

# ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ



*Αναφορά στο μάθημα Ειδικά Θέματα Τεχνολογίας Λογισμικού του*

*Οικονόμου Χρήστου (ΑΜ 8120096)*

*Διδάσκων Καθ. Διομήδης Σπινέλλης*

## Περιεχόμενα

1. Κατανόηση και τεκμηρίωση του έργου.....	3
2. Πλάτος των αλλαγών.....	3
3. Ολοκλήρωση.....	3
4. Έλεγχος.....	6
5. Συνεργασία με την ομάδα ανάπτυξης.....	7
6. Σχετικοί Σύνδεσμοι.....	7

## 1. Κατανόηση και τεκμηρίωση του έργου

Η κατανόηση και η τεκμηρίωση του έργου ήταν πολύ σημαντική διότι το έργο είναι σημαντικά μεγάλο και πολύπλοκο. Θα μπορούσα να πω ότι η κατανόηση του έργου και του τρόπου αλληλεπίδρασης των τμημάτων του ήταν το 60% του χρόνου που αφιέρωσα στην εργασία. Ο κύριος λόγος είναι ότι υπάρχει χρήση 2 γλωσσών προγραμματισμού της Java και Groovy. Αξίζει σε αυτό το σημείο να αναφέρω ότι η συνεισφορά μου είναι να βοηθήσω στην κατανόηση του έργου μέσω της παρουσίασης της javadocs τεκμηρίωσης στο περιβάλλον του gremlin-console.

## 2. Πλάτος των αλλαγών

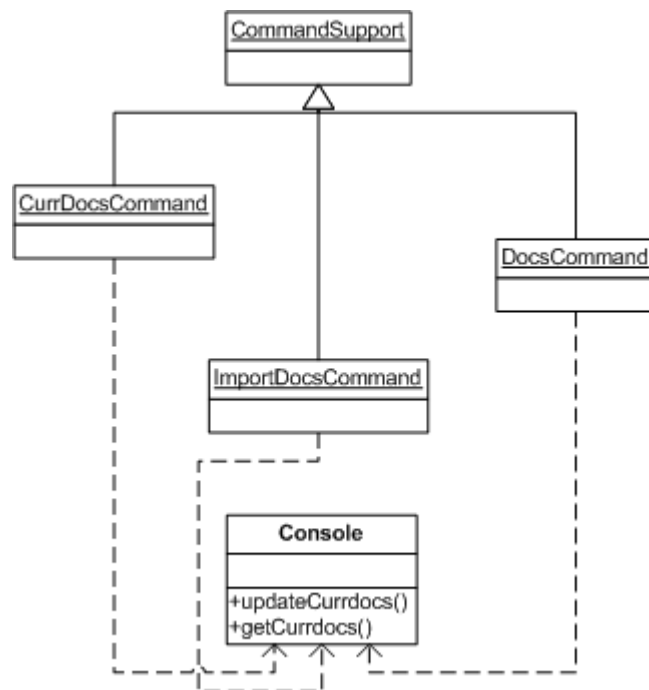
Το μέγεθος των αλλαγών που έγιναν στο έργο είναι η προσθήκη 828 γραμμών κώδικα. Πιο αφαιρετικά προσέθεσα ένα πακέτο, πιο συγκεκριμένα το `org.apache.tinkerpop.gremlin.console.groovy.DisplayDocs` (πιο συγκεκριμένα παρουσιάζεται πιο κάτω). Το πακέτο αυτό έχει ως σκοπό να «φορτώνει» την τεκμηρίωση javadoc. Επίσης προσέθεσα 3 εντολές groovy οι οποίες έχουν την ευθύνη να μεταβιβάζουν δεδομένα από την κονσόλα στις κλάσεις του πακέτου. Παράλληλα προσέθεσα ένα αρχείο που κάνει κάποια junits ώστε να βεβαιωθώ για την συμπεριφορά των μεθόδων. Συνολικά προστέθηκαν 7 αρχεία.

## 3. Ολοκλήρωση

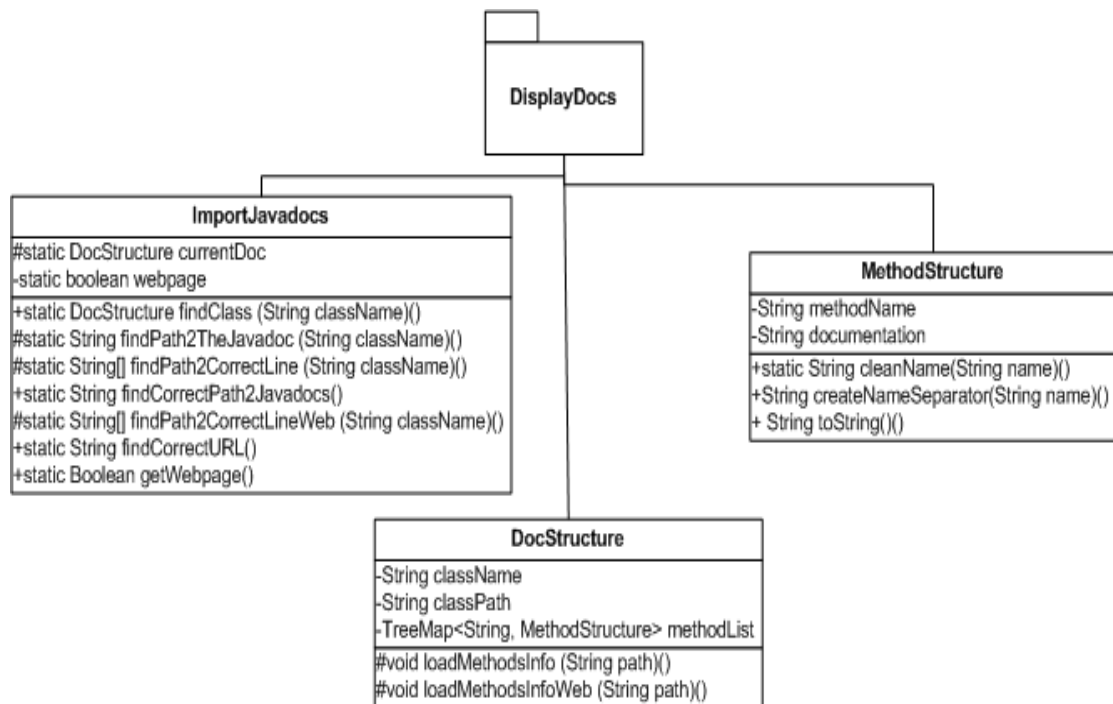
Το έργο στα όρια που είχα θέσει ολοκληρώθηκε. Συγκεκριμένα υλοποιήθηκαν οι εντολές “:importdocs”, “:docs” όπως είχαν αναφερθεί αλλά και η εντολή “:currdocs” που χρησιμεύει στο να απαντά στο χρήστη πια τεκμηρίωση έχει στην κονσόλα του εκείνη την στιγμή.

Η υλοποίηση περιλαμβάνει κώδικα σε 2 γλώσσες Java και Groovy. Η Groovy για τις εντολές της κονσόλας. Πιο κάτω παρατίθεται ένα διάγραμμα UML ώστε να

διευκολύνεται η κατανόηση της δομής. Το μόνο που δεν εμφανίζεται στο παρακάτω διάγραμμα είναι η μεταβλητή περιβάλλοντος που χρησιμοποιώ για να αποθηκεύσω την τρέχουσα τεκμηρίωση.



Η Java χρησιμοποιείται για την υλοποίηση του πακέτου ***org.apache.tinkerpop.gremlin.console.groovy.DisplayDocs*** που είναι υπεύθυνο για την λειτουργικότητα των εντολών. Πιο κάτω παρατίθεται ένα διάγραμμα UML για την παρουσίαση των κυριότερων μεθόδων κάθε κλάσης του πακέτου.



Για την υλοποίηση της λειτουργικότητας έχουν ακολουθηθεί δύο στρατηγικές. Αρχικά πρέπει να αναφέρουμε ότι η gremlin-console διανέμεται σε μορφή .zip που περιλαμβάνει τα javadocs. Ο σχεδιασμός είναι να επιχειρείται να αναγνωστεί η τεκμηρίωση από αυτό το φάκελο. Το επικίνδυνο με αυτήν την στρατηγική είναι ότι υπάρχει περίπτωση ο χρήστης να έχει μετακινήσει το αρχείο εκκίνησης με συνέπεια να μην είναι δυνατή η εύρεση του σωστού path μέχρι τα αρχεία. Εδώ δημιουργείται η ανάγκη για μια εφεδρική στρατηγική. Αυτή είναι ότι αν δεν βρει το φάκελο εκτός από το να ειδοποιήσει τον χρήστη, προσπαθεί να αναγνώσει την javadoc του έργου από το site με την τεκμηρίωση (<http://tinkerpop.incubator.apache.org/>). Έτσι η εντολή δουλεύει πάντα αν υπάρχει σύνδεση στο διαδίκτυο. Ποιο κάτω μπορείτε να δείτε ένα παράδειγμα χρήσης της εντολής για την μέθοδο addVertex της κλάσης Graph (<http://tinkerpop.incubator.apache.org/javadocs/3.0.0.M9-incubating/core/org/apache/tinkerpop/gremlin/structure/Graph.html>).

```

gremlin> :docs "addVertex"
=====
addVertex
=====
Vertex addVertex(Object... keyValues)

Add a Vertex to the graph given an optional series of key/value pairs. These key/values must be provided in an even number where the odd numbered arguments are String property keys and the even numbered arguments are the related property values.

Parameters:

keyValues - The key/value pairs to turn into vertex properties

Returns:

The newly created vertex

=====
addVertex
=====
default Vertex addVertex(String label)

Add a Vertex to the graph with provided vertex label.

Parameters:

label - the label of the vertex

Returns:

The newly created labeled vertex

gremlin>

```

## 4. Έλεγχος

Στο τομέα του έλεγχου χρησιμοποιώ Junit. Η φιλοσοφία των περιπτώσεων που ελέγχω είναι δύο. Ότι η αρχική τιμή (Graph) φορτώνεται χωρίς προβλήματα, ο λόγος είναι ότι αν δεν φορτώνει αυτή η τιμή δεν θα μπορεί να ανοίξει η gremlin-console. Άρα με αυτό τον τρόπο αν κάτι αλλάξει για αυτήν την κλάση θα πρέπει κάποιος να αλλάξει και την αρχική τιμή στην console για να μπορέσει να κάνει build. Η δεύτερη είναι ότι σε περίπτωση που δοθούν λανθασμένες είσοδοι οι εντολές θα δώσουν λογικό αποτέλεσμα. Για παράδειγμα δεν θα αλλάξουν την ήδη φορτωμένη κλάση εκτός αν έχει εξασφαλιστεί η φόρτωση της άλλης!

## 5. Συνεργασία με την ομάδα ανάπτυξης

Η συνεργασία με την ομάδα ανάπτυξης ίσως ήταν ένας από τους πιο σημαντικούς λόγους που επέλεξα το έργο. Η αλήθεια είναι ότι όταν έστειλα το πρώτο email δεν ήξερα αν θα λάβω απάντηση γενικά και όμως η απάντηση ήρθε σε 5 λεπτά και ήταν απίστευτα ενθαρρυντική! Κάτι που με ενθάρρυνε να ασχοληθώ και να δουλεύω σκληρά διότι δεν ήξερα τίποτα για Graph computing πριν δω το tinkerpop. Η πρώτη κίνηση που μου είπαν είναι να εγγραφώ στην development list ώστε να μπορώ να επικοινωνώ με την ομάδα ανάπτυξης άμεσα. Από τότε που το έκανα έχω ανταλλάξει πολλά μηνύματα για συμβουλές και για να σχολιάσω την δουλειά άλλων. Γενικά για δικές μου συμβουλές-απορίες έχουμε ανταλλάξει πάνω από 40 μηνύματα. Το τελευταίο μάλιστα ήρθε σήμερα το πρωί ως απάντηση στο μήνυμα που είχα στείλει εχθές για το pull request που ο Stephen μου αναφέρει ότι παρότι δεν έχει προλάβει να δει ενδελεχώς το κώδικα του άρεσε το εμφανισιακό αποτέλεσμα το αποτέλεσμα.

## 6. Σχετικοί Σύνδεσμοι

**Pull request:** <https://github.com/apache/incubator-tinkerpop/pull/71>

**Apache Tinkerpop3:** <https://github.com/apache/incubator-tinkerpop>

**Folk:** <https://github.com/xristosoik/incubator-tinkerpop>