ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ ΑΝΑΛΥΤΗ ΡΟΗΣ ΔΕΔΟΜΕΝΩΝ

Παρακάτω θα αναφερθούμε συνοπτικά στο τρόπο λειτουργίας και τη βασική αρχιτεκτονική του αναλυτή ροής δεδομένων. Σκοπός του εργαλείου είναι να δέχεται ως είσοδο τον γράφο χρήσης ορισμού ενός προγράμματος ή μιας μονάδας κώδικα, ο οποίος είναι ο γράφος ελέγχου ροής του προγράμματος όπου στους κόμβους και τις ακμές του έχουμε συσχετίσει το σύνολο των μεταβλητών που ορίζονται καθώς και το σύνολο των μεταβλητών που χρησιμοποιούνται στο κόμβο ή την ακμή, και να μας παράγει τα μονοπάτια που χρειάζεται να καλύψουμε για να ικανοποιήσουμε το κριτήριο που επιλέγουμε.

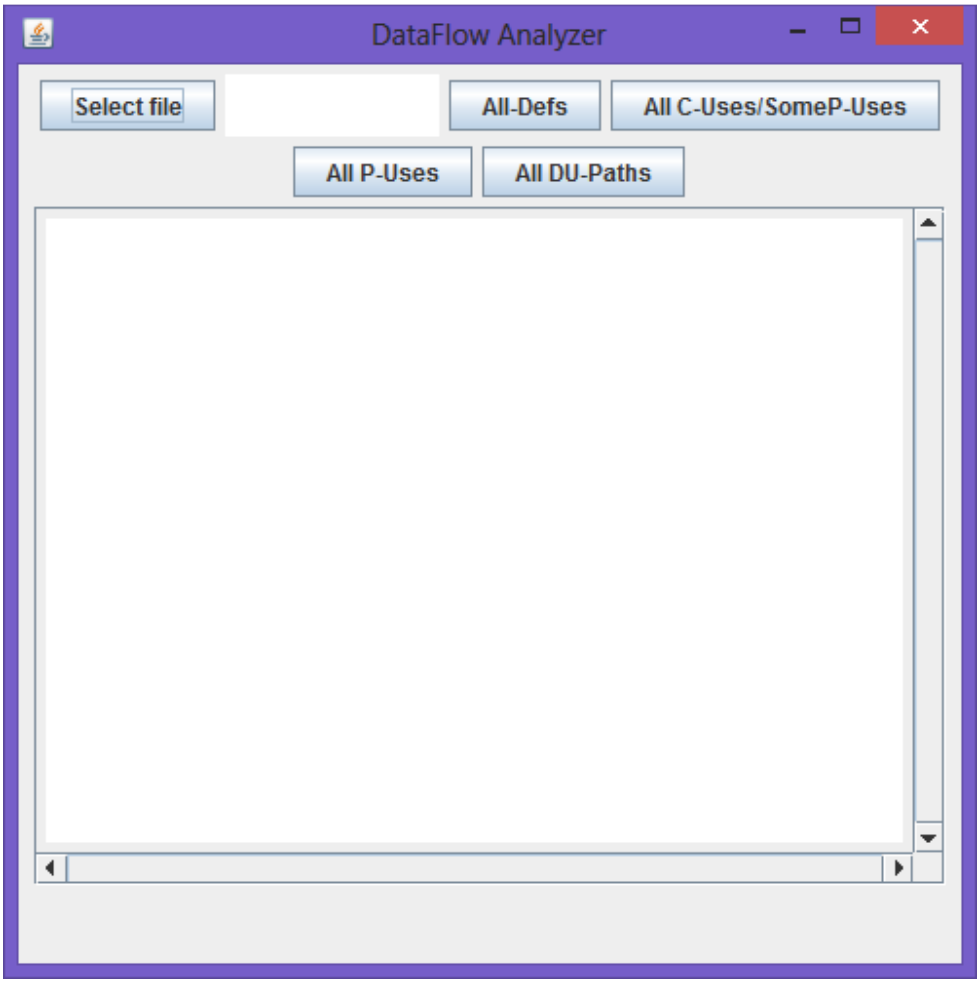
Τα κριτήρια που μπορούμε να επιλέξουμε είναι τα all-defs,all-c-uses/some-p-uses,all-p-useskαι all-du-paths.Η είσοδος γίνεται σ'ενα γραφικό περιβάλλον αναπτυγμένο σε περιβάλλον Swing. Το αρχείο που φορτώνουμε είναι σε μορφή XML με μια συκεκριμένη δομή που θα παρουσιάσουμε παρακάτω.

Γραφικό περιβάλλον

Το βασικό παράθυρο της εφαρμογής μας είναι αναπτυγμένο σε JavaSwingκαι περιλαμβάνει ένα κουμπί που φορτώνουμε το αρχείο XML, τέσσερα κουμπιά για τα κριτήρια μας και μια περιοχή κειμένου textarea που μας δείχνει τ'αποτελέσματα. Δείχνουμε σε βήματα πως βρίσκουμε τα μονοπάτια έχοντας το γράφο ορισμού χρήσης

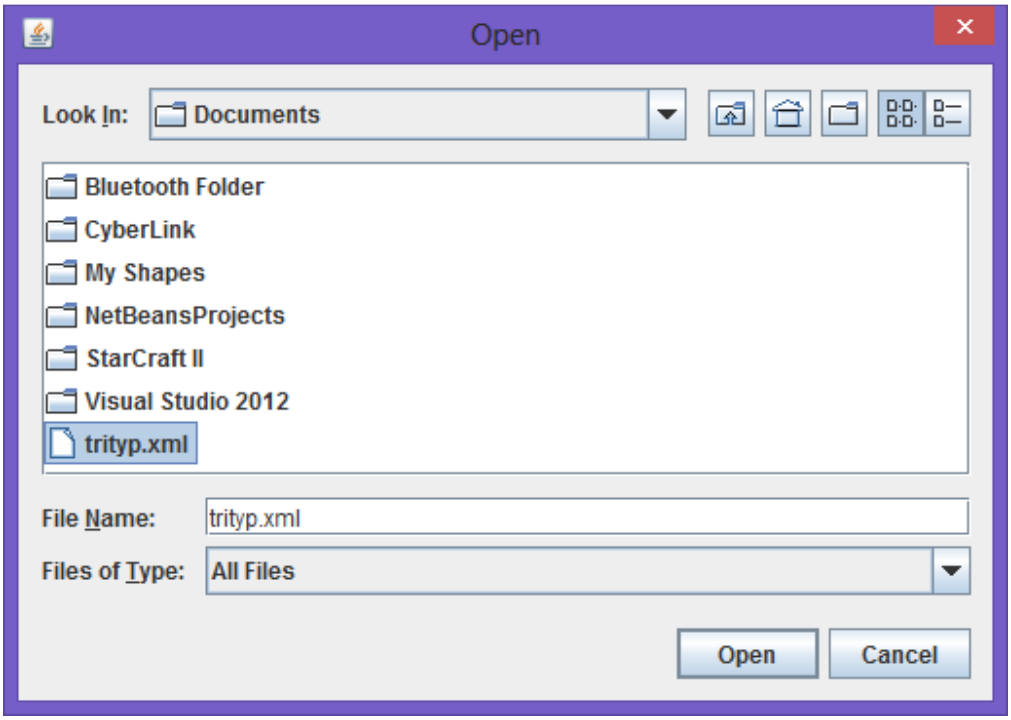
Βήμα 1.

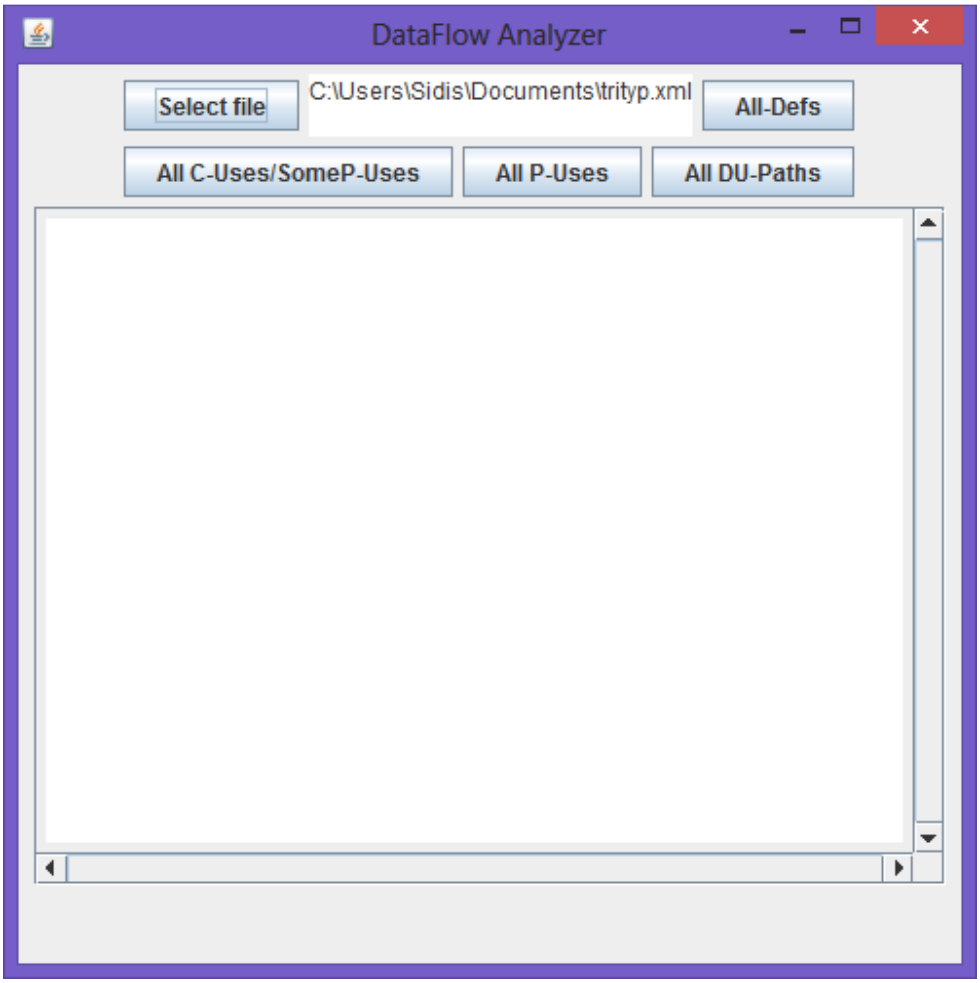
Με διπλό κλικ στο αρχείο dataFlowAnalyzer.jaremφανίζεται το παράθυρο της εφαρμογής μας



Βήμα 2

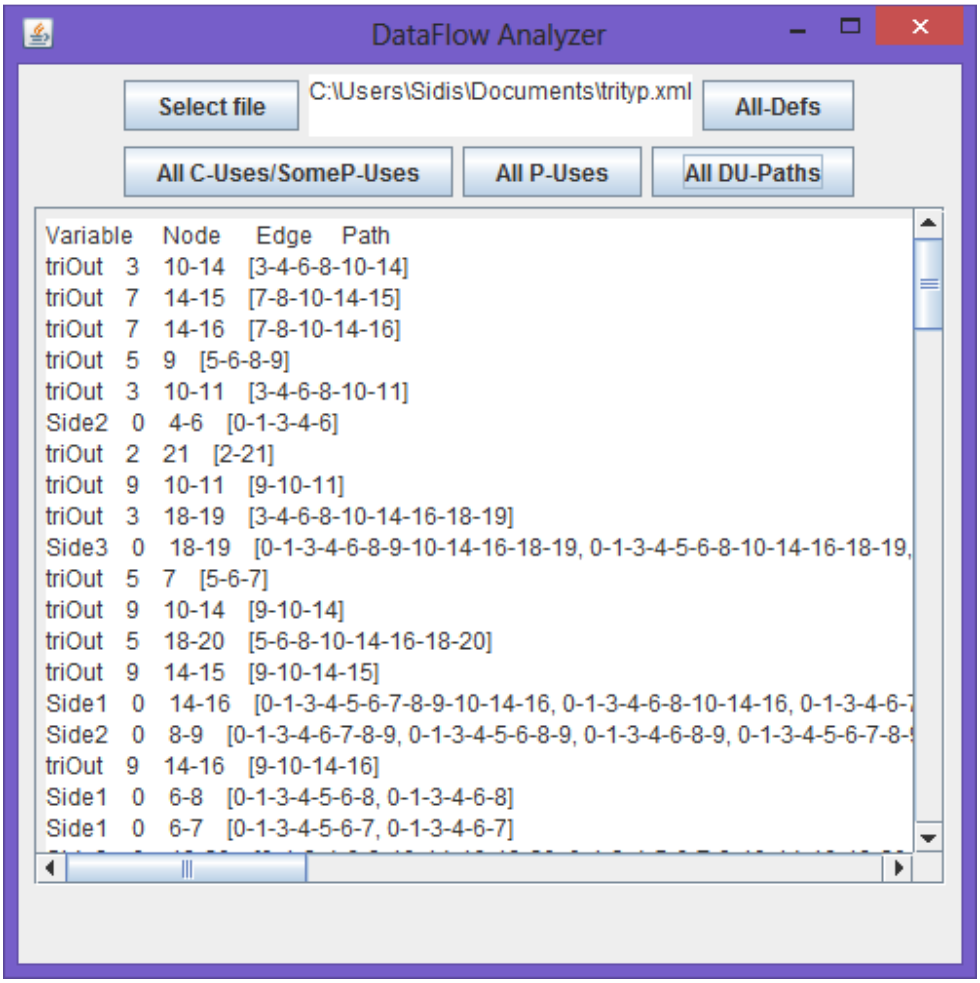
Επιλέγουμε το αρχείο XML επιλέγωντας Selectfile

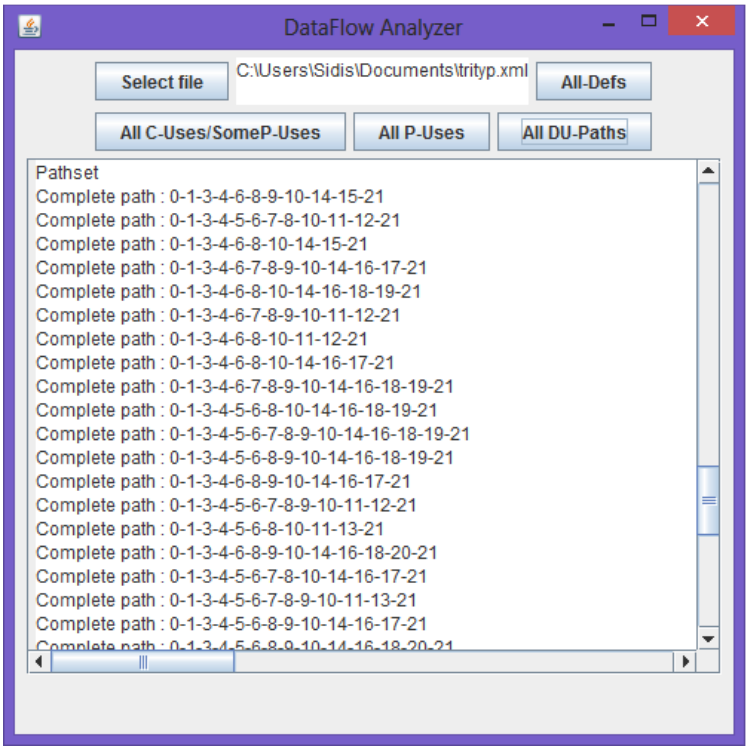

Open
Look In:
Documents
Bluetooth Folder
CyberLink
My Shapes
NetBeansProjects
StarCraft II
Visual Studio 2012
trityp.xml
File Name:
trityp.xml
Files of Type:
All Files
Open
Cancel



Βήμα 3

Επιλέγουμε το κριτήριο που επιθυμούμε


DataFlow Analyzer
Select file
C:\Users\Sidis\Documents\trityp.xml
All-Defs
All C-Uses/SomeP-Uses
All P-Uses
All DU-Paths
Variable Node Edge Path
triOut 3 10-14 [3-4-6-8-10-14]
triOut 7 14-15 [7-8-10-14-15]
triOut 7 14-16 [7-8-10-14-16]
triOut 5 9 [5-6-8-9]
triOut 3 10-11 [3-4-6-8-10-11]
Side2 0 4-6 [0-1-3-4-6]
triOut 2 21 [2-21]
triOut 9 10-11 [9-10-11]
triOut 3 18-19 [3-4-6-8-10-14-16-18-19]
Side3 0 18-19 [0-1-3-4-6-8-9-10-14-16-18-19, 0-1-3-4-5-6-8-10-14-16-18-19,
triOut 5 7 [5-6-7]
triOut 9 10-14 [9-10-14]
triOut 5 18-20 [5-6-8-10-14-16-18-20]
triOut 9 14-15 [9-10-14-15]
Side1 0 14-16 [0-1-3-4-5-6-7-8-9-10-14-16, 0-1-3-4-6-8-10-14-16, 0-1-3-4-6-
Side2 0 8-9 [0-1-3-4-6-7-8-9, 0-1-3-4-5-6-8-9, 0-1-3-4-6-8-9, 0-1-3-4-5-6-7-8-
triOut 9 14-16 [9-10-14-16]
Side1 0 6-8 [0-1-3-4-5-6-8, 0-1-3-4-6-8]
Side1 0 6-7 [0-1-3-4-5-6-7, 0-1-3-4-6-7]



Βλέπουμε ότι η εφαρμογή μας έχει δώσει για κάθε μεταβλητή το κόμβο που ορίζεται, το κόμβο ή την ακμή που χρησιμοποιείται και το μονοπάτι που περνάει από αυτά τα σημεία. Τέλος μας δίνει και ένα σύνολο από μονοπάτια που καλύπτουν το κριτήριο που επιλέξαμε. Τα μονοπάτια αυτά είναι μοναδικά.

Γράφος ορισμού χρήσης

Βασικό σημείο για την εφαρμογή μας είναι η είσοδος που της δίνουμε. Αυτό που δίνουμε ως είσοδο είναι ο γράφος ορισμού χρήσης του προγράμματος προς εξέταση με μορφή XML. Η μορφή αυτή πρέπει να είναι συγκεκριμένη.

Το αρχικό στοιχείο του αρχείου XML είναι το root element Graph. Ύστερα υπάρχουν ως στοιχεία παιδιά του Graph το στοιχείο Node και το Edge. Το στοιχείο Node έχει παιδιά τα στοιχεία NodeName, που είναι το όνομα του κόμβου, στοιχεία Def που είναι οι μεταβλητές που ορίζονται στο κόμβο και Cuse, που είναι μεταβλητές που γίνεται χρήση υπολογισμού στο κόμβο. Το στοιχείο Edge έχει στοιχεία παιδιά τα EdgeName, που είναι το όνομα της ακμής και Puse που είναι οι μεταβλητές που γίνεται χρήση βεβαίωσης στο κόμβο. Ένα παράδειγμα ενός τέτοιου αρχείου XML δίνεται παρακάτω:

```
<Graph>
<Node>
<NodeName>0</NodeName>
        <Def>triOut</Def>
        <Def>Side2</Def>
        <Cuse></Cuse>
</Node>
<Node>
<NodeName>1</NodeName>
        <Def>Side1</Def>
        <Cuse>triOut</Cuse>
</Node>
<Node>
<NodeName>2</NodeName>
        <Def>Side1</Def>
        <Cuse>triOut</Cuse>
</Node>
<Edge>
<EdgeName>1-2</EdgeName>
        <Puse>triOut</Puse>
</Edge>
</Graph>
```

Η δομή αυτή πρέπει ν'ακολουθηθεί για όλο το γράφο. Ένα βασικό στοιχείο είναι ότι τα ονόματα των γράφων πρέπει να είναι αριθμοί και να ξεκινάνε από το 0, γιατί η σχέση των κόμβων μεταξύ τους έχει αναπαρασταθεί μ'ενα διδιάστατο πίνακα σε Java.

Κώδικας της εφαρμογής

Οι κλάσεις που έχουμε χρησιμοποιήσει για την ανάπτυξη της εφαρμογής είναι οι Graph.java, Node.java, Edge.java, DataFlowAnalyzer.java, ReadXML.javaκαι MainWindow.java

Ο γράφος ορισμού χρήσης έχει υλοποιηθεί σε Java δημιουργώντας μια κλάση, την Graph.java. Οι βασικές μεταβλητές αυτής της κλάσης είναι η nodeList, η edgeList και η adMatrix. Η nodeList είναι μια μεταβλητή τύπου Map<String, Node> που περιέχει όλους τους κόμβους του γράφου ορισμού χρήσης και αντιστοιχεί τον κάθε κόμβο με το όνομα του. Η edgeList είναι μια μεταβλητή τύπου Map<String, Edge> που περιέχει όλες τις ακμές του γράφου ορισμού χρήσης και αντιστοιχεί την κάθε ακμή με το όνομα της. Τέλος η adMatrix είναι μια μεταβλητή πίνακας τύπου Integer[][] που περιέχει τις συνδέσεις των κόμβων μεταξύ τους.

Οι βασικέςμέθοδοιτης Graph.java είναι η addNode(Node node), η addEdge(Edge edge), η createAdjMatrix() και η adjacentNodes(String last). Η addNode(Nodenode) είναι μια publicvoid μέθοδος η οποία δέχεται ως όρισμα μια μεταβλητή τύπου Node και την προσθέτει στο γράφο. Η addEdge(Edgeedge) είναι επίσης μια publicvoid μέθοδος η οποία δέχεται ως όρισμα μια μεταβλητή τύπου Edge και προσθέτει στο γράφο μια ακμή. HcreateAdjMatrix()είναι μια publicvoid η οποία δημιουργεί τον πίνακα adMatrix. Τέλος η adjacentNodes(Stringlast) είναι μια μέθοδος publicLinkedList<String> η οποία δέχεται ως είσοδο μια μεταβλητή τύπου String και επιστρέφει πίσω μια συνδεδεμένη λίστα με τους γειτονικούς κόμβους ενός κόμβου του γράφου.

Μια απ' τις οντότητες που αποτελούν το γράφο είναι ο κόμβος. Αυτή η οντότητα έχει αναπαρασταθεί σε Java ως μια κλάση με το όνομα Node.java. Οι βασικές της μεταβλητές είναι η name, η defs και η cUses. Η name είναι μια μεταβλητή τύπου String που αντιπροσωπεύει το όνομα του κόμβου. Η defs είναι μια μεταβλητή τύπου List<String> που είναι μια λίστα με όλες τις μεταβλητές του προγράμματος που ορίζονται στο συγκεκριμένο κόμβο. Τέλος η cUses είναι μια μεταβλητή τύπου List<String> που είναι μια λίστα με όλες τις μεταβλητές του προγράμματος που γίνεται χρήση υπολογισμού στο συγκεκριμένο κόμβο.

Οι βασικές μέθοδοι είναι δύο publicvoid μέθοδοι που είναι setters για να θέτουν τιμές για τις μεταβλητές defs και cUses, οι setDefs και setCUses.

Η επόμενη οντότητα που απαρτίζει το γράφο είναι η ακμή. Αυτή έχει αναπαρασταθεί σε Java με μια κλάση με το όνομα Edge.java. Οι βασικές της μεταβλητές είναι η name και η pUses. Η name είναι μια μεταβλητή τύπου String που αναπαριστά το όνομα της ακμής. Σαν όνομα μιας ακμής χρησιμοποιούνται τα ονόματα δύο κόμβων χωρισμένα με το χαρακτήρα -, π.χ. «1-2». Η

pUseseείναι μια μεταβλητή τύπου List<String> που είναι μια λίστα με όλες τις μεταβλητές του προγράμματος που γίνεται χρήση βεβαίωσης στη συγκεκριμένη ακμή.

Η μοναδική μέθοδος της κλάσης είναι μια publicvoid μέθοδος που είναι setter για να θέτει τιμή για τη μεταβλητή pUses.

Ο αναλυτής ροής δεδομένων είναι η βασική κλάση της εφαρμογής μας. Έχει αναπαρασταθεί σε Java ως μια κλάση, την dataFlowAnalyzer.java. Η κλάση αυτή περιέχει τις βασικές μεθόδους με τις οποίες αναζητούμε και βρίσκουμε μονοπάτια που πληρούν τα κριτήρια που έχουμε ήδη εξετάσει. Ο κατασκευαστής της κλάσης δέχεται ως παράμετρο το γράφο χρήσης ορισμού, δημιουργεί τον πίνακα γειτνίασης αλλά και αρχικοποιεί τις μεταβλητές που αντιστοιχίζουν τους ορισμούς και τις χρήσεις των μεταβλητών του προς εξέταση προγράμματος με τους κόμβους του γράφου που γίνεται ο ορισμός και οι χρήσεις. Ο κώδικας του κατασκευαστή είναι ο ακόλουθος:

```
publicDataFlowAnalyzer(Graphgraph){
        this.g=graph;
        g.createAdjMatrix();
        this.createCUsesMap();
        this.createDefsMap();
        this.createPUsesMap();
    }
```

Η κλάση επίσης χρησιμοποιεί κάποιες βοηθητικές μεθόδους που είναι publicstatic οι οποίες επιτελούν διάφορες λειτουργικότητες που βοηθούν στην εύρεση των μονοπατιών. Αυτές βρίσκονται στο τέλος της κλάσης. Οι τρεις μεταβλητές που αντιστοιχίζουν τους ορισμούς και τις χρήσεις των μεταβλητών του προς εξέταση προγράμματος με τους κόμβους του γράφου που γίνεται ο ορισμός και οι χρήσεις είναι οι cUsesMap, defsMap και pUsesMap και είναι τύπου Map<String,String>. Παρακάτω θα αναφερθούμε στις βασικές μεθόδους με τις οποίες βρίσκουμε τα μονοπάτια ανάλογα με το κριτήριο που επιθυμούμε.

All-defs

Η μέθοδος για την εύρεση των μονοπατιών με βάση το κριτήριο all-defs είναι μια public μέθοδος με το όνομα findalldefs().Στη μέθοδο αυτή βρίσκουμε όλα τα μονοπάτια για κάθε κόμβο i και κάθε μεταβλητή x που ανήκει στο def(i) προς κάποιο κόμβο του dcu(x,i) ή κάποια ακμή του dpu(x,j,k) τα οποία είναι καθαρά από ορισμούς. Στη μέθοδο γίνεται χρήση βοηθητικών μεθόδων που βρίσκουν μονοπάτι από τον αρχικό κόμβο πρός το κόμβο που ορίζεται η μεταβλητή, μονοπατιού από το κόμβο που ορίζεται η μεταβλητή πρός το κόμβο που

χρησιμοποιείται, όπου εξετάζεται αν το μονοπάτι είναι καθαρό από ορισμούς, αλλιώς βρίσκεται κάποιο άλλο, και τέλος απ΄το κόμβο που χρησιμοποιείται πρός τον τελικό κόμβο.

All-c-uses/some-p-uses

Η μέθοδος για την εύρεση των μονοπατιών με βάση το κριτήριο all-c-uses είναι μια public μέθοδος με το όνομα findcuses(). Στη μέθοδο αυτή βρίσκουμε όλα τα μονοπάτια για κάθε κόμβο i και κάθε μεταβλητή x που ανήκει στο def(i) προς κάθε κόμβο του dcu(x,i) τα οποία είναι καθαρά από ορισμούς.Αν το dcu(x,i) είναι άδειο τότε βρίσκουμε ένα μονοπάτι απ' το dpu(x,i).Στη μέθοδο γίνεται χρήση βοηθητικών μεθόδων που βρίσκουν μονοπάτι από τον αρχικό κόμβο πρός το κόμβο που ορίζεται η μεταβλητή, μονοπατιού από το κόμβο που ορίζεται η μεταβλητή πρός το κόμβο που χρησιμοποιείται, όπου εξετάζεται αν το μονοπάτι είναι καθαρό από ορισμούς, αλλιώς βρίσκεται κάποιο άλλο, και τέλος απ΄το κόμβο που χρησιμοποιείται πρός τον τελικό κόμβο.

All-p-uses

Η μέθοδος για την εύρεση των μονοπατιών με βάση το κριτήριο all-p-uses είναι μια public μέθοδος με το όνομα findAllPuses(). Στη μέθοδο αυτή βρίσκουμε όλα τα μονοπάτια για κάθε κόμβο i και κάθε μεταβλητή x που ανήκει στο def(i) προς κάθε ακμή του dpu(x,i) τα οποία είναι καθαρά από ορισμούς. Στη μέθοδο γίνεται χρήση βοηθητικών μεθόδων που βρίσκουν μονοπάτι από τον αρχικό κόμβο πρός το κόμβο που ορίζεται η μεταβλητή, μονοπατιού από το κόμβο που ορίζεται η μεταβλητή πρός το πρώτο κόμβο της ακμής που χρησιμοποιείται, όπου εξετάζεται αν το μονοπάτι είναι καθαρό από ορισμούς, αλλιώς βρίσκεται κάποιο άλλο, και τέλος απ΄το δεύτερο κόμβο της ακμής που χρησιμοποιείται πρός τον τελικό κόμβο.

All-DU-Paths

Η μέθοδος για την εύρεση των μονοπατιών με βάση το κριτήριο all-du-paths είναι μια public μέθοδος με το όνομα findAllDUpaths(). Στη μέθοδο αυτή βρίσκουμε κάθε μονοπάτι για κάθε κόμβο i και κάθε μεταβλητή x που ανήκει στο def(i) προς κάθε κόμβο του dcu(x,i) και κάθε ακμή του dpu(x,i) τα οποία είναι καθαρά από ορισμούς. Στη μέθοδο γίνεται χρήση των βοηθητικών μεθόδων που γίνονται και στις μεθόδους findcuses() και findAllPuses(). Η εύρεση όλων των μονοπατιών από το κόμβο που ορίζεται η μεταβλητή πρός τον κόμβο που χρησιμοποιείται ή το πρώτο κόμβο της ακμής που χρησιμοποιείται, γίνεται με τη χρήση του αλγορίθμου DFS. Τα μονοπάτια τα βρίσκουμε με τη μέθοδο getPathsFromANodeToAnother(), μέσα στην οποία υλοποιείται ο αλγόριθμος DFS. Με τη μέθοδο αυτή θέτουμε τιμές σε μια μεταβλητή Set<String>που είναι όλα τα δυνατά μονοπάτια από έναν κόμβο προς έναν άλλο και ύστερα το κάθε ένα απ' αυτά εξετάζουμε αν είναι def-clearμε την μέθοδο isDefClear() που μας επιστρέφει μια τιμή boolean. Ύστερα βρίσκουμε ένα μονοπάτι απ'το rootκόμβο πρός τον πρώτο

κόμβο, κάνουμε concatenateto Stringαπ’το Set<String>και αυτό το κάνουμε concatenateμε ένα μονοπάτι απ’το δεύτερο κόμβο προς τον τελικό.

ReadXML.java

Η κλάση η οποία διαβάζει το αρχείο και δημιουργεί ένα αντικείμενο τηε κλάσης Graph, το οποίο περνάει ως είσοδο σ’ ένα αντικείμενο της κλάσης DatFlowAnalyzer, είναι η ReadXML.java. Η βασική της μέθοδος είναι η createGraph, η οποία αρχικοποιεί αντικείμενο τηε κλάσης Graph, διαβάζει το αρχείο και θέτει τιμές για τους κόμβους και τις ακμές του γράφου, καθώς και θέτει τιμές για τις μεταβλητές που ορίζονται και χρησιμοποιούνται σε κάθε κόμβο και ακμή

MainWindow.java

Η κλάση που δημιουργεί το παράθυρο της εφαρμογής μας είναι η MainWindow.java. Αυτή χρησιμοποιεί το Swingγια να δημιουργήσει τη διεπαφή του χρήστη με την εφαρμογή.