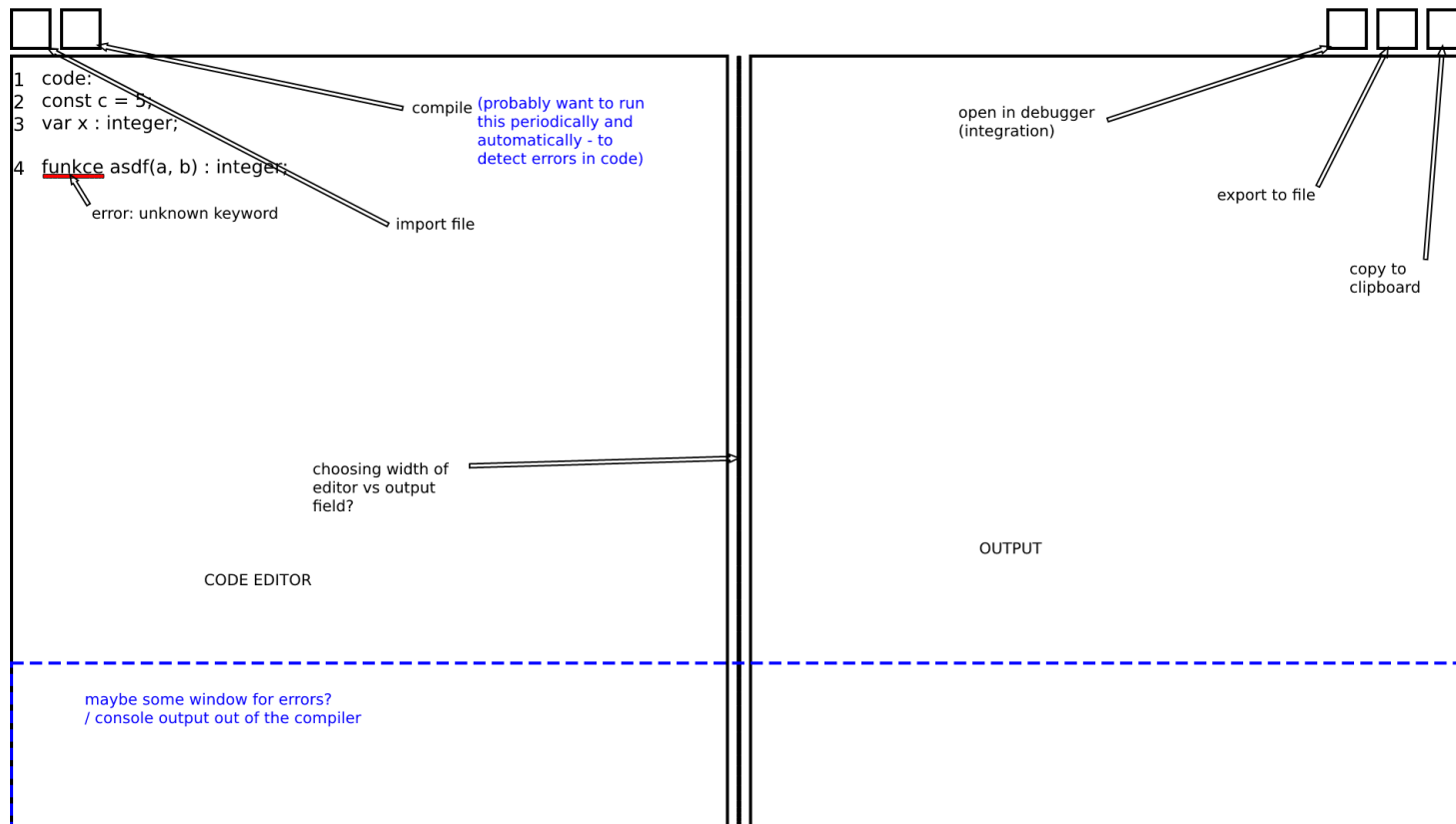


Webový překladač PL/0



Tomáš Linhart
Pavel Třeštík

Grafický návrh



Funkce překladače



- Překlad rozšířeného PL/0 jazyka na webu
- Náhrada předchozího PHP překladače
- Funkce:
 - Překlad programu na instrukce
 - Náповěda programátorovi
 - Lint
 - Práce se soubory – Načtení a uložení
 - Napojení na již existující debugger z předchozích let
- Využití alternativy yacc a lex pro JS: <https://github.com/GerHobbelt/jison>



Původní gramatika PL/0



```
program = block "." ;
block = [ "const" ident "=" number {"," ident "=" number} ";" ]
      [ "var" ident {"," ident} ";" ]
      { "procedure" ident ";" block ";" } statement ;
statement = [ ident ":=" expression
            | "call" ident
            | "?" ident
            | "!" expression
            | "begin" statement {"," statement} "end"
            | "if" condition "then" statement
            | "while" condition "do" statement ] ;
condition = "odd" expression |
            expression ("=" | "<" | "<=" | ">" | ">=") expression ;
expression = [ "+" | "-" ] term { ("+" | "-") term };
term = factor { ("*" | "/" ) factor };
factor = ident | number | "(" expression ")";
```

Změny v block



block =

```
[ "const" ident "=" number {"," ident "=" number} ";" ]
```

VS

```
[ "const" ident [ ":" data_type ] "=" value {"," ident [ ":" data_type ] "=" value } ";" ]
```

```
[ "var" ident {"," ident} ";" ]
```

VS

```
[ "var" ident [ ":" data_type ] {"," ident [ ":" data_type ] } ";" ]
```

```
{ "procedure" ident ";" block ";" } statement ;
```

VS

```
{ "procedure" ident [ "(" ident [ : data_type ] {"," ident [ : data_type ] } ")" ] ";" block ";" } statement ;
```

Změny ve statement



```
statement = [ ident "!=" {ident "!="} expression  
| "{" ident {, ident} "}" := {" value{, value} "}"  
| "call" ident  
| "?" ident  
| "!" expression  
| "begin" statement {";" statement } "end"  
| "if" condition "then" statement [ "else" statement ]  
| "(" condition ")" ? " "return" statement ":" "return" statement  
| "while" condition "do" statement  
| "for" number "to" number "do" statement  
| "foreach" ident "in" array_ident "do" statement ]  
| "return" value;
```

Změna ve factor



factor = ident | number | value | "("expression ");

Výsledná gramatika



```
program = block "." ;  
block = [ "const" ident [ ":" data_type ] "=" value { "," ident [ ":" data_type ] "=" value } ";" ]  
      [ "var" ident [ ":" data_type ] { "," ident [ ":" data_type ] } ";" ]  
      { "procedure" ident [ "(" ident [ : data_type ] { "," ident [ : data_type ] } ")" ] ";" block ";" } statement ;
```

```
statement = [ ident ":" { ident ":" } expression  
            | "{" ident { , ident } "}" := { " value{ , value } " }"  
            | "call" ident  
            | "?" ident  
            | "!" expression  
            | "begin" statement { ";" statement } "end"  
            | "if" condition "then" statement [ "else" statement ]  
            | "(" condition ")" ? " " "return" statement ":" "return" statement  
            | "while" condition "do" statement  
            | "for" number "to" number "do" statement  
            | "foreach" ident "in" array_ident "do" statement ]  
            | "return" value;
```

```
condition = "odd" expression  
           | expression ("="|"#"|"<"|<="|>"|>=") expression ;
```

```
expression = [ "+"|"-" ] term { ("+"|"-" ) term};  
term = factor { ("*"|"/" ) factor};  
factor = ident | number | value | "(" expression ")";
```




Děkujeme za pozornost