

## Дәріс - 6.

**Тақырыбы:** Файл. Режим. Файл соңына қосу. CIN, COUT операторлары. Тізбекті ену файлы.

### Жоспар:

1. Файл. Тізбекті ену файлы
2. CIN, COUT операторлары

C++ тілінде байттар жиынтығы не дискіге жазылған түрлі құрылымдардан тұратын бір мәлімет **файл** деп аталады. Мысалы, файл:

- 1001 Мәдіходжаев Сұлтан 1985 қазақ
- 1002 Нұрланқызы Маржан 1987 қазақ
- 1003 Нұрғалиқызы Мәлика 1988 қазақ

(Мұндай файл кез келген n жолдан тұруы да мүмкін). Тізімдік файлдың әр жолында (құрылымда) бір студент жөнінде мәлімет берілген, оны **жазба** не **файл элементі** деп атайды. Файл элементі сандық және символдық типті бес мәннен (**өрістен**) тұр.

Жалпы, **Си++** тілінде файлдар тек **байттар ағыны** ретінде қарастырылған, сондықтан элементтердің жазба түрінде құрылуы міндетті емес.

Файл дискіде 1-8 символдан тұруы мүмкін жеке атау бойынша сақталады, атаудың толықтырмасын **.dat**, **.txt** деп белгілеу не толықтырманы енгізбей кету де мүмкін.

Файлмен жүргізілуі мүмкін жұмыстар:

- дискіге тиеп, сақтау;
- экранға шығару не принтер арқылы қағазға басу;
- файлға жаңа элементтер қосу не бір, не бірнеше элементін алып тастау және т.б.

Файлмен **тізбекті ену** және **еркін** (тікелей, кездейсоқ) **ену** тәсілдері бойынша жұмыс істеу мүмкін.

C++ тілінде енгізу және шығару тәсілдері қиынырақ, олар **символдар тасқыны** (**ағыны, поток**) бойынша орындалады. Мысалы, double типті —5.27 санын файлға жазу, шығару үшін 8 байттық ұяшық пайдаланбай, бөлек байттардан тұратын -, 5, . , 2 және 7 символдары ағын бойынша ретімен өзгеріссіз енгізіліп, шығарылады (*өзгеріссіз* дегеніміз — ол үшін функцияларды өзгертудің қажеті жоқтығы).

Файлмен жұмыс істеу **файл көрсеткіші** арқылы орындалады. Ол файлдың атауын, тұрған орнын, ағынды пайдалану тәсілін де анықтайды. Файл көрсеткіші - стандартты **stdio.h** файлында анықталған құрылымдық **FILE** типіне нұсқама. Ол арқылы файлды дискіге тиеу (жүктеу), оны дискіден машинаның жадына енгізу әр элементтің басынан бастап ретімен жүргізіледі.

C++ - те рұқсат етілген файлдық объектілерді **fout**, **fin** арқылы белгілесек, файлмен жұмыс істеу программасында пайдаланатын функциялық операторлар мен командалар:

1. **#include <fstream.h>** // препроцессор нұсқауы
2. **ofstream fout;** // жазу үшін файл объектісін жариялау
3. **ifstream fin;** // оқу үшін файл объектісін жариялау
4. **fout.open (Файл аты, ios :: in)** // **Файл құру**
5. **fin.open (Файл аты, ios :: out)** // **Файлды оқу**
6. **fout.close()** // Құрылған файлды жабу
7. **fin.close()** // Оқылған файлды жабу

(**Ескерту.** **fout**, **fin** белгілеулерін кез келген басқа түрде алуға да болады, мысалы: fp, xx).

Мұндағы **Файл аты** — файлға берілетін атау; **ios :: in**, **ios :: out** — қол жеткізу мүмкін режимдер (4.1-кесте):

#### Қол жеткізу режимдері (4.1- кесте)

Режим	Пайдалануы
ios :: out	Жазу үшін файлды ашу.
ios :: in	Оқу үшін файлды ашу.
ios :: app	Элемент қосу үшін файлды ашу.
ios :: binary	Файлды екілік жүйеде ашу.
ios :: ate	Файлды ашып, көрсеткішті соңына орналастыру.
ios :: trunc	Файл ішіндегісін алып тастау.
ios :: nocreate	Файл жоқ болса, ашу операциясын орындамау.

C++ тілінде ағындар мен буферлерді іске қосу қиынырақ, бірақ файл кодына **iostream.h** стандартты файлы қосылған кезде оларды басқаруға арналған **кластар** да іске қосылады: **streambuf**, **ios\_base**, **ostream** және **fstream** ( **класс** — анықталған тип).

**streambuf** - буферді бөлу, буферді толтыру әдістерін, буфер ішіндегісіне қол жеткізу түрлерін және буферді тазалауды орындау үшін жадты басқарады.

**ios\_base** — ағынның жалпы қасиеттерін (оқу үшін ашықтығын, мәліметтердің екілік не мәтіндік типті болуын) айқындайды да, көрсеткішті **streambuf** объектісіне қосады. Оған түрлі форматтау тұрақтылары да енгізілген.

**ios** класы **ios\_base** класына негізделген, **ostream** және **fstream** кластары - **ios** класының туындылары. Олар сәйкес шығару және енгізу кластарының мұрагерлері. **Файлға жазу ostream** класының << операциясы арқылы, **файлдан оқу fstream** класының >> операциясы арқылы орындалады. Программаға **iostream (ios)** класының кітапханасы қосылған кезде автоматты түрде программада ағымдық объектілер құрылады. Олардың негізгілері: **cin, cout, cerr, clog**.

**cerr, clog** объектілері (айнымалылары) программада кеткен қатені айқындап, экранға шығарады.

#### Файл соңына қосу

Мысал. (34)-программа бойынша құрылған программада жазылған үш жазудан тұратын файл соңына бір элемент қосу керек (ескерту: "Соңына жазу қосатын файл аты-?" сұрағына сақталған файл атын енгізу керек).

Программаны мынадай түрде құруға болады:

```
#include <stdio.h>
#include <conio.h>
#include <fstream.h>
ofstream fout;
main()
{
char атау[8]; clrscr();
puts ("Соңына жазу қосатын файл аты-? "); gets (атау);
fout.open (атау, ios :: app);
fout << " 1004 Ахметов Ержан 1985.қазақ\n";
fout. close(); getch(); return 0;
}
```

#### CIN, COUT операторлары

**COUT** - **ostream** класындағы ауыстырып тиеу объектісі (информацияны экранға шығару операторы). Жазылу түрі: **cout << inform;** Мұндағы << алдыңғы тарауларда пайдаланылған **биттік** операция емес, арнайы әрекетті орындайтын команда, мысалы, оның

*форматтау* қасиеттері жеткілікті. Оны **кірістіру операторы** деп атайды. **Inform** айнымалысы жад облысын көрсетеді, ол тұрақты, жол не өрнек болуы да мүмкін. Мысалы:

char ms[10]="on adam"; cout << ms; **inform** тұрақты болса, тырнақшаларға алынып жазылады. Ал, бірнеше тұрақтыдан тұрса, араларына бос орын не бірнеше символдық бос орын тастап кететін табуляция ( \t ) белгісін қойып кеткен дұрыс. Мысалы:

```
cmd1 cmd2 cmd3
2      3      1
1      2      3
2      1      3
```

мәліметін форматтап шығарудың программасы:

```
#include <iostream.h >
main() (33)
{
cout << "cmd1\t cmd2\t cmd3\t \n";
cout <<"2\t 3\t 1\t \n";
cout <<"1\t2\t3\t\n";
cout << "2\t 1\t 3\t \n";
}
```

CIN — *байттар ағынын* енгізу операторы (объектісі). Жазылу түрі: **cin >>** **айнымалы;** (айнымалы мәні — мәлімет не сан). Ол **пернетақпадан** пайдаланушының мәліметтерді **дұрыс енгізуін талап** етеді. Мысалы, мәлімет *мәтін* болса, **тек бір сөзден** ғана тұруы тиіс, егер ол бірнеше сөзден тұрса, енгізуде **get** операторын пайдаланған жөн. Жалпы, cout және cin - *форматталған шығару, енгізу операторлары, яғни* олар **форматтық %d, %s сияқты** символдарды енгізуді қажет етпей, өздері жоспарлайды. **Яғни,** олар **printf, scanf операторларын** пайдаланбай, **программа жазуды оңайлатады.**

### Тізбекті ену файлы

Басынан бастап элементтері бойынша ретімен құрылған файлды *тізбекті ену файлы* деп атайды. С++ тілінде көлемі үлкен емес файлды бірден енгізуге болады.

**Ескерту.** Құрылатын файл атауын **.dat** не **.txt** кеңейтілуімен берген жөн, мысалы, **tizim.dat** (атауды кеңейтілуісз беруге де болады).

**Мысал.** С++ тілінде 4.2-тақырыпта берілген тізімді тізбекті ену файлы түрінде дискіге жазу керек.

Программаны мынадай түрде құруға болады:

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h> // енгізбеу де мүмкін
#include <fstream.h>
ofsream fout;
main() (34)
charatau[8]; clrscr();
puts ("Файл аты-? "); gets (atau);
fout.open (atau, ios :: out);
fout << " 1001 Мәдіходжаев Сұлтан 1985 қазақ \n";
fout << " 1002 Нұрланқызы Маржан 1987 қазақ \n";
fout << " 1003 Нұрғалиқызы Мәлика 1988 қазақ \n";
fout. close(); getch(); return 0;
}
```

Мұндағы **iostream.h, fstream.h** файл құру мен оқу файлдары. Файлды құруға арналған **fout.open (atau, ios :: uut);** операторы **fout** үшін буферден орын бөліп қояды, ал, << операциясы онан соң жазылған **fout** операторларының мазмұндарын берілген *atau* бойынша бірден дискіге көшіріп жазады.

**Ескерту.** 1. Файл құру кезінде енгізілген жаңа **cout** операторы **cout** операторын алмастырады, яғни файл құру үшін **cout** орнына **cout** операторы пайдаланылады. Ол файлды дискіге жазу мен қоса ағымдық каталогқа да енгізіп қояды. Осы сияқты, оқу кезінде **cin** операторының орнына **fin** операторы қолданылады (оның >> операциясы құрылған файлды берілген атау бойынша буферге енгізеді).

2. Программа жаңа файл құруды сұраған кезде бұрыннан құрылған файл атын енгізуге болмайды, бұл кезде құрулы файл жойылып кетеді.

3. Жеке сандардан не жоғарыдағы сияқты, тұтас қарастырылған жолдардан файл құру ықшамды.

**Тізбекті ену файлын оқып, экранға шығару программасы:**

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
#include <fstream.h>
ifstream fin;
main()
{
charatau[8]; char in_char; clrscr();
cout << "Файл аты-? " cin >> atau;
fin.open (atau, ios :: in);
while ( fin.get ( in_char ))
cout << in_char; /* символдарды ретімен экранға шығару */
fin. close(); getch(); return 0;
}
```

Файл аты енгізілген соң **fin.open (atau, ios :: in);** операторы файлды оқуға ашады, **fin.get** операторы файлдың жеке символдарын (**in\_char**) ретімен буферге енгізеді де, **while** операторының құрамындағы **cout << in\_char** операторы оларды ретімен оқып экранға шығарады.

Ал, **С** тілінде файлдық ағынға қол жеткізудің **тәсілі** - оны көрсеткіш (**fp**) арқылы сипаттау. Оның сипатталу түрі:

**FILE \* fp;**

Мұндағы **FILE** - **<stdio.h>** файлында анықталған тип. Көрсеткішті нақты файлмен байланыстыру мынадай оператор арқылы орындалады:

**fopen ( "Файлға жол", "қол жеткізу типі");**

**С** тілінде **файлға жол** үшін ағымдық каталогта пайдаланатын файл идентификаторын не файлдың толық атын енгізуге болады, **қол жеткізу типі** үшін "wt" не "w" параметрі енгізіледі (Бұл - жазу үшін файлды ашу параметрі). Файл атын енгізуде қате кетсе, оператор көрсеткішті бос файлға (NULL) қайтарып береді. Бұл операторды пайдалану кезінде жаңа файл құру үшін бұрынғы **printf()**, **scanf()** операторлары пайдаланылады. Соңғысы әр жазба элементін ретімен буферге енгізеді де, арнайы **fprintf()** операторы файл элементтерін ретмен дискіге жазып шығады. Мысалы,

**fp = fopen ("Fail.dat", "wt");**

операторының орындалу кезінде **Fail.dat** атаулы файл **fp** көрсеткішінің көмегімен **жазуға** ашылады.

Жоғарғы операторды **Турбо Си++** тілінде де пайдалануға болады, **мысалы**, (33)- программаның Турбо Си++ тілінде құрылуының жалпы түрі:

```
#include <stdio.h>
#include <conio.h>
void main()
{
int shifr, tjli, k; char gam[15], ati[15], ulti[6];
charatau[8]; clrscr();
```

```

FILE * out;          /* fp=out */
printf("атау-?"); scanf("%s", &атау);
if ((out=fopen(атау,"wt")) == NULL)
{ printf("қате"); getch(); return; }
for (k=1; k<=3; k++)
{
printf("Шифр- "); scanf("%d ", &shifr);
printf("Фамилия - "); scanf("%s ", &fam);
printf("Аты- "); scanf("%s",&ати);
printf("Туғанжылы- "); scanf("%d", &tjli);
printf("Ұлты- "); scanf("%s", &ulti); /* буферден дискіге көшіріп жазу операторы-
fprintf */
fprintf( out, "%d %s %s %d %s\n",
shifr, fam, ati, tjli, ulti);
}
fclose(out); getch(); return;
}

```

Мұндағы ықшамдылық: кез келген құрылымдық файлдың элементтерін буферге енгізіп, ретімен дискіге көшіріп жазуға болатынында.

### Бақылау сұрақтары

- 1 Файл дегеніміз не?
- 2 Файлмен қандай тәсілдер бойынша жұмыс істеуге болады?
- 3 Сin стандартты құрылғандарды енгізу операторы, оның кемшіліктері?
- 4 Тізбекті ену файлына ауызша мысал келтір?
- 5 СOUT экранға шығару операторын сипаттап беріңіз?

### Әдебиеттер тізімі

1. Берн Страуструп. Язык программирования С++. Москва: 1999г.
2. Бабэ Б. Просто и ясно о Borland С++: пер. с англ. – СПб.: Питер, 1997.
3. Бад Т. Объектно-ориентированное программирование в действии: пер. с англ. СПб.: Питер, 1997. – 464с.
4. Ирэ П. Объектно-ориентированное программирование с использованием С++: пер. с англ. К.: НИПФ ДиаСофтЛтд, 1995. – 480с.
5. Подбельский В.В. Язык С++. – М.: Финансы и статистика, 1966. – 558с.
6. Шамис В.А. Borland С++ Builder. Программирование на С++: пер. с англ. –М.: БИНОМ.-480с.
7. Шилдт Г. Теория и практика С++. Пер. с англ. – СПб.: БХВ-Петербург, 1996.-416с.
8. Архангельский А.Я. Программирование в С++ Builder 6. –М.: ЗАО «Издательство БИНОМ», 2003.-1152с.
9. Б.Пахомов. С/С++ и Borland С++ 2006. Санкт-Петербург.: БХВ-Петербург., 2006.
10. Т.Павловская. С/С++. Структурное программирование. Питер. 2005.
11. Программирование на С++. Учебное пособие. Под ред. А.Д.Хомоненко. Москва.: Альтекс-А, 2003.