Kompilátory

Cvičenie 7: Generovanie (medzi)kódu

Peter Kostolányi

28. novembra 2017

StringTemplate

- Knižnica na generovanie kódu resp. iného formátovaného textu

StringTemplate

- Knižnica na generovanie kódu resp. iného formátovaného textu
- Umožňuje definovať šablóny s atribútmi, za ktoré možno počas behu programu dosadzovať vhodný text

StringTemplate

- Knižnica na generovanie kódu resp. iného formátovaného textu
- Umožňuje definovať šablóny s atribútmi, za ktoré možno počas behu programu dosadzovať vhodný text
- Užitočné na generovanie LLVM (medzi)kódu

StringTemplate

- Knižnica na generovanie kódu resp. iného formátovaného textu
- Umožňuje definovať šablóny s atribútmi, za ktoré možno počas behu programu dosadzovať vhodný text
- Užitočné na generovanie LLVM (medzi)kódu

Stránka projektu:

StringTemplate

- Knižnica na generovanie kódu resp. iného formátovaného textu
- Umožňuje definovať šablóny s atribútmi, za ktoré možno počas behu programu dosadzovať vhodný text
- Užitočné na generovanie LLVM (medzi)kódu

Stránka projektu:

- `http://www.stringtemplate.org/`

StringTemplate

- Knižnica na generovanie kódu resp. iného formátovaného textu
- Umožňuje definovať šablóny s atribútmi, za ktoré možno počas behu programu dosadzovať vhodný text
- Užitočné na generovanie LLVM (medzi)kódu

Stránka projektu:

- `http://www.stringtemplate.org/`
- StringTemplate je súčasťou balíka ANTLR4

StringTemplate

- Knižnica na generovanie kódu resp. iného formátovaného textu
- Umožňuje definovať šablóny s atribútmi, za ktoré možno počas behu programu dosadzovať vhodný text
- Užitočné na generovanie LLVM (medzi)kódu

Stránka projektu:

- `http://www.stringtemplate.org/`
- StringTemplate je súčasťou balíka ANTLR4

Všeobecná dokumentácia k StringTemplate:

StringTemplate

- Knižnica na generovanie kódu resp. iného formátovaného textu
- Umožňuje definovať šablóny s atribútmi, za ktoré možno počas behu programu dosadzovať vhodný text
- Užitočné na generovanie LLVM (medzi)kódu

Stránka projektu:

- `http://www.stringtemplate.org/`
- StringTemplate je súčasťou balíka ANTLR4

Všeobecná dokumentácia k StringTemplate:

- `https://github.com/antlr/stringtemplate4/blob/master/doc/index.md`

StringTemplate

- Knižnica na generovanie kódu resp. iného formátovaného textu
- Umožňuje definovať šablóny s atribútmi, za ktoré možno počas behu programu dosadzovať vhodný text
- Užitočné na generovanie LLVM (medzi)kódu

Stránka projektu:

- `http://www.stringtemplate.org/`
- StringTemplate je súčasťou balíka ANTLR4

Všeobecná dokumentácia k StringTemplate:

- `https://github.com/antlr/stringtemplate4/blob/master/doc/index.md`

Dokumentácia k StringTemplate API:

StringTemplate

- Knižnica na generovanie kódu resp. iného formátovaného textu
- Umožňuje definovať šablóny s atribútmi, za ktoré možno počas behu programu dosadzovať vhodný text
- Užitočné na generovanie LLVM (medzi)kódu

Stránka projektu:

- `http://www.stringtemplate.org/`
- StringTemplate je súčasťou balíka ANTLR4

Všeobecná dokumentácia k StringTemplate:

- `https://github.com/antlr/stringtemplate4/blob/master/doc/index.md`

Dokumentácia k StringTemplate API:

- `http://www.stringtemplate.org/api/index.html`

„Hello, World!“ s použitím StringTemplate

Vytvorenie šablóny priamo v zdrojovom kóde:

„Hello, World!“ s použitím StringTemplate

Vytvorenie šablóny priamo v zdrojovom kóde:

```
1 import org.stringtemplate.v4.*;

3 public class Hello {
      public static void main(String[] args) {
5         ST hello = new ST("Hello, <name>!");
          hello.add("name", "world");
7         System.out.println(hello.render());
      }
9 }
```

„Hello, World!“ s použitím StringTemplate

Súbor `templates.stg` s definíciami šablón:

„Hello, World!“ s použitím StringTemplate

Súbor `templates.stg` s definíciami šablón:

```
1 group templates;
2
3 SomeTemplate(name, meno) ::= <<
4 Hello, <name>!
5 Po slovensky: Ahoj, <meno>!
6 >>
```

„Hello, World!“ s použitím StringTemplate

Súbor `templates.stg` s definíciami šablón:

```
1 group templates;
2
3 SomeTemplate(name, meno) ::= <<
4 Hello, <name>!
5 Po slovensky: Ahoj, <meno>!
6 >>
```

Načítanie šablóny zo súboru `templates.stg`:

„Hello, World!“ s použitím StringTemplate

Súbor `templates.stg` s definíciami šablón:

```
group templates;

SomeTemplate(name, meno) ::= <<
Hello, <name>!
Po slovensky: Ahoj, <meno>!
>>
```

Načítanie šablóny zo súboru `templates.stg`:

```
import org.stringtemplate.v4.*;

public class Hello2 {
    public static void main(String[] args) {
        STGroup group = new STGroupFile("templates.stg");
        ST template = group.getInstanceOf("SomeTemplate");
        template.add("name", "world");
        template.add("meno", "svet");
        System.out.println(template.render());
    }
}
```

„Skupiny“ šablón

Trieda STGroup:

„Skupiny“ šablón

Trieda STGroup:

- „Skupina“ šablón realizovaná napríklad súborom so šablónami alebo adresárom s niekoľkými súbormi

„Skupiny“ šablón

Trieda STGroup:

- „Skupina“ šablón realizovaná napríklad súborom so šablónami alebo adresárom s niekoľkými súbormi
- Metóda `ST getInstanceOf(String name)` vráti šablónu s požadovaným názvom

„Skupiny“ šablón

Trieda STGroup:

- „Skupina“ šablón realizovaná napríklad súborom so šablónami alebo adresárom s niekoľkými súbormi
- Metóda `ST getInstanceOf(String name)` vráti šablónu s požadovaným názvom

Trieda STGroupFile:

„Skupiny“ šablón

Trieda STGroup:

- „Skupina“ šablón realizovaná napríklad súborom so šablónami alebo adresárom s niekoľkými súbormi
- Metóda `ST getInstanceOf(String name)` vráti šablónu s požadovaným názvom

Trieda STGroupFile:

- Podtrieda triedy `STGroup`

„Skupiny“ šablón

Trieda STGroup:

- „Skupina“ šablón realizovaná napríklad súborom so šablónami alebo adresárom s niekoľkými súbormi
- Metóda `ST getInstanceOf(String name)` vráti šablónu s požadovaným názvom

Trieda STGroupFile:

- Podtrieda triedy `STGroup`
- Pre „skupiny“ šablón realizované jediným súborom (zvyčajne s príponou `.stg`)

„Skupiny“ šablón

Trieda STGroup:

- „Skupina“ šablón realizovaná napríklad súborom so šablónami alebo adresárom s niekoľkými súbormi
- Metóda `ST getInstanceOf(String name)` vráti šablónu s požadovaným názvom

Trieda STGroupFile:

- Podtrieda triedy `STGroup`
- Pre „skupiny“ šablón realizované jediným súborom (zvyčajne s príponou `.stg`)
- Súbor so šablónami treba špecifikovať ako argument konštruktora

Šablóny

Trieda ST:

Šablóny

Trieda ST:

- Najdôležitejšia trieda zodpovedajúca samotnej šablóne

Šablóny

Trieda ST:

- Najdôležitejšia trieda zodpovedajúca samotnej šablóne
- Metóda `ST add(String name, Object value)`:

Šablóny

Trieda ST:

- Najdôležitejšia trieda zodpovedajúca samotnej šablóne
- Metóda `ST add(String name, Object value)`:
  - „Prireťazí“ objekt `value` do atribútu s názvom `name`

Šablóny

Trieda ST:

- Najdôležitejšia trieda zodpovedajúca samotnej šablóne
- Metóda `ST add(String name, Object value)`:
  - „Prireťazí“ objekt `value` do atribútu s názvom `name`
  - Detaily v dokumentácii k API

Šablóny

Trieda ST:

- Najdôležitejšia trieda zodpovedajúca samotnej šablóne
- Metóda `ST add(String name, Object value)`:
  - „Prireťazí“ objekt `value` do atribútu s názvom `name`
  - Detaily v dokumentácii k API
- Metóda `String render()`:

Šablóny

Trieda ST:

- Najdôležitejšia trieda zodpovedajúca samotnej šablóne
- Metóda `ST add(String name, Object value)`:
  - „Prireťazí“ objekt `value` do atribútu s názvom `name`
  - Detaily v dokumentácii k API
- Metóda `String render()`:
  - Vráti šablónu s atribútmi nahradenými zodpovedajúcimi „postupnosťami zreťazených objektov“

Šablóny

Trieda ST:

- Najdôležitejšia trieda zodpovedajúca samotnej šablóne
- Metóda `ST add(String name, Object value)`:
  - „Prireťazí“ objekt `value` do atribútu s názvom `name`
  - Detaily v dokumentácii k API
- Metóda `String render()`:
  - Vráti šablónu s atribútmi nahradenými zodpovedajúcimi „postupnosťami zreťazených objektov“
  - Ak niektorý z objektov nie je znakový reťazec, použije sa jeho metóda `toString()`

Šablóny

Trieda ST:

- Najdôležitejšia trieda zodpovedajúca samotnej šablóne
- Metóda `ST add(String name, Object value)`:
  - „Prireťazí“ objekt `value` do atribútu s názvom `name`
  - Detaily v dokumentácii k API
- Metóda `String render()`:
  - Vráti šablónu s atribútmi nahradenými zodpovedajúcimi „postupnosťami zreťazených objektov“
  - Ak niektorý z objektov nie je znakový reťazec, použije sa jeho metóda `toString()`

V šablónach možno používať rôzne ďalšie „pokročilé prvky“:

Šablóny

Trieda ST:

- Najdôležitejšia trieda zodpovedajúca samotnej šablóne
- Metóda `ST add(String name, Object value)`:
  - „Prireťazí“ objekt `value` do atribútu s názvom `name`
  - Detaily v dokumentácii k API
- Metóda `String render()`:
  - Vráti šablónu s atribútmi nahradenými zodpovedajúcimi „postupnosťami zreťazených objektov“
  - Ak niektorý z objektov nie je znakový reťazec, použije sa jeho metóda `toString()`

V šablónach možno používať rôzne ďalšie „pokročilé prvky“:

- My z nich budeme používať predovšetkým podmienené atribúty

Šablóny

Trieda ST:

- Najdôležitejšia trieda zodpovedajúca samotnej šablóne
- Metóda `ST add(String name, Object value)`:
  - „Prireťazí“ objekt `value` do atribútu s názvom `name`
  - Detaily v dokumentácii k API
- Metóda `String render()`:
  - Vráti šablónu s atribútmi nahradenými zodpovedajúcimi „postupnosťami zreťazených objektov“
  - Ak niektorý z objektov nie je znakový reťazec, použije sa jeho metóda `toString()`

V šablónach možno používať rôzne ďalšie „pokročilé prvky“:

- My z nich budeme používať predovšetkým podmienené atribúty
- Detaily vo všeobecnej dokumentácii k StringTemplate