

Kompilator języka opartego o Ruby do HTML

Weronika Witek
wwitek@student.agh.edu.pl

21 stycznia 2023

1 Założenia programu

Projekt ma na celu stworzenie prostego języka Rhtml, kompilowanego do HTML (oraz kompilatora), o nieco przyjaźniejszej użytkownikowi składni i z możliwością używania pętli for, aby nie było konieczności wielokrotnego pisania tych samych linii.

Projekt został zrealizowany w języku Python, z wykorzystaniem biblioteki antlr4.

Programy napisane w Rhtml mają też takie rozszerzenie, .Rhtml.

2 Opis tokenów

2.1 Tokeny pętli

Token	Wartość	Znaczenie
LEFT	[Początek iteratora.
RIGHT]	Koniec iteratora.
ITER	iter	Oznaczenie, gdzie ma być wstawiana wartość z iteratora.
FOR	for	Początek pętli.
EACH	each	Fragment oznaczenia pętli.
LEFT	[Początek iteratora

2.2 Tokeny HTML

Token	Wartość	Znaczenie
HTML	html	Odpowiednik <code><html></code>
LIST	ul lub ol	Odpowiada listom nieuprządkowanym i uprządkowanym.
LIST_ITEM	li	Element listy
MAIN_TAG	body, footer, head, main lub nav	Główne tagi HTML, które mogą być tylko wewnątrz tagu <code><html></code>
DOUBLE_TAG	a, abbr, address, area, article, aside, b, bdi, bdo, blockquote, button, cite, del, details, div, em, h[1-6], header, hr, i, iframe, ins, kbd, mark, meta, meter, noscript, object, p, pre, progress, q, s, script, section, small, span, strong, sub, summary, sup, textarea, time, title, u, var lub wbr	Tag HTML, inny niż <code><html></code> lub tag główny, który ma początek i koniec (<code><></code> i <code>< / ></code>).
SINGLE_TAG	base, br, embed, img, link lub param	Pojedynczy tag HTML (samo <code><></code>).
TAG_OPTION	![a-z]+	Nazwa atrybutu tagu HTML.

2.3 Ogólne tokeny

Token	Wartość	Znaczenie
END	end	Koniec zawartości bloku (jak tag, czy pętla).
DO	end	Początek zawartości bloku (jak tag, czy pętla).
INT	0 — -?[1-9][0-9]*	Wartość int.
STRING	"[a-zA-Z0-9\n?!.,:;]*"	Wartość string.
WHITESPACE	[\t\n\r]	Fragment oznaczenia pętli.

3 Gramatyka formatu

```
1 // enter point for a program
  prog: html_expr EOF;
```

```

3 // integer range of for loop
5 range: LEFT INT '..' INT RIGHT;

7 // for loop, it iterates either through integers from number a to b, list
  of ints, or list of strings
  // inside of a for loop can be another for loop, a list or tag
9 for_loop: FOR (range | int_list | string_list) '..' EACH DO (tag_expr |
  for_loop | list_inside)+ END;

11 // list of integers
  int_list: LEFT int_elems? RIGHT;
13 int_elems: INT ( ',' INT)*;

15 // list of strings
  string_list: LEFT string_elems? RIGHT;
17 string_elems: STRING ( ',' STRING)*;

19 // html block
  html_expr: HTML DO main_tag_expr* END;
21
  // main tag block
23 main_tag_expr: MAIN_TAG DO (tag_expr | tag_expr | for_loop | list)* END;

25 // double or single tag block, both can have options, but only double can
  have a tag_inside
  tag_expr: (DOUBLE_TAG DO tag_option_expr* tag_inside* END) | (SINGLE_TAG
  DO tag_option_expr* END);
27
  // options of what can be inside a tag block: string or int, another tag
  block, a for loop, value from for loop on
29 // higher level or a list
  tag_inside: (string_int_inside | tag_expr | for_loop | ITER | list);
31
  // string or an int, for easier displaying in visitor
33 string_int_inside: STRING | INT;

35 // tag attribute
  tag_option_expr: TAG_OPTION '=' STRING;
37
  // unordered or ordered list block, can have only for loop or list_inside
39 list: LIST DO (for_loop | list_inside)* END;

41 // point of list
  list_inside: LIST_ITEM DO (string_int_inside | ITER) END;

```

4 Instrukcja obsługi

4.1 Wymagania

Uruchomienie kompilatora wymaga posiadania interpretera Python ≥ 3.8 wraz z zainstalowaną biblioteką antlr4-tools.

4.2 Używanie programu

Aby skompilować swój program, należy przejść do folderu Rhtml i wykonać polecenie:

```
$ python main.py input_path output_path
```

gdzie `input_path` to ścieżka do programu, a `output_path` to ścieżka do pliku wyjściowego (który zostanie nadpisany).

5 Przykład stosowania

Program `test.Rhtml`:

```
1  html do
    body do
3      div do
        !id = "car"
5        !name = "test"
        "something"
7        div do
            "test"
9            for [1, 10].each do
                h1 do
11                    iter
                    "car"
13                    iter
                end
15            end
        end
17    end
    img do
19    end
    for [1, 10].each do
21        h1 do
            iter
23            "car"
            iter
25        end
    end
27    div do
        "hey"
29    end
    ul do
31        for ["b", "c"].each do
            li do
33                iter
            end
35        end
    end
37    end
```

en

Wyjściowy plik html:

```
<html>
2 <body>
    <div id="car" name="test">
4         something
        <div>
6             test
            <h1>
8                 1
                car
10                 1
            </h1>
12            <h1>
                10
14            car
                10
16            </h1>
        </div>
18    </div>
    <img>
20    <h1>
        1
22        car
        1
24    </h1>
    <h1>
26        10
        car
28        10
    </h1>
30    <div>
        hey
32    </div>
    <ul>
34        <li>b</li>
        <li>c</li>
36    </ul>
    </body>
38 </html>
```