

# OO Programozás

## Konstansok

Dr. Hatwagner F. Miklós

Széchenyi István Egyetem, Győr

[https://github.com/wajzy/GKxB\\_INTM085](https://github.com/wajzy/GKxB_INTM085)

2023. szeptember 13.

A `const` típusmódosító változókkal használva

- *Konstans változó* (**paradoxon!** → *elnevezett konstans*)
- Memóriában helyezik el,
- értéke megjelenik a nyomkövető programokban (debugger),
- csak olvasható (fordító biztosítja),
- van típusa (vö. `#define`).
- Láthatóságuk az őket definiáló állományra terjed csak ki.

Tömbök definiálása, méret megadása

- konstans kifejezéssel,
- konstans kifejezéssel inicializált elnevezett konstanssal,
- vagy tetszőleges kifejezéssel C99 / C++14-től

## constMain.cpp

```
1 #include <iostream>
2 #include "constHeader.h"
3 #define MEANING_M 42 // no type, cumbersome debugging
4
5 void array(int);
6
7 int main() {
8     const int MEANING = 42;
9     int meaningCopy = MEANING; // read
10    // MEANING++; // error
11    // MEANING = -42; // error
```

## constMain.cpp

```
12  // size expressed with constant expression
13  int oldArray1[MEANING_M * sizeof(double)]; // OK
14  int oldArray2[MEANING * sizeof(double)]; // OK
15  array(21);
16  // std::cout << i << std::endl; // error
17  std::cout << ci << std::endl; // OK
18  std::cout << si << std::endl; // OK
19  }
20
21  void array(int size) {
22      // C99+ / C++14+ -> variable size is OK, allocated on stack
23      int newArray[size * sizeof(double)];
24  }
```

## constHeader.h

```
1 // int i = 1; // error, multiple definition of 'i'  
2 const int ci = 2; // OK, visible only in defining file scope  
3 static int si = 3; // OK, file scope
```

## constSource.cpp

```
1 #include "constHeader.h"
```