НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут ім. Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

##### Лабораторна робота №1

**«Розробка лексичного аналізатора»**

***з дисципліни "Теорія формальних мов і компіляції"***

**Варіант № 11**

Виконав: Худер К.Н.

#### Група: КП-81мп

Номер залікової книжки: 4118

Оцінка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

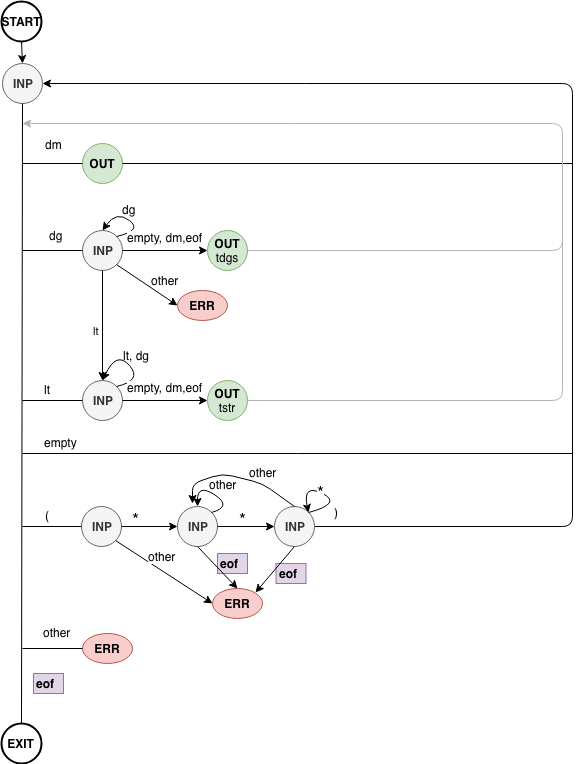
1 семестр 2018/2019

**Завдання**

Варіант 11:

1. <signal-program> --> <program>
2. <program> --> PROGRAM <procedure-identifier>; <block>
3. <block> --> <declarations> BEGIN <statements-list> END
4. <declarations> --> <label-declarations>
5. <label-declarations> --> LABEL <unsigned-integer> <labels-list>; | <empty>
6. <labels-list> --> , <unsigned-integer> <labels-list> | <empty>
7. <statements-list> --> <statement> <statements-list> | <empty>
8. <statement> --> <unsigned-integer> : <statement>; |  
   GOTO <unsigned-integer> ; |  
   LINK <variable-identifier> , <unsigned- integer> ; |  
   IN <unsigned-integer>; |  
   OUT <unsigned-integer>;
9. <variable-identifier> --> <identifier>
10. <procedure-identifier> --> <identifier>
11. <identifier> --> <letter><string>
12. <string> --> <letter><string> | <digit><string> | <empty>
13. <unsigned-integer> --> <digit><digits-string>
14. <digits-string> --> <digit><digits-string> | <empty>
15. <digit>-->0|1|2|3|4|5|6|7|8| 9
16. <letter>-->A|B|C|D|...|Z

**Граф автомату**



**Лістинг програми**

Codes.py

keywords = {  
 **'PROGRAM'**: 401,  
 **'BEGIN'**: 402,  
 **'END'**: 403,  
 **'LABEL'**: 404,  
 **'GOTO'**: 405,  
 **'LINK'**: 406,  
 **'IN'**: 407,  
 **'OUT'**: 408  
}  
empty = [9, 10, 11, 12, 13, 32]  
delims = [**':'**, **';'**, **'.'**, **','**]  
**def** is\_lt(c):  
 **return** c >= **'A' and** c <= **'Z'  
def** is\_dg(c):  
 **return** c >= **'0' and** c <= **'9'  
def** is\_dm(c):  
 **return** c **in** delims  
**def** is\_empty(c):  
 **return** ord(c) **in** empty  
**def** is\_keyword(str):  
 **return** str **in** keywords.keys()

DataSet.py

**class** DataSet(object):  
 **def** \_\_init\_\_(self, start\_id, max\_id):  
 self.id = start\_id  
 self.max = max\_id  
 self.set = {}  
 **def** add(self, value):  
 new\_id = self.get\_id\_by\_value(value)  
 **if not** new\_id:  
 self.id += 1  
 new\_id = self.id  
 **if** new\_id >= self.max:  
 **raise** Exception(**'DataSet overflow'**)  
 self.set.\_\_setitem\_\_(new\_id, value)  
 **return** new\_id  
 **def** get\_id\_by\_value(self, value):  
 **for** key, val **in** self.set.items():  
 **if** val == value:  
 **return** key  
 **return** False

Errors.py

**class** CustomException(Exception):  
 **def** \_\_init\_\_(self, position, value):  
 self.position = position  
 self.value = value  
**class** UnexpectedSymbolException(CustomException):  
 **def** \_\_str\_\_(self):  
 **return "Unexpected symbol: {0}. Line: {1}:{2}"**.format(self.value, self.position[0], self.position[1])  
**class** EndOfFileException(CustomException):  
 **def** \_\_str\_\_(self):  
 **return "End of File. Line: {0}:{1}"**.format(self.position[0], self.position[1])

Scanner.py

**class** Scanner:  
 **def** \_\_init\_\_(self, filename):  
 self.line = 1  
 self.column = 1  
 self.lineP = 1  
 self.columnP = 1  
 self.f = open(filename, **'r'**)  
 self.out = []  
 self.positions = []  
 self.c = **''  
 def** decreaseCol(self):  
 self.columnP -=1  
 **def** read(self, flag=False):  
 c = self.f.read(1)  
 self.c = c  
 **if** c:  
 **if** flag:  
 self.lineP = self.line  
 self.columnP = self.column  
 **if** ord(c) == 10:  
 self.line += 1  
 self.column = 1  
 **else**:  
 self.column += 1  
 **return** c  
 **def** append(self, code, res=**'none'**):self.out.append(code)  
 self.positions.append([self.lineP, self.columnP])  
 self.lineP = self.line  
 self.columnP = self.column  
 **def** exception(self, ExceptionClass, c=**''**):  
 **raise** ExceptionClass([self.lineP, self.columnP], c)  
 **def** \_\_exit\_\_(self, exc\_type, exc\_value, traceback):  
 self.f.close()

Main.py  
**def** translate(filename):  
 DIGITS = DataSet(500, 750)  
 STRINGS = DataSet(750, 1000)  
 scanner = Scanner(filename)  
 buf = **''  
 while** True:  
 **if** buf:  
 c = buf  
 buf = **''** scanner.decreaseCol()  
 **else**:  
 c = scanner.read(True)  
 **if not** c:  
 **break  
 if** is\_empty(c):  
 **continue  
 if** is\_dm(c):  
 scanner.append(ord(c), c)  
 **continue  
 if** is\_lt(c):  
 res = get\_string(c, scanner)  
 str = res[**'str'**]  
 buf = res[**'c'**]  
 **if** is\_keyword(str):  
 scanner.append(keywords[str], str)  
 **else**:  
 scanner.append(STRINGS.add(str), str)  
 **continue  
 if** is\_dg(c):  
 dgstr = c  
 is\_digit\_flag = True  
 **while** True:  
 c = scanner.read()  
 **if** is\_dg(c):  
 dgstr += c  
 **continue  
 elif** is\_dm(c) **or** is\_empty(c) **or not** c:  
 **break  
 elif** is\_lt(c):  
 res = get\_string(c, scanner)  
 dgstr += res[**'str'**]  
 buf = res[**'c'**]  
 is\_digit\_flag = False  
 **break  
 else**:  
 **raise** UnexpectedSymbolException([scanner.line, scanner.column - 1], c)  
  
 **if** is\_digit\_flag:  
 scanner.append(DIGITS.add(dgstr), dgstr)  
 **elif** is\_keyword(dgstr):  
 scanner.append(keywords[dgstr], dgstr)  
 **else**:  
 scanner.append(STRINGS.add(dgstr), dgstr)  
 **if** is\_dm(c):  
 buf = c  
 **continue  
 if** c == **'('**:  
 c = scanner.read(True)  
 **if** c == **'\*'**:  
 **while** True:  
 c = scanner.read(True)  
 **if** c == **'\*'**:  
 **while** c == **'\*'**:  
 c = scanner.read(True)  
 **if** c == **')'**:  
 flag = True  
 **break  
 else**:  
 **continue  
 elif not** c:  
 scanner.exception(EndOfFileException)  
 **else**:  
 **continue  
 if** flag:  
 **continue  
 else**:  
 **raise** UnexpectedSymbolException([scanner.line, scanner.column - 2], **'('**)  
 scanner.exception(UnexpectedSymbolException, c)  
 **return** {**'out'**: scanner.out, **'STRINGS'**: STRINGS.set, **'DIGITS'**: DIGITS.set, **'positions'**: scanner.positions}

**def** get\_string(c, f):  
 str = c  
 **while** True:  
 c = f.read()  
 **if** is\_lt(c) **or** is\_dg(c):  
 str += c  
 **continue  
 else**:  
 **break  
 return** {**'str'**: str, **'c'**: c}

**Контрольні приклади**

|  |  |
| --- | --- |
| **True - тести** | |
| **Test1** | |
| PROGRAM LAB1; | 1 1 401 PROGRAM  1 9 751 LAB1  1 13 59 ; |
| Strings | {751: ' LAB1 '} |
| Digits | {} |
| **Test2** | |
| PROGRAM TEST2;  (\* some comment \*)  LABEL 4118,  24,  05,  11  BEGIN   15:GOTO 17;  LINK MYVAR, 15;  IN 4118;  OUT 11;   234ABC;  END. | 1 1 401 PROGRAM  1 9 751 TEST2  1 14 59 ;  5 1 404 LABEL  5 7 501 4118  5 11 44 ,  6 9 502 24  6 11 44 ,  7 9 503 05  7 11 44 ,  8 9 504 11  10 1 402 BEGIN  12 5 505 15  12 7 58 :  12 8 405 GOTO  12 13 506 17  12 15 59 ;  13 5 406 LINK  13 10 752 MYVAR  13 15 44 ,  13 17 505 15  13 19 59 ;  14 5 407 IN  14 8 501 4118  14 12 59 ;  15 5 408 OUT  15 9 504 11  15 11 59 ;  17 5 753 234ABC  17 11 59 ;  19 1 403 END  19 4 46 . |
| Strings | {751: 'TEST2', 752: 'MYVAR', 753: '234ABC'} |
| Digits | {501: '4118', 502: '24', 503: '05', 504: '11', 505: '15', 506: '17'} |
| **False-тести** | |
| **Test3** | |
| PROGRAM TEST2;  (\* some comment \*)  LABEL 4118&,  24,  05,  BEGIN  END. | 1 1 401 PROGRAM  1 9 751 TEST2  1 14 59 ;  5 1 404 LABEL  Errors.UnexpectedSymbolException: Unexpected symbol: &. Line: 5:11 |
| Strings | {751: 'TEST2'} |
| Digits | {} |
| **Test4** | |
| PROGRAM TEST2;  (\* some comment \*)  LABEL 4118,  24,  05,  11  BEGIN   15:GOTO 17;  LINK MYVAR, 15;  IN 4118;  OUT 11;   234ABC;  END.  (\* kjbkjbkjbkjbg | 1 1 401 PROGRAM  1 9 751 TEST2  1 14 59 ;  5 1 404 LABEL  5 7 501 4118  5 11 44 ,  6 9 502 24  6 11 44 ,  7 9 503 05  7 11 44 ,  8 9 504 11  10 1 402 BEGIN  12 5 505 15  12 7 58 :  12 8 405 GOTO  12 13 506 17  12 15 59 ;  13 5 406 LINK  13 10 752 MYVAR  13 15 44 ,  13 17 505 15  13 19 59 ;  14 5 407 IN  14 8 501 4118  14 12 59 ;  15 5 408 OUT  15 9 504 11  15 11 59 ;  17 5 753 234ABC  17 11 59 ;  19 1 403 END  19 4 46 .  Errors.EndOfFileException: End of File. Line: 21:16 |
| Strings | {751: 'TEST2', 752: 'MYVAR', 753: '234ABC'} |
| Digits | {501: '4118', 502: '24', 503: '05', 504: '11', 505: '15', 506: '17'} |
| **Test5** | |
| PROGRAM TEST2; LABEL+ 4118 BEGIN END. | 1 1 401 PROGRAM  1 9 751 TEST2  1 14 59 ;  2 1 404 LABEL  Errors.UnexpectedSymbolException: Unexpected symbol: &. Line: 5:6 |
| Strings | {751: 'TEST2'} |
| Digits | {} |