

const a metody (o)

- const u parametrů – parametry nepředávat hodnotou (paměťově a časově náročné), const slouží jako ochrana proti nechtěnému přepisu hodnot v metodě (funkci)
- const u návratové hodnoty – návratovou hodnotu nelze měnit (či použít na místě ne-const argumentu metody/funkce)
- const parametry by neměly být předávány na místě nonconst parametrů
- na const parametry nelze volat metody, které je změní – kontrola překladač
- metody, které nemění objekt je nutno označit (programátorem) jako const a pouze takto označené metody lze volat na const objekty

```
float f1(void) const { ... }  
//míní float f1(T const * const this) {}
```

```
float f2(void) {}
```

```
const T a;
```

```
T b;
```

```
a.f1(); b.f1(); b.f2(); // lze, převod na const je v pořádku  
a.f2(); // nelze konstantní proměnnou na místě nekonstantní
```

const a metody – použití v programu

- u parametrů metod/funkcí se doporučuje ty, které se nemění označit const
- u metod, které nemění parametr, který je vyvolal, je nutné za definici přidat const,
- případný pokus o změnu „this“ v metodě označené const hlídá překladač
- T a const T jsou dva různé typy. Proto můžeme mít dvě metody stejného jména, jednu volanou pro konstantní proměnné a druhou pro nekonstantní