ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Π.Μ.Σ. «Προηγμένα Συστήματα Πληροφορικής - Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης»

Προηγμένα Θέματα Αντικειμενοστραφούς Προγραμματισμού

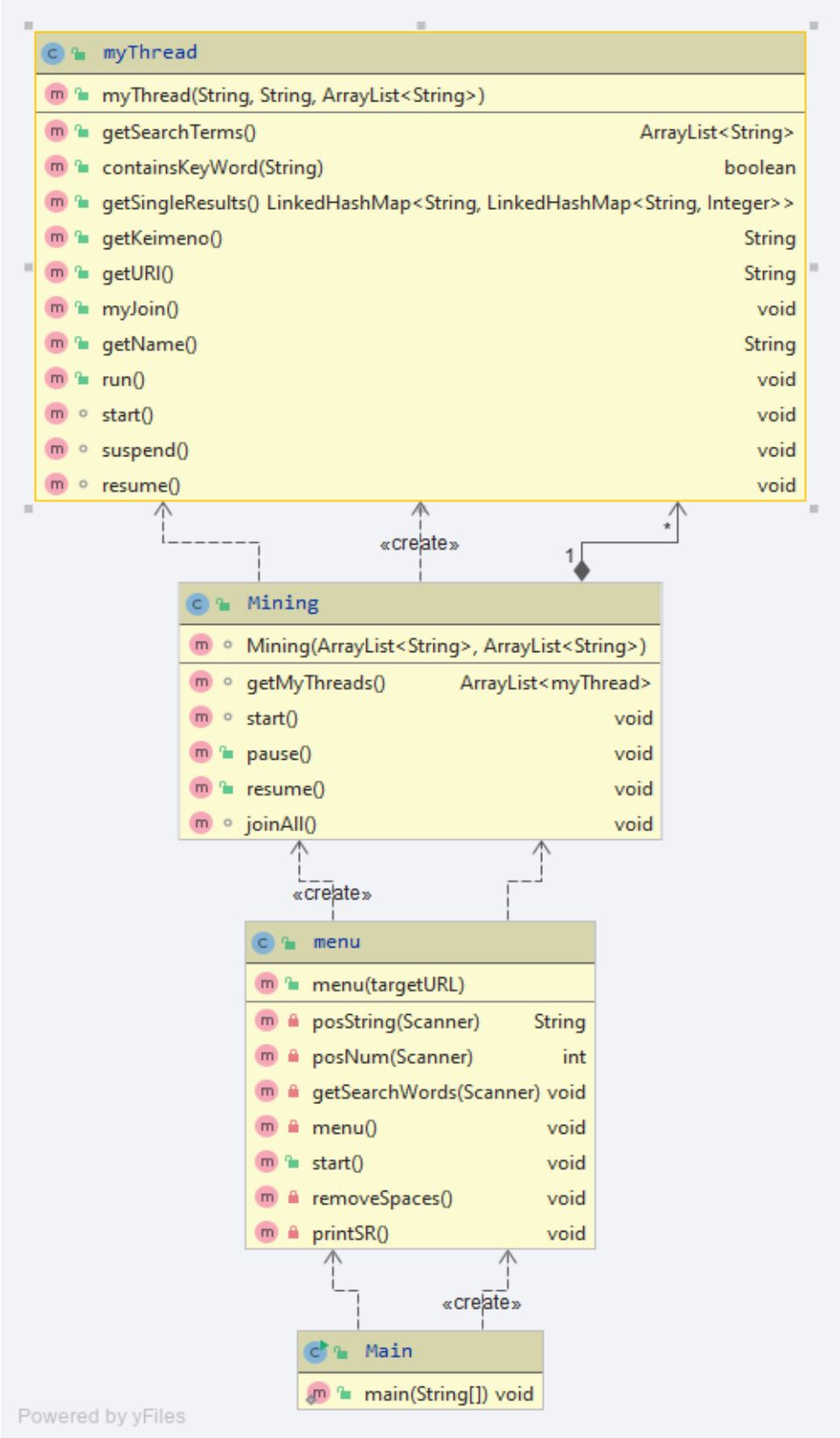
Εργασία: “Unipi Impression Miner”

Ζαριδάκης Αλέξανδρος 18013
Μπενάκης Νικόλαος 18016

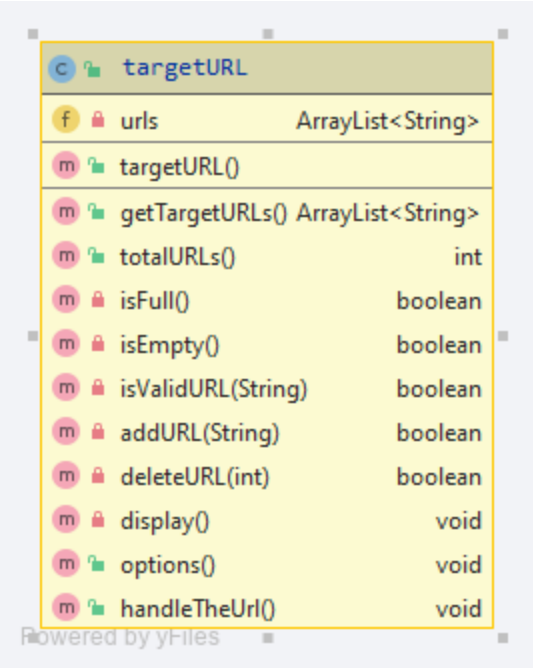
Καθηγητής: Ευθύμιος Αλέπης

Αθήνα
10/11/2019

Η εκτέλεση της εφαρμογής ξεκινάει από την κλάση Main.



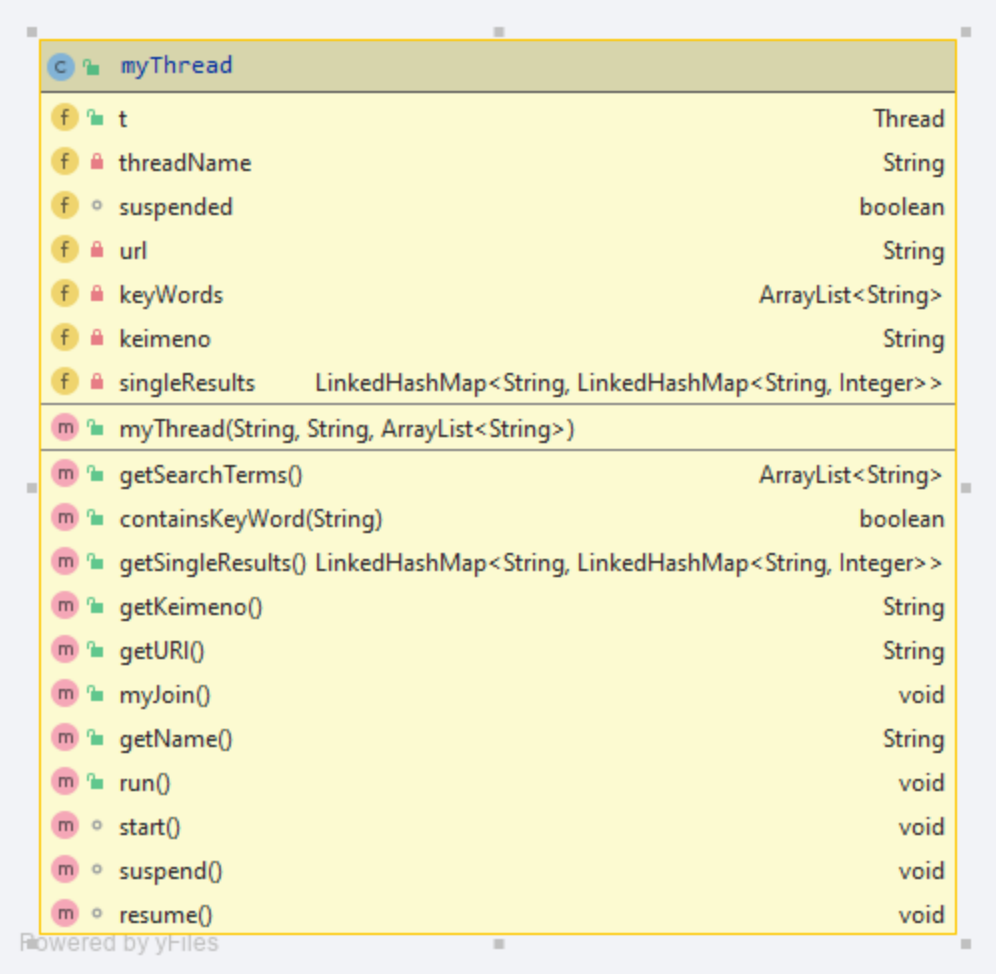
Μέσα στη main δημιουργείται και καλείται instance της κλάσης menu. Σε αυτή υπάρχουν όλες οι λειτουργίες για το αρχικό μενού που βλέπει ο χρήστης. Για τον σκοπό αυτό υπάρχουν συναρτήσεις που διαβάζουν και ελέγχουν για τυχόν λάθη εισόδου του χρήστη. Έχει υλοποιηθεί μηχανισμός για την αντιμετώπιση λάθος εισόδου από τον χρήστη όπου θα αναλυθεί αργότερα. Στο αρχικό μενού του χρήστη η επιλογή 1 είναι να αλλάξει τα URL στα οποία θα πραγματοποιηθεί η αναζήτηση. Για την διεκπεραίωση αυτής της λειτουργικότητας δημιουργήθηκε η κλάση targetURL:



Όλα τα url αποθηκεύονται σε μία λίστα και με κατάλληλες συναρτήσεις ο χρήστης έχει τη δυνατότητα να προσθέσει ή να διαγράψει διευθύνσεις. Ο μέγιστος αριθμός διευθύνσεων που μπορεί να έχει είναι 10. Έχουν υλοποιηθεί έλεγχοι για την εγκυρότητα των url καθώς και για άλλα λάθη που μπορεί να προκύψουν από την είσοδο του χρήστη. Επιστρέφοντας στο κύριο μενού η επόμενη επιλογή που δίνεται στο χρήστη είναι η εκκίνηση της διαδικασίας mining.

Για την διαδικασία mining υπάρχει αντίστοιχα η κλάση mining. Στον constructor της δέχεται την λίστα με όλα τα URL που θα γίνει αναζήτηση ως πρώτο όρισμα και ως δεύτερο παίρνει αντίστοιχα όλους τους όρους αναζήτησης. Επίσης, μέσα στον constructor της δημιουργεί τόσα στιγμιότυπα της κλάσης myThread όσο είναι και το πλήθος των URL που θα γίνει η αναζήτηση. Δηλαδή, κάθε ένα thread αναλαμβάνει ένα URL. Ακόμα, με χρήστη join η εφαρμογή περιμένει να τελειώσουν όλα τα thread την αναζήτησή τους και μετά συνεχίζει στη προβολή αποτελεσμάτων.

Η κλάση myThread:



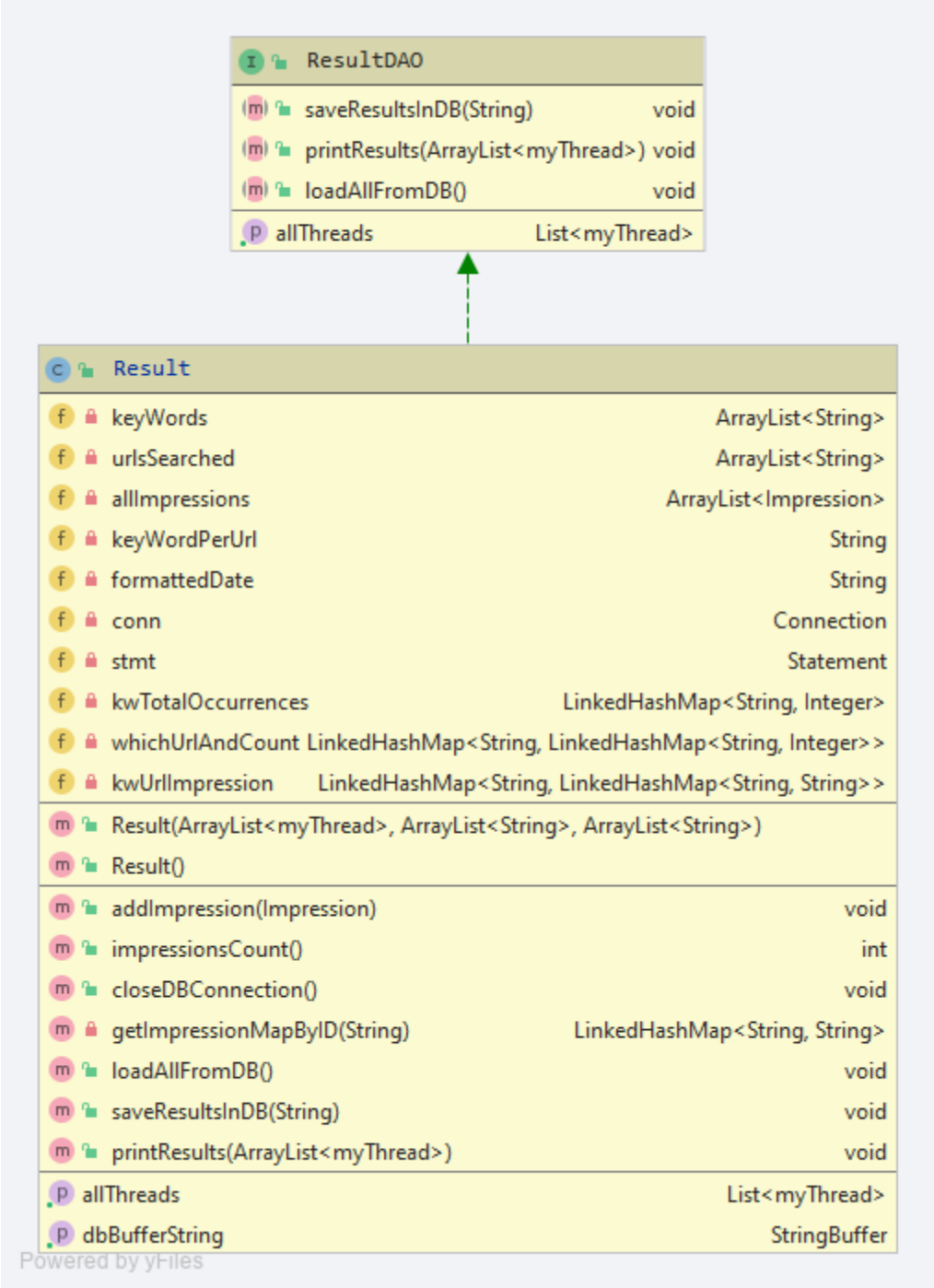
Στην κλάση αυτή γίνεται η αναζήτηση των στα url των λέξεων που έδωσε ο χρήστης. Αφού ολοκληρωθεί η αναζήτηση για την πρώτη λέξη αποθηκεύεται στη δομή

```
LinkedHashMap<String,LinkedHashMap <String,Integer> > singleResults;
```

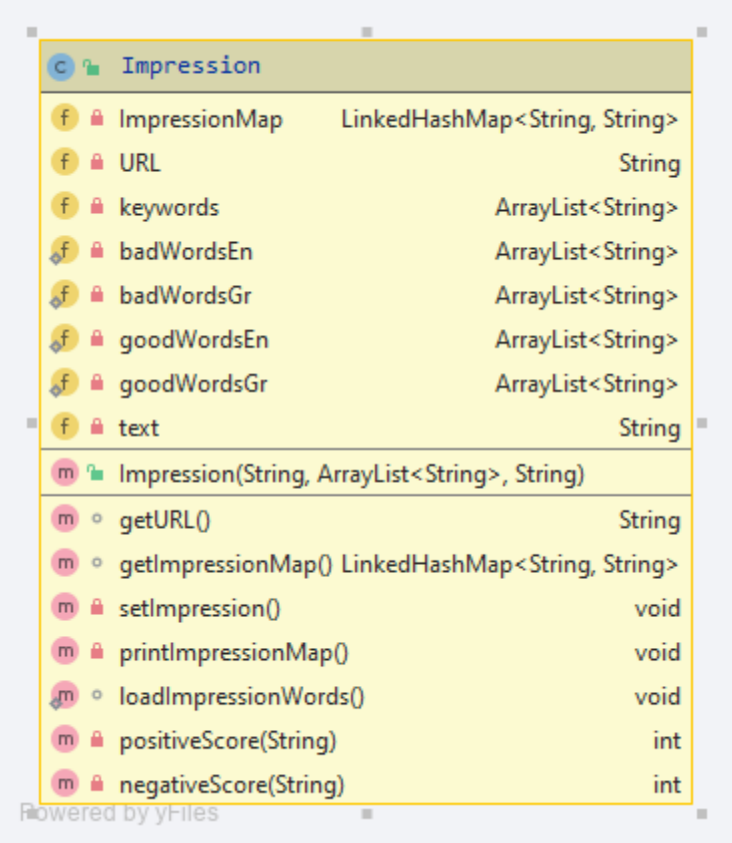
Στην map singleResults κάθε θέση για key το url στο οποίο έγινε η αναζήτηση και για value έχει ένα εσωτερικό map που έχει ως Key τον όρο αναζήτησης που έψαξε και value το πόσες φορές βρέθηκε. Έτσι, με αυτή την δομή το κάθε thread γνωρίζει πόσα keywords έχει βρει αλλά και πόσες φορές. Το ταυτοποίηση των όρων αναζήτησης γίνεται με χρήση regular expression. Μέρος κώδικα:

```
for(String tmpKey : keyWords){
    timesFound = 0;
    pattern = Pattern.compile(tmpKey);
    matcher = pattern.matcher(got);
    while (matcher.find())
        timesFound++;
    this.keimeno = got.toString();
    //System.out.println("Eimai to thread "+this.threadName+" brika to "+tmpKey+" sto "+this.url+" "+timesFo
    keyWordCount.put(tmpKey,timesFound);
}
singleResults.put(this.url,keyWordCount); // adding the results to map
for(Map.Entry<String, LinkedHashMap<String, Integer> > tmp : singleResults.entrySet()){
    if(tmp!=null){
        System.out.println("At url "+tmp.getKey()+"  found keywords:");
        if(!tmp.getValue().isEmpty()){
            tmp.getValue().forEach((k,v)-> System.out.println("Keyword: ->"+k+"<-  Found: "+v+" Times."));
        }
    }
}
```

Όταν η παραπάνω διαδικασία τελειώσει για όλα τα threads επιστρέφει η εκτέλεση στη κλάση menu. Επόμενο στάδιο εκτέλεσης είναι δημιουργία αντικειμένου Result.



Η κλάση Result υλοποιεί τις συναρτήσεις που έχουν οριστεί στο Interface ResultDao και αφορούν την πρόσβαση στη βάση δεδομένων της εφαρμογής. Δηλαδή, αποθήκευση της κάθε αναζήτησης αλλά και εμφάνιση όλων των αποτελεσμάτων σε περίπτωση που το ζητήσει ο χρήστης από το αρχικό μενού. Η βάση δεδομένων που χρησιμοποιήθηκε είναι postgres. Επιπλέον, για την εξαγωγή των αποτελεσμάτων η κλάση Result χρησιμοποιεί την κλάση Impression.



Αρχικά, η κλάση Impression χρησιμοποιεί τέσσερα βοηθητικά αρχεία. Συγκεκριμένα, badWordsEn, badWordsGr, GoodWordsEn, GoodWordsGr που το καθένα περιέχει αντίστοιχα «κακές και καλές» λέξεις για Ελληνικά και Αγγλικά. Τα περιεχόμενα αυτά των αρχείων φορτώνονται μία φορά στη κεντρική ραμ. Στη συνάρτηση, setImpression γίνεται αναζήτηση στο κείμενο του κάθε URL για τις λέξεις που όρισε ο χρήστης. Αλγοριθμικά, ανιχνεύεται η θέση της κάθε λέξης μέσα στο κείμενο και έπειτα ελέγχονται οι λέξεις δεκαπέντε θέσεις μετά και πριν(αν υπάρχουν τόσες λέξεις) για ανίχνευση κάποιας λέξεις από τα τέσσερα αρχεία καλών/κακών λέξεων. Υπάρχει εσωτερικό σκορ για κάθε μία λέξη και στο τέλος αποθηκεύεται με θετικές εντυπώσεις, αρνητικές ή ουδέτερες.