yaddaşı tapılmış fayllar qrupundan təmizləyir.

done — Fayllar qrupunun yaddaşdakı unikal nömrəsi.

Fayllarla işləmək üçün program nümunəsi

```
//Burada ad dəyişdirmə və silmə funksiyaları var
#include <stdio.h>
//Burada faylların axtarılması funksiyası var
#include <io.h>
#include <string.h>
#include <iostream>
using namespace std;

//Mövcud faylın adının dəyişdirilməsi
void RenameFile();

//Mövcud faylın silinməsi
void RemoveFile();

//Faylların kataloqda axtarılması
void Dir();

void main()
{
    // İcra etmək üçün menyu bölməsinin seçilməsini
    //təklif edirik
    cout << "Please, select preffer number...\n";

    //istifadəçinin seçilməsi

    char ch;
    do{
        //Adın dəyişdirilməsi
        cout << "\n1 - Rename\n";
        //Silmə
        cout << "2 - Remove\n";
        //Bəzi kataloqlara baxmaq
        cout << "3 - View some folder\n";
        //Çıxış
        cout << "0 - Exit\n\n";
        cin >> ch;
```

```

//analiz edirik və uyğun
//funksiyanı çağırırıq
switch(ch)
{
    case '1':
        RenameFile();
        break;
    case '2':
        RemoveFile();
        break;
    case '3':
        Dir();
        break;
}
} while(ch != '0'); //Proqramdan çıxış
}

//Mövcud faylın adının dəyişdirilməsi
void RenameFile()
{
    char oldName[50], newName[50];

    //Mövcud adı bir dəyişəndə saxlayırıq
    //(oldName),
    cout << "Enter old name:";
    cin >> oldName;

    //Digərində isə yeni adı(newName)
    cout << "Enter new name:";
    cin >> newName;
    //Ad dəyişdirməni və nəticənin yoxlanılmasını
    //həyata keçiririk
    if (rename(oldName, newName) != 0)
        cout << "Error!\n Couldn't rename file.
            Check old and new filename...\n\n";
    else

```

```

        cout << "Ok...\n\n";
    }

    //Mövcud faylı silmək
    void RemoveFile()
    {
        char Name[50];
        //Silinəcək faylın adını və yolunu əldə
        //edirik
        cout << "Enter name:"; cin >> Name;

        //Faylı silirik və nəticəni
        //yoxlayırıq
        if (remove(Name) != 0)
            cout << "Error!\n Couldn't remove file.
                Check filename...\n";
        else
            cout << "Ok...\n" ;
    }

    //Fayl və kataloqların axtarılması
    void Dir()
    {
        //YOLU soruşuruq (məsələn, C diskindəki Temp qovluğu)
        //növbəti şəkildə yazılacaq: c:\temp\
        char path[70];
        cout << "\nEnter full path (for example,
            C:\\):"; cin >> path;

        //Fayllar maskasının soruşulması
        char mask[15];
        cout << "\nEnter mask (for example, *.* or
            *.txt):"; cin >> mask;

        //İki sətiri birləşdirməklə biz nəticə alırıq
        //yəni, istifadəçi nəyi və harda tapmaq
        //isteyir
        strcat(path, mask);

```

```

//_finddata_t strukturunun fileinfo
//göstəricisinin təyin edilməsi
//_finddata_t strukturunun
//dinamik obyektinin yaradılması
_finddata_t *fileinfo=new _finddata_t;

//Axtarışa başlayırıq
long done = _findfirst(path,fileinfo);

//əgər done -1-ə bərabər olarsa,
//onda axtarışın aparılması mənasızdır
int MayWeWork = done;

//Sayğac tapılmış faylların sayı haqqında
//informasiya saxlayır
int count = 0;
while (MayWeWork!=-1)
{
    count++;
    //tapılmış faylın adını çap edirik
    cout << fileinfo->name << "\n\n";
    //Qrupda növbəti faylı axtarmağa cəhd edirik
    MayWeWork = _findnext(done, fileinfo);
}
//Tapılmış faylların sayı haqqında məlumatın çap
//edilməsi
cout << "\nInformation: was found " << count; cout
<< " file(s) in folder..." << path << "\n\n";

//Yaddaşın təmizlənməsi
_findclose(done);
delete fileinfo;
}

```

Kataloqlarla işləmək üçün əməliyyatlar

Dərsin əvvəlki bölməsində baxılan fayllardan başqa kataloqlar da mövcuddur, elə deyilmi? İndi biz sizinlə kataloqlarla iş üsullarının öyrənilməsi ilə məşğul olacağıq.

Funksiyalar dəsti

direct.h kitabxanası

```
int _mkdir( const char *dirname )
```

Verilmiş yola görə kataloq yaradır.

dirname — yaradılacaq kataloq üçün yol və ad.

```
int _rmdir( const char *dirname )
```

Verilmiş yola görə kataloqu silir.

dirname — Silinəcək kataloq üçün yol və ad.

Səhv baş verərkən, hər iki funksiya -1 qaytarır.

Qeyd: Yeri gəlmişkən!!! Kataloqun adının dəyişdirilməsi üçün stdio.h kitabxanasının rename funksiyası istifadə edilir.

Diqqət!!! Yalnız boş kataloqu silmək və ya adını dəyişdirmək olar!!!

Kataloqlarla işə aid nümunə

```
#include <iostream>
#include <direct.h>
#include <stdio.h>

using namespace std;

//Mövcud kataloqun adının dəyişdirilməsi
void RenameDirectory();
//Mövcud kataloqun silinməsi
void RemoveDirectory();

//Kataloqun yaradılması
void CreateDirectory();

void main()
{
    //İcra etmək üçün menyu bölməsinin seçilməsini
    //təşklif edir
    cout << "Please, select preffer number...\n";
    //istifadəçinin seçilməsi
    char ch;
    do{
        //Ad dəyişdirmək
        cout << "\n1 - Rename\n";
        //Silmək
        cout << "2 - Remove\n";
        //Yaratmaq
        cout << "3 - Create\n";
        //Çıxış
        cout << "0 - Exit\n\n";
        cin >> ch;

        //analiz edirik və uyğun funksiyanı çağırırıq
        switch(ch)
        {
```

```
        case '1':
            RenameDirectory();
            break;
        case '2':
            RemoveDirectory();
            break;
        case '3':
            CreateDirectory();
            break;
        }
    } while(ch != '0'); //Proqramdan çıxış
}

//Mövcud kataloqun adının dəyişdirilməsi
void RenameDirectory()
{
    char oldName[50], newName[50];

    //Bir dəyişəndə mövcud adı (oldName) saxlayırıq
    cout << "Enter old name:";
    cin >> oldName;

    //Digərində isə yeni adı (newName)
    cout << "Enter new name:";
    cin >> newName;

    //AD dəyişdirməni və nəticənin yoxlanılmasını
    //həyata keçiririk
    if (rename(oldName, newName) != 0)
        cout << "Error!\n Couldn't rename directory.\n\n";
    else
        cout << "Ok...\n\n";
}

//Mövcud kataloqun silinməsi
void RemoveDirectory()
{
```



```

char Name[50];
//Silinən kataloqun adını və yolunu əldə
//edirik
cout << "Enter name:"; cin >> Name;

//Kataloqu silirik və nəticəni yoxlayırıq
if (_rmdir(Name) == -1)
    cout << "Error!\n Couldn't remove directory.\n";
else
    cout << "Ok...\n" ;
}
//Kataloq yaratmaq
void CreateDirectory()
{
    char Name[50];
    //Yaradılan kataloqun adını və yolunu əldə
    //edirik
    cout << "Enter name:";
    cin >> Name;
    //Kataloq yaradırıq və nəticəni
    //yoxlayırıq
    if (_mkdir(Name) == -1)
        cout << "Error!\n Couldn't create directory.\n";
    else
        cout << "Ok...\n" ;
}

```

Praktik nümunə.

Kataloqun məzmununun göstərilməsi

Kataloqun məzmununun göstərilməsini həyata keçirən proqram nümunəsi. Proqram yüklənən zaman cari kataloqun məzmununu göstərir, sonra isə istifadəçiyə seçim imkanı verir. Növbəti əməlləri daxil etmək olar:

1. **cd Yol** — başqa kataloqa keçmək.
2. **cd ..** — ana kataloqun məzmununun göstərilməsi və keçid.
3. **cd** və ya **cd .** — cari kataloqun məzmununun göstərilməsi.
4. **exit** — proqramdan çıxış.
5. **root** — ana kataloqa keçid.

Qeyd: Yuxarıda göstərilənlərin hamısı registrasiyalı olmayan əməllərdir.

```

#include <iostream>
#include <windows.h>
#include <io.h>
#include <stdio.h>

using namespace std;

const int size=255;

//sağdakı artıq slash və boşluqları silən funksiya
void RemoveRSpacesAndRSlashes(char *str){
    int index=strlen(str)-1;
    while(str[index]=='\\'||str[index]==' '){
        index--;
    }
    strncpy(str,str,index);
    str[index+1]='\0';
}

//Cari kataloqu göstərən funksiya
void ShowCurrentDir(char path[],char temp[]){
    CharToOem(path,temp);
    printf("%s>",temp);
}

// Windows kodlaşdırmadan DOS kodlaşdırmaya çevirmə
//funksiyası

```

```
//Kril əlifbasının düzgün əks etdirilməsi üçün

void RussianMessage(char path[]){
    CharToOem(path,path);
}

//Kataloqun məzmununun ekranda göstərilməsi
bool ShowDir(char path[]){
    //Cari kataloqun məzmununun
    //göstərilməsi
    _finddata_t find;
    char pathfind[MAX_PATH];
    strcpy(pathfind,path);
    strcat(pathfind,"\\*.");
    char info[MAX_PATH];

    //Axtarışa başlanması
    int result=_findfirst(pathfind,&find);
    //Ekranın təmizlənməsi
    system("cls");
    int flag=result;
    if (flag!=-1) {
        strcpy(info,"Belə kataloq yoxdur");
        RussianMessage(info);
        printf("%s\n",info);
        return false;
    }
    while(flag!=-1){
        if(strcmp(find.name,".")&&strcmp(find.name,"..")){
            //Kataloq olmasının yoxlanılması
            find.attrib&_A_SUBDIR?strcpy(info," Kataloq "):
                strcpy(info," Файл ");
            RussianMessage(info);
            RussianMessage(find.name);
            printf("%30s %10s\n",find.name,info);
        }
    }
}
```

```
//Axtarışı davam etdiririk
flag=_findnext(result,&find);
}
ShowCurrentDir(path,info);
//Axtarış üçün ayrılmış resursları
silirik
_findclose(result);
return true;
}

void main(){
    //Verilmiş dəyişəndə kataloqun yolu
    //saxlanılacaq
    char path[MAX_PATH];
    //Verilmiş dəyişəndə isitifadəçinin daxil
    //etdiyi əmr olacaq
    char action[size];
    //Müvəqqəti dəyişən
    char temp[MAX_PATH];
    //Cari kataloqun yolunu əldə edirik
    GetCurrentDirectory(sizeof(path),path);

    bool flag=true;
    //Cari kataloqun yolunun göstərilməsi
    ShowDir(path);
    do{
        //İstifadəçinin əmrlərinin daxil edilməsi
        cin.getline(action,size);
        //Sağdakı boşluq və slaşları silirik
        RemoveRSpacesAndRSlashes(action);
        //Ana kataloqa keçid
        if(!strcmpi(action,"root")){
            path[2]='\0';
            ShowDir(path);
        }

        //İstifadəçinin proqramdan çıxma
        //istəyinin yoxlanılması
        else if(!strcmpi(action,"exit")){

```

```

        flag=false;
    }
    //cd əmriniin yoxlanılması
    else if(!strnicmp(action,"cd",2)){
        //Cari kataloqun məzmununun göstərilməsi
        if((!strcmpi(action,"cd"))){
            //Kataloqun göstərilməsi
            ShowDir(path);
        }
        //cd əmri parametrləri ilə verilmişdir
        else if(!strnicmp(action,"cd ",3)){
            //Parametrin indeksini tapırıq
            int index=strspn(action+2," ");
            if(index){
                //Kataloqun tam adının yoxlanılması
                if(strchr(action+index+2,':')){
                    //Kataloqun məzmununun əks
                    //etdirilməsinə cəhd edirik
                    if(ShowDir(action+index+2)){
                        strcpy(path,action+index+2);
                    }
                    else{
                        //Səhv baş vermişdir
                        ShowCurrentDir(path,temp);
                    }
                }
                //Ana kataloqa keçirik
            }
            else if(!strcmp(action+index+2,"..")){
                char *result=strrchr(path,'\\');
                if(result){
                    int delta=result-path;
                    strncpy(temp,path,delta);
                    temp[delta]='\0';
                }
                else{
                    strcpy(temp,path);
                }
            }
        }
    }

```

```

        if(ShowDir(temp)){
            strcpy(path,temp);
        }
        else{
            //Səhv baş vermişdir
            ShowCurrentDir(path,temp);
        }
    }
    //Kataloqun göstərilməsi
    else if(!strcmp(action+index+2,".")){
        ShowDir(path);
    }
    else if(!strcmp(action+index+2,"/")){
        ShowDir(path);
    }
    else{
        //tam olmayan yol verilmişdir
        strcpy(temp,path);
        strcat(temp,"\\");
        strcat(temp,action+index+2);
        //Kataloqun məzmununun
        //göstərilməsinə cəhd edilmişdir
        if(ShowDir(temp)){
            strcpy(path,temp);
        }
        else{
            //Səhv baş vermişdir
            ShowCurrentDir(path,temp);
        }
    }
}
else{
    //kataloqun göstərilməsi
    ShowDir(path);
}
}
else{

```

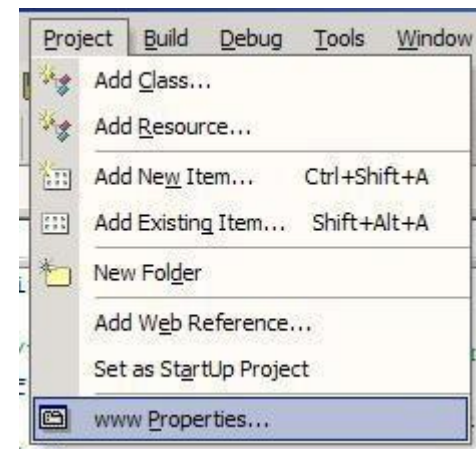
```
        //Kataloqun göstərilməsi
        ShowDir(path);
    }
}
else{
    //Kataloqun göstərilməsi
    ShowDir(path);
}
}while(flag);
}
```

Əmrlər sətirinin argumentlərinin istifadə edilməsi

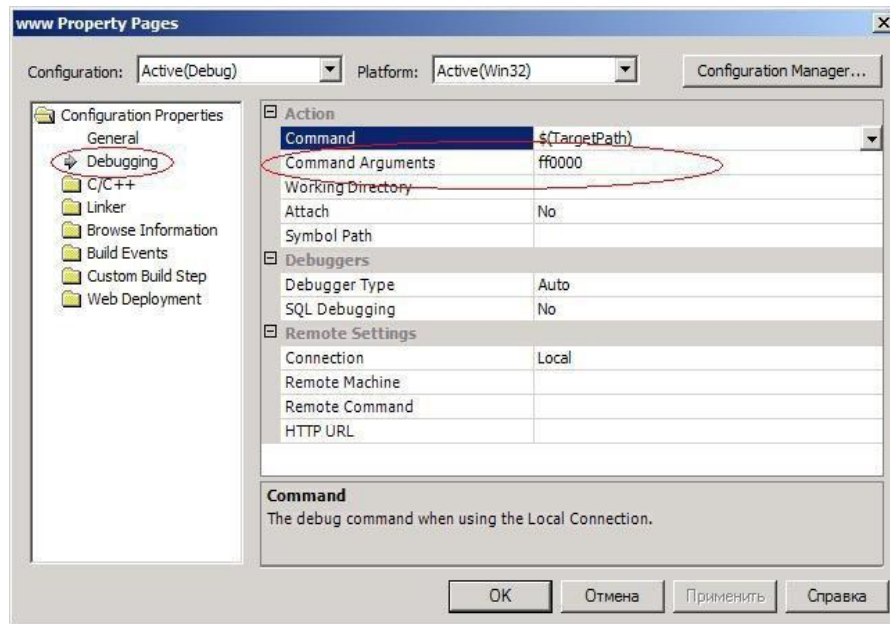
Elə əməliyyat sistemləri vardır ki, main funksiyasına argumentləri əmrlər sətirindən göndərmək olur. Bu cür ötürmənin üç üsuluna baxacağıq.

Birinci üsul Visual Studio vasitəsilə ötürmə

Bunun üçün Project menyusuna daxil olmalı və **"proyektin adı" Properties....** seçməli.

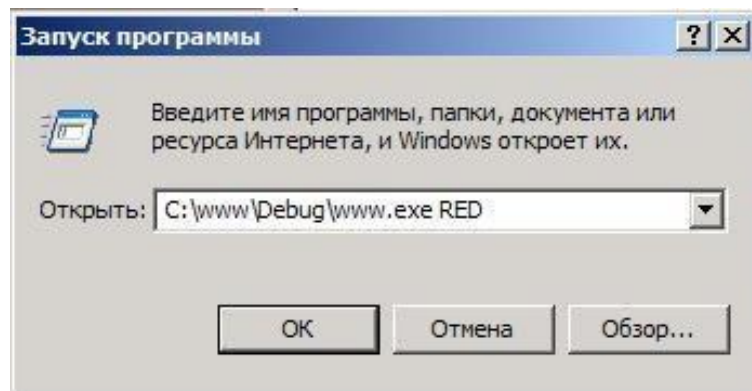


Sonra açılmış pəncərnin Debugging bölməsinin Command Arguments sahəsinə əmrlər sətirinin argumentlərini yazmaq lazımdır.



İkinci üsul

"Пуск" menyusunun "Выполнить" əmri vasitəsilə
ötürmək



Üçüncü üsul Cansole (cmd.exe) vasitəsilə ötürmək



Main funksiyasında əmrlər sətirinin argumentlərinin verilməsi.

Əmrlər sətirinin argumentlərinin verilməsi üçün main funksiyasının parametrlər siyahısına iki xüsusi parametrlər daxil etmək lazımdır:

1. **int argc** — verilmiş parametrlər əmrlər sətirinin argumentlərinin sayına bərabər olan qiymət yazılır.
2. **char *argv[]** — əmrlər sətirinin argumentlərinin qiymətlərinin birbaşa yerləşdirildiyi sətirlər massivi
Qeyd etmək lazımdır ki, argc massivində həmişə bir element – proqramın yolu olur.

Qeyd: DİQQƏT!!! main funksiyasındakı parametrlərin adı vacib deyil.

Dərsin növbəti bölməsində əmrlər sətirinin parametrlərinin istifadəsinə aid nümunəyə baxacağıq.

Praktik nümunə

HTML sənədi hazırlayan və onun fon rəngini əmrlər sətirindən parametrlər kimi ötürülən rənglə rəngləyən program nümunəsinə baxaq

```
#include <iostream>
#include <string.h>
#include <stdio.h>

using namespace std;

void main(int argc, char * argv[])
{ //Susmaya görə qara rəng veririk
  char str[7]="000000";
  //HTML - sənədin əvvəlini formalaşdırırıq
  char filehtml[256]="<html><head><title>
    New file!!!</title></head><body bgcolor =
    \'#"; //Faylı yazmaq üçün açırıq
  FILE*f=fopen("C:\\1.html", "w+");
  //Alınmadısa, dayanırıq
  if(!f) exit(0);
  //Əgər rəng parametri ötürülərsə, onu
  //istifadə edirik
  if(argc==2){
    strcpy(str,argv[1]);
  }
  //Rəngi sənədə qoşuruq
  strcat(filehtml,str);
  //Sonluğu sənədə qoşuruq
  strcat(filehtml,"\'></body></html>");
  //Faylda saxlayırıq
  fputs(filehtml,f);
```

```
//Faylı bağlayırıq
fclose(f);
cout<<"\nOK.....\n";
}
```

Ev tapşırığı

1. Növbəti proqramlar dəstini yaratmaq lazımdır:

- ■ Kataloqların köçürülməsi üçün proqram (bütün daxili kataloq və fayllar köçürülür).
- ■ Kataloqların yerinin dəyişdirilməsi üçün proqram (bütün daxili kataloq və fayllar yerini dəyişdirir).
- ■ Kataloqların silinməsi üçün proqram (bütün daxili kataloq və fayllar silinir).

Əgər silinən faylda (kataloqda) Read-Only atributu təyin edilmişdirsə, növbəti menyunu vermək lazımdır:

1. Silmək? 2. Keçmək? 3. Həmişə üçün silmək? 4. İmtina?

2. Köçürmə və yerdəyişmə funksiyaları üçün eyni əməliyyatları yerinə yetirməli: 1. Yenidən yazmaq? 2. Keçmək? 3. Həmişə üçün yenidən yazmaq? 4. İmtina?

Proqramın parametrləri əmrlər sətri vasitəsilə ötürülür.

Məsələn,

■ ■ copy..exe c:\A d:\B

■ ■ move..exe c:\A d:\B

■ ■ delete..exe c:\A