

NPRG041: Programování v C++

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

Cvičení 8

Dědičnost, virtuální funkce

Martin Svoboda

martin.svoboda@matfyz.cuni.cz

1. 12. 2021

Univerzita Karlova, Matematicko-fyzikální fakulta

P1: Aritmetické výrazy: Uzly

Upravte kód našeho parseru aritmetických výrazů

- Nahradte stávající třídu pro uzel následující hierarchií tříd
 - Abstraktní třída `Node` jako společný předek
 - Odvozená finální třída `NumberNode` pro listové uzly s čísly
 - Abstraktní odvozená třída `OperationNode` pro vnitřní uzly
 - Třídy `...OperationNode` pro jednotlivé operace
 - `Addition...`, `Subtraction...`, `Multiplication...`, `Division...`
- Vhodně rozmístěte naše datové položky do jednotlivých tříd
 - Zcela se obejděte bez položky pro uložení operátoru a typu uzlu
- Zachovejte funkci na vypsání výrazu v postfixové notaci

P1: Aritmetické výrazy: Uzly

Pokračování...

- Základní použití dědičnosti
 - `class NumberNode final : public Node { ... }`
- Vhodně používejte virtuální metody
 - `virtual Type get_type();`
 - `Type get_type() override;`
 - `virtual Type get_type() = 0;`
 - Nezapomeňte na virtuální destruktor `virtual ~Node();`
- Definujte jen následující konstruktory
 - `NumberNode(std::string& number);`
 - `OperatorNode(Node* left, Node* right);`
 - Následně použijte `using OperatorNode::OperatorNode;`
- Vhodně používejte inlinové metody

P2: Aritmetické výrazy: Vyhodnocení

Rozšiřte náš parser

- Přidejte virtuální metodu na spočítání výsledku výrazu