Kompilátory

Cvičenie 5: Syntaktická analýza v ANTLR4 (2. časť)

Peter Kostolányi

14. novembra 2017

Prístup používaný doposiaľ

Spracovanie výstupu parsovania prostredníctvom „listenerov“
a „visitorov“:

Prístup používaný doposiaľ

Spracovanie výstupu parsovania prostredníctvom „listenerov“ a „visitorov“:

- Ideálne z pohľadu softvérového inžiniera

Prístup používaný doposiaľ

Spracovanie výstupu parsovania prostredníctvom „listenerov“ a „visitorov“:

- Ideálne z pohľadu softvérového inžiniera
- Gramatika je oddelená od hlavnej aplikácie kompilátora

Prístup používaný doposiaľ

Spracovanie výstupu parsovania prostredníctvom „listenerov“ a „visitorov“:

- Ideálne z pohľadu softvérového inžiniera
- Gramatika je oddelená od hlavnej aplikácie kompilátora
- Jednu gramatiku môžno použiť aj pri viacerých projektoch

Prístup používaný doposiaľ

Spracovanie výstupu parsovania prostredníctvom „listenerov“ a „visitorov“:

- Ideálne z pohľadu softvérového inžiniera
- Gramatika je oddelená od hlavnej aplikácie kompilátora
- Jednu gramatiku môžno použiť aj pri viacerých projektoch
- Zvyčajne ide o odporúčaný prístup

Atribúty a akcie

Alternatívne možno priamo do gramatiky zakomponovať kusy kódu, tzv. akcie:

Atribúty a akcie

Alternatívne možno priamo do gramatiky zakomponovať kusy kódu, tzv. akcie:

- Metóda podobná prekladovým schémam z Aho et al.

Atribúty a akcie

Alternatívne možno priamo do gramatiky zakomponovať kusy kódu, tzv. akcie:

- Metóda podobná prekladovým schémam z Aho et al.
- Pri malých projektoch môže byť jednoduchšie pridať akcie, než konštruovať „visitor“

Atribúty a akcie

Alternatívne možno priamo do gramatiky zakomponovať kusy kódu, tzv. akcie:

- Metóda podobná prekladovým schémam z Aho et al.
- Pri malých projektoch môže byť jednoduchšie pridať akcie, než konštruovať „visitor“
- Pri aplikáciách s vysokými požiadavkami na výkon môže byť výhodné explicitne nekonštruovať syntaktický strom

Atribúty a akcie

Alternatívne možno priamo do gramatiky zakomponovať kusy kódu, tzv. akcie:

- Metóda podobná prekladovým schémam z Aho et al.
- Pri malých projektoch môže byť jednoduchšie pridať akcie, než konštruovať „visitor“
- Pri aplikáciách s vysokými požiadavkami na výkon môže byť výhodné explicitne nekonštruovať syntaktický strom
- `parser.setBuildParseTree(false);`

Atribúty a akcie

Alternatívne možno priamo do gramatiky zakomponovať kusy kódu, tzv. akcie:

- Metóda podobná prekladovým schémam z Aho et al.
- Pri malých projektoch môže byť jednoduchšie pridať akcie, než konštruovať „visitor“
- Pri aplikáciách s vysokými požiadavkami na výkon môže byť výhodné explicitne nekonštruovať syntaktický strom
- `parser.setBuildParseTree(false);`
- Akcie sa po volaní `antlr4` preložia na kód v Jave, ktorý je pridaný do triedy parsera

Pridávanie záznamov do triedy parsera

```
1  grammar Gramatika;

3  import LexikalnePravidla;

5  @parser::header {
   import java.io.*;
7  import java.util.*;
   }
9
   @parser::members {
11     int pomocnaPremenna = 0;
       int nejakaMetoda(int cokolvek) {
13         ...
       }
15 }

17 init
       :     ...
19     ;

21 ...
```

Pridávanie záznamov do triedy parsera

```
1  grammar Gramatika;

3  import LexikalnePravidla;

5  @parser::header {
   import java.io.*;
7  import java.util.*;
   }
9
   @parser::members {
11     int pomocnaPremenna = 0;
       int nejakaMetoda(int cokolvek) {
13         ...
       }
15 }

17 init
       :    ...
19     ;

21 ...
```

- `@header` resp. `@members`: pridajú zodpovedajúce záznamy aj do triedy lexera

Atribúty a akcie

Vkladanie akcií do gramatiky:

Atribúty a akcie

Vkladanie akcií do gramatiky:

```
1 grammar Gramatika;

3 import LexikalnePravidla;

5 init
      :     NUM COMMA init EOF {System.out.println($NUM.int);}
7     |     NUM {System.out.println($NUM.int);}
      ;
9
  ...
```

Atribúty a akcie

Vkladanie akcií do gramatiky:

```
1 grammar Gramatika;

3 import LexikalnePravidla;

5 init
      :     NUM COMMA init EOF {System.out.println($NUM.int);}
7     |     NUM {System.out.println($NUM.int);}
      ;
9
  ...
```

- `$NUM.int` je referencia na preddefinovaný atribút `int` tokenu `NUM`

Atribúty a akcie

Vkladanie akcií do gramatiky:

```
1 grammar Gramatika;

3 import LexikalnePravidla;

5 init
      :     NUM COMMA init EOF {System.out.println($NUM.int);}
7     |     NUM {System.out.println($NUM.int);}
      ;
9
  ...
```

- `$NUM.int` je referencia na preddefinovaný atribút `int` tokenu `NUM`
- Prekladá sa na `Integer.valueOf(NUM().getText())`

Atribúty a akcie

Vkladanie akcií do gramatiky:

```
1 grammar Gramatika;

3 import LexikalnePravidla;

5 init
      :     NUM COMMA init EOF {System.out.println($NUM.int);}
7     |     NUM {System.out.println($NUM.int);}
      ;
9
  ...
```

- `$NUM.int` je referencia na preddefinovaný atribút `int` tokenu `NUM`
- Prekladá sa na `Integer.valueOf(NUM().getText())`
- Okrem preddefinovaných atribútov možno definovať aj vlastné, napríklad prostredníctvom `returns`

Atribúty a akcie

Vkladanie akcií do gramatiky:

```
1 grammar Gramatika;

3 import LexikalnePravidla;

5 init
      :     NUM COMMA init EOF {System.out.println($NUM.int);}
7     |     NUM {System.out.println($NUM.int);}
      ;
9
  ...
```

- `$NUM.int` je referencia na preddefinovaný atribút `int` tokenu `NUM`
- Prekladá sa na `Integer.valueOf(NUM().getText())`
- Okrem preddefinovaných atribútov možno definovať aj vlastné, napríklad prostredníctvom `returns`
- Zoznam všetkých preddefinovaných atribútov pre tokeny a neterminály možno nájsť v dokumentácii k ANTLR4

Sémantické predikáty

Umožňujú deaktivovať niektoré z pravidiel počas parsovania:

Sémantické predikáty

Umožňujú deaktivovať niektoré z pravidiel počas parsovania:

- Vhodné napríklad ak samotný vstup určuje syntaktickú štruktúru jeho zvyšku

Sémantické predikáty

Umožňujú deaktivovať niektoré z pravidiel počas parsovania:

- Vhodné napríklad ak samotný vstup určuje syntaktickú štruktúru jeho zvyšku
- Vhodné tiež na parsovanie viacerých verzií toho istého jazyka

Sémantické predikáty

Umožňujú deaktivovať niektoré z pravidiel počas parsovania:

- Vhodné napríklad ak samotný vstup určuje syntaktickú štruktúru jeho zvyšku
- Vhodné tiež na parsovanie viacerých verzií toho istého jazyka
- Popisná sila gramatík v ANTLR4 je vďaka nim veľmi veľká

Sémantické predikáty

Umožňujú deaktivovať niektoré z pravidiel počas parsovania:

- Vhodné napríklad ak samotný vstup určuje syntaktickú štruktúru jeho zvyšku
- Vhodné tiež na parsovanie viacerých verzií toho istého jazyka
- Popisná sila gramatík v ANTLR4 je vďaka nim veľmi veľká

```
1  vstupy
       :    vstup+
3      ;

5  vstup
       :    NUM postupnost[$NUM.int]
7      ;

9  postupnost[int n]
   locals [int i = 1;]
11     :    ({$i <= $n}? NUM {$i++;})*
       ;
```

Sémantické predikáty

Umožňujú deaktivovať niektoré z pravidiel počas parsovania:

- Vhodné napríklad ak samotný vstup určuje syntaktickú štruktúru jeho zvyšku
- Vhodné tiež na parsovanie viacerých verzií toho istého jazyka
- Popisná sila gramatík v ANTLR4 je vďaka nim veľmi veľká

```
1  vstupy
       :    vstup+
3      ;

5  vstup
       :    NUM postupnost[$NUM.int]
7      ;

9  postupnost[int n]
   locals [int i = 1;]
11     :    ({$i <= $n}? NUM {$i++;})*
       ;
```

- Používajú sa aj pri automatickom odstraňovaní ľavej rekurzie

Sémantické predikáty

Umožňujú deaktivovať niektoré z pravidiel počas parsovania:

- Vhodné napríklad ak samotný vstup určuje syntaktickú štruktúru jeho zvyšku
- Vhodné tiež na parsovanie viacerých verzií toho istého jazyka
- Popisná sila gramatík v ANTLR4 je vďaka nim veľmi veľká

```
1  vstupy
       :    vstup+
3      ;

5  vstup
       :    NUM postupnost[$NUM.int]
7      ;

9  postupnost[int n]
   locals [int i = 1;]
11     :    ({$i <= $n}? NUM {$i++;})*
       ;
```

- Používajú sa aj pri automatickom odstraňovaní ľavej rekurzie
- `antlr4 -Xlog Gramatika.g4`