# 10. gyakorlat anyaga

## String

A C++-ban bevezették a string osztályt, amely leegyszerűsíti a stringkezelést. Eddig is használhatjuk a C-ben megismert, char\* -os megoldást, de ehelyett használhatjuk a string osztályt is. Először, include-olni kell a string header fájlt (csak string, **nem** string.h!!!). Ezek után használhatjuk a következő kifejezéseket:

string a = "zoldseg";

string b("gyumolcs");

Sok szempontból is előnyös, nem kell kézzel helyett foglalni adott méretű tömbnek, és rengeteg függvényt megvalósítottak, hogy még könnyebben menjen a sztringek használata. Emellett megfelelően túlterhelték az operátorokat, melyek segítségével még gyorsabban tudunk műveletek végrehajtani sztringekkel.

|  |  |
| --- | --- |
| size | sztring mérete |
| append | hozzáfűzés egy meglévő sztringhez. megegyezik a hatása a + (vagy +=) operátoréval. |
| at | sztring adott pozícióján lévő karakter visszaadása. hatása megegyezik a tömbindex operátoréval |
| replace | Sztringben rész-sztring cserét hajt végre |

Egy egyszerű példakód:

#include <iostream>

#include <string>

using namespace std;

int main(){

string a = "zoldseg";

string b("gyumolcs");

string c = a;

string d = b+"leves";

c.append("leves");

cout << "a valtozo:" << a << " b valtozo: " << b << " c valtozo: " << c << " d valtozo: " << d << endl;

int meret = c.size();

cout << "a c merete: " << meret << " az a valtozo 2. poziiojan levo betu"

<< a.at(2) << " ami ugyanaz, mint: " << a[2] << endl;

if(c == d) {

cout << "A ket leves ugyanaz!" << endl;

} else {

cout << " Nem ugyanaz a ket leves" << endl;

}

return 0;

}

Bővebben:

http://www.tutorialspoint.com/cplusplus/cpp\_strings.htm

http://www.cplusplus.com/reference/string/string/

## Alapvető file input/output

A C++ nyelvben a file műveleteket is megvalósíthatjuk streamek segítségével, így az alap file műveletek egyszerűen használhatóak. Használatukhoz include-olni kell az fstream header-t, mely tartalmazza az alapvető műveleteket.

#include <fstream>

Ezek uán használhatjuk a különböző osztályokat:

* Beolvasáshoz: ifstream
* Kiíráshoz: ofstream
* Mindkettőhöz: fstream

Megnyitási módok:

* + **in** - a fájlt olvasásra nyitjuk meg (fstream esetében ez az alapértelmezett)
  + **out** - a fájlt írásra nyitjuk meg (ofstream esetében ez az alapértelmezett)
  + **binary** – bináris módban nyitjuk meg
  + **ate** – az output pozícója a fájl végére mutat
  + **app** – hozzáfűzésre nyitjuk meg
  + **truncate** – meglévő fájl esetén a tartalmát töröljük

Példányosítás írásra: (olvasásra ugyanez, csak ifstream-mel!)

* + ofstream out;
  + out.open( fileName ); // out.open( fileName , ios::out | ios::binary);

VAGY

* + ofstream out( fileName); // ofstream out( fileName , ios::out | ios::binary);

Példányosítás olvasásra:

* + ifstream in;
  + in.open( fileName );

VAGY

* + ifstream in( fileName);

ofstream alapértelmezetten írásra nyitja meg a fájlokat.

ifstream alapértelmezetten olvasásra nyitja meg a fájlokat.

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

int main() {

int a = 5;

double b = 3.4;

//char\* str = "valamivalami";

string str = "Valami";

ofstream out( "out.txt", ofstream::app );

out << a << b << str << endl;

out.close();

return 0;

}

Kivételkezelés alapjai

Az olyan kódrészleteknél, melyek „hibát” eredményezhetnek, érdemes lehet try-catch blokkokat használnunk.A „hibát okozó”, és az azzal közvetlen összefüggésben lévő program részletet tesszük try blokkba.  
Ekkor, ha hiba történik, lehetőség van egy exception-t (kivételt) dobni (throw). Ezt a kivételt „kaphatjuk el”, és kezelhetjük a catch blokkban.