



Interpreter

Interpreter

Definuje reprezentáciu gramatiky daného jazyka spolu s interpreterom, ktorý využíva túto reprezentáciu a interpretuje jej vety v tomto jazyku.

Given a language, define a representation for its grammar along with an interpreter that uses the representation to interpret sentences in the language.

Návrhové vzory



Motivácia

- Niekedy sa stáva, že namiesto vytvárania inštancii tried je ľahšie popísať problém pomocou jednoduchých viet vo vlastnom jazyku a postaviť interpreter, ktorý týmto vetám rozumie a vie ich vykonať.
- Typickým príkladom je vyhľadávanie reťazca, ktorý zodpovedá zadanej maske.
- Interpreter popisuje, ako definovať gramatiku jednoduchého jazyka, reprezentovať jednotlivé vety a následne ich interpretovať.

Gramatika

Výraz ::= Literál | Voľba | Postupnosť |
Opakovanie | '(' Výraz ')'

Voľba ::= Výraz '|' Výraz

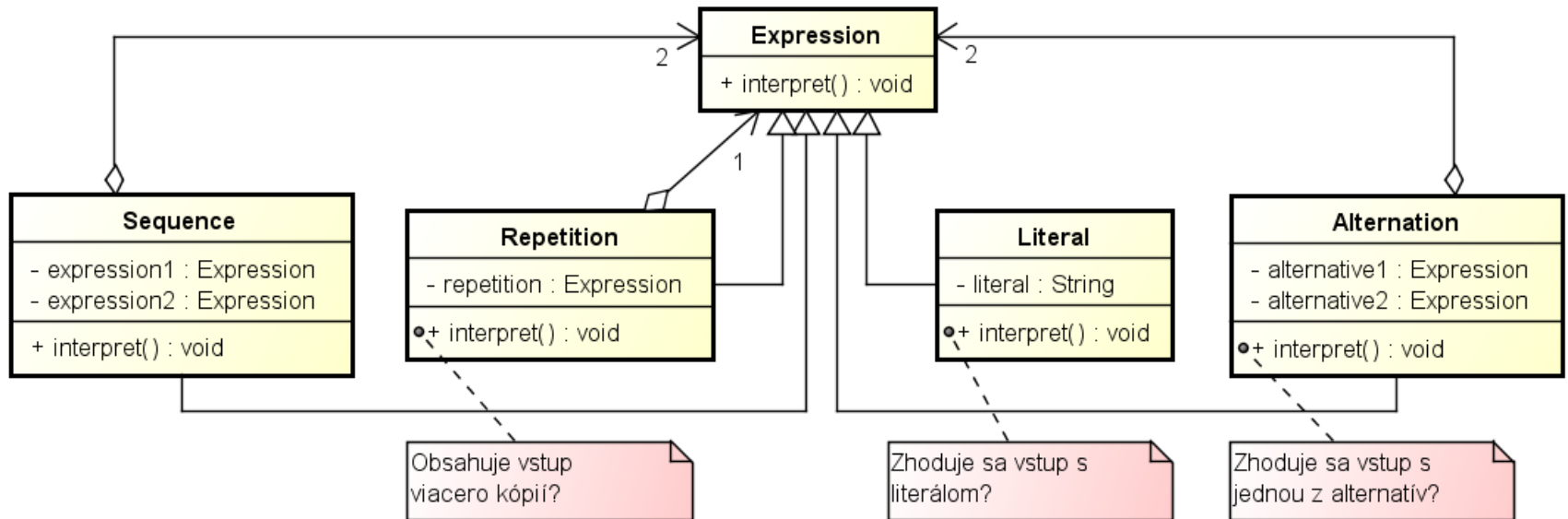
Postupnosť ::= Výraz '&' Výraz

Opakovanie ::= Výraz '*'

Literál ::= 'a' | 'b' | ... { 'a' | 'b' | ... }*



Gramatika

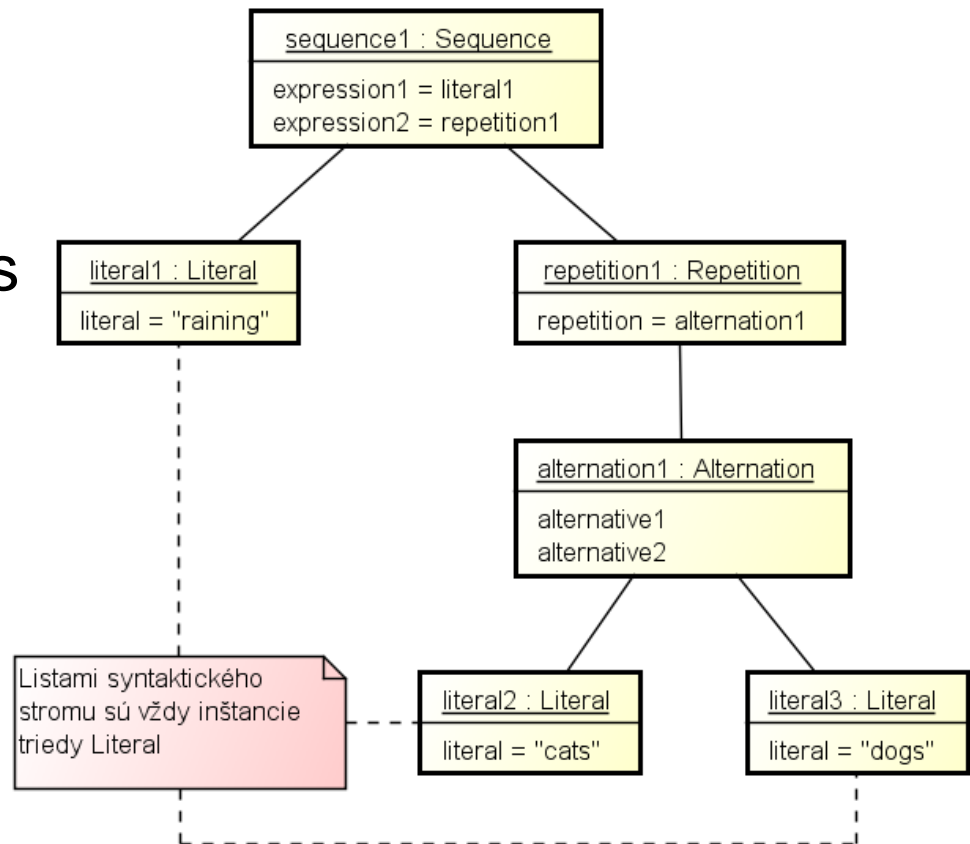


powered by Astah

Syntaktický strom

raining & (cats | dogs) *

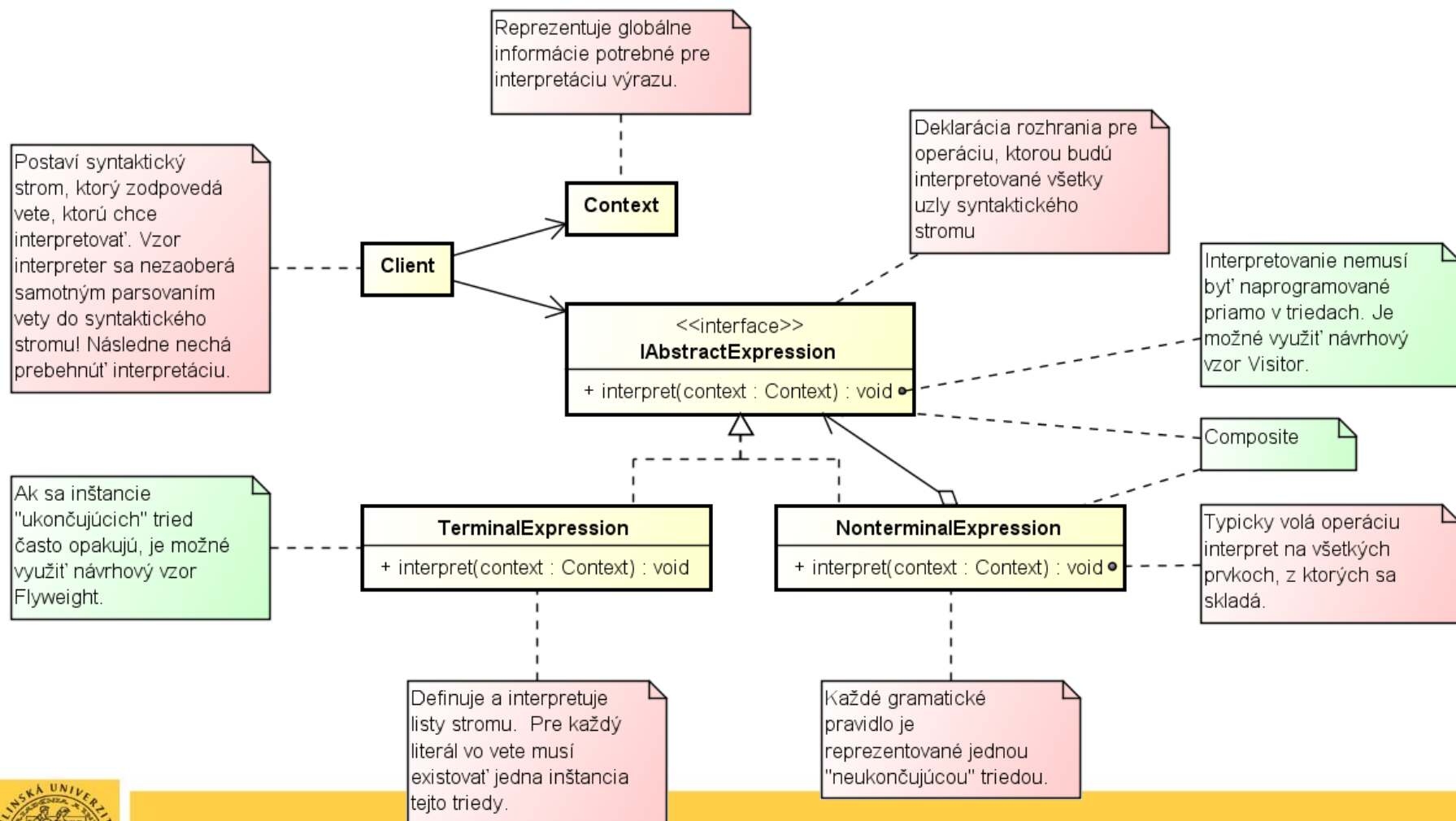
- Vie overiť, či mu zodpovedajú nasledujúce výrazy:
 - ✓ Raining cats cats dogs
 - ✓ Raining
 - ✓ Raining dogs dogs
- Raining raining cats
- Raining dogs raining



Aplikovateľnosť

- Interpréter je vhodné použiť keď existuje jazyk, v ktorom sa dajú jednoducho previesť jeho výrazy do formy syntaktického stromu a keď:
 - Existuje jednoduchá gramatika jazyka. Zložité hierarchie spôsobia vytvorenie komplikovanej štruktúry tried syntaktického stromu.
 - Efektívnosť nie je kritickým prvkom pri návrhu systému. Interprétery často netransformujú vety do syntaktického stromu priamo (najskôr použijú pomocné štruktúry, ktoré opäť interpretujú), čo samotné interpretovanie spomalí.

Implementácia



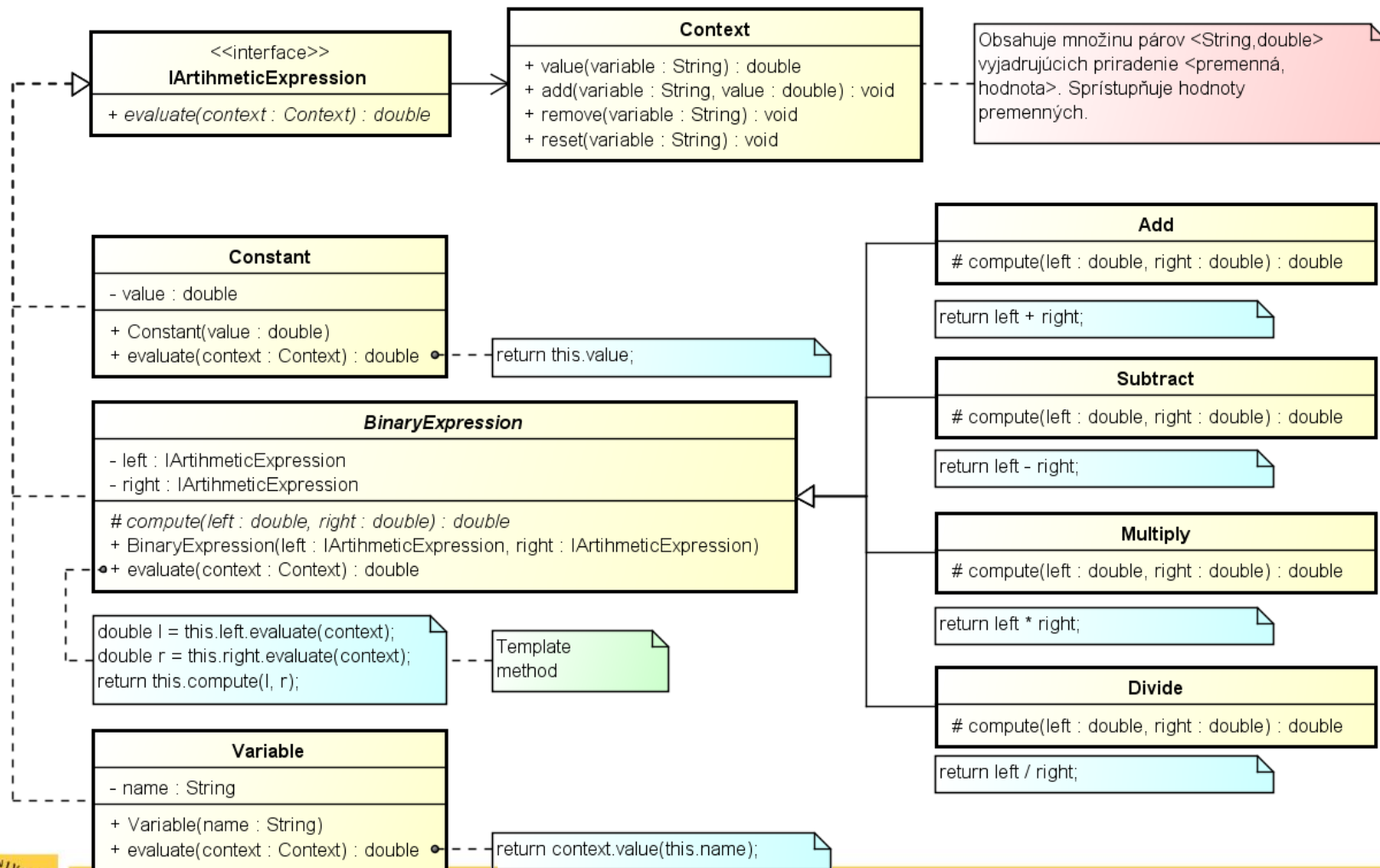
Dôsledky

- Keďže je gramatika jazyka definovaná vo forme tried, je jednoduché ju implementovať a rozšíriť.
- Problémom sú komplexné gramatiky, ktoré je ťažké udržiavať.
- Rovnaký syntaktický strom je možné interpretovať rôznymi spôsobmi. Jedna možnosť je zavedenie ďalšej metódy do rozhrania `IAbstractExpression`, lepšia možnosť využíva návrhový vzor Návštevník.

Príklad

Konštantá ::= '# ' Číslo '# '
Premenná ::= Identifikátor
Poločlen ::= Premenná | Konštantá | '(' Výraz ') '
Člen ::= Premenná | '+' Poločlen | '-' Poločlen
Súčin ::= Člen | Súčin '*' Člen | Súčin '/' Člen
Súčet ::= Súčin | Súčet '+' Súčin | Súčet '-' Súčin
Výraz ::= Súčet

Príklad



Príbuzné vzory

- *Composite* – syntaktický strom je inštanciou tohto vzoru.
- *Flyweight* – umožňuje efektívne zdieľanie listov stromu.
- *Iterator* – umožňuje prechádzať syntaktickým stromom.
- *Visitor* – definuje správanie každého uzla syntaktického stromu.

Upozornenie

- Tieto študijné materiály sú určené výhradne pre študentov predmetu 5I132 Návrhové vzory (Design Patterns) na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline.
- Reprodukovanie, šírenie (i častí) materiálov bez písomného súhlasu autora nie je dovolené.

Ing. Michal Varga, PhD.
Katedra informatiky
Fakulta riadenia a informatiky
Žilinská univerzita v Žiline
Michal.Varga@fri.uniza.sk

