

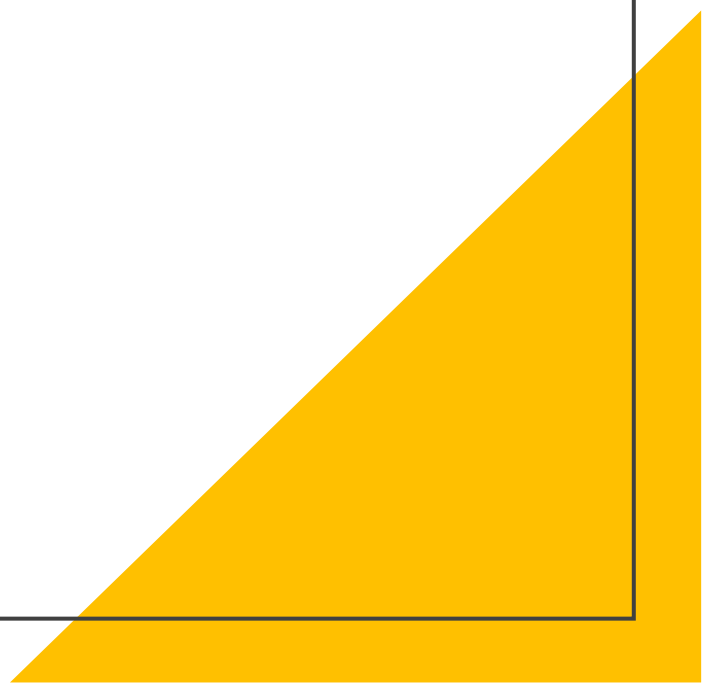
Επεξεργασία Φυσικής Γλώσσας

Τελική Εργασία

Κωνσταντίνος Καλογερόπουλος

Π19057

Θέμα 1



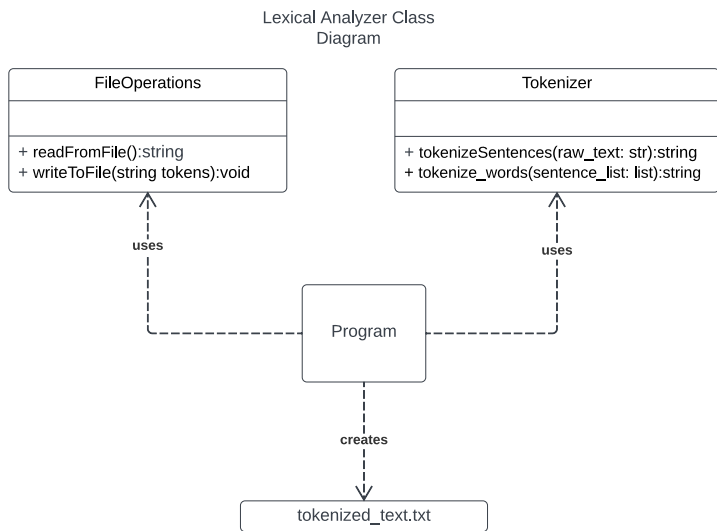
Εισαγωγή

- Η λεκτική ανάλυση (lexical analysis) είναι η διαδικασία που μετατρέπει μια ακολουθία από χαρακτήρες σε μια ακολουθία από λεκτικές μονάδες (tokens). Ένα πρόγραμμα ή συνάρτηση που κάνει λεκτική ανάλυση ονομάζεται λεκτικός αναλυτής (lexical analyzer, lexer ή scanner).
- Δημιουργία λεκτικού αναλυτή που με είσοδο μια μικρή ιστορία επιστρέφει τις επιμέρους προτάσεις της ιστορίας αυτής.

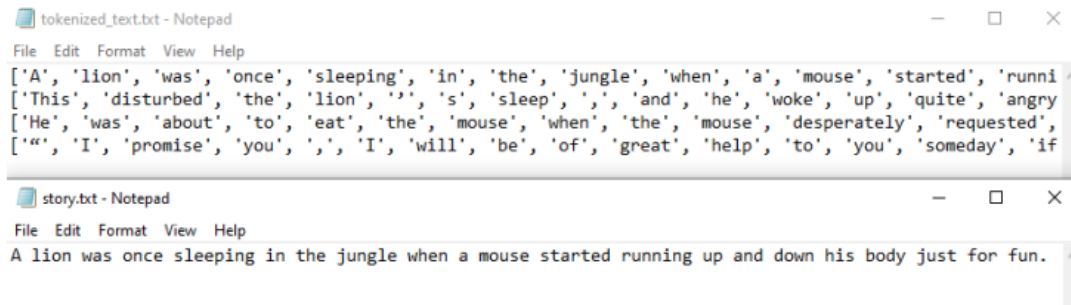
- Το πρόγραμμα αναπτύχθηκε σε python3.10.5. Ακόμα, έγινε χρήση της βιβλιοθήκης NLTK (Natural Language Toolkit). Το Εργαλείο Φυσικής Γλώσσας, ή πιο συχνά NLTK, είναι μια σειρά από βιβλιοθήκες και προγράμματα για συμβολική και στατιστική επεξεργασία φυσικής γλώσσας για τα Αγγλικά.
- Το πρόγραμμα αποτελείται από δύο κλάσεις όπως φαίνεται από το παρακάτω σχεδιάγραμμα στην επόμενη διαφάνεια.

Περιγραφή Προγράμματος

Lexical Analyzer Class Diagram



- Η κλάση FileOperations, περιέχει τις μεθόδους για το άνοιγμα και την αποθήκευση ενός αρχείου.
- Η κλάση Tokenizer, η οποία ενσωματώνει τις μεθόδους της βιβλιοθήκης NLTK, για τους σκοπούς της δημιουργίας μιας λίστας που περιέχει λέξεις και μιας λίστας που περιέχει προτάσεις.



```
tokenized_text.txt - Notepad
File Edit Format View Help
['A', 'lion', 'was', 'once', 'sleeping', 'in', 'the', 'jungle', 'when', 'a', 'mouse', 'started', 'runni
['This', 'disturbed', 'the', 'lion', '', 's', 'sleep', ',', 'and', 'he', 'woke', 'up', 'quite', 'angry
['He', 'was', 'about', 'to', 'eat', 'the', 'mouse', 'when', 'the', 'mouse', 'desperately', 'requested',
["", 'I', 'promise', 'you', ',', 'I', 'will', 'be', 'of', 'great', 'help', 'to', 'you', 'someday', 'if

story.txt - Notepad
File Edit Format View Help
A lion was once sleeping in the jungle when a mouse started running up and down his body just for fun.
```

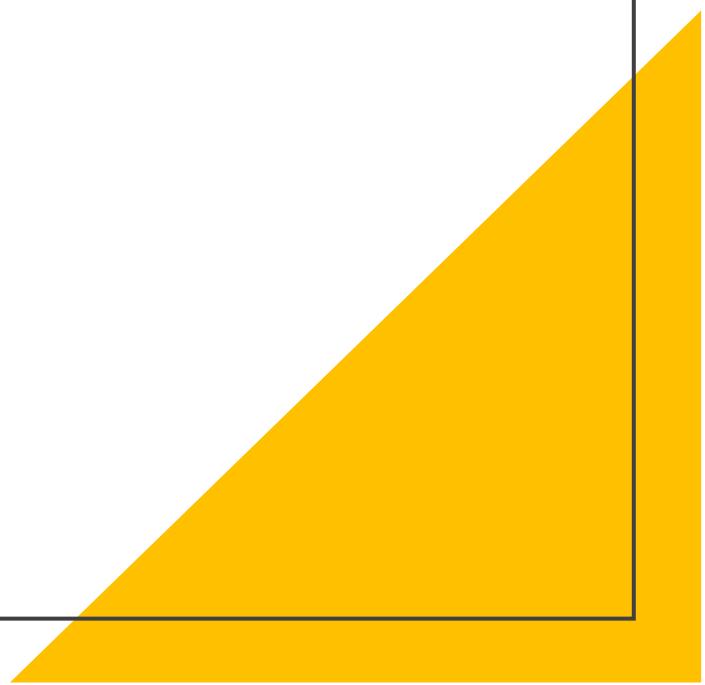
Εικόνα 3 Output αρχικής(κάτω) και τελικής(πάνω) ιστορίας μετά την λεκτική ανάλυση.

Εκτέλεση προγράμματος

- Λεκτική ανάλυση δοθέντος κειμένου από αρχείο. Έξοδος: tokenized_text.txt
- Κάτω δεξιά ακολουθεί μια εικόνα που αποτελεί ένα δείγμα του τελικού αρχείου που παράγεται σε σύγκριση με το αρχικό.

```
PS C:\Users\khalo\OneDrive\Documents\GitHub\Natural-L
alo/AppData/Local/Programs/Python/Python310/python.exe
cessing/Assignments/1/Lexical Analyzer/main.py"
Enter the file for Lexical Analysis: story.txt
Created tokenized text successfully.
```

Θέμα 2



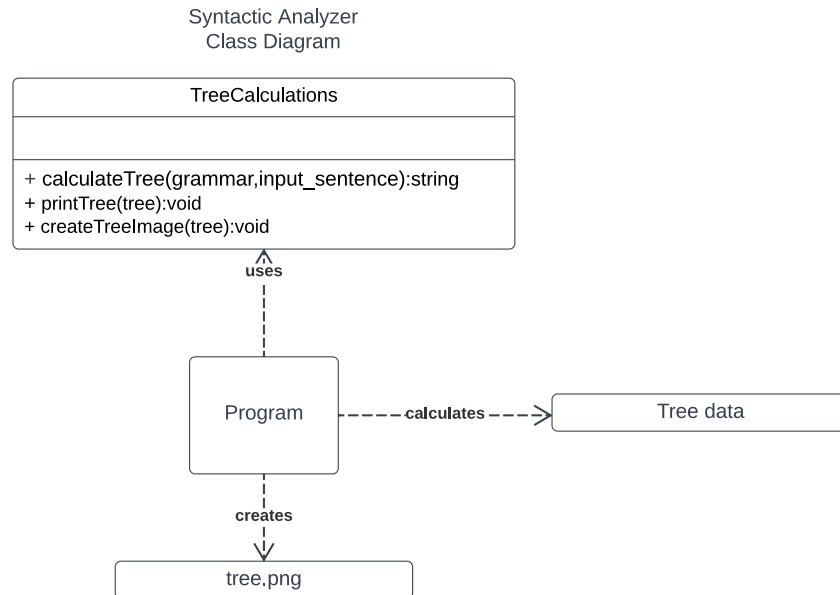
Εισαγωγή

- Ο συντακτικός αναλυτής υλοποιεί τη συντακτική ανάλυση μιας συγκεκριμένης γλώσσας. Αρχικά, δέχεται ως είσοδο ένα πρόγραμμα με τη μορφή ακολουθίας λεκτικών μονάδων. Έπειτα, ελέγχει αν το πρόγραμμα είναι σύμφωνο με τη γραμματική της γλώσσας που υλοποιεί (αν ανήκει στη συγκεκριμένη γλώσσα). Σε περίπτωση συντακτικού λάθους ενημερώνει τον χρήστη. Τέλος, παράγει το συντακτικό δέντρο που αντιστοιχεί στην ακολουθία εισόδου.
- Δημιουργία συντακτικού αναλυτή που με είσοδο μια πρόταση, θα παράγει το δέντρο της

Περιγραφή Προγράμματος

- Το πρόγραμμα αναπτύχθηκε σε python3.10.5. Ακόμα, έγινε χρήση της βιβλιοθήκης NLTK (Natural Language Toolkit). Επιπλέον, έγινε χρήση της βιβλιοθήκης Pillow και του διερμηνέα (interpreter) Ghostscript.
- Το πρόγραμμα αποτελείται από μια κλάση όπως φαίνεται από το παρακάτω σχεδιάγραμμα στην επόμενη διαφάνεια.

Lexical Analyzer Class Diagram



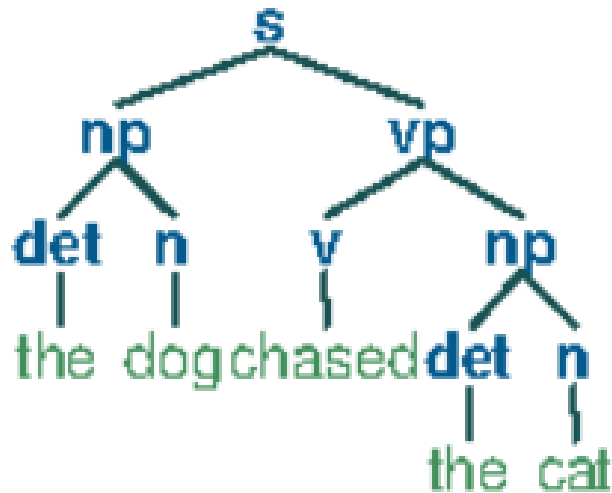
- Η κλάση TreeCalculations λαμβάνει τα δεδομένα για το δέντρο και ενσωματώνει τις κατάλληλες λειτουργίες για την δημιουργία του.

```
PS C:\Users\kkalo\OneDrive\Documents\GitHub\Natural-Language-Processing\kkalo/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/roccessing/Assignments/2/Syntactic Analyzer/main.py"
Syntactic Tree is:
(s (np (det the) (n dog)) (vp (v chased) (np (det the) (n cat)))))

Creating syntactic tree image file..

Converting file to PNG so we can view it..

Done! Locate the file as 'tree.png' in the working directory.
```

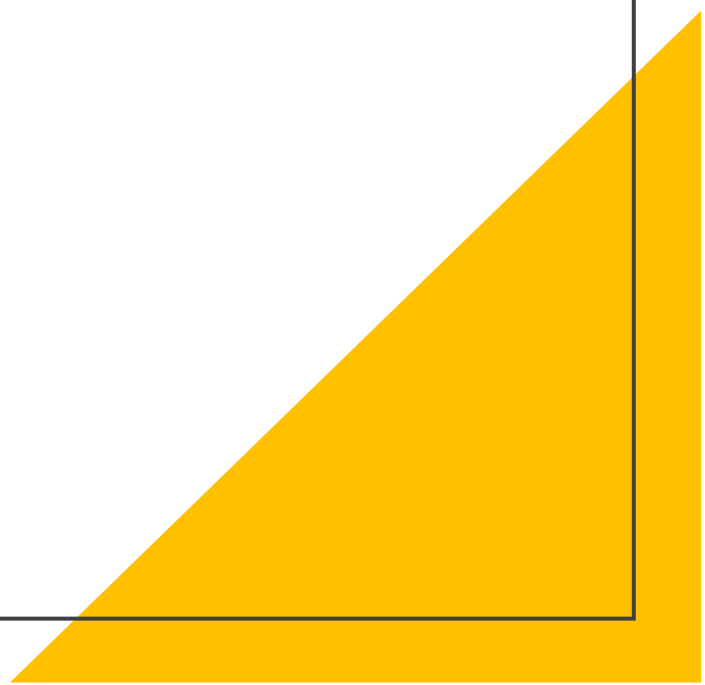


Εικόνα 6 Συντακτικό δέντρο της δοσμένης πρότασ

Εκτέλεση προγράμματος

- Το πρόγραμμα παράγει και εκτυπώνει το συντακτικό δέντρο με βάση τη γραμματική και την πρόταση που του δόθηκε.
- Στην συνέχεια δημιουργεί το δέντρο σε εικόνα μορφής PostScript(.ps) και τέλος το μετατρέπει σε εικόνα png.

Θέμα 3



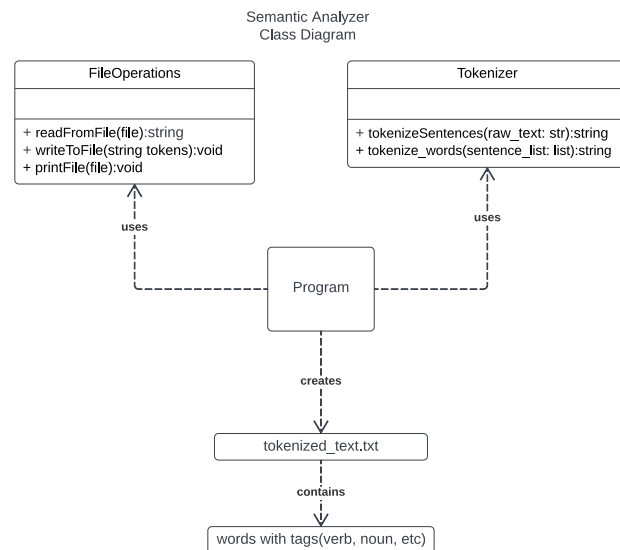
Εισαγωγή

- Η σημασιολογική ανάλυση είναι η διαδικασία εξαγωγής νοήματος από το κείμενο. Επιτρέπει στους υπολογιστές να κατανοούν και να ερμηνεύουν προτάσεις, παραγράφους ή ολόκληρα έγγραφα, αναλύοντας τη γραμματική τους δομή και προσδιορίζοντας τις σχέσεις μεταξύ μεμονωμένων λέξεων σε ένα συγκεκριμένο πλαίσιο (όπως σε κάποια πρόταση).
- Δημιουργία σημασιολογικού αναλυτή που με είσοδο μια πρόταση, θα παράγει το σημανόμενο της όπως είναι οι σχέσεις μεταξύ ρημάτων, ουσιαστικών, επιθέτων κ.λπ.

Περιγραφή Προγράμματος

- Το πρόγραμμα αναπτύχθηκε σε python3.10.5. Επίσης, έγινε χρήση της βιβλιοθήκης NLTK (Natural Language Toolkit).
- Το πρόγραμμα αποτελείται από δύο κλάσεις όπως φαίνεται από το παρακάτω σχεδιάγραμμα στην επόμενη διαφάνεια.

Lexical Analyzer Class Diagram



- Η κλάση **FileOperations**, περιέχει τις μεθόδους για το άνοιγμα και την αποθήκευση ενός αρχείου.
- Η κλάση **Tokenizer**, η οποία ενσωματώνει τις μεθόδους της βιβλιοθήκης **NLTK**, για τους σκοπούς της δημιουργίας μιας λίστας που περιέχει λέξεις και μιας λίστας που περιέχει προτάσεις.

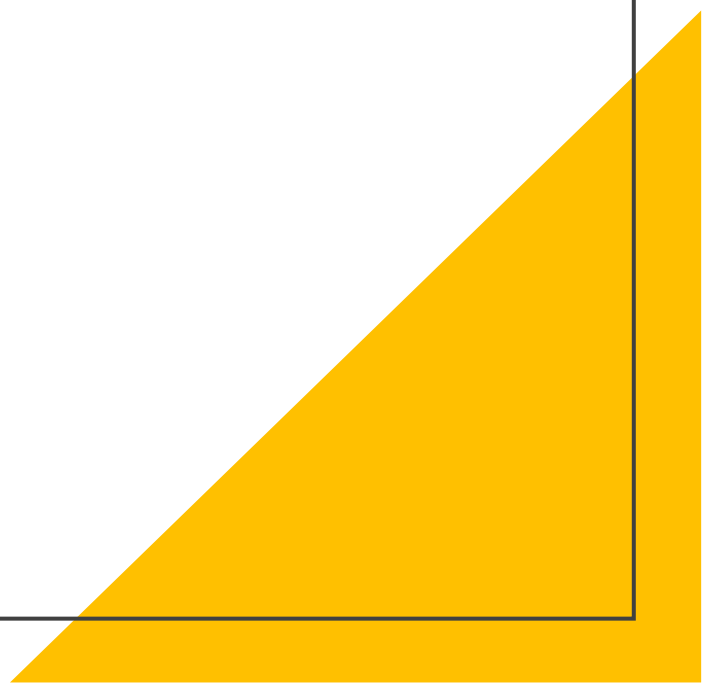
Εκτέλεση προγράμματος

- Όταν ξεκινάει η εκτέλεση του προγράμματος, ζητάει από τον χρήστη να εισάγει ένα κείμενο με προτάσεις.
- Στην συνέχεια, το πρόγραμμα το επεξεργάζεται και παράγει τα σημαινόμενα της κάθε πρότασης.
- Τέλος, τα αποθηκεύει στο αρχείο `tokenized_text.txt`, το οποίο προβάλλει στο output του προγράμματος ώστε να δει ο χρήστης το αποτέλεσμα της σημασιολογικής ανάλυσης.

```
yzer> & C:/Users/kka10/AppData/Local/Programs/Python/Python310/python.exe C:/Users/k
Documents/GitHub/Natural-Language-Processing/Assignments/3/Semantic Analyzer/main.py"
Enter the file for Semantic Analysis: story.txt
Created tokenized text successfully.
[('A', 'DET'), ('lion', 'NOUN'), ('was', 'VERB'), ('once', 'ADV'), ('sleeping', 'VERB'
), ('in', 'ADP'), ('the', 'DET'), ('jungle', 'NOUN'), ('when', 'ADV'), ('a', 'DET'), (
'mouse', 'NOUN'), ('started', 'VERB'), ('running', 'VERB'), ('up', 'ADV'), ('and', 'CO
NJ'), ('down', 'ADP'), ('his', 'PRON'), ('body', 'NOUN'), ('just', 'ADV'), ('for', 'AD
P'), ('fun', 'NOUN'), ('.', '.')]

```


Θέμα 4



Εισαγωγή

- Η βάση γνώσεων είναι ένα ολοκληρωμένο αποθετήριο που περιέχει τις πληροφορίες που απαιτούνται για την κατανόηση των ερωτήσεων και την απάντηση τους. Ακόμα δίνει τη δυνατότητα να βελτιστοποιηθεί η επεξεργασία φυσικής γλώσσας και η απάντηση ερωτήσεων για να επιτύχουμε ακρίβεια που μοιάζει όσο τον δυνατό πιο φιλική με τον άνθρωπο.
- Ανάπτυξη ενός προγράμματος για την πραγματοποίηση ερωταποκρίσεων σε μία βάση γνώσης. Οι ερωτήσεις και οι απαντήσεις θα δίνονται σε φυσική γλώσσα.

Περιγραφή Προγράμματος

- Το πρόγραμμα αναπτύχθηκε σε python3.10.5. Η βάση γνώσης στην python αντικατοπτρίζεται σε μια βάση δεδομένων. Για αυτό, έγινε χρήση της SQLite. Έτσι, εκτελούμε τις παρακάτω εντολές στο query tool για να δημιουργηθεί ο πίνακας στην βάση και οι στήλες του.
- Το πρόγραμμα αποτελείται από μια κλάση. Στην συνέχεια, θα προβληθεί το διάγραμμα κλάσεων.

```

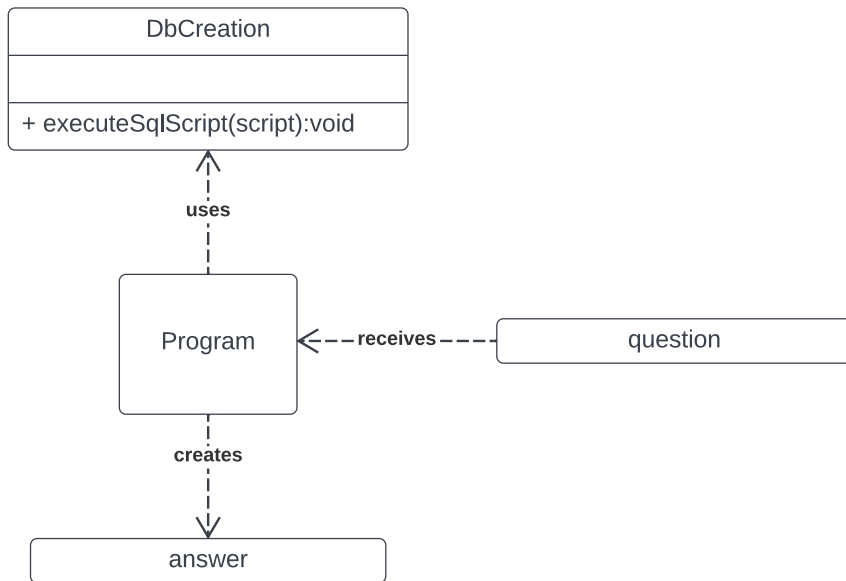
main.py U  knowledge_base.sql U X
knowledge_base.sql
1  CREATE TABLE [IF NOT EXISTS] knowledge
2  (
3      id integer default 1 constraint knowledge_pk primary
4      Name text,
5      Verb text,
6      Noun text,
7      Adjective text,
8      Intransitive_Verb text,
9      Transitive_Verb text,
10     Adverb text,
11     Receiver text
12 );
13
14 INSERT INTO knowledge (Name, Verb, Noun, Adjective, I
15 ('mary', null, 'dog', null, null, 'give', null, 'john
16 ('mary', null, 'book', null, null, 'give', null, 'tomy
17 ('mary', null, null, null, 'run', null, 'quickly', null)
18 ('mary', null, null, 'tall', null, null, null, null),
19 ('mary', null, null, 'slim', null, null, null, null),
20 ('mary', null, null, 'blonde', null, null, null, null),
21 ('mary', 'love', 'book', null, null, null, null, null),
22 ('dog', 'need', 'food', null, null, null, null, null),
23 ('cat', 'have', 'food', null, null, null, null, null),
24 ('dog', 'hate', 'cat', null, null, null, null, null),
25 ('dog', 'chase', 'cat', null, null, null, null, null),
26 ('cat', null, null, 'scary', null, null, null, null);

```

- Τα queries για την δημιουργία του πίνακα και των στηλών του

Lexical Analyzer Class Diagram

KnowledgeBase Application
Class Diagram



- Η κλάση DbCreation περιέχει την μέθοδο:
- `executeSqlScript(script):void`
- Η μέθοδος αυτή εκτελεί το SQL αρχείο με τις sql εντολές για την αρχικοποίηση του πίνακα knowledge στην βάση δεδομένων.
- Μετά, το πρόγραμμα χρησιμοποιεί τα δεδομένα στην βάση τα οποία επεξεργάζεται κάθε φορά μετά την ερώτηση του χρήστη σε αυτό.
- Στο τέλος, του επιστρέφει την απάντηση σε φυσική γλώσσα.

```
Examples of questions you can make to the NLP: 'who loves books', 'who gave dog', 'who hates cat'..
Enter your question: who loves books
mary
mary
mary
mary
mary
mary
mary
mary
mary
mary
mary
mary
mary
mary
mary
```

```
Loading grammar..
Examples of questions you can make to the NLP: 'who loves books', 'who gave dog', 'who hates cat'..
Enter your question: who hates cat
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
dog
```

Εκτέλεση προγράμματος

- Αρχικά, ο χρήστης πραγματοποιεί μια ερώτηση στο πρόγραμμα.
- Στην συνέχεια, το πρόγραμμα ψάχνει στην βάση γνώσης εάν υπάρχει απάντηση.
- Στην περίπτωση που υπάρχει, του επιστρέφει το αποτέλεσμα.