

**NPRG041: Programování v C++**

<http://www.ksi.mff.cuni.cz/~svoboda/courses/211-NPRG041/>

Cvičení 6

# Parsování výrazů, zásobník

**Martin Svoboda**

[martin.svoboda@matfyz.cuni.cz](mailto:martin.svoboda@matfyz.cuni.cz)

10. 11. 2021

**Univerzita Karlova**, Matematicko-fyzikální fakulta

# P1: Aritmetické výrazy: Parsování

Vytvořte jednoduchý parser pro infixové aritmetické výrazy

- Předpokládejme následující základní binární operace
  - Sčítání +, odčítání -, násobení \* a dělení /
    - Všechny uvedené operace jsou zleva asociativní
    - Operace \* a / mají vyšší precedenci než operace + a -
  - Výraz může obsahovat i pomocné kulaté závorky ( )
- Pracujeme jen s přirozenými čísly včetně nuly
  - Jinými slovy před čísly nemohou být unární -
- Uvažujeme jen syntakticky dobře formované vstupy
- Převeďte vstupní výraz do postfixové (reverzní polské) notace
  - Vstupní infixový výraz vložte napevno přímo do kódu
  - Výstupní postfixový výraz vypište na standardní výstup
    - Operátory a čísla oddělte vždy právě jednou mezerou

# P1: Aritmetické výrazy: Parsování

Pokračování...

- Příklad transformace
  - Vstup:  $10*2+3*((1+14)-18)-10$
  - Výstup: `10 2 * 3 1 14 + 18 - * + 10 -`
- Použijte algoritmus shunting-yard
- Využijte standardní zásobník
  - `#include <stack>`
  - `std::stack<char>`
  - Metody `push(...)`, `top()`, `pop()`, `empty()`

# P1: Aritmetické výrazy: Parsování

---

- 1 **foreach** token  $t$  ve vstupním infixovém výrazu **do**
  - 2     **if**  $t$  je číslo **then** vypiš  $t$  na standardní výstup
  - 3     **else if**  $t$  je otevírací závorka ( **then** dej ( na zásobník
  - 4     **else if**  $t$  je zavírací závorka ) **then**
  - 5         **while** na vrcholu zásobníku je nějaký operátor  $o$  **do**
  - 6             └ odeber  $o$  ze zásobníku a vypiš jej na výstup
  - 7             odeber ( ze zásobníku
  - 8     **else**  $t$  je operátor  $n$
  - 9         **while** na zásobníku je operátor  $o$  s precedencí vyšší než  $n$  nebo
  - 10             také stejnou, je-li ovšem  $n$  zleva asociativní **do**
  - 11             └ odeber  $o$  ze zásobníku a vypiš jej na výstup
  - 11             └ přidej  $n$  na zásobník
  - 12 **while** zásobník je neprázdný **do**
  - 13     └ odeber  $o$  ze zásobníku a vypiš jej na výstup
-