

Programování překladačů

Errata k učebnímu textu

Šárka Vavrečková
Ústav informatiky, FPF SU v Opavě
<http://fpf.slu.cz/~vav10ui>

Poslední aktualizace: 13. února 2009
Platí pro verzi skript vydanou na podzim roku 2008

Str. 24, kód:

```
procedure DejZnak; // zajistí posun na další znak ze vstupu
var i: byte;
begin
  if eof(zdroj) then
    znak.rad[delka] := #0 // zastupuje symbol konce souboru
  else with Znak do begin // je nutné načíst další řádek
    if delka = pozice then begin
      readln(zdroj, rad);
      rad := rad + ' ';
      inc(cislo);
      delka := length(rad);
      pozice := 1;
      for i := 1 to delka do
        rad [i] := UpCase(rad [i]); // převod na velká písmena
      end else inc(pozice); // ještě nejsme na konci řádku
    end;
  end;
end;
```

Str. 180 (příloha A), gramatika, oprava neterminálů I, J :

$S \rightarrow DbTe.$
 $D \rightarrow vI$ deklarace
 $I \rightarrow iJ$
 $J \rightarrow;$
 $J \rightarrow, I$

Str. 181 nahoře, totéž:

$J \rightarrow;$
 $J \rightarrow, I$

Str. 183, kód, za smyčkou while, opačný relační operátor:

```
if pozice > delka then stav := k_chyba; // případ, kdy slovo jen začíná jako
```

Str. 183, kód procedury Lex, doplnit řádek:

```
case znak.rad[znak.pozice] of
  #0: symbol.typ := S_ENDOFFILE;
  'A'..'Z': begin // identifikátor nebo klíčové slovo, délka > 0
```

Str. 186, kód nahoře, záměna symbolu:

```
procedure S;  
begin  
  if symbol.typ = S_VAR then begin
```

Str. 186, kód funkcí pro neterminály *I, J*:

```
procedure I;  
begin  
  if symbol.typ = S_ID then begin  
    Pridaj(symbol.atribstr);  
    expect(S_ID);  
    J;  
  end else  
    error('místo symbolu '+VypisHodn(sym)+' očekáván '+VypisTyp(S_ID));  
end;  
  
procedure J;  
begin  
  case symbol.typ of  
    S_CARKA: begin  
      expect(S_CARKA);  
      I;  
    end;  
    S_STREDNIK: expect(S_STREDNIK);  
    else error('místo '+VypisHodn(sym)+' očekáván '+VypisTyp(S_CARKA)+' nebo '  
      +VypisTyp(S_STREDNIK));  
  end;  
end;
```