Μορφοποίηση Κώδικα με το ANTLR

Διαλέξεις στο μάθημα: Μεταφραστές II

Γεώργιος Μανής

Περιγραφή απαιτήσεων

- Να μην υπάρχουν κενές γραμμές στον κώδικα.
- Τα άγκιστρα να ανοίγουν στην επόμενη γραμμή που βρίσκονται τα `if`, `else` και `while` και στοιχισμένα κάτω από αυτά.
- Τα άγκιστρα να κλείνουν σε δική τους ξεχωριστή γραμμή, στοιχισμένα με τα `if`, `else`, `while` και το άνοιγμα των αγκίστρων.
- Το μπλοκ κώδικα που υπάρχει μέσα στα άγκιστρα να βρίσκεται ένα tab πιο μέσα σε σχέση με τα `if`, `else` και `while`. Το υπόλοιπο μπλοκ να έχει την ίδια στοίχιση. Αν μέσα στο μπλοκ εμφανιστούν φωλιασμένα άγκιστρα, τότε αυτά ακολουθούν τους ίδιους κανόνες, οπότε το φωλιασμένο μπλοκ βρίσκεται ακόμα ένα tab πιο μέσα.
- Ανάμεσα στο `if`, `else` και το `while` και το άνοιγμα της παρένθεσης που τα ακολουθεί πρέπει να υπάρχει ένας κενός χαρακτήρας. Πριν και μετά το σύμβολο `:=` πρέπει να υπάρχει ένας κενός χαρακτήρας. Δεν υπάρχουν άλλοι κενοί χαρακτήρες στο πρόγραμμα.

Παράδειγμα

```
        program Beautiful
a :=-2; b:= 1;
    if (a=5 ) {
        a := 6
    }
else
        {
    if      (b>1
    )
        {
a :=1 +a }
;
while
        (a<10)
{
        a:= a*1;
            b :=b/   1
        }
            }; print(a)
```

```
program Beautiful
    a := -2;
    b := 1;
    if (a=5)
    {
        a := 6
    }
    else
    {
        if (b>1)
        {
            a := 1+a
        };
        while (a<10)
        {
            a := a*1;
            b := b/1
        }
    };
    print(a)
```

Γραμματική

```
grammar Beautifier;

startRule
    :   'program' ID statements
    ;

statements
    :   statements ';' statement
    |   statement
    ;

statement
    :   assignment_stat
    |   if_stat
    |   while_stat
    |   print_stat
    |
    ;
```

```
assignment_stat
    :   ID ':=' signed_expression
    ;

signed_expression
    :   op=('+'|'-') expression
    |   expression
    ;

if_stat
    :   'if' '(' expression RELAT_OPER expression')'
        '{'
            statements
        '}'
        else_part
    ;

else_part
    :   'else'
        '{'
            statements
        '}'
    |
    ;
```

Γραμματική

```
while_stat
    :   'while' '(' expression RELAT_OPER expression')'
        '{'
        statements
        '}'
    ;

print_stat
    :   'print' '('expression ')'
    ;
```

```
expression
    :   expression ('+'|'-') term
    |   term
    ;

term
    :   term ('*'|'/') factor
    |   factor
    ;

factor
    :   INTEGER
    |   ID
    |   '(' expression ')'
    ;
```

Γραμματική

```
INTEGER     :   [0-9]+ ('.' [0-9]+)? ([eE] [+-]? [0-9]+)?;
ID          :   [a-zA-Z]+;
RELAT_OPER  :   '<' | '>'  | '<>' | '<=' | '>=' | '=';
WS          :   [ \t\r\n] -> skip;
```

Members

```
@parser::members {

    # field with the current depth
    tabs = 0

    # line feed and print the necessary empty space
    def spaces(self):
        print()
        print(self.tabs*'  ',end='')

    # move indent one position to the right
    def indent(self):
        self.tabs = self.tabs + 1

    # move indent one position to the left
    def deindent(self):
        self.tabs = self.tabs - 1
}
```

Σημασιολογικοί κανόνες

```
startRule
    :   'program' ID        {print('\n\n\n\n\nprogram',$ID.text)}
                            {self.indent()                      }
        statements          {print('\n\n')                      }
    ;
```

Σημασιολογικοί κανόνες

```
statements
    :
        statements ';'      {print(';',end='')                    }
        statement
    |   statement
    ;
```

Σημασιολογικοί κανόνες

```
while_stat
    :   'while' '('                     {self.spaces()              }
                                        {print('while (',end='')     }
        expression
        op=RELAT_OPER                   {print($op.text,end='')      }
        expression')'                   {print(')',end='')          }
        '{'                             {self.spaces()              }
                                        {print('{',end='')          }
                                        {self.indent()              }
        statements
        '}'                             {self.deindent()            }
                                        {self.spaces()              }
                                        {print('}',end='')          }
    ;
```

Σημασιολογικοί κανόνες

```
if_stat
    :   'if' '('                    {self.spaces()              }
                                    {print('while (',end='')     }
        expression
        op=RELAT_OPER               {print($op.text,end='')      }
        expression')'               {print(')',end='')           }
        '{'                         {self.spaces()              }
                                    {print('{',end='')           }
                                    {self.indent()              }
        statements
        '}'                         {self.deindent()            }
                                    {self.spaces()              }
                                    {print('}',end='')           }
        else_part
    ;
```

Σημασιολογικοί κανόνες

```
else_part
    :   'else'                  {self.spaces()              }
                                {print('else',end='')        }
        '{'                     {self.spaces()              }
                                {print('{',end='')           }
                                {self.indent()              }
        statements
        '}'                     {self.deindent()            }
                                {self.spaces()              }
                                {print('}',end='')           }
    |
    ;
```

Σημασιολογικοί κανόνες

```
assignment_stat
    :   ID                          {self.spaces()              }
                                    {print($ID.text,end='')     }
        ':='                        {print(':=',end='')         }
        signed_expression
    ;
```

Σημασιολογικοί κανόνες

```
signed_expression
    :   op=('+'|'-')                {print($op.text,end='')     }
        expression
    |   expression
    ;
```

Σημασιολογικοί κανόνες

```
print_stat
    :   'print' '('                 {self.spaces()              }
                                    {print('print (',end='')     }
        expression ')'              {print(')',end='')          }
    ;
```

Σημασιολογικοί κανόνες

```
expression
    :   expression op=('+'|'-')     {print($op.text,end='')     }
        term
    |   term
    ;

term
    :   term op=('*'|'/')           {print($op.text,end='')     }
        factor
    |   factor
    ;

factor
    :   INTEGER                     {print($INTEGER.text,end='') }
    |   ID                          {print($ID.text,end='')      }
    |   '(' expression ')'
    ;
```