



A C++ programozási nyelv alapjai

A legalapvetőbb dolgok

Példaprogram

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Szia vilag" << endl;
    system("pause");
    return 0;
}
```

A C++ programok felépítése - 1

Egy C++ program egy vagy több .cpp kiterjesztésű állományból áll (forrásfájlok). Pontosan az egyik forrásfájlnak tartalmaznia kell egy **main** nevű függvényt (ez a program „belépési pontja” – innen indul a program)

A C++ programok felépítése - 2

A legtöbb program tartalmaz egy # jellel kezdődő *direktívát*, amely az előfordítónak szóló parancs.

A C++ program egy vagy több függvényből áll.

Mindegyik függvény a következő alakú:

```
típus függvéynév(argumentumlista)
```

```
{
```

```
    utasítások
```

```
}
```

Minden utasítás pontosvesszővel zárul - ;

A C++ programok felépítése - 3

Az `#include <iostream>` direktíva hatására az előfordító a programhoz fűzi a beolvasáshoz és kiíráshoz szükséges definíciókat.

`using namespace std;` - a program azonosítói a standard (std) névtérhez tartoznak

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Szia vilag" << endl;
    system("pause");
    return 0;
}
```

A C++ programok felépítése - 4

`int` – a függvény
visszatérési értéke
integer – egész szám.
A `main` függvény
visszatérési értéke
általában 0, más
értéket hibajelzésre
használunk

A `return 0` utasítás adja a
függvény visszatérési
értékét, ebben az
esetben nullát.

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Szia vilag" << endl;
    system("pause");
    return 0;
}
```

A C++ programok felépítése - 5

az utasítások (parancsok) a main függvény törzsében (a kapcsos zárójelek között: { ... }) egyenként végrehajthatók.

Mindegyik utasítás pontosvesszővel (;) zárul.

Az idézőjelek közötti írásjelek sorozatát ("Szia világ") karakterláncnak (string) nevezzük.

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Szia világ" << endl;
    system("pause");
    return 0;
}
```

A C++ programok felépítése - 6

Az endl (vagy '\n') szimbólum az újsor jelet helyettesíti.

A system("pause") utasítás az operációs rendszer pause (szünet) parancsát hívja, amely egy billentyű lenyomásáig felfüggeszti a program futását.

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Szia világ" << endl;
    system("pause");
    return 0;
}
```


A tipikus c++ program szerkezete

előfordítói direktívák

külső deklarációk és definíciók

```
int main()
```

```
{  
    definíciók és deklarációk  
    utasítások  
    return 0;  
}
```

függvénydefiníciók

Programhibák

Szintaktikai vagy fordítási hiba (syntax-error):

Valami megsérti a nyelv szabályait.

A fordító hibát talált.

```
cout << "Szia világ!" << endl;
```

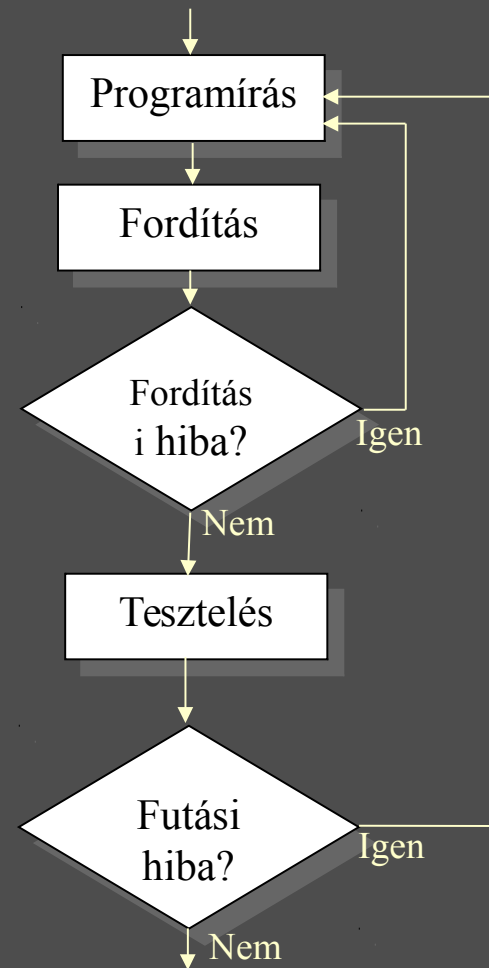
```
cout << "Szia világ" << end;
```

Logikai vagy futás közbeni hiba (runtime-error):

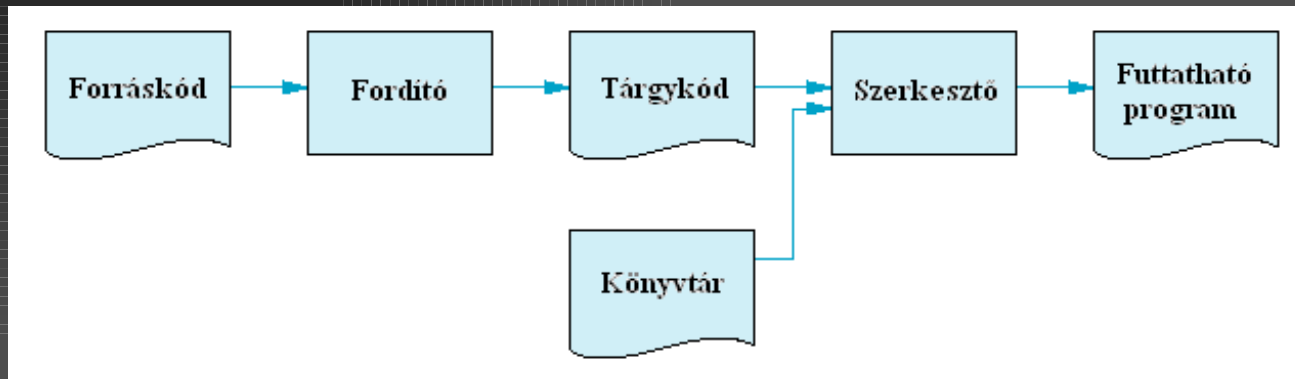
A program nem a kívánalmaknak megfelelően működik.

A programozónak kell megtalálnia a hibát – programtesztelés.

```
cout << "Sza vlag" << endl;
```



A fordítási folyamat



A C++ *forráskód* a fordítani kívánt program.

A *compiler* az a program, amelyik a C++ forráskód fordítását végzi.

A *tárgykód* (object code) gépi utasításokat és a program memóriába olvasását meghatározó információkat tartalmaz.

A *szerkesztő* (linker) összeállítja a tárgykódokat és a könyvtárakban szereplő kódokat egyetlen *futtatható állománnyá* (executable file).

A *könyvtárak* előre lefordított programrészeket tartalmaznak.

Fejlesztői környezetek

Borland C++

Microsoft C++, Visual C++

Dev-C++

Gcc

stb.

A Dev-C++ fejlesztői környezet

Új programkód

A meglévő program megnyitása

Fordítás

Futtatás

Fordítás és futtatás

