

Not so Swift Language

SP z předmětu KIV/FJP

Daniel Schnurpfeil & Jiří Trefil

Fakulta aplikovaných věd Západočeské univerzity v Plzni

6. prosince 2022

- Tvorba překladače zvoleného jazyka:
 - 👍 základní zadání (viz následující snímky)
 - 👍 vstup – značně zjednodušené konstrukce jazyka Swift
 - 👍 výstup – instrukce PL/0
 - ? technologie – PLY (Python lex-Yacc) / C++ (flex, bison)
 - 👍 silně typovaný jazyk
 - 👍 statická deklarace proměnných a konstant
 - 🤔 případné rozšíření od datový typ `bool`, `double`...

Konstrukce jazyka (1)

- definice celočíselných proměnných
 - př.: `{ var a: Int = 10 }`
- definice celočíselných konstant
 - př.: `{ let A: Int = 100 }`
- přiřazení
 - př.:

```
{  
  var a: Int = 10  
  var b: Int = a  
}
```

Konstrukce jazyka (2)

- základní aritmetika a logika
 - +, -, *, /, &&, ||, !, (), <, >, ==, <=, >=
- cyklus (while)
 - př.: { while (A > 1) {a = a + 1 + index} }
- definice procedury její volání
 - př.: {
 func count(number: Int) {
 number = number + 1
 }
 var a: Int = 10
 count(a)
}
- jednoduchou podmínku (if bez else)
 - př.: { if (a < 5) {a = a + 1} }

```
[program] := [dekl_list]
[dekl_list] := [dekl] || [dekl_list][dekl]
[dekl] := [var_dekl] || [fun_dekl]
[var_dekl] := [dtype] [var_list]
[var_list] := id || [var_list] , id
[fun_dekl] := func id ([params]) → [dtype] [comp_block]
[params] := [params],[type] id || [type] id || ee
[comp_block] := {[block]}
[block] := [exp_block] || [comp_block] || [iter_block] || [cond_block]
[exp_block] := [exp][block] || [exp]
[exp] := [var] = [value] || [call] || [var] *= [value] || ...
[value] := number || [num_expr]
[num_expr] := [num_expr] + [term] || +[term] || -[term]
[term] := [term] * [factor] || [term] / [factor]
[factor] := ([num_expr]) || id || number
[iter_block] := for([var]=[value], id[relation][value], [exp]) [block]
[cond_block] := if ([expr] [relation] [expr])[block]
[call] := id([args])
[args] := [args],[value] || [value] || ee
[value] := id || number
[dtype] := int || double || string
[var] := [type] id || id
[relation] := == || != || < || > || <= || >=
```

Regulární výrazy pro čísla a identifikátory

- `int:^[0-9]+$`
- `double:^[0-9]*[0-9]+$`
- `id: [A-Za-z][A-Za-z0-9_]*$`