Juhász Tibor:

Code::Blocks modulok

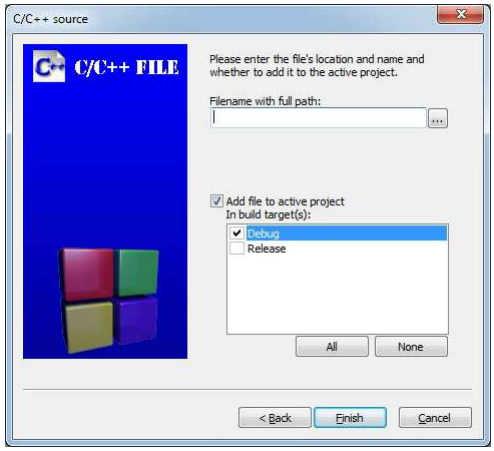
Kiegészítés a Programozási ismeretek című könyvhöz (Műszaki Kiadó 2011)

A C++ nyelven megírt programok általában több modulból állhatnak. A modulokat külön fájlokban tároljuk. Tankönyvünkben főleg a tömbök elemeinek kezdőértékét adjuk meg modulokban. A modulokhoz fejállományok is tartoznak, melyek típusokat, valamint a változók és függvények deklarációit tartalmazzák. Egyszerűbb függvények esetén a definíciót szintén a fejállományba írhatjuk (*inline* definíció).

A fejállományokat az *#include* direktíva (a fordítóprogramnak szóló utasítás) segítségével építjük be a modulokba.

Több modulból álló program készítése

1. Hozzunk létre egy új projektet (a *main*() függvényt tartalmazó *main.cpp* modullal)!
2. Készítsünk egy új C++ forrásfájlt, ami az újabb modul kódját fogja tartalmazni:
   File/*New*/*File*, *C/C++ source*, *Go*
3. A fájlnév megadásakor adjuk hozzá a fájlt az aktív projekthez, és jelöljük be legalább a *Debug* célt!
4. Írjuk meg a forráskódot! Építsük be a forrásfájlhoz tartozó fejállományt (lásd az 5. pontot)!
5. Hozzunk létre egy új fejállományt:
   File/*New*/*File*, *C/C++ header*
   A fejállomány neve célszerűen ugyanaz legyen, mint a *cpp* fájl neve!
   A fejállományt is adjuk hozzá a projekthez! Itt szintén jelöljük be legalább a *Debug* célt!
6. Vegyük fel a fejállományba az újabb modulban szereplő azon változók és függvények deklarációit, melyekre más modulból szeretnénk hivatkozni! A kezdőérték nélküli változóknál adjuk meg az *extern* (külső) tárolási osztályt (a függvények prototípusánál az *extern* elhagyható)!



A kódszerkesztő automatikusan kiegészíti a fejállomány kódját az *#ifndef*, *#define*, *#endif* direktívákkal. Alkalmazásukkal elkerüljük, hogy ugyanazt a modult esetleg többször építsük be a programba, ami hibához vezethet.

Az alábbiakban megadjuk egy C++ program forráskódját, amely a *main* modulon kívül egy tömböt és egy függvényt definiáló modult tartalmaz.

main.cpp

```
#include <iostream>
using namespace std;
#include "adatok.h"

int main()
{
    cout << nev[0] << endl << endl;
    kiir();

    cin.get();
    return 0;
}
```

adatok.cpp

```
#include <iostream>
#include <string>
using namespace std;
#include "adatok.h"

string nev[] = {"Alfonz", "Berta", "Cili", "Dani", "Eszter"};
const int meret = sizeof(nev)/sizeof(nev[0]);

void kiir() {
    cout << "Nevek:\n";
    for (int i=0; i<meret; i++)
        cout << nev[i] << endl;
}
```

adatok.h

```
#include <string>
using namespace std;

extern const int meret;
extern string nev[];

void kiir();
```

Létező modul felvétele a projektbe

1. Másoljuk be a modult a fejállománnyal együtt a projekt könyvtárába!
2. Adjuk hozzá a fájlokat a projekthez:
   Project/Add files
   Jelöljük be legalább a *Debug* célt!
3. Építsük be a forrásfájlhoz tartozó fejállományt a *main.cpp* modulba!

Az alábbi program kilistázza a forrásfájlok között található *Diakok.cpp* modulban inicializált *nev* tömb elemeit.

main.cpp

```
#include <iostream>
using namespace std;
#include "Diakok.h"

int main()
{
    for (int i=0; i<nevSzam; i++)
        cout << i+1 << ". " << nev[i] << endl;

    cin.get();
    return 0;
}
```