File input/output alapok

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2023 szeptember 26.

Streamek

- A C++ -ban az adatok bevitele és kiírása ún `stream`-eken keresztül történik
- ilyennel már találkoztunk:
  - a standard bevitel (ún `istream`) amelyet a `cin`-nel használtunk
  - a standard output (ún `ostream`) amelyet a `cout`-tal használtunk

Pl. mit csinál egy pl egy `ostream`?

- különböző adattípusokat karaktersorozattá konvertál
- ezt a karaktersorozatot "valahová" elküldi (pl konzolablak, fájl stb)

Fájl streamek

Egy fájl olvasásához vagy írásához

- a fájlt meg kell nyitni
- átadni az olvasó/író függvénynek
- a végén a fájlt be kell zárni (a C++ -ban általában nem kell explicit ezzel foglalkozni)

Fájl streamek

Egy fájl olvasásához vagy írásához

- a fájlt meg kell nyitni
- átadni az olvasó/író függvénynek
- a végén a fájlt be kell zárni (a C++ -ban általában nem kell explicit ezzel foglalkozni)

- A fájlokkal kapcsolatos I/O streameket a `fstream` header fájl segítségével tudjuk használni
- három fajta fájl stream van definiálva
  - **ifstream** olvasásra
  - **ofstream** kiírásra
  - **fstream** olvasásra és írásra is

Fájl streamek megnyitása

```
#include <fstream>

main(int argc, char* argv[])
{
ifstream inpfile(argv[1]);

ofstream outfile(argv[2]);
}
```

- parancssori argumentumként megadunk egy fájlnevet és megnyitjuk olvasásra
- parancssori argumentumkent megadunk egy fájlnevet és megnyitjuk írásra

Fájl streamek megnyitása

```
#include <fstream>

main(int argc, char* argv[])
{
ifstream inpfile(argv[1]);

ofstream outfile(argv[2]);
}
```

- parancssori argumentumként megadunk egy fájlnevet és megnyitjuk olvasásra
- parancssori argumentumkent megadunk egy fájlnevet és megnyitjuk írásra

- fontos ellenőrizni, hogy sikerült-e megnyitni az inputfájlt

```
if (!inpfile)
{cout << "nem talalom a fajlt" << argv[1] << endl;}
```

- használhatjuk az `is_open()` operációt is:

```
if ( inpfile.is_open() )
{ \\ read data }
else
{cout << "nem talalom a fajlt" << argv[1] << endl;}
```

Fájl stream olvasás/írás

Itt most szöveges fájlokkal foglalkozunk

- 10-es számrendszerbeli számokat olvasunk be/írunk ki
- (másik lehetőség: binárisan tárolva, mint a memóriában)
- miután megnyitottuk az input/output fájlt, használhatjuk a « és » operátorokat

Fájl stream olvasás/írás

Itt most szöveges fájlokkal foglalkozunk

- 10-es számrendszerbeli számokat olvasunk be/írunk ki
- (másik lehetőség: binárisan tárolva, mint a memóriában)
- miután megnyitottuk az input/output fájlt, használhatjuk a « és » operátorokat

```
ifstream inpfile("inp_data.dat");
ofstream outfile("out_data.dat");

int i=0;

inpfile >> i;

outfile << i;

for (int j; inpfile >> j;)
{
//do something
}
```

- a streamből beolvasott karaktereket átalakítja `int`-re és az `i` változóba kerülnek
- a `int` típusú szám karakterré alakul és kiíródik
- amíg a fájl végére nem ér, olvasson be `int`-ket

Megjegyzés:

- a `for` ciklus szintaxisa: `for (`*[init]*`;` *[condition]*`;` *[increment]*`)`
- de egyes részek elhagyhatóak (pl *[increment]*))

Fájl stream olvasás/írás

- fájlba kiírni általában könnyebb, mint onnan beolvasni, különösen struktúrált adatokat
- struktúrált adatok beolvasására még visszatérünk