


Νεκταρία Ευαγγελία Καλλιούπη

Ανάπτυξη Λεκτικού Αναλυτή σε JAVA

Ο κώδικας , παρέχεται στο αρχείο με όνομα LexicalAnalyzer.java και είναι εμπλουτισμένος με <<πυκνά>> σχόλια που περιγράφουν την διαδικασία με την οποία έχει γίνει η λεκτική ανάλυση .

Συγκεκριμένα ,στο πρόγραμμα αρχικά <<φορτώνουμε >> ένα text αρχείο ,το οποίο διαβάζουμε χαρακτήρα –χαρακτήρα .

```
File myFile= new File("C:\\Users\\nek\\Desktop\\NLP Project\\text.txt"); //Creation of File Descriptor for input file
FileReader readMyFile=new FileReader(myFile); //Creation of File Reader object
BufferedReader br=new BufferedReader(readMyFile); //Creation of BufferedReader object
```



Εκεί που δείχνει το κόκκινο βελάκι ,μπορούμε να προσθέσουμε /αλλάξουμε το μονοπάτι ,τοποθετώντας το μονοπάτι του αρχείου που θέλουμε να αναλύσουμε .

```
while((c = br.read()) != -1)           //Read char by Char
{
    char character = (char) c; //converting integer to char
```

Έχουμε 3 λίστες :

```
//list of sentences the file includes
ArrayList<String> sentences= new ArrayList<String>();
//list of words ,each sentence of the file has .
ArrayList<ArrayList<String>> wordsPerSentence=new ArrayList<ArrayList<String>>();
//words per sentence
ArrayList<String> words=new ArrayList<String>();
```

Ελέγχουμε για κάθε χαρακτήρα που διαβάζουμε αν ταυτίζεται με τον χαρακτήρα της τελείας που σηματοδοτεί το τέλος μια πρότασης .Αν όχι προσθέτουμε τον χαρακτήρα αυτό στην String μεταβλητή sentence και συνεχίζουμε στον επόμενο χαρακτήρα .Αν ναι προσθέτουμε την τελεία στην μεταβλητή sentence , στην συνέχεια προσθέτουμε την μεταβλητή sentence στην λίστα sentences και τέλος αρχικοποιούμε την μεταβλητή sentence σε null ,έτσι ώστε στην επόμενη επανάληψη να αποθηκεύσουμε σε αυτήν την επόμενη πρόταση .

```
int c = 0;
String sentence="";
System.out.println("Original Text from file :");
System.out.println("");

while((c = br.read()) != -1)           //Read char by Char
{

    char character = (char) c; //converting integer to char

    System.out.print(character); // print the whole text from file char by char

    if(String.valueOf(character).equals(".")) { // find the end of a sentence

        sentence=sentence+character; // add period to sentence
        sentences.add(sentence); //add sentence to sentences list
        sentence="";// initialize sentence variable to null in order to store a new sentence

    }else {

        sentence=sentence+character; //if not the end ,add character to sentence

    }

}

System.out.println("~~~~~");
```

Στην συνέχεια για όλες τις προτάσεις της λίστας sentences (δηλαδή για όλες τις προτάσεις του κείμενου που βρίσκεται στο αρχείο txt) ,ελέγχουμε για κάθε χαρακτήρα που διαβάζουμε αν ταυτίζεται με τον χαρακτήρα της τελείας που σηματοδοτεί το τέλος μια πρότασης και ίσως μιας λέξης ή αν ταυτίζεται με τον χαρακτήρα κενού. Αν όχι προσθέτουμε τον χαρακτήρα αυτό στην String μεταβλητή word και συνεχίζουμε στον επόμενο χαρακτήρα. Αν ναι προσθέτουμε την μεταβλητή word στην λίστα words και τέλος αρχικοποιούμε την μεταβλητή word σε null ,έτσι ώστε στην επόμενη επανάληψη να αποθηκεύσουμε σε αυτήν την επόμενη λέξη.

Τέλος προσθέτουμε την λίστα words που περιέχει όλες τις λέξεις(word) της πρότασης i (sentence i) στην λίστα wordsPerSentence στην οποία αποθηκεύουμε τις λίστες των λέξεων που περιέχει κάθε πρόταση .

```
for( int i=0;i< sentences.size();i++) { //loop as much times as the size of sentences list

    System.out.print("Sentence : ");
    System.out.println(sentences.get(i)); // print each sentence in sentences list
    System.out.println();

    String word=""; //here we store each word our sentence has

    for(int j=0;j<sentences.get(i).length();j++) {

        if( (String.valueOf(sentences.get(i).charAt(j)).equals(".")) || (String.valueOf(sentences.get(i).charAt(j)).equals(" "))) {

            words.add(word);
            word="";

        }else {

            word=word+sentences.get(i).charAt(j);

        }

    }

    wordsPerSentence.add(words); // add sentence's i words into wordsPerSentence list
```

Μέσα στον κώδικα υπάρχουν επιπλέον εντολές εκτύπωσης .

Παράδειγμα εκτέλεσης



Console

<terminated> LexicalAnalyzer [Java Application] C:\Program Files\Java\jdk1.8.0_121\bin\javaw.exe (23 Ιουλ 2020, 9:33:20 μ.μ.)

Lexical Analyzer

~~~~~

Original Text from file :

mary runs quickly.mary is tall.mary is slim.mary is blonde.mary gives john a dog.mary gives tomy a book.mary loves books.

~~~~~

Sentence : mary runs quickly.

Words : [mary,runs,quickly]

~~~~~

Sentence : mary is tall.

Words : [ mary,is,tall ]

~~~~~

Sentence : mary is slim.

Words : [mary,is,slim]

~~~~~

Sentence : mary is blonde.

Words : [ mary,is,blonde ]

~~~~~

Sentence : mary gives john a dog.

Words : [mary,gives,john,a,dog]

~~~~~

Sentence : mary gives tomy a book.

Words : [ mary,gives,tomy,a,book ]

~~~~~

Sentence : mary loves books.

Words : [mary,loves,books]

~~~~~