

ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων



Πτυχιακή Εργασία

Οπτική αντιπροσώπευση δεδομένων
(Visual representation of data)

Χαρίλαος Λεμενιτάκης

Μάρτιος 2010

Επιβλέπων καθηγητής : Δρ Βιδάκης Νικόλαος
Επιτροπή Αξιολόγησης : Βιδάκης Ν, Παχουλάκης Γ, Μηλολιδάκης Γ

Περίληψη

Σκοπός της πτυχιακής αυτής ήταν η οπτική αντιπροσώπευση δεδομένων βιοπληροφορίας του προγράμματος αναπαράστασης βιολογικών δομών Jmol. Μέσω της οπτικής αντιπροσώπευσης δεδομένων του προγράμματος Jmol στόχος ήταν να επιτευχθεί η βελτίωση της διεπαφής χρήσης του προγράμματος.

Το Jmol είναι ένα πρόγραμμα ανοικτού κώδικα που ανήκει στην κατηγορία του ανοικτού λογισμικού και ο κώδικας του είναι γραμμένος στην αντικειμενοστραφή γλώσσα προγραμματισμού Java. Για να επιτευχθεί η βελτιστοποίηση της οπτικής αντιπροσώπευσης δεδομένων του προϋπάρχοντος προγράμματος Jmol δημιουργήθηκαν νέα περιβάλλοντα προς την κατεύθυνση αυτή, τα οποία ενσωματώθηκαν στην εφαρμογή του ήδη υπάρχοντος προγράμματος Jmol.

Τα νέα περιβάλλοντα που δημιουργήθηκαν γράφτηκαν σε γλώσσα προγραμματισμού Java ώστε να είναι απολύτως συμβατά με τον κώδικα του Jmol. Χρησιμοποιήθηκαν ιδιαίτερα στην ανάπτυξη τους τα πακέτα της Java Swing και Awt, για την ανάπτυξη του περιβάλλοντος των παραθύρων της νέας εφαρμογής.

Λόγω του γνωστικού αντικείμενου που πραγματεύεται η εφαρμογή που δημιουργήθηκε, δηλαδή την παροχή πληροφορίας που αφορά βιολογικές δομές και μόρια, πριν αλλά και κατά την διάρκεια της πτυχιακής μελετήθηκαν διάφορες πληροφορίες σχετικά με της βιολογικές δομές και τους τρόπους που αναπαρίστανται. Μελετήθηκαν ακόμα προγράμματα που πραγματεύονται το ίδιο γνωστικό αντικείμενο με το πρόγραμμα Jmol, ώστε τα περιβάλλοντα και οι εφαρμογές που θα δημιουργηθούν να καλύπτουν της επιμέρους ανάγκες ενός προγράμματος αναπαράστασης βιολογικών δομών.

Τα περιβάλλοντα που δημιουργήθηκαν στο πλαίσιο της παρούσας πτυχιακής έχουν σαν αντικειμενικό στόχο την βελτίωση της διαδραστικότητας μεταξύ του προγράμματος και του χρήστη, για αυτό το λόγω το αποτέλεσμα είναι, να είναι εύκολα και αποτελεσματικά στην χρήση τους με απλουστευμένες λειτουργίες και να παρέχουν ευελιξία στις ενέργειες του χρήστη μέσα από τα παρεχόμενα μενού. Τα περιβάλλοντα αυτά παρέχουν στον χρήστη ένα σύνολο πληροφοριών που είναι επικεντρωμένες στο γνωστικό αντικείμενο των βιολογικών δομών και μορίων και λειτουργίες ώστε ο χρήστης να έχει την δυνατότητα αλληλεπίδρασης με αυτήν την πληροφορία που του παρέχεται, ο σκοπός που επιτυγχάνεται μέσω της αλληλεπίδρασης είναι ακριβώς η κατανόηση της πληροφορίας.

ABSTRACT

The aim of this exercise they was the optical representation of given biodata in the program of representation of biological structures Jmol. Via the optical representation of data of program Jmol, the aim was is achieved the improvement of contact of use of program.

Jmol it is a program of open code that belong in the category of open software and his code is written in language of planning Java. In order to is achieved the optimisation of optical representation of data of program Jmol were created new environments to this direction that were incorporated in his application goods of existing program Jmol. The new environments that were created were written in language of planning Java so that they are also absolutely compatible with the code of Jmol , were particularly used in their growth her parcels Java Swing and Awt, for the growth of environment of windows of new application.

For the reason of cognitive object that deals the application that created that is to say the benefit of information that concerns biological structures and molecules, before but also at the duration of final were studied different information with regard to biological structures and the ways that represented, Were still studied programs that they contain the same cognitive object with the program Jmol , in order that the environments and the applications that will be created they cover individual needs of program of representation of biological structures.

The environments that were created in the frame of this exercise they have as objective aim the improvement of interaction between the program and the user, for this reason the result is, that is easily also effectively in their use with simplified operations and provide flexibility in actions of user in from provided menu. This environments provide in the user a total of information that is focus in the cognitive object of biological structures and molecules and operations in order that the user has the possibility of interaction with this information that for him is provided, the aim that is achieved means of interaction is precisely the comprehension of information.

Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ	8
1.1. ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΠΤΥΧΙΑΚΗΣ	8
1.2. ΟΡΓΑΝΩΣΗ ΤΟΥ ΤΟΜΟΥ	9
2. ΧΡΗΣΙΜΟΙ ΟΡΙΣΜΟΙ ΚΑΙ ΠΛΗΡΟΦΟΡΙΕΣ	10
2.1. Η ΚΡΥΣΤΑΛΛΟΓΡΑΦΙΑ	10
2.2. ΑΝΑΠΑΡΑΣΤΑΣΗ ΜΟΡΙΩΝ	10
2.3. Η PROTEIN DATA BANK.....	10
2.4. ΤΑ PDB ΑΡΧΕΙΑ	11
2.5. Η ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ	11
2.6. Η ΒΙΟΜΟΡΙΑΚΗ ΠΡΟΣΟΜΟΙΩΣΗ	11
2.7. Η ΑΚΤΙΝΑ VAN DER WAALS ΕΝΟΣ ΑΤΟΜΟΥ	12
2.8. ΜΟΡΙΑΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ	12
3. Η ΕΦΑΡΜΟΓΗ	13
3.1. ΣΚΟΠΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	13
3.2. ΕΜΒΕΛΕΙΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ (APPLICATION SCORE)	13
3.3. ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	13
4. ΠΡΟΓΡΑΜΜΑΤΑ ΑΝΑΠΑΡΑΣΤΑΣΗΣ ΒΙΟΛΟΓΙΚΩΝ ΜΟΡΙΩΝ.....	18
4.1. ΤΙ ΕΞΥΠΗΡΕΤΟΥΝ	19
4.2. ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΟΥΝΤΑΙ	19
4.3. ΑΠΟ ΠΟΥ ΑΝΤΛΟΥΝ ΠΛΗΡΟΦΟΡΙΕΣ	20
4.4. ΤΟ ΠΡΟΓΡΑΜΜΑ ΑΠΕΙΚΟΝΙΣΗΣ ΒΙΟΛΟΓΙΚΩΝ ΔΟΜΩΝ Jmol	20
4.4.1. Η αρχιτεκτονική συστήματος του Jmol.....	21
4.5. ΆΛΛΑ ΠΡΟΓΡΑΜΜΑΤΑ ΑΠΕΙΚΟΝΙΣΗΣ ΒΙΟΛΟΓΙΚΩΝ ΔΟΜΩΝ.....	22
4.5.1. Το PyMol.....	22
5. ΤΕΧΝΟΛΟΓΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΕ ΓΙΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	25
5.1. ΕΙΣΑΓΩΓΗ.....	25
5.2. Η ΓΛΩΣΣΑ HTML.....	25
5.2.1. Οι εντολές της HTML που χρησιμοποιήθηκαν στην εφαρμογή.....	25
5.3. Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVA	26
5.3.1. Αντικειμενοστρεφής προγραμματισμός.....	26
5.3.2. Κλάσεις και αντικείμενα	26
5.3.3. Κληρονομικότητα (Inheritance).....	26
5.4. ΤΟ ΠΑΚΕΤΟ JAVA ABSTRACT WINDOWING TOOLKIT(AWT).....	27
5.4.1. Γεγονότα στα GUI (GUI Events)	28
5.5. ΤΟ ΠΑΚΕΤΟ JAVA SWING	29
5.5.1. Γραφική διασύνδεση χρήστη με χρήση Swing.....	30
5.5.2. Σύνθεση γραφικής διασύνδεσης με χρήση Swing.....	30
5.5.3. Top-Level Swing Containers και Swing Components	30
5.5.4. Διαχειριστές Γεγονότων - Event Handlers	31
5.6. ΤΑ ΚΥΡΙΟΤΕΡΑ ΣΥΣΤΑΤΙΚΑ ΤΟΥ ΠΑΚΕΤΟΥ SWING ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ ΣΤΗΝ ΥΛΟΠΟΙΗΣΗ.....	32
5.6.1. Το συστατικό Jtree (δέντρο) του Swing	32
5.6.2. Το συστατικό JEditorPane του Swing.....	32
5.6.3. Ο ενδιάμεσος υποδοχέας Jpanel (Πάνελ) του Swing	32

5.6.4.	Ο ενδιάμεσος υποδοχέας <i>JScrollPane</i> (Πάνελ κύλισης) του <i>Swing</i>	33
5.6.5.	Ο κύριος υποδοχέας <i>JFrame</i> (Πλαίσιο) του <i>Swing</i>	33
6.	ΑΝΑΛΥΣΗ ΛΟΓΙΣΜΙΚΟΥ	33
6.1.	ΕΙΣΑΓΩΓΗ	33
6.2.	ΜΕΘΟΔΟΛΟΓΙΑ ΑΝΑΠΤΥΞΗΣ	34
6.2.1.	Καταγραφή Απαιτήσεων	34
6.2.2.	<i>Use Cases</i>	35
6.2.3.	Σενάρια χρήσης (<i>Scenarios of Use</i>).	36
6.3.	ΓΕΝΙΚΟ ΣΕΝΑΡΙΟ ΧΡΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	37
6.3.1.	Καταγραφή απαιτήσεων της εφαρμογής	37
6.3.2.	Περιπτώσεις χρήσης και σενάρια της εφαρμογής	40
6.3.3.	Λίστα σεναρίων περίπτωσης χρήσης: Απαιτήσεις χρήσης της εφαρμογής	42
7.	ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	63
7.1.	USER INTERFACE.	63
7.1.1.	Η εφαρμογή <i>SequenceView</i>	64
7.1.2.	Η λειτουργία εισαγωγής σχολίου (<i>comment mode</i>)	68
7.1.3.	Η εφαρμογή <i>TreeView</i>	69
8.	ΣΥΜΠΕΡΑΣΜΑΤΑ	72
9.	ΒΙΒΛΙΟΓΡΑΦΙΑ	73
10.	ΠΑΡΑΡΤΗΜΑΤΑ	74
10.1.	ΠΑΡΑΡΤΗΜΑ Α: ΟΙ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΒΙΟΠΛΗΡΟΦΟΡΙΑΣ	74
10.2.	ΠΑΡΑΡΤΗΜΑ Β: ΤΑ ΑΡΧΕΙΑ <i>PDB</i>	75
10.2.1.	Ανάλυση πεδίων εγγραφών	77
10.3.	ΠΑΡΑΡΤΗΜΑ Γ: ΟΙ ΒΙΟΛΟΓΙΚΕΣ ΔΟΜΕΣ ΠΟΥ ΣΥΝΘΕΤΟΥΝ ΈΝΑ ΜΟΡΙΟ	80
10.3.1.	Οι βιολογικές δομές	80
10.4.	ΠΑΡΑΡΤΗΜΑ Δ: ΑΝΑΦΟΡΑ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	86
10.4.1.	Το περιβάλλον της εφαρμογής <i>SequenceView</i>	86
10.4.2.	Η λειτουργία εισαγωγής σχολίου (<i>addComment</i>)	88
10.4.3.	Το περιβάλλον της εφαρμογής <i>TreeView</i>	93

Πίνακες

Table 1: GUI-components AWT	28
Table 2: Γεγονότα και οι σχετικοί ακροατές γεγονότος τους.....	31
Table 3: Λίστα σεναρίων ανά περίπτωση χρήσης.....	41
Table 4: Δράσεις σεναρίων περίπτωσης χρήσης «Απαιτήσεις χρήσης της εφαρμογής» .	42
Table 5: Δράσεις σεναρίων περίπτωσης χρήσης «Απαιτήσεις λειτουργίας της εφαρμογής».....	45
Table 6: Δράσεις σεναρίων περιπτώσεις χρήσης «Απαιτήσεις λειτουργιών της εφαρμογής».....	52
Table 7: Επεξήγηση εγγραφών.....	76
Table 8: Επεξήγηση εγγραφών ισότιμου τμήματος.....	77
Table 9: Επεξήγηση εγγραφών δευτεροβάθμιας δομής.....	79
Table 10: Πίνακας αμινοξέων	84

Εικόνες

Figure 1: Λειτουργία εφαρμογής SequenceView.....	15
Figure 2: Λειτουργία εφαρμογής TreeView.....	16
Figure 3: Λειτουργία εφαρμογής Comment mode.....	17
Figure 4: Περιβάλλον εργασίας του Jmol.....	21
Figure 5: Αρχιτεκτονική συστήματος,τα βασικά συστατικά.....	22
Figure 6: Το μενού του Pymol.....	23
Figure 7: Το μενού του Pymol.....	24
Figure 8: Ιεραρχία των GUI-components κλάσεων.....	28
Figure 9: Ακροατές γεγονότων.....	29
Figure 10: Ιεραρχία του top-level Swing containers JFrame.....	30
Figure 11: Ιεραρχία της κλάσης JComponent.....	31
Figure 12: Επισημάνση αμινοξέων.....	65
Figure 13: Τα χαρακτηριστικά χρώματα των αμινοξέων όπως παρουσιάζονται στο περιβάλλον Amino acid color.....	66
Figure 14: Μεγέθυνση επιλεγμένων αμινοξέων.....	67
Figure 15: Εισαγωγή γραφικών σχολίου.....	68
Figure 16: Περιβάλλον εισαγωγής σχολίου.....	69
Figure 17: Η εφαρμογή TreeView.....	70
Figure 18: Επισημάνση της επιλεγμένης βιολογικής δομής στην σκηνή του Jmol.....	71
Figure 19: Σχήμα αναπαράστασης της δομής της πρωτεΐνης.....	82
Figure 20: Το αμινοξύ Αργινίνη(Arg) σε αναπαράσταση στο Jmol.....	83

1. Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζουμε μια γενική περιγραφή του αντικείμενου της πτυχιακής αυτής και την οργάνωση των κεφαλαίων του υπόλοιπου τόμου .

1.1. Αντικείμενο της πτυχιακής

Στην συγκεκριμένη πτυχιακή ασχοληθήκαμε με την βελτίωση της οπτικής αντιπροσώπευσης δεδομένων βιοπληροφορίας που αφορούν βιολογικές δομές στο πρόγραμμα απεικόνισης βιολογικών δομών και μορίων Jmol .Αντικείμενο της πτυχιακής είναι να σχεδιάσουμε και να δημιουργήσουμε νέα περιβάλλοντα και εφαρμογές τα οποία να ενσωματωθούν στο ήδη υπάρχον περιβάλλον του προγράμματος Jmol ,ως αποτέλεσμα την βελτίωση της οπτικής αντιπροσώπευσης δεδομένων βιοπληροφορίας μέσω της διεπαφής του προγράμματος .

Πρίν αλλά και κατά την διάρκεια της πτυχιακής μελετήθηκαν διάφορα προγράμματα αναπαράστασης βιολογικών δομών και τα επιμέρους περιβάλλοντα που παρείχαν .Η τελική επιλογή ήταν το πρόγραμμα Jmol ως το πρόγραμμα που θα βελτιώναμε την διαδραστικότητα του.Οι λόγοι που επιλέχθηκε το Jmol ήταν κυρίως ότι διατίθονταν ως πρόγραμμα ανοικτού κώδικα από τους δημιουργούς του,ότι ο κώδικας που είχε δημιουργηθεί το πρόγραμμα ήταν σε γλώσσα Java οπότε και στην εφαρμογή που θα δημιουργούσαμε θα μπορούσαμε να χρησιμοποιήσουμε την γλώσσα Java ως γλώσσα συγγραφής ,ότι το Jmol ύστερα από μελέτη που κάναμε στο διαδίκτυο διαπιστώθηκε ότι είναι ένα ευρέως διαδεδομένο αποδεκτό και έγκυρο πρόγραμμα στον τομέα που πραγματεύεται ,την απεικόνιση βιολογικών δομών και μορίων .Ένας ακόμη λόγος της επιλογής του συγκεκριμένου προγράμματος ήταν ότι υπήρχαν ήδη κοινότητες χρηστών στο διαδίκτυο όπως η Wiki που κατέβαλαν προσπάθεια για την βελτίωση της εφαρμογής του προγράμματος .

Τα συστατικά που δημιουργήσαμε παρέχουν στο χρήστη δεδομένα βιοπληροφορίας που αφορούν το μόριο που απεικονίζεται στην σκηνή του Jmol,ο χρήστης έχει την δυνατότητα να αλληλεπιδρά με αυτά τα δεδομένα βιοπληροφορίας και μέσω των συστατικών που τα παρέχουν να αλληλεπιδρά με το γραφικό περιβάλλον αναπαράστασης των μορίων ,την σκηνή του Jmol, βελτιώνοντας με αυτό τον τρόπο την διεπαφή του προγράμματος.Το Jmol είναι πρόγραμμα ανοικτού κώδικα γραμμένο σε γλώσσα προγραμματισμού Java , πρώτα έγινε πλήρη κατανόηση των περιοχών εκείνων του κώδικα του προγράμματος που μας ενδιέφεραν για να εισάγουμε τα στοιχεία που θέλαμε και στην συνέχεια έγινε η εισαγωγή των καινούριων συστατικών στο πρόγραμμα , στόχος ήταν να αλληλεπιδρούν αρμονικά με αυτό . Έτσι ο χρήστης μέσω των καινούργιων περιβαλλόντων που δημιουργήθηκαν εκτός από της δυνατότητες που του παρείχε πριν το Jmol , τώρα μπορεί να κατανοήσει καλύτερα την ατομική δομή των μορίων που αναπαρίστανται γραφικά και την πρωτεϊνική τους δομή .Με τα περιβάλλοντα που δημιουργήθηκαν επιτυγχάνεται η επισήμανση των αμινοξέων που συγκροτούν την πρωτεΐνη στο γραφικό περιβάλλον αναπαράστασης , το χαρακτηριστικό χρώμα των αμινοξέων όπως είναι επιστημονικά αποδεκτό και το σημείο που βρίσκονται τα συγκεκριμένα άτομα που σχηματίζουν τα αμινοξέα μέσα στην δομή του μορίου.Ο χρήστης έχει την δυνατότητα ακόμα να δει ορισμένα δομικά συστατικά του βιολογικού μορίου που αναπαρίστανται ιεραρχικά δομημένα σε μορφή δέντρου,όπως τα ονόματα των

ατόμων που αποτελούν το μόριο. Ένα άλλο χρήσιμο χαρακτηριστικό που προστέθηκε στο Jmol με της εφαρμογές που δημιουργήθηκαν είναι η εισαγωγή σχολίου η διαφόρων πληροφοριών σε μορφή κειμένου σε καθένα άτομο ξεχωριστά της μοριακής δομής που απεικονίζεται, όπως και η αποθήκευση αλλά και επανάκτηση αυτής της πληροφορίας(σχολίου) που συνοδεύει τα άτομα.

1.2. Οργάνωση του τόμου

Η συνέχεια της πτυχιακής έχει οργανωθεί στα ακόλουθα κεφάλαια:

- Στο Κεφάλαιο 2, παρουσιάζονται χρήσιμες έννοιες και ορισμοί που αφορούν τα προγράμματα μοριακής απεικόνισης και γενικότερα τον γνωστικό τομέα που ανήκουν.
- Στο Κεφάλαιο 3, παρουσιάζεται η εφαρμογή που δημιουργήθηκε και αναλύεται η περαιτέρω λειτουργία της.
- Στο Κεφάλαιο 4, παρουσιάζονται τα προγράμματα αναπαράστασης βιολογικών δομών και οι λειτουργίες τους και αναφέρεται ιδιαίτερα το πρόγραμμα αναπαράστασης Jmol ,το οποίο χρησιμοποιήθηκε στην παρούσα εργασία.
- Στο Κεφάλαιο 5, παρουσιάζεται η τεχνολογία που χρησιμοποιήθηκε για την ανάπτυξη του συστήματος.
- Στο Κεφάλαιο 6, παρουσιάζονται γενικές αλλά και ειδικές πληροφορίες για την ανάλυση και σχεδίαση του συστήματος.
- Στο Κεφάλαιο 7, αναφέρεται η υλοποίηση της εφαρμογής ,τα επιμέρους περιβάλλοντα και οι λειτουργίες που υλοποιήθηκαν.
- Στο Κεφάλαιο 8, παρουσιάζονται τα συμπεράσματα της έρευνας και της υλοποίησης.
- Στο Κεφάλαιο 9, παρουσιάζεται όλη η σχετική βιβλιογραφία στην οποία βασιστήκαμε για τη μελέτη και εγγραφή της πτυχιακής.
- Στο Κεφάλαιο 10, παρουσιάζονται τα παραρτήματα στα οποία αναφερόμαστε σε διάφορα σημεία της πτυχιακής.

2. Χρήσιμοι ορισμοί και πληροφορίες

Στα επόμενα κεφάλαια θα αναλύσουμε καλύτερα το πρόγραμμα που χρησιμοποιήθηκε αλλά και της εφαρμογές που δημιουργήσαμε, για να γίνουν καλύτερα κατανοητά σε αυτό το κεφάλαιο θα εξηγήσουμε ορισμένους σχετικούς ορισμούς και θα αναφέρουμε σχετικές πληροφορίες.

2.1. Η Κρυσταλλογραφία

Κρυσταλλογραφία ονομάζεται γενικά η μελέτη του κρυσταλλικού πλέγματος, δηλαδή της γεωμετρικής δόμησης των κρυστάλλων. Με τη βοήθεια της κρυσταλλογραφίας και των μεθόδων που ακολουθεί αυτή, οι επιστήμονες ανακαλύπτουν τη γεωμετρική διάταξη (κρυσταλλικό σύστημα) των σωματιδίων που συγκροτούν τους κρυστάλλους. Δέσμη ακτίνων X περνώντας μέσα από τα μικροσκοπικά σωματίδια του κρυστάλλου δημιουργούν ένα σχήμα από κηλίδες πάνω σε φιλμ που τελικά στη συνέχεια προσκρούουν. Από το σχήμα αυτό που αποτυπώνεται στο φιλμ οι ειδικοί μελετητές αναγνωρίζουν έτσι τη γεωμετρική δόμηση. Αυτή η τεχνική χρησιμοποιείται επίσης και στη μελέτη κραμάτων και ορυκτών, ακόμη δε και στη διερεύνηση της δομής του DNA με καθοριστικό μάλιστα ρόλο στη Γενετική.[4]

- Η κρυσταλλογραφία που εφαρμόζεται με ακτίνες X καλείται κρυσταλλογραφία ακτίνων X.

2.2. Αναπαράσταση Μορίων

Η 3D δομή ενός μορίου αντιπροσωπεύεται από ένα σύνολο ατόμων. Κάθε άτομο που ανήκει σε αυτό το σύνολο έχει τη δομή σφαίρας με ορισμένη ατομική ακτίνα. Η τρισδιάστατη τοποθέτηση των ατόμων στο μοντέλο αναπαράστασης που παρουσιάζεται στην σκηνή (scene) ενός προγράμματος απεικόνισης βιολογικών μορίων καθορίζεται από τις εσωτερικές αποστάσεις των ατόμων και τις σχετικές γωνίες τους που σχηματίζουν μεταξύ τους και με τους δεσμούς που τα συγκρατούν.

- Ως Επιφάνειες Μορίου μπορούμε να ορίσουμε τα όρια των σφαιρών.
- Οι σφαίρες αντιπροσωπεύουν τα άτομα και τα ραβδιά τους μεταξύ τους δεσμούς. Το μήκος των ραβδίων και οι γωνίες που σχηματίζουν αποτελούν βαθμούς ελευθερίας για κάθε μόριο.

2.3. Η Protein Data Bank

Η Protein Data Bank (PDB) αποτελεί τη μοναδική βάση παγκοσμίως όπου είναι κατατεθειμένες οι τρισδιάστατες δομές βιολογικών μακρομορίων. Ιδρύθηκε το 1971 στα Brookhaven National Laboratories (BNL) των ΗΠΑ και περιελάβανε 7 δομές μακρομορίων όπως αυτές προέκυψαν από κρυσταλλογραφικές μελέτες. Ο ρυθμός αύξησης των εγγραφών στη δεκαετία του '70 ήταν πολύ μικρός. Από το 1980 και μετά λόγω της τεχνολογικής εξέλιξης σε κάθε στάδιο του προσδιορισμού δομών ο ρυθμός προσθήκης δεδομένων στην PDB αυξήθηκε δραματικά. Πλέον στην βάση περιλαμβάνονται και δομές όπως προκύπτουν με φασματοσκοπία Πυρηνικού Μαγνητικού Συντονισμού (NMR) ή κρυσταλλογραφία ακτίνων X. Στην παρούσα φάση (Μάρτιος 2002) η PDB περιλαμβάνει 17493 δομές βιομορίων. Οι εγγραφές στην PDB εκτός από τις συντεταγμένες των ατόμων που απαρτίζουν τη δομή περιλαμβάνουν και επιπρόσθετα βοηθητικά στοιχεία όπως βιβλιογραφικές αναφορές, λεπτομέρειες για τον

προσδιορισμό της δομής καθώς και άλλα στοιχεία που προκύπτουν από τη συγκεκριμένη δομή. Κάθε δομή προτού διατεθεί στο κοινό υφίσταται έλεγχο για την ορθότητα της με τη χρήση ειδικού λογισμικού. Στη συνέχεια εφόσον περάσει τις δοκιμές με επιτυχία αποκτά ένα χαρακτηριστικό κωδικό και προστίθεται στη βάση.

2.4. Τα PDB αρχεία

Το είδος αρχείων που χρησιμοποιήθηκε αρχικά από την Protein Data Bank αποκαλούνταν PDB αρχεία . Αυτό το αρχικό σχήμα περιορίστηκε από το πλάτος των διάτρητων καρτών υπολογιστών σε 80 χαρακτήρες ανά γραμμή. Περίπου το 1996, το σχήμα "μακρομοριακών κρυσταλλογραφικών αρχείων πληροφοριών", mmCIF, άρχισε να εκσυγχρονίζεται. Μια έκδοση XML αυτού του σχήματος, αποκαλούμενη PDBML, περιγράφηκε στις δομές αρχείων το 2005. Στην πραγματικότητα, τα μεμονωμένα αρχεία μεταφορτώνονται εύκολα στα πακέτα γραφικής παράστασης χρησιμοποιώντας τις διευθύνσεις Ιστού:

Για τα αρχεία σχήματος PDB, χρήση, π.χ.,
<http://www.pdb.org/pdb/files/4hhb.pdb.gz>

Για τα αρχεία PDBML (XML), χρήση, π.χ.,
<http://www.pdb.org/pdb/files/4hhb.xml.gz>

Η "4hhb" είναι το προσδιοριστικό PDB. Κάθε δομή που κοινοποιητε στην Protein Data Bank λαμβάνει τέσσερις προσδιοριστικούς χαρακτήρες, είναι το PDB ID. (Αυτό δεν μπορεί να χρησιμοποιηθεί ως προσδιοριστικό για τα βιομόρια, επειδή συχνά διάφορες δομές για το ίδιο μόριο- σε διαφορετικά περιβάλλοντα ή συνθήκες έχει περιληφθεί στο PDB με διαφορετικό PDB IDs.).

2.5. Η βιοπληροφορική

Ένας γενικός ορισμός .Η Βιοπληροφορική μπορεί να οριστεί ως: “η εφαρμογή υπολογιστικών τεχνικών και μεθόδων στην προσπάθεια κατανόησης και οργάνωσης των δεδομένων και πληροφοριών που σχετίζονται με τα βιολογικά μακρομόρια
Βιοπληροφορική αντιστοιχεί στην ανάπτυξη αλγορίθμων και βάσεων δεδομένων για την Μελέτη και την κατανόηση των βιολογικών συστημάτων,
Δομική Βιοπληροφορική ,αντιπροσωπεύει τον τομέα ο οποίος πραγματεύεται την δομή των βιομακρομορίων. Ο όρος Δομική Βιοπληροφορική συμπεριλαμβάνει την μελέτη των δομών του DNA, RNA των πεπτιδίων και των πρωτεϊνών.
Η γνώση των πρωτεϊνικών δομών επιτρέπει την διερεύνηση των βιολογικών διεργασιών άμεσα, σε βάθος και με υψηλή ανάλυση .

2.6. Η Βιομοριακή Προσομοίωση

Η (Βιο)Μοριακή Προσομοίωση επιτρέπει την μελέτη της δομής και της αλληλεπίδρασης πρωτεϊνών-πρωτεϊνών DNA RNA μέσω υπολογιστικών προσεγγίσεων σε υψηλή ανάλυση (λεπτομέρεια σε ατομικό επίπεδο) μέσω της μελέτης της τρισδιάστατης(3D) μοριακής δομής της πρωτεΐνης ή του πρωτεϊνικού συμπλόκου. Τέτοιες προσεγγίσεις είναι η προσομοίωση δομών μέσω υπολογιστικών μεθόδων:

- Πρόβλεψη μέσω ομολογίας ή Συγκριτική προσομοίωση
- Πρόβλεψη μέσω Threading
- ab initio πρόβλεψη

2.7. Η ακτίνα Van Der Waals ενός ατόμου

Ακτίνα Van Der Waals, είναι το μισό της απόστασης που μπορούν να πλησιάσουν οι πυρήνες δυο ομοίων ατόμων που συνδέονται με δυνάμεις Van Der Waals. Το όνομα δημιουργήθηκε στην αρχή της δεκαετίας του '70 μετά που ο Ολλανδός επιστήμονας Johannes Diderik Van Der Waals, παρατήρησε τις δυνάμεις που αναπτύσσονται είτε είναι η ελκυστική είτε η αποκρουστική δύναμη ανάμεσα στα μόρια η μεταξύ μερών των μορίων αυτών.

2.8. Μοριακή μοντελοποίηση

Η Μοριακή Μοντελοποίηση, αποτελεί ένα νέο και ταυτόχρονα γοργά αναπτυσσόμενο επιστημονικό κλάδο που συνδυάζει σε μεγάλο βαθμό τις επιστήμες της Βιολογίας και της Πληροφορικής. Η Μοριακή Μοντελοποίηση προσπαθεί να μιμηθεί τη συμπεριφορά των μοριακών συστημάτων, βασισμένη σε μεγάλο βαθμό στη σχεδίαση μοντέλων μορίων με τη βοήθεια ηλεκτρονικού υπολογιστή.

Τα σύγχρονα πακέτα λογισμικού μοριακής σχεδίασης, αποτελούν χρήσιμα εργαλεία στα χέρια των ερευνητών, οι οποίοι έχουν τη δυνατότητα να συνδυάσουν τη θεωρία και το πείραμα. Το μόνο ίσως μειονέκτημα είναι ότι δεν υπάρχει ένα γενικότερο και διευρυμένο εργαλείο μοριακής σχεδίασης. Το σύνολο των βιολογικών μορίων που μελετάμε στη μοριακή σχεδίαση ποικίλλει από μεμονωμένα μόρια και απλές ατομικές δομές σε πολυμερή και βιολογικά μακρομόρια όπως οι πρωτεΐνες και το DNA. Έτσι το επιλεγόμενο κάθε φορά μοντέλο πρέπει σαφώς να καλύπτει τις ιδιαιτερότητες και τα χαρακτηριστικά του προς σχεδίαση συστήματος.

Η Μοριακή Μοντελοποίηση ακολουθεί τα εξής 3 βασικά βήματα: α)επιλογή του κατάλληλου μοντέλου που περιγράφει ικανοποιητικά τις ενδομοριακές και εσωμοριακές συσχετίσεις του μορίου, β)υπολογισμός της ενεργειακής κατάστασης του συστήματος και ελαχιστοποίησή της και γ)ανάλυση των παραπάνω υπολογισμών και έλεγχος της τελικής διαμόρφωσης ώστε να ικανοποιούνται όλες οι συνθήκες και περιορισμοί που ο σχεδιαστής έχει θέσει.

Αν και στη Μοριακή Σχεδίαση το σύνολο των συντεταγμένων παρέχει μια χρήσιμη απεικόνιση του μορίου σε δισδιάστατο και τρισδιάστατο επίπεδο, είναι αναγκαία και η χρήση συμπληρωματικών πληροφοριών που έχουμε στη διάθεσή μας προκειμένου να αναπαραστήσουμε και τις βιοχημικές ιδιότητες του μορίου. Σημαντική πρόκληση σε αυτή την κατεύθυνση αποτελεί η ικανοποιητική απεικόνιση των επιφανειών των μορίων οι οποίες είναι υπεύθυνες για τη συμπεριφορά των μορίων και τις φυσικοχημικές τους ιδιότητες.

Οι σύγχρονες τεχνικές των μοριακών γραφικών (molecular graphics) επιτρέπουν την απεικόνιση της τρισδιάστατης αρχιτεκτονικής των μορίων στην ενεργειακά ευνοϊκότερη διαμόρφωση. Στη δομή αυτή είναι δυνατόν να επέμβουμε απομονώνοντας τμήματα των μορίων, αλλάζοντας τον προσανατολισμό ορισμένων ομάδων ή ψάχνοντας για άλλες δυνατές διαμορφώσεις. Τα μοριακά αυτά μοντέλα επιτρέπουν επίσης την απεικόνιση φυσικοχημικών χαρακτηριστικών που επηρεάζουν τις αλληλεπιδράσεις ενός μορίου με άλλα μόρια. Είναι π.χ. δυνατόν να παρασταθούν οι ακτίνες Van der Waals των ατόμων, η μοριακή επιφάνεια ή ο μοριακός όγκος, το μοριακό ηλεκτροστατικό δυναμικό, η ηλεκτρονιακή πυκνότητα κ.ά.. Η κατασκευή μοριακών μοντέλων επιτρέπει εξάλλου τη σύγκριση ανάμεσα σε διαφορετικά μόρια προσδιορίζοντας περιοχές ομοιοτήτων και διαφορών. Παράλληλα είναι δυνατή η απεικόνιση της τρισδιάστατης προσαρμογής ενός

μικρομορίου (φαρμάκου) σε ένα μακρομόριο (υποδοχέα). Σε αυτή την κατεύθυνση είναι δυνατό να σχεδιαστούν στην οθόνη του ηλεκτρονικού υπολογιστή νέα μόρια τα οποία μιμούμενα το σχήμα μιας δραστικής ένωσης ή του φυσιολογικού υποστρώματος έχουν θεωρητικά τη δυνατότητα να καταλάβουν την ενεργό θέση ενός υποδοχέα. Η μοριακή μοντελοποίηση με χρήση γραφικών σε υπολογιστές παρουσιάστηκε το 1960 και αντικατέστησε τα φυσικά μοντέλα αναπαράστασης.[4]

3. Η εφαρμογή

3.1. Σκοπός της εφαρμογής

Η εφαρμογή που δημιουργήθηκε είχε σαν σκοπό την οπτική αντιπροσώπευση δεδομένων βιοπληροφορίας, τα δεδομένα περιέχονται μέσα από τα παραθυρικά περιβάλλοντα που έχουν δημιουργηθεί. Μέσα από αυτά τα περιβάλλοντα που δημιουργήθηκαν πετύχαμε την βελτίωση της διεπαφής του προγράμματος Jmol. Μέσα από τα νέα συστατικά που δημιουργήθηκαν έγινε προσπάθεια να παραχθεί πληροφορία σχετική με την δομή και τα επιμέρους στοιχεία κυρίως ενός βιολογικού μακρομορίου, ώστε όσο είναι δυνατόν να απλουστεύσουμε την διαδικασία της κατανόησης της δομής και των επιμέρους στοιχείων ενός μορίου που αναπαρίσταται γραφικά. Η εφαρμογή δημιουργήθηκε για να λειτουργήσει στο περιβάλλον του Jmol και δεν λειτουργεί σαν ένα αυτόνομο πρόγραμμα, οπότε σε πολλές περιπτώσεις χρησιμοποιεί πόρους που παρέχει το πρόγραμμα Jmol.

3.2. Εμβέλεια της εφαρμογής (application scope)

Η εφαρμογή έχει ως στόχο την βελτίωση της παροχής δεδομένων σε ένα είδη υπάρχον πρόγραμμα παρουσίασης γραφικά βιολογικών μορίων, σαν άμεση συνέπεια η εφαρμογή που δημιουργήθηκε θα πραγματεύεται στοιχεία βιολογίας, σε αυτό το σημείο θέλει ιδιαίτερη προσοχή στην σωστή και έγκυρη αναπαραγωγή των στοιχείων αυτών. Για να μπορεί να χαρακτηριστεί η εφαρμογή ως επιτυχημένη θα πρέπει να πλήρη κάποιες βασικές προϋποθέσεις. Οι βασικές προϋποθέσεις είναι.

- Λειτουργικότητα
- Εναρμόνιση με το ήδη υπάρχον περιβάλλον του Jmol
- Απλούστευση διαδικασιών
- Ευκολία στην κατανόηση και την πρόσβαση στα δεδομένα
- Αναπαραγωγή έγκυρης και σωστής πληροφορίας

3.3. Περιγραφή της εφαρμογής

Η εφαρμογή γράφτηκε στην αντικειμενοστραφή γλώσσα προγραμματισμού Java και αναπτύχθηκε στην πλατφόρμα Java Net Beans, χρησιμοποιήθηκαν κατά κύριο λόγο συστατικά της τεχνολογίας Java Swing και λιγότερο της τεχνολογίας Java awt. Σε κάποια σημεία του κώδικα χρησιμοποιήθηκε και η γλώσσα προγραμματισμού HTML, περισσότερο για την μορφοποίηση κειμένου.

Η εφαρμογή που δημιουργήσαμε περιλαμβάνει μια λίστα μενού (VisualTools) που έχει προστεθεί στην κύρια μπάρα μενού του περιβάλλοντος του Jmol. Αυτή η λίστα

μενού περιέχει τρία υπομενού(submenu) που ονομάζονται Sequence , Visual , TreeView .

Το υπομενού Sequence ενεργοποιεί το παραθυρικό περιβάλλον SequenceView.

Το υπομενού Visual περιέχει άλλη μια κατηγορία υπομενού ,το addComment και το itemColor .Με το addComment ενεργοποιείτε η λειτουργία της εισαγωγής σχολίου και με το itemColor μπορούμε να επιλέξουμε από μια λίστα χρωμάτων τι χρώμα θέλουμε να είναι το γραφικό σχολίου που θα δημιουργήσουμε .Η επιλογή χρώματος έγινε κυρίως για να επιλέγουμε κάθε φορά τέτοιο χρώμα για τα γραφικά ώστε να ξεχωρίζουν από την υπόλοιπη δομή του μορίου που αναπαρίσταται,αναλογα με την επιλογη του χρηστη. Το υπομενού TreeView ενεργοποιεί το παραθυρικό περιβάλλον TreeView όπου απεικονίζεται το δέντρο της πρωτοταγούς δομής του μορίου που αναπαρίσταται.

Η εφαρμογή περιλαμβάνει τρία διαφορετικά περιβάλλοντα υλοποίησης-ενότητες όπως περιγράφηκαν περιληπτικά και παραπάνω.

- Η πρώτη ενότητα αναφέρεται στην δημιουργία ενός παραθυρικού περιβάλλοντος που ονομάστηκε άποψη ακολουθίας (SequenceView).Το συγκεκριμένο παραθυρικό περιβάλλον περιλαμβάνει την ακολουθία αμινοξέων (παράρτημα Γ) της βιολογικής δομής που αναπαρίσταται κάθε φορά από το Jmol.Δίνει την δυνατότητα στο χρήστη να επιλέξει κάθε φορά οποιοδήποτε αμινοξύ επιθυμεί ,με την επιλογή επισημαίνονται τα άτομα που συγκροτούν το αμινοξύ στην σκηνή του Jmol.Ακόμα παρέχεται η λειτουργία maxatom όπου ο χρήστης μπορεί να μεγεθύνει τα αμινοξέα που επισημαίνονται κάθε φορά .Μέσα από την συγκεκριμένη εφαρμογή ο χρήστης έχει την δυνατότητα να πληροφορηθεί τα χαρακτηριστικά χρώματα των αμινοξέων ,αυτή η πληροφορία γίνεται δημόσια ενεργοποιώντας το παράθυρο με τίτλο amino acid color (περισσότερες πληροφορίες σχετικά παρέχονται στο παράρτημα Γ).Η ακολουθία αμινοξέων που περιγράφεται στο συγκεκριμένο περιβάλλον περιέχεται και ανακτάται από τα αρχεία pdb (protein data bank files) που περιγράφουν την δομή ενός μορίου .Τα αρχεία pdb αναλύονται περαιτέρω στο παράρτημα Β της παρούσας εργασίας.
- Η δεύτερη ενότητα περιλαμβάνει ένα παραθυρικό περιβάλλον ,το οποίο ονομάστηκε άποψη δέντρου (TreeView).Το συγκεκριμένο περιβάλλον περιέχει σχετική πληροφορία που περιγράφει με την «μορφή δέντρου» την πρωτοταγή αναπαράσταση(παράρτημα Γ) της βιολογικής δομής που απεικονίζεται κάθε φορά στο Jmol. Κάθε δομικό συστατικό που αποτελεί την πρωτοταγή αναπαράσταση του μορίου αναπαρίσταται με την μορφή κόμβου του δέντρου,ο χρήστης μπορεί να επιλέξει οποιοδήποτε κόμβο του συγκεκριμένου δέντρου ,ο οποίος επισημαίνεται στην συνέχεια πάνω στο μόριο στην σκηνή του Jmol.Η επισήμανση του κόμβου επιτυγχάνεται επισημαίνοντας ένα ένα το μικρότερο δομικό συστατικό που τον αποτελεί,το άτομο.
- Η τρίτη ενότητα βασίστηκε στην δημιουργία ενός γραφικού αντικειμένου τύπου σχολίου (Comment mode) που θα παρουσιάζεται πάνω από τα άτομα που θα απεικονίζονται γραφικά στην σκηνή του προγράμματος (κάνοντας κλικ απλά κάθε φορά στο άτομο που θα επιλέξουμε) και θα επιτρέπει στον χρήστη να αλληλεπιδρά με την σκηνή του Jmol και να εισάγει μέσω αυτού του γραφικού κάποιο κείμενο η σχόλιο που θα συνοδεύει οποιοδήποτε άτομο επιλεγεί .Η συγκεκριμένη πληροφορία είναι δυνατό να αποθηκεύεται σε ένα αρχείο αποθήκευσης και να επανακτάται από το αρχείο αυτό.

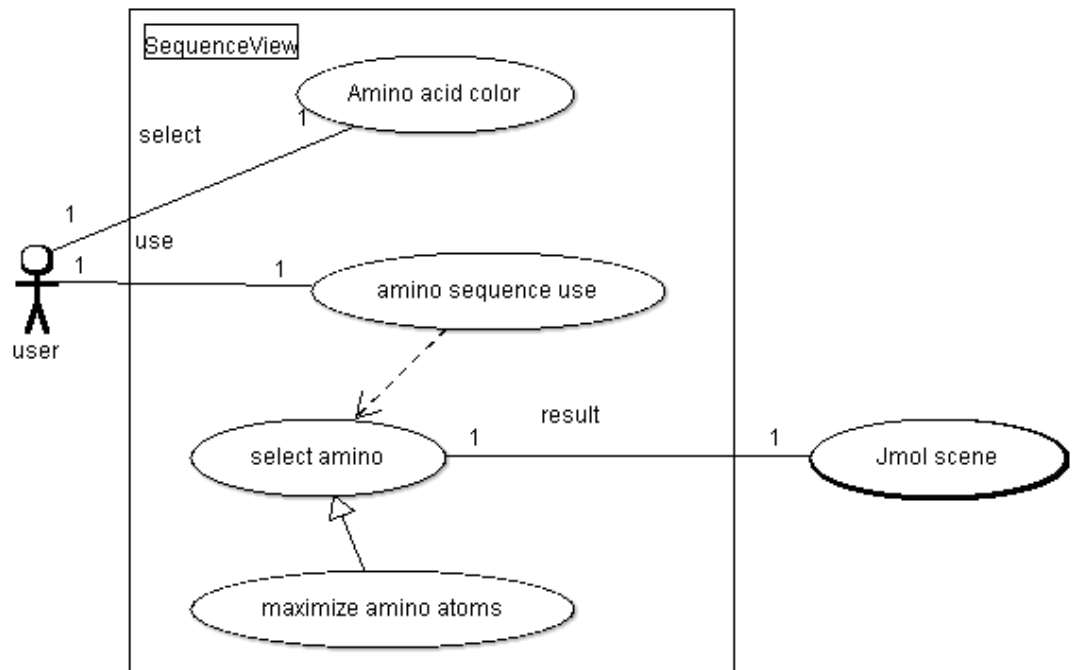


Figure 1: Λειτουργία εφαρμογής SequenceView

Η διαδικασία λειτουργίας της εφαρμογής SequenceView. Ο χρήστης μέσω του μενού επιλογής ενεργοποιεί το παραθυρικό περιβάλλον της εφαρμογής SequenceView, έτσι ξεκινάει η αλληλεπίδραση χρήστη και εφαρμογής. Η εφαρμογή παρουσιάζει την ακολουθία αμινοξέων, ο χρήστης έχει την δυνατότητα να επιλέξει να ανοίξει το παράθυρο επεξήγησης των αμινοξέων και των χρωμάτων τους (Amino acid color) ώστε να κατανοήσει καλύτερα την ακολουθία αμινοξέων και τα χαρακτηριστικά τους χρώματα που αναπαρίστανται. Ο χρήστης μπορεί να χρησιμοποιήσει την ακολουθία αμινοξέων επιλέγοντας ένα αμινοξύ την φορά και βλέποντας την αναπαράσταση πάνω στο μόριο στην σκηνή του Jmol. Η εφαρμογή δίνει την δυνατότητα στο χρήστη να μεγεθύνει το αμινοξύ που έχει επιλέξει ώστε να είναι ευδιάκριτο στην σκηνή του Jmol.

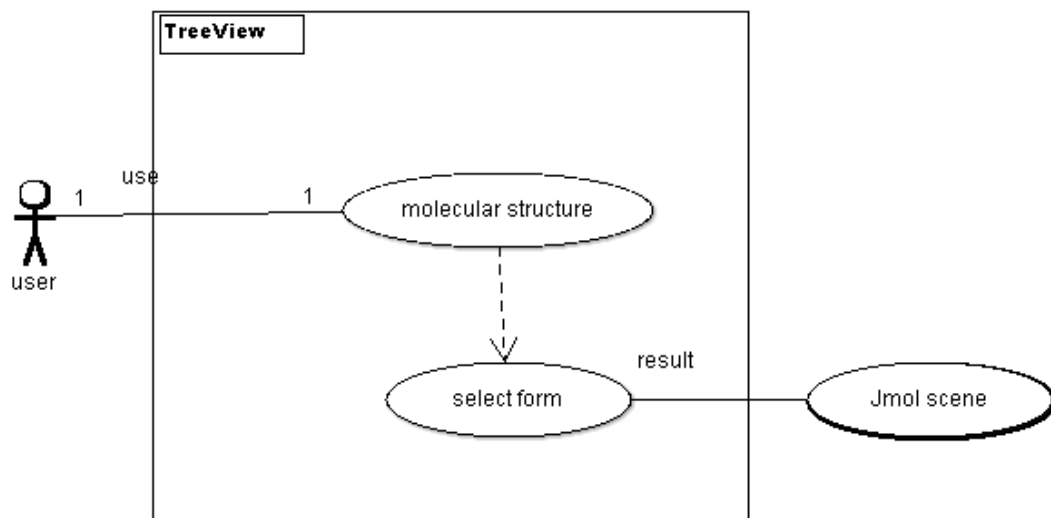


Figure 2: Λειτουργία εφαρμογής TreeView

Η διαδικασία λειτουργίας της εφαρμογής TreeView .Ο χρήστης μέσω του μενού επιλογής ενεργοποιεί το παραθυρικό περιβάλλον της εφαρμογής TreeView,έτσι ξεκινάει η αλληλεπίδραση χρήστη και εφαρμογής,στο παραθυρικό περιβάλλον της εφαρμογής απεικονίζεται σε μορφή δέντρου η πρωτοταγής δομή του μορίου .Ο χρήστης επιλέγοντας οποιοδήποτε δομικό συστατικό του μορίου από το παραθυρικό περιβάλλον της εφαρμογής ,μπορεί να δει μαρκαρισμένα τα άτομα που αποτελούν το δομικό αυτό συστατικό στην σκηνή του Jmol .

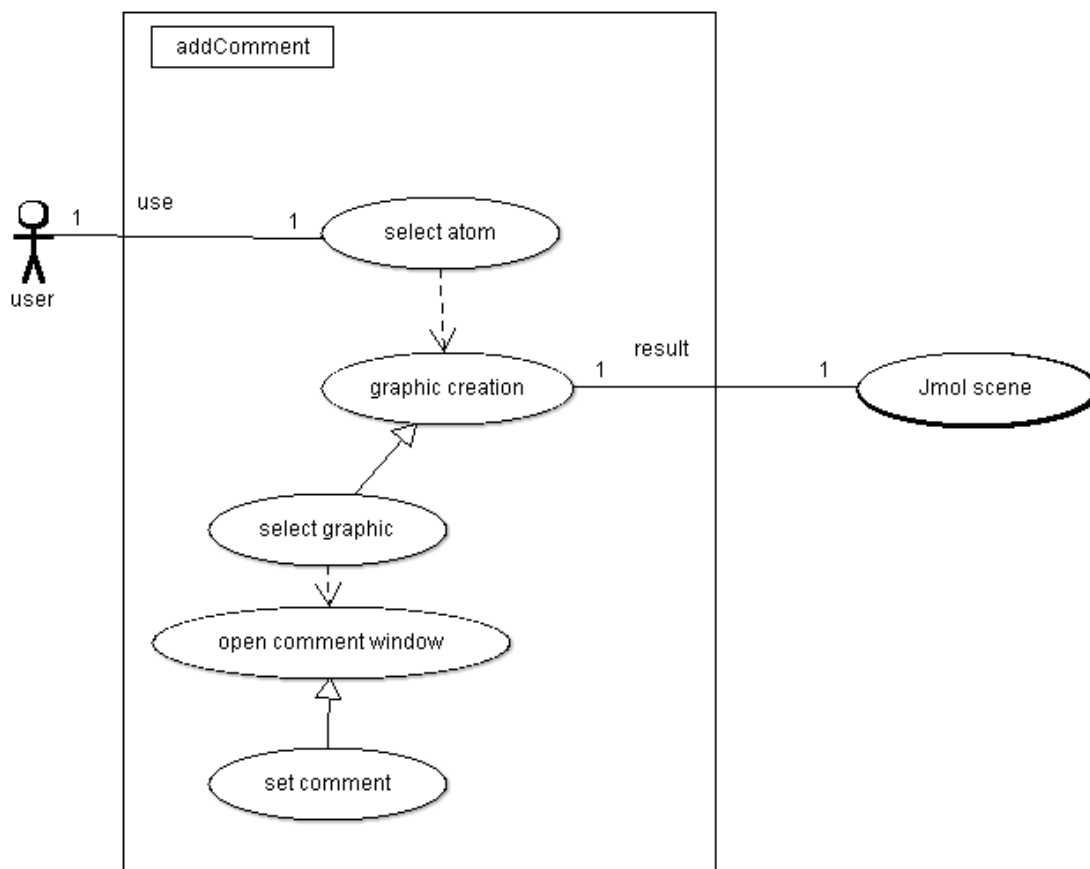


Figure 3: Λειτουργία εφαρμογής Comment mode

Η διαδικασία λειτουργίας της εφαρμογής Comment mode. Ο χρήστης επιλέγοντας από το μενού Visual την επιλογή addComment, εισάγει την εφαρμογή στην κατάσταση εισαγωγής σχολίου (Select item mode), όπου η εφαρμογή περιμένει να λάβει ως είσοδο της συντεταγμένες ενός ατόμου στην σκηνή (αυτή η επιλογή γίνεται με ένα απλό κλικ πάνω στο άτομο που επιλέγουμε κάθε φορά). Επιλέγοντας το άτομο που θέλουμε να εισαχθεί το σχόλιο παραμετροποιούμε την εφαρμογή εμφανίζοντας το αποτέλεσμα στην σκηνή (scene result). Στην συνέχεια για να εισάγουμε οποιοδήποτε σχόλιο θα πρέπει να κάνουμε κλικ πάνω στο γραφικό εισαγωγής σχολίου που δημιουργήθηκε, ανοίγοντας το παραθυρικό περιβάλλον όπου μπορούμε να εισάγουμε το σχόλιο.

4. Προγράμματα αναπαράστασης βιολογικών μορίων

Η πολύπλοκη δομή των μορίων γίνεται ευκολότερα κατανοητή με τρισδιάστατα μοντέλα και όχι με δυσδιάστατες αναπαραστάσεις. Επομένως, η χρήση τρισδιάστατων γραφικών συστημάτων για την προσομοίωση μοριακών ενώσεων, βοηθά κατά πολύ στην αντίληψή τους αλλά μπορεί να χρησιμοποιηθεί και σαν εργαλείο για την ανίχνευση καινούριων ενώσεων.

Η δομή στο χώρο ενός βιομορίου αποτελεί το βασικότερο παράγοντα για τον καθορισμό των χημικών του ιδιοτήτων και της λειτουργίας του. Έτσι κατά τη μελέτη των χημικών ιδιοτήτων βιολογικών μακρομορίων η αναπαράσταση της δομής τους στο χώρο είναι ιδιαίτερα σημαντική για την κατανόηση της σχέσης μεταξύ των δομικών χαρακτηριστικών των μορίων και της λειτουργίας τους. Για το σκοπό αυτό κατά το παρελθόν χρησιμοποιούνταν διάφορων τύπων μοντέλα (σκελετικά, χωροπληρωτικά) κατασκευασμένα από μέταλλο, πλαστικό ή ξύλο δίνοντας μας μία άποψη της δομής των μορίων στο χώρο. Το βασικό τους μειονέκτημα βέβαια είναι ότι απαιτείται μεγάλο χρονικό διάστημα για την κατασκευή τους το οποίο γίνεται ακόμη μεγαλύτερο όταν πρόκειται να αναπαρασταθούν μεγαλομόρια όπως δομές πρωτεϊνών και νουκλεϊκών οξέων καθώς και το γεγονός ότι είναι πολύ ευαίσθητα στη χρήση τους.

Η χρήση μοριακών γραφικών με τη βοήθεια υπολογιστή έχει διευκολύνει σημαντικά τη διαδικασία αναπαράστασης δομών βιομορίων στο χώρο. Βέβαια είναι ιδιαίτερα δύσκολο να καταδειχθεί όλη η σημαντική δομική πληροφορία σε μια αναπαράσταση δύο διαστάσεων όπως συμβαίνει στην οθόνη του υπολογιστή. Καμία αναπαράσταση σε 2 διαστάσεις δεν είναι δυνατό να καταδείξει όλες τις λεπτομέρειες μιας πραγματικής τρισδιάστατης αναπαράστασης. Για αυτό το σκοπό έχουν αναπτυχθεί μια σειρά από τρόπους αναπαράστασης με καθένα από αυτούς να παρουσιάζει μια συγκεκριμένη άποψη του μορίου. Ανάλογα λοιπόν με τα χαρακτηριστικά της δομής που επιθυμεί ο χρήστης να μελετήσει μπορεί να επιλέξει και τον κατάλληλο τρόπο αναπαράστασης.

Μέσω της χρήσης του υπολογιστή προσπαθείται να δημιουργηθεί μια εικόνα στην οθόνη κατά τέτοιο τρόπο ώστε να δίνει την εντύπωση τρισδιάστατης απεικόνισης των αντικείμενων. Αυτό επιτυγχάνεται με τις κατάλληλες εναλλαγές φωτισμού στα αντικείμενα δίνοντας την εντύπωση του βάθους, προσομειώνοντας τελικά τη στερεοσκοπική όραση. Στις μέρες μας έχουν αναπτυχθεί μια σειρά από προγράμματα μοριακών γραφικών που προσφέρουν πολλούς δυνατούς τρόπους αναπαράστασης ενώ ταυτόχρονα μπορούν να εκτελέσουν διαφόρων ειδών υπολογισμούς. Πολλά από αυτά έχουν εκδόσεις ώστε να εκτελούνται σε μια πληθώρα λειτουργικών συστημάτων όπως UNIX, MS-WINDOWS, MS-DOS. Για την απεικόνιση των δομών στο χώρο τα προγράμματα αυτά συνήθως διαβάζουν αρχεία συντεταγμένων ατόμων υπό κατάλληλη μορφοποίηση. Ένας χαρακτηριστικός τύπος αρχείων συντεταγμένων είναι τα αρχεία PDB. Πρόκειται για ένα πρότυπο οργάνωσης πληροφορίας που πρωτοεφαρμόστηκε στην Protein Data Bank μια βάση δεδομένων όπου κατατίθενται οι δομές βιολογικών μακρομορίων προσδιορισμένες με κρυσταλλογραφία ακτίνων-X και φασματοσκοπία NMR.

Παρακάτω ακολουθεί μια συλλογή από προγράμματα που διατίθενται (ημι-)ελεύθερα μέσω του Διαδικτύου για ακαδημαϊκή χρήση.

- PYMOL: <http://pymol.sourceforge.net/>
- GRASP: <http://trantor.bioc.columbia.edu/grasp/>
- O: <http://origo.imsb.au.dk/~mok/o/>
- MOLSCRIPT: <http://www.avatar.se/molscript/>
- RASMOL: <http://www.umass.edu/microbio/rasmol/>
- Swiss-PdbViewer: <http://www.expasy.ch/spdbv/>
- CHIME plug-in: <http://www.mdlchime.com/chime/>

4.1. Τι εξυπηρετούν

Είναι προγράμματα τρισδιάστατης μοριακής οπτι-κοποίησης που μοντελοποιούν τις ιδιότητες συμμετρίας και τους κανονικούς τρόπους δόνησης της μοριακής δομής. Έτσι τα προγράμματα αυτά παρέχουν:

- Μια πληθώρα χαρακτηριστικών μορίων που καλύπτουν όλες τις περιπτώσεις στα γνωστικά αντικείμενα της μοριακής συμμετρίας και των κανονικών τρόπων δόνησης.
- Παραδείγματα τα οποία δεν έχουν οπτικοποιηθεί ποτέ ως σήμερα, όπως για παράδειγμα τα στοιχεία και οι διεργασίες συμμετρίας μορίων που ανήκουν στην ομάδα σημείου I_h .
- Τρισδιάστατη οπτικοποίηση των μοριακών δομών και των αντίστοιχων χημικών εννοιών. Ο χρήστης είναι ελεύθερος να χειριστεί τα μοριακά μοντέλα, να τα περιστρέψει, να τα μετακινήσει και να αυξομειώσει το μέγεθός του ώστε να αντιληφθεί με τον καλύτερο τρόπο τη δομή και τη δυναμική φύση των διεργασιών συμμετρίας και των κανονικών τρόπων δόνησης των μορίων.
- Νέα και πρωτότυπα συμβολικά συστήματα με δυναμικά επιφανειακά χαρακτηριστικά που επιτρέπουν τη νοητική σύνδεση των οπτικοποιήσεων με τις υποκείμενες χημικές έννοιες.
- Ελεύθερη και διερευνητική πρακτική εξάσκηση με τις οπτικοποιημένες χημικές έννοιες.
- Δυναμική οπτικοποίηση της εσωτερικής κίνησης και παραμόρφωσης των μοριακών δομών.

4.2. Που χρησιμοποιούνται

Τα Προγράμματα απεικόνισης βιολογικών δομών χρησιμοποιούνται σήμερα ευρύτατα από βιολόγους και χημικούς για την αναπαράσταση μοριακών συστημάτων καθώς και στον υπολογισμό η πρόβλεψη των φυσικοχημικών και βιοχημικών τους ιδιοτήτων . Χρησιμοποιούνται ακόμα για τον σχεδιασμό και την κατασκευή φαρμάκων , η τεχνική ονομάζεται «Βασισμένος στη Δομή Σχεδιασμός Φαρμάκων» (Structure-based drug design) και έχει ως αντικείμενο τον σχεδιασμό τρισδιάστατων φαρμακευτικών μορίων . Ακόμα χρησιμοποιούνται για:

- την κατανόηση των βιολογικών διαδικασιών ,
- την μοντελοποίηση και τη σύγκριση γονιδιωμάτων ,
- την μοντελοποίηση της σύγκρισης προτεϊνών ,
- και την κατασκευή φαρμάκων.

4.3. Από πού αντλούν πληροφορίες

Τα προγράμματα απεικόνισης βιολογικών δομών στις επιτραπέζιες (desktop) εκδόσεις τους, όπως και το πρόγραμμα μοριακής αναπαράστασης Jmol αντλούν της πληροφορίες δομής των μοντέλων που αναπαριστούν από ειδικά σχεδιασμένα αρχεία με κατάλληλη δομή πληροφοριών και μορφοποίηση, που ονομάζονται pdb files (protein data bank files) ή από εξειδικευμένες βάσεις δεδομένων που παρέχουν τέτοια πληροφορία στο διαδίκτυο.

4.4. Το πρόγραμμα απεικόνισης βιολογικών δομών Jmol

Το Jmol είναι ένα ανοικτού κώδικα πρόγραμμα απεικόνισης τρισδιάστατων μοντέλων μορίων, το πρόγραμμα έχει γραφεί στην γλώσσα προγραμματισμού Java για την έκδοση που χρησιμοποιείται ως εγκατεστημένο πρόγραμμα στους υπολογιστές. Υπάρχει και έκδοση του Jmol που χρησιμοποιείτε δικτυακά χρησιμοποιώντας applets ως ενσωματωμένο αντικείμενο σε ιστοσελίδες. Την πληροφορία της δομής ενός μορίου ώστε να το απεικονίσει το Jmol την αντλεί από ειδικά αρχεία που περιγράφουν την δομή ενός μορίου με κατάλληλο τρόπο ώστε να μπορούν να διαβάζονται από τα προγράμματα απεικόνισης μορίων και συνάμα και από το Jmol. Τα αρχεία αυτά έχουν κατάληξη pdb και παρέχονται δικτυακά από την πρωτεϊνική τράπεζα δεδομένων (protein data bank), περισσότερες πληροφορίες για τα συγκεκριμένα αρχεία παρέχονται στο παράρτημα αυτού του συγγράμματος. Πριν από μερικά χρόνια το Jmol αναδύθηκε ως ένα εφικτό σχέδιο αντικατάστασης για δυο παλαιότερα προγράμματα απεικόνισης βιολογικών δομών το Rasmol και το Chime. Το Jmol τρέχει σε λειτουργικό Windows, Linux και Mac.

Ως εφαρμογή το Jmol χρησιμοποιείται στους τομείς της επιστήμης της βιολογίας και της χημείας για την αναπαράσταση μορίων με κατανοητό τρόπο ακόμα και σε έναν απλό χρήστη με στοιχειώδη γνώση βιολογίας και χημείας. Το πρόγραμμα παρέχει πάρα πολλά εργαλεία ώστε να μπορεί η πληροφορία που απεικονίζει να γίνει κατανοητή και επεξεργάσιμη ώστε να μπορούν να βγούν από αυτή την πληροφορία συμπεράσματα από τα πιο απλά όπως από ποια άτομα αποτελείται το μόριο που απεικονίζεται, πόσα άτομα οξυγόνου, άζωτου, άνθρακα και άτομα σιδήρου περιέχει η δομή του, ακόμα ποια άτομα είναι διαλύτες, ποια είναι άτομα νερού και άλλες πολλές πληροφορίες που είναι εύκολο να αποσπάσει κανείς από μια αναπαράσταση ενός μορίου από το Jmol. Το Jmol περιλαμβάνει πάρα πολλές λειτουργίες και εργαλεία ώστε η πληροφορία που παρέχει να είναι χρήσιμη, εύκολα προσβάσιμη και κατανοητή.

Κύριο μενού του
προγράμματος

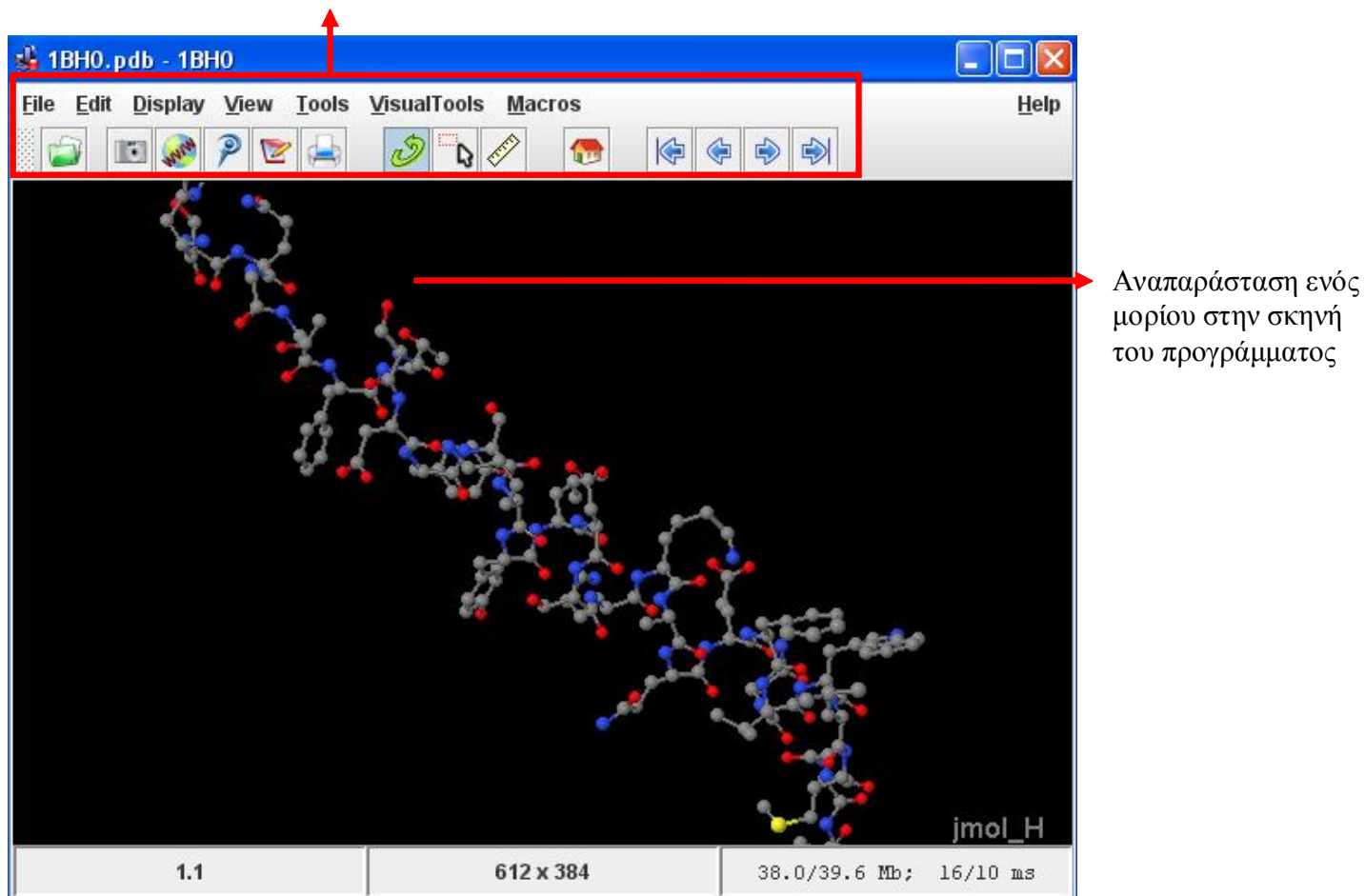


Figure 4: Περιβάλλον εργασίας του Jmol

4.4.1. Η αρχιτεκτονική συστήματος του Jmol

Για την κατασκευή της εφαρμογής που δημιουργήθηκε ήταν αναγκαίο μιας και η εφαρμογή αυτή δημιουργήθηκε στα πλαίσια του προγράμματος Jmol και χρησιμοποιεί πόρους του προγράμματος αυτού, να μελετηθεί και να αναλυθεί η αρχιτεκτονική του πηγαίου προγράμματος Jmol. Σε αυτή την ενότητα θα μιλήσουμε για τα πιο σημαντικά συστατικά της αρχιτεκτονικής του προγράμματος Jmol και θα δώσουμε μια γενική περιγραφή τους.

Στο πρόγραμμα Jmol υπάρχει υλοποιημένος ένας διαχειριστής γεγονότων που χειρίζεται την εισαγωγή δεδομένων, την εξαγωγή της πληροφορίας και διανέμει τους απαραίτητους πόρους στα επιμέρους συστατικά του συστήματος. Ο χρήστης εισάγει στο σύστημα την κατάλληλη πληροφορία για το μόριο που πρόκειται να αναπαρασταθεί είτε

είναι από αρχείο pdb ,είτε από το δίκτυο . Η πληροφορία για την δομή του μορίου που περιέχεται στο αρχείο pdb η βρίσκεται στο δίκτυο φορτώνεται από το PDB Loader Module.Ο PDB Loader Module δημιουργεί ένα αντικείμενο του μορίου, το οποίο είναι αποτέλεσμα των διάφορων java κλάσεων , μετά την δημιουργία του το αντικείμενο του μορίου περνά στο main event handler module (χειριστής γεγονότος). Αυτό το αντικείμενο του μορίου περνά έπειτα στην ενότητα γραφικής παράστασης(Graphics Module) ,η οποία είναι αρμόδια για την παραγωγή του τρισδιάστατου προτύπου του αντίστοιχου μορίου γραφικά .Το συστατικό της γραφικής παράστασης (Graphics Module) ,σε μια τέτοια εφαρμογή είναι το πιο κρίσιμο για την επιτυχία της εφαρμογής . Στην εικόνα παρακάτω αποτυπώνεται το γενικό σχέδιο λειτουργίας του συστήματος του Jmol.

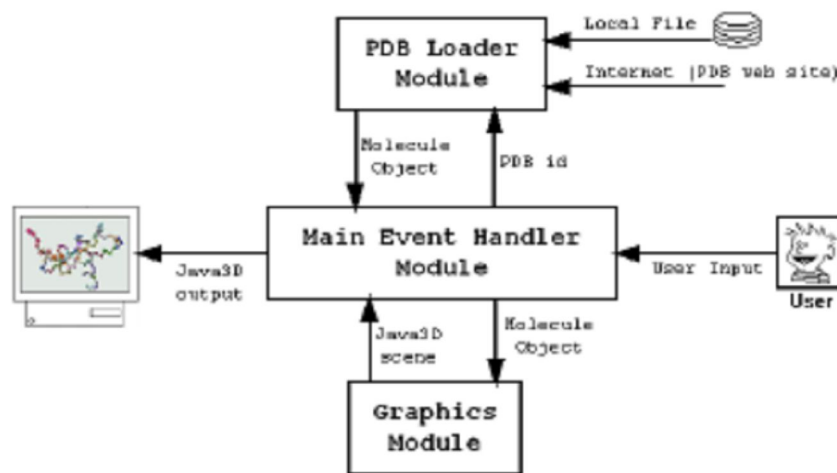


Figure 5: Αρχιτεκτονική συστήματος,τα βασικά συστατικά

4.5. Άλλα πρόγραμματα απεικόνισης βιολογικών δομών

4.5.1. Το Pymol

Η εφαρμογή Pymol είναι μια εφαρμογή αναπαράστασης βιολογικών μορίων γραφικά ,χρησιμοποιεί τρισδιάστατες αναπαραστάσεις μορίων και διαφόρων χημικών δομών .Η εφαρμογή σχεδιάστηκε στο πνεύμα παλαιότερων προγραμμάτων του RasMol και της ανοικτής έκδοσης του RasMol και προορίζεται για αναπαράσταση βιολογικών και χημικών δομών περιλαμβάνοντας την ατομική ανάλυση X-ray κρυσταλλικές κατασκευές των πρωτεϊνών των νουκλεοτιδίων (DNA ,RNA ,tRNA) και υδατάνθρακες καθώς και μικρές δομές μορίων και άλλα σύμπλοκα συμπεριλαμβανομένων των ανόργανων αλάτων και διαλυτών μόριων .Το Pymol παρέχει ακαδημαϊκή έρευνα και εκπαίδευση στην χημεία και στην βιολογία και στην βιοτεχνολογία και την φαρμακευτική επιστήμη αναπαριστώντας απεικόνιση ποιότητας των μακρομορίων .Ακόμα το πρόγραμμα παρέχει μοριακή απεικόνιση για της βιολογικά μικρότερες δομές περιλαμβανομένων το space-filling , atoms, chemical bonds, cartoon ribbons, molecular surfaces and solvent accessible surface meshes.

Το Pymol κάνοντας στερεοχημικές αναλύσεις μπορεί να απεικονίζει της τρισδιάστατες στερεοχημικές σχέσεις της οργανικής χημείας .Απεικονίζοντας την κατασκευή της κρυσταλλογραφικής πρωτεΐνης ,το Pymol μπορεί να απεικονίσει εκατομμύρια άτομα που περιέχονται σε μια δομή κρυσταλλικού πλέγματος.

Η μοριακές αλληλεπιδράσεις μπορούν να αναπαρασταθούν χρησιμοποιώντας interatomic distances, van der Waals clashes, property-based, potential-based, & proximity-based coloring schemes, solid and mesh isosurface potentials, slice planes, and gradient field lines.

Η εφαρμογή υποστηρίζει επίσης χάρτες πυκνότητας ηλεκτρονίων ,ηλεκτροστατικούς δυναμικούς χάρτες και άλλα ογκομετρικά σύνολα δεδομένων, καθώς και αυθαίρετες 3D γεωμετρίες αποτελούμενες από τρίγωνα ,σφαίρες ,κυλίνδρους ,κώνους, ελληπσοιδή και κείμενο . Η επιλογές εξόδου περιλαμβάνουν την αντιγραφή και επικόλληση ,PNG εικόνες, QuickTime βίντεο, VRML, Maya OBJ, and PovRay αρχεία εισόδου.

Με την εφαρμογή Pymol μπορούν να δημιουργηθούν animations ενός μορίου ,τα animations μπορούν να δημιουργηθούν μέσω των απλών κινήσεων ενός αντικειμένου & της κίνησης φωτογραφικών μηχανών ή μέσω της εισαγωγής των τροχιών από τις μοριακές προσομοιώσεις δυναμικής και άλλων δυναμικών διαμορφωτικών συνόλων όπως οι τρισδιάστατες NMR δομές.

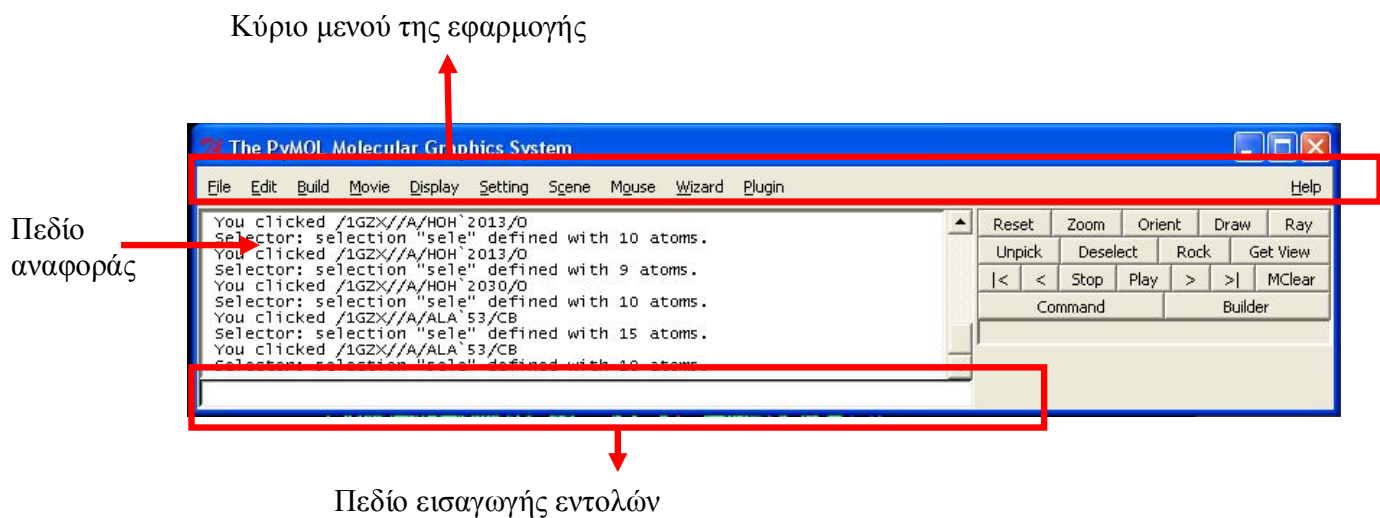


Figure 6: Το μενού του Pymol

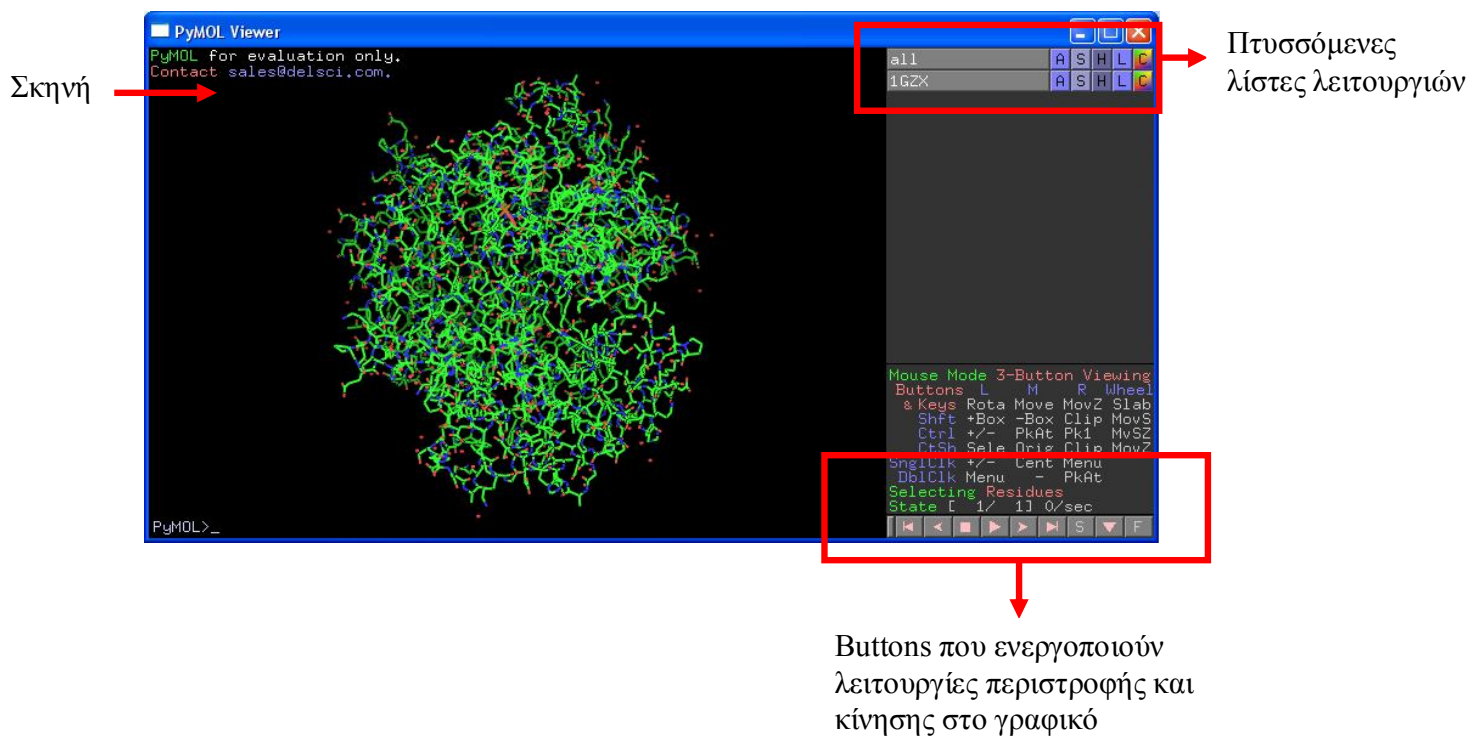


Figure 7: Το μενού του Pymol

5. Τεχνολογία που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής

5.1. Εισαγωγή

Η εφαρμογή που δημιουργήσαμε γράφτηκε εξ ολοκλήρου σε κώδικα Java. Για τα παραθυρικά περιβάλλοντα της εφαρμογής χρησιμοποιήθηκαν τα πακέτα της Java που παρέχουν συστατικά για την δημιουργία Graphical User Interface-GUI (στα ελληνικά Γραφική Διεπαφή με το Χρήστη) όπως είναι το πακέτο Java Swing, που χρησιμοποιήθηκε κατά κύριο λόγο και το πακέτο Java AWT. Σε κάποια σημεία του κώδικα χρησιμοποιήθηκε και η γλώσσα HTML, κατά κύριο λόγο στην μορφοποίηση κειμένου. Όλη η εφαρμογή αναπτύχθηκε στην πλατφόρμα Java Net Beans.

5.2. Η γλώσσα HTML

Τα αρχικά HTML προέρχονται από τις λέξεις HyperText Markup Language. Η html δεν είναι μια γλώσσα προγραμματισμού. Είναι μια γλώσσα σήμανσης (markup language), δηλαδή ένας ειδικός τρόπος γραφής κειμένου. Ο καθένας μπορεί να δημιουργήσει ένα αρχείο HTML χρησιμοποιώντας απλώς έναν επεξεργαστή κειμένου. Αποτελεί υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language) που επινοήθηκε από την IBM προκειμένου να λυθεί το πρόβλημα της μη τυποποιημένης εμφάνισης κειμένων στα διάφορα υπολογιστικά συστήματα. Ο browser αναγνωρίζει αυτόν τον τρόπο γραφής και εκτελεί τις εντολές που περιέχονται σε αυτόν. Αξίζει να σημειωθεί ότι η html είναι η πρώτη και πιο διαδεδομένη γλώσσα περιγραφής της δομής μιας ιστοσελίδας. Η html χρησιμοποιεί τις ειδικές ετικέτες (τα tags) να δώσει τις απαραίτητες οδηγίες στον browser. Τα tags είναι εντολές που συνήθως ορίζουν την αρχή ή το τέλος μιας λειτουργίας. Τα tags βρίσκονται πάντα μεταξύ των συμβόλων < και >. Π.χ. <BODY> Οι οδηγίες είναι case insensitive, δεν επηρεάζονται από το αν έχουν γραφτεί με πεζά (μικρά) ή κεφαλαία. Ένα αρχείο HTML πρέπει να έχει κατάληξη htm ή html.

5.2.1. Οι εντολές της HTML που χρησιμοποιήθηκαν στην εφαρμογή.

Για να ξεκινήσουμε να γράφουμε HTML πρέπει να χρησιμοποιήσουμε την εντολή <html>, αυτή η εντολή είναι η πρώτη για να ξεκινήσουμε τον κώδικα HTML, όταν τελειώσουμε τον κώδικα χρησιμοποιούμε την εντολή </html>.

- Για να παρουσιάσουμε έντονη γραφή χρησιμοποιούμε την εντολή και την κλείνουμε .

```
<b>Έντονη γραφή</b>
```

- Για να παρουσιάσουμε χρωματισμένο κείμενο χρησιμοποιούμε την εντολή μετά το ίσον δίνουμε το επιθυμητό χρώμα, η εντολή κλείνει .

```
<font color=#6F6F6F>Αυτό το κείμενο έχει μια απόχρωση του  
γκρι</font>
```

- Για να επιλέξουμε το μέγεθος της γραμματοσειράς υπάρχει η εντολή ``μετά το ίσον δίνουμε το μέγεθος και τέλος ``.

``Το μέγεθος των γραμμάτων``

5.3. Η γλώσσα προγραμματισμού Java

Η Java είναι υψηλού επιπέδου γλώσσα με τα ακόλουθα χαρακτηριστικά :

- **Αντικειμενοστραφής**
- **Ασφαλής (secure):** Απαγορεύει στους εισβολείς να γράφουν προγράμματα που μπορούν να χαλάσουν τα συστήματα των χρηστών.
- **Μεταφερτή (portable):** Μπορεί να εκτελείται σε διαφορετικές πλατφόρμες χωρίς τροποποίηση όπως σε Windows, σε Macintosh.
- **Κατανεμημένη (distributed):** Δηλαδή, διαφορετικά κομμάτια ενός προγράμματος μπορούν να προέλθουν από διαφορετικές πηγές.
- **Πολυνηματική (multithreaded):** Δηλαδή ένα απλό πρόγραμμα σε Java μπορεί να κάνει πολλά προγράμματα ανεξάρτητα και αλληλεπιδρώντα.
- **Ανεξαρτησία από πλατφόρμα** είναι η δυνατότητα το ίδιο πρόγραμμα να εκτελείται σε διαφορετικές πλατφόρμες και λειτουργικά συστήματα .

5.3.1. Αντικειμενοστρεφής προγραμματισμός

Στον αντικειμενοστρεφή προγραμματισμό (object oriented programming, OOP) ένα πρόγραμμα υπολογιστή είναι ένα σύνολο αλληλεπιδρώντων αντικειμένων. Μπορεί να ειπωθεί ότι αντικειμενοστρεφής προγραμματισμός είναι ένα τρόπος καλύτερης οργάνωσης προγραμμάτων. Στον αντικειμενοστρεφή προγραμματισμό, ένα πρόγραμμα δεν αποτελείται από τα δεδομένα και τον κώδικα του, αλλά από αντικείμενα (objects) τα οποία εμπεριέχουν τα δεδομένα και τα οποία (αντικείμενα) ανταλλάσσουν μεταξύ τους πληροφορίες και μηνύματα προκειμένου να επιτευχθεί ο στόχος του προγράμματος.

5.3.2. Κλάσεις και αντικείμενα

Η κλάση (class) είναι ένα πρότυπο που χρησιμοποιείται για δημιουργία πολλαπλών αντικειμένων με παρόμοια χαρακτηριστικά .Οι κλάσεις περιέχουν όλα τα χαρακτηριστικά ενός συγκεκριμένου συνόλου αντικειμένων. Όταν γράφετε ένα πρόγραμμα σε μια αντικειμενοστρεφή γλώσσα προγραμματισμού, δεν ορίζετε μεμονωμένα αντικείμενα αλλά κλάσεις αντικειμένων.Ο προγραμματιστής έχει πλεονεκτήματα δημιουργώντας ένα προσχέδιο για αντικείμενα με ίδιες ιδιότητες. Δηλαδή μπορεί να λεχθεί σε πιο ελεύθερη μορφή ότι κλάση είναι το προσχέδιο ή το πρωτότυπο, το οποίο ορίζει κοινές μεταβλητές και κοινές μεθόδους (συναρτήσεις) για τα αντικείμενα του ίδιου είδους. Σαν συμπέρασμα μπορεί να λεχθεί ότι ένα αντικείμενο (object) αναπαριστά μια ομάδα χαρακτηριστικών που εκτελούν κάποιες εργασίες και είναι αυτόνομο στοιχείο ενός προγράμματος. Τα αντικείμενα καλούνται επίσης και ομότυπα (instances).

5.3.3. Κληρονομικότητα (Inheritance)

Μία τάξη Java μπορεί να επεκταθεί (extend) από μία άλλη τάξη. Αυτή η άλλη τάξη ονομάζεται υποτάξη της αρχικής που είναι η υπερτάξη της. Μια υποτάξη κληρονομεί από την υπερτάξη της:

(α) Όλα τα μέλη ιδιωτικά, προστατευμένα και δημόσια, αν και μπορεί απευθείας να αναφερθεί μόνο στα προστατευμένα και στα δημόσια.

(β) Όλες τις μεθόδους ιδιωτικές, προστατευμένες και δημόσιες αν και μπορεί απευθείας να αναφερθεί μόνο στις προστατευμένες και στις δημόσιες.

(γ) Τον τύπο της υπερτάξης της. Για το σύστημα τύπων της Java αν η τάξη B είναι υποτάξη της τάξης A, τότε αντικείμενα της B θα γίνουν δεκτά οπουδήποτε θα απαιτούνταν αντικείμενα της A. Αυτή η συμβατότητα τύπων είναι μία πολύ ισχυρή ιδιότητα του αντικειμενοστραφούς προγραμματισμού.

5.4. Το πακέτο Java Abstract Windowing Toolkit(AWT)

Ένα Graphical User Interface-GUI (στα ελληνικά Γραφική Διεπαφή με το Χρήστη) είναι το μέρος του προγράμματος, που φροντίζει για τον τρόπο εμφάνισης και χειρισμού του προγράμματος από τον χρήστη. Ένα GUI αποτελείται από GUI-components. Οι κλάσεις που χρησιμοποιούνται για την κατασκευή αντικειμένων τύπου GUI-components ανήκουν στο java.awt (Abstract Windowing Toolkit) package. Οι βασικότερες από αυτές είναι η κλάση Component και η κλάση Container. Κάθε κλάση που κληρονομεί την κλάση Component είναι και αυτή ένα Component. Επίσης κάθε κλάση που κληρονομεί την κλάση Container είναι και αυτή ένα Container.

Οι κλάσεις που χρησιμοποιήθηκαν για την κατασκευή GUI-components στην εφαρμογή που δημιουργήθηκε περιγράφονται παρακάτω:

ΚΛΑΣΕΙΣ	ΛΕΙΤΟΥΡΓΙΑ
Button	Περιοχή που προξενεί ένα γεγονός (event) όταν επιλέγεται με το ποντίκι.
TextArea	Περιοχή όπου ο χρήστης εισάγει δεδομένα από το πληκτρολόγιο. Σε ένα TextArea μπορούμε να έχουμε πολλές γραμμές κειμένου σε αντίθεση με το TextField που μπορούμε να έχουμε μόνο μία.
List	Λίστα στοιχείων από τα οποία ο χρήστης μπορεί να επιλέξει ένα από αυτά με το ποντίκι. Διπλό κλικ σε ένα στοιχείο της λίστας προξενεί ένα action event.
Panel	Είναι ένα container αντικείμενο στο οποίο μπορούν να τοποθετηθούν component αντικείμενα.
ScrollPane	Είναι ένα container αντικείμενο στο οποίο μπορεί να τοποθετηθεί ένα component, συνήθως ένα Panel, το οποίο θα εμφανίζεται στον χρήστη αλλά κι όταν αυτό δεν είναι εφικτό θα εμφανίζονται Scrollbars που θα επιτρέψουν την διολίσθηση του component έτσι ώστε να γίνεται ορατό και το τμήμα του που δεν φαίνεται.
Label	Εμφανίζει κείμενο που δεν μπορεί να τροποποιηθεί από τον χρήστη.

TextField	Περιοχή όπου ο χρήστης εισάγει δεδομένα από το πληκτρολόγιο. Σε ένα TextField μπορούμε επίσης να εμφανίζουμε πληροφορίες.
------------------	---

Table 1: GUI-components AWT

Το επόμενο σχήμα εμφανίζει την ιεραρχία των GUI-components κλάσεων μαζί με κάποιες επιπλέον συμπληρωματικές κλάσεις που διαχειρίζονται ένα GUI.

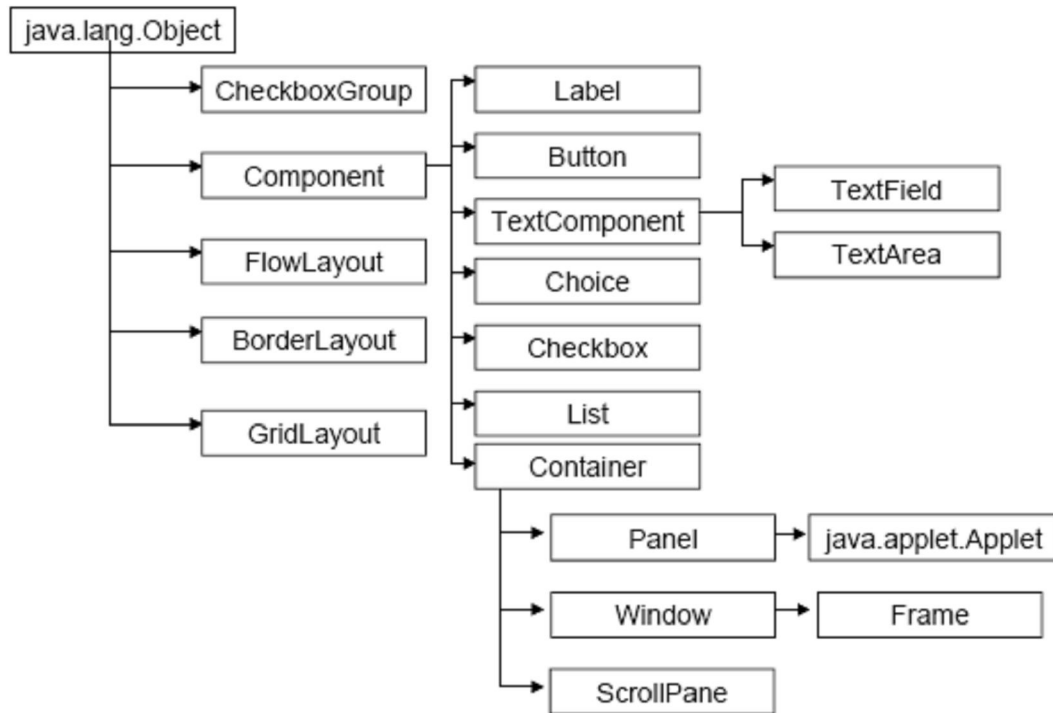


Figure 8: Ιεραρχία των GUI-components κλάσεων

5.4.1. Γεγονότα στα GUI (GUI Events)

Γεγονότα στέλνονται σε ένα πρόγραμμα Java από τα Windows. Η πληροφορία σχετικά με ένα GUI event, αποθηκεύεται σε ένα αντικείμενο που ανήκει στην κλάση ή σε υποκλάσεις της AWT event. Η AWT event είναι ορισμένη στο πακέτο java.awt.event. Το επόμενο σχήμα εμφανίζει την ιεραρχία των κλάσεων του πακέτου java.awt.event.

Για να μπορέσει ο προγραμματιστής να επεξεργαστεί κάποιο event το πετυχαίνει με δυο βήματα:
 A) Να δηλώσει έναν ακροατή γεγονότων (event listener) που θα ακούει τέτοιου τύπου events.

B) Να υλοποιήσει τον χειριστή γεγονότων (event handler) που θα πιάνει events που προκύπτουν από το βήμα A.

Ένας event listener πρέπει να είναι αντικείμενο κάποιας κλάσης, η οποία υλοποιεί ένα ή περισσότερα event-listener interface του πακέτου java.awt.event.

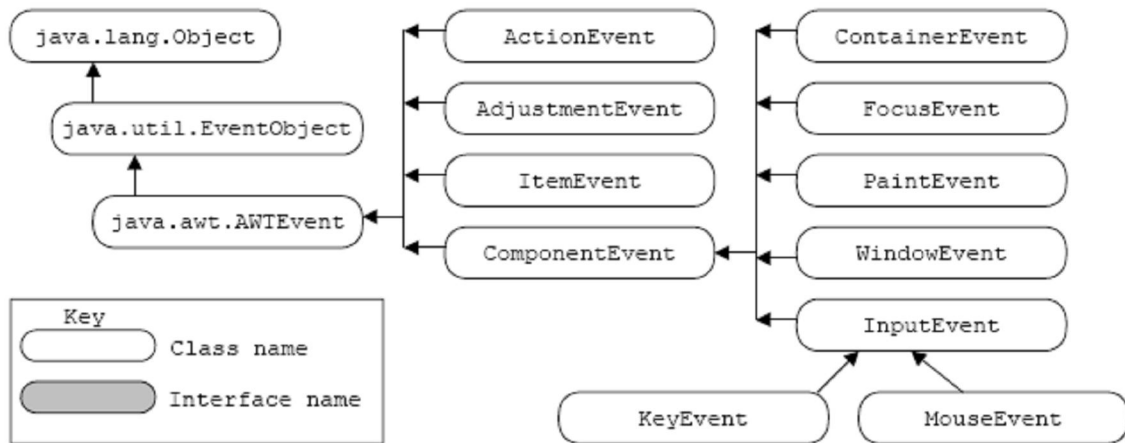


Figure 9: Ακροατές γεγονότων

Κάθε event-listener interface παρέχει μια ή περισσότερες event handler μεθόδους. Μια event handler μέθοδος καλείται αυτόματα όταν συμβαίνει κάποιο συγκεκριμένου τύπου γεγονός.

- Στην κλάση του event listener (είναι αυτονόητο ότι) θα πρέπει να υλοποιούνται όλες οι event handler μέθοδοι των event-listener interfaces που η κλάση αυτή υλοποιεί.

- Έτσι το event listener αντικείμενο της κλάσης αυτής, είναι σε θέση να ακούει συγκεκριμένου τύπου events που παράγονται από GUI components του προγράμματος - και σε περίπτωση που ένα τέτοιο event παραχθεί, εκτελείται η κατάλληλη event handler μέθοδος για αυτό το event.

5.5. Το πακέτο Java Swing

JFC είναι το ακρωνύμιο των λέξεων Java Foundation Classes το οποίο περιλαμβάνει μια ομάδα χαρακτηριστικών για την δημιουργία γραφικής διασύνδεσης χρήστη (GUI). Δύο από τα χαρακτηριστικά του JFC είναι το Swing και το Look and Feel (εμφάνιση και αίσθηση). Το Swing είναι ουσιαστικά μια επέκταση του AWT. Περιλαμβάνει ότι και το AWT (κουμπιά, ετικέτες, πεδία κειμένου κ.α) αλλά με νέα χαρακτηριστικά. Οι καινοτομίες αυτές είναι νέα συστατικά, επεκταμένα χαρακτηριστικά συστατικών και καλύτερη αίσθηση και εμφάνιση. Το Swing ενσωματώθηκε στην Java στην έκδοση 2. Τα στοιχεία του Swing είναι μέρος του πακέτου javax.swing. Όπως έγινε αναφορά πιο πάνω, όλα τα συστατικά του AWT υπάρχουν και στο Swing. Η διαφορά στον ορισμό των αντικειμένων των κλάσεων είναι ότι τα συστατικά του Swing αρχίζουν από το γράμμα J. Δηλαδή το αντίστοιχο συστατικό του Button στο AWT θα είναι στο Swing το JButton κ.ο.κ.

Άλλες σημαντικές διαφορές είναι: Τα κουμπιά και ετικέτες του Swing εκτός από κείμενο μπορούν να περιέχουν και εικονίδια. Εικονίδιο είναι ένα γραφικό μικρών διαστάσεων. Ένα ακόμα χαρακτηριστικό είναι η δυνατότητα πρόσθεσης ή αλλαγής περιθωρίων γύρω από τα περισσότερα συστατικά.

5.5.1. Γραφική διασύνδεση χρήστη με χρήση Swing

Η δημιουργία μιας γραφικής διασύνδεσης χρήστη με Swing είναι περίπου ίδια με αυτήν του AWT και δημιουργείται με τα παρακάτω βήματα:

- Σύνθεση μιας γραφικής διασύνδεσης χρήστη προσθέτοντας συστατικά (components) στους ενδιαμέσους υποδοχείς (content pane – τομέας περιεχομένων). Οι ενδιαμέσοι υποδοχείς τίθενται στους ανώτερους υποδοχείς.
- Δημιουργία χειριστών συμβάντων που ανταποκρίνονται στην αλληλεπίδραση χρήστη με το GUI.

5.5.2. Σύνθεση γραφικής διασύνδεσης με χρήση Swing

Η σύνθεση γραφικής διασύνδεσης με Swing είναι ίδια με την AWT με την διαφορά ότι υπάρχουν δύο επίπεδα υποδοχέων. Η διαδικασία σύνθεσης είναι η εξής: Αρχικά τα συστατικά προστίθενται σε ένα **ενδιάμεσο υποδοχέα** που καλείται τομέας περιεχομένων (content pane). Ο **τομέας περιεχομένων** είναι υπεύθυνος για την διάταξη των συστατικών. Οι διαχειριστές διάταξης ορίζονται σε αυτόν. Για τους διαχειριστές διάταξης ισχύουν οι έννοιες που εξετάζονται στο κεφάλαιο .Τέλος οι ενδιαμέσοι υποδοχείς προστίθενται στους κύριους (ανώτερους) υποδοχείς. Οι ανώτεροι υποδοχείς είναι οι κλάσεις JFrame, JApplet, JOptionPane. Οι ενδιαμέσοι υποδοχείς είναι οι JPanel, JScrollPane, JSplitPane, JTabbedPane, JToolBar. Οι ενδιαμέσοι υποδοχείς προστίθενται στους ανώτερους υποδοχείς με την μέθοδο setContentPane(...). Οι διαχειριστές διάταξης ορίζονται στους ενδιαμέσους υποδοχείς με την μέθοδο setLayout(...). Τα συστατικά προστίθενται στους ενδιαμέσους υποδοχείς με την μέθοδο add(...). Οι ενδιαμέσοι υποδοχείς μπορούν να προστεθούν σε άλλους ενδιαμέσους υποδοχείς. Οι ανώτεροι υποδοχείς έχουν προεπιλεγμένους τομείς περιεχομένων, δηλαδή σε ένα πρόγραμμα μπορεί να δείτε την εντολή getContentPane() (δηλαδή επιστροφή του τομέα περιεχομένων) χωρίς να έχει οριστεί ενδιαμέσος υποδοχέας.

5.5.3. Top-Level Swing Containers και Swing Components

Κάθε γραφική διεπαφή τύπου Swing πρέπει να έχει τουλάχιστον ένα top-level Swing container. Ένα top-level Swing container παρέχει την απαραίτητη υποστήριξη που χρειάζονται τα Swing components για την εμφάνισή τους και την διαχείριση των γεγονότων που αυτά παράγουν. Υπάρχουν τρία top-level Swing containers: το JFrame, το JDialog, και (για applets) το JApplet. Κάθε JFrame αντικείμενο δημιουργεί ένα κύριο γραφικό παράθυρο, κάθε JDialog αντικείμενο δημιουργεί ένα δευτερεύον παράθυρο (δηλ παράθυρο που εξαρτάται από κάποιο άλλο παράθυρο). Κάθε JApplet αντικείμενο δημιουργεί την περιοχή εμφάνισης ενός applet στο παράθυρο του Web Browser. Σε αυτό το κείμενο θα ασχοληθούμε κατά κύριο λόγο με το συστατικό JFrame .Η ιεραρχία του top-level Swing containers JFrame δίνεται παρακάτω:

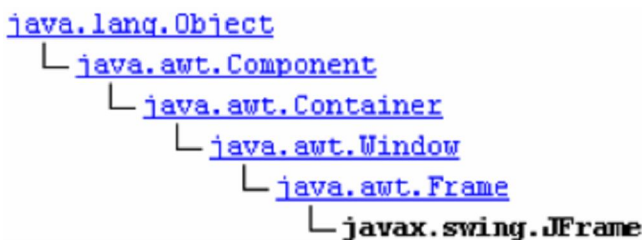


Figure 10: Ιεραρχία του top-level Swing containers JFrame

Με εξαίρεση τα top-level containers, πχ το JFrame, όλα τα Swing components είναι υποκλάσεις της κλάσης JComponent. Η ιεραρχία της JComponent φαίνεται στο παρακάτω σχήμα:

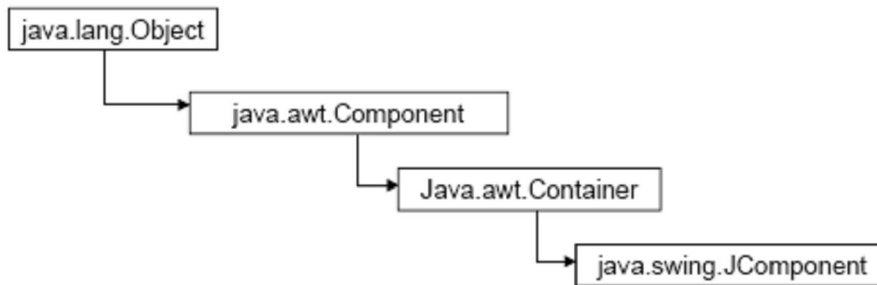


Figure 11: Ιεραρχία της κλάσης JComponent

5.5.4. Διαχειριστές Γεγονότων - Event Handlers

Τα Swing components παράγουν διάφορους τύπους γεγονότων (events). Ο παρακάτω πίνακας περιέχει τους διαχειριστές γεγονότων που χρησιμοποιήθηκαν στην εφαρμογή που δημιουργήθηκε.

ΜΕΡΙΚΑ ΓΕΓΟΝΟΤΑ ΚΑΙ ΟΙ ΣΧΕΤΙΚΟΙ ΑΚΡΟΑΤΕΣ ΓΕΓΟΝΟΤΟΣ ΤΟΥΣ	
Λόγος που οδηγεί στο γεγονός	Τύπος ακροατή
Ο χρήστης πατάει ένα κουμπί ,πατάει enter όταν γράφει κείμενο σε ένα text area ,η επιλέγει ένα αντικείμενο μενού	ActionListener
Ο χρήστης κλείνει ένα frame(κύριο παράθυρο)	WindowListener
Ο χρήστης πατάει ένα κουμπί του mouse όταν ο κέρσορας βρίσκεται πάνω σε ένα συστατικό	MouseListener
Ο χρήστης περιστρέφει το mouse γύρω από ένα συστατικό	MouseMotionListener
Το συστατικό γίνεται ορατό(visible)	ComponentListener
Το συστατικό παίρνει την εστίαση του πληκτρολογίου	FocusListener
Σε πίνακα η σε λίστα η επιλογή άλλαξε	ListSelectionListener
Οποιαδήποτε ιδιοκτησία σε ένα συστατικό αλλάζει ,όπως το κείμενο σε μια ετικέτα	PropertyChangeListener

Table 2: Γεγονότα και οι σχετικοί ακροατές γεγονότος τους

5.6. Τα κυριότερα συστατικά του πακέτου Swing που χρησιμοποιήθηκαν στην υλοποίηση

Σε αυτή την ενότητα θα μιλήσουμε για τα κύρια συστατικά της Java που χρησιμοποιήθηκαν στην υλοποίηση της εφαρμογής που δημιουργήθηκε. Αρχικά θα αναλυθούν οι ενδιάμεσοι υποδοχείς (intermediate containers) που χρησιμοποιήθηκαν και στην συνέχεια οι κύριοι υποδοχείς (top-level container).

5.6.1. Το συστατικό Jtree (δέντρο) του Swing

Σε ένα δέντρο τα δεδομένα εμφανίζονται ιεραρχικά. Ένα δέντρο δημιουργείται από την κλάση JTree. Ένα αντικείμενο της JTree δεν περιέχει πραγματικά τα δεδομένα, αλλά περιέχει μια άποψη από αυτά. Ο προσανατολισμός του συστατικού JTree (δέντρο) συνήθως είναι κάθετος. Κάθε σειρά από αυτό το συστατικό περιέχει ένα στοιχείο δεδομένων που καλείται κόμβος. Κάθε δέντρο έχει μια ρίζα κόμβου από όπου όλοι οι απόγονοι κόμβοι εξαρτώνται. Προεπιλεγμένα, τα δέντρα εμφανίζουν την ρίζα κόμβου. Οι κόμβοι που εξαρτώνται από έναν άλλο κόμβο είναι παιδιά του κόμβου αυτού ο οποίος λέγεται κλάδος (branch node), ή πατέρας τους. Οι κόμβοι που δεν έχουν απογόνους λέγονται φύλλα (leaf). Η λειτουργία ενός αντικειμένου της κλάσης JTree είναι απλή. Κάνοντας διπλό κλικ σε ένα κόμβο εμφανίζονται οι επόμενοι κόμβοι της ιεραρχίας. Αν γίνει πάλι διπλό κλικ στον ίδιο κόμβο οι απόγονοι του κόμβου σταματούν να είναι ορατοί.

Σε ένα αντικείμενο της κλάσης JTree η ρίζα του και οι υπόλοιποι κόμβοι προστίθενται με την κλάση DefaultMutableTreeNode. Οι κόμβοι προστίθενται σε άλλους κόμβους με την μέθοδο add(...). Για την ρίζα, τους κλάδους και τα φύλλα υπάρχουν προεπιλεγμένα εικονίδια δίπλα στους κόμβους. Κάθε φορά που γίνεται διπλό κλικ σε ένα κλάδο αυτός αλλάζει εικονίδιο ανάλογα αν κρύβει ή εμφανίζει τους απογόνους του. Η κλάση που αλλάζει τα προεπιλεγμένα εικονίδια είναι η DefaultTreeCellRenderer.

Οι μέθοδοι της κλάσης αυτής είναι οι setLeafIcon(Icon i), setOpenIcon(Icon i), setClosedIcon(Icon i) που θέτουν τα εικονίδια σε ένα φύλλο, σε έναν κλάδο που εκτείνεται και σε έναν κλάδο που συρρικνώνεται αντίστοιχα. Η μέθοδος που ενεργοποιεί την αλλαγή εικονιδίων στο δέντρο είναι η setCellRender(DefaultTreeCellRenderer d).

5.6.2. Το συστατικό JEditorPane του Swing

Το JEditorPane είναι ένα συστατικό του Swing το οποίο μπορεί να αναπαραστήσει κείμενο μορφοποιημένο με την γλώσσα μορφοποίησης HTML η RTF (Rich Text Format).

5.6.3. Ο ενδιάμεσος υποδοχέας JPanel (Πάνελ) του Swing

Η κλάση JPanel είναι ο πιο ευέλικτος και ο πιο συχνά χρησιμοποιούμενος ενδιάμεσος υποδοχέας. Η JPanel μπορεί να χρησιμοποιήσει οποιονδήποτε διαχειριστή διάταξης. Προεπιλεγμένα, η JPanel δεν χρωματίζει τίποτα εκτός από το παρασκήνιο (background) και έχει σαν διαχειριστή διάταξης την διάταξη ροής (κλάση FlowLayout).

Οι βασικοί δημιουργοί της κλάσης JPanel είναι:

- JPanel(): Δημιουργεί ένα αντικείμενο της κλάσης JPanel.

- **JPanel(LayoutManager layout):** Δημιουργεί ένα αντικείμενο της κλάσης JPanel με διαχειριστή διάταξης το αντικείμενο layout. Τα συστατικά προστίθενται στον ενδιαμέσο υποδοχέα με την μέθοδο add(...). Μέσα στην μέθοδο αυτή μπορεί να προσδιοριστεί και ο προσανατολισμός των συστατικών.

5.6.4. Ο ενδιάμεσος υποδοχέας JScrollPane (Πάνελ κύλισης) του Swing

Ο ενδιάμεσος υποδοχέας κύλισης υλοποιείται με την κλάση JScrollPane. Χρησιμοποιείται όταν ένα συστατικό που θέλουμε να βάλουμε σε ένα πρόγραμμα καταλαμβάνει περισσότερη από την προσφερόμενη επιφάνεια. Ένα αντικείμενο της κλάσης JScrollPane έχει μπάρες κύλισης στις άκρες του και έτσι είναι δυνατή η εμφάνιση όλων των σημείων του συστατικού που περικλείει. Βασικά η JScrollPane αποτελείται από τα στοιχεία δύο κλάσεων της JScrollPane και της JViewport. Η κλάση JViewport είναι υπεύθυνη για το τι εμφανίζεται στην οθόνη όταν μετακινούνται οι λίστες κύλισης. Οι δημιουργοί της κλάσης JScrollPane είναι:

- **JScrollPane():** Δημιουργεί ένα άδειο αντικείμενο της κλάσης JScrollPane με τις κυλιόμενες μπάρες να εμφανίζονται όταν χρειάζονται.
- **JScrollPane(int x, int y):** Δημιουργεί ένα άδειο αντικείμενο της κλάσης JScrollPane όπου με x, y ορίζεται η κατάσταση των μπαρών. Το όρισμα x αναφέρεται στην οριζόντια μπάρα και μπορεί να πάρει τις τιμές HORIZONTAL_SCROLLBAR_ALWAYS, HORIZONTAL_SCROLLBAR_NEVER, HORIZONTAL_SCROLLBAR_AS_NEEDED που αντιστοιχούν στην εμφάνιση της μπάρας, πάντα, όταν χρειάζεται και ποτέ αντίστοιχα. Τα ίδια ισχύουν και για το όρισμα y που ισχύει για τον κατακόρυφο άξονα, στην θέση όμως της λέξης HORIZONTAL μπαίνει η VERTICAL.

5.6.5. Ο κύριος υποδοχέας JFrame (Πλαίσιο) του Swing

Τα πλαίσια στο Swing υλοποιούνται με την κλάση JFrame. Ένα αντικείμενο της JFrame έχει περιθώρια, τίτλο και κουμπιά για το κλείσιμο, την ελαχιστοποίηση και μεγιστοποίηση του παραθύρου. Δημιουργοί της κλάσης JFrame είναι:

- **JFrame():** Δημιουργεί ένα πλαίσιο το οποίο είναι αόρατο.
 - **JFrame(String str):** Δημιουργεί ένα αόρατο πλαίσιο με τίτλο str.
- Η μέθοδος setVisible(boolean bl) είναι υπεύθυνη για την εμφάνιση ή μη ενός πλαισίου. Η setSize(int x, int y) θέτει σε x, y pixels το μέγεθος του πλαισίου. Τέλος, η μέθοδος setTitle(String str) θέτει σε str τον τίτλο ενός πλαισίου.

6. Ανάλυση Λογισμικού

6.1. Εισαγωγή

Η ανάλυση λογισμικού είναι ένα δομημένο σύνολο δραστηριοτήτων που απαιτούνται για την ανάπτυξη ενός συστήματος λογισμικού. Οι δραστηριότητες αυτές αφορούν την: Ανάλυση (Specification), Σχεδίαση (Design), Τεκμηρίωση (Validation) και Εξέλιξη (Evolution) του συστήματος. Με την ανάλυση προσδιορίζονται όλα όσα πρέπει να υλοποιεί το σύστημα (απαιτήσεις συστήματος) καθώς και εντοπίζονται οι περιορισμοί του, με την σχεδίαση υλοποιείται το σύστημα, με την τεκμηρίωση ελέγχεται αν το λογισμικό ανταπεξέρχεται στις

αρχικές απαιτήσεις και τέλος στη φάση της εξέλιξης το σύστημα εξελίσσεται για να καλύψει τυχόν μελλοντικές ανάγκες.

Για τη διεκπεραίωση των παραπάνω δραστηριοτήτων έχουν αναπτυχθεί διάφορα μοντέλα διαδικασίας λογισμικού, τα οποία είναι απλοποιημένες αναπαραστάσεις της διαδικασίας μέσα από συγκεκριμένη οπτική γωνία. Τέτοια μοντέλα είναι το Waterfall (σειριακό μοντέλο που προαπαιτεί πλήρως καταγεγραμμένες απαιτήσεις), το Evolutionary (εξελικτικό μοντέλο με πολλές ενδιάμεσες εκδόσεις) και το Reuse-based (μοντέλο που βασίζεται στην επαναχρησιμοποίηση υπάρχοντων στοιχείων τα οποία ενοποιούνται για τη δημιουργία του νέου συστήματος).

Σε κάθε περίπτωση η επιλογή του μοντέλου εξαρτάται από τις απαιτήσεις του συστήματος καθώς και από τη διάρκεια και το μέγεθος του έργου.

6.2. Μεθοδολογία ανάπτυξης

Για την ανάπτυξη της εφαρμογής θα χρησιμοποιηθούν αντικειμενοστραφείς τεχνολογίες, οι οποίες είναι και οι πιο διαδεδομένες τα τελευταία χρόνια. Υπάρχουν διάφορες μεθοδολογίες και προσεγγίσεις για τον κύκλο ζωής μιας τέτοιας εφαρμογής. Μία μεθοδολογία ορίζει την διαδικασία ανάπτυξης του λογισμικού, στην οποία συμπεριλαμβάνονται τα βήματα, οι πρακτικές και οι τεχνικές που πρέπει να ακολουθηθούν. Στο μοντέλο του καταρράκτη (waterfall model) όλη η ανάλυση και ο σχεδιασμός γίνεται πριν την υλοποίηση (συγγραφή του κώδικα). Σε μία επαναληπτική διαδικασία ανάπτυξης (π.χ. Extreme Programming) αναπτύσσεται ένα μικρό υποσύνολο του συστήματος, το οποίο ενισχύεται και βελτιώνεται επαναληπτικά μέχρι την πλήρη υλοποίηση του συστήματος. Σε κάθε επανάληψη γίνονται οι αναγκαίες σχεδιαστικές τροποποιήσεις. Στην Ενοποιημένη Διαδικασία Rational (Rational Unified Process) πρώτα γίνεται η ανάλυση απαιτήσεων και οι δραστηριότητες ανάλυσης και σχεδιασμού για ολόκληρο το σύστημα και στη συνέχεια το σύστημα κατασκευάζεται με επαναληπτικό τρόπο. Αν και δεν χρησιμοποιήθηκε καμία από τις παραπάνω μεθοδολογίες στην ολότητά τους, ωστόσο ακολουθήθηκαν ορισμένες από τις πρακτικές που αυτές προτείνουν. Οι κυριότερες από αυτές είναι:

- **Διαγράμματα UML**- Τα διαγράμματα UML χρησιμοποιούνται σε όλες τις φάσεις του κύκλου ζωής του λογισμικού, από την ανάλυση και τον σχεδιασμό μέχρι την τεκμηρίωση.
- **Επαναληπτική ανάπτυξη (iterative development)** – Επιτρέπει τη διασαφήνιση των απαιτήσεων και την βελτίωση του σχεδιασμού, καθώς και τη γρήγορη ανάδραση από τους χρήστες.

6.2.1. Καταγραφή Απαιτήσεων

Η μελέτη των απαιτήσεων ενός νέου συστήματος λογισμικού αποτελεί ιδιαίτερα σημαντικό τμήμα της συνολικής φάσης ανάπτυξης του συστήματος και έχει μελετηθεί από πολλές και διαφορετικές μεθοδολογικές προσεγγίσεις. Συνήθως οι απαιτήσεις λογισμικού διαχωρίζονται σε δύο κατηγορίες – αυτές που αφορούν τα λειτουργικά χαρακτηριστικά του συστήματος (λειτουργικές απαιτήσεις) και αυτές που συνιστούν ‘οριζόντια’ ποιοτικά χαρακτηριστικά (quality attributes) και που αναφέρονται ως μη λειτουργικές απαιτήσεις. Οι λειτουργικές απαιτήσεις καταγράφονται, αναλύονται και ενσωματώνονται σε διαδικασίες που συνιστούν το σύστημα λογισμικού. Υπάρχουν πληθώρα τεχνικών τόσο για τη καταγραφή όσο

και για την ανάλυση και επικύρωση των λειτουργικών απαιτήσεων ενός συστήματος λογισμικού.

Σε αντίθεση με τις λειτουργικές απαιτήσεις λογισμικού, οι μη-λειτουργικές απαιτήσεις είναι λιγότερο γνωστές και περισσότερο σύνθετες στη μελέτη τους. Παραδείγματα μη-λειτουργικών απαιτήσεων για ένα σύστημα λογισμικού μπορεί να αφορούν μεγέθη όπως η χρηστικότητα του συστήματος, η μεταφερσιμότητά του από μια πλατφόρμα σε μια άλλη η δυνατότητα προσαρμογής του συστήματος, η ασφάλεια και η υποστήριξη χρηστών μετά την παράδοση.

Από τα παραπάνω προκύπτει ότι εξαιτίας των χαρακτηριστικών των μη-λειτουργικών απαιτήσεων (π.χ. αφηρημένες έννοιες, οριζόντια εφαρμογή, κλπ) η μελέτη και ανάλυσή τους δεν είναι ούτε εύκολη ούτε συγκρίσιμη με τη μελέτη και ανάλυση των λειτουργικών απαιτήσεων ενός συστήματος λογισμικού. Η δυσκολία συνίσταται σε διάφορους λόγους. Πρώτα από όλα, οι απαιτήσεις της κατηγορίας αυτής δεν προσδιορίζονται και δεν επικυρώνονται εύκολα. Δεύτερον, δεν υπάρχει ένας ενιαίος και πλήρης κατάλογος μη-λειτουργικών απαιτήσεων που να μπορούν να εξειδικευτούν ανάλογα με το εκάστοτε πρόβλημα. Τρίτον, στην πλειοψηφία των περιπτώσεων οι μη-λειτουργικές απαιτήσεις είναι αντιμαχόμενες η μία την άλλη με αποτέλεσμα να μην καθίσταται εφικτή η προοδευτική ενσωμάτωση και υποστήριξη τους σε ένα σύστημα κάτι που ισχύει για τις λειτουργικές απαιτήσεις. Τέλος, οι μη-λειτουργικές απαιτήσεις αφορούν κυρίως κατασκευαστές συνιστωσών πλατφόρμας (platform developers) με αποτέλεσμα συχνά να μην εξετάζονται κατά το σχεδιασμό ενός συστήματος λογισμικού.

6.2.2. Use Cases.

Περίπτωση χρήσης είναι η περιγραφή ενός σεναρίου ή ενός συνόλου σεναρίων τα οποία είναι στενά συνδεδεμένα μεταξύ τους. Μια περίπτωση χρήσης ορίζει και περιγράφει την αλληλεπίδραση ανάμεσα στους δράστες (actors) και συγκεκριμένα λειτουργικά τμήματα της εφαρμογής (υπηρεσίες του συστήματος). Δηλαδή μια περίπτωση χρήσης περιγράφει σε υψηλό επίπεδο, συστηματικά και μεθοδικά πώς μια εφαρμογή θα χρησιμοποιηθεί για την υλοποίηση συγκεκριμένων στόχων. Πολλές περιπτώσεις χρήσεις καλύπτουν τελικά όλες τις απαιτήσεις και την επιθυμητή συμπεριφορά της εφαρμογής. Με τις περιπτώσεις χρήσης περιγράφουμε την συμπεριφορά του συστήματος χωρίς να μπαίνουμε σε λεπτομέρειες όπως τρόποι υλοποίησης.

Μια περίπτωση χρήσης συνήθως συμπεριλαμβάνει πολλά σενάρια και δίνει την δυνατότητα αποφυγής περιττών πληροφοριών και περιγραφών των λειτουργιών ενός συστήματος. Αυτό το οποίο προσπαθούμε να επιτύχουμε με τις περιπτώσεις χρήσης είναι μια ολοκληρωμένη άποψη των λειτουργιών και της συμπεριφοράς του λογισμικού του συστήματος.

Οι περιπτώσεις χρήσης είναι δομημένες με συγκεκριμένο και τυπικό τρόπο. Αυτό γίνεται μέσω δομημένης περιγραφής. Τα συστατικά αυτής της δομής είναι:

- Όνομα / σύντομη περιγραφή
- Χαρακτήρες (actors): που επικοινωνούν με την περίπτωση χρήσης
- Προϋποθέσεις (Ποιές συνθήκες πρέπει να ικανοποιούνται στην έναρξη της περίπτωσης χρήσης)
- Μέτα-Συνθήκη (Ποιές συνθήκες πρέπει να ικανοποιούνται στην λήξη της περίπτωσης χρήσης)
- Ερέθισμα (Trigger): ένα συμβάν που προκαλεί την περίπτωση χρήσης
- Σχέσεις: Συσχέτιση (association), Περιλαμβάνει (include), Επεκτείνει (extend), Εξειδικεύει (generalization) :
- Ομαλή Ροή Γεγονότων (Περιγραφή της συμπεριφοράς βάση της περίπτωσης χρήσης)

- Συνιστώσες ροές γεγονότων
- Εναλλακτικές ροές – Εξαιρέσεις: αποκλίσεις από το κύριο σενάριο επιτυχίας

6.2.3. Σενάρια χρήσης (Scenarios of Use).

Τα σενάρια αποτελούν μια μικρογραφία των επιμέρους λειτουργιών ενός συστήματος και ανάλογα με την πιστότητά τους μπορεί να είναι είτε γενικά σενάρια που εμπεριέχουν επιλεγμένες περιπτώσεις χρήσης είτε σενάρια αλληλεπίδρασης που καταγράφουν τρόπους εκτέλεσης μιας περίπτωσης χρήσης. Η χρήση των σεναρίων κατά την διαδικασία ανάλυσης ενός συστήματος λογισμικού βασίζεται στην ιδιότητα που έχουν τα σενάρια να προσφέρουν αφενός λεπτομερείς καταγραφές πρακτικών (δηλαδή το τι κάνουν οι χρήστες ή το ίδιο το σύστημα) και ταυτόχρονα να συνεισφέρουν στον οραματισμό νέων λύσεων με τη χρήση προηγμένων τεχνολογικών εργαλείων. Συγκεκριμένα, ένα σενάριο προσφέρεται τόσο για την αποκάλυψη καθηκόντων που εκτελούν οι χρήστες και που συνιστούν υπάρχουσες πρακτικές όσο και για τον οραματισμό λύσεων που γίνονται εφικτές με τη χρήση νέων τεχνολογιών και εργαλείων.

Τα σενάρια χρήσης θα πρέπει να ικανοποιούν συγκεκριμένα ποιοτικά χαρακτηριστικά που καθορίζουν την πληρότητα του σεναρίου.

- Ένα σενάριο για να είναι πλήρες θα πρέπει να καταγράφεται με μια μορφή που να επιτρέπει περαιτέρω ανάλυση.
- Το σενάριο θα πρέπει να προσδιορίζει το στόχο ή το σκοπό που εξυπηρετείται και να γίνεται σαφής αναφορά σε εμπλεκόμενους φορείς (συνιστώσες συστήματος και χρήστες).
- Το σενάριο θα πρέπει να συνθέτεται από επιμέρους δραστηριότητες (βήματα) που όταν εκτελεστούν να αποφέρουν το επιθυμητό αποτέλεσμα.
- Το σενάριο θα πρέπει να προσδιορίζει την ροή των δραστηριοτήτων (βημάτων) που πρέπει να εκτελεστούν
- Για κάθε επιμέρους δραστηριότητα ή βήμα θα πρέπει να υπάρχουν ή να μπορούν να προσδιοριστούν όρια που εξασφαλίζουν την έναρξη και λήξη της δραστηριότητας και του σεναρίου γενικότερα.
- Ένα σενάριο θα πρέπει να προσδιορίζει τις προϋποθέσεις που το ολοκληρώνουν και τυχόν εξαιρέσεις που συνήθως συνιστούν εναλλακτικά σενάρια.

Αναφορικά με τις τεχνικές καταγραφής, ένα σενάριο ανάλογα με το βαθμό ωριμότητάς του μπορεί να καταγράφεται με πολλούς διαφορετικούς τρόπους, όπως:

- κείμενο / αφήγηση γεγονότων,
- αριθμημένη ακολουθία βημάτων,
- φόρμα δομημένης περιγραφής
- διαχωρισμένη αφήγηση,
- μοντέλο περιπτώσεων χρήσης,
- διάγραμμα σεναρίου VORD
- μοντέλο ακολουθίας της UML, κλπ.

Επίσης, είναι δυνατόν ένα σενάριο αρχικά καταγεγραμμένο με ένα γενικό τύπο (π.χ. κείμενο) να εξειδικευτεί με τη χρήση μιας άλλης μορφής (π.χ. διαχωρισμένη ακολουθία βημάτων) προκειμένου να γίνει δυνατή η καλύτερη κατανόηση επιμέρους συστατικών του.

6.3. Γενικό σενάριο χρήσης της εφαρμογής

Ας υποθέσουμε ότι ένας χρήστης του προγράμματος αναπαράστασης μορίων γραφικά Jmol χρησιμοποιήσει την εφαρμογή που δημιουργήθηκε στο πλαίσιο της παρούσας πτυχιακής και η οποία είναι ενσωματωμένη στο πρόγραμμα Jmol και επιλέξει κάποια λειτουργία της εφαρμογής μας. Υποθέτοντας ότι ο χρήστης θα θελήσει να μελετήσει με περισσότερη λεπτομέρεια την δομή του μορίου που αναπαρίσταται από το Jmol. Αρχικά ο χρήστης μπορεί να επιλέξει το κομμάτι της εφαρμογής που ονομάζεται Sequence View, με αυτό το εργαλείο ο χρήστης μπορεί να πληροφορηθεί την ακολουθία αμινοξέων που χαρακτηρίζει το βιολογικό μόριο ή την πρωτεΐνη που αναπαρίσταται την δεδομένη στιγμή από το Jmol και επιλέγοντας ένα συγκεκριμένο αμινοξύ την φορά να διαπιστώσει τα σημεία που βρίσκονται τα αμινοξέα κάθε είδους πάνω στην βιολογική δομή, επισημαίνοντας η εφαρμογή ένα ένα τα μόρια που αποτελούν τα συγκεκριμένα αμινοξέα κάθε φορά.

Στην συνέχεια ο χρήστης ανοίγοντας το κομμάτι της εφαρμογής Jtree θα μπορέσει να διακρίνει τα δομικά συστατικά, δηλ. της πολυπεπτιδικές αλυσίδες, τα αμινοξέα και τέλος τα μόρια που αποτελούν και το μικρότερο συστατικό στην αναπαράσταση ενός μορίου, όλα τα παραπάνω συστατικά ενός μορίου βρίσκονται κατηγοριοποιημένα στην εφαρμογή Jtree ανάλογα το μεγαλύτερο συστατικό στο οποίο ανήκουν. Επιλέγοντας ο χρήστης με ένα απλό κλικ κάθε άτομο που δηλώνεται με το όνομα του θα μπορεί να δει την οπτική επισήμανση του επιλεγμένου ατόμου στην σκηνή του Jmol. Στο κάθε επιλεγμένο άτομο ο χρήστης θα μπορεί ενεργοποιώντας την εφαρμογή Comment mode να εισάγει τα δικά του σχόλια και να τα αποθηκεύσει.

6.3.1. Καταγραφή απαιτήσεων της εφαρμογής

Όπως προαναφέρθηκε, η καταγραφή των απαιτήσεων είναι ένα ιδιαίτερα σημαντικό κομμάτι στη διαδικασία της ανάπτυξης ενός συστήματος. Παρακάτω παρουσιάζουμε τις απαιτήσεις της εφαρμογής που δημιουργήθηκε.

Απαιτήσεις χρήσης της εφαρμογής

Προσδιορισμός / Τίτλος Απαίτησης

Πρόσβαση και λειτουργία της εφαρμογής

Περιγραφή Απαίτησης

- Εισαγωγή στο πρόγραμμα Jmol
- Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
- Πρόσβαση στην εφαρμογή επιλέγοντας το μενού Visual Tools
- Επιλογή περιβάλλοντος της εφαρμογής
- Λειτουργία του περιβάλλοντος που επιλέχθηκε

Απαιτήσεις λειτουργίας της εφαρμογής

Προσδιορισμός / Τίτλος Απαίτησης

Προβολή ακολουθίας αμινοξέων

Περιγραφή Απαίτησης

- Εισαγωγή στο πρόγραμμα Jmol
- Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
- Εισαγωγή στην εφαρμογή Sequence View
- Επισκόπηση ακολουθίας αμινοξέων

Προσδιορισμός / Τίτλος Απαίτησης

Προβολή των βιολογικών δομικών συστατικών ενός μορίου

Περιγραφή Απαίτησης

- Εισαγωγή στο πρόγραμμα Jmol
- Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
- Εισαγωγή στην εφαρμογή Jtree
- Επισκόπηση των δομικών συστατικών ενός μορίου

Προσδιορισμός / Τίτλος Απαίτησης

Ενεργοποίηση της λειτουργίας εισαγωγής σχολίου

Περιγραφή Απαίτησης

- Εισαγωγή στο πρόγραμμα Jmol
- Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
- Εισαγωγή στην λειτουργία addComment

Απαιτήσεις λειτουργιών της εφαρμογής

Προσδιορισμός / Τίτλος Απαίτησης

Επισημάνση των αμινοξέων πάνω στο μόριο που αναπαρίσταται στην σκηνή του Jmol

Περιγραφή Απαίτησης

- Εισαγωγή στο πρόγραμμα Jmol
- Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
- Εισαγωγή στην εφαρμογή Sequence View
- Επιλογή αμινοξέου από την αμινοξική ακολουθία

Προσδιορισμός / Τίτλος Απαίτησης

Επίσημανση των βιολογικών δομικών συστατικών πάνω στο μόριο που αναπαρίσταται στην σκηνή του Jmol

Περιγραφή Απαίτησης

- Εισαγωγή στο πρόγραμμα Jmol
- Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
- Εισαγωγή στην εφαρμογή TreeView
- Επιλογή της βιολογικής δομής από την εφαρμογή του δέντρου

Προσδιορισμός / Τίτλος Απαίτησης

Αύξηση μεγέθους των επιλεγμένων ατόμων του μορίου που αναπαρίσταται στην σκηνή του Jmol

Περιγραφή Απαίτησης

- Εισαγωγή στο πρόγραμμα Jmol
- Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
- Εισαγωγή στην εφαρμογή Sequence View
- Επιλογή αμινοξέου από την αμινοξική ακολουθία
- Επιλογή από την πτυσσόμενη λίστα της επιλογής max atom

Προσδιορισμός / Τίτλος Απαίτησης

Κοινοποίηση πληροφορίας σχετική με τα χαρακτηριστικά χρώματα των αμινοξέων στο χρήστη

Περιγραφή Απαίτησης

- Εισαγωγή στο πρόγραμμα Jmol
- Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
- Εισαγωγή στην εφαρμογή Sequence View
- Άνοιγμα του παραθυρικού περιβάλλοντος amino acid color
- Επισκόπηση πληροφορίας

Προσδιορισμός / Τίτλος Απαίτησης

Εισαγωγή σχολίου

Περιγραφή Απαίτησης

- Εισαγωγή στο πρόγραμμα Jmol
- Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
- Εισαγωγή στην λειτουργία addComment
- Επιλογή του ατόμου για την εισαγωγή του σχολίου
- Κλικ πάνω στο γραφικό εισαγωγής σχολίου
- Εισαγωγή σχολίου στο text field
- Κλικ στο κουμπί ok

Προσδιορισμός / Τίτλος Απαίτησης

Αφαίρεση σχολίου

Περιγραφή Απαίτησης

- Εισαγωγή στην εφαρμογή Comment mode
- Επιλογή του ατόμου για την αφαίρεση του ήδη υπάρχοντος σχολίου

Προσδιορισμός / Τίτλος Απαίτησης

Λειτουργία αποθήκευσης

Περιγραφή Απαίτησης

- Επιλογή της λειτουργίας αποθήκευσης
- Επιλογή της τοποθεσίας που θέλουμε να αποθηκευτεί το αρχείο
- Όνομα αρχείου αποθήκευσης
- Επιλογή επικύρωσης αποθήκευσης

6.3.2. Περιπτώσεις χρήσης και σενάρια της εφαρμογής

Σε αυτή την ενότητα θα αναλύσουμε τις περιπτώσεις χρήσης που προκύπτουν από την παραπάνω περιγραφή του γενικού σεναρίου έτσι ώστε να προσδιορίσουμε τα σενάρια χρήσης της εφαρμογής και να καταγράψουμε όλες τις περιπτώσεις αλληλεπίδρασης των χρηστών της εφαρμογής που δημιουργήσαμε με το σύστημα. Στο πίνακα παρακάτω βλέπουμε όλα τα σενάρια ανά περίπτωση χρήσης.

Συγκεκριμένα, για την περίπτωση χρήσης «Απαιτήσεις χρήσης της εφαρμογής» έχουμε το εξής σενάριο:

- Πρόσβαση και λειτουργία της εφαρμογής που δημιουργήσαμε

Για την περίπτωση χρήσης «Απαιτήσεις λειτουργίας της εφαρμογής» έχουμε τα εξής σενάρια:

- Προβολή ακολουθίας αμινοξέων
- Προβολή των δομικών συστατικών ενός μορίου
- Ενεργοποίηση της λειτουργίας εισαγωγής σχολίου

Για την περίπτωση χρήσης «Απαιτήσεις λειτουργιών της εφαρμογής» έχουμε τα εξής σενάρια:

- Επισήμανση των αμινοξέων πάνω στο μόριο που αναπαρίσταται στην σκηνή του Jmol
- Επισήμανση των βιολογικών δομικών συστατικών πάνω στο μόριο που αναπαρίσταται στην σκηνή του Jmol
- Αύξηση μεγέθους των επιλεγμένων ατόμων του μορίου που αναπαρίσταται στην σκηνή του Jmol
- Κοινοποίηση πληροφορίας σχετική με τα χαρακτηριστικά χρώματα των αμινοξέων στο χρήστη
- Εισαγωγή σχολίου
- Αφαίρεση σχολίου
- Λειτουργία αποθήκευσης

Περίπτωση χρήσης	Σενάρια
Απαιτήσεις χρήσης της εφαρμογής	<ul style="list-style-type: none"> • Πρόσβαση και λειτουργία της εφαρμογής που δημιουργήσαμε
Απαιτήσεις λειτουργίας της εφαρμογής	<ul style="list-style-type: none"> • Προβολή ακολουθίας αμινοξέων • Προβολή των βιολογικών δομικών συστατικών ενός μορίου • Επιλογή ατόμου για την εισαγωγή σχολίου
Απαιτήσεις λειτουργιών της εφαρμογής	<ul style="list-style-type: none"> • Επισήμανση των αμινοξέων πάνω στο μόριο που αναπαρίσταται στην σκηνή του Jmol • Αύξηση μεγέθους των επιλεγμένων ατόμων του μορίου που αναπαρίσταται στην σκηνή του Jmol • Κοινοποίηση πληροφορίας σχετική με τα χαρακτηριστικά χρώματα των αμινοξέων στο χρήστη • Εισαγωγή σχολίου • Αφαίρεση σχολίου • Λειτουργία αποθήκευσης

Table 3: Λίστα σεναρίων ανά περίπτωση χρήσης

6.3.3. Λίστα σεναρίων περίπτωσης χρήσης: Απαιτήσεις χρήσης της εφαρμογής

Σε περαιτέρω ανάλυση ,βρίσκουμε τις δράσεις για κάθε σενάριο της περίπτωσης χρήσης «Απαιτήσεις χρήσης της εφαρμογής» ξεχωριστά , όπως φαίνεται στον παρακάτω πίνακα .

User: Απλός χρήστης ,εξειδικευμένος χρήστης (βιολόγος ,χημικός)	
Περίπτωση χρήσης: Απαιτήσεις χρήσης της εφαρμογής	
Σενάρια	Δράσεις
Πρόσβαση και λειτουργία της εφαρμογής	Εισαγωγή στο πρόγραμμα Jmol
	Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
	Πρόσβαση στην εφαρμογή επιλέγοντας το μενού Visual Tools
	Επιλογή περιβάλλοντος της εφαρμογής
	Λειτουργία του περιβάλλοντος που επιλέχθηκε

Table 4: Δράσεις σεναρίων περίπτωσης χρήσης «Απαιτήσεις χρήσης της εφαρμογής»

Περίπτωση χρήσης: Απαιτήσεις χρήσης της εφαρμογής

Σενάριο: Πρόσβαση και λειτουργία της εφαρμογής

Η εφαρμογή που δημιουργήθηκε είναι προσβάσιμη από όποιον επιθυμεί να την χρησιμοποιήσει ,είτε είναι απλός χρήστης είτε είναι εξειδικευμένος χρήστης η εφαρμογή του παρέχει τους ίδιους πόρους .Εξειδικευμένοι χρήστες θεωρούνται χρήστες που έχουν ειδικότητα βιολόγου η χημικού και έχουν παραπάνω γνώση στην συγκεκριμένη θεματική ενότητα που πραγματεύεται το Jmol και συνάμα και η εφαρμογή που δημιουργήσαμε .

Βήματα σεναρίου:

1. Εισαγωγή στο πρόγραμμα Jmol
2. Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
3. Πρόσβαση στην εφαρμογή επιλέγοντας το μενού Visual Tools
4. Επιλογή περιβάλλοντος της εφαρμογής
5. Λειτουργία του περιβάλλοντος που επιλέχθηκε

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Εισαγωγή στο πρόγραμμα Jmol

Άνοιγμα του προγράμματος αναπαράστασης μορίων Jmol

Βήμα 2: Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα

Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα

Βήμα 3: Πρόσβαση στην εφαρμογή επιλέγοντας το μενού Visual Tools

Το μενού Visual Tools έχει τοποθετηθεί στην κεντρική μπάρα μενού του Jmol και περιέχει ως συστατικά υπομενού τα τρία μέρη που αποτελούν την εφαρμογή μας .

Βήμα 4: Επιλογή περιβάλλοντος της εφαρμογής

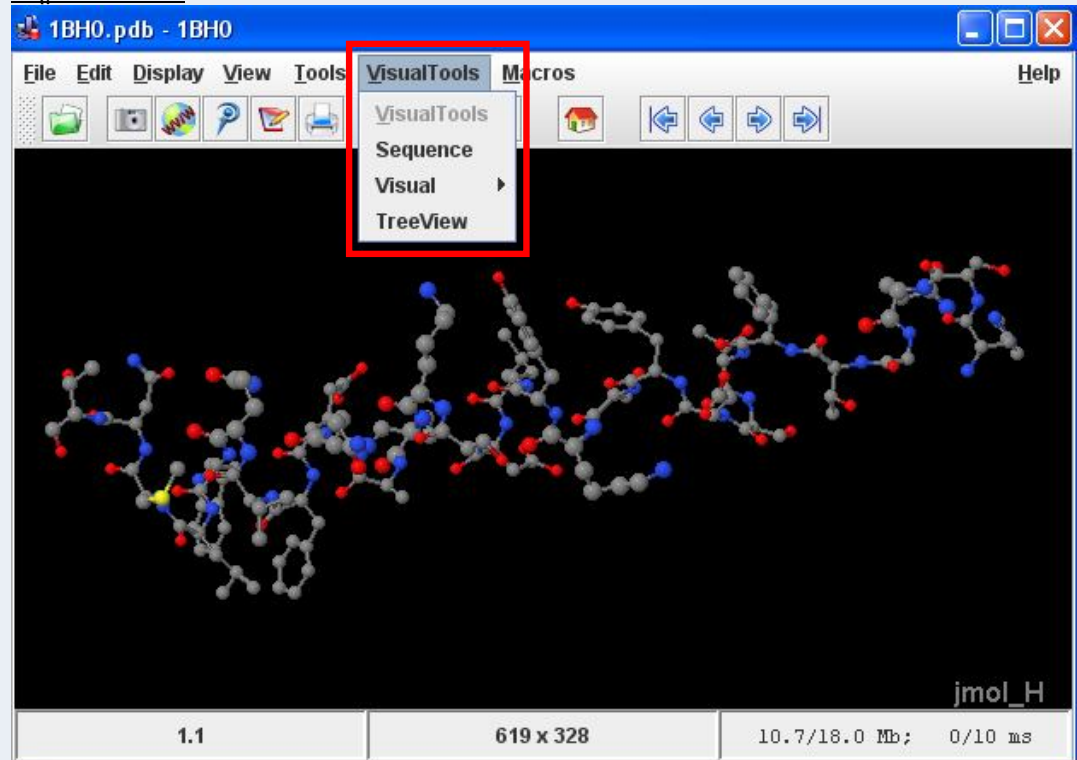
Απλά επιλέγουμε ένα από τα τρία υπομενού (Sequence ,Visual ,TreeView) που ανοίγουν ένα από τα τρία μέρη της εφαρμογής μας αντίστοιχα (Sequence view,Comment mode ,TreeView) .

Βήμα 5: Λειτουργία του περιβάλλοντος που επιλέχθηκε

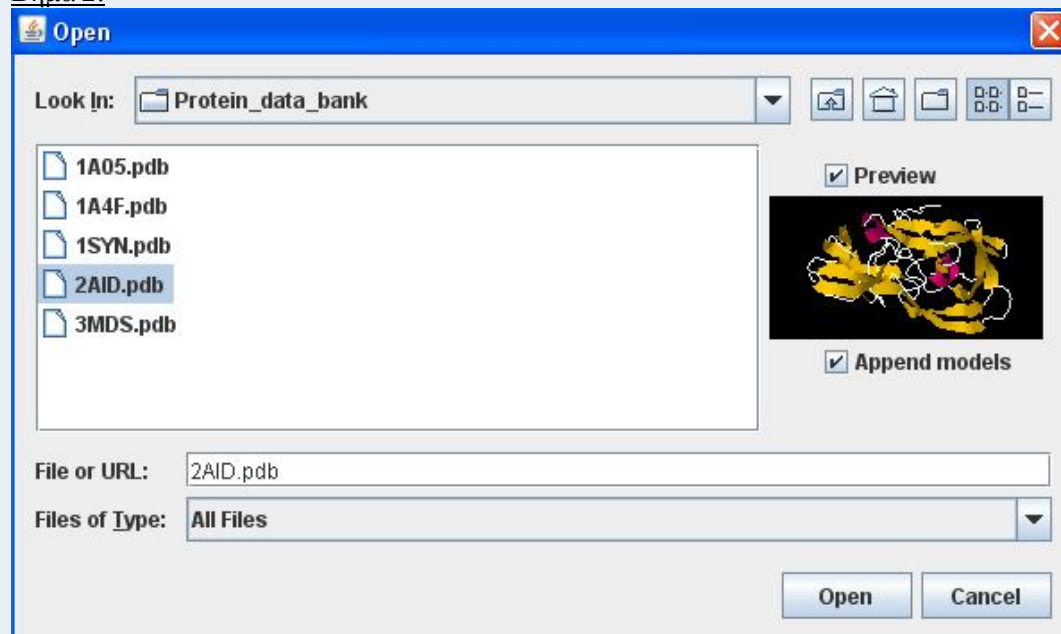
Μετά την επιλογή περιβάλλοντος ,ο χρήστης αντιμετωπίζει την λειτουργία του περιβάλλοντος που έχει επιλέξει .

Σχηματική απεικόνιση σεναρίου:

Βήμα 1 & 3&4:



Βήμα 2:



User: Απλός χρήστης ,εξειδικευμένος χρήστης (βιολόγος ,χημικός)	
Περίπτωση χρήσης: Απαιτήσεις λειτουργίας της εφαρμογής	
Σενάρια	Δράσεις
Προβολή ακολουθίας αμινοξέων	Εισαγωγή στο πρόγραμμα Jmol
	Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
	Εισαγωγή στην εφαρμογή Sequence View
	Επισκόπηση ακολουθίας αμινοξέων
Προβολή των δομικών συστατικών ενός μορίου	Εισαγωγή στο πρόγραμμα Jmol
	Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
	Εισαγωγή στην εφαρμογή Jtree
	Επισκόπηση των δομικών συστατικών ενός μορίου
Ενεργοποίηση της λειτουργίας εισαγωγής σχολίου	Εισαγωγή στο πρόγραμμα Jmol
	Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
	Εισαγωγή στην λειτουργία addComment
	Επιλογή του ατόμου για την εισαγωγή του σχολίου

Table 5: Δράσεις σεναρίων περίπτωσης χρήσης «Απαιτήσεις λειτουργίας της εφαρμογής»

Περίπτωση χρήσης: Απαιτήσεις λειτουργίας της εφαρμογής

Σενάριο: Προβολή ακολουθίας αμινοξέων

Η εφαρμογή που δημιουργήθηκε παρέχει μέσα από την εφαρμογή Sequence View την επισκόπηση της ακολουθίας αμινοξέων που είναι δυνατή από τον χρήστη

Βήματα σεναρίου:

1. Εισαγωγή στο πρόγραμμα Jmol
2. Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
3. Εισαγωγή στην εφαρμογή Sequence View
4. Επισκόπηση ακολουθίας αμινοξέων
5. Πληροφορία των χαρακτηριστικών χρωμάτων των αμινοξέων

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Εισαγωγή στο πρόγραμμα Jmol

Άνοιγμα του προγράμματος αναπαράστασης μορίων Jmol

Βήμα2: Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα

Εισαγωγή στο πρόγραμμα ,πληροφορίας που παρέχει όλα τα δεδομένα για την αναπαράσταση των βιολογικών δομών ,είτε είναι pdb αρχείο ,είτε είναι πληροφορία παρεχόμενη μέσω διαδικτύου

Βήμα3: Εισαγωγή στην εφαρμογή Sequence View

Άνοιγμα της εφαρμογής από το μενού επιλογής

Βήμα4: Επισκόπηση ακολουθίας αμινοξέων

Εμφάνιση της ακολουθίας αμινοξέων

Βήμα5: Πληροφορία των χαρακτηριστικών χρωμάτων των αμινοξέων

Άνοιγμα του παραθυρικού περιβάλλοντος που περιέχει τα χαρακτηριστικά χρώματα των αμινοξέων
























Σχηματική απεικόνιση σεναρίου:

Βήμα 3&4:

```
frame1=new JFrame("SequenceView");
tmp = new JEditorPane();
panel = new JPanel();
.....
panel.add(pane);
panel.add(button2);
panel.add(combo);
panel.add(button);
frame1.add(panel, BorderLayout.SOUTH);
frame1.setSize(viewer.getScreenWidth(), 100);
frame1.setLocation(0, 615);
frame1.pack();
frame1.show();
```



Βήμα 5:

Amino acids color					
	<i>Ala-->A</i>		<i>Arg-->R</i>		<i>Asn-->N</i>
	<i>Asp-->D</i>		<i>Cys-->C</i>		<i>Gln-->Q</i>
	<i>Glu-->E</i>		<i>Gly-->G</i>		<i>His-->H</i>
	<i>Ile-->I</i>		<i>Leu-->L</i>		<i>Lys-->K</i>
	<i>Met-->M</i>		<i>Phe-->F</i>		<i>Pro-->P</i>
	<i>Ser-->S</i>		<i>Thr-->T</i>		<i>Trp-->W</i>
	<i>Tyr-->Y</i>		<i>Val-->V</i>		<i>Asx-->X</i>
	<i>Glx-->X</i>		<i>Other</i>		

Περίπτωση χρήσης: Απαιτήσεις λειτουργίας της εφαρμογής

Σενάριο: Προβολή των βιολογικών δομικών συστατικών ενός μορίου

Η εφαρμογή παρέχει μέσα από το περιβάλλον TreeView την επισκόπηση των επιμέρους βιολογικών δομικών συστατικών που αποτελούν ένα μόριο ή μια πρωτεΐνη

Βήματα σεναρίου:

1. Εισαγωγή στο πρόγραμμα Jmol
2. Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
3. Εισαγωγή στο περιβάλλον Jtree
4. Επισκόπηση των δομικών συστατικών ενός μορίου

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Εισαγωγή στο πρόγραμμα Jmol

Άνοιγμα του προγράμματος αναπαράστασης μορίων Jmol

Βήμα2: Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα

Εισαγωγή στο πρόγραμμα ,πληροφορίας που παρέχει όλα τα δεδομένα για την αναπαράσταση των βιολογικών δομών ,είτε είναι pdb αρχείο ,είτε είναι πληροφορία παρεχόμενη μέσω διαδικτύου

Βήμα3: Εισαγωγή στο περιβάλλον Jtree

Άνοιγμα της εφαρμογής από το μενού επιλογής

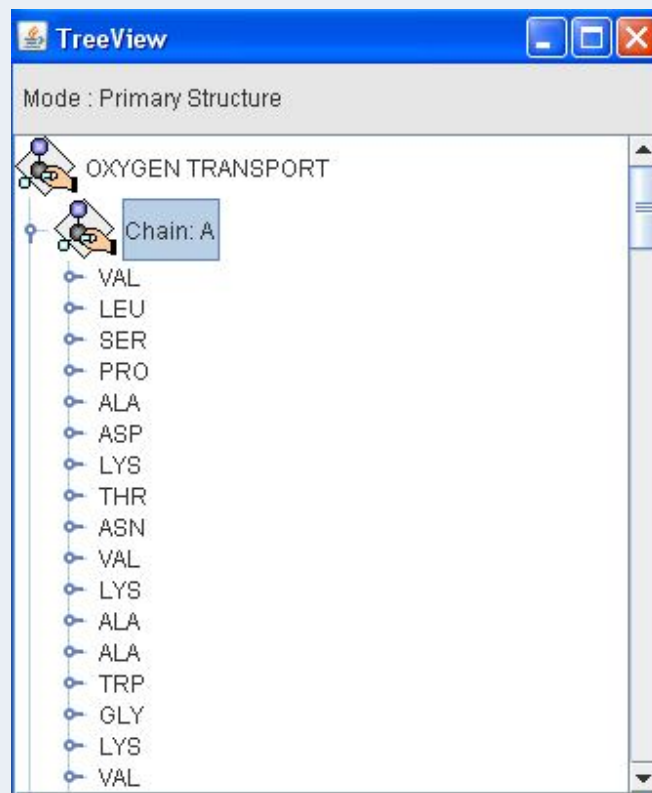
Βήμα4: Επισκόπηση των δομικών συστατικών ενός μορίου

Εμφάνιση των δομικών συστατικών του μορίου που αναπαρίσταται

Σχηματική απεικόνιση σεναρίου:

Βήμα 3&4:

```
super("TreeView");  
Container content = getContentPane();  
.....  
primTree = createPrimTree();  
treeCanvas = new JTree(primTree);  
.....
```



Περίπτωση χρήσης: Απαιτήσεις λειτουργίας της εφαρμογής

Σενάριο: Ενεργοποίηση της λειτουργίας εισαγωγής σχολίου

Μέσω της λειτουργίας εισαγωγής σχολίου μπορεί ο χρήστης να εισάγει σχόλια σε κάθε άτομο ξεχωριστά ,επιλέγοντας το άτομο που επιθυμεί να εισαχθεί το σχόλιο

Βήματα σεναρίου:

1. Εισαγωγή στο πρόγραμμα Jmol
2. Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
3. Εισαγωγή στην εφαρμογή addComment

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Εισαγωγή στο πρόγραμμα Jmol

Άνοιγμα του προγράμματος αναπαράστασης μορίων Jmol

Βήμα2: Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα

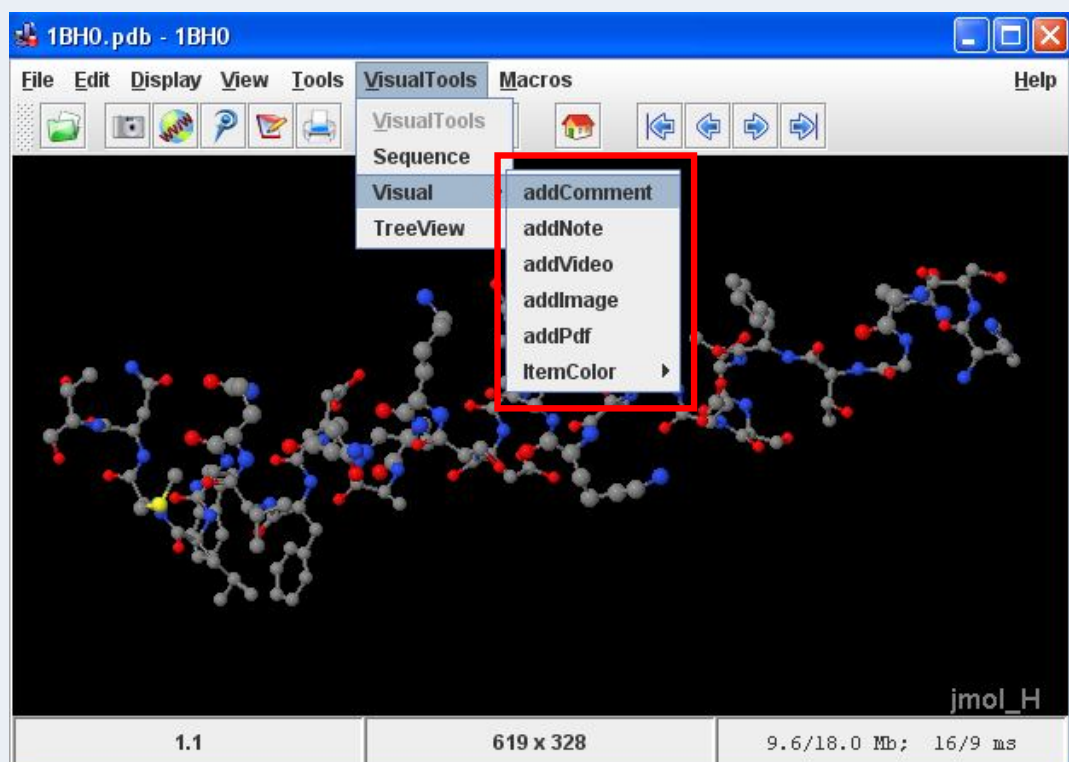
Εισαγωγή στο πρόγραμμα ,πληροφορίας που παρέχει όλα τα δεδομένα για την αναπαράσταση των βιολογικών δομών ,είτε είναι pdb αρχείο ,είτε είναι πληροφορία παρεχόμενη μέσω διαδικτύου

Βήμα3: Εισαγωγή στην εφαρμογή addComment

Άνοιγμα της εφαρμογής από το μενού επιλογής

Σχηματική απεικόνιση σεναρίου:

Βήμα 3:



User: Απλός χρήστης ,εξειδικευμένος χρήστης (βιολόγος ,χημικός)	
Περίπτωση χρήσης: Απαιτήσεις λειτουργιών της εφαρμογής	
Σενάρια	Δράσεις
Επισήμανση των αμινοξέων πάνω στο μόριο που αναπαρίσταται στην σκηνή του Jmol	Εισαγωγή στο πρόγραμμα Jmol
	Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
	Εισαγωγή στην εφαρμογή Sequence View
	Επιλογή αμινοξέου από την αμινοξική ακολουθία
Επισήμανση των βιολογικών δομικών συστατικών πάνω στο μόριο που αναπαρίσταται στην σκηνή του Jmol	Εισαγωγή στο πρόγραμμα Jmol
	Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
	Εισαγωγή στην εφαρμογή TreeView
	Επιλογή της βιολογικής δομής από την εφαρμογή του δέντρου
Αύξηση μεγέθους των επιλεγμένων αμινοξέων του μορίου που αναπαρίσταται στην σκηνή του Jmol	Εισαγωγή στο πρόγραμμα Jmol
	Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
	Εισαγωγή στην εφαρμογή Sequence View
	Επιλογή αμινοξέου από την αμινοξική ακολουθία
	Μεγέθυνση επιλεγμένων αμινοξέων
Κοινοποίηση πληροφορίας σχετική με τα χαρακτηριστικά χρώματα των αμινοξέων στο χρήστη	Εισαγωγή στο πρόγραμμα Jmol
	Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
	Εισαγωγή στην εφαρμογή Sequence View
	Άνοιγμα του παραθυρικού περιβάλλοντος amino acid color
	Επισκόπηση πληροφορίας
Εισαγωγή σχολίου	Εισαγωγή στο πρόγραμμα Jmol
	Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
	Εισαγωγή στην εφαρμογή Comment mode
	Εισαγωγή σχολίου σε άτομο
	Επιλογή γραφικού σχολίου
	Εισαγωγή σχολίου στο text field
	Επιλογή επικύρωσης σχολίου
Αφαίρεση σχολίου	Εισαγωγή στην εφαρμογή Comment mode
	Επιλογή του ατόμου για την αφαίρεση του ήδη υπάρχοντος σχολίου

Λειτουργία αποθήκευσης	Επιλογή της λειτουργίας αποθήκευσης
	Επιλογή της τοποθεσίας που θέλουμε να αποθηκευτεί το αρχείο
	Όνομα αρχείου αποθήκευσης
	Επιλογή επικύρωσης αποθήκευσης

Table 6: Δράσεις σεναρίων περιπτώσεις χρήσης «Απαιτήσεις λειτουργιών της εφαρμογής»

Περίπτωση χρήσης: Απαιτήσεις λειτουργιών της εφαρμογής

Σενάριο: Επισήμανση των βιολογικών δομικών συστατικών πάνω στο μόριο που αναπαρίσταται στην σκηνή του Jmol

Η εφαρμογή TreeView μας δίνει την αναπαράσταση των κυριότερων δομικών συστατικών ενός μορίου σε δενδροειδή μορφή .Ο χρήστης μπορεί να επιλέξει οποιοδήποτε δομικό συστατικό που αναπαρίσταται στην εφαρμογή TreeView και αυτό να επισημανθεί στην σκηνή του Jmol

Βήματα σεναρίου:

1. Εισαγωγή στο πρόγραμμα Jmol
2. Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
3. Εισαγωγή στην εφαρμογή TreeView
4. Επιλογή της βιολογικής δομής από την εφαρμογή του δέντρου

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Εισαγωγή στο πρόγραμμα Jmol

Άνοιγμα του προγράμματος αναπαράστασης μορίων Jmol

Βήμα2: Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα

Εισαγωγή στο πρόγραμμα ,πληροφορίας που παρέχει όλα τα δεδομένα για την αναπαράσταση των βιολογικών δομών ,είτε είναι pdb αρχείο ,είτε είναι πληροφορία παρερχομένη μέσω διαδικτύου

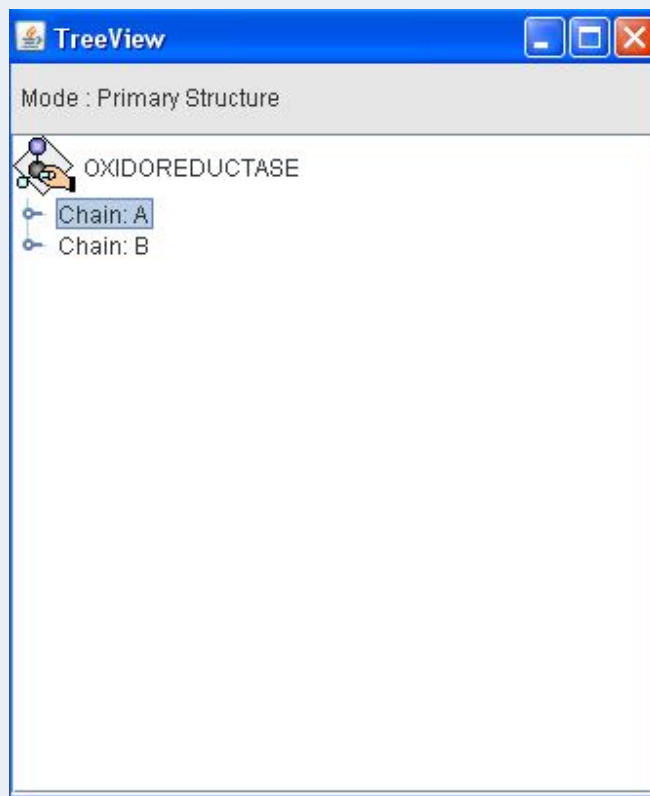
Βήμα3: Εισαγωγή στην εφαρμογή TreeView

Άνοιγμα της εφαρμογής από το μενού επιλογής

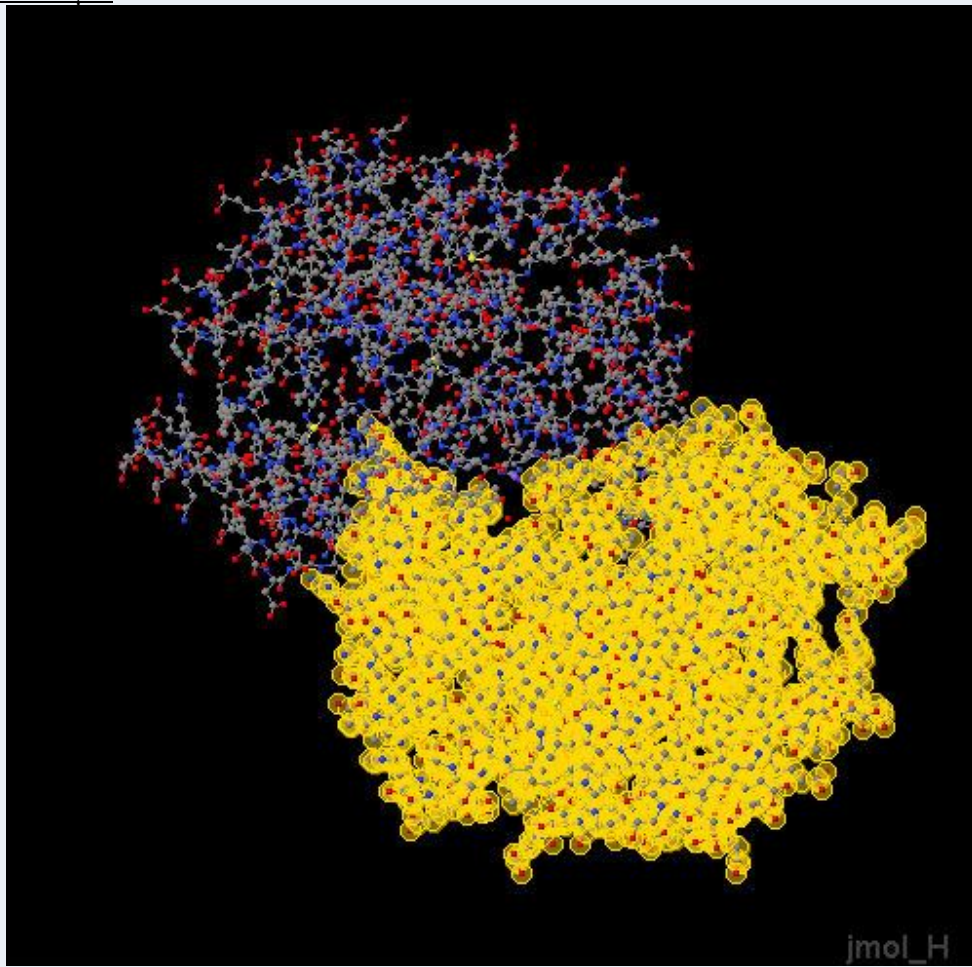
Βήμα4: Επιλογή της βιολογικής δομής από την εφαρμογή του δέντρου

Ο χρήστης με ένα απλό κλικ με το mouse πάνω στην βιολογική δομή που επιθυμεί να επιλέξει από την εφαρμογή δένδρου ,επισημαίνεται η δομή και στην σκηνή του Jmol(επισημαίνονται τα άτομα που αποτελούν την δομή)

Σχηματική απεικόνιση σεναρίου:
Βήμα 3&4:



Αποτέλεσμα:



Περίπτωση χρήσης: Απαιτήσεις λειτουργιών της εφαρμογής

Σενάριο: Επισήμανση των αμινοξέων πάνω στο μόριο που αναπαρίσταται στην σκηνή του Jmol

Επιλέγοντας ένα οποιοδήποτε αμινοξύ από την αμινοξική ακολουθία που μας δίδεται από την εφαρμογή Sequence View, το επιλεγμένο αμινοξύ επισημαίνεται στην σκηνή αναπαράστασης του Jmol

Βήματα σεναρίου:

1. Εισαγωγή στο πρόγραμμα Jmol
2. Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
3. Εισαγωγή στην εφαρμογή Sequence View
4. Επιλογή αμινοξέου από την αμινοξική ακολουθία

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Εισαγωγή στο πρόγραμμα Jmol

Άνοιγμα του προγράμματος αναπαράστασης μορίων Jmol

Βήμα2: Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα

Εισαγωγή στο πρόγραμμα ,πληροφορίας που παρέχει όλα τα δεδομένα για την αναπαράσταση των βιολογικών δομών ,είτε είναι pdb αρχείο ,είτε είναι πληροφορία παρεχόμενη μέσω διαδικτύου

Βήμα3: Εισαγωγή στην εφαρμογή Sequence View

Άνοιγμα της εφαρμογής από το μενού επιλογής

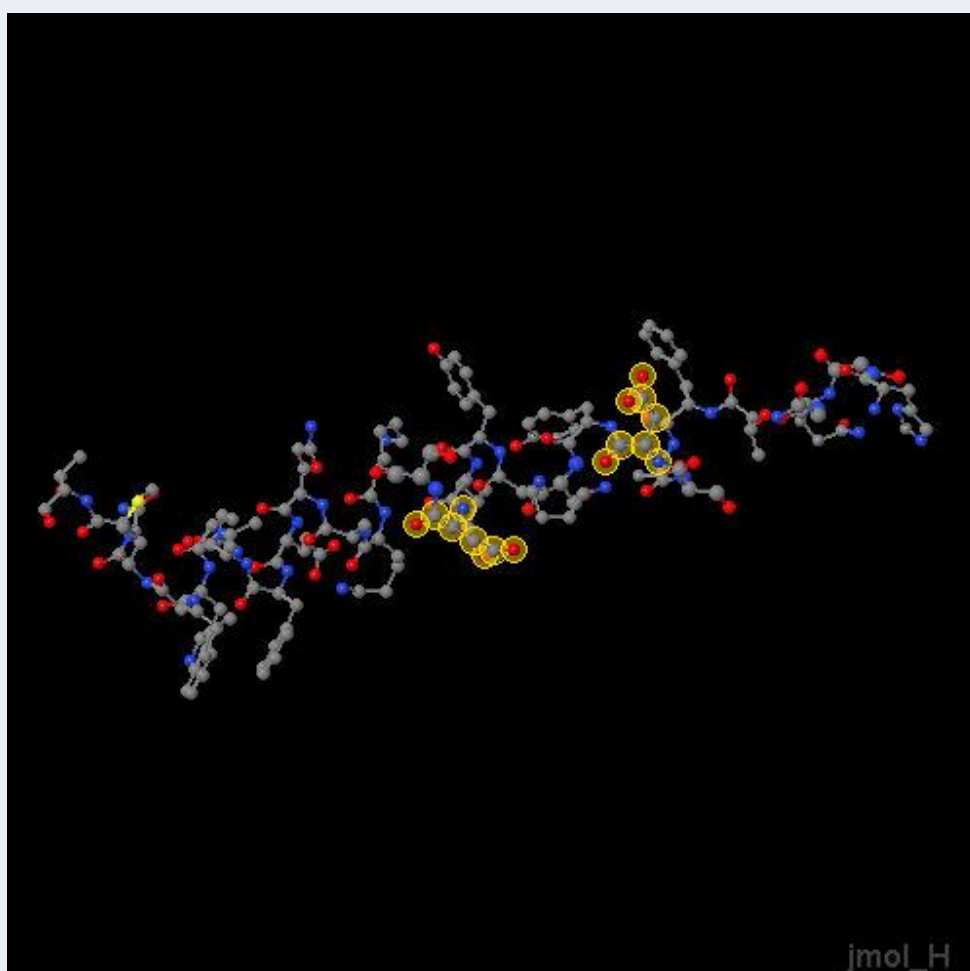
Βήμα4: Επιλογή αμινοξέου από την αμινοξική ακολουθία

Ο χρήστης με ένα απλό κλικ με το mouse πάνω στην αμινοξική ακολουθία μπορεί να επιλέξει το αμινοξύ που επιθυμεί

Σχηματική απεικόνιση σεναρίου:
Βήμα 4:



Αποτέλεσμα:



Περίπτωση χρήσης: Απαιτήσεις λειτουργιών της εφαρμογής

Σενάριο: Αύξηση μεγέθους των επιλεγμένων ατόμων του μορίου που αναπαρίσταται στην σκηνή του Jmol

Επιλέγοντας από την πτυσσόμενη λίστα της εφαρμογής Sequence View την επιλογή max atom ,τα είδη επιλεγμένα αμινοξέα «παχαίνουν» μεγαλώνουν σε μέγεθος και γίνονται πιο ευδιάκριτα

Βήματα σεναρίου:

1. Εισαγωγή στο πρόγραμμα Jmol
2. Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
3. Εισαγωγή στην εφαρμογή Sequence View
4. Επιλογή αμινοξέου από την αμινοξική ακολουθία
5. Μεγέθυνση επιλεγμένου αμινοξέου

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Εισαγωγή στο πρόγραμμα Jmol

Άνοιγμα του προγράμματος αναπαράστασης μορίων Jmol

Βήμα2: Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα

Εισαγωγή στο πρόγραμμα ,πληροφορίας που παρέχει όλα τα δεδομένα για την αναπαράσταση των βιολογικών δομών ,είτε είναι pdb αρχείο ,είτε είναι πληροφορία παρεχόμενη μέσω διαδικτύου

Βήμα3: Εισαγωγή στην εφαρμογή Sequence View

Άνοιγμα της εφαρμογής από το μενού επιλογής

Βήμα4: Επιλογή αμινοξέου από την αμινοξική ακολουθία

Ο χρήστης με ένα απλό κλικ με το mouse πάνω στην αμινοξική ακολουθία μπορεί να επιλέξει το αμινοξέο που επιθυμεί

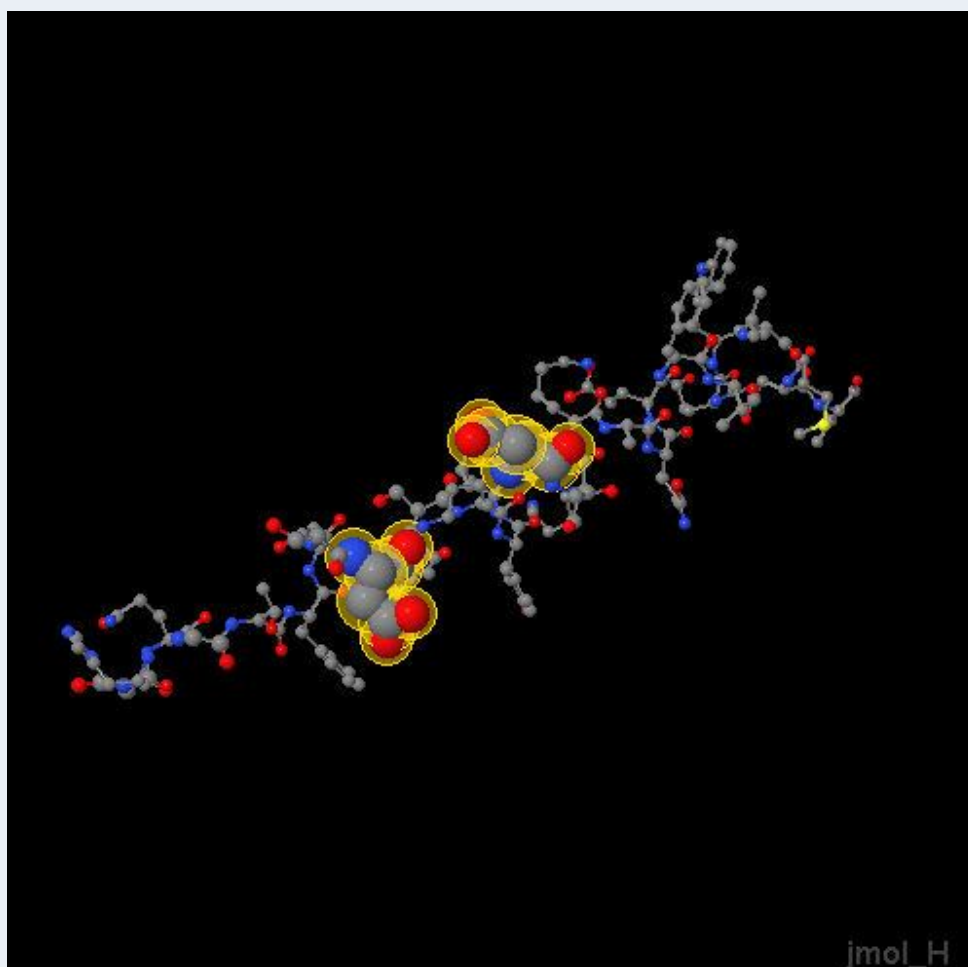
Βήμα5: Μεγέθυνση επιλεγμένου αμινοξέου

Επιλέγοντας ο χρήστης την επιλογή max atom από την πτυσσόμενη λίστα μεγεθύνει τα είδη επιλεγμένα αμινοξέα κάνοντας τα πιο ευδιάκριτα

Σχηματική απεικόνιση σεναρίου:
Βήμα 3&4&5:



Αποτέλεσμα:



Περίπτωση χρήσης: Απαιτήσεις λειτουργιών της εφαρμογής

Σενάριο: Εισαγωγή σχολίου

Ο χρήστης έχει την δυνατότητα εισαγωγής σχολίου σε κάθε άτομο ξεχωριστά και αποθήκευσης του σχολίου αυτού

Βήματα σεναρίου:

1. Εισαγωγή στο πρόγραμμα Jmol
2. Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα
3. Εισαγωγή στην εφαρμογή Comment mode
4. Εισαγωγή σχολίου σε άτομο
5. Επιλογή γραφικού σχολίου
6. Εισαγωγή σχολίου στο text field
7. Επιλογή επικύρωσης σχολίου

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Εισαγωγή στο πρόγραμμα Jmol

Άνοιγμα του προγράμματος αναπαράστασης μορίων Jmol

Βήμα2: Εισαγωγή της κατάλληλης πληροφορίας στο πρόγραμμα

Εισαγωγή στο πρόγραμμα ,πληροφορίας που παρέχει όλα τα δεδομένα για την αναπαράσταση των βιολογικών δομών ,είτε είναι pdb αρχείο ,είτε είναι πληροφορία παρεχόμενη μέσω διαδικτύου

Βήμα3: Εισαγωγή στην εφαρμογή Comment mode

Άνοιγμα της εφαρμογής από το μενού επιλογής

Βήμα4: Εισαγωγή σχολίου σε άτομο

Ο χρήστης με ένα απλό κλικ με το mouse πάνω σε ένα άτομο, μπορεί να εμφανίσει πάνω στο άτομο το γραφικό εισαγωγής σχολίου

Βήμα5: Επιλογή γραφικού σχολίου

Ο χρήστης με ένα κλικ πάνω στο γραφικό σχολίου που εμφανίστηκε ανοίγει το παράθυρο εισαγωγής κειμένου

Βήμα6: Εισαγωγή σχολίου στο text field

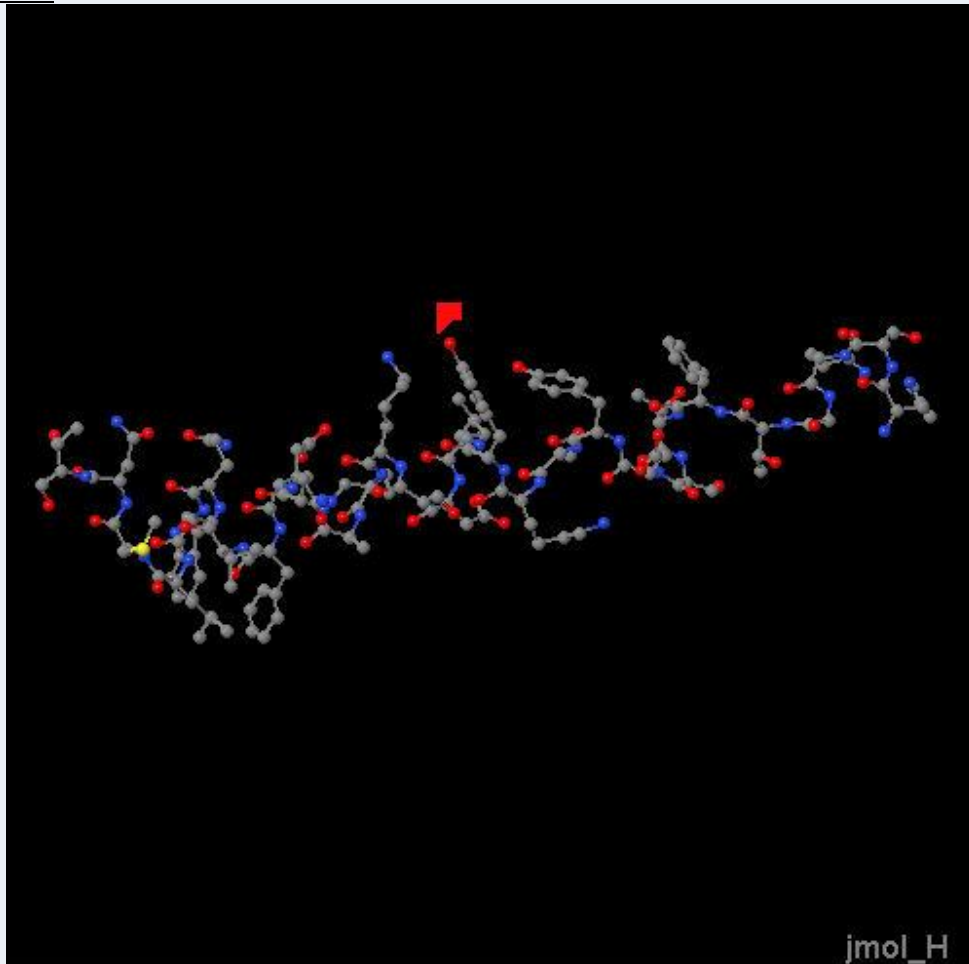
Ο χρήστης εισάγει το επιθυμητό σχόλιο στην περιοχή κειμένου

Βήμα7: Επιλογή επικύρωσης σχολίου

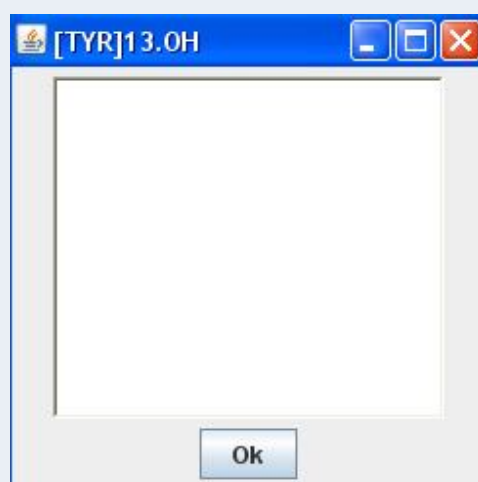
Ο χρήστης επιλέγει το κουμπί επικύρωσης για την αποθήκευση(προσωρινή) του σχολίου και την σύνδεση του με το συγκεκριμένο γραφικό και συνάμα με το άτομο που αντιπροσωπεύει το γραφικό αυτό

Σχηματική απεικόνιση σεναρίου:

Βήμα 4:



Βήμα 6:



Περίπτωση χρήσης: Απαιτήσεις λειτουργιών της εφαρμογής

Σενάριο: Αφαίρεση σχολίου

Στον χρήστη παρέχεται η δυνατότητα αφαίρεσης ενός σχολίου από ένα άτομο

Βήματα σεναρίου:

1. Εισαγωγή στην εφαρμογή Comment mode
2. Επιλογή του ατόμου για την αφαίρεση του ήδη υπάρχοντος σχολίου

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα1: Εισαγωγή στην εφαρμογή Comment mode

Ανοιγμα της εφαρμογής από το μενού επιλογής

Βήμα2: Επιλογή του ατόμου για την αφαίρεση του ήδη υπάρχοντος σχολίου

Ο χρήστης με ένα απλό κλικ με το mouse πάνω σε ένα άτομο που έχει εισαχθεί σχόλιο ,αναιρείτε το γραφικό σχολίου που συνοδεύει το άτομο και συνάμα και το σχόλιο που συνοδεύει το συγκεκριμένο άτομο

Περίπτωση χρήσης: Απαιτήσεις λειτουργιών της εφαρμογής

Σενάριο: Λειτουργία αποθήκευσης

Στον χρήστη παρέχεται η δυνατότητα αποθήκευσης των σχολίων που έχει εισάγει σε άτομα που αναπαρίστανται στην σκηνή σε αρχείο αποθήκευσης της πληροφορίας

Βήματα σεναρίου:

1. Επιλογή της λειτουργίας αποθήκευσης
2. Επιλογή τοποθεσίας που θέλουμε να αποθηκευτή το αρχείο
3. Όνομα αρχείου αποθήκευσης
4. Επιλογή επικύρωσης αποθήκευσης

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα1: Επιλογή της λειτουργίας αποθήκευσης

Ο χρήστης επιλέγει την λειτουργία αποθήκευσης είτε από το μενού του Jmol είτε από την γραμμή εργαλείων του προγράμματος

Βήμα2: Επιλογή τοποθεσίας που θέλουμε να αποθηκευτή το αρχείο

Ο χρήστης επιλέγει την τοποθεσία αποθήκευσης του αρχείου που περιέχει τα σχόλια εισαγωγής

Βήμα3: Όνομα αρχείου αποθήκευσης

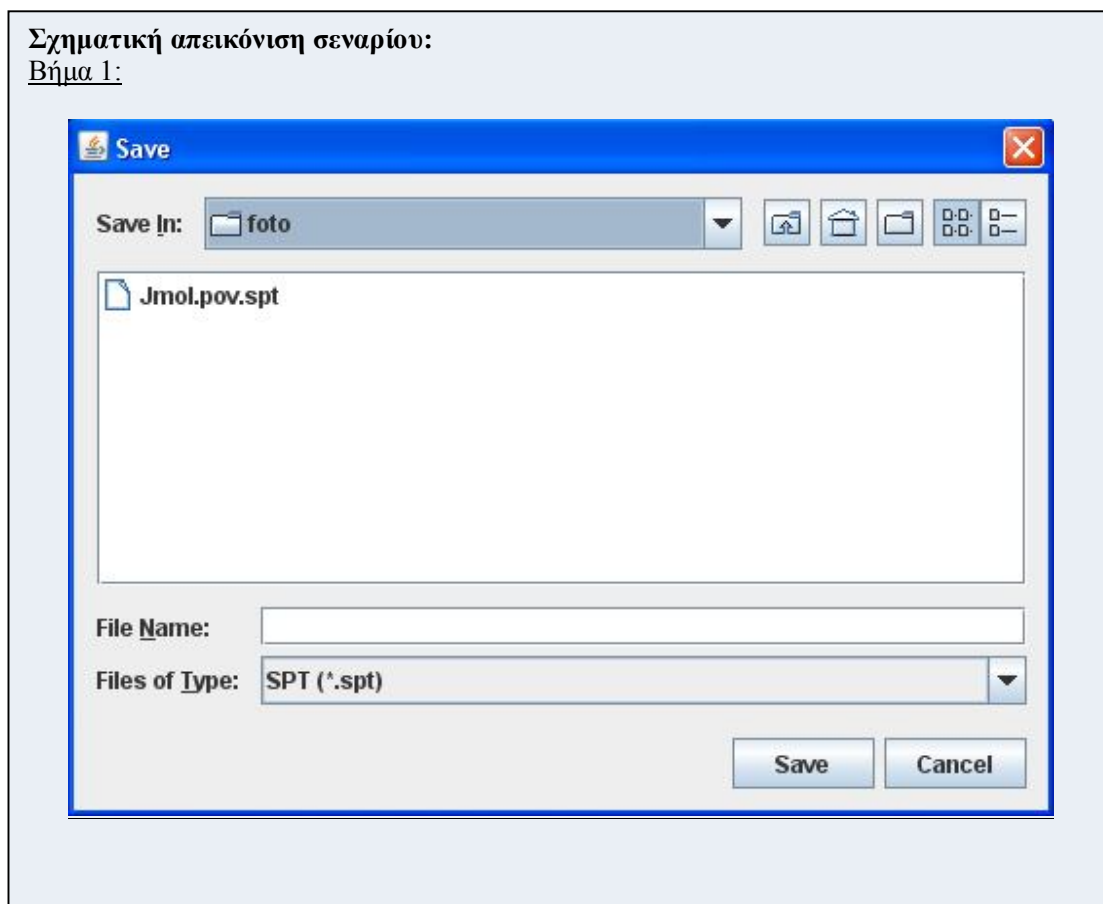
Ο χρήστης δίνει ένα επιθυμητό όνομα του αρχείου

Βήμα4: Επιλογή επικύρωσης αποθήκευσης

Ο χρήστης επικυρώνει την αποθήκευση της πληροφορίας στο αρχείο που έχει ονομάσει ,επιλέγοντας την επιλογή αποθήκευσης (Save)

Σχηματική απεικόνιση σεναρίου:

Βήμα 1:



7. Υλοποίηση της εφαρμογής

Σε αυτό το κεφάλαιο παρατίθεται η υλοποίηση της εφαρμογής, τα παραθυρικά περιβάλλοντα διεπαφής και οι λειτουργίες που εκτελούνται μέσα από αυτά τα περιβάλλοντα.

Η υλοποίηση της εφαρμογής αφορά δυο αντικείμενα αναφοράς την διεπαφή της εφαρμογής και της λειτουργίες που ενεργοποιούνται μέσα από την διεπαφή της εφαρμογής. Τα περιβάλλοντα που έχουν υλοποιηθεί περιέχουν την οπτική αντιπροσώπευση της πληροφορίας και των δεδομένων των βιολογικών δομών που αναπαρίστανται στην σκηνή του Jmol. Οι λειτουργίες που υλοποιήθηκαν και ενεργοποιούνται μέσα από τα περιβάλλοντα διεπαφής βοηθούν τον χρήστη να κατανοήσει καλύτερα τα δεδομένα που αναπαρίστανται από τα περιβάλλοντα και να τα επεξεργαστεί.

7.1. User Interface.

Η διεπαφή χρήστη (User Interface) είναι το σύνολο των μέσων με τα οποία οι χρήστες αλληλεπιδρούν με ένα συγκεκριμένο μηχάνημα, συσκευή, υπολογιστικό πρόγραμμα ή άλλο περίπλοκο εργαλείο (το σύστημα). Η διεπαφή χρήστη παρέχει μέσα εισόδου (input), που επιτρέπουν στους χρήστες να χειρίζονται το σύστημα και μέσα εξόδου (output), που επιτρέπουν στο σύστημα να παράγουν τα εφέ από τη διαχείριση των χρηστών.

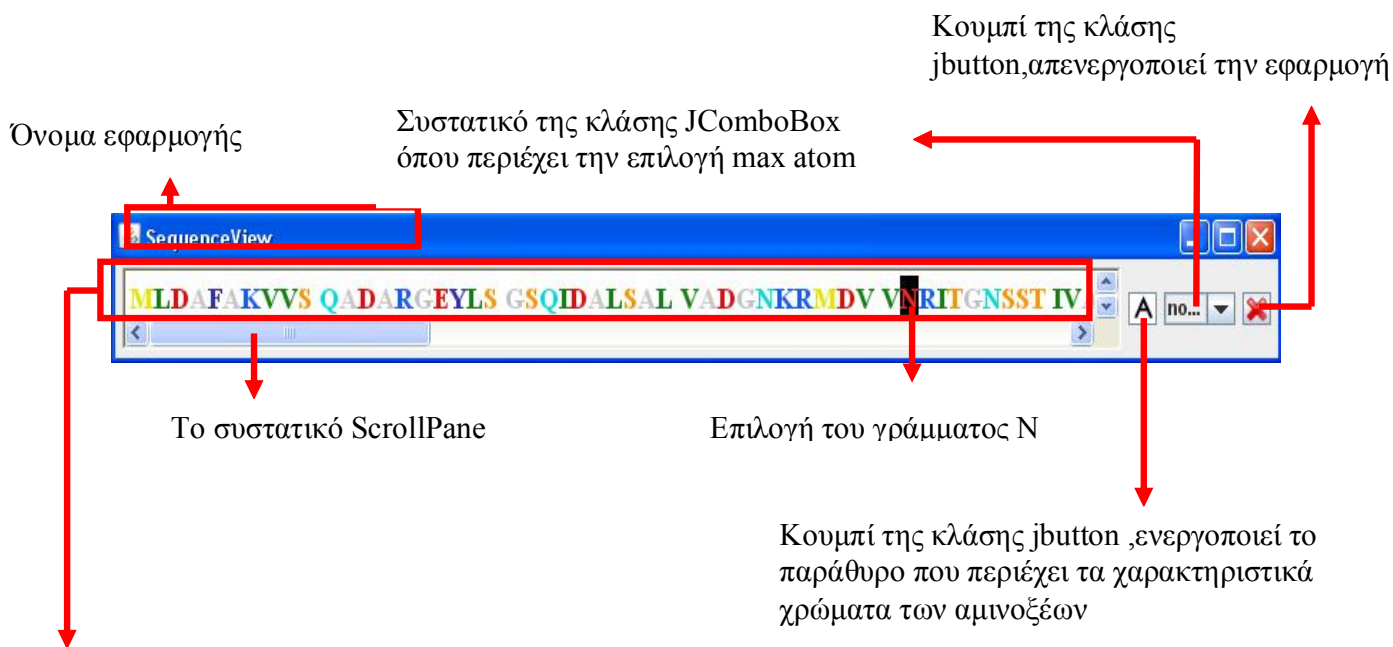
Στην επιστήμη των υπολογιστών και στην αλληλεπίδραση ανθρώπου-υπολογιστή, το user interface αναφέρεται στην πληροφορία που αφορά γραφικά, κείμενο και ήχο που παρουσιάζει ένα πρόγραμμα στο χρήστη καθώς και τις ακολουθίες ελέγχου (όπως πάτημα κουμπιών μέσω του πληκτρολογίου, κινήσεις μέσω του ποντικιού και επιλογές μέσω του touchscreen) τις οποίες απασχολεί ο χρήστης για τον έλεγχο του προγράμματος.

Ένα κοινό user interfaces είναι το ακόλουθο:

- *Graphical user interfaces (GUI)*: Τα γραφικά user interfaces δέχονται είσοδο από συσκευές όπως το πληκτρολόγιο του υπολογιστή και το ποντίκι και παρέχουν αρθρωτή γραφική έξοδο στην οθόνη του υπολογιστή. Υπάρχουν τουλάχιστον δύο διαφορετικές αρχές που χρησιμοποιούνται ευρέως στο σχεδιασμό GUI: Object-oriented user interfaces (OOUIs) και application oriented interfaces.

7.1.1. Η εφαρμογή SequenceView

Η εφαρμογή SequenceView που το περιβάλλον της φαίνεται στην παρακάτω εικόνα μας παρέχει την ακολουθία αμινοξέων του βιολογικού μορίου που αναπαρίσταται στην σκηνή του Jmol. Ο χρήστης επιλέγοντας κάθε φορά ένα γράμμα (το κάθε αμινοξύ αντιπροσωπεύεται από ένα γράμμα) αμινοξέου από την ακολουθία αμινοξέων (όπως διακρίνεται παρακάτω έχει επιλεγεί το γράμμα N που αντιπροσωπεύει το αμινοξύ Ασπαραγίνη) ενεργοποιεί την λειτουργία επισήμανσης (περιβάλλεται με κίτρινο χρώμα το άτομο) στην σκηνή του Jmol όλων των ατόμων που αποτελούν το κάθε αμινοξύ που αντιπροσωπεύεται από το γράμμα N και υπάρχουν στην βιολογική δομή που αναπαρίσταται (figure 12). Παρακάτω στην εικόνα 12 βλέπουμε στην σκηνή του Jmol την επισήμανση των αμινοξέων που χαρακτηρίζονται από το γράμμα N που έχει επιλεγεί.



Η ακολουθία αμινοξέων όπου περιέχεται σε ένα συστατικό JEditorPane του πακέτου Java Swing

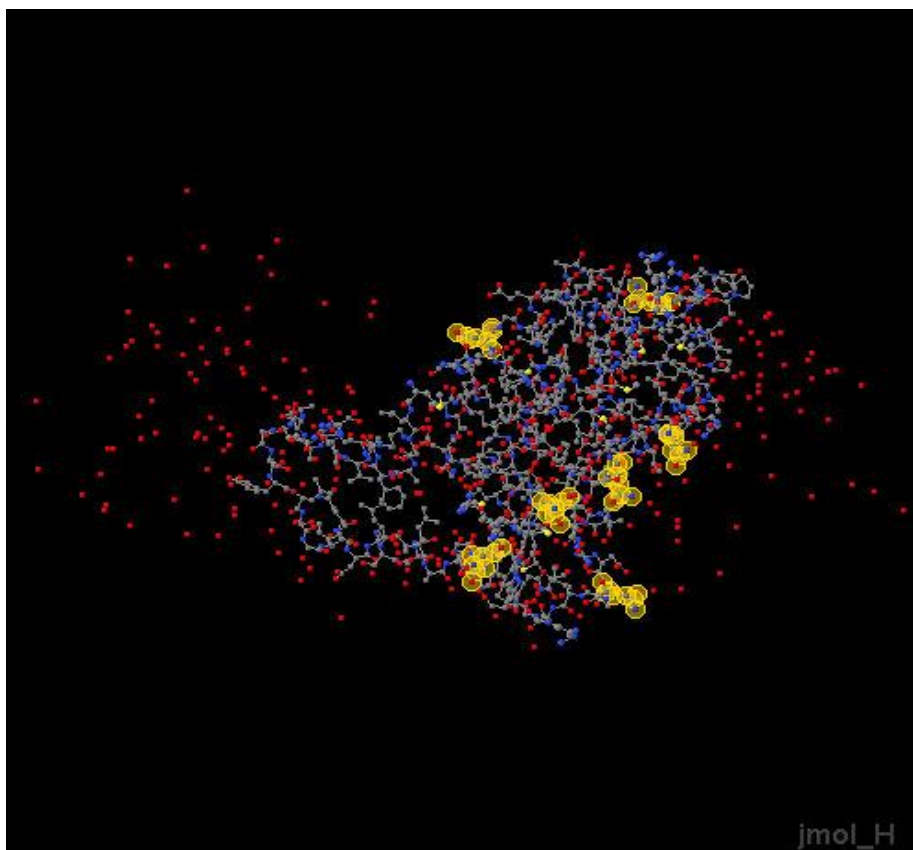

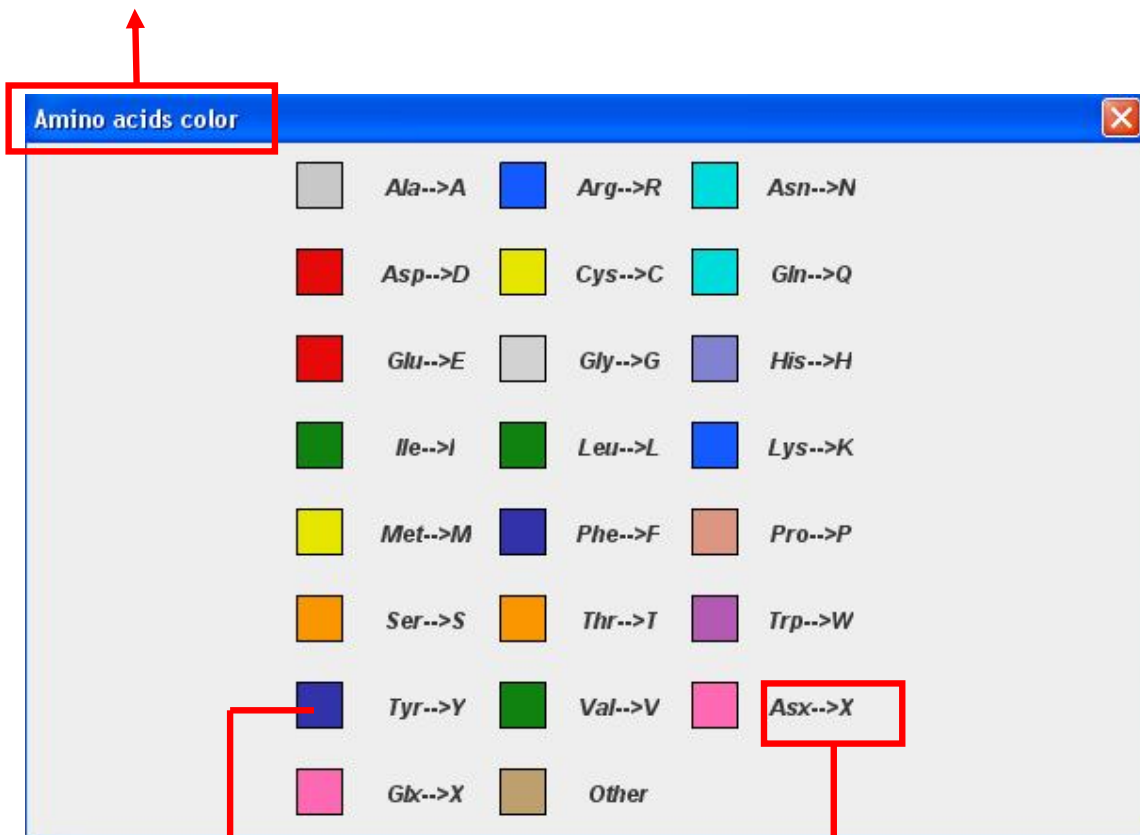


Figure 12: Επισημάνση αμινοξέων

Το κάθε είδος αμινοξέου αντιπροσωπεύεται από ένα χαρακτηριστικό χρώμα ,αν παρατηρήσουμε την ακολουθία αμινοξέων όπως αυτή αναπαρίσταται από την εφαρμογή SequenceView θα διαπιστώσουμε ότι κάθε αντιπροσωπευτικό γράμμα αμινοξέου είναι χρωματισμένο με ένα συγκεκριμένο χαρακτηριστικό χρώμα του είδους του αμινοξέου που αντιπροσωπεύει .Ο χρήστης έχει την επιλογή να πληροφορηθεί μέσω της εφαρμογής SequenceView τα χαρακτηριστικά χρώματα των αμινοξέων καθώς και σε ποιο είδος αμινοξέου αντιστοιχεί κάθε αντιπροσωπευτικό γράμμα της ακολουθίας .Όλη η πληροφορία που περιγράφηκε γίνεται γνωστή στο χρήστη ενεργοποιώντας το παραθυρικό περιβάλλον Amino acid color όπως φαίνεται παρακάτω .Το παραθυρικό περιβάλλον ενεργοποιείτε με το button  που βρίσκεται στο περιβάλλον της εφαρμογής SequenceView .

Όνομα παραθύρου



Αντιστοιχία του ονόματος του αμινοξέου με τον χαρακτήρα που αντιπροσωπεύει το όνομα του

Χαρακτηριστικό χρώμα αμινοξέου

Figure 13: Τα χαρακτηριστικά χρώματα των αμινοξέων όπως παρουσιάζονται στο περιβάλλον Amino acid color

Στον χρήστη παρέχεται η δυνατότητα αύξησης του μεγέθους των ατόμων κατά 25 τοις εκατό του μεγέθους τους, που έχουν είδη επισημανθεί επιλέγοντας την επιλογή max atom, αυτή η λειτουργία ενεργοποιείται από το περιβάλλον της εφαρμογής SequenceView μέσα από το συστατικό JComboBox και την επιλογή max atom. Με την αύξηση του μεγέθους των ατόμων επιτυγχάνετε η καλύτερη επισκόπηση τους, με αυτόν τον τρόπο γίνεται πιο σαφής η θέση τους μέσα στην δομή του ατόμου που αναπαρίσταται. Στην εικόνα παρακάτω βλέπουμε την μεγέθυνση των ατόμων που έχουν επισημανθεί με την λειτουργία max atom.

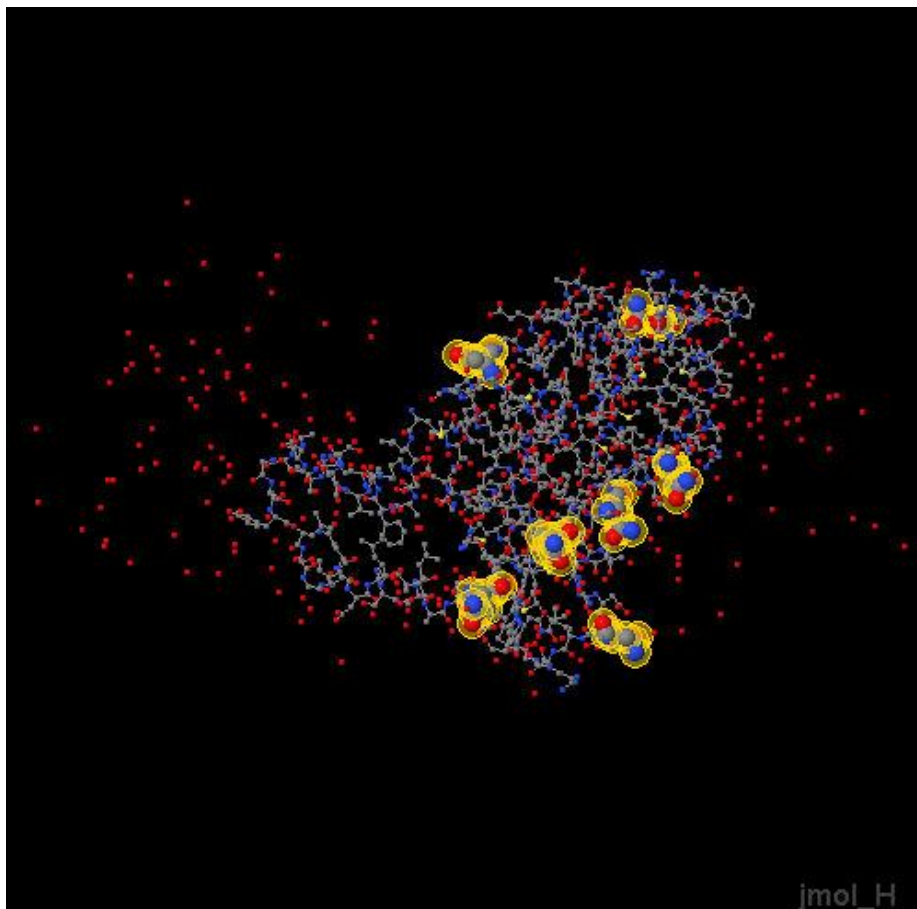


Figure 14: Μεγέθυνση επιλεγμένων αμινοξέων

7.1.2. Η λειτουργία εισαγωγής σχολίου (comment mode)

Με την λειτουργία εισαγωγής σχολίου δίνεται η δυνατότητα στο χρήστη να εισάγει οποιοδήποτε σχόλιο επιθυμεί σε άτομο της επιλογής του .Για την ενεργοποίηση της λειτουργίας αυτής επιλέγεται η επιλογή addComment που βρίσκεται στο υπομενού Visual του κυρίως μενού VisualTools στην μπάρα μενού του Jmol.Μετά την ενεργοποίηση της λειτουργίας εισαγωγής σχολίου ο χρήστης μπορεί να επιλέξει οποιοδήποτε άτομο της δομής ενός μορίου που παρίσταται στην σκηνή του Jmol,η επιλογή του ατόμου γίνεται τοποθετώντας τον κέρσορα του ποντικιού πάνω στο άτομο και κάνοντας κλικ με το αριστερό πλήκτρο του ποντικιού τότε εμφανίζεται το γραφικό σχόλιου πάνω από το συγκεκριμένο άτομο που έχουμε επιλέξει. Στην εικόνα παρακάτω φαίνεται η εισαγωγή σχολίου σε άτομο.

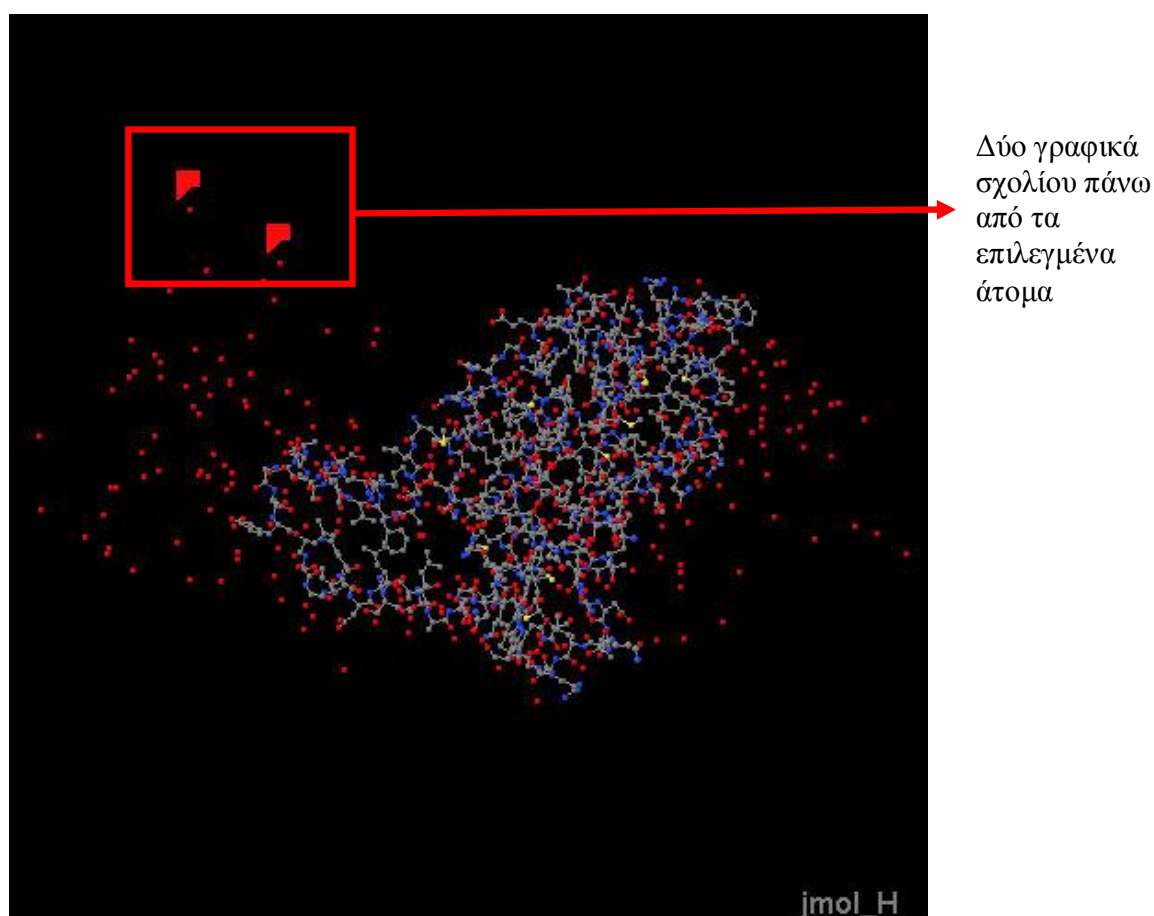


Figure 15: Εισαγωγή γραφικών σχολίου

Κάνοντας κλικ με το ποντίκι πάνω στο γραφικό που έχει δημιουργηθεί για να υποδείξει την ύπαρξη σχολίου στο επιλεγμένο άτομο ,εμφανίζεται το παράθυρο εισαγωγής σχολίου που περιέχει μια περιοχή (JTextArea) που μπορούμε να εισάγουμε το σχόλιο που επιθυμούμε και ένα κουμπί επικύρωσης το οποίο πατώντας το επιτυγχάνουμε την προσωρινή (όσο η εφαρμογή του Jmol είναι σε λειτουργία)αποθήκευση του σχολίου που εισήγαμε .Η εφαρμογή μας δίνει την

δυνατότητα αφαίρεσης του σχολίου από ένα άτομο .Η διαδικασία αφαίρεσης επιτυγχάνεται τοποθετώντας τον κέρσορα του ποντικιού πάνω στο άτομο που έχει εισαχθεί το σχόλιο και κάνοντας κλικ με το αριστερό πλήκτρο του ποντικιού.

Το παράθυρο εισαγωγής του σχολίου περιέχει στην κεφαλίδα του κάποια χρήσιμα στοιχεία που μας δίνουν πληροφορίες για το συγκεκριμένο άτομο που έχει επιλεγεί ,το πρώτο από αυτά τα στοιχεία δηλώνει το είδος του αμινοξέου που ανήκει το άτομο ,το δεύτερο κατά σειρά στοιχείο δηλώνει τον αριθμό ακολουθίας του αμινοξέου ,το τρίτο κατά σειρά στοιχείο δηλώνει το προσδιοριστικό γράμμα της πολυπεπτιδικής αλυσίδας που ανήκει το άτομο ,το τέταρτο στοιχείο δηλώνει το όνομα του ατόμου(το πρώτο γράμμα του ονόματος ενός ατόμου δηλώνει το χημικό είδος του ατόμου ,αν πχ το όνομα είναι CGI το C δηλώνει ότι το άτομο είναι άνθρακας(carbon)).Παρακάτω στην εικόνα φαίνεται ένα παράθυρο εισαγωγής σχολίου .

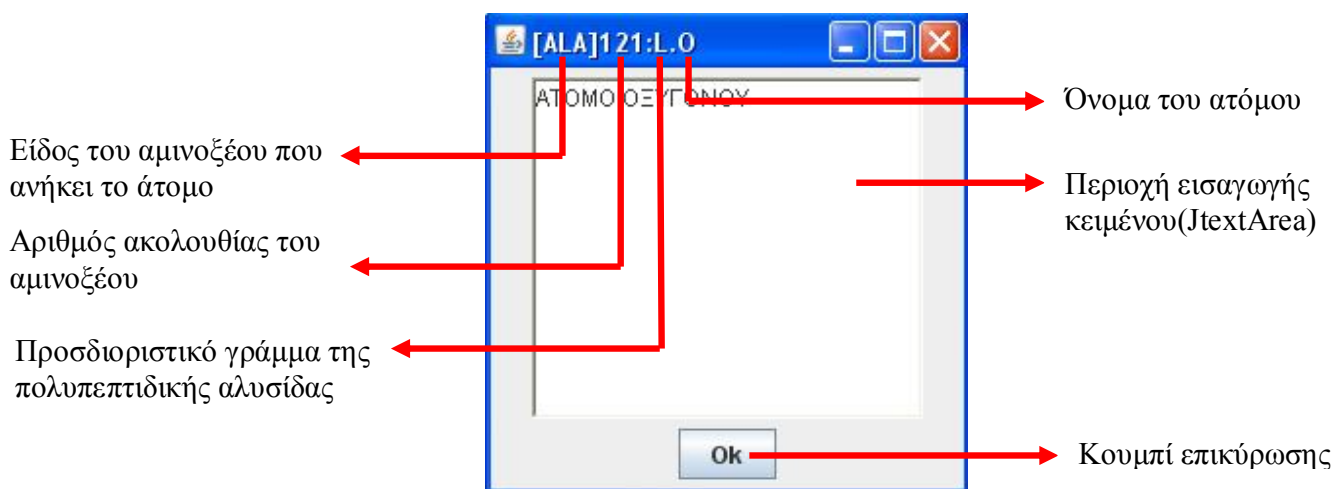


Figure 16: Περιβάλλον εισαγωγής σχολίου

7.1.3. Η εφαρμογή TreeView

Η εφαρμογή TreeView αναπαριστά τα επιμέρους συστατικά από τα οποία αποτελείται το μόριο που αναπαρίσταται γραφικά μέσω του προγράμματος Jmol. Η εφαρμογή χρησιμοποιεί την κλάση JTree της Java, η κλάση JTree ουσιαστικά τοποθετεί τα συστατικά με την ιεραρχική μορφή ενός δέντρου και αναπαριστά τα δεδομένα ιεραρχικά. Τα επιμέρους συστατικά που αποτελούν ένα μόριο αναπαρίστανται από την εφαρμογή TreeView σαν κόμβοι ,οι κόμβοι μπορεί να είναι οι πολυπεπτιδικές αλυσίδες ,τα αμινοξέα και τα άτομα. Ο κάθε κόμβος μπορεί να έχει έναν ανώτερο κόμβο που ονομάζεται «πατέρας» του κομβού αλλά πολλούς χαμηλότερους ιεραρχικά κόμβους που ονομάζονται «απόγονοι» του ανώτερου κόμβου . Η ιεραρχική αυτή δομή τελειώνει στον χαμηλότερο ιεραρχικά κόμβο που είναι το μεμονωμένο άτομο ,ένας κόμβος που αναπαριστά ένα μεμονωμένο άτομο δεν έχει «απογόνους» .Ο κύριος κόμβος της ιεραρχικής αυτής δομής θα είναι ο κόμβος που θα αναπαριστά το ίδιο το μόριο.

Κάνοντας διπλό κλικ σε ένα κόμβο εμφανίζονται οι επόμενοι κόμβοι της ιεραρχίας. Αν γίνει πάλι διπλό κλικ στον ίδιο κόμβο οι απόγονοι του κόμβου σταματούν να είναι ορατοί.

Επιλέγοντας κάθε φορά έναν κόμβο επισημαίνεται και στην σκηνή του Jmol τα άτομα που συγκροτούν τον κόμβο αυτόν(figure 18),αν επιλεγεί ο κόμβος που αντιπροσωπεύει το μόριο τότε

θα επισημανθούν στην σκηνή όλα τα άτομα που αποτελούν το μόριο ,αν επιλεγεί ο κόμβος που αντιπροσωπεύει ένα μεμονομένο άτομο τότε θα στην σκηνή θα επισημανθεί το συγκεκριμένο άτομο . Παρακάτω στην εικόνα παρουσιάζεται το περιβάλλον της εφαρμογής TreeView .

Όνομα εφαρμογής

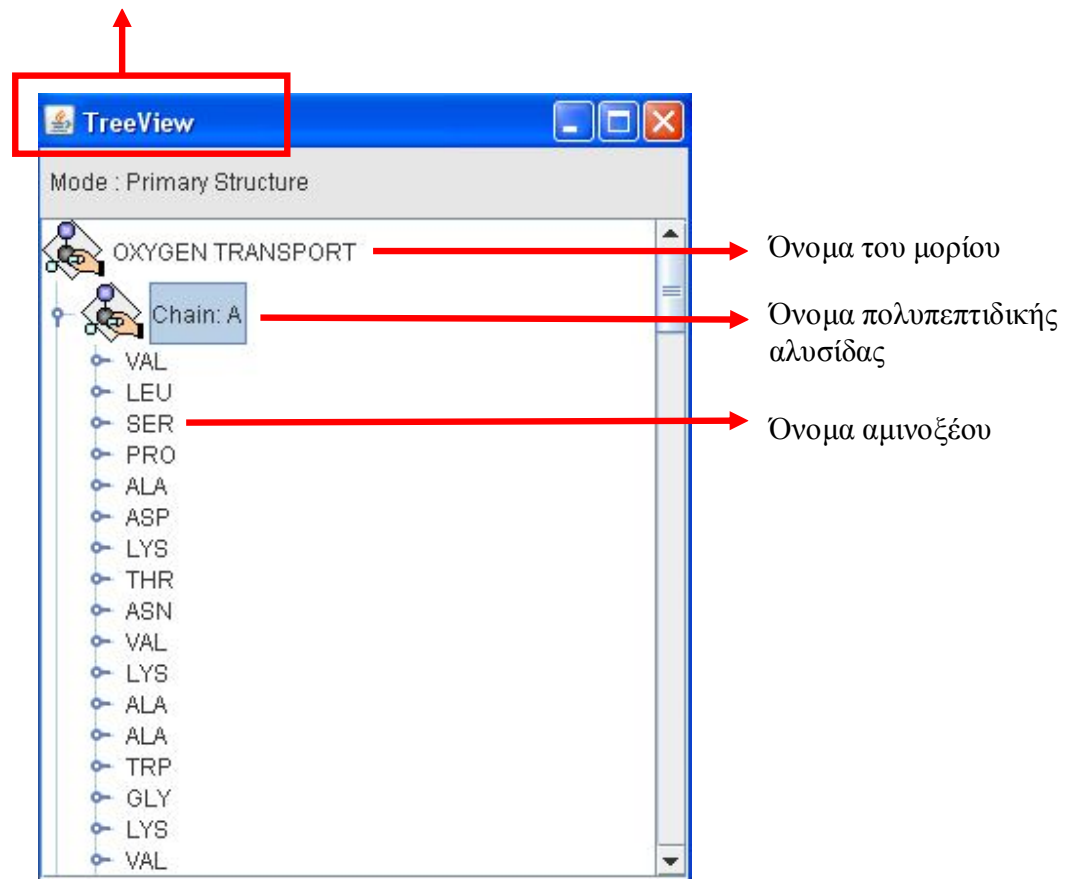


Figure 17: Η εφαρμογή TreeView

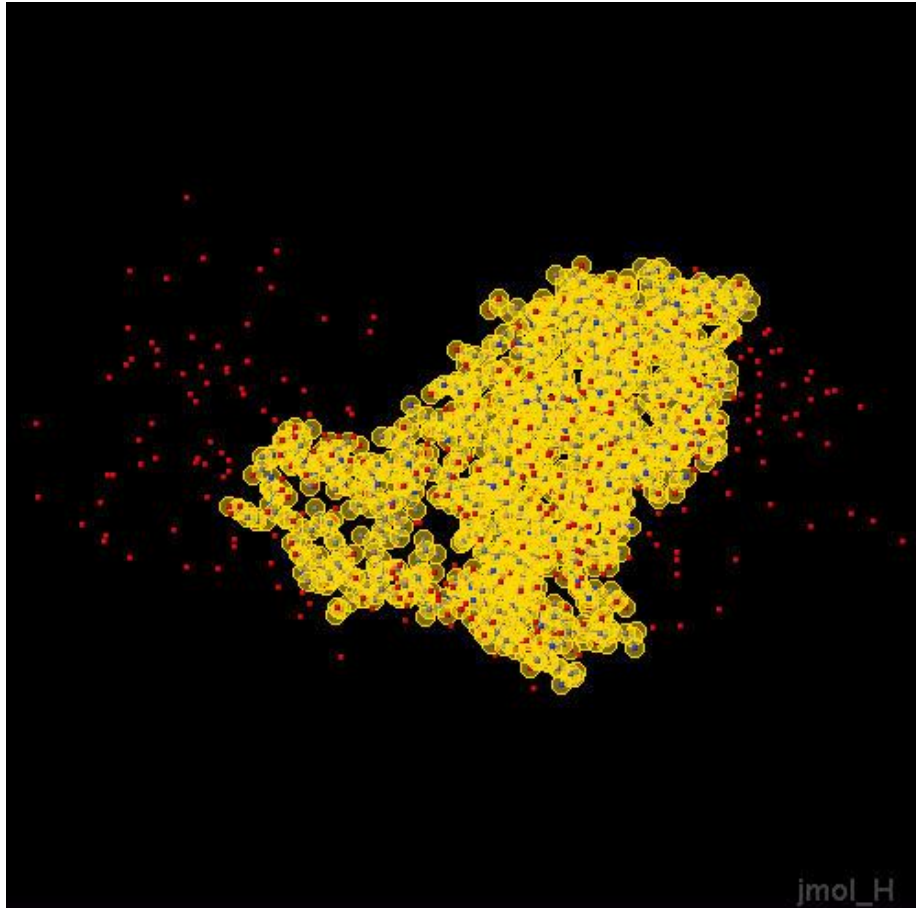


Figure 18: Επισήμανση της επιλεγμένης βιολογικής δομής στην σκηνή του Jmol

8. Συμπεράσματα

Σημείο αναφοράς της συγκεκριμένης πτυχιακής ήταν η αντιπροσώπευση δεδομένων βιοπληροφορίας μέσα από την εφαρμογή που δημιουργήθηκε ,τα δεδομένα βιοπληροφορίας αφορούν την δομή των βιολογικών μακρομορίων που απεικονίζονται στο πρόγραμμα Jmol.Σαν συνέπεια της δημιουργίας της εφαρμογής αυτής και την αντιπροσώπευση δεδομένων μέσα από αυτή ήταν η βελτίωση της διεπαφής ανάμεσα στον χρήστη και το περιβάλλον του προγράμματος αναπαράστασης βιολογικών μορίων και δομών Jmol.Η εφαρμογή σχεδιάστηκε αναπτύχθηκε και ενσωματώθηκε στο Jmol ,περιλάμβανε διάφορα περιβάλλοντα διεπαφής που επιτρέπουν στον χρήστη ,είτε είναι απλός χρήστης ,είτε περισσότερο εξειδικευμένος στον τομέα που πραγματεύονται τα προγράμματα αναπαράστασης βιολογικών δομών ,να αλληλεπιδρά με την σκηνή που αναπαρίσταται η βιολογική δομή στο Jmol.

Η αλληλεπίδραση βοηθά τον χρήστη στην κατανόηση της πληροφορίας που παρέχεται και στην συγκεκριμένη περίπτωση μέσα από το περιβάλλον της εφαρμογής SequenceView όπου παρουσιάζεται η ακολουθία αμινοξέων του κάθε μορίου που αναπαρίσταται την δεδομένη χρονική στιγμή στην σκηνή του Jmol, επιλέγοντας ο χρήστης ένα αμινοξέο από την ακολουθία μπορεί να δει και την αναπαράσταση του στην σκηνή του Jmol.Στο περιβάλλον της εφαρμογής TreeView παρουσιάζεται η πρωτοταγής δομή ενός μορίου σε μορφή δέντρου ,κάθε κόμβος του δέντρου περιγράφει και μια δομή του μορίου ,επιλέγοντας ένα κόμβο αναπαρίσταται στην σκηνή η δομή που αντιπροσωπεύει ο συγκεκριμένος κόμβος .Η τελευταία λειτουργία δίνει την δυνατότητα στον χρήστη να εισάγει σχόλιο σε οποιοδήποτε άτομο του μορίου που αναπαρίσταται μια δεδομένη χρονική στιγμή.

Μέσα από την μελέτη του προγράμματος Jmol και άλλων σχετικών προγραμμάτων αλλά και γενικότερα του συγκεκριμένου γνωστικού αντικείμενου .δηλαδή της αναπαράστασης βιολογικών δομών με υπολογιστή συμπαιρένουμε ότι ο συγκεκριμένος τομέας είναι ήδη ιδιαίτερα ανεπτυγμένος αλλά υπάρχουν περιθώρια και περαιτέρω βελτίωσης όσον αφορά της εφαρμογές που δραστηριοποιούνται σε αυτόν τον τομέα ,βελτίωση γραφικών αναπαράστασης και αλγορίθμων ,πιο φιλικά περιβάλλοντα στο χρήστη ,περισσότερη και καλύτερης ποιότητας ανάλυση της πληροφορίας που παρέχεται.

Όσον αφορά την εφαρμογή που δημιουργήθηκε θα μπορούσε να βελτιωθεί περισσότερο σε δυο συγκεκριμένα σημεία τουλάχιστον .Θα μπορούσε να δημιουργηθούν εκτός την λειτουργία της εισαγωγής σχολίου σε κάθε μόριο και άλλες λειτουργίες που θα διευκόλυναν τον χρήστη ,όπως η εισαγωγή βίντεο ,η εισαγωγή εικόνας ,η κειμένου μορφής pdf.Στην εφαρμογή TreeView θα μπορούσε να δημιουργηθεί εκτός την αναπαράσταση της πρωτοταγούς δομής ενός μορίου σε δενδροειδή μορφή και η δευτεροταγής δομή και οι υπόλοιπες ανώτερου επιπέδου δομές .Βέβαια οι συγκεκριμένες βελτιώσεις που αναφέρθηκαν παραπάνω θα ήταν αντικείμενο μιας άλλης πτυχιακής εργασίας και για να επιτευχθούν θα απαιτούσαν περισσότερες γνώσεις όσον αφορά το κομμάτι της βιολογίας για να είναι μια τέτοια εργασία αξιόπιστη.

9. Βιβλιογραφία

Μέρος του υλικού που παρουσιάστηκε στην συγκεκριμένη πτυχιακή προέκυψε από τα παρακάτω συγγράμματα.

[1]Τμήμα επιστήμης υπολογιστών του πανεπιστημίου Κρήτης .Ηλεκτρονικά βιβλία Java ,Java Swing tutorial,Java Awt tutorial .

[2]Understanding molecular simulation: from algorithms to applications ,Dan Freckle, Berend Smit – 2002

[3] Molecular biology and biotechnology: a comprehensive desk reference
Robert A. Meyers – 1995

[4]Research in Computational Molecular Biology: 13th Annual International ...
Serafim Batzoglou

[5]Java Swing,Marc Loy, Robert Eckstein - 2002

[6]The definitive guide to Java Swing,John Zukowski

[7]Java AWT reference,John Zukowski

[8]Effective Java: programming language guide,Joshua Bloch – 2001

[9]UML for Java programmers,Robert C. Martin – 2003

[10]Java 3D programming,Daniel Selman - 2002

[11]Computer graphics using Java 2D and 3D ,Hong Zhang, Y. Daniel Liang – 2007

[12]How to Use Jmol to Study and Present Molecular Structures, Τόμος 1
Angel Herrez – 2008

[13]The Java native interface: programmer's guide and specification
Sheng Liang – 1999

Παρακάτω αναφέρονται σελίδες του διαδικτύου που αντλήθηκε υλικό για την διεκπεραίωση της πτυχιακής.

Jmol wiki->http://wiki.jmol.org/index.php/Main_Page

Jmol-><http://jmol.sourceforge.net/>

Pymol-><http://www.pymol.org/>

A tutorial on using-> http://137.189.50.96/kbwong/teaching/pymol/pymol_tutorial.html

10. Παραρτήματα

10.1. Παράρτημα Α: Οι βάσεις δεδομένων βιοπληροφορίας

Οι βάσεις δεδομένων ακολουθιών και δομών DNA και πρωτεϊνών αποτελούν το χαρακτηριστικότερο δείγμα της αλματώδους εξέλιξης που έχει επιτελεστεί τα τελευταία χρόνια στη Βιολογία. Όταν πρωτοξεκίνησε η δημιουργία τους ο όγκος της πληροφορίας ήταν τόσο μικρός που και ένας μικρός αριθμός ερευνητών αρκούσε για την συντήρηση και για την ανανέωση των βάσεων αυτών. Αν κάποιος ερευνητής ενδιαφέρονταν να έχει πρόσβαση στις εγγραφές της βάσης επικοινωνούσε με τους επιστημονικούς υπευθύνους και εκείνοι του έστελναν με συμβατικό ταχυδρομείο όλη τη βάση η οποία αρκούσε να αποθηκευτεί ακόμη και σε μερικές δισκέτες ή μια μαγνητοταινία. Την τελευταία δεκαετία όμως η τεχνολογική εξέλιξη βοήθησε στη διεκπεραίωση μεγάλου όγκου πειραματικής εργασίας η οποία σε συνάρτηση με τον διαρκή προσδιορισμό γωνιωμάτων διαφόρων οργανισμών αύξησε τον όγκο της πληροφορίας στο επίπεδο της ακολουθίας και όχι μόνο, σε δυσθεώρητα μεγέθη. Οι βάσεις πλέον δεν περιέχουν απλώς πολλά δεδομένα αλλά και η διαδικασία ανανέωσης τους είναι απαραίτητα καθημερινή υπόθεση. Πλέον η συντήρηση μιας βάσης απαιτεί ένα πολυάριθμο επιτελείο επιστημόνων οι οποίοι ασχολούνται αποκλειστικά με το σχολιασμό (annotation) των νεοεισερχόμενων δεδομένων καθώς και με τη διόρθωση λαθών των ήδη υπαρχόντων. Χαρακτηριστικά παραδείγματα αποτελούν η βάση πρωτεϊνικών ακολουθιών SWISS-PROT που περιέχει 104948 ακολουθίες (Rel. 40.9 - Ιανουάριος 2002), ενώ η EMBL Nucleotide Sequence Database που περιέχει νουκλεοτιδικές αλληλουχίες έχει 14366182 εγγραφές (Rel. 69 - Δεκέμβριος 2001). Η πρόσβαση στις βάσεις αυτές είναι πλέον εύκολη μέσω της χρήσης του Διαδικτύου. Ο χρήστης μπορεί να επισκεφτεί την ιστοσελίδα που διατηρείται από τους υπευθύνους της βάσης και να κάνει αναζητήσεις αποθηκεύοντας στον υπολογιστή του δεδομένα του άμεσου ενδιαφέροντος του. Παράλληλα έχουν δημιουργηθεί και μια σειρά από βάσεις που αποσκοπούν στην ταξινόμηση της πληροφορίας στο επίπεδο της ακολουθίας και της δομής προκειμένου να οργανωθεί η πληροφορία και να εξαχθούν συμπεράσματα για την βιολογική τους σημασία.

Παρακάτω παρατίθενται οι κυριότερες κατηγορίες βάσεων νουκλεϊνικών οξέων και πρωτεϊνών και οι κυριότεροι αντιπρόσωποι τους επιγραμματικά.

A. Βάσεις δεδομένων νουκλεοτιδικών αλληλουχιών.

1. GENBANK
2. EMBL-Bank
3. DDJB

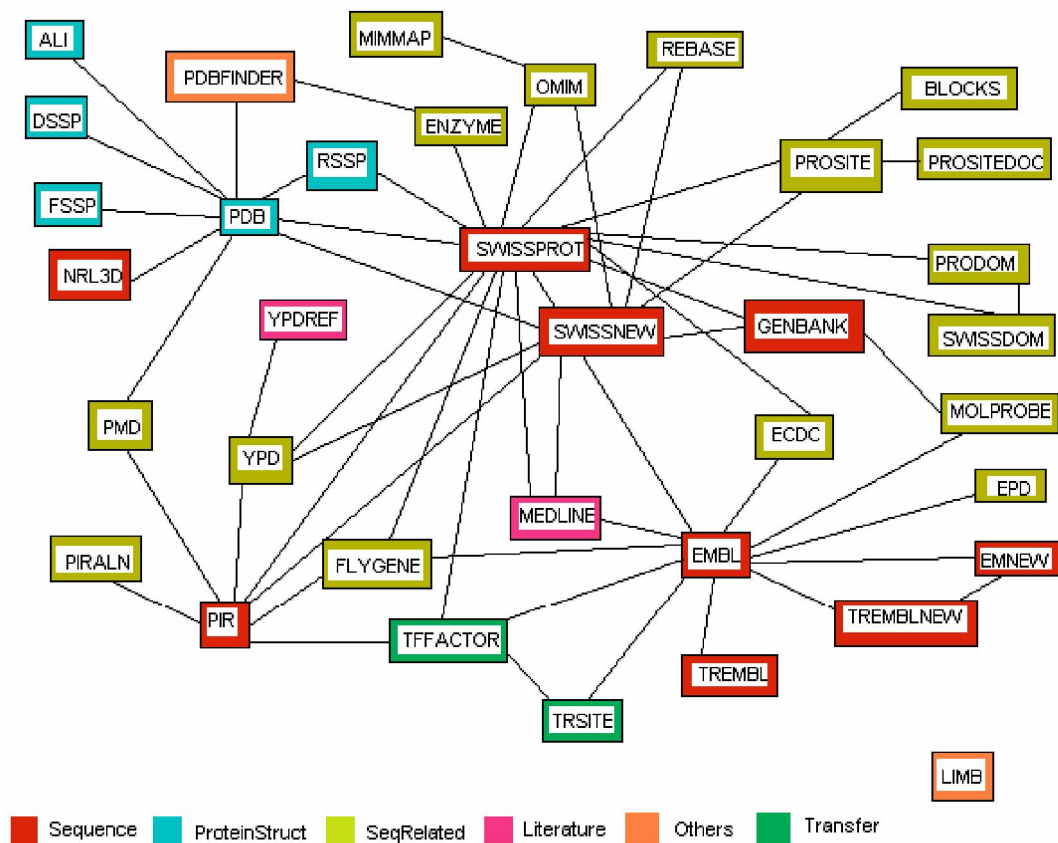
B. Βάσεις δεδομένων πρωτεϊνικών ακολουθιών και βάσεις για την ανάλυση ακολουθιών.

1. SWISS-PROT
2. Protein Information Resource
3. PROSITE

Γ. Βάσεις δεδομένων δομικής βιολογίας.

1. Protein Data Bank
2. CATH
3. SCOP

Μερικές βάσεις δεδομένων που είναι διαθέσιμες στο διαδίκτυο.



10.2. Παράρτημα Β: Τα αρχεία PDB

Σε αυτήν την ενότητα θα αναφερθούμε και θα αναλύσουμε την δομή των αρχείων του είδους PDB που παρέχονται από την δικτυακή βάση δεδομένων Protein Data Bank και τα χρησιμοποιούμε στην εφαρμογή που δημιουργήσαμε για να αντλήσουμε την πληροφορία που αναπαρίσταται στα περιβάλλοντα της εφαρμογής και είναι επεξεργάσιμη από το χρήστη.

Τα αρχεία της Protein Data Bank με επέκταση pdb είναι αρχεία που περιέχουν δεδομένα σχετικά με την δομή των μορίων στο χώρο, τις ατομικές συντεταγμένες, την αρχική και δευτεροβάθμια δομή των μορίων, διάφορες άλλες πληροφορίες που έχουν σχέση με την δομή που περιγράφουν και το κρυσταλλογραφικό στοιχείο της δομής NMR. Τα συγκεκριμένα αρχεία διαβάζονται από τα προγράμματα δημιουργίας τρισδιάστατων μοντέλων μορίων, όπως το Jmol για να αναπαραστήσουν τα μόρια που περιγράφονται στα αρχεία γραφικά. Παρακάτω θα αναλυθεί η δομή αυτών των αρχείων με λεπτομέρεια.

Το αρχείο PDB αποτελείται από γραμμές πληροφοριών. Κάθε γραμμή πληροφοριών στο αρχείο καλείται **εγγραφή**. Ένα αρχείο PDB περιέχει γενικά διάφορες εγγραφές, που τακτοποιούνται σε μια συγκεκριμένη σειρά ώστε να περιγραφεί μια δομή.

ΤΥΠΟΣ ΕΓΓΡΑΦΗΣ	ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ
HEADER	Η πρώτη γραμμή κάθε καταχώρησης. Περιέχει τον Κωδικό PDB ID, μια βασική ταξινόμηση καθώς και την ημερομηνία κατάθεσης.
TITLE	Περιγραφή της Δομής που αναφέρεται στην καταχώρηση.
COMPND	Περιγραφή των 'Μακρομοριακών Περιεχομένων'.
KEYWDS	'Λέξεις Κλειδιά' που περιγράφουν το Μακρομόριο.
SOURCE	Βιολογική προέλευση του μακρομορίου.
EXPDTA	Η πειραματική τεχνική που χρησιμοποιήθηκε για τη λύση της δομής.
AUTHOR	Κατάλογος των ερευνητών που εργάστηκαν στον καθορισμό της δομής.
HYDBND	Καθορισμός των θέσεων Υδρογονικών Δεσμών.
SEQRES	Η Βιολογική Ακολουθία. Για πρωτεΐνες ακολουθείται ο κωδικός των τριών γραμμάτων (π.χ. ALA για Αλανίνη).
HELIX	Καθορισμός περιοχών με ελικοειδή δομή.
SHEET	Καθορισμός περιοχών με εκτεταμένη δομή.
CRYST1	Παράμετροι Μοναδιαίας Κυψελίδας, Ομάδα Συμμετρίας Χώρου
ORIGXn(n=1,2,ή 3)	Πίνακας Μετατροπής από σύστημα ορθογωνίων συντεταγμένων στις συντεταγμένες που κατατέθηκαν αρχικά.
SCALEn(n=1,2,ή 3)	Πίνακας Μετατροπής από σύστημα ορθογωνίων συντεταγμένων σε κρυσταλλογραφικές συντεταγμένες.
ATOM	Εγγραφές των ατομικών συντεταγμένων.
HETATM	Εγγραφές των ατομικών συντεταγμένων για τα Ετεροάτομα.
TER	Τερματισμός μιας Αλυσίδας.
END	Τελευταία εγγραφή κάθε καταχώρησης.
SSBOND	Προσδιορισμός των δεσμών δισουλφιδίου.
REVDAT	Ημερομηνία αναθεώρησης και σχετικές πληροφορίες.
SEQADV	Προσδιορισμός των συγκρούσεων μεταξύ του PDB και της ονομασμένης βάσης δεδομένων ακολουθίας.
SIGATM	Σταθερές αποκλίσεις των ατομικών παραμέτρων.
SIGUIJ	Σταθερές αποκλίσεις των ανισότροπων παραγόντων θερμοκρασίας.
SITE	Προσδιορισμός των ομάδων που περιλαμβάνουν τις σημαντικές περιοχές.
LINK	Προσδιορισμός των δεσμών διά-υπολειμμάτων.

Table 7: Επεξήγηση εγγραφών

10.2.1. Ανάλυση πεδίων εγγραφών

Ο πίνακας παρακάτω αναλύει της εγγραφές που περιγράφουν το ισότιμο τμήμα του μορίου ,το ισότιμο τμήμα περιλαμβάνει της εγγραφές των ατόμων των ετεροάτομων και των τερματισμό της αλυσίδας .Η περιγραφή που γίνεται στον πίνακα παρακάτω αναλύει τα πεδία των εγγραφών κατά σειρά από αριστερά προς τα δεξιά .Στο πεδίο του πίνακα που ονομάζεται στήλη αναφέρονται οι στήλες σε ένα pdb αρχείο που περιέχουν κάθε πεδίο μιας εγγραφής από αριστερά προς τα δεξιά .

Ο πίνακας παρακάτω αναλύει της εγγραφές που περιγράφουν την πρωτεϊνικά δευτεροβάθμια δομή,η πρωτεϊνικά δευτεροβάθμια δομή περιλαμβάνει της εγγραφές της έλικας του φύλλου και του δεσμού .

ΠΡΩΤΕΪΝΙΚΗ ΤΡΑΠΕΖΑ ΔΕΔΟΜΕΝΩΝ (PROTEIN DATA BANK) ΙΣΟΤΙΜΟ ΤΜΗΜΑ (COORDINATE SECTION)			
Τύπος εγγραφής	Στήλη	Δεδομένα(πεδία εγγραφών)	Τύπος δεδομένων
Ατομο(atom)	7-11 [#]	Αύξων αριθμός ατόμων	Ακέραιος (integer)
	13-16	Όνομα ατόμων	Χαρακτήρες (character)
	18-20 ^{\$}	Όνομα υπολειμμάτων	Χαρακτήρες (character)
	22	Προσδιοριστικό αλυσίδων	Χαρακτήρα (character)
	23-26	Αριθμός ακολουθίας υπολειμμάτων	Ακέραιος (integer)
	31-38	X ορθογώνια συντεταγμένη	Πραγματικός(real)
	39-46	Y ορθογώνια συντεταγμένη	Πραγματικός(real)
	47-54	Z ορθογώνια συντεταγμένη	Πραγματικός(real)
	55-60	Κατοχή	Πραγματικός(real)
	61-66	Παράγοντας θερμοκρασίας	Πραγματικός(real)
	77-78	Σύμβολο στοιχείων	Χαρακτήρα (character)
Ετεροάτομα (hetatm)	7-80	Ίδιες εγγραφές όπως το άτομο	
Τερματισμός αλυσίδας (ter)	7-11 [#]	Αύξων αριθμός αλυσίδας	Ακέραιος (integer)
	18-20 ^{\$}	Όνομα υπολειμμάτων	Χαρακτήρες (character)
	23-26	Αριθμός ακολουθίας υπολειμμάτων	Ακέραιος (integer)

Table 8: Επεξήγηση εγγραφών ισότιμου τμήματος

ΠΡΩΤΕΪΝΙΚΑ ΔΕΥΤΕΡΟΒΑΘΜΙΑ ΔΟΜΗ			
Τύπος εγγραφής	Στήλη	Δεδομένα(πεδία εγγραφώ	Τύπος δεδομένων
Έλικα (helix)	8-10	Αύξων αριθμός ελικας	Ακέραιος (integer)
	12-14	Προσδιοριστικό ελικας	Ακέραιος (integer)
	16-18	Αρχικό όνομα υπολειμμάτων	Χαρακτήρες (character)
	20	Προσδιοριστικό αλυσίδων	Χαρακτήρες (character)
	22-25	Αριθμός ακολουθίας υπολειμμάτων	Ακέραιος (integer)
	28-30	Τελικό όνομα υπολειμμάτων	Χαρακτήρες (character)
	32	Προσδιοριστικό αλυσίδων	Χαρακτήρες (character)
	34-37	Αριθμός ακολουθίας υπολειμμάτων	Ακέραιος (integer)
	39-40	Τυπος ελικας	Ακέραιος (integer)
	72-76	Μήκος του έλικα	Ακέραιος (integer)
Φύλλο (sheet)	8-10	Αριθμός σκελών	Ακέραιος (integer)
	12-14	Προσδιοριστικό φύλλων	Χαρακτήρες (character)
	15-16	Αριθμός σκελών (στο τρέχον φύλλο)	Ακέραιος (integer)
	18-20	Αρχικό όνομα υπολειμμάτων	Χαρακτήρες (character)
	22	Προσδιοριστικό αλυσίδων	Χαρακτηρες (character)
	23-26	Αριθμός ακολουθίας υπολειμμάτων	Ακέραιος (integer)
	29-31	Τελικό όνομα υπολειμμάτων	Χαρακτήρες (character)
	33	Προσδιοριστικό αλυσίδων	Χαρακτήρες (character)
	34-37	Αριθμός ακολουθίας υπολειμμάτων	Ακέραιος (integer)
	39-40	Αίσθηση σκελών όσον αφορά προηγούμενο	Χαρακτήρες (character)
Οι ακόλουθοι τομείς προσδιορίζουν δύο άτομα που περιλαμβάνονται σε έναν δεσμό υδρογόνου, ο πρώτος στο τρέχον σκέλος και ο δεύτερος στο προηγούμενο σκέλος.			
	42-45	Όνομα ατόμου που ανηκει στον δεσμο υδρογονου	Χαρακτήρες (character)
	46-48	Όνομα υπολειμμάτων	Χαρακτήρες (character)

	50	Προσδιοριστικό αλυσίδων	Χαρακτήρες (character)
	51-54	Αριθμός ακολουθίας υπολειμμάτων	Ακέραιος (integer)
	55	Κώδικας για τις εισαγωγές των υπολειμμάτων	Χαρακτήρες (character)
	57-60	Όνομα ατόμου που ανηκει δεσμο υδρογονου	Χαρακτήρες (character)
	61-63	Όνομα υπολειμμάτων	Χαρακτήρες (character)
	65	Προσδιοριστικό αλυσίδων	Χαρακτήρες (character)
	66-69	Αριθμός ακολουθίας υπολειμμάτων	Ακέραιος (integer)
	70	Κώδικας για τις εισαγωγές των υπολειμμάτων	Χαρακτήρες (character)
Δεσμός(sbond)	8-10	Αύξων αριθμός	Ακέραιος (integer)
	12-14	Όνομα υπολειμμάτων	Χαρακτήρες (character)
	16	Προσδιοριστικό αλυσίδων	Χαρακτήρες (character)
	18-21	Αριθμός ακολουθίας υπολειμμάτων	Ακέραιος (integer)
	26-28	Όνομα υπολειμμάτων	Χαρακτήρες (character)
	30	Προσδιοριστικό αλυσίδων	Χαρακτήρες (character)
	32-35	Αριθμός ακολουθίας υπολειμμάτων	Ακέραιος (integer)
	60-65	Χειριστής συμμετρίας για το πρώτο υπόλειμμα	Ακέραιος (integer)
	67-72	Χειριστής συμμετρίας για το δεύτερο υπόλειμμα	Ακέραιος (integer)
	74-78	Μήκος του δεσμού δισουλφιδίου	Πραγματικός(real)

Table 9: Επεξήγηση εγγραφών δευτεροβάθμιας δομής

10.3. Παράρτημα Γ: Οι βιολογικές δομές που συνθέτουν ένα μόριο

Σε αυτό το κεφάλαιο αναφέρεται ιδιαίτερα η πληροφορία και τα δεδομένα που πραγματεύεται η εφαρμογή που έχει δημιουργηθεί στα πλαίσια της παρούσας πτυχιακής. Συγκεκριμένα αναφέρονται οι βιολογικές δομές που αποτελούν την πρωτοταγή δομή του μορίου.

10.3.1. Οι βιολογικές δομές

Τα προγράμματα απεικόνισης βιολογικών δομών, καθώς και το Jmol που ανήκει σε αυτήν την κατηγορία προγραμμάτων αναπαριστούν στο περιβάλλον τους διάφορες βιολογικές δομές, η ανώτερη σε μέγεθος δομή που αναπαρίσταται από αυτά τα προγράμματα είναι το βιολογικό μακρομόριο και η κατώτερη σε μέγεθος είναι το άτομο και ο ατομικός δεσμός. Τα άτομα και οι δεσμοί που είναι η μικρότερη τάξη μεγέθους στην αναπαράσταση ενός μορίου με γραφικό τρόπο σχηματίζουν μεταξύ τους διάφορες πολυπλοκότερες δομές. Ένα μόριο αναφέρεται σαν μια βιολογική δομή και αποτελείται από πολλές άλλες βασικές βιολογικές δομές μικρότερης κλίμακας, ο συνδυασμός όλων αυτών των δομών σχηματίζουν την ενιαία δομή του μορίου.

Ένα μόριο από χημικής άποψης είναι το σύνολο δύο ή περισσοτέρων ατόμων τα οποία ενώνονται μεταξύ τους με χημικούς δεσμούς. Τα άτομα μπορεί να είναι του ίδιου ή διαφορετικών χημικών στοιχείων. Τώρα για να αναπαρασταθεί ένα μόριο γραφικά, τα άτομα που θα το αποτελούν θα απεικονίζονταν στο μοντέλο του μορίου σαν σφαίρες, η κάθε σφαίρα έχει μέγεθος σύμφωνα με την ακτίνα Van Der Waals του ατόμου που αντιπροσωπεύει. Όσα άτομα θα περιλάμβανε το μόριο στην πραγματικότητα, τόσες σφαίρες θα περιείχε η απεικόνιση του μοντέλου του μορίου, αυτές οι σφαίρες (άτομα) θα ενώνονται μεταξύ τους με κυλίνδρους που αυτοί οι κύλινδροι γραφικά αναπαριστούν τους χημικούς δεσμούς. Στην εικόνα παρακάτω αναπαρίσταται με γραφικό τρόπο ένα δίπολο μόριο νερού τα άτομα αναπαρίστανται με σφαίρες ενώ οι δεσμοί με κυλίνδρους, το συγκεκριμένο μόριο αποτελείται από ένα άτομο οξυγόνου και δυο άτομα υδρογόνου, το κάθε άτομο υδρογόνου συνδέεται με το άτομο οξυγόνου με έναν μόνο χημικό δεσμό. Η γωνιά που σχηματίζουν τα άτομα μεταξύ τους είναι 104.5 μοίρες και ονομάζεται (γωνία δεσμού).

Πρωτοταγής δομή

Η δομή των πρωτεϊνών αναφέρεται συνήθως στα πλαίσια τεσσάρων επιπέδων, της πρωτοταγής, δευτεροταγής, τριτοταγής και τεταρτοταγής δομής.

Η καθορισμένη ακολουθία αμινοξέων μιας πρωτεΐνης αποτελεί την πρωτοταγή της δομή, κάθε πρωτεΐνη μπορεί να αποτελείται από οποιαδήποτε ακολουθία αμινοξέων.

Η πρωτοταγής δομή της πρωτεΐνης περιλαμβάνει σε σειρά μεγέθους συστατικών από το μικρότερο στο μεγαλύτερο, τα άτομα, τα αμινοξέα και την αλυσίδα πολυπεπτιδίων. Όλα τα παραπάνω δομικά συστατικά συγκροτούν ένα ολοκληρωμένο μόριο την πρωτεΐνη. Τα άτομα συνθέτουν τα αμινοξέα, τα αμινοξέα με την σειρά τους συνθέτουν μια μεγαλύτερη δομή την αλυσίδα πολυπεπτιδίων και μια η περισσότερες αλυσίδες πολυπεπτιδίων συνθέτουν την πρωτεΐνη.

Το άτομο

Στη χημεία και τη φυσική, ένα άτομο (ετυμ. = άτμητο < α στερητ. + τέμνω) είναι το μικρότερο δυνατό σωματίδιο ενός χημικού στοιχείου το οποίο διατηρεί χημικές ιδιότητες. Η λέξη άτομο αρχικά εννοούσε το μικρότερο δυνατό άτμητο σωματίδιο, αλλά στη συνέχεια ο όρος αυτός απέκτησε ειδικό νόημα στην επιστήμη όταν βρέθηκε πως και τα άτομα αποτελούνται από μικρότερα υποατομικά σωματίδια. Τα περισσότερα άτομα αποτελούνται από τρεις τύπους υποατομικών σωματιδίων τα οποία διέπουν τις ιδιότητες των πρώτων:

- Ηλεκτρόνια, τα οποία έχουν αρνητικό φορτίο και έχουν τη μικρότερη μάζα από όλα τα τρία.
- Πρωτόνια, τα οποία έχουν θετικό φορτίο και έχουν μάζα περίπου 1836 φορές μεγαλύτερη από αυτή των ηλεκτρονίων και
- Νετρόνια, τα οποία δε φέρουν φορτίο και έχουν μάζα περίπου 1838 φορές μεγαλύτερη από αυτή των ηλεκτρονίων.

Τα πρωτόνια και τα νετρόνια είναι και τα δύο νουκλεόνια και σχηματίζουν τον συμπαγή ατομικό πυρήνα. Τα ηλεκτρόνια σχηματίζουν το πολύ μεγαλύτερης έκτασης ηλεκτρονικό νέφος το οποίο περιβάλλει τον πυρήνα.

Τα άτομα είναι οι θεμελιώδεις οικοδομικοί λίθοι της χημείας, και διατηρούνται στις χημικές αντιδράσεις. Τα άτομα είναι ικανά να σχηματίζουν μεταξύ τους δεσμούς δίνοντας μόρια και άλλου τύπου χημικά στοιχεία. Τα μόρια αποτελούνται από πολλαπλά άτομα. Για παράδειγμα, ένα μόριο νερού είναι συνδυασμός δύο ατόμων υδρογόνου και ενός ατόμου οξυγόνου. [3]

Το μόριο

Ως μόριο στην επιστήμη εννοείται το ελάχιστο συστατικό ενός στοιχείου ή μιας ένωσης (μιάς καθαρής ουσίας) που μπορεί να υπάρξει σε ελεύθερη κατάσταση, διατηρώντας τις φυσικές και χημικές ιδιότητες των μερών του στοιχείου ή της ένωσης που το αποτελούν. Από χημικής άποψης είναι το σύνολο δύο ή περισσότερων ατόμων τα οποία ενώνονται μεταξύ τους με χημικούς δεσμούς. Η απεικόνιση ενός μορίου είναι δυνατόν να γίνει με διαφορετικούς τρόπους. Ένας είναι με την δομική του παράσταση, στην οποία τα σύμβολα των ατόμων ενώνονται με ευθείες γραμμές που αντιπροσωπεύουν τους χημικούς δεσμούς ή πιο απλά γράφοντας την μοριακή του παράσταση. (Πηγή wikipedia). Παρακάτω στην εικόνα βλέπουμε την απεικόνιση του μορίου του νερού, βλέπουμε ότι αποτελείται από τρία άτομα (σφαίρες) και δυο δεσμούς (κύλινδροι). Το μόριο του νερού είναι ένα μόριο με πολύ απλή δομή, υπάρχουν στην φύση πολύ πιο σύνθετα μόρια. [3]

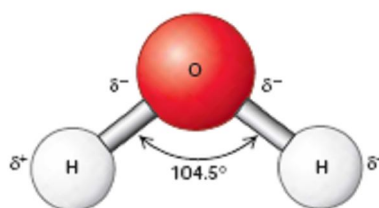


Figure 22 :Το μόριο του νερού

Η αλυσίδα πολυπεπτιδίων

Με τον όρο αλυσίδα πολυπεπτιδίων ή *πολυπεπτιδίου* (polypeptide chain) ονομάζεται οποιαδήποτε αλληλουχία αμινοξέων που ενώνονται με πεπτιδικούς δεσμούς προκειμένου να σχηματίσουν πρωτεΐνες. [3]

Οι πρωτεΐνες

Οι πρωτεΐνες είναι μεγάλα σύνθετα βιομόρια, με μοριακό βάρος από 10.000 μέχρι πάνω από 1 εκατομμύριο άτομα, αποτελούμενα από αμινοξέα, τα οποία ενώνονται μεταξύ τους με πεπτιδικούς δεσμούς σχηματίζοντας μια γραμμική αλυσίδα, καλούμενη αλυσίδα πολυπεπτιδίων. Όλες οι πρωτεΐνες περιέχουν άνθρακα, οξυγόνο και άζωτο και οι περισσότερες εξ αυτών και θείο.[3]

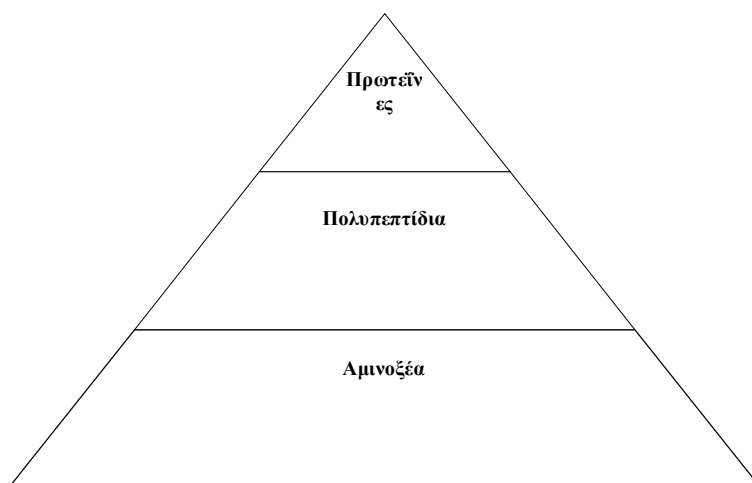


Figure 19: Σχήμα αναπαράστασης της δομής της πρωτεΐνης

Τα αμινοξέα

Αμινοξέα ονομάζονται οι χημικές ενώσεις που περιέχουν μία τουλάχιστον καρβονική ομάδα (από τα καρβονικά οξέα (RCOOH) και μία τουλάχιστον αμινομάδα ($-\text{NH}_2$). Τα αμινοξέα αποτελούν τα βασικά δομικά στοιχεία των πρωτεϊνών που καθορίζουν και τις χαρακτηριστικές ιδιότητές τους. Παρακάτω στον πίνακα αναφέρονται τα 20 πιο σημαντικά αμινοξέα που συνθέτουν της πρωτεΐνες των ζωντανών οργανισμών.[3]

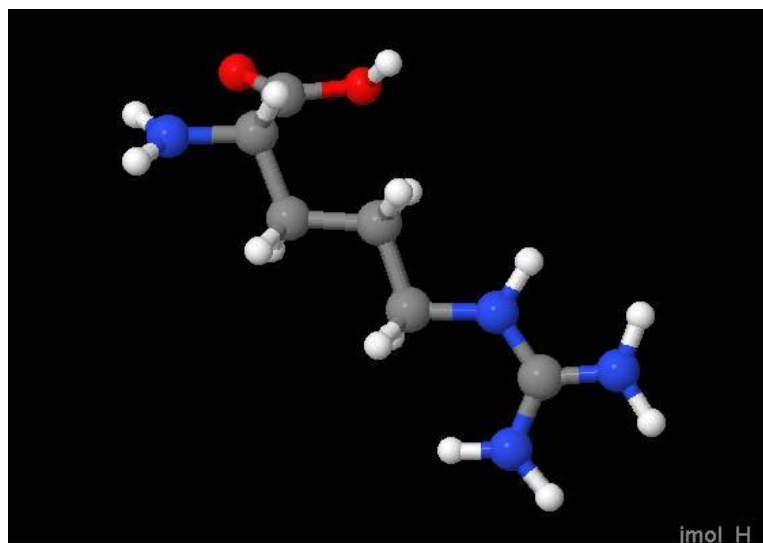


Figure 20: Το αμινοξύ Αργινίνη(Arg) σε αναπαράσταση στο Jmol

ΕΛΛΗΝΙΚΗ ΟΝΟΜΑΣΙΑ	ΔΙΕΘΝΗΣ ΣΥΝΤΜΗΣΗ	ΑΝΤΙΠΡΟΣΩΠΕΥΤΙΚΟ ΓΡΑΜΜΑ	ΑΝΤΙΠΡΟΣΩΠΕΥΤΙΚΟ ΧΡΩΜΑ(ΔΕΚΑΕΞΑΔΙΚΟ)
Αλανίνη	Ala	A	C8C8C8
Αργινίνη	Arg	R	145AFF
Ασπαραγίνη	Asn	N	00DCDC
Ασπαργανικό οξύ	Asp	D	E60A0A
Κυστεΐνη	Cys	C	E6E600
Γλουταμίνη	Gln	Q	00DCDC
Γλουταμικό οξύ	Glu	E	E60A0A
Γλυκίνη	Gly	G	EBEBEB

Ιστιδίνη	His	H	8282D2
Ισολευκίνη*	Ile	I	0F820F
Λευκίνη*	Leu	L	0F820F
Λυσίνη*	Lys	K	145AFF
Μεθιονίνη*	Met	M	E6E600
Φαινυλαλανίνη*	Phe	F	3232AA
Προλίνη	Pro	P	DC9682
Σερίνη	Ser	S	FA9600
Θρεονίνη*	Thr	T	FA9600
Θρυπτοφάνη*	Trp	W	B45AB4
Τυροσίνη	Tyr	Y	3232AA
Βαλίνη*	Val	V	0F820F

Table 10: Πίνακας αμινοξέων

Τα φερόμενα με αστερίσκο (*) είναι τα 8 βασικά αμινοξέα.

Ακολουθία αμινοξέων(Amino acid sequence)

Τα αμινοξέα αποτελούν τα βασικά δομικά στοιχεία των πρωτεϊνών που καθορίζουν και τις χαρακτηριστικές ιδιότητές τους. Οι πρωτεΐνες είναι μόρια που αποτελούνται από ένα ή περισσότερα πολυπεπίδια. Ένα πολυπεπίδιο αποτελείται από πολλά αμινοξέα .

Η ακολουθία αμινοξέων είναι μια γραμματοσειρά που κάθε γράμμα της είναι το αντιπροσωπευτικό γράμμα και ενός αμινοξέος, μια ακολουθία αμινοξέων μπορεί να αποτελείται από 124 αμινοξέα έως και μεγάλες ακολουθίες που αποτελούνται από 4.563 αμινοξέα.

Ουσιαστικά μια ολοκληρωμένη ακολουθία αμινοξέων περιγράφει όλα τα αμινοξέα που αποτελούν μια πρωτεΐνη. Αν και τα βασικά στοιχεία μιας ακολουθίας είναι μόνο 20 αμινοξέα, ο θεωρητικός αριθμός των διαφορετικών πρωτεϊνών είναι τεράστιος δεδομένου ότι για μία ακολουθία N αμινοξέων υπάρχουν 20^N συνδυασμοί που οδηγούν σε διαφορετικές πρωτεΐνες.

Η αλληλουχία των αμινοξέων σε μια ακολουθία μπορεί να δώσει πληροφορίες για το είδος της πρωτεΐνης, καθορίζει σε μεγάλο βαθμό την τρισδιάστατη δομή της πρωτεΐνης (δομή της πρωτεΐνης σημαίνει, είδη αμινοξέων και διάταξη τους στο χώρο). Παρακάτω ακολουθεί μια ακολουθία αμινοξέων όπως αναπαρίσταται στην εφαρμογή Sequence View.

Ακολουθία αμινοξέων: HSQGTFTSDY SKYLDSKKAQ EFVQWLMNT

Παρακάτω φαίνεται η ίδια ακολουθία όπως αναπαρίσταται σε ένα αρχείο της protein data bank

```
SEQRES   1      29  HIS  SER  GLN  GLY  THR  PHE  THR  SER  ASP  TYR  SER  LYS  TYR
SEQRES   2      29  LEU  ASP  SER  LYS  LYS  ALA  GLN  GLU  PHE  VAL  GLN  TRP  LEU
SEQRES   3      29  MET  ASN  THR
```

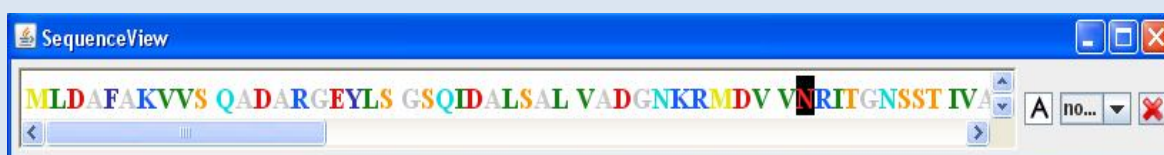
10.4. Παράρτημα Δ: Αναφορά στον κώδικα της εφαρμογής

Σε αυτό το κεφάλαιο αναφέρονται και αναλύονται ορισμένα σημαντικά τμήματα του κώδικα της εφαρμογής που δημιουργήθηκε.

10.4.1. Το περιβάλλον της εφαρμογής SequenceView

```
frame1=new JFrame("SequenceView");  
tmp = new JEditorPane();  
panel = new JPanel();
```

Απεικόνιση:



Στην πρώτη γραμμή του παραπάνω πλαισίου δημιουργείται το κύριο πλαίσιο(frame) που θα φιλοξενήσει τα υπόλοιπα συστατικά που θα αποτελέσουν το περιβάλλον του παραθύρου SequenceView.

Στην δεύτερη γραμμή δημιουργείται το συστατικό JEditorPane όπου θα φιλοξενήσει την ακολουθία αμινοξέων ,το συγκεκριμένο συστατικό επιλέχθηκε λόγω ότι έχει την δυνατότητα να αναπαράγει κώδικα της μορφής Html για την μορφοποίηση κειμένου.

Στην τρίτη γραμμή δημιουργείται το συστατικό JPanel όπου είναι το δεύτερο πιο κύριο συστατικό μετά το JFrame και φιλοξενεί τα υπόλοιπα συστατικά του παραθύρου εκτός το κύριο συστατικό JFrame.

```
SeqView sequenceView = new SeqView(molecule);  
data = sequenceView.createInfo();  
hold = "<html><font size=5><b>";  
for (int i = 0; i < dat.length(); i++) {  
    if (dat.charAt(i) == 'A') {  
        hold += "<font color=#C8C8C8>" + "A" + "</font>";  
    }  
    if (dat.charAt(i) == 'R') {  
        hold += "<font color=#145AFF>" + "R" + "</font>";  
    }  
}
```

Με την δεύτερη γραμμή που αναφέρεται στο παραπάνω πλαίσιο ,η μεταβλητή data περιέχει την ακολουθία αμινοξέων που επιστρέφεται από την μέθοδο createInfo του αντικείμενου sequenceView που είναι αντικείμενο της κλάσης SeqView,όπου η συγκεκριμένη κλάση παίρνει σαν όρισμα την μεταβλητή molecule ,η συγκεκριμένη μεταβλητή περιέχει τα δεδομένα του αρχείου pdb που φορτώθηκε στο πρόγραμμα.

Με την εντολή for στην τέταρτη γραμμή και στην συνέχεια με την εντολή if προσπευλαύνονται ένα ένα τα δεδομένα της μεταβλητής dat ανάλογα το είδος του αμινοξέου που αντιπροσωπεύουν και χρωματίζονται με την χρήση της γλώσσας μορφοποίησης Html με το ανάλογο χρώμα και αποθηκεύονται εκ νέου στην μεταβλητή hold, στην γραμμή 6 & 9.

```
public void focusGained(FocusEvent e) {
    CaretListener caret=new CaretListener() {
public void caretUpdate(CaretEvent e) {
    .....
    }
};
tmp.addCaretListener(caret);
```

Στην πρώτη γραμμή στο πλαίσιο παραπάνω υλοποιείται η μέθοδος focusGained που «αντιλαμβάνεται» τα γεγονότα εστίασης που συμβαίνουν στο συστατικό που φιλοξενεί την ακολουθία αμινοξέων JEditorPane. Στην δεύτερη γραμμή δημιουργείται ο δείκτης με όνομα αντικειμένου caret, με τον συγκεκριμένο δείκτη θα επιλέγονται τα αντιπροσωπευτικά γράμματα των αμινοξέων στην ακολουθία αμινοξέων. Στην τρίτη γραμμή υλοποιείται η μέθοδος που «αντιλαμβάνεται» της ενέργειες του δείκτη και υλοποιεί της ανάλογες διαδικασίες που έχουμε ορίσει. Στην τελευταία γραμμή «εντάσσεται» στο αντικείμενο tmp μέσω της μεθόδου addCaretListener ο δείκτης caret.

```
ScrollPane pane = new ScrollPane(ScrollPane.SCROLLBARS_AS_NEEDED);
Icon img2 = new ImageIcon(getClass().getResource("../images/atomSet copy.jpg"));
JButton button2 = new JButton(img2);
Dimension dimm2 = new Dimension(20, 20);
button2.setPreferredSize(dimm2);
```

Στην πρώτη γραμμή δημιουργείται η μπάρα κύλισης με όνομα μεταβλητής pane, η συγκεκριμένη μπάρα κύλισης θα ενσωματωθεί στο συστατικό JEditorPane για την κύλιση της ακολουθίας αμινοξέων, όταν αυτή ξεπερνάει το μέγεθος του πεδίου του συστατικού JEditorPane. Στην δεύτερη γραμμή δημιουργείται το αντικείμενο εικονιδίου img2, που περιέχει την εικόνα που θα μορφοποιήσει το αντικείμενο button2 (είναι το κουμπί που ενεργοποιεί το παραθυρικό περιβάλλον που ονομάζεται amino acid colors). Στην τέταρτη και πέμπτη γραμμή ορίζονται οι διαστάσεις στο αντικείμενο dimm2 και με την μέθοδο setPreferredSize ορίζονται οι διαστάσεις στο κουμπί.

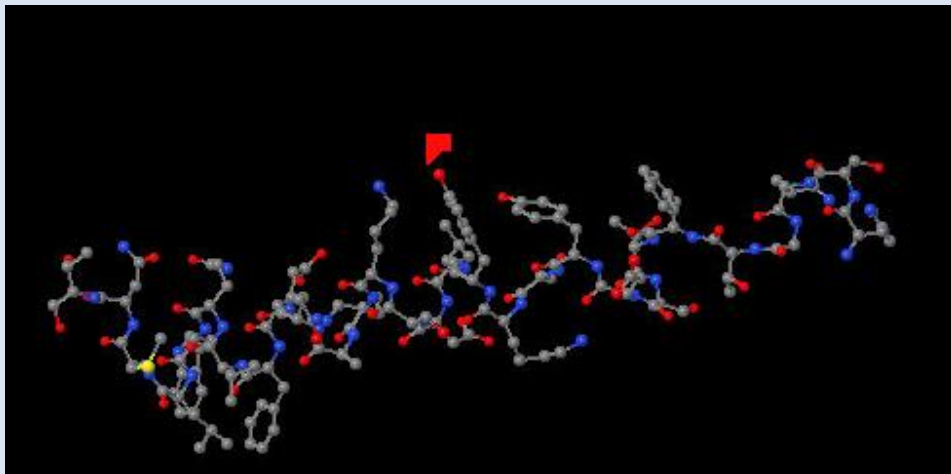
```
panel.add(pane);
panel.add(button2);
panel.add(combo);
panel.add(button);
frame1.add(panel, BorderLayout.SOUTH);
```

Στην πρώτη αλλά και στις επόμενες γραμμές του παραπάνω πλαισίου δίνονται οι εντολές για την ενσωμάτωση των επιμέρους συστατικών στο συστατικό panel του παραθυρικού περιβάλλοντος της εφαρμογής. Στην τελευταία γραμμή το συστατικό panel ενσωματώνεται στο κύριο πλαίσιο παραθύρου JFrame και στην σωστή θέση που το τοποθετεί ο διαχειριστής διάταξης BorderLayout.SOUTH.

10.4.2. Η λειτουργία εισαγωγής σχολίου (addComment)

```
public void fillComment(int x, int y, int z, int zSlab, int widthFill, int heightFill) {  
.....  
int w = widthFill - 1;  
int t=y-heightFill;  
    t=t+11;  
while(--heightFill-1>=0){ drawHLine(x, t--, z, w+3);          }  
while(--a>=0){          drawHLine(x,y++, z, w--);          }  
}
```

Απεικόνιση:



Στην πρώτη γραμμή ορίζεται η συνάρτηση fillComment που δημιουργεί το γραφικό που παριστά την εισαγωγή σχολίου και ορίζονται και οι μεταβλητές εισόδου της μεθόδου .Οι x,y,z μεταβλητές περιέχουν την θέση του γραφικού στην σκηνή του Jmol και οι μεταβλητές zSlab , widthFill, heightFill περιέχουν το μέγεθος του γραφικού .Στης παρακάτω γραμμές δημιουργείται το γραφικό ακολουθώντας το σύστημα δημιουργίας γραφικών του Jmol και χρησιμοποιώντας μεθόδους που ήδη έχουν υλοποιηθεί από το πρόγραμμα Jmol.(Εδώ πρέπει να αναφερθεί ότι για να εισαχθεί ένα γραφικό σχολίου σε ένα συγκεκριμένο άτομο πρέπει να επιλεγεί το άτομο αυτό πιέζοντας ο χρήστης το αριστερό πλήκτρο του mouse πάνω του).

```
void mouseClicked(long time, int x, int y, int modifiers, int clickCount) {
.....
text1.gemismaAtomScreen(x, y,viewer.findNearestAtomIndex(x, y+15));
}
```

Η μέθοδος mouseClicked υλοποιείτε στα συμβάντα του αριστερού πλήκτρου του mouse,κάνοντας κλικ ο χρήστης πάνω στην σκηνή απεικόνισης των μορίων στο πρόγραμμα Jmol.Η μέθοδος που υλοποιείτε στη συνέχεια είναι η gemismaAtomScreen ,η συγκεκριμένη μέθοδος παίρνει ως ορίσματα της τιμές x,y που είναι οι συντεταγμένες του σημείου της σκηνής που έγινε κλικ με το mouse,η μέθοδος findNearestAtomIndex,παίρνοντας ως ορίσματα της συντεταγμένες του σημείου ,αν 15 μονάδες παρακάτω από το σημείο που έγινε το κλικ υπάρχει κάποιο άτομο ,αν υπάρχει επιστρέφει τον αύξων αριθμό του ατόμου ,αν δεν υπάρχει επιστρέφει τιμή 0.

```
public void gemismaAtomScreen(int x,int y,int atomindex){
    boundsXY c=new boundsXY(0,0,0,"");
```

Στο πλαίσιο παραπάνω φαίνεται ένα μέρος της υλοποίησης της μεθόδου gemismaAtomScreen.Η συγκεκριμένη μέθοδος αντιστοιχεί το σχόλιο που έχει εισαχθεί ,στο αντίστοιχο γραφικό σχολίου που επέλεξε ο χρήστης μετά από το πάτημα πάνω του, του αριστερού πλήκτρου του mouse. Η κλάση boundsXY δημιουργεί ένα αντικείμενο όπου αποθηκεύεται κάθε φορά το σχόλιο που έχει εισάγει ο χρήστης σε μορφή κειμένου. Τα ορίσματα της boundsXY για κάθε σχόλιο είναι οι συντεταγμένες x,y του ατόμου που αφορά το σχόλιο ,ο αύξων αριθμός του ατόμου και το σχόλιο σε μορφή κειμένου που έχει εισάγει ο χρήστης.

```

        for(int e=0;e<v.size();e++){
            c =(boundsXY)v.get(e);
            if((atomindex==c.getatomindex())&&(((c.getx()+10)>x)&&((c.getx()-
10)<x))&&(((c.gety()+10)>y)&&((c.gety()-10)<y)))){
                String textlabel=c.getlabeltext();
                im.checkAreaN(x,y,atomindex,true,textlabel);
                break;    }
        }
        .....

```

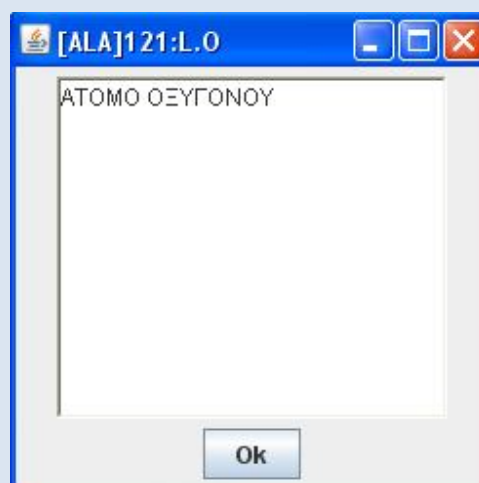
Στο πλαίσιο παραπάνω φαίνεται η υπόλοιπη υλοποίηση της gemismaAtomScreen ,με την χρήση της εντολής for προσπελούνται τα υποθηκευμένα δεδομένα του διανύσματος v. Το διάνυσμα v είναι μια διανυσματική μορφή αποθήκευσης και περιέχει όλα τα αντικείμενα που έχει δημιουργήσει η κλάση boundsXY .Με την χρήση της εντολής της δεύτερης γραμμής η τρέχουσα εγγραφή του διανύσματος v αποθηκεύεται προσωρινά στο κενό αντικείμενο της κλάσης boundsXY.Στην τρίτη γραμμή με την χρήση της εντολής if προσπαθεί να γίνει η αντιστοίχιση του κλικ του αριστερού πλήκτρου του mouse από το χρήστη ,που έχει λάβει η μέθοδος mouseClicked και των δεδομένων του διανύσματος v.Ουσιαστικά γίνεται η διαδικασία να αποσαφηνιστεί ποιο συγκεκριμένο σχόλιο επέλεξε ο χρήστης .Στην συνέχεια αν ο χρήστης έχει επιλέξει κάποιο σχόλιο ,δηλαδή αν η εντολή if επιστρέφει true,προχωράμε στην πέμπτη γραμμή και μετά όπου αποθηκεύεται σε μια String μεταβλητή το συγκεκριμένο σχόλιο σε μορφή κειμένου και καλείται η μέθοδος checkAreaN της κλάσης newImport με ορίσματα της συντεταγμένες του x,y γραφικού σχολίου ,το άτομο που ανήκει το γραφικό σχολίου και το κείμενο που έχει εισαχθεί στο συγκεκριμένο γραφικό και αποτελεί το σχόλιο .Η εντολή break τερματίζει πλέον την λειτουργία της εντολής for.

```

public void checkAreaN(int x,int y,int atomindex1,boolean smart,String strLabel){
    frame=new JFrame(strLabel);
    JPanel panel1=new JPanel();
    JScrollPane jScrollPane1 = new JScrollPane();
    JTextArea1 = new JTextArea();
    .....

```

Απεικόνιση:



Στο πλαίσιο παραπάνω φαίνεται η υλοποίηση της μεθόδου `checkAreaN`. Η συγκεκριμένη μέθοδος είναι «υπεύθυνη» για της λειτουργίες του παραθυρικού περιβάλλοντος εισαγωγής σχολίου. Στην δεύτερη γραμμή του πλαισίου υλοποιείτε ο κύριος υποδοχέας των υπόλοιπων συστατικών το `Jframe`. Στην τρίτη γραμμή υλοποιείτε το `Jpanel` που είναι ο δεύτερος κυριότερος υποδοχέας συστατικών. Στην τέταρτη γραμμή υλοποιείτε η μπάρα κύλισης που θα ενσωματωθεί στο πεδίο εισαγωγής κειμένου `JTextArea`.

```
for(int e=0;e<v.size();e++){
    c =(Comment)v.get(e);
    if(atomindex==c.getatomindex()){
        pos=e;
        fine=true;
        break;
    }else
        fine=false;
}
```

Οι παραπάνω γραμμές κώδικα εκτελούνται κάθε φορά που καλείται η μέθοδος `checkAreaN`. Ουσιαστικά η παραπάνω επαναληπτική εντολή `for` προσπευλάνει ένα ένα τα σχόλια που έχουν αποθηκευτεί στο διάνυσμα `v`, η εντολή `if` ελέγχει αν ο τρέχων αριθμός του ατόμου που ανήκει το σχόλιο που έχει επιλέξει ο χρήστης υπάρχει στο διάνυσμα `v`, αν υπάρχει κρατείται η τρέχουσα θέση που βρίσκεται στην μεταβλητή `pos` και γίνεται η flag `fine` ίση με `true` και τερματίζετε η `for` με την εντολή `break`, αν δεν υπάρχει η μεταβλητή `fine` γίνεται ίση με `false` και συνεχίζεται η `for` έως προσπελασθούν όλα τα στοιχεία του διανύσματος `v`. Συνοψίζοντας ο παραπάνω κώδικας ψάχνει να βρει αν υπάρχει ήδη υποθηκευμένο σχόλιο για κάποιο συγκεκριμένο άτομο που έχει πατήσει ο χρήστης, αν ναι τότε το `fine` γίνεται `true`, αν όχι γίνεται `false`.

```
if(fine==true){
    c=(Comment)v.get(pos);
    jTextArea1.setText(c.getText());
}
```

Αν η μεταβλητή `fine` είναι αληθής (`true`), πάει να πει ότι υπάρχει ήδη κάποιο σχόλιο αποθηκευμένο για το τρέχον άτομο στην θέση που εκφράζει η μεταβλητή `pos`, οπότε παίρνουμε από το διάνυσμα `v` το σχόλιο που έχει στη θέση που περιέχει η μεταβλητή `pos` και το εμφανίζουμε στην περιοχή εισαγωγής κειμένου του παραθύρου.

```
text1=jTextArea1.getText();
public void actionPerformed(ActionEvent r) {
    text=jTextArea1.getText();
    .....
}
```

Στην πρώτη γραμμή του παραπάνω πλαισίου αποθηκεύεται το περιεχόμενο του πεδίου εισαγωγής κειμένου `jTextArea1` στην μεταβλητή `text1`. Στην δεύτερη γραμμή υλοποιείτε η μέθοδος `actionPerformed` που ενεργοποιείτε στο πάτημα του `button` αποθήκευσης του σχολίου που έχει εισαχθεί. Στην υλοποίηση της μεθόδου περιλαμβάνετε και η αποθήκευση του περιεχομένου του πεδίου εισαγωγής κειμένου `jTextArea1` στην μεταβλητή `text`.

```

if(text1.equals(text)){
    v.insertElementAt(new Comment(id,name,x1,y1,text,atomindex),i);
}
else{
    v.insertElementAt(new Comment(id,name,x1,y1,text,atomindex),pos);
    i++;}

```

Στην πρώτη γραμμή του παραπάνω πλαισίου με την χρήση της μεθόδου equals συγκρίνονται τα δυο αλφαριθμητικά text1 και text αν είναι ίδια μεταξύ τους το πρόγραμμα στην δεύτερη γραμμή εισάγει στο διάνυσμα v ένα αντικείμενο σχολίου στην τρέχουσα θέση .Αν δεν υπάρχει ομοιότητα μεταξύ του αλφαριθμητικού text1 και του text τότε στην πέμπτη γραμμή εισάγεται ένα αντικείμενο σχολίου στο διάνυσμα v στη θέση που αναφέρει η μεταβλητή pos,δηλαδή σε αυτή την περίπτωση υπάρχει ήδη κάποιο σχόλιο που αντιστοιχεί σε κάποιο άτομο ,οπότε αλλάζει το παλιό με το νέο σχόλιο και αποθηκεύεται στην θέση pos του «παλιού» σχολίου.

```

commands.append(" set plithoscomment "+v.size()+"\n");
commands.append(" set atomindexcomment"+e+" "+c.getatomindex()+"\n");
commands.append(" set textcomment"+e+" "+sbf+c.getText()+sbf+"\n");
commands.append(" set idcomment"+e+" "+c.getId()+"\n");
commands.append(" set namecomment"+e+" "+sbf+c.getName()+sbf+"\n");
commands.append(" set xcomment"+e+" "+c.getX()+"\n");
commands.append(" set ycomment"+e+" "+c.getY()+"\n");

```

Απεικόνιση:

```

set plithoscomment 1;
set atomindexcomment0 106;
set textcomment0 "atomo oxygenou";
set idcomment0 0;
set namecomment0 "comment1";
set xcomment0 232;
set ycomment0 168;

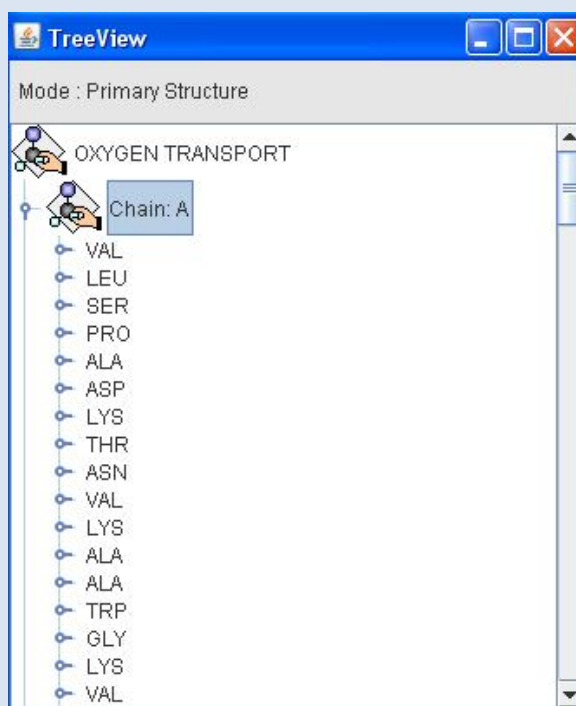
```

Παραπάνω στο πλαίσιο παρατίθεται ένα μέρος του κώδικα αποθήκευσης των σχολίων που εισάγει ο χρήστης στο πρόγραμμα .Τα σχόλια αποθηκεύονται σε αρχείο αποθήκευσης με κατάληξη SPT ,η συγκεκριμένη λειτουργία αποθήκευσης ήταν ήδη υλοποιημένη από το πρόγραμμα Jmol,η λειτουργία αποθήκευσης σχολίου ενσωματώθηκε στην προϋπάρχουσα λειτουργία .Στον κώδικα που παρατίθεται το αντικείμενο commands είναι αντικείμενο της κλάσης StringBuffer και περιέχει αλφαριθμητικά δεδομένα .Η μέθοδος append εισάγει τα δεδομένα στο αντικείμενο commands.Τα δεδομένα προσδιορίζουν κατ'αρχην τα σχόλια που έχουν εισαχθεί(plithoscomment),τον αριθμό που προσδιορίζει το άτομο που αντιστοιχεί το σχόλιο (atomindexcomment),το σχόλιο (textcomment),τον αύξων αριθμό σχολίου (idcomment),το όνομα του σχολίου (namecomment),οι συντεταγμένες του γραφικού σχολίου (xcomment, ycomment).

10.4.3. Το περιβάλλον της εφαρμογής TreeView

```
DefaultMutableTreeNode primTree = createPrimTree();  
Jtree treeCanvas = new JTree(primTree);  
treeCanvas.addTreeSelectionListener(this);  
JScrollPane tmpScroll = new JScrollPane(treeCanvas);
```

Απεικόνιση:



Στην πρώτη γραμμή του παραπάνω πλαισίου καλούμε την μέθοδο `createPrimTree`, στη συγκεκριμένη μέθοδο δημιουργείται το δέντρο αναπαράστασης των δεδομένων και οι επιμέρους κόμβοι του. Η συνάρτηση `createPrimTree` επιστρέφει ένα αντικείμενο της κλάσης `DefaultMutableTreeNode`, το αντικείμενο ονομάζεται `primTree`. Το αντικείμενο `primTree` περιέχει τους κόμβους που περιγράφουν την δομή του μορίου και δίνεται σαν όρισμα στην κλάση `Jtree` ώστε να δημιουργηθεί το δέντρο αναπαράστασης των δεδομένων. Η κλάση `Jtree` επιστρέφει το αντικείμενο που περιγράφει το δέντρο, το `treeCanvas`.

Στην τρίτη γραμμή εισάγεται στο αντικείμενο `treeCanvas` ένας ακροατής συμβάντων επιλογής (`addTreeSelectionListener`), με αυτό τον τρόπο θα μπορούμε να χειριστούμε της εκάστοτε επιλογές κόμβων από το χρήστη. Στην τέταρτη γραμμή το αντικείμενο `treeCanvas` εισάγεται στην κλάση `JScrollPane`, που αναπαριστά την μπάρα κύλισης που θα εισαχθεί σε ένα πλαίσιο `JFrame`.

```

public DefaultMutableTreeNode createPrimTree()
{
root = new DefaultMutableTreeNode(new TreeNode(0,molecule.getName(),-1,-1,-1));
for (i=0;i<molecule.getNumChains();i++)
{
DefaultMutableTreeNode chain = new DefaultMutableTreeNode(new TreeNode(1,"Chain:
"+molecule.getChain(i).getID(),i,-1,-1));

```

Στο παραπάνω πλαίσιο δίνεται ο κώδικας υλοποίησης της μεθόδου createPrimTree. Στην τρίτη γραμμή εισάγεται στο αντικείμενο root που αναπαριστά τον κόμβο που περιγράφει ολόκληρο το μόριο, το όνομα του μορίου (molecule.getName()). Στην τέταρτη γραμμή του παραπάνω πλαισίου με την εντολή for προσπελούνται μια μια η πολυπεπτιδικές αλυσίδες (molecule.getNumChains()) που υπάρχουν στο αντικείμενο molecule που περιέχει τα στοιχεία της δομής του μορίου.

Στην έκτη γραμμή του πλαισίου δημιουργείται το αντικείμενο chain που περιγράφει τον κόμβο της πολυπεπτιδικής αλυσίδας στο δέντρο αναπαράστασης του μορίου. Η συνέχεια του κώδικα υλοποίησης της μεθόδου createPrimTree δίνεται παρακάτω.

```

for (j=0;j<molecule.getChain(i).getNumGroups();j++)
{
DefaultMutableTreeNode group = new DefaultMutableTreeNode(new TreeNode(3,
molecule.getChain(i).getGroup(j).getSymbol(),i,-1,j,-1));

```

Στην πρώτη γραμμή δίνεται ο κώδικας προσπέλασης των αμινοξέων που ανήκουν στην τρέχουσα πολυπεπτιδική αλυσίδα (molecule.getChain(i).getNumGroups()).

Στην τρίτη γραμμή εισάγεται στο αντικείμενο group ο κόμβος που αντιστοιχεί σε κάθε αμινοξύ.

```

for (k=0;k<molecule.getChain(i).getGroup(j).getNumAtoms();k++)
{
DefaultMutableTreeNode atom = new DefaultMutableTreeNode(new TreeNode(4,atomcount+"
"+molecule.getChain(i).getGroup(j).getAtom(k).getName(),i,-1,j,k));
group.add(atom);
}
chain.add(group);
} root.add(chain);
}

return (root);}

```

Στην πρώτη γραμμή προσπελούνται ένα ένα τα άτομα που ανήκουν σε κάθε αμινοξύ (molecule.getChain(i).getGroup(j).getNumAtoms()).

Στην τρίτη γραμμή εισάγεται σε κάθε επανάληψη της εντολής for το όνομα του τρέχον ατόμου (molecule.getChain(i).getGroup(j).getAtom(k).getName()) στο αντικείμενο atom που αντιπροσωπεύει τον κόμβο του δέντρου για το συγκεκριμένο άτομο.

Στην πέμπτη γραμμή εισάγεται στο αντικείμενο group που αντιπροσωπεύει τον κόμβο του αμινοξέου το κάθε άτομο που ανήκει στο συγκεκριμένο αμινοξύ.

Στην έβδομη γραμμή εισάγεται στο αντικείμενο chain που αντιπροσωπεύει τον κόμβο των πολυπεπτιδικών αλυσίδων ένα ένα τα αμινοξέα που ανήκουν σε κάθε πολυπεπτιδική αλυσίδα.

Στην όγδοη γραμμή εισάγεται στο αντικείμενο root που αναπαριστά τον κόμβο ολόκληρου του μορίου ,με την σειρά οι κόμβοι που αναπαριστούν της πολυπεπτιδικές αλυσίδες.
Στην τελευταία γραμμή επιστρέφεται ο κύριος κόμβος του δέντρου που ονομάζεται root, όπου περιέχει όλους τους υπόλοιπους κόμβους και περιγράφει την πρωτοταγή δομή του μορίου.