

const, const parametry (no)

složí k vytvoření konstantní proměnné, nebo k ochraně proměnných před nechtěnou změnou

vytvoření konstantní (neměnné) proměnné

- nelze ji měnit – kontroluje překladač – chyba
- obdoba **#define PI 3.1415** z jazyka C
- typ proměnné je součástí definice **const float PI=3.1415;**
- obvykle dosazení přímé hodnoty při překladu
- **const int** a **int** jsou dva různé/rozlišitelné typy
- při snaze předat (odkazem) **const** proměnnou na místě nekonstantního parametru funkce dojde k chybě (starší překladače vytvoří dočasnou proměnnou)

U použití ve více modulech (deklarace-definice v h souboru) – rozdíl v C a C++

- u C je **const char a='b';** ekvivalentní **extern const char a='b';**
- pro lokální viditelnost – **static const char a='b';**
- u C++ je **const char a='b';** ekvivalentní **static const char a='b';**
- pro globální viditelnost – **extern const char a='b';**
- výhodné psát včetně modifikátorů **extern/static**

Potlačení možnosti změn u parametrů předávaných funkcím (především) ukazatelem a odkazem

```
int fce(const int *i)  
int fce1(int const &ii)  
const int & fce2(double aa)...
```

- proměnná označená const je hlídána překladačem před změnou
- volání const parametrem na místě nonconst parametru (fce) – nelze

shrnutí definicí (typ, ukazatel, reference, const):

T	je proměnná daného typu
T *	je ukazatel na daný typ
T &	reference na T
const T T const	deklaruje konstantní T (const char a='b';)
T const * const T*	deklaruje ukazatel na konstantní T
T const & const T&	deklaruje referenci na konstantní T
T * const	deklaruje konstantní ukazatel na T
T const * const const T* const	deklaruje konstantní ukazatel na konstantní T