ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΔΙΚΤΥΑΚΗ ΕΚΜΑΘΗΣΗ ΣΧΕΔΙΟΥ

ΑΣΗΜΑΚΟΠΟΥΛΟΣ ΙΩΑΝΝΗΣ

ΕΠΙΒΛΕΠΟΝΤΕΣ:

ΖΕΡΒΑΚΗΣ ΜΙΧΑΗΛ
ΑΝΤΩΝΙΑΔΗΣ ΑΡΙΣΤΟΜΕΝΗΣ

ΧΑΝΙΑ 2013

Αφιερώνεται στους γονείς μου
και στα αδέλφια μου

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ 2

2. ΣΤΑΘΜΗ ΤΩΝ ΓΝΩΣΕΩΝ 3
2.1 Η χρήση του Διαδικτύου στην εκπαίδευση 3
2.2 Εκπαίδευση με εφαρμογές των Java Applets 13

3. ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΤΟΥ APPLET 102

4. ΣΥΝΟΨΗ 123

5. ΒΙΒΛΙΟΓΡΑΦΙΑ 124

1. ΕΙΣΑΓΩΓΗ

Σκοπός αυτής της εργασίας είναι η σχεδίαση Java Applets μέσω των οποίων θα μπορεί να γίνεται εκμάθηση Μηχανολογικού Σχεδιασμού. Τα Java Applets της εργασίας επικεντρώνονται στην παρουσίαση διαφόρων μηχανολογικών σχεδίων με επερωτήσεις πάνω σε αυτά. Τα σχέδια αυτά είναι εικόνες και τρισδιάστατα σχέδια. Η προγραμματιστική συστηματοποίηση των Applets υλοποιήθηκε με τη γλώσσα Java στο πρόγραμμα Eclipse ενώ για την τελική παρουσίαση των Applets στο χρήστη, τα Applets ενσωματώθηκαν σε αρχεία της γλώσσας XHTML τα οποία διαμορφώθηκαν κατάλληλα για το μοτίβο της εργασίας.

Οι απαντήσεις του χρήστη αξιολογούνται από το πρόγραμμα. Σε κάθε απάντηση εμφανίζεται η πληροφορία στο χρήστη αν έδωσε τη σωστή ή την εσφαλμένη απάντηση. Σε περίπτωση που έδωσε σωστή απάντηση παίρνει ένα βαθμό και προχωρεί στην επόμενη ερώτηση ενώ στην αντίθετη περίπτωση δεν παίρνει βαθμό αλλά του δίνεται η δυνατότητα να προσπαθήσει για δεύτερη και τελευταία φορά στη συγκεκριμένη επερώτηση. Αν απαντήσει ετούτη τη φορά σωστά παίρνει κανονικά το βαθμό της σωστής απάντησης ενώ αν απαντήσει ξανά λάθος αυτομάτως το πρόγραμμα τον μεταφέρει στην επόμενη επερώτηση, ο χρήστης δεν παίρνει βαθμό και συγχρόνως εμφανίζεται η πληροφορία της σωστής απάντησης.

Από τα παραπάνω μπορεί να γίνει εύκολα αντιληπτό ο λόγος της επιλογής των Java Applets για την εκπαίδευση στη Μηχανολογική Σχεδίαση. Το Java Applet διευκολύνει το χρήστη με ποικίλους τρόπους. Καταρχάς το πρόγραμμα μπορεί να το χρησιμοποιήσει ο εκπαιδευόμενος χωρίς να απαιτείται να είναι ειδήμων της πληροφορικής. Αυτό επιτυγχάνεται με τη βαθιά έμφαση που δόθηκε στην προγραμματιστική σχεδίαση για τον απόλυτο προσανατολισμό των χειρισμών του χρήστη μέσα στο Applet. Επιπρόσθετα δίνεται η δυνατότητα στο χρήστη να έχει μία πολύ σαφή οπτική αντίληψη επί του μηχανολογικού απεικονίσματος. Ακόμη του δίνεται η άνεση στην επιλογή της απάντησης διότι αν επιλέξει κάποια χωρίς να το επιβεβαιώσει με το πάτημα ενός άλλου κουμπιού μπορεί αν το επιθυμεί να αλλάξει την επιλογή του. Επίσης παρέχεται στο χρήστη πλήρη ευχέρεια των κινήσεων του στο πρόγραμμα όπου μπορεί αν δε γνωρίζει την απάντηση να πάει στην επόμενη ερώτηση ή αν κάνει λάθος να δύναται να ξαναπροσπαθήσει στην απάντηση της επερώτησης. Η δυνατότητα προσφυγής στην επόμενη επερώτηση υπάρχει πάντοτε ενώ το πρόγραμμα είναι σχεδιασμένο έτσι ώστε να παρουσιάζει τη σωστή απάντηση είτε όταν ο χρήστης απάντησε δις φορές λάθος είτε όταν μεταπήδησε στην επόμενη ερώτηση οποιαδήποτε χρονική στιγμή.

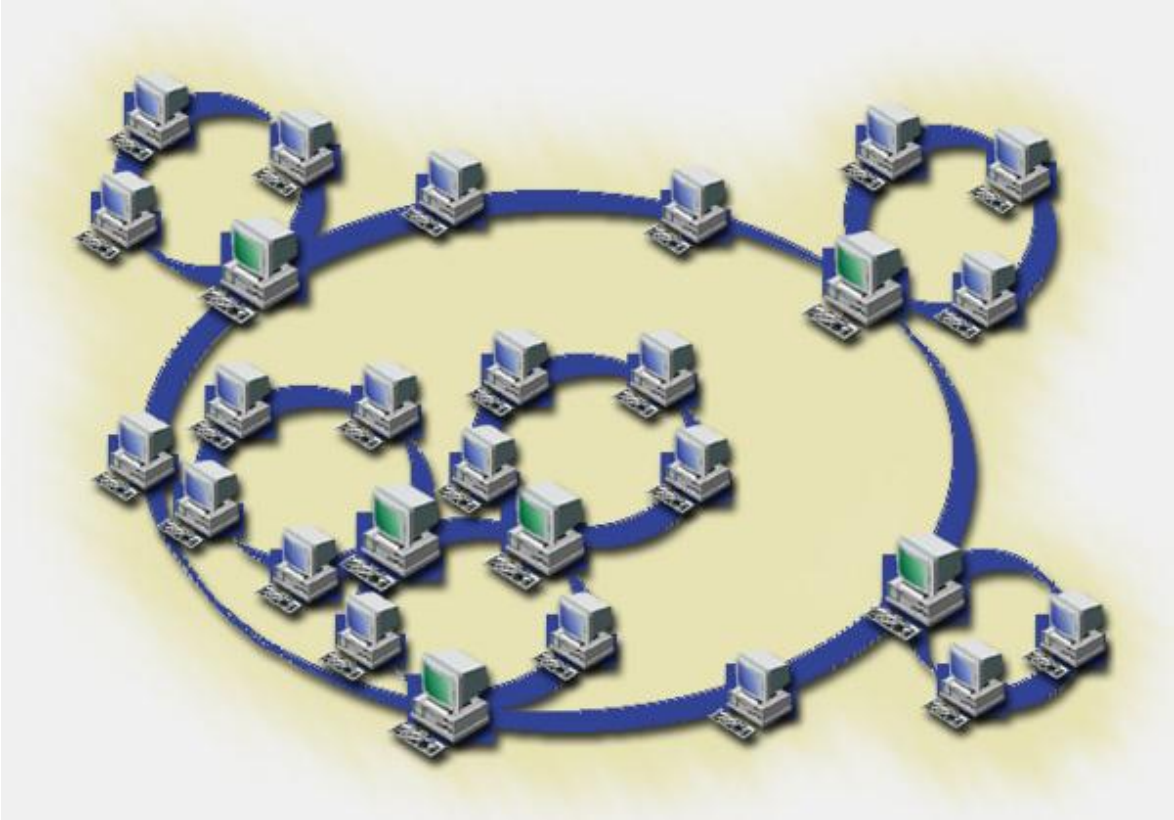
Η ευελιξία των κινήσεων του χρήστη συμπληρώνεται με το να μπορεί στο τέλος των επερωτήσεων να ξεκινήσει το Applet από την αρχή. Τέλος το όμορφο γραφικό περιβάλλον που έχει δημιουργηθεί στο πρόγραμμα κάνει το χρήστη υποσυνείδητα να έχει πολλή περισσότερη διάθεση για εκμάθηση σε σύγκριση με τον κλασικό τρόπο εκπαίδευσης.

Συνοψίζοντας η εργασία συνδυάζει άπταιστα μηχανολογία με προγραμματισμό. Τα Applets είναι σχεδιασμένα έτσι ώστε να παρέχουν την όσο το δυνατότερο προσιτή προς τον εκπαιδευόμενο διδασκαλία σε ένα όμορφο γραφικό περιβάλλον στο οποίο συγκαταλέγονται και τρισδιάστατοι σχεδιασμοί των μηχανολογικών σχεδίων με τη Java. Το πρόγραμμα είναι ικανό να καταρτίσει στη μηχανολογική σχεδίαση είτε αρχάριους Μηχανολόγους Μηχανικούς είτε εμπειρότερους και αποτελεί πλέον ένα αναπόσπαστο μέρος της ευρέως διαδεδομένης σήμερα προσέγγισης της εκπαιδεύσεως μέσω του Διαδικτύου.

2. ΣΤΑΘΜΗ ΤΩΝ ΓΝΩΣΕΩΝ

2.1 Η χρήση του Διαδικτύου στην εκπαίδευση

Το Διαδίκτυο (Internet) είναι ένα ευρύτατο δίκτυο που συνδέει μεταξύ τους πολλά μικρότερα δίκτυα ηλεκτρονικών υπολογιστών. Ένα ή και περισσότερα από τα επιμέρους δίκτυα του Διαδικτύου σηκώνει το βάρος της διασύνδεσης μεταξύ όλων των μικρότερων δικτύων και ονομάζεται ραχοκοκαλιά (backbone) του Διαδικτύου. Αυτή η εξέλιξη έλυσε πάρα πολλά προβλήματα συμβατότητας στην επικοινωνία μεταξύ των υπαρχόντων δικτύων. Το Internet αποτελεί σήμερα ένα από τα ευρύτερα μέσα επικοινωνίας και το χρησιμοποιούν πάνω από ένα δισεκατομμύριο άνθρωποι.



Σχήμα 2.1 : Το Διαδίκτυο υπολογιστών που συνδέει τα επιμέρους δίκτυα υπολογιστών μεταξύ τους.

Το Διαδίκτυο προσφέρει πολλές διαφορετικές υπηρεσίες. Ο αριθμός των υπηρεσιών δεν είναι σταθερός καθόσον ανακύπτουν νέες ανάγκες από τη χρήση του Διαδικτύου. Συνεπώς ο αριθμός των υπηρεσιών αυξάνεται ραγδαία. Οι πιο γνωστές από τις υπηρεσίες αυτές είναι το ηλεκτρονικό ταχυδρομείο, η εκπαίδευση από το Διαδίκτυο, η μεταφορά αρχείων, η σύνδεση με απομακρυσμένο υπολογιστή, η ηλεκτρονική συνδιάλεξη, η ψηλάφηση, οι ομάδες συζήτησης, οι ταχυδρομικές λίστες, οι πίνακες ανακοινώσεων και η αναμεταδιδόμενη συζήτηση.

Η αλματώδης λοιπόν τεχνολογική ανάπτυξη στις μέρες μας έχει κάνει πραγματικότητα κάτι που πριν έναν αιώνα κανείς δε θα φανταζόταν. Τη δημιουργία του Διαδικτύου. Μέσα από αυτό όπως προαναφέρθηκε είναι δυνατό να γίνει εκπαίδευση δηλαδή διδασκαλία και μάθηση. Οι εκπαιδευτές πραγματοποίησαν μέσω του Web αποτελεσματικές στρατηγικές εκπαίδευσης οι οποίες ανταποκρίνονται άψογα στα ενδιαφέροντα και τις δυνατότητες του κάθε μαθητή ξεχωριστά. Οι παραπάνω μέθοδοι των εκπαιδευτών είναι κυρίως η αξιοποίηση των τεχνολογιών ήχου, βίντεο και εικόνας που προσφέρονται από το Internet. Έτσι παρέχονται ευκαιρίες

εκμάθησης διαφορετικές από εκείνες που προσφέρει η παραδοσιακή εκπαίδευση στο πλαίσιο των κανονικών τάξεων. Αυτό γίνεται από τους εκπαιδευτές με την επέκταση και υποστήριξη των πολυμέσων καθώς και με την καθοδήγηση για πρόσβαση στους μαθητές σε πηγές πληροφορίας, σε εικόνες και σε κατάλληλους ήχους καθώς το Web είναι κυριολεκτικά μία ανεξάντλητη κρήνη γνώσης. Άλλες τακτικές των εκπαιδευτών είναι οι ηλεκτρονικές συλλογές υλικού τις οποίες διαχειρίζονται χρήστες ή εκπαιδευτές, οι προσωπικοί υπολογιστές παλάμης, οι ψηφιακοί δίσκοι πολυμέσων, το λογισμικό υποστήριξης συνεργασίας, το ηλεκτρονικό ταχυδρομείο, τα ημερολόγια Διαδικτύου, οι εγκυκλοπαίδειες διαχειριζόμενες από τους χρήστες, η σύγχρονη συζήτηση με κείμενο και η αξιολόγηση υποβοηθούμενη από τον υπολογιστή. Επιπλέον είναι οι εξομοιωτές, τα παιχνίδια, το εικονικό περιβάλλον εκπαίδευσης, η διανομή συλλογών ψηφιακών αρχείων σε πολλούς παραλήπτες και το ηλεκτρονικό σύστημα υποστήριξης της απόδοσης όπου είναι το πρόγραμμα που διευκολύνει την πρόσβαση σε πληροφορίες. Με τους τρόπους αυτούς οι δάσκαλοι ελαχιστοποίησαν υπέροχα μία συνοχή από γεωγραφικούς, θεσμικούς, φυλετικούς και χρονικούς περιορισμούς της γνώσης.

Αν και τα μέσα της Διαδικτυακής εκπαίδευσης είναι αναρίθμητα το καθένα μπορεί να αποτελείται από διαφορετικά προγράμματα τα οποία συνεργάζονται. Το πιο διάχυτο σύστημα που χρησιμοποιείται στην ανώτατη εκπαίδευση είναι το σύστημα διαχείρισης της μάθησης ή αλλιώς εικονικό περιβάλλον εκπαίδευσης. Οι όροι χρησιμοποιούνται για να περιγράψουν το λογισμικό που σχεδιάστηκε για τη διαχείριση των δραστηριοτήτων εκπαίδευσης. Μετεξέλιξη τους είναι το σύστημα διαχείρισης εκπαιδευτικού περιεχομένου που προσαρτεί τη λειτουργικότητα της επαναχρησιμοποίησης του υλικού ή μέρος αυτού. Οι λειτουργίες εμπεριέχουν την παιδεία από το Διαδίκτυο, την εγγραφή χρηστών, το ημερολόγιο μαθημάτων, τη ροή κατάρτισης, τη διαχείριση των χρηστών, την αξιολόγηση των εκπαιδευομένων, τις υπηρεσίες της τηλεδιάσκεψης και τη συνεργατική μάθηση ήτοι τις συζητήσεις και ανταλλαγές αρχείων.

Σχήμα 2.2 : Με τη Διαδικτυακή εκπαίδευση μαθαίνεις σε όποιο σημείο της γης και αν βρίσκεσαι.

Η εκπαίδευση μέσω του Διαδικτύου μπορεί να διαχωριστεί στη σύγχρονη και την ασύγχρονη εκπαίδευση. Στη σύγχρονη εκπαίδευση η διαδικασία της διδασκαλίας και της μάθησης γίνονται ταυτόχρονα. Ο εκπαιδευτής παραδίδει το μάθημα σε ζωντανή σύνδεση όχι απαραίτητα αμφίδρομη και ο εκπαιδευόμενος αν και βρίσκεται σε διαφορετικό τόπο παρακολουθεί το μάθημα στον ίδιο χρόνο. Χαρακτηριστικό παράδειγμα είναι η Διαδικτυακή εκπαίδευση με την υποστήριξη της τηλεδιάσκεψης. Το ίδιο και η χρήση δωματίου ζωντανής συζήτησης - live chartroom.

Στην ασύγχρονη εκπαίδευση που είναι πιο διαδεδομένη ο εκπαιδευόμενος μαθαίνει όχι μόνο σε διαφορετικό χώρο από τον εκπαιδευτή αλλά και σε διαφορετικό χρόνο από τη διαδικασία της παράδοσης ή της δημιουργίας του μαθήματος. Χαρακτηριστικά παραδείγματα είναι τα αμέτρητα βίντεο στο youtube που έχουν ανεβάσει καθηγητές.

Σήμερα υπάρχουν μέθοδοι εκπαίδευσης από το Διαδίκτυο που χρησιμοποιούν και τα δύο είδη ώστε να παρέχουν ολοκληρωμένη εκπαιδευτική εμπειρία. Με τη βοήθεια της ασύγχρονης εκπαίδευσης υπάρχει το πλεονέκτημα της μάθησης στο χρόνο και με το ρυθμό που επιθυμεί ο εκπαιδευόμενος. Με τη χρήση σύγχρονων συζητήσεων σε τακτά χρονικά διαστήματα ο εκπαιδευτής γνωρίζει τους μαθητές του, οι εκπαιδευόμενοι επικοινωνούν μεταξύ τους και με τον εκπαιδευτή και οι μαθητές δεν αισθάνονται αποξενωμένοι από την εκπαιδευτική κοινότητα και διαδικασία.

Η εκπαίδευση μέσω του Διαδικτύου όπως είναι κατανοητό έχει εξελιχθεί σε μεγάλο βαθμό. Ο χρήστης έχει πρόσβαση σε μεγάλο όγκο πληροφοριών και τα εργαλεία που δύναται να χρησιμοποιήσει είναι τόσα πολλά και διαφορετικά ώστε μπορούμε να μιλούμε για μία τελείως διαφορετική μεθοδολογία από την παραδοσιακή εκμάθηση. Αυτή η πραγματικότητα μας δείχνει το πέρασμα σε μία νέα επιμορφωτική εποχή. Σε αυτήν τη νέα εποχή οι χρήστες καθώς εξοικειώνονται όλο και περισσότερο με το Διαδίκτυο και τις υπηρεσίες του το διαμορφώνουν ομοίως όλο και περισσότερο. Είναι γεγονός ότι σήμερα αρίφνητοι χρήστες του Διαδικτύου συνεισφέρουν με ευκολία στη δημιουργία και διαμόρφωση του περιεχομένου του Internet. Σε αυτόν ακριβώς το σκοπό έχει επικεντρωθεί η πλειοψηφία των υπηρεσιών του Web ώστε να καθιστούν ακόμη και τον σχετικά αρχάριο χρήστη σε εκπαιδευτή.

Τα βασικά χαρακτηριστικά αυτών των υπηρεσιών είναι η ευκολία χρήσης του Web καθώς και η ευελιξία περιήγησης στο περιεχόμενο. Έτσι όχι μόνο υπάρχει απεριόριστη ποσότητα πληροφοριών, υλικού και γνώσης αλλά ο χρήστης έχει επιπλέον και τη δυνατότητα να τα διαχειριστεί. Ένα πολύ ενδιαφέρον κομμάτι που σχετίζεται με την ευκολία διαχείρισης των πληροφοριών είναι αυτό το οποίο ονομάζεται συλλογική ευφυΐα. Παράδειγμα είναι η ιστοσελίδα του ηλεκτρονικού βιβλιοπωλείου Amazon όπου δίνεται η δυνατότητα στους χρήστες να βαθμολογήσουν και να σχολιάσουν τα προϊόντα. Έτσι προηγούμενοι χρήστες επηρεάζουν τη συμπεριφορά μελλοντικών χρηστών μεταφέροντας την εμπειρία και τη γνώμη τους. Η μέθοδος αυτή έχει ασύνορες δυνατότητες και τρομακτική δύναμη επιτρέποντας με έναν πολύ απλό τρόπο τη συμβολή στο περιεχόμενο του Διαδικτύου.

Σχήμα 2.3 : Το ηλεκτρονικό βιβλιοπωλείο Amazon.

Οι πιο πολύτιμες υπηρεσίες για το σκοπό της Διαδικτυακής εκπαίδευσης είναι το wiki και τα δικτυακά ημερολόγια ενώ υπάρχει ήδη τεράστιος αριθμός εκπαιδευτικών

κοινοτήτων που αξιοποιεί τις δυνατότητες τους. Πρέπει επίσης να αναφερθούν και οι υπηρεσίες RSS - Really Simply Syndication που συμβάλλουν στην παρακολούθηση και ενημέρωση των χρηστών για τα τελευταία νέα και τις αλλαγές σε ένα δικτυακό τόπο. Ο συνδυασμός των τεχνολογιών αυτών μπορεί να αποτελέσει εκπαιδευτική διαδικασία και χωρίς την ύπαρξη επίσημου οργανωμένου μαθήματος.

Σχήμα 2.4 : Η βικιπαιδεία εξηγεί αναλυτικά αυτό που ζητά ο χρήστης λειτουργώντας ως μία ηλεκτρονική εγκυκλοπαίδεια.

Η εκπαίδευση μέσω του Διαδικτύου έχει συγκεκριμένες τεχνικές απιτήσεις τόσο σε λογισμικό όσο και σε εξοπλισμό.Το απαιτούμενο λογισμικό αφορά την όλη διαδικασία της δημιουργίας του συστήματος της Διαδικτυακής εκπαίδευσης από τη δημιουργία του μέχρι την ολοκλήρωση της εκπαιδευτικής διαδικασίας στον εκπαιδευόμενο. Το ενδιαφέρον είναι ότι σε όλα τα παραπάνω επίπεδα είναι δυνατό να χρησιμοποιηθεί λογισμικό ανοιχτού κώδικα που διατίθεται δωρεάν. Το λογισμικό αυτό περιλαμβάνει:

- Λειτουργικό σύστημα εξυπηρετητή - server και πελάτη - client, π.χ. κάποια διανομή Linux.
- Λογισμικό εξυπηρετητή, π.χ. Apache Server.
- Λογισμικό δημιουργίας εκπαιδευτικού υλικού, π.χ. OpenOffice.
- Λογισμικό διαχείρισης της μάθησης ή εικονικού περιβάλλοντος εκπαίδευσης, π.χ. Atutor, Claroline, Efront, Dokeos ή Moodle.
- Περιηγητή Διαδικτύου ή αλλιώς φυλλομετρητής π.χ. Mozilla Firefox. Τις περισσότερες φορές το λογισμικό αυτό είναι το μόνο απαραίτητο για τον εκπαιδευόμενο ώστε να έχει πρόσβαση στο εκπαιδευτικό πρόγραμμα. Το ίδιο συμβαίνει και με τον εκπαιδευτή ο οποίος μπορεί να διαχειριστεί και να ανεβάσει το εκπαιδευτικό υλικό μόνο με τη χρήση ενός περιηγητή Διαδικτύου.
- Λογισμικό υποστήριξης πληροφοριών που χρησιμοποιούνται από το σύστημα διαχείρισης της μάθησης ή το εικονικό περιβάλλον εκπαίδευσης, π.χ. Java, Flash, RealMedia, Quicktime, Windows Media Files.

Οι απαιτήσεις σε επίπεδο εξοπλισμού είναι:

- Εξυπηρετητής ιστού (web server).
- Εξυπηρετητής αρχείων (file server).
- Τερματικά.
- Σύνδεση Διαδικτύου για τους υπολογιστές.

Η χρήση του Διαδικτύου ως πηγή άντλησης πληροφοριών για τα μαθήματα ή για τη διενέργεια ερευνών μπορεί να κεφαλαιοποιήσει τις πραγματικές δυνάμεις της έρευνας και της πρακτικής στη διδασκαλία των τάξεων. Ένα WebQuest για παράδειγμα αποτελεί μία δραστηριότητα έρευνας στην οποία οι περισσότερες ή και όλες οι πληροφορίες που χρησιμοποιούν οι μαθητές προέρχονται από το Internet. Ωστόσο τα WebQuests δεν είναι βασισμένα στην απλή ανάκληση πληροφοριών αλλά στην έρευνα. Οι εκπαιδευτικοί επομένως μπορούν να σχεδιάσουν μία έρευνα την οποία οι μαθητές τους θα αναλάβουν να διεκπεραιώσουν με αναζήτηση στο Web. Μπορούν επίσης να προγραμματίσουν πληθώρα πολυπολιτισμικών δραστηριοτήτων ή ακόμη και να αναπτύξουν μόνοι ή σε στενή συνεργασία με άλλα σχολεία ολοκληρωμένα εκπαιδευτικά προγράμματα. Σε μία διαπολιτισμική μελέτη για παράδειγμα οι εκπαιδευτικοί χρησιμοποίησαν εικονικές επισκέψεις παιδαγωγικού περιεχομένου για να αναδιφήσουν απομακρυσμένες περιοχές της Αλάσκας. Άλλοι εκπαιδευτικοί έχουν αξιοποιήσει διαπολιτισμικά αναλυτικά προγράμματα μέσα από σχέδια μαθημάτων, πλασματικά μουσεία, πολιτισμικά βίντεο και ηχητικά ακούσματα από διαφορετικές χώρες. Ετούτη η συμπίεση των χωροχρονικών περιορισμών ως αποτέλεσμα της Διαδικτυακής εκμάθησης δημιουργεί πλέον πρωτόγνωρες και ευρηματικές δυνατότητες στην εκπαιδευτική διαδικασία. Ένα απλό παράδειγμα είναι η εκπόνηση κοινών εργασιών ανάμεσα σε μαθητές που ζουν σε διαφορετικούς τόπους του πλανήτη, με ανόμοιο εθνοπολιτισμικό και κοινωνικοοικονομικό υπόβαθρο γεγονός το οποίο μπορεί να εισφέρει στην προώθηση της διαπολιτισμικής κατανόησης και συνεργασίας.

Το Διαδίκτυο παρέχει πάμπολλους νέους τρόπους συνεργασίας σε εκπαιδευτικούς με εντελώς διαφορετικό εθνοπολιτισμικό κρηπίδωμα. Ένας εκπαιδευτικός από την Ελλάδα, για παράδειγμα, με αρκετούς μαθητές αλβανικής καταγωγής στην τάξη του, θα μπορούσε να επικοινωνήσει με έναν εκπαιδευτικό από την Αλβανία, για να αντλήσει χρήσιμες πληροφορίες σχετικά με τον τρόπο μάθησης και διδασκαλίας, τον τρόπο αξιολόγησης των μαθητών, τον πολιτισμό ή τα ήθη και έθιμα, έτσι ώστε να κατανοήσει καλύτερα τις ανάγκες των αλλοδαπών μαθητών του. Η επικοινωνία μπορεί να πραγματοποιηθεί μέσα σε περιοχές πολλών χρηστών, σε σχετικών φόρουμ συζήτησης, μέσω ηλεκτρονικού ταχυδρομείου ή με τηλεδιάσκεψη.

Επιπρόσθετα το Διαδίκτυο εκχωρεί περισσότερες ευκαιρίες στους πολιτισμικά ανόμοιους μαθητές στο να μπορούν να χρησιμοποιήσουν τόσο τη μητρική όσο και την επίσημη γλώσσα της χώρας υποδοχής. Οι μαθητές, για παράδειγμα, μπορούν να αναβαθμίσουν τις δεξιότητές τους στην ανάγνωση, τη γραπτή και την προφορική έκφραση ή τη μορφοποίηση κειμένου και να εμπλουτίσουν το λεξιλόγιο τους. Τέλος, τόσο τα εικονικά όσο και τα σε απευθείας σύνδεση περιβάλλοντα, μπορούν να σχεδιαστούν με τρόπο ώστε να είναι ικανά να προωθήσουν την επίλυση των προβλημάτων, τη ζωντανή αλληλεπίδραση και τη δημιουργική συνεργασία.

Η παράδοση μαθημάτων μέσω του Διαδικτύου δύναται να μεταβάλλει ολοκληρωτικά το εκπαιδευτικό τοπίο και να επηρεάσει αποφασιστικά την ανάπτυξη και τον έλεγχο του αναλυτικού προγράμματος σπουδών. Το Διαδίκτυο επιτρέπει την αναζήτηση και εξεύρεση πληροφοριών μέσα από την υπεραφθονία των διαθέσιμων

πηγών. Η εύκολη και γρήγορη πρόσβαση σε ένα εξαιρετικά μεγάλο αριθμό δικτυακών τόπων, καθώς και η αλληλεπίδραση με ανθρώπους που βρίσκονται σε μεγάλη απόσταση και ζουν σε διαφορετικό γεωγραφικό και πολιτισμικό περιβάλλον, παρέχει τη δυνατότητα πρόσκτησης καθαρά εξατομικευμένων μορφών γνώσης και μπορεί να ωφελήσει το ίδιο μαθητές και εκπαιδευτικούς. Το Διαδίκτυο παρέχει επίσης περισσότερες ευκαιρίες πρόσβασης στην ακαδημαϊκή κοινότητα, σε άτομα τα οποία ήταν αποκλεισμένα από αυτήν λόγω γεωγραφικών ή πολιτισμικών περιορισμών.

Σχήμα 2.5 : Η φιλελεύθερη άποψη για την εκπαίδευση είναι ότι αυτή πρέπει να παρέχεται σε όλα τα παιδιά ανεξαρτήτως χρώματος, εθνικότητας και κοινωνικής τάξης πράγμα που το καθιστά δυνατόν η εκπαίδευση μέσω του Διαδικτύου.

Η προσέγγιση της εκπαίδευσης μέσα από ένα παιδαγωγικό πλαίσιο εμπλουτισμένο με διαπολιτισμικές προσεγγίσεις βασισμένες στο Διαδίκτυο συνάδει με τις ανάγκες της τεχνολογικά προσανατολισμένης κοινωνίας μας. Το ζήτημα δεν είναι πλέον αν θα χρησιμοποιήσουμε το Διαδίκτυο στην εκπαίδευση, αλλά μάλλον με ποια τακτική θα το χρησιμοποιήσουμε, για να υποστηρίξουμε τη διδασκαλία και την εκμάθηση μέσα στις σύγχρονες πολυπολιτισμικές τάξεις. Οι προσπάθειες συμπόρευσης της διαπολιτισμικής εκπαίδευσης με τις παιδαγωγικές εφαρμογές του Διαδικτύου βρίσκονται σε συνεχή εξέλιξη. Θα μπορούσαμε να ισχυριστούμε ότι οι σχέσεις του ζεύγους Διαδίκτυο - διαπολιτισμική εκπαίδευση μπορούν να είναι αρμονικές.

Το Διαδίκτυο μπορεί να διευρύνει τα διαθέσιμα μέσα διδασκαλίας και μάθησης και να προσφέρει νέες προκλήσεις και εκπαιδευτικές δυνατότητες. Μέσα από το σχεδιασμό και την εφαρμογή των κατάλληλων στρατηγικών, μπορεί να αποτελέσει ένα σημαντικό συμπληρωματικό παιδαγωγικό και διδακτικό εργαλείο στην ανάπτυξη διαπολιτισμικών διαστάσεων στην εκπαιδευτική διαδικασία. Σε καμία περίπτωση, ωστόσο, δεν αποτελεί πανάκεια ή υποκατάστατο των ωφέλιμων βιβλίων, των προσωπικών επαφών, του εποικοδομητικού διαλόγου ή άλλων παραδοσιακών μορφών διδασκαλίας.

Η δημιουργία ομάδων συνομιλίας στο Διαδίκτυο, για εκπαιδευτικούς και μαθητές, παρέχει στην εκπαίδευση τη δυνατότητα να κινηθεί πέρα από αυτό που μερικοί ερευνητές αναφέρουν ως κάτι το επιφανειακό και επουσιώδες. Η τεχνολογία μπορεί ενδεχομένως να ωθήσει εκπαιδευτικούς και μαθητές να επανεξετάσουν τις υποθέσεις τους για άλλους πολιτισμούς ή να αμφισβητήσουν οποιεσδήποτε στερεοτυπικές τους αντιλήψεις μέσα από την ανάπτυξη εποικοδομητικών μορφών. Ένα από τα απλούστατα παράδειγματα θα μπορούσε να είναι η αναθεώρηση ενός Τούρκου στρατιωτικού μετά από συζήτηση με Έλληνα πολίτη μέσω του Web για τις κάκιστες απόψεις που έχει εναντίον των Ελλήνων υπηκόων.

Σχήμα 2.6 : Στην εικόνα φαίνεται ένα παράδειγμα συζήτησης στο Web.

Οι συζητήσεις στο Διαδίκτυο βοηθούν τους μαθητές να αναπτύξουν και τις προφορικές τους δεξιότητες μέσα στην τάξη, όπως προκύπτει από έρευνες των Pagnucci και Mauriello (1999) και Takayoshi και συνεργάτες, (1999). Συγκεκριμένα οι ταυτόχρονες συζητήσεις στο Διαδίκτυο παρουσιάζουν τα παρακάτω πλεονεκτήματα: Πρώτα από όλα βοηθούν τους μαθητές που είναι επιφυλακτικοί στις πρόσωπο με πρόσωπο συζητήσεις να ξεπεράσουν τη διστακτικότητα τους. Έπειτα η πλειονότητα των μαθητών είναι σε θέση να εκκινήσει μία συζήτηση εκ περιτροπής. Ακόμη οι μαθητές μπορούν να σκεφτούν και να αναθεωρήσουν τη συνεισφορά τους στη συζήτηση ανετότερα ενώ επιπλέον ενθαρρύνεται και η ισότιμη επικοινωνία.

Το μεγάλο πλεονέκτημα του Internet είναι ο ανοικτός χαρακτήρας του ο οποίος επιτρέπει την ταχύτατη δημιουργία, ανάπτυξη και εμπορική αξιοποίηση νέων τεχνολογιών. Για αυτό και οι εφαρμογές του στον τομέα της εκπαίδευσης είναι ήδη άπλετες και προβλέπεται πως θα αυξηθούν ακόμη περισσότερο στο άμεσο μέλλον. Παρά τη μεγάλη τους ποικιλία όμως όλες οι εκπαιδευτικές εφαρμογές του Internet μπορούν να χωριστούν σε τρεις κατηγορίες ανάλογα με τον τρόπο που χρησιμοποιούν οι διδάσκοντες για να έρθουν σε επαφή με τους μαθητές.

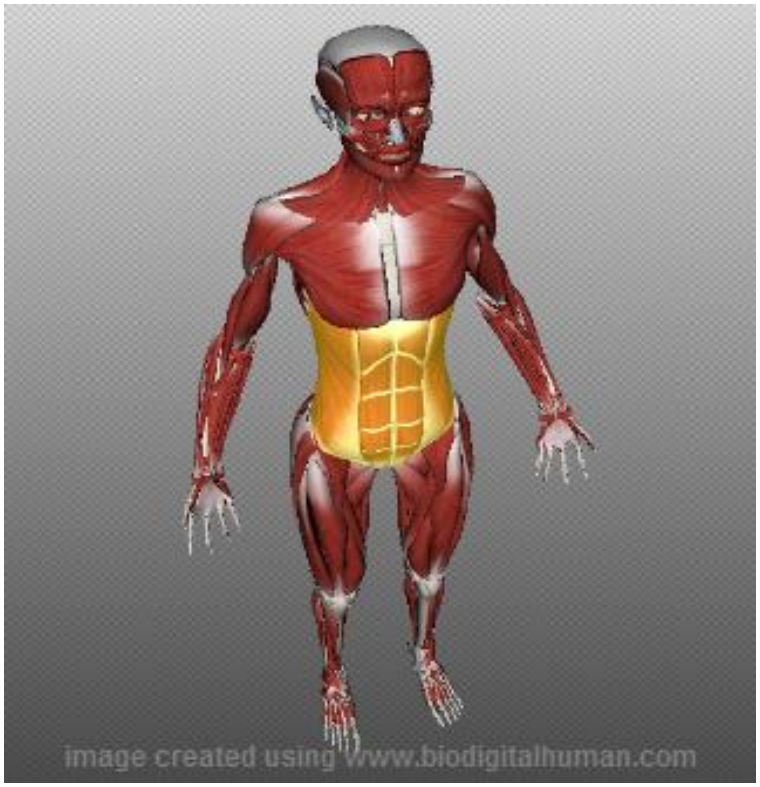
Οι τρόποι αυτοί είναι:

- Η πληροφοριακή επικοινωνία.
- Η προσωπική επικοινωνία με τρίτους.
- Η προσωπική επικοινωνία με εξειδικευμένο εκπαιδευτή.

Παραδείγματα πληροφορικής επικοινωνίας είναι οι μελέτες, εργασίες και συμπεράσματα ερευνών από ιδιώτες, πανεπιστήμια, μη κερδοσκοπικούς οργανισμούς και επιχειρήσεις τα οποία βρίσκονται στο Διαδίκτυο. Επιπρόσθετα είναι τα online μαθήματα ήτοι σειρές κειμένων με αντικείμενο τη διδασκαλία συγκεκριμένων θεμάτων όπως ο οδηγός κατασκευής Web σελίδων. Ακόμη είναι τα μηνύματα

ενημερωτικού περιεχομένου τα οποία έχουν δημοσιευθεί σε δημόσιους χώρους του Internet όπως για παράδειγμα τα usenet groups ή τα mailing lists.

Το κύριο χαρακτηριστικό της πληροφοριακής επικοινωνίας είναι η απουσία οποιασδήποτε ατομικής επαφής του εκπαιδευτή με το μαθητή. Στις περιπτώσεις δημοσίευσης μελετών ή online μαθημάτων, ο δημιουργός τους τοποθετεί απλώς την εργασία του στο δίκτυο και τη θέτει στη διάθεση κάθε ενδιαφερόμενου. Στις περιπτώσεις των μηνυμάτων ενημερωτικού περιεχομένου ο κάτοχος κάποιας πληροφορίας τη δηλοποιεί σε ένα χώρο ανακοινώσεων ως αυτόνομη ανακοίνωση όπως: "Μάθατε πως η θάλασσα κατάπιε δύο νησιά στο Κιριμπάτι;" Επίσης μπορεί να την κοινοποιήσει σε ένα χώρο ανακοινώσεων ως απάντηση σε κάποιον τρίτο όπως: "Η εποχή του Μπαρόκ για την οποία ρωτούσες διήρκεσε από το 1600 έως το 1750 μ.Χ. περίπου." Τα μηνύματα αυτά αποθηκεύονται σε διάφορες βάσεις δεδομένων όπως για παράδειγμα το deja news και είναι σήμερα διαθέσιμα σε όλους τους χρήστες του δικτύου. Μία αναζήτηση με τις λέξεις "Μπαρόκ" και "διήρκεσε" θα μας φέρει την απάντηση την οποία έδωσε πριν δύο χρόνια ο χρήστης Χ στον Ψ.



Σχήμα 2.7 : Η μελέτη της ανθρώπινης ανατομίας από μη γιατρό μέσω ενός τρισδιάστατου μοντέλου του Web αποτελεί ένα τέλειο παράδειγμα της πληροφορικής επικοινωνίας.

Η πληροφοριακή επικοινωνία αποτελεί τον πιο ευρέως διαδεδομένο τρόπο εκπαιδευτικής αξιοποίησης του Internet και χρησιμοποιείται από ιδιώτες οι οποίοι επιδιώκουν να αυτομορφωθούν σε κάποιο αντικείμενο π.χ. το πως καλλιεργούνται οι τουλίπες. Επιπλέον χρησιμοποιείται από ερευνητές οι οποίοι συγκεντρώνουν πληροφορίες όπως για παράδειγμα στελέχη επιχειρήσεων που αξιολογούν μία νέα τεχνολογία. Άλλοι χρήστες της πληροφορικής επικοινωνίας είναι οι εκπαιδευόμενοι που συγκεντρώνουν υλικό για τη σύνταξη σχολικών εργασιών ή την καλύτερη αντίληψη των μαθημάτων τους. Ένα απλό παράδειγμα είναι οι μαθητές οι οποίοι διδάχθηκαν ότι ο Αρχιμήδης έκαψε το ρωμαϊκό στόλο χρησιμοποιώντας κάτοπτρα μπορούν να μελετήσουν τους νόμους της οπτικής για να κατανοήσουν με ποιο τρόπο έγινε αυτό.

Σχήμα 2.8 : Ένα παράδειγμα επικοινωνίας στο Διαδίκτυο μέσω του προγράμματος Skype η οποία μπορεί να εξελιχθεί σε εκπαίδευση από το Internet.

Η πληροφοριακή επικοινωνία χρησιμοποιείται κυρίως από ενήλικες και είναι μία ως επί το πλείστον μοναχική απασχόληση. Η μελέτη των πηγών πληροφοριών του δικτύου γίνεται στο χρόνο που εξυπηρετεί καλύτερα το χρήστη με την ταχύτητα και με τη σειρά που ο ίδιος επιθυμεί. Δηλαδή σε μία σειρά 20 online μαθημάτων δύναται να παραλείψει το 6ο ή να μελετήσει πρώτα το 12ο και μετά το 8ο. Συνήθως, αυτή η μορφή χρήσης του Διαδικτύου είναι ανοργάνωτη, αποσπασματική και ευκαιριακή καθόσον οι περισσότεροι χρήστες του δικτύου εκτελούν ευκαιριακές αναζητήσεις χωρίς μακροπρόθεσμο σχέδιο. Σήμερα ο χρήστης χρειάζεται πληροφορίες για τις ασθένειες των μελισσών ενώ αύριο μπορεί να ανασκαλεύει για το κατά κεφαλήν εισόδημα της Βραζιλίας.

Το μεγαλύτερο πλεονέκτημα της πληροφοριακής επικοινωνίας είναι το χαμηλό της κόστος. Δηλαδή απλώς τοποθετούμε το υλικό online. Επιπρόσθετα άλλο πλεονέκτημα είναι η εύκολη πρόσβαση στο περιεχόμενο της όλο το 24ωρο. Το ακρογωνιαίο μειονέκτημα της είναι η έλλειψη επαφής του εκπαιδευτή με τον εκπαιδευόμενο. Δεν απαντώνται απορίες, ούτε γίνονται διευκρινήσεις. Έτσι αν ένα μέρος του κειμένου δεν είναι κατανοητό ή σε κάποιο θέμα δε δίδεται η αναμενόμενη βαρύτητα η μόνη λύση για τον εκπαιδευόμενο είναι η εύρεση κάποιας άλλης πηγής πληροφοριών. Η μέχρι σήμερα εμπειρία έχει δείξει πως λόγω της χαώδους φύσης του δικτύου οι μαθητές δαπανούν μεγάλο μέρος του χρόνου τους στην αναζήτηση πληροφοριών. Έτσι ο χρόνος ο οποίος τους μένει διαθέσιμος για μελέτη είναι αρκετά μικρότερος από ότι στις άλλες μορφές αυτομόρφωσης.

Σχετικά με την προσωπική επικοινωνία με τρίτους στο Internet υπάρχουν χιλιάδες χώροι προσωπικής επικοινωνίας μεταξύ των χρηστών. Δεκάδες χιλιάδες mailing lists, usenet groups, chat rooms και άλλοι χώροι φιλοξενούν καθημερινά ένα τεράστιο νούμερο συζητήσεων, μηνυμάτων και ανακοινώσεων, ταξινομημένων κατά θέμα και κατηγορία. Παραδείγματα είναι η java, το ms-access, η μεσαιωνική ιστορία, η αστροφυσική, η οικολογία και αμέτρητα άλλα. Πολλοί χρήστες του Διαδικτύου

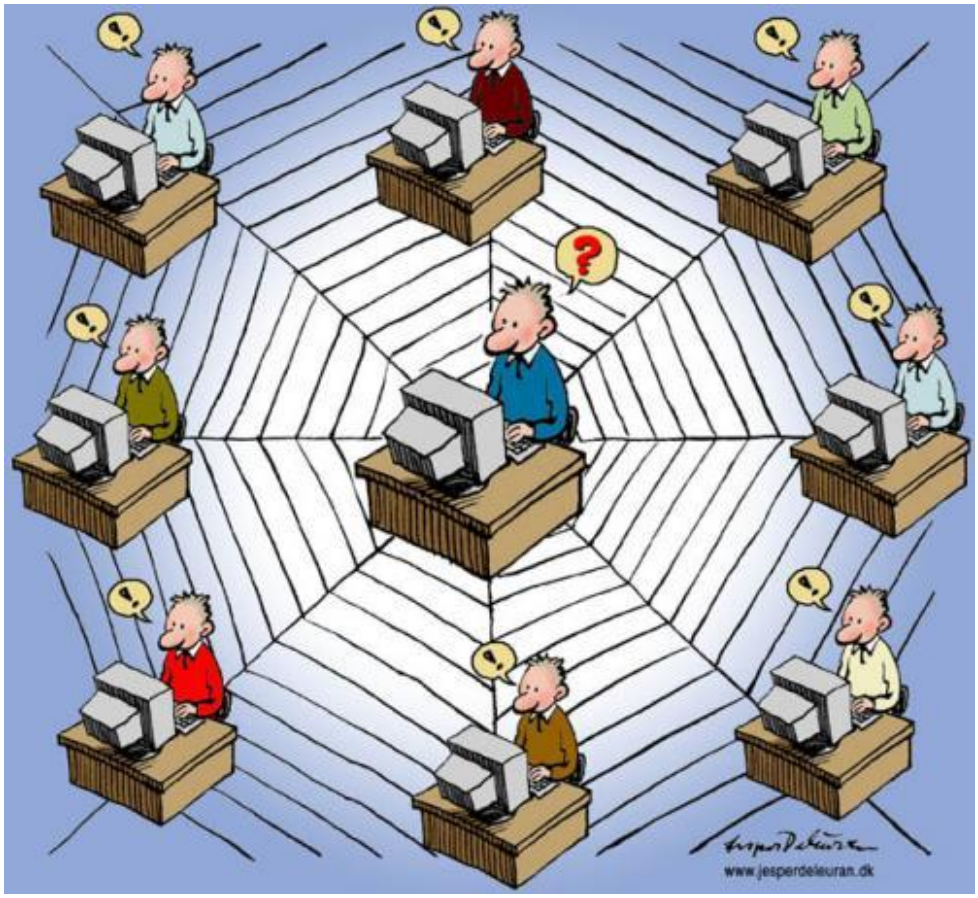
καταφεύγουν σε αυτούς τους χώρους για την υποβολή δημόσιων ερωτήσεων σχετικά με θέματα που τους απασχολούν. Αν και υπάρχουν περιορισμοί σε αυτήν τη μέθοδο όπως το ότι δε μπορείς να υποβάλλεις πολλές ερωτήσεις, η διατύπωση τους πρέπει να υπακούει σε ειδικούς κανόνες, δεν είναι βέβαιο πως θα λάβεις απάντηση καθώς και ποικίλες άλλες συγκρατήσεις, πολλές χιλιάδες άνθρωποι αξιοποιούν καθημερινά το πνεύμα συνεργασίας και αλληλοϋποστήριξης που διέπει τους χώρους αυτούς για να λάβουν απαντήσεις στα ερωτήματα τους. Ένα παράδειγμα είναι οι ελάχιστες ερωτήσεις που μένουν δίχως απάντηση στην ομάδα χρηστών της Microsoft Access στο comp.databases.ms-access.

Μολονότι η προσωπική επικοινωνία με τρίτους δεν παρέχει εχέγγυα αξιοπιστίας ή διαθεσιμότητας όταν παρέχεται σε εθελοντική βάση η εμπειρία των τελευταίων ετών έχει δείξει πως μπορεί να αποτελέσει ένα χρήσιμο συμπληρωματικό βοήθημα για όσους χρησιμοποιούν το Internet για εκπαιδευτικούς σκοπούς. Δε μπορεί φυσικά να παράσχει ολοκληρωμένη γνώση, αλλά έχει αποδειχθεί αρκετά αποτελεσματική στη διευθέτηση συγκεκριμένων αποριών ή την αναζήτηση πολύ ειδικών θεμάτων. Για αυτό και η επί αμοιβή παροχή της φαίνεται πως έχει μεγάλο και κερδώο μέλλον.

Η προσωπική επικοινωνία με εξειδικευμένο εκπαιδευτή είναι πιο γνωστή με το όνομα εκπαίδευση από απόσταση και για τους περισσότερους συμβολίζει τις πραγματικές δυνατότητες του Internet και το δρόμο προς τον οποίο αναπόφευκτα κατευθύνεται η εκπαιδευτική διαδικασία. Ο όρος εκπαίδευση από απόσταση χρησιμοποιείται συνήθως υπό την πολύ ευρεία έννοια του και αναφέρεται σε κάθε μορφή διδασκαλίας όπου χάρη στη χρήση της τεχνολογίας, ο εκπαιδευτής και οι εκπαιδευόμενοι συμμετέχουν στο μάθημα από διαφορετικούς χώρους όπως έχει ήδη προαναφερθεί για τη Διαδικτυακή εκμάθηση.

Οι μαθητές αλλά και οι δάσκαλοι τους, διαθέτουν ο καθένας διαφορετικούς τρόπους με τους οποίους μαθαίνουν, με την έννοια πως έχουν προτιμώμενες οδούς για την πρόσληψη και την επεξεργασία των πληροφοριών. Μερικοί ενδιαφέρονται για γεγονότα, δεδομένα και αλγορίθμους, άλλοι αισθάνονται περισσότερο άνετα με θεωρίες και μαθηματικά μοντέλα. Κάποιοι επιθυμούν την οπτικοποιημένη παρουσίαση των πληροφοριών (εικόνες, σχήματα, διαγράμματα), άλλοι προτιμούν τις προφορικές παρουσιάσεις και τις γραπτές ή προφορικές ερμηνείες. Τέλος υπάρχουν αυτοί που εμπιστεύονται περισσότερο τις διαδικασίες όπου υπάρχει δυνατότητα για δράση και αλληλοπάθεια και εκείνοι που έλκονται από μία εσωστρεφή και εξατομικευμένη αντιμετώπιση της μαθησιακής διαδικασίας.

Σε αυτήν τη γνωστική περιπέτεια, στην οποία υποτίθεται πως όλοι θα πρέπει να συμμετέχουν με ενδιαφέρον και αποτελεσματικότητα, εμπλέκονται πολυσχιδείς παράγοντες που σχετίζονται με καθαρά βιολογικά υποκείμενα, (εγκέφαλος, λειτουργία αισθήσεων, χρονοβιολογία κ.τ.λ.π.), με ψυχολογικές παραμέτρους και με κοινωνικές καταστάσεις οι οποίες αφορούν τόσο τους εκπαιδευόμενους όσο και τους εκπαιδευτές τους. Πρόκειται για ένα ζήτημα υψηλής πολυπλοκότητος που γίνεται ακόμη πιο σύνθετο, εάν δεν εξασφαλισθεί η συνδρομή όλων των εμπλεκομένων τους οποίους αντιλαμβανόμαστε εδώ ωσάν βιο - ψυχο - κοινωνικά γνωστικά συστήματα που αυτοοργανώνονται. Εάν στα παραπάνω προστεθεί η αναγκαία, όπως φαίνεται σήμερα, συνύπαρξη με τους υπολογιστές και την επαναστατική λογική που συνοδεύει πολλές από τις εφαρμογές τους, που είναι κατά κανόνα μη γραμμικές, τότε αντιλαμβανόμαστε πόσο ενδιαφέρον αποκτά η γνώση από τον καθένα των τρόπων με τους οποίους προτιμά να προσλαμβάνει ή και να μεταδίδει τις πληροφορίες, να πλοηγείται με άλλα λόγια και να αλληλεπιδρά στο Διαδίκτυο.



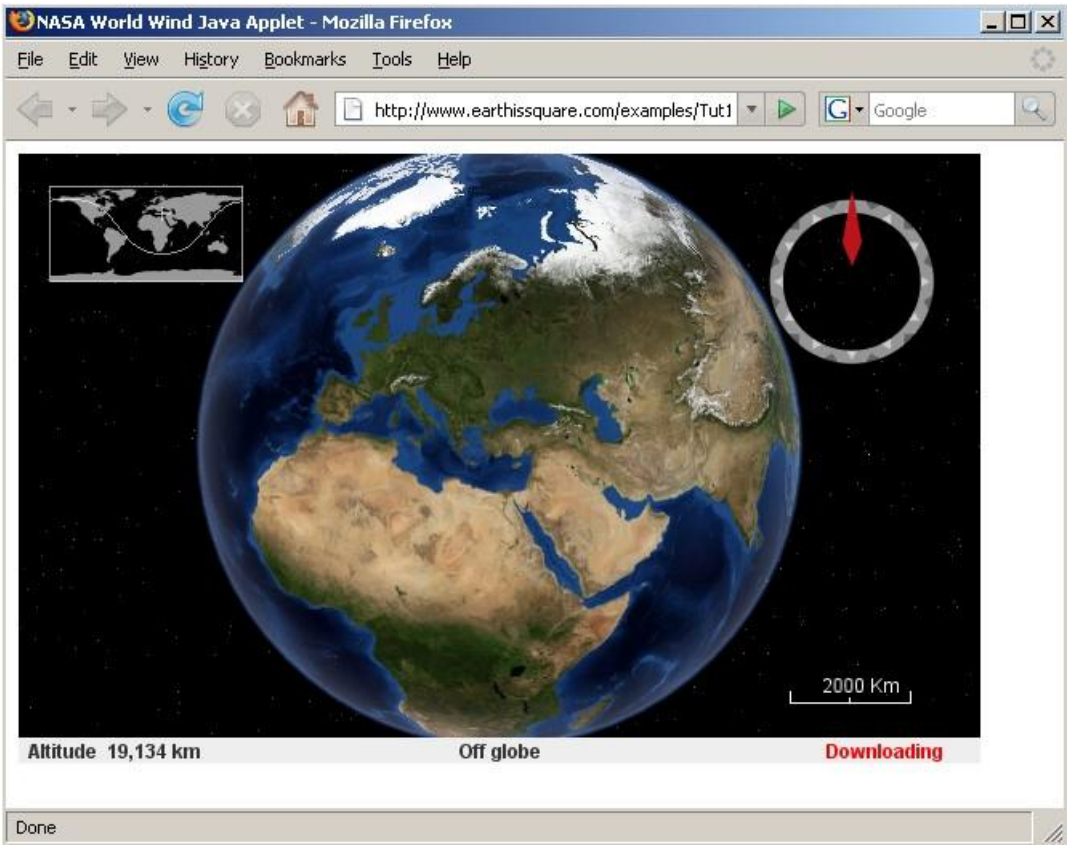
Σχήμα 2.9 : Στο Διαδίκτυο ψάχνουμε συνήθως όλοι σήμερα για απαντήσεις σε απορίες καθόσον αυτό αποτελεί μία ανεξάντλητη κρήνη γνώσης.

Τελικό συμπέρασμα είναι πως επιβάλλεται να αναγνωρίσουμε τα όρια, τους περιορισμούς αλλά και τις δυνατότητες που παρέχει το Διαδίκτυο με τελικό στόχο όχι την επιβίωση αλλά τη λειτουργική συνοίκηση μαζί του, αναγνωρίζοντάς το ως ένα δυναμικό, μη γραμμικό και συνεχώς εξελισσόμενο γνωστικό πληροφοριακό περιβάλλον. Στο περιβάλλον αυτό είμαστε αναγκασμένοι να πλοηγηθούμε, αποφασίζοντας ο καθένας, άλλοτε μόνος του και άλλοτε ως μέλος μίας συγκροτημένης ομάδας, τις στρατηγικές επιλογές μας και τις τακτικές μας κινήσεις, μέσα σε έναν κόσμο που χαρακτηρίζεται από πολυπλοκότητα και ασάφεια και για τον οποίο οι διαδικασίες και οι κανόνες λειτουργίας συγκροτούνται και αναδιατάσσονται αδιάλειπτα.

2.2 Εκπαίδευση με εφαρμογές των Java Applets

Ένα Applet ή αλλιώς Java Applet, είναι ένα πρόγραμμα το οποίο είναι γραμμένο στην κορυφαία αυτήν τη στιγμή γλώσσα προγραμματισμού στον πλανήτη δηλαδή στη γλώσσα Java. Το Applet περιλαμβάνεται πάντοτε σε ιστοσελίδα και παρουσιάζεται στο Internet. Αν και ο κώδικας της ιστοσελίδας είναι εμφανής σε περίπτωση που κάποιος ψάξει για τον κώδικα της εντούτοις ο κώδικας του Applet είναι αδιόρατος για οποιονδήποτε με εξαίρεση τον προγραμματιστή του Applet. Ήτοι ο προγραμματιστής γράφει τον κώδικα της Java στον υπολογιστή του και τον τρέχει αλλού δηλαδή στο Διαδίκτυο. Επίσης πολύς κόσμος έχει ταυτίσει εσφαλμένως την ορολογία του Applet με το μήλο. Το Applet σημαίνει μικροεφαρμογή και όχι μήλο ενώ χρησιμοποιείται κυρίως για διαχείριση γραφικών στο Web όπως αντίστοιχα και το Java Swing για προγραμματισμό με γραφικά αλλά σε τοπικούς υπολογιστές. Πρέπει επίσης να αναφερθεί ότι το Applet τρέχει στο Web πάντα από την πλευρά του πελάτη.

Τα Applets χρησιμοποιούνται για να παρέχουν διαδραστικές λειτουργίες σε Web εφαρμογές που δε μπορούν να παρέχονται από HTML γλώσσα και μόνο. Μπορούν μέσω του ποντικιού να μετακινούν αντικείμενα, να ζωγραφίζουν και γενικά έχουν πάμπολλες δυνατότητες με το ποντίκι (και όχι μόνο με αυτό) τις οποίες δεν τις έχει η γλώσσα HTML και αρκετές από αυτές δεν τις έχει ούτε η δυναμική γλώσσα από την πλευρά του πελάτη - client η JavaScript. Το γεγονός αυτό καθιστά τα Applets ολοσχερώς κατάλληλα για την επίδειξη, την απεικόνιση και τη διδασκαλία. Υπάρχουν σε απευθείας σύνδεση συλλογές Applets για τη μελέτη διαφόρων θεμάτων, από τη Φυσική μέχρι την Ιατρική και κυρίως σε αυτόν τον τομέα για τη φυσιολογία της καρδιάς. Επιπρόσθετα τα Applets χρησιμοποιούνται ευρέως από τη NASA ενώ ακόμη εφαρμόζονται για να δημιουργήσουν σε απευθείας σύνδεση συλλογές παιχνιδιών όπου επιτρέπεται στους παίκτες να ανταγωνιστούν με τους αντιπάλους τους σε πραγματικό χρόνο.

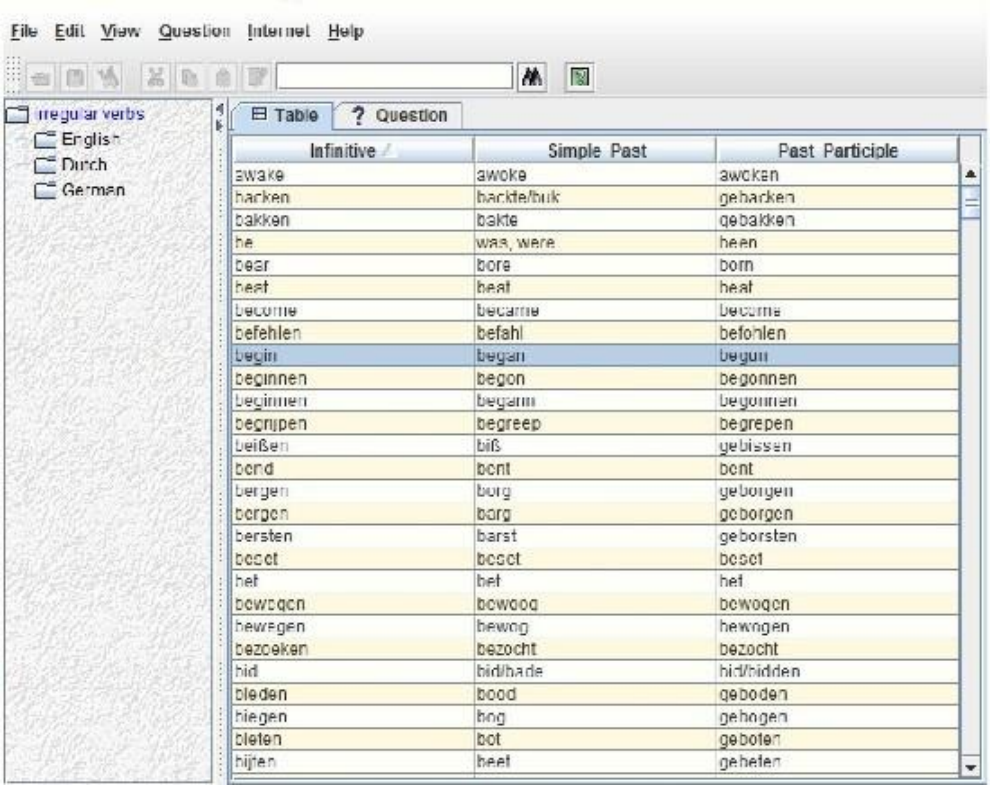


Σχήμα 2.10 : Χρήση του Applet από τη NASA όπου με το Applet επισκέπτεσαι οποιαδήποτε μεριά της γης μέχρι και σε πολύ χαμηλό ύψος.

Οι πρόσφατες εξελίξεις όσον αφορά την κωδικοποίηση των εφαρμογών, συμπεριλαμβανομένων των κινητών και των ενσωματωμένων συστημάτων έχουν οδηγήσει στη συνειδητοποίηση της ασφάλειας των Applets. Ας υποθέσουμε πως κάποιος έχει δημιουργήσει ένα Java Applet στον Ιστό. Ο κώδικας byte αυτού του Applet εκτελείται στο φυλλομετρητή που βρίσκεται στον υπολογιστή ενός αναγνώστη. Δηλαδή το Applet αυτού του κάποιου εκτελείται στους υπολογιστές άλλων. Πανομοιότυπα τα Applets που έχουν σχεδιασθεί από άλλους προγραμματιστές εκτελούνται και στον υπολογιστή αυτού του κάποιου. Είναι καθολική διαπίστωση πως όταν εκτελούνται προγράμματα άλλων χρηστών στον υπολογιστή οποιουδήποτε υπάρχει κίνδυνος ασφάλειας. Ένας πιθανός κίνδυνος είναι ότι το πρόγραμμα μπορεί να μεταδώσει κάποιον ιό στους υπολογιστές. Άλλος κίνδυνος είναι μήπως το πρόγραμμα είναι ικανό να διαβάσει εμπιστευτικές πληροφορίες οι οποίες βρίσκονται αποθηκευμένες στα αρχεία των υπολογιστών. Ακόμη υπάρχει φόβος μήπως το πρόγραμμα εσκεμμένα επιφέρει δυσεπίλυτα προβλήματα στο λειτουργικό σύστημα των υπολογιστών. Ευτυχώς όμως το Applet είναι εξακριβωμένο ότι

παρέχει ασφάλεια και δε δύναται να προβεί σε καμία τέτοιου είδους ενέργεια και ούτε μπορεί να διαβάσει ή να γράψει κάποιο από τα αρχεία των υπολογιστών.

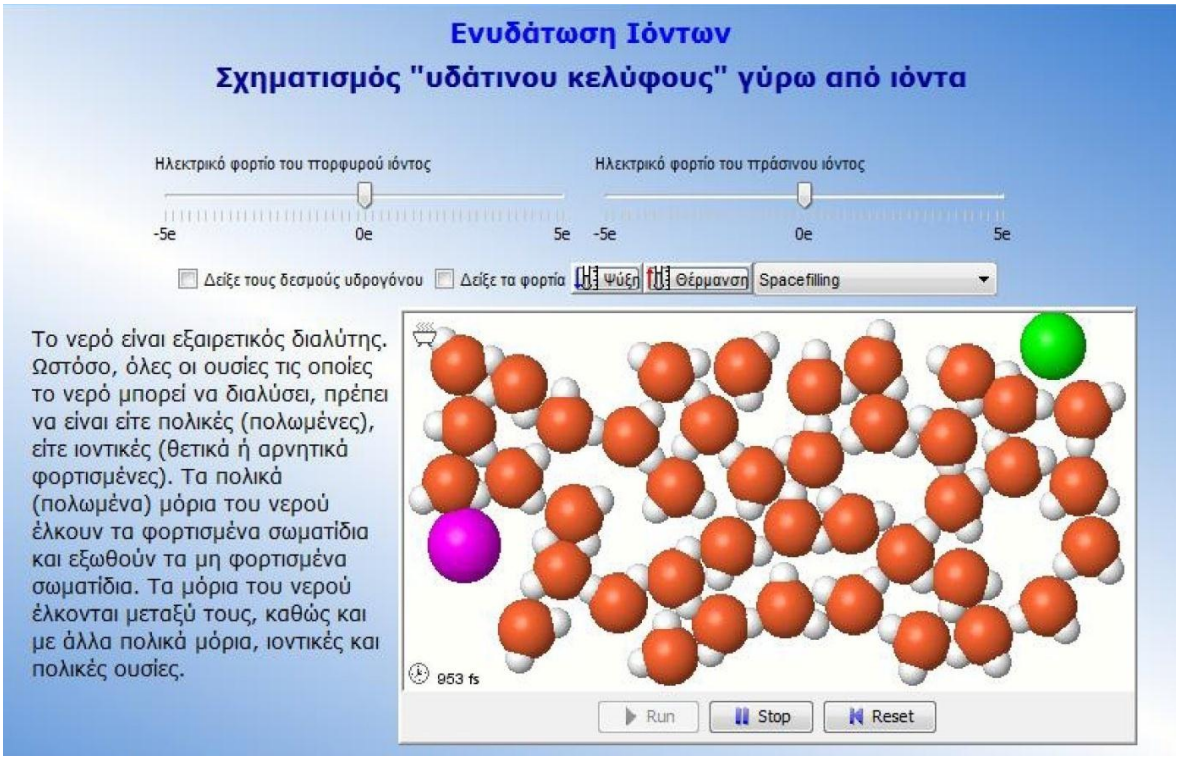
Ως εκπαιδευτικά μέσα, τα Java Applets επηρεάζουν τη διδασκαλία των μαθημάτων μεταφέροντας την εστίαση τους από τις ξεκομμένες δεξιότητες στην περιοχή της διερεύνησης των σχέσεων μεταξύ των εννοιών του εκάστοτε μαθήματος. Ως εκ τούτου, τα τελευταία χρόνια πολλά εκπαιδευτικά ιδρύματα στρέφουν την προσοχή τους στην παραγωγή portals με μαθησιακά εργαλεία τα οποία επιτρέπουν στους εκπαιδευτικούς την online χρήση Applets και άλλων υλικών που παρέχουν μάθηση. Η ένταξη των Applets στην εκπαιδευτική διαδικασία εκτός από το ότι την εμπλουτίζει με ένα νέο μέσο που είναι ενδιαφέρον και προσελκύει τους μαθητές, δίνει τη δυνατότητα να υλοποιηθεί διερευνητική εκμάθηση. Έτσι οι μαθητές έχουν ένα εργαλείο πειραματισμού και μόρφωσης, χωρίς περιορισμούς χρόνου και τόπου, το οποίο συμβάλλει στην κατανόηση εννοιών και φαινομένων και στη διατύπωση γόνιμων αποριών.



Σχήμα 2.11 : Applet το οποίο διδάσκει τους χρόνους της αγγλικής, γερμανικής και ολλανδικής γλώσσας.

Οι εκπαιδευτικοί από την άλλη αποκτούν ρόλο συντονιστή, καθοδηγητή, οργανωτή και ισότιμα συμμετέχοντα στη διαδικασία της πνευματικής κατάρτισης. Τα ικανοποιητικά σχεδιασμένα προγράμματα προσομοίωσης Java Applets μπορούν να παρέχουν στους μαθητές ένα κανάλι για να συζητήσουν τις έννοιες της μάθησης μέσα σε ένα αλληλεπιδραστικό περιβάλλον εκμάθησης. Στο πλαίσιο αυτών οι μαθητευόμενοι δεν εκλαμβάνονται ως παθητικοί δέκτες, αλλά ως αυτοτελή και υπεύθυνα άτομα τα οποία συμμετέχουν ενεργά στη διαδικασία της διαπαιδαγώγησης, υποβοηθούμενα από τις δυνατότητες της αλληλεπιδραστικής υπολογιστικής τεχνολογίας. Η έννοια της ενεργητικής μύησης προϋποθέτει ότι ο μαθητής θα αναπτύξει ένα ιδιαίτερο είδος ψυχολογικής σχέσης με τη διαδικασία και το περιεχόμενο. Η ρυθμιστική παρέμβαση του εκπαιδευτικού συντελείται στο πλαίσιο μίας διηνεκούς υποστήριξης των μαθησιακών επιδιώξεων, με ταυτόχρονη σταδιακή απόσυρση του ελέγχου και της υποστήριξης από το σύστημα και τον εκπαιδευτικό με στόχο την αυτόνομη μάθηση.

Παράλληλα η γραφιστική προσέγγιση στην αλληλεπίδραση, έχει βαρυσήμαντο ρόλο στα μαθησιακά κίνητρα. Οι μαθητές συχνά έλκονται από την καινοτομία των αλληλεπιδραστικών πολυμέσων. Η διάταξη της πληροφορίας στην οθόνη και ο τρόπος παρουσίασης επηρεάζουν σοβαρά την κατεύθυνση, τη διάρκεια και το βαθμό προσοχής του μαθητή. Στο πλαίσιο αυτό το χρώμα, η ταχύτητα, η κίνηση και ο ήχος, μπορούν να επηρεάσουν την αλληλεπιδραστική διαδικασία. Ακόμη οι αναπαραστάσεις που εξασφαλίζουν τα Applets αποτελούν σημαντικό συστατικό εννοιών καθόσον έχουν πρακτικό προσανατολισμό. Ένα παράδειγμα αποτελεί η παρουσίαση του τρόπου με τον οποίο αλλάζει το εμβαδό χωρίου καμπύλης καθώς αυξάνουν οι διαμερίσεις, κάτι το οποίο σε περίπτωση που δεν το δουν οι μαθητές να συμβαίνει, δυσκολεύονται να το αντιληφθούν. Συν τοις άλλοις, γίνεται εύκολα προσέγγιση συναφών αντικειμένων με άμεση αναφορά τους μέσα στο ίδιο Applet, δίχως να διακόπτεται έτσι η ροή και η καθαρότητα της διδασκαλίας, όπως συμβαίνει λόγου χάρη με τα συχνά «βάλε - βγάλε» των διαφανειών ή το «σβήσε - γράψε» του πίνακα.



Σχήμα 2.12 : Στα διαλύματα τα μόρια του νερού σχηματίζουν ένα κέλυφος γύρω από τα διαλυμένα ιόντα. Σε αυτό το Applet γίνεται προσομοίωση της κίνησης των διαλυμένων ιόντων ή αλλιώς ενυδατωμένων ιόντων.

Το GUI σημαίνει Γραφική Διεπαφή Χρήστη - Graphical User Interface. Ήτοι η έννοια του GUI είναι τα παραθυρικά συστήματα που αλληλεπιδρούν με τους χρήστες ή απλούστερα τα γραφικά της γλώσσας Java. Όπως έχει ήδη προαναφερθεί Swing GUI είναι τα γραφικά που προγραμματίζονται μόνο για τοπικούς υπολογιστές ενώ τα Java Applets είναι τα ίδια σχεδόν γραφικά τα οποία προγραμματίζονται αποκλειστικά για το Διαδίκτυο. Γενικά τα προγράμματα της γλώσσας Java διαιρούνται σε δύο κατηγορίες εφαρμογές και μικροεφαρμογές. Τις μικροεφαρμογές τις αποτελούν τα Java Applets ενώ τις εφαρμογές οποιαδήποτε άλλου είδους προγραμματισμός με τη γλώσσα Java συγκαταλέγοντας φυσικά σε αυτές και το Java Swing.

Οι εφαρμογές πρέπει να περιέχουν πάντοτε τη μέθοδο main. Ετούτη η μέθοδος είναι επιφορτισμένη με το να ξεκινάει την εκτέλεση κάθε προγράμματος της Java το οποίο είναι εφαρμογή. Σε αντιδιαστολή τα Java Applets χρησιμοποιούν αντί της

μεθόδου main για να τρέξουν τη μέθοδο init. Στη μέθοδο init γίνονται οι αρχικοποιήσεις που θα γίνονταν από τους δημιουργούς ενός τυπικού Swing GUI. Ο ορισμός μίας κλάσης μικροεφαρμογής δηλαδή Applet δεν ενσωματώνει δημιουργούς. Επιπρόσθετα τα Applets δεν περιέχουν τις μεθόδους setTitle, setVisible και setSize τις οποίες δύναται να χρησιμοποιήσει το Swing. Στις μικροεφαρμογές αφού μπαίνουν σε HTML αρχείο μπορεί να προστεθεί οποιοσδήποτε τίτλος από τη γλώσσα HTML ενώ ακόμη η γλώσσα HTML είναι αυτή που ρυθμίζει το μέγεθος του Applet. Επίσης οι μικροεφαρμογές παρουσιάζονται αυτομάτως σε μία ιστοσελίδα ή σε ένα πρόγραμμα επισκόπησης μικροεφαρμογών και για τους λόγους αυτούς είναι άσκοπη η χρήση των προειρημένων μεθόδων στα Applets. Προσέτι οι μικροεφαρμογές δεν έχουν κουμπί για το κλείσιμο του παραθύρου με συνέπεια να μη διαθέτουν τη μέθοδο setDefaultCloseOperation που χρησιμοποιεί το Swing. Όταν κλείνει το έγγραφο HTML το οποίο εμπεριέχει τη μικροεφαρμογή τότε το Applet κλείνει ομόχρονα.

Για να εισχωρήσει ένα Applet στο αρχείο HTML πρέπει να γραφτεί στο αρχείο αυτό ο παρακάτω κώδικας:

<applet code = "Όνομα κλάσης.class" width = ένας ακέραιος αριθμός height = ένας ακέραιος αριθμός </applet>

Αυτή η ετικέτα της μικροεφαρμογής προϋποθέτει ότι το αρχείο HTML και η κλάση της Java βρίσκονται στον ίδιο φάκελο. Εάν δεν ισχύει αυτό τότε πρέπει να χρησιμοποιηθεί ένα πλήρες όνομα διαδρομής για την κλάση της Java. Εδώ παρατηρούμε ότι δε δίνεται η ονομασία του αρχείου της Java που έχει κατάληξη .java αλλά αυτού που έχει την κατάληξη .class. Όσο για τα width και height είναι εκείνα τα οποία καθορίζουν το μέγεθος της μικροεφαρμογής. Το width θέτει το πλάτος της μικροεφαρμογής σε εικονοστοιχεία (pixels) ενώ το height το ύψος του Applet ομοίως σε εικονοστοιχεία.

Για να οριστεί ένα Applet ο κώδικας της Java πρέπει να είναι ο εξής:

public class Όνομα κλάσης extends JApplet { }

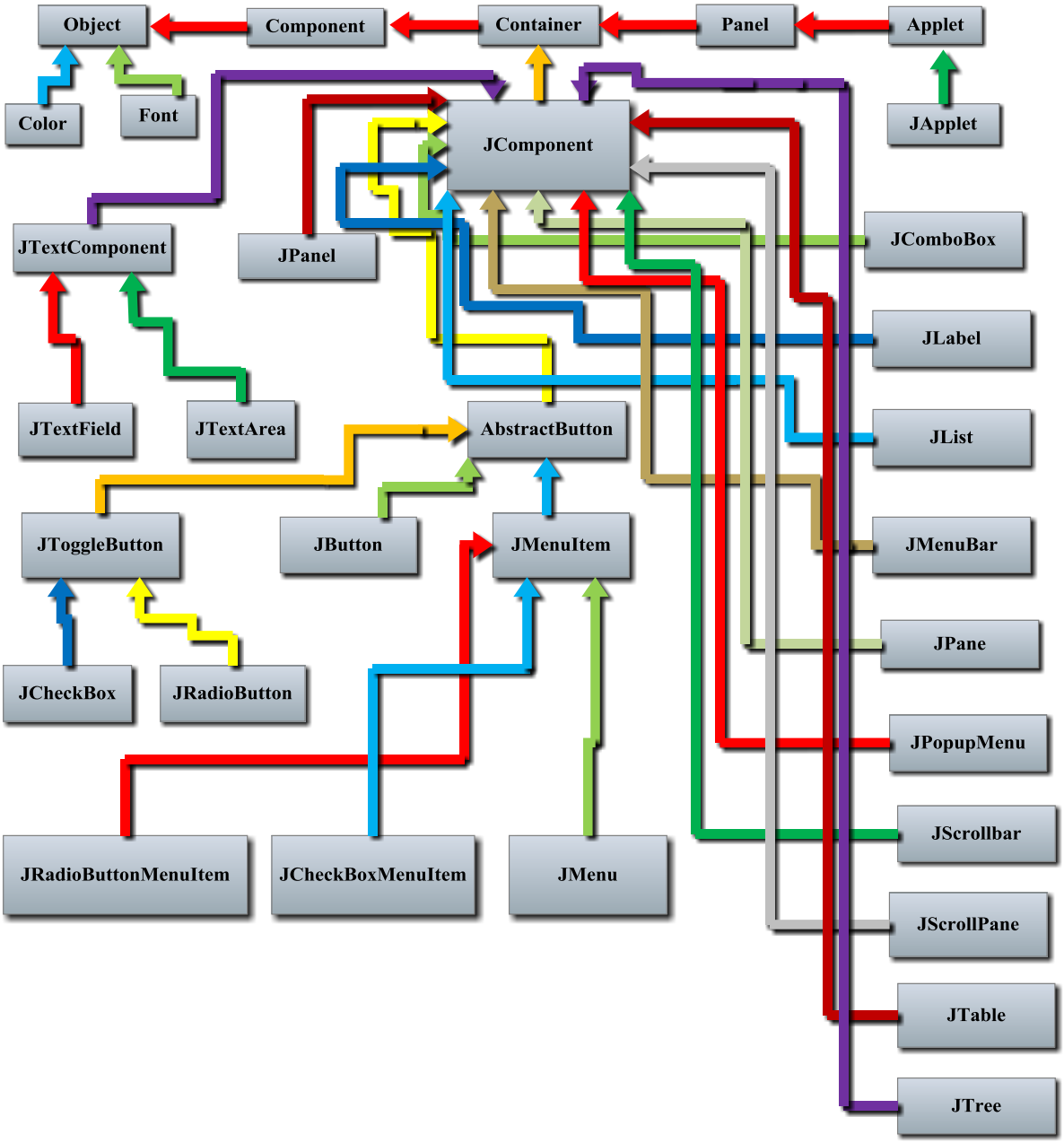
Κατόπιν μέσα στα άγκιστρα της κλάσης που ορίστηκε πρέπει να δημιουργηθεί η μέθοδος init:

```
public void init()
{
        Container a = getContentPane();
        a.setLayout(null);
}
```

Στη θέση του a μπορεί να μπει οποιαδήποτε άλλη μεταβλητή. Η κλάση Container είναι μία κλάση που ανήκει στο πακέτο java.awt το οποίο πρέπει να δηλωθεί στην αρχή του κώδικα. Την κλάση αυτή μπορούμε να τη φανταστούμε ως έναν υποδοχέα για στοιχεία διασύνδεσης χρήστη μέσω γραφικών (GUI components container). Στον κώδικα γίνεται λήψη αναφοράς στον υποδοχέα a. Στην αμέσως επόμενη γραμμή του κώδικα θέτουμε ότι ο υποδοχέας ήτοι ολόκληρη η μικροεφαρμογή θα έχει διαχειριστή διάταξης απόλυτο. Σε αυτήν την περίπτωση ο προγραμματιστής πρέπει κάθε φορά να προδιαγράφει τα όρια του κάθε συστατικού με τη μέθοδο setBounds. Αυτό μπορεί να φαίνεται πολύμοχθο ωστόσο δίνει τη δυνατότητα στον προγραμματιστή να βάλει σε όποιο ακριβώς σημείο εκείνος επιθυμεί τα συστατικά πράγμα που ίσως να μη γινόταν αταλαιπώρητα αν είχε προσδιοριστεί διαχειριστής διάταξης. Στην εργασία ετούτη ο διαχειριστής διάταξης έχει οριστεί ως απόλυτος. Όσο για το extends αυτό σημαίνει πως η κλάση που δημιουργούμε είναι υποκλάση της κλάσης JApplet με απαύγασμα η κλάση μας να έχει όλες τις δυνατότητες της κλάσης JApplet. Καθόσον η υποκλάση πάντοτε κληρονομεί τις δυνατότητες της υπερκλάσης της. Η κλάση JApplet ανήκει στο πακέτο javax.swing

το οποίο πρέπει ομοίως να δηλωθεί στην αρχή του κώδικα. Γενικά όλα τα πακέτα πρέπει να δηλώνονται στην αρχή του κώδικα. Η μέθοδος public void setLayout (LayoutManager x) είναι μία μέθοδος της κλάσης Container ενώ η κλάση LayoutManager ανήκει στο πακέτο java.awt και καθορίζει το διαχειριστή διάταξης. Η μέθοδος public void setBounds (int k, int m, int z, int b) αποτελεί μία μέθοδο που χρησιμοποιούν οι υποκλάσεις της κλάσης Component η οποία κλάση ανήκει στο πακέτο java.awt. Ετούτη η μέθοδος μετακινεί και αλλάζει το μέγεθος του συστατικού. Η νέα θέση της πάνω αριστερής γωνίας του δίνεται από τις συντεταγμένες k, m ενώ το νέο του μέγεθος προσδιορίζεται από τις παραμέτρους z και b όπου z είναι το πλάτος του συστατικού και b το ύψος του. Οι παράμετροι k, m, z, b είναι ακέραιοι αριθμοί και αναπαριστούν εικονοστοιχεία. Αν επιθυμούσαμε να ορίσουμε κάποιο διαχειριστή διάταξης ο οποίος είναι επιφορτισμένος στο να δίνει κατάλληλες διαστάσεις στα συστατικά μέρη του υποδοχέα και να τα εισάγει σε ορισμένες θέσεις στην οθόνη απαλλάσσοντας τοιουτοτρόπως τον προγραμματιστή από το να προδιαγράφει τις λεπτομέρειες αυτές τότε έπρεπε να συντάξουμε τον εξής κώδικα:

```
LayoutManager d = new FlowLayout();
a.setLayout(d);
```



Σχήμα 2.13 : Στην εικόνα φαίνεται η ιεραρχία των βασικών κλάσεων που χρησιμοποιούνται στα Java Swings και τα Java Applets. Η υπερκλάση είναι το κουτί της εικόνας στο οποίο πάει το βέλος και η υποκλάση είναι το κουτί της εικόνας από όπου προέρχεται το βέλος.

Στην περίπτωση αυτή έχουμε ορίσει ως διαχειριστή διάταξης τον FlowLayout. Ετούτος ο διαχειριστής διάταξης είναι ο απλούστερος και τοποθετεί τα συστατικά στον υποδοχέα από αριστερά προς τα δεξιά, με τη σειρά που αυτά προστίθενται σε αυτόν ενώ είναι ο εξ ορισμού διαχειριστής διάταξης για τον υποδοχέα JPanel και ανήκει στο πακέτο java.awt. Όταν φτάσει στο δεξιό όριο του υποδοχέα συνεχίζει να προσθέτει συστατικά από την επόμενη σειρά του. Η κλάση FlowLayout επιτρέπει στα συστατικά να είναι στοιχισμένα προς τα αριστερά, στο κέντρο ή προς τα δεξιά. Πρέπει να επισημανθεί ότι είναι απαραίτητο πάντοτε να ορίζουμε διαχειριστή διάταξης ή να τον θέτουμε ως απόλυτο εφόσον δημιουργούμε αντικείμενο της κλάσης Container το οποίο το εξισώνουμε με getContentPane() όπως στα applets αυτής της εργασίας ώστε να γίνει συστηματοποιημένος ο προγραμματισμός των applets. Ακόμη αρμόζει να σημειωθεί ότι έτσι όπως έχει αρχικοποιηθεί ο διαχειριστής διάταξης FlowLayout, η

στοίχιση των συστατικών θα γίνεται στο κέντρο. Τα συστατικά στοιχίζονται προς τα αριστερά αν γίνει αρχικοποίηση όπως:

LayoutManager d = new FlowLayout(FlowLayout.LEFT);

Ομοίως αν θέλαμε να κάνουμε στοίχιση των συστατικών στο κέντρο θα θέταμε:

LayoutManager d = new FlowLayout(FlowLayout.CENTER);

Αν επιθυμούσαμε στοίχιση προς τα δεξιά έπρεπε να γράψουμε:

LayoutManager d = new FlowLayout(FlowLayout.RIGHT);

Οι παράμετροι FlowLayout.LEFT, FlowLayout.RIGHT και FlowLayout.CENTER είναι τύπου integer τουτέστιν αποτελούν ακέραιες τιμές. Όσο για την κλάση JPanel ανήκει στο πακέτο javax.swing. Η κλάση αυτή χρησιμοποιείται όταν επιδιώκουμε να δημιουργήσουμε μεγάλα παράθυρα με πολλά στοιχεία ώστε να έχουμε τη δυνατότητα να χωρίσουμε τον υποδοχέα σε πολλούς μικρότερους και να ορίσουμε τον κατάλληλο διαχειριστή διάταξης σε καθέναν από αυτούς τους μικρούς υποδοχείς ξεχωριστά. Τοιουτοτρόπως με τα πανό παρέχεται μεγάλη ευελιξία σχεδίασης προγραμμάτων με γραφικά ενώ τα αντικείμενα της κλάσης JPanel είναι εξ ορισμού αδιαφανή. Οι βασικοί κατασκευαστές των κλάσεων FlowLayout και JPanel είναι οι εξής:

- public FlowLayout()
  Με αυτόν τον κατασκευαστή δημιουργείται ο διαχειριστής διάταξης FlowLayout με τα συστατικά όπως προαναφέρθηκε να διατάσσονται στο κέντρο.

- public FlowLayout(int x)
  Με αυτόν τον κατασκευαστή δημιουργείται ο διαχειριστής διάταξης FlowLayout και όπως μνημονεύτηκε πριν τα συστατικά διατάσσονται δεξιά ή αριστερά ή στο κέντρο.

- public FlowLayout(int x,int y,int z)
  Με αυτόν τον κατασκευαστή δημιουργείται ο διαχειριστής διάταξης FlowLayout και στη θέση του x θέτουμε αν τα συστατικά θα στοιχηθούν στο κέντρο, στα δεξιά ή στα αριστερά όπως προηγουμένως. Στις θέσεις των y και z βάζουμε δύο ακέραιους αριθμούς που αναπαριστούν εικονοστοιχεία και καθορίζουν αντιστοίχως το οριζόντιο και κατακόρυφο διάκενο μεταξύ των στοιχείων. Σε κάθε περίπτωση όπου δεν προδιαγράφουμε τα διάκενα χρησιμοποιούνται αυτομάτως διάκενα των πέντε εικονοστοιχείων.

- public JPanel()
  Με αυτόν τον κατασκευαστή δημιουργείται ένα πανό με διαχειριστή διάταξης τον FlowLayout.

- public JPanel(LayoutManager a)
  Με αυτόν τον κατασκευαστή δημιουργείται ένα πανό με ένα συγκεκριμένο διαχειριστή διάταξης.

Αν δεν ορίσουμε συγκεκριμένο διαχειριστή διάταξης για κάποιον υποδοχέα, εφαρμόζεται ο διαχειριστής διάταξης BorderLayout. Ο διαχειριστής αυτός μας επιτρέπει να ορίσουμε τη θέση του στοιχείου με βάση τις θέσεις north (βόρεια), south (νότια), east (ανατολικά), west (δυτικά) και center (κεντρικά). Τα συστατικά τα οποία τοποθετούνται north και south επεκτείνονται οριζοντίως μέχρι τις πλευρές του υποδοχέα. Το ύψος των περιοχών αυτών είναι όσο και το ύψος των συστατικών που περιέχουν. Οι περιοχές east και west επεκτείνονται κατακόρυφα και έχουν

πλάτος ίσο με το πλάτος των συστατικών που τοποθετούνται σε αυτές. Το συστατικό που τίθεται στην περιοχή center διευρύνεται τοιουτοτρόπως ώστε να γεμίσει όλο τον υπόλοιπο χώρο του υποδοχέα. Αν και οι πέντε περιοχές είναι κατειλημμένες τότε προφανώς όλος ο χώρος είναι κορεσμένος με συστατικά. Σε περίπτωση όπου οι περιοχές north ή south δεν είναι κατειλημμένες, τα συστατικά στις περιοχές east, center και west αναπτύσσονται κατακόρυφα έτσι ώστε να γεμίσουν το χώρο που απομένει. Αν οι περιοχές east ή west δεν είναι πιασμένες το συστατικό στην περιοχή center επεκτείνεται οριζόντια και καταλαμβάνει την έκταση την οποία εναπομένει. Τέλος άμα η περιοχή center δεν είναι κατειλημμένη τότε αυτή παραμένει κενή καθόσον τα υπόλοιπα συστατικά δεν εξαπλώνονται για να συμπληρώσουν ετούτο το άδειο από συστατικά μέρος. Ο διαχειριστής διάταξης BorderLayout ανήκει στο πακέτο java.awt ενώ έχει τους εξής κατασκευαστές:

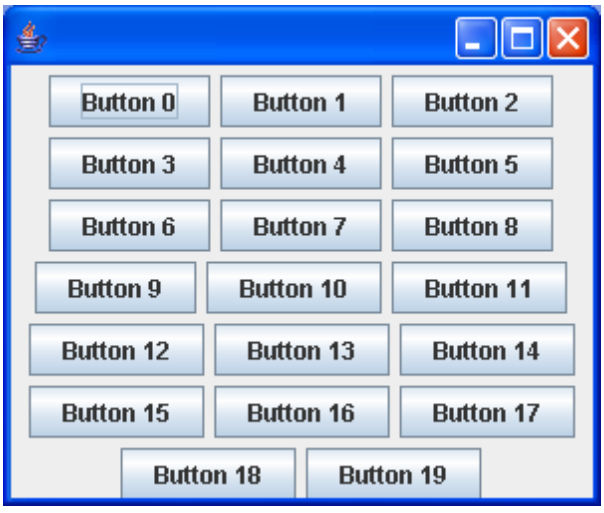
- public BorderLayout()
  Αυτός ο κατασκευαστής δημιουργεί ένα διαχειριστή διάταξης BorderLayout χωρίς κενά μεταξύ των συστατικών.

- public BorderLayout(int x,int y)
  Ετούτος ο κατασκευαστής δημιουργεί ένα διαχειριστή διάταξης BorderLayout όπου η ακέραιη τιμή του x δηλώνει σε εικονοστοιχεία το οριζόντιο διάκενο μεταξύ των συστατικών ενώ η ακέραιη τιμή του y δηλώνει το κατακόρυφο διάκενο μεταξύ των συστατικών ομοίως σε εικονοστοιχεία.

Η προσθήκη ενός στοιχείου σε έναν υποδοχέα Container στον οποίο έχουμε θέσει ως διαχειριστή διάταξης ένα αντικείμενο του τύπου BorderLayout επιτυγχάνεται με την εντολή:

Container.add(συστατικό,περιοχή);

Η περιοχή είναι μία από τις σταθερές BorderLayout.NORTH, BorderLayout.SOUTH, BorderLayout.EAST, BorderLayout.WEST ή BorderLayout.CENTER. Αν δεν προσδιοριστεί η περιοχή στην παραπάνω μέθοδο τότε το στοιχείο τοποθετείται αυτομάτως στην περιοχή BorderLayout.CENTER. Η προσάρτηση περισσότερων του ενός συστατικού σε μία ορισμένη περιοχή έχει ως αποτέλεσμα να αναφύεται μονάχα το τελευταίο στοιχείο που προστέθηκε. Υπάρχουν και άλλοι διαχειριστές διάταξης όπως ο GridLayout, ο GridBagLayout και ο CardLayout αλλά θα μελετήσουμε μόνο έναν ακόμη τον GridLayout.



Σχήμα 2.14 : Στην εικόνα τα γραφικά στοιχεία τα τοποθετεί ο διαχειριστής διάταξης FlowLayout. Τα συστατικά εγκαθίστανται από το διαχειριστή διάταξης σε γραμμές από τα αριστερά προς τα δεξιά χωρίς καμία στοίχιση.

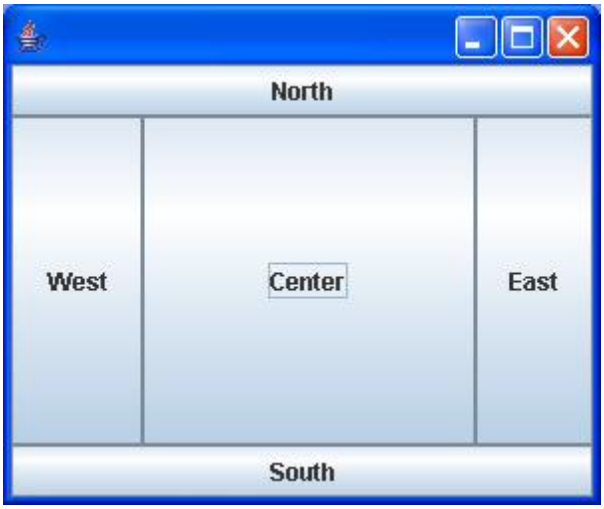
Ο διαχειριστής διάταξης GridLayout διαιρεί έναν υποδοχέα σε πλαίσια έτσι ώστε τα συστατικά που προστίθενται σε αυτόν να δύνανται να τοποθετηθούν οριζόντια σε γραμμές και κατακόρυφα σε στήλες. Όλα τα συστατικά τα οποία εγκαθίστανται σε υποδοχέα με διαχειριστή διάταξης τον GridLayout έχουν το ίδιο πλάτος καθώς και το ίδιο ύψος. Τα στοιχεία προστίθενται αρχίζοντας από το πάνω αριστερά κελί του πλαισίου και συνεχίζουν αριστερά προς τα δεξιά μέχρι να συμπληρωθεί η πρώτη γραμμή. Η διαδικασία ετούτη εξακολουθεί για τη δεύτερη γραμμή κ.ο.κ. Ο διαχειριστής διάταξης GridLayout ανήκει στο πακέτο java.awt και έχει τους ακόλουθους κατασκευαστές:

- public GridLayout()
  Αυτός ο κατασκευαστής δημιουργεί ένα διαχειριστή διάταξης GridLayout όπου τα συστατικά θέτονται σε μία στήλη ανά συστατικό σε μία γραμμή.

- public GridLayout(int x,int y)
  Αυτός ο κατασκευαστής δημιουργεί ένα διαχειριστή διάταξης GridLayout όπου η τιμή του x εκφράζει έναν ακέραιο αριθμό ο οποίος καθορίζει τον αριθμό των γραμμών όπου θα τοποθετηθούν τα συστατικά ενώ η τιμή του y έναν ακέραιο αριθμό που προδιαγράφει τον αριθμό των στηλών.

- public GridLayout(int x,int y,int k,int z)
  Αυτός ο κατασκευαστής δημιουργεί ένα διαχειριστή διάταξης GridLayout ενώ οι τιμές των x και y παίζουν τον ίδιο ακριβώς ρόλο όπως στον προηγούμενο κατασκευαστή. Η τιμή του k είναι ένας ακέραιος αριθμός ο οποίος δηλώνει εικονοστοιχεία και προσδιορίζει τον οριζόντιο κενό χώρο μεταξύ των γραμμών. Ομοίως η τιμή του z είναι ένας ακέραιος αριθμός που εκφράζει εικονοστοιχεία και καθορίζει τον κατακόρυφο άδειο χώρο μεταξύ των στηλών.

Συνοψίζοντας με τους διαχειριστές διάταξης, ο προγραμματιστής μπορεί να χρησιμοποιήσει έναν ή περισσότερους διαχειριστές διάταξης. Δηλαδή έχει την πολυτέλεια να κάνει συνδυασμό των διαχειριστών διάταξης. Ωστόσο ο προγραμματιστής δύναται και να χρησιμοποιήσει διαχειριστή διάταξης απόλυτο και να προδιαγράφει εκείνος τα όρια του κάθε στοιχείου. Τοιουτοτρόπως υπάρχει αξιοθαύμαστη ευκινησία στην επιλογή της μεθόδου προγραμματισμού, γεγονός που επιβεβαιώνει για μία ακόμη φορά τη φήμη της γλώσσας Java ως την ασυναγώνιστη γλώσσα προγραμματισμού σε ολόκληρη την υφήλιο.



Σχήμα 2.15 : Στην εικόνα φαίνονται τα γραφικά στοιχεία όπως έχουν τεθεί από το

διαχειριστή διάταξης BorderLayout. Τα συστατικά βρίσκονται βόρεια (north), δυτικά (west), ανατολικά (east), νότια (south) και κεντρικά (center). Το συγκεκριμένο παράθυρο δεν είναι Applet δηλαδή δε δημιουργήθηκε από την κλάση JApplet αλλά από την κλάση JFrame. Το ίδιο ισχύει και για το σχήμα 14. Η κλάση JFrame είναι αυτή που παράγει γραφικά σε τοπικούς υπολογιστές περίπου όπως συνθέτει γραφικά η κλάση JApplet στο Διαδίκτυο.

Σε ένα Applet έχουμε τη δυνατότητα αν το επιθυμούμε να δημιουργήσουμε νέο παράθυρο με την κλάση JFrame. Δηλαδή δεν απαγορεύεται να ενσωματώσουμε σε ένα Applet τουτέστιν σε μία μικροεφαρμογή, μία εφαρμογή. Αυτό μπορεί να πραγματοποιηθεί για παράδειγμα με το που πατήσουμε ένα κουμπί στο Applet να εμφανιστεί ένα παράθυρο γραφικών το οποίο διαμορφώθηκε από την κλάση JFrame. Γενικά όμως τα γραφικά που δημιουργεί η κλάση JFrame προορίζονται για τοπικούς υπολογιστές και όχι για το Internet όπως έχει ήδη προαναφερθεί. Η τάξη JFrame παρέχει τις δυνατότητες και συμπεριφορές ενός παραθύρου δηλαδή μία μπάρα τίτλου και κουμπιά για ελαχιστοποίηση, μεγιστοποίηση και κλείσιμο του παραθύρου. Εδώ αξίζει να αναφερθεί πως η χρησιμοποίηση της κλάσης JFrame για τον προγραμματισμό γραφικών σε τοπικούς υπολογιστές είναι ο πιο ευρέως διαδεδομένος τρόπος για μηχανογραφήσεις τραπεζών, φαρμακείων, βιβλιοπωλείων και άλλα. Η κλάση JFrame ανήκει στο πακέτο javax.swing ενώ οι βασικοί της κατασκευαστές είναι οι παρακάτω:

- public JFrame()
  Ο κατασκευαστής αυτός δημιουργεί ένα παράθυρο το οποίο αρχικά είναι αόρατο. Για να γίνει ορατό το παράθυρο πρέπει να χρησιμοποιηθούν οι μέθοδοι setSize και setVisible όπου με την πρώτη μέθοδο δίνουμε δύο ακέραιους αριθμούς οι οποίοι αναπαριστούν εικονοστοιχεία και εκφράζουν το πλάτος και το ύψος του παραθύρου. Στη δεύτερη μέθοδο πρέπει να δώσουμε την τιμή true η οποία είναι τύπου boolean και έτσι το παράθυρο γίνεται πλέον ορατό.

- public JFrame(String a)
  Ο κατασκευαστής αυτός παράγει ένα πλαίσιο που είναι αρχικά αόρατο και έχει τίτλο τη μεταβλητή a.

Για να προγραμματίσουμε ένα JFrame πρέπει να γράψουμε τον εξής κώδικα:

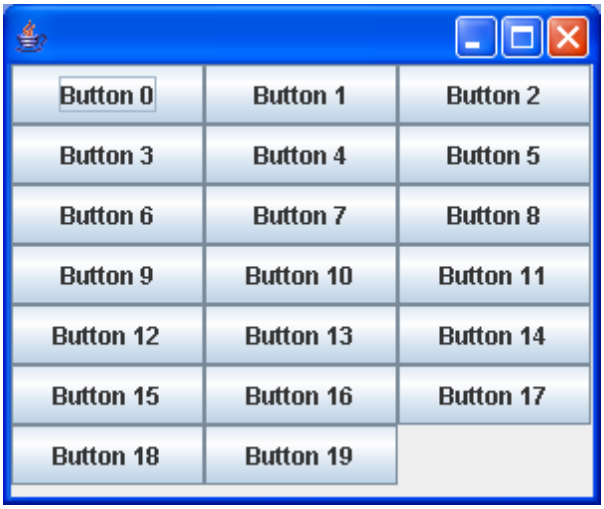
```
public class Όνομα κλάσης extends JFrame
{
}
```

Κατόπιν μέσα στα άγκιστρα της κλάσης που ορίστηκε πρέπει να δημιουργηθεί ο ακόλουθος κατασκευαστής:

```
public void Όνομα κλάσης ()

{

Container a = getContentPane();
```

Κατόπιν γράφουμε τα ίδια όπως στην κλάση JApplet ενώ πρέπει να χρησιμοποιήσουμε και τη μέθοδο main για να τρέξει η εφαρμογή.

```
}
```



Σχήμα 2.16 : Στην εικόνα φαίνονται τα γραφικά στοιχεία όπως έχουν τεθεί από το διαχειριστή διάταξης GridLayout. Και αυτό το παράθυρο δεν είναι Applet και έχει δημιουργηθεί από την κλάση JFrame. Τα συστατικά και πιο συγκεκριμένα τα κουμπιά έχουν τοποθετηθεί σε εφτά γραμμές και τρεις στήλες. Το τελευταίο στοιχείο προφανώς δεν είναι κουμπί απλά έχει αφεθεί ως κενός χώρος. Έτσι μπορούμε εύκολα να συμπεράνουμε πως αφού με ετούτον το διαχειριστή διάταξης είμαστε ικανοί να στοιχίζουμε τα συστατικά με τη μορφή ενός πίνακα, ο GridLayout μας χρησιμεύει υπεράγαν σε πληθώρα εφαρμογών.

Κάθε υποκλάση της κλάσης JApplet που επιθυμεί να χειριστεί ένα σημασιολογικό γεγονός πρέπει να υλοποιεί τη διεπαφή ActionListener η οποία ανήκει στο πακέτο java.awt.event και τη (μοναδική της) μέθοδο actionPerformed. Το αντικείμενο το οποίο δημιουργείται από την κλάση ActionListener εγγράφεται ακολούθως με ένα συστατικό με τη χρήση της μεθόδου addActionListener του συστατικού αυτού. Όταν συμβεί ένα σημασιολογικό γεγονός καλείται η μέθοδος actionPerformed. Το γεγονός αυτό μπορεί για παράδειγμα να είναι το πάτημα ενός αντικειμένου της τάξης JButton όπου η κλάση JButton ανήκει στο πακέτο javax.swing και δημιουργεί ένα παραλληλόγραμμο κουμπί.

Η κλάση JButton έχει τους εξής βασικούς κατασκευαστές:

- public JButton()
  Ο κατασκευαστής αυτός δημιουργεί ένα παραλληλόγραμμο κουμπί χωρίς να υπάρχει μέσα στο κουμπί εικόνα ή συμβολοσειρά.

- public JButton(String a)
  Ο κατασκευαστής αυτός δημιουργεί ένα παραλληλόγραμμο κουμπί όπου μέσα στο κουμπί υπάρχει η συμβολοσειρά a.

- public JButton(Icon a)
  Ο κατασκευαστής αυτός δημιουργεί ένα παραλληλόγραμμο κουμπί όπου μέσα στο κουμπί περιέχεται η εικόνα a.

- public JButton(String a,Icon b)
  Αυτός ο κατασκευαστής δημιουργεί ένα παραλληλόγραμμο κουμπί όπου μέσα στο κουμπί περιέχεται μία συμβολοσειρά a και μία εικόνα b.

Μερικές από τις μεθόδους της τάξης JButton είναι οι επόμενες:

- public void setText(String a)
  Αυτή η συνάρτηση τοποθετεί τη συμβολοσειρά a μέσα στο κουμπί.

- public String getText()
  Ετούτη η συνάρτηση επιστρέφει τη συμβολοσειρά που περιέχεται στο κουμπί.

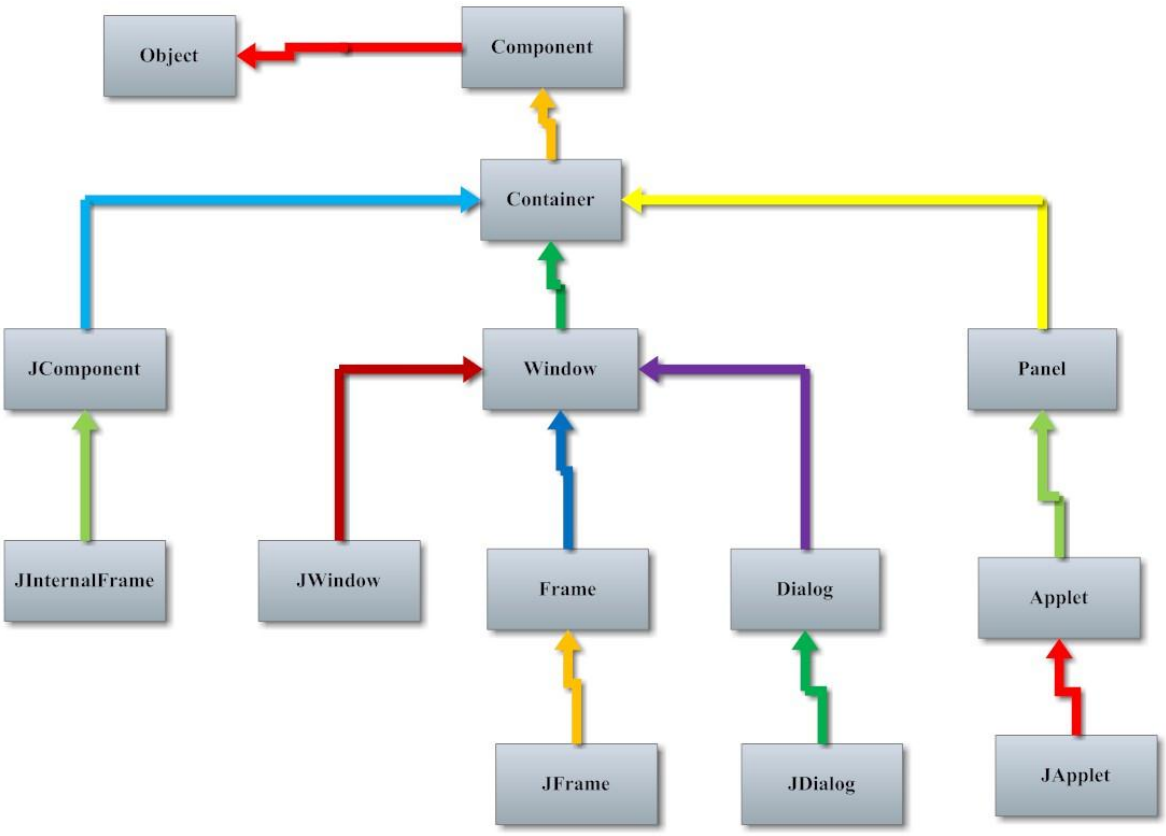
- public void setIcon(Icon a)
  Αυτή η μέθοδος θέτει την εικόνα a εντός του κουμπιού.

- public Icon getIcon()
  Ετούτη η μέθοδος επιστρέφει την εικόνα του κουμπιού.

Για να βάλουμε μία εικόνα σε ένα αντικείμενο της κλάσης JButton χρησιμοποιούμε τον εξής κώδικα:

```
Icon a=new ImageIcon("Διεύθυνση της εικόνας");
JButton b=new JButton(a);
```

Οι κλάσεις Icon και ImageIcon ανήκουν στο πακέτο javax.swing ενώ χρησιμοποιούμε στη συγκεκριμένη περίπτωση έναν από τους κατασκευαστές της τάξης ImageIcon. Δηλαδή υπάρχουν πολλοί άλλοι κατασκευαστές της ενώ η δουλειά που επιτελεί η κλάση ImageIcon με τον κατασκευαστή τον οποίο ορίσαμε, όπως είναι αυτόδηλο είναι το να φορτώνει εικόνες από συγκεκριμένες διευθύνσεις. Επιβάλλεται ακόμη να αναφερθεί ότι το πάτημα του κουμπιού γίνεται όταν ο χρήστης τοποθετήσει το δρομέα του ποντικιού πάνω στην περιοχή του κουμπιού και ύστερα πατήσει το αριστερό πλήκτρο του ποντικιού. Δίχως όμως να χρησιμοποιήσουμε τη μέθοδο addActionListener όσο και να πατάμε το κουμπί η ενέργεια μας δε θα έχει κανένα απολύτως αποτέλεσμα. Για να έχουμε κάποιο αποτέλεσμα με το πάτημα του κουμπιού καταρχάς πρέπει να συγκροτήσουμε μία εσωτερική κλάση στην υποκλάση της κλάσης JApplet ως private (ιδιωτική) δηλαδή που να χρησιμεύει μόνο στην υποκλάση αυτή της κλάσης JApplet όπως παρακάτω:



Σχήμα 2.17 : Στην εικόνα φαίνεται ένα απόσπασμα της ιεραρχίας των βασικών κλάσεων που χρησιμοποιούνται στα Java Swings και τα Java Applets ενώ συμπεριλαμβάνεται στην εικόνα και η τάξη JFrame. Η υπερκλάση είναι το κουτί της εικόνας στο οποίο πάει το βέλος και η υποκλάση είναι το κουτί της εικόνας από όπου προέρχεται το βέλος.

```
private class buttonListener implements ActionListener
{
        public void actionPerformed(ActionEvent event)
        {
                String buttonb=event.getActionCommand();
                sunartisi(buttonb);
        }
 }
```

Εδώ ορίσαμε μία εσωτερική κλάση με το όνομα buttonListener η οποία υλοποιεί τη διεπαφή ActionListener. Κάθε διασύνδεση περιέχει μία συλλογή από επικεφαλίδες μεθόδων και όποια κλάση υλοποιεί τη διασύνδεση πρέπει να υλοποιεί όλες τις μεθόδους της διεπαφής. Η διασύνδεση ActionListener έχει μόνο μία μέθοδο την actionPerformed την οποία και χρησιμοποιούμε. Η κλάση ActionEvent ανήκει στο πακέτο java.awt.event ενώ έχει τη μέθοδο public String getActionCommand. Με ετούτη τη μέθοδο αρχικοποιήσαμε τη συμβολοσειρά buttonb. Κατόπιν βάλαμε μέσα στη συνάρτηση sunartisi τη συμβολοσειρά buttonb καθόσον δεν είναι εφικτό να γράψουμε τεράστιο κώδικα μέσα στη μέθοδο actionPerformed. Τώρα αν έχουμε για παράδειγμα εντός ενός κουμπιού τη συμβολοσειρά “Click” και θέσουμε μία συνθήκη που να λέει πως αν η συμβολοσειρά buttonb είναι ίση με την “Click” να γίνει κόκκινο το παρασκηνιακό χρώμα του υποδοχέα τότε σε περίπτωση που πατήσουμε το εν λόγω κουμπί θα μετατραπεί σε κόκκινο το παρασκηνιακό χρώμα του υποδοχέα. Τουτέστιν θα πραγματοποιηθεί ένα συμβάν. Κοντολογίς η συμβολοσειρά buttonb εξαιτίας του ότι αρχικοποιήθηκε με τη μέθοδο getActionCommand είναι ικανή να παράγει συμβάντα αν θεωρήσουμε πως είναι ίση με τη συμβολοσειρά του κουμπιού που πατιέται. Όσο για τη μέθοδο public void addActionListener (ActionListener a) αποτελεί

μία συνάρτηση της κλάσης AbstractButton η οποία τάξη ανήκει στο πακέτο javax.swing και είναι υπερκλάση της τάξης JButton. Συνεπώς τα αντικείμενα της κλάσης JButton μπορούν να τη χρησιμοποιήσουν διότι οι υποκλάσεις πάντοτε κληρονομούν τις μεθόδους των υπερκλάσεων τους. Οπότε για να γίνει το συμβάν πρέπει να έχουμε γράψει αλλού γενικά και στη μέθοδο init ειδικά τον εξής κώδικα:

```
public class firstApplet extends JApplet
{

private buttonListener blistener;
private JButton button1;
public void init()

{
```

Στη μέθοδο init γράφουμε κώδικα αλλά θα επικεντρωθούμε μόνο στον κώδικα των κατασκευαστών ενός κουμπιού και του ακροατή καθώς και της σύνδεσης τους.

```
blistener=new ButtonListener();
button1=new JButton(“Click”);
button1.addActionListener(blistener);

}
private class buttonListener implements ActionListener

{

public void actionPerformed(ActionEvent event)

{
```

Εδώ βάζουμε ότι παρουσιάσαμε προηγουμένως.

```
}

}

public void sunartisi(String a)

{
```

Εδώ γράφουμε τον κώδικα για να δημιουργήσουμε τα συμβάντα.

```
}

}
```

Η κλάση Component αποτελεί την υπερκλάση όλων των στοιχείων GUI και ορίζει όλες τις κοινές ιδιότητες και συμπεριφορές των υποτάξεων της. Περιέχει μεθόδους οι οποίες προσδιορίζουν το μέγεθος ενός συστατικού, το αν είναι ορατό ή όχι, τη γραμματοσειρά των χαρακτήρων του, το χρώμα του και τρόπους με τους οποίους αυτό δύναται να ανταποκριθεί σε γεγονότα. Η εν λόγω κλάση είναι αφηρημένη δηλαδή δε μπορεί να δημιουργήσει αντικείμενα. Ωστόσο οι υποτάξεις της επιτρέπεται να παράγουν αντικείμενα για να χρησιμοποιήσουν τις μεθόδους της κλάσης Component. Μερικές από τις μεθόδους της τάξης Component είναι οι παρακάτω:

- public void setLocation(int x,int y)
  Η μέθοδος αυτή μετακινεί το συστατικό. Η νέα θέση της πάνω αριστερής γωνίας του δίνονται από τις συντεταγμένες x και y. Τα x και y είναι ακέραιοι αριθμοί που εκφράζουν εικονοστοιχεία.

- public void setBounds(int x,int y,int z,int k)
  Η μέθοδος αυτή έχει περιγραφεί ήδη.

- public void setSize(int x,int y)
  Η συνάρτηση ετούτη μεταβάλλει το μέγεθος του στοιχείου ώστε αυτό να έχει πλάτος x και ύψος y. Τα x και y είναι ακέραιοι αριθμοί που εκφράζουν εικονοστοιχεία.

- public void setFont(Font f)
  Η μέθοδος αυτή προσδιορίζει τη μορφή της γραμματοσειράς του συστατικού. Αν η παράμετρος f είναι null τότε το στοιχείο κληρονομεί τη μορφή της γραμματοσειράς του γονικού του συστατικού.

- public void setForeground(Color c)
  Η συνάρτηση αυτή προσδιορίζει το προσκηνιακό χρώμα του στοιχείου. Εάν η παράμετρος c είναι ίση με null τότε το συστατικό κληρονομεί το προσκηνιακό χρώμα του γονικού του στοιχείου.

- public void setBackground(Color c)
  Η μέθοδος ετούτη προσδιορίζει το παρασκηνιακό χρώμα του στοιχείου. Αν η παράμετρος c είναι ίση με null τότε το συστατικό κληρονομεί το παρασκηνιακό χρώμα του γονικού του συστατικού.

- public void setVisible(boolean b)
  Η συνάρτηση αυτή κάνει ορατό το στοιχείο όταν η τιμή b είναι ίση με true ενώ το κάνει αόρατο αν η τιμή b είναι false.

- public Color getBackground()
  Η μέθοδος αυτή επιστρέφει μία αναφορά για το παρασκηνιακό χρώμα του συστατικού.

- public Color getForeground()
  Η συνάρτηση αυτή επιστρέφει μία αναφορά σχετικά με το προσκηνιακό χρώμα του στοιχείου.

- public void setEnabled(boolean b)
  Η μέθοδος ετούτη ενεργοποιεί το συστατικό αν η τιμή του b είναι true. Διαφορετικά εάν η τιμή του b είναι false το στοιχείο απενεργοποιείται.

- public void setFocusable(boolean b)
  Η συνάρτηση ετούτη δίνει την εστίαση σε ένα στοιχείο εάν η τιμή του b είναι true. Αν η τιμή του b είναι false χάνεται η εστίαση από το συστατικό.

- public void addKeyListener(KeyListener a)
  Η μέθοδος αυτή προσθέτει τον ακροατή a ώστε να λαμβάνει γεγονότα του πληκτρολογίου από το στοιχείο αυτό.

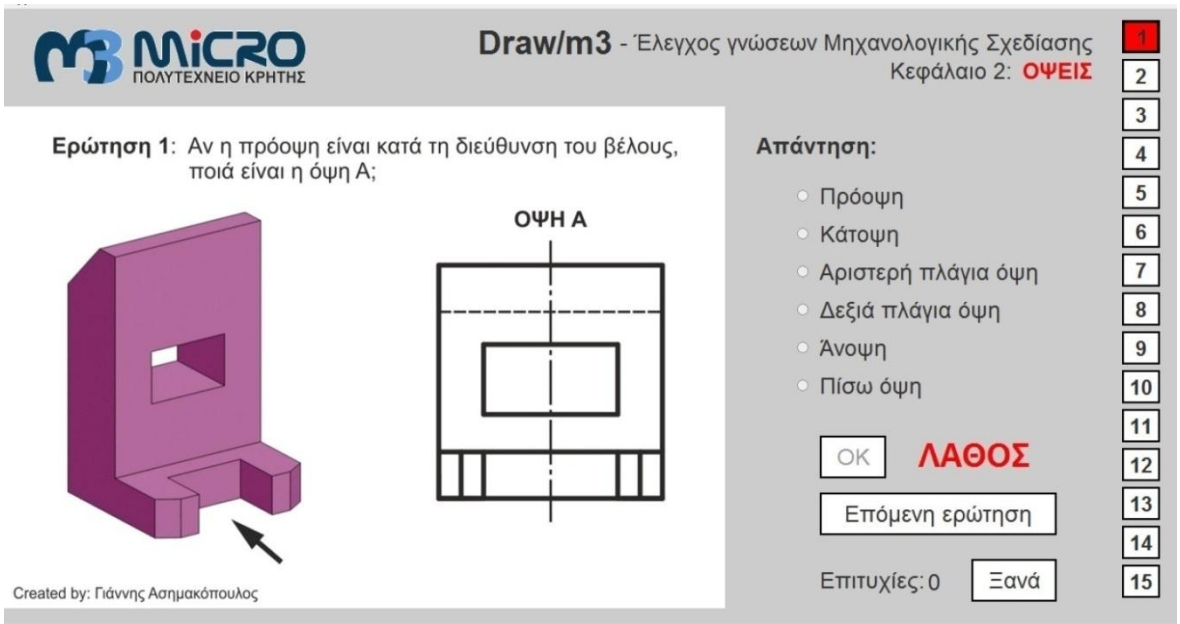
- public void addMouseListener(MouseListener a)
  Η συνάρτηση αυτή προσθέτει τον ακροατή a ώστε να λαμβάνει γεγονότα του ποντικιού από το συστατικό.

- public void addMouseMotionListener(MouseMotionListener a)
  Η μέθοδος ετούτη προσθέτει τον ακροατή a για να λαμβάνει γεγονότα κίνησης του ποντικιού από το στοιχείο αυτό.

- public void paint(Graphics g)
  Η συνάρτηση αυτή ζωγραφίζει το συστατικό. Η μέθοδος ετούτη καλείται αυτομάτως από το σύστημα όταν το στοιχείο πρέπει να ζωγραφιστεί. Για παράδειγμα όταν αυτό εμφανίζεται για πρώτη φορά ή έχει καταστραφεί και χρειάζεται να διορθωθεί. Η παράμετρος g είναι μία αναφορά της αφηρημένης τάξης Graphics η οποία ανήκει στο πακέτο java.awt και προσδιορίζει την περιοχή όπου πρέπει να ζωγραφιστεί το συστατικό. Εφόσον η συνάρτηση paint καλείται αυτόματα από το σύστημα πολύ σπάνια απαιτείται να κληθεί από τον προγραμματιστή. Συνήθως ακυρώνεται από αυτόν. Στην περίπτωση που ο προγραμματιστής επιθυμεί να ξαναζωγραφίσει το στοιχείο τότε πρέπει να καλέσει τη μέθοδο repaint.

- public void repaint(Graphics g)
  Η μέθοδος αυτή ξαναζωγραφίζει το συστατικό. Όταν το στοιχείο είναι ελαφρύ τουτέστιν όχι πολύπλοκα σχεδιασμένο τότε η κλήση ετούτης της συνάρτησης προξενεί την κλήση της μεθόδου paint το συντομότερο δυνατό. Διαφορετικά σε περίπτωση όπου το συστατικό είναι βαρύ δηλαδή περίπλοκα σχεδιασμένο τότε η κλήση ετούτης της μεθόδου προκαλεί την κλήση της συνάρτησης update το συντομότερο δυνατόν.

- public void update(Graphics g)
  Η συνάρτηση ετούτη ενημερώνει το στοιχείο. Αν το συστατικό είναι βαρύ τότε η κλήση της μεθόδου update είναι το αποτέλεσμα μίας κλήσης της μεθόδου repaint. Η συνάρτηση update καλεί τη μέθοδο paint του στοιχείου για να το ξανασχεδιάσει. Όπως και η μέθοδος paint τοιουτοτρόπως και η συνάρτηση update πολύ σπάνια χρειάζεται να κληθεί άμεσα από τον προγραμματιστή αν και η τελευταία μπορεί να ακυρωθεί για να ελαττώσει το τρεμοπαίξιμο των κινούμενων γραφικών.



Σχήμα 2.18 : Στην εικόνα φαίνονται τα αποτελέσματα της εφαρμογής των συναρτήσεων setEnabled(true) και setEnabled(false). Όλα τα κυκλικά κουμπιά καθώς και ένα από τα παραλληλόγραμμα κουμπιά είναι απενεργοποιημένα ενώ τα άλλα δύο παραλληλόγραμμα κουμπιά είναι ενεργοποιημένα.

Η κλάση Container (υποδοχέας) είναι υποτάξη της κλάσης Component (συστατικό). Αφού η τάξη Container κληρονομεί την Component κάθε αντικείμενο της Container είναι υποδοχέας και συστατικό. Πανομοιότυπα τα αντικείμενα των υποκλάσεων της Container αποτελούν υποδοχείς και συστατικά. Συνεπώς τα αντικείμενα της Container και όλα τα αντικείμενα των υποτάξεων της κλάσης Container έχουν τη δυνατότητα να ενσωματωθούν σε έναν άλλον υποδοχέα γεγονός το οποίο μας δίνει την ικανότητα να δημιουργήσουμε σύνθετες γραφικές διεπαφές χρήστη. Η τάξη Container εκτός του ότι ορίζει όλες τις κοινές ιδιότητες και συμπεριφορές των αντικειμένων των υποκλάσεων της, κληρονομεί και τις μεθόδους της τάξης Component. Ορισμένες από τις μεθόδους της κλάσης Container είναι οι εξής:

- public Component add(Component a)
  Η συνάρτηση αυτή προσθέτει το συστατικό a σε έναν υποδοχέα.

- public void setLayout(LayoutManager a)
  Η μέθοδος ετούτη έχει ήδη περιγραφεί.

- public void paint(Graphics g)
  Η συνάρτηση ετούτη ζωγραφίζει αυτόν τον υποδοχέα. Για παράδειγμα αν ο χρήστης καλύψει το παράθυρο ενός υποδοχέα στην οθόνη με ένα άλλο ανοικτό παράθυρο και αργότερα ξεσκεπάσει το παράθυρο του υποδοχέα τότε αυτομάτως καλείται η μέθοδος paint. Η συνάρτηση αυτή λειτουργεί αντιστοίχως όπως η μέθοδος paint της τάξης Component.

- public void update(Graphics g)
  Η συνάρτηση αυτή ενημερώνει τον υποδοχέα και λειτουργεί αντιστοίχως όπως λειτουργεί η μέθοδος update της κλάσης Component.

Η τάξη JComponent είναι η υπερκλάση των περισσότερων συστατικών του πακέτου swing, τα ονόματα των οποίων αρχίζουν με 'J'. Αποτελεί μία αφηρημένη κλάση και επομένως δε μπορεί να δημιουργήσει αντικείμενα. Μολαταύτα οι υποτάξεις της κλάσης JComponent δύνανται να παράγουν αντικείμενα για να χρησιμοποιήσουν τις μεθόδους της τάξης JComponent. Οποιαδήποτε αφηρημένη - abstract κλάση σχηματίζεται με αποκλειστικό σκοπό το να παραχθούν υποτάξεις της οι οποίες θα μπορούν να κάνουν χρήση των συναρτήσεων της. Η κλάση JComponent ανήκει στο πακέτο javax.swing και δύο από τις μεθόδους της είναι οι παρακάτω:

- public boolean setOpaque(boolean b)
  Ετούτη η συνάρτηση καθιστά το συστατικό αδιαφανές αν η παράμετρος b είναι true αλλιώς αν είναι false το στοιχείο μετατρέπεται σε διαφανές. Για παράδειγμα αν έχουμε σχηματίζει μία ορθογώνια περιοχή με την κλάση JLabel και θέλουμε να δώσουμε κάποιο χρώμα μέσα σε αυτήν την περιοχή αυτό είναι δυνατό να γίνει μόνο στην περίπτωση που θα έχουμε θέσει την παράμετρο b ως true.

- public void setToolTipText(String a)
  Ετούτη η μέθοδος προσδιορίζει τη συμβολοσειρά a η οποία θα εμφανιστεί όταν ο δείκτης του ποντικιού παραμείνει για λίγο χρονικό διάστημα πάνω από το συγκεκριμένο συστατικό.

- public void setBorder (Border border)
  Αυτή η μέθοδος θέτει το όριο border του στοιχείου. Η τάξη Border ανήκει στο πακέτο javax.swing.border.

Η τάξη JLabel έχει αντικείμενα τα οποία δημιουργούν μία περιοχή όπου θα δύναται να παρουσιαστεί ένα κείμενο, μία εικόνα ή και τα δύο. Αρκετές φορές αυτή η ετικέτα

τοποθετείται δίπλα σε κάποιο άλλο στοιχείο γραφικών διεπαφών χρήστη περιέχοντας μία συμβολοσειρά ή και μία εικόνα που χρησιμοποιείται για να υποδεικνύει την αποστολή ετούτου του άλλου συστατικού γραφικών διεπαφών χρήστη. Κάθε ετικέτα δηλαδή η κλάση JLabel έχει μία περιοχή εμφάνισης. Η ετικέτα αποτελεί υπερβολικά σημαντική τάξη καθόσον παρέχει στον προγραμματιστή αξιόλογη ευελιξία στο σχεδιασμό προγραμμάτων με γραφικά γεγονός στο οποίο συντελούν εκτός από τους εξαιρετικούς κατασκευαστές της και οι σπουδαίες συναρτήσεις που διαθέτει. Η κλάση JLabel ανήκει στο πακέτο javax.swing ενώ έχει τους επόμενους κατασκευαστές:

- public JLabel()
  Αυτός ο κατασκευαστής παράγει μία ετικέτα η οποία δεν περιέχει κείμενο ή εικόνα και είναι κατακόρυφα κεντραρισμένη μέσα στην περιοχή εμφάνισης της. Έπειτα από τη δημιουργία της στην ετικέτα δύναται να προστεθεί κείμενο ή εικόνα από μεθόδους της τάξης JLabel.

- public JLabel(String s)
  Ετούτος ο κατασκευαστής δημιουργεί μία ετικέτα που περιέχει τη συμβολοσειρά s. Η ετικέτα είναι στοιχισμένη ως προς την αρχική άκρη της περιοχής εμφάνισης της και κεντραρισμένη κατακόρυφα μέσα σε αυτήν.

- public JLabel(String s,int a)
  Αυτός ο κατασκευαστής σχηματίζει μία ετικέτα η οποία περιέχει τη συμβολοσειρά s. Η ετικέτα είναι κεντραρισμένη κατακόρυφα στην περιοχή εμφάνισης της. Η παράμετρος a είναι ένας ακέραιος αριθμός ο οποίος στοιχίζει οριζόντια την ετικέτα αναλόγως με ποια από τις παρακάτω σταθερές της κλάσης JLabel λάβει:

Σταθερά	**Στοίχιση**
SwingConstants.LEFT	Προς τα αριστερά
SwingConstants.CENTER	Στο κέντρο
SwingConstants.RIGHT	Προς τα δεξιά
SwingConstants.LEADING	Στην αρχή της περιοχής
SwingConstants.TRAILING	Στο τέλος της περιοχής

- public JLabel(Icon a)
  Ετούτος ο κατασκευαστής παράγει μία ετικέτα που περιέχει την εικόνα a. Η ετικέτα είναι κατακόρυφα και οριζόντια κεντραρισμένη στην περιοχή εμφάνισης της.

- public JLabel(Icon a,int b)
  Αυτός ο κατασκευαστής δημιουργεί μία ετικέτα η οποία περικλείει μία εικόνα a. Η παράμετρος b καθορίζει την οριζόντια στοίχιση της ετικέτας μέσα στην περιοχή παρουσίασης της και παίρνει μία από τις τιμές που προσδιορίστηκαν παραπάνω. Η ετικέτα είναι κατακόρυφα κεντραρισμένη μέσα στον τομέα εμφάνισης της.

- public JLabel(String s,Icon a,int b)
  Ετούτος ο κατασκευαστής σχηματίζει μία ετικέτα που εμπεριέχει μία συμβολοσειρά s και μία εικόνα a ενώ η ακέραιη τιμή b παίρνει μία από τις τιμές οι οποίες αναφέρθηκαν παραπάνω. Η ετικέτα είναι κατακόρυφα κεντραρισμένη εντός του πεδίου εμφάνισης της ενώ η συμβολοσειρά s βρίσκεται στην τελική - trailing άκρη της εικόνας a.

Μερικές από τις μεθόδους της τάξης JLabel είναι οι εξής:

- public void setHorizontalAlignment(int a)
  Αυτή η συνάρτηση θέτει τη στοίχιση του τίτλου της ετικέτας κατά μήκος του άξονα x (οριζοντίως) όπου η ακέραιη τιμή a λαμβάνει μία από τις τιμές οι οποίες προσδιορίστηκαν προηγουμένως. Ως τίτλος της ετικέτας εννοείται η συμβολοσειρά ή και η εικόνα που βρίσκονται μέσα στην ετικέτα.

- public int getHorizontalAlignment()
  Ετούτη η μέθοδος επιστρέφει τη στοίχιση του τίτλου της ετικέτας κατά μήκος του άξονα x ως μία τιμή από αυτές που δηλώθηκαν πρωτύτερα.

- public void setVerticalAlignment(int a)
  Ετούτη η συνάρτηση θέτει τη στοίχιση του τίτλου της ετικέτας κατά μήκος του άξονα y (κάθετα). Η παράμετρος a δύναται να λάβει μία από τις ακόλουθες τιμές:

Σταθερά	**Στοίχιση**
SwingConstants.TOP	Στοίχιση στην κορυφή
SwingConstants.CENTER	Στοίχιση στο κέντρο
SwingConstants.BOTTOM	Στοίχιση στη βάση

- public int getVerticalAlignment()
  Αυτή η μέθοδος επιστρέφει τη στοίχιση του τίτλου της ετικέτας κατά μήκος του άξονα y ως μία από τις παραπάνω τιμές.

- public void setHorizontalTextPosition(int a)
  Ετούτη η συνάρτηση με βάση την τιμή της παραμέτρου a η οποία μπορεί να είναι μία από τις τιμές που αναφέρθηκαν πιο πάνω καθορίζει την οριζόντια θέση της συμβολοσειράς της ετικέτας σχετικά με τη θέση της εικόνας της.

- public int getHorizontalTextPosition()
  Αυτή η συνάρτηση επιστρέφει τη σχετική με τη θέση της εικόνας οριζόντια θέση της συμβολοσειράς της ετικέτας.

- public void setVerticalTextPosition(int a)
  Ετούτη η μέθοδος καθορίζει την κατακόρυφη στοίχιση της συμβολοσειράς της ετικέτας σε σχέση με τη θέση της εικόνας της. Η παράμετρος a δύναται να λάβει μία από τις τιμές οι οποίες προσδιορίστηκαν παραπάνω.

- public int getVerticalTextPosition()
  Αυτή η συνάρτηση επιστρέφει τη σχετική με τη θέση της εικόνας θέση της συμβολοσειράς της ετικέτας ως προς τον κατακόρυφο άξονα.

- public void setText(String a)
  Ετούτη η μέθοδος τοποθετεί τη συμβολοσειρά a μέσα στην ετικέτα.

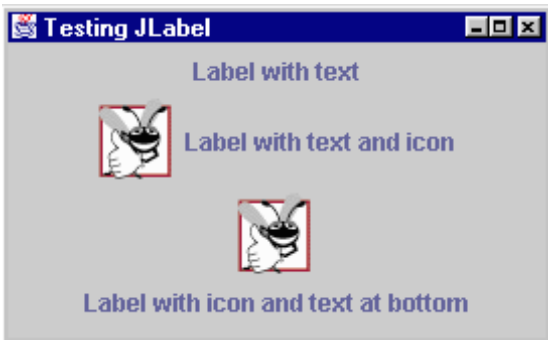
- public String getText()
  Ετούτη η συνάρτηση επιστρέφει τη συμβολοσειρά η οποία βρίσκεται μέσα στην ετικέτα.

- public void setIcon(Icon a)
  Αυτή η μέθοδος τοποθετεί την εικόνα a ένδον της ετικέτας.

- public Icon getIcon()
  Αυτή η συνάρτηση επιστρέφει την εικόνα που περιέχει η ετικέτα.

- public void setLabelfor(Component a)
  Ετούτη η μέθοδος θέτει την ετικέτα στο συστατικό a. Η παράμετρος a μπορεί να είναι null εάν η συγκεκριμένη ετικέτα δεν αντιστοιχεί σε κάποιο συστατικό.

- public Component getLabelfor()
  Αυτή η συνάρτηση επιστρέφει το συστατικό στο οποίο αντιστοιχεί η εν λόγω ετικέτα. Σε περίπτωση που η ετικέτα δεν αντιστοιχεί σε κάποιο συστατικό η τιμή που επιστρέφει η μέθοδος είναι null.



Σχήμα 2.19 : Στην εικόνα φαίνεται ένα παράθυρο το οποίο έχει δημιουργηθεί από την κλάση JFrame και όχι από την JApplet. Στο συγκεκριμένο παράθυρο διακρίνεται η εφαρμογή της τάξης JLabel. Υπάρχουν τρία στοιχεία της κλάσης JLabel. Το πρώτο βρίσκεται στο πάνω μέρος του παραθύρου όπου φαίνεται μία ετικέτα η οποία περιέχει μία συμβολοσειρά. Το δεύτερο στοιχείο διακρίνεται στο κέντρο όπου η ετικέτα συγκαταλέγει μία εικόνα και μία συμβολοσειρά. Το τρίτο είναι τοποθετημένο στο κάτω μέρος του παραθύρου και η ετικέτα περικλείει ομοίως μία εικόνα και μία συμβολοσειρά αλλά όχι στη σειρά αφού η εικόνα είναι πάνω από τη συμβολοσειρά.
Η κλάση JTextField δημιουργεί ένα συστατικό γραφικών διεπαφών χρήστη το οποίο χρησιμοποιείται είτε για την εισαγωγή μίας γραμμής κειμένου που πληκτρολογεί ο χρήστης είτε για να εμφανίσει μία γραμμή κειμένου ως έξοδο του προγράμματος. Η γραμμή κειμένου πληκτρολογείται ή εμφανίζεται μέσα σε ένα ορθογώνιο. Όταν ο χρήστης πατήσει Enter για να στείλει τη συμβολοσειρά που πληκτρολόγησε στο πρόγραμμα, συμβαίνει ένα γεγονός ενέργειας - action event. Αν το πρόγραμμα διαθέτει έναν ακροατή ενέργειας συνδεδεμένο με το πεδίο κειμένου τότε αυτός μπορεί να αντιμετωπίσει το εν λόγω γεγονός. Η τάξη JTextField ανήκει στο πακέτο javax.swing ενώ μερικοί από τους κατασκευαστές της είναι οι παρακάτω:

- public JTextField()
  Αυτός ο κατασκευαστής παράγει ένα νέο πεδίο κειμένου που είναι κενό δηλαδή έχει μηδέν στήλες και περιέχει την κενή συμβολοσειρά.

- public JTextField(String a)
  Ετούτος ο κατασκευαστής σχηματίζει ένα νέο πεδίο κειμένου το οποίο τοποθετεί τη συμβολοσειρά a εντός αυτού. Ο αριθμός των στηλών του πεδίου τίθεται στο μηδέν.

- public JTextField(int a)
  Αυτός ο κατασκευαστής δημιουργεί ένα νέο πεδίο κειμένου με αριθμό στηλών a όπου a ακέραιος αριθμός.

- public JTextField(String a,int b)
  Ετούτος ο κατασκευαστής συγκροτεί ένα νέο πεδίο κειμένου που περιλαμβάνει τη συμβολοσειρά a και έχει αριθμό στηλών b όπου b ακέραιος αριθμός.

Μερικές από τις μεθόδους της κλάσης JTextField είναι οι εξής:

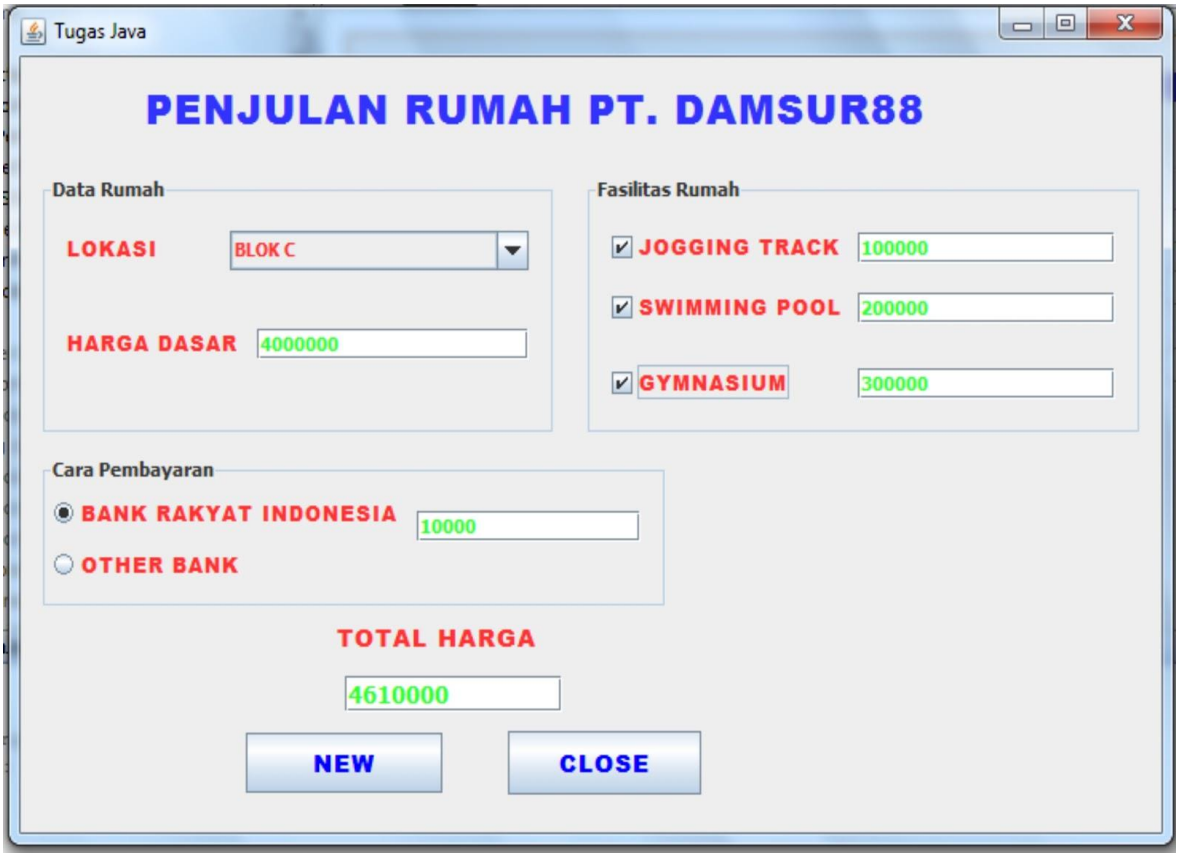
- public String getText()
  Αυτή η συνάρτηση επιστρέφει το κείμενο που υπάρχει μέσα στο πεδίο κειμένου ως μία συμβολοσειρά.

- public void setText(String a)
  Ετούτη η συνάρτηση θέτει τη συμβολοσειρά a στο πεδίο κειμένου. Εάν η συμβολοσειρά a είναι κενή τότε διαγράφεται το κείμενο το οποίο βρισκόταν μέσα στο πεδίο κειμένου.

- public void setEditable(boolean b)
  Σε αυτήν τη μέθοδο αν η παράμετρος b είναι ίση με false τότε ο χρήστης δε δύναται να τροποποιήσει τα περιεχόμενα του αντικειμένου της τάξης JTextField. Αντιθέτως εάν η παράμετρος b είναι true τότε ο χρήστης μπορεί να πληκτρολογήσει μέσα στο πεδίο κειμένου.

- public boolean isEditable()
  Ετούτη μέθοδος επιστρέφει true ή false ανάλογα με το αν τα περιεχόμενα του πεδίου κειμένου είναι τροποποιήσιμα.

- public void addActionListener(ActionListener a)
  Αυτήν τη συνάρτηση την έχουμε δει προηγουμένως και προσθέτει τον ακροατή ενέργειας a στο πεδίο κειμένου.



Σχήμα 2.20 : Στην εικόνα φαίνεται ένα παράθυρο το οποίο έχει δημιουργηθεί από την κλάση JFrame και όχι από την JApplet. Στην εικόνα οι ορθογώνιες περιοχές στις οποίες έχουν τεθεί οι αριθμοί με πράσινο χρώμα αποτελούν παραδείγματα πεδίου κειμένου τουτέστιν παραδείγματα της τάξης JTextField.

Η κλάση JTextArea δηλαδή μία περιοχή κειμένου είναι παρόμοια με την τάξη JTextField (πεδίο κειμένου) με τη διαφορά πως η πρώτη επιτρέπει την εμφάνιση ή την εισαγωγή πολλών γραμμών κειμένου σε αυτήν. Στην τάξη JTextArea το πάτημα του πλήκτρου Enter επιφέρει τη δημιουργία μίας νέας γραμμής ενώ όπως προαναφέρθηκε στην κλάση JTextField το πάτημα του πλήκτρου Enter προκαλεί ένα συμβάν ActionEvent το οποίο συμβάν είναι πανομοιότυπο με τα συμβάντα από κουμπιά διαταγών με τη διαφορά ότι η εστίαση δεν εγκαταλείπει το πεδίο. Και εάν είναι επιθυμητή η έξοδος πρέπει να προγραμματιστεί. Η τάξη JTextArea ανήκει στο πακέτο javax.swing ενώ ορισμένοι από τους κατασκευαστές της είναι οι ακόλουθοι:

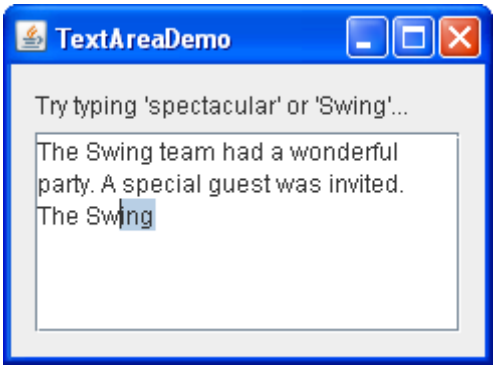
- public JTextArea()
  Αυτός ο κατασκευαστής σχηματίζει μία νέα περιοχή που είναι κενή δηλαδή έχει μηδέν γραμμές, μηδέν στήλες και κενή συμβολοσειρά.

- public JTextArea(String a)
  Ετούτος ο κατασκευαστής παράγει μία νέα περιοχή κειμένου η οποία εμπεριέχει τη συμβολοσειρά a. Το πλήθος των γραμμών και των στηλών της περιοχής τίθεται στο μηδέν.

- public JTextArea(int a,int b)
  Αυτός ο κατασκευαστής δημιουργεί μία νέα κενή περιοχή κειμένου, οι διαστάσεις της οποίας καθορίζονται από τους ακέραιους αριθμούς a, b. Ο ακέραιος αριθμός a προσδιορίζει το νούμερο των γραμμών της περιοχής κειμένου και ο ακέραιος αριθμός b το πλήθος των στηλών της.

Η κλάση JTextArea διαθέτει τις μεθόδους getText, setText, setEditable και isEditable οι οποίες είναι ίδιες με τις συναρτήσεις της τάξης JTextField. Αυτό συμβαίνει καθόσον και οι δύο κλάσεις κληρονομούν την αφηρημένη τάξη JTextComponent στην οποία ανήκουν ετούτες οι συναρτήσεις. Και αφού οι κλάσεις JTextField και JTextArea είναι υποτάξεις της κλάσης JTextComponent κληρονομούν τις μεθόδους της. Η τάξη JTextArea έχει μία ακόμη σημαντική συνάρτηση:

- public void append(String a)
  Αυτή η μέθοδος επισυνάπτει τη συμβολοσειρά a στο τέλος ενός αντικειμένου της κλάσης JTextArea.



Σχήμα 2.21 : Στην εικόνα φαίνεται ένα παράθυρο το οποίο έχει δημιουργηθεί από την κλάση JFrame και όχι από την JApplet. Εδώ απεικονίζεται η εφαρμογή της τάξης JTextArea με το στοιχείο το οποίο ενέχει κείμενο ενώ επίσης περιλαμβάνει και το άσπρο χρώμα φόντου που πλαισιώνει το κείμενο.

Μία άλλη πολύ σημαντική κλάση είναι η JPasswordField. Η τάξη αυτή δημιουργεί ένα στοιχείο γραφικών διεπαφών χρήστη το οποίο χρησιμοποιείται για την εισαγωγή μίας γραμμής κειμένου που πληκτρολογεί ο χρήστης. Η γραμμή κειμένου πληκτρολογείται μέσα σε ένα ορθογώνιο. Όταν ο χρήστης πατήσει Enter για να στείλει τη συμβολοσειρά που πληκτρολόγησε στο πρόγραμμα, συμβαίνει ένα γεγονός ενέργειας - action event. Αν το πρόγραμμα διαθέτει έναν ακροατή ενέργειας συνδεδεμένο με το αντικείμενο της τάξης JPasswordField τότε αυτός μπορεί να αντιμετωπίσει το εν λόγω γεγονός. Δηλαδή η κλάση ετούτη μοιάζει πάρα πολύ με την τάξη JTextField με τη διαφορά πως το κείμενο το οποίο καταχωρίζει ο χρήστης κρύβεται με αστερίσκους. Η κλάση JPasswordField ανήκει στο πακέτο javax.swing ενώ ορισμένοι από τους κατασκευαστές της είναι οι παρακάτω:

- public JPasswordField()
  Αυτός ο κατασκευαστής σχηματίζει ένα αντικείμενο της τάξης JPasswordField που είναι κενό δηλαδή έχει μηδέν στήλες και περιέχει την κενή συμβολοσειρά.

- public JPasswordField(String a)
  Ετούτος ο κατασκευαστής παράγει ένα αντικείμενο της κλάσης JPasswordField στο οποίο τοποθετείται η συμβολοσειρά a εντός αυτού. Ο αριθμός των στηλών τίθεται στο μηδέν.

- public JPasswordField(int a)
  Αυτός ο κατασκευαστής δημιουργεί ένα αντικείμενο της κλάσης JPasswordField με αριθμό στηλών a όπου a ακέραιος αριθμός.

- public JPasswordField(String a,int b)
  Ετούτος ο κατασκευαστής συγκροτεί ένα αντικείμενο της τάξης JPasswordField

που περιλαμβάνει τη συμβολοσειρά a και έχει αριθμό στηλών b όπου b ακέραιος αριθμός.

Μία αξιοσημείωτη μέθοδος της κλάσης JPasswordField είναι η εξής:

- public char [] getPassword()
  Η συνάρτηση αυτή επιστρέφει το κείμενο στο πεδίο ως πίνακα από χαρακτήρες. Για λόγους ασφαλείας δεν πρέπει να χρησιμοποιηθεί το αντικείμενο String. Για αυτό και η μέθοδος getText που κληρονομείται από την τάξη JTextComponent θεωρείται παρωχημένη.

Για να δημιουργήσουμε πλαίσια ελέγχου χρησιμοποιούμε την κλάση JCheckBox. Τα αντικείμενα αυτής της τάξης παρουσιάζουν στην οθόνη μικρές τετράγωνες περιοχές. Όταν πατήσουμε κάποια από ετούτες τις τετράγωνες περιοχές εμφανίζεται ένα γραφικό ‘‘v’’, ενδεικτικό ότι την έχουμε επιλέξει. Δηλαδή χρησιμοποιούμε αντικείμενα της κλάσης JCheckBox όταν επιθυμούμε να συντάξουμε πολλαπλές επιλογές, χωρίς όμως να είναι αλληλοαποκλειόμενες. Η κλάση JCheckBox ανήκει στο πακέτο javax.swing ενώ έχει τους εξής βασικούς κατασκευαστές:

- public JCheckBox()
  Αυτός ο κατασκευαστής δημιουργεί ένα πλαίσιο ελέγχου το οποίο αρχικά δεν είναι επιλεγμένο ενώ δεν περιέχει ούτε συμβολοσειρά, ούτε εικόνα.

- public JCheckBox(Icon a)
  Ετούτος ο κατασκευαστής παράγει ένα πλαίσιο ελέγχου που αρχικά δεν είναι επιλεγμένο και περιέχει την εικόνα a.

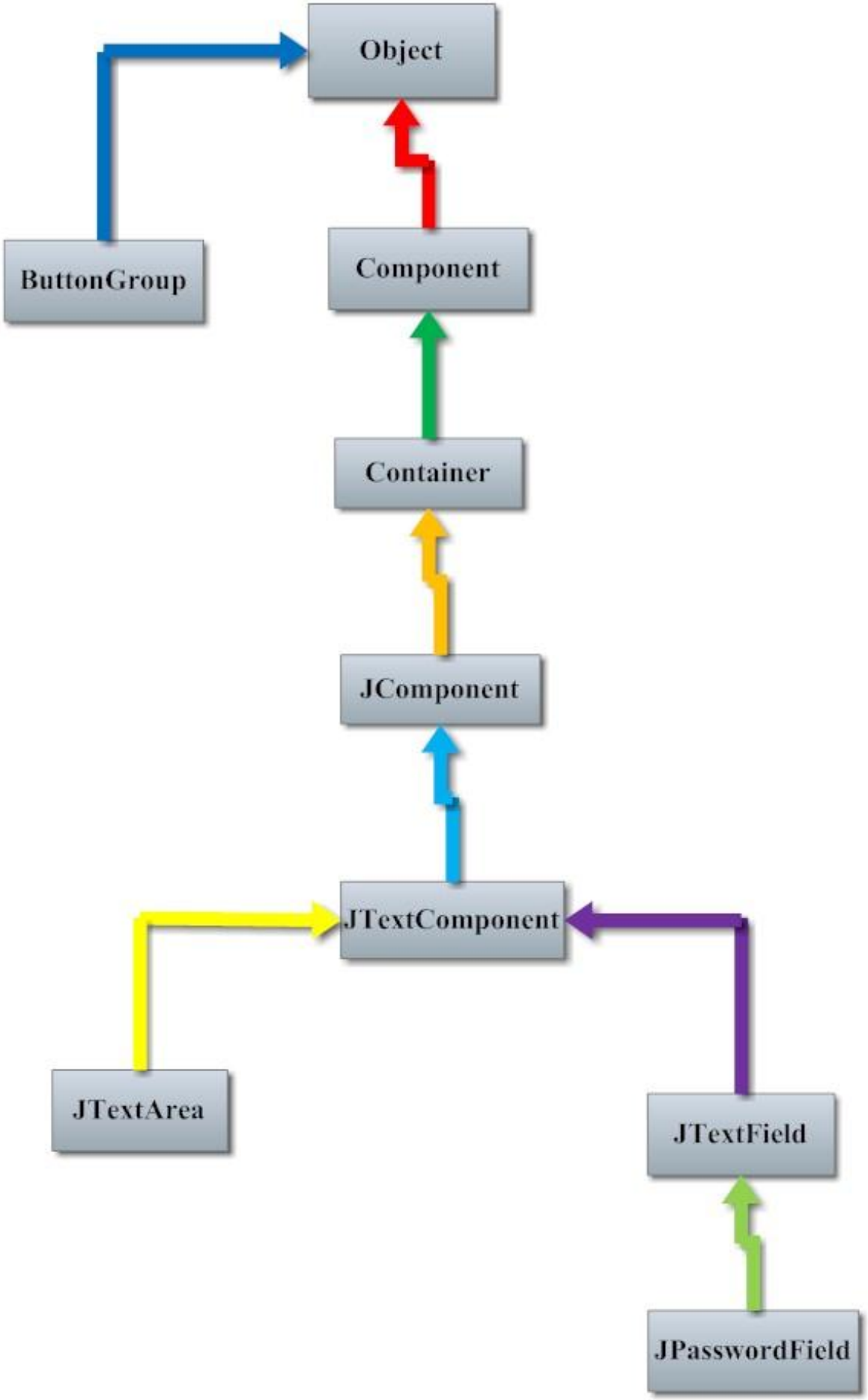
- public JCheckBox(Icon a,boolean b)
  Αυτός ο κατασκευαστής σχηματίζει ένα πλαίσιο ελέγχου το οποίο εμπεριέχει την εικόνα a. Επίσης αν η παράμετρος b είναι true τότε το πλαίσιο ελέγχου είναι αρχικά επιλεγμένο ενώ εάν είναι false δεν είναι επιλεγμένο.

- public JCheckBox(String s)
  Ετούτος ο κατασκευαστής δημουργεί ένα πλαίσιο ελέγχου που αρχικά δεν είναι επιλεγμένο ενώ περιέχει τη συμβολοσειρά s.

- public JCheckBox(String s,boolean b)
  Αυτός ο κατασκευαστής παράγει ένα πλαίσιο ελέγχου το οποίο περιέχει τη συμβολοσειρά s και αν η παράμετρος b είναι true τότε το πλαίσιο ελέγχου είναι αρχικά επιλεγμένο ενώ όταν είναι false δεν είναι επιλεγμένο.

- public JCheckBox(String s,Icon a)
  Ετούτος ο κατασκευαστής σχηματίζει ένα πλαίσιο ελέγχου που εμπεριέχει τη συμβολοσειρά s και την εικόνα a ενώ δεν είναι αρχικά επιλεγμένο.

- public JCheckBox(String s,Icon a,boolean b)
  Αυτός ο κατασκευαστής δημιουργεί ένα πλαίσιο ελέγχου το οποίο περιέχει τη συμβολοσειρά s και την εικόνα a ενώ είναι αρχικά επιλεγμένο αν η παράμετρος b είναι true. Εάν είναι false το πλαίσιο ελέγχου δεν είναι αρχικά επιλεγμένο.



Σχήμα 2.22 : Στην εικόνα φαίνεται ένα απόσπασμα της ιεραρχίας των βασικών κλάσεων

που χρησιμοποιούνται στα Java Swings και τα Java Applets ενώ συμπεριλαμβάνονται στην εικόνα και οι τάξεις JPasswordField και ButtonGroup. Η υπερκλάση είναι το κουτί της εικόνας στο οποίο πάει το βέλος και η υποκλάση είναι το κουτί της εικόνας από όπου προέρχεται το βέλος.

Δύο βαρυσήμαντες μέθοδοι της τάξης JCheckBox τις οποίες κληρονομεί από την αφηρημένη κλάση AbstractButton είναι οι επόμενες:

- public boolean isSelected()
  Η συνάρτηση ετούτη χρησιμοποιείται για να διαπιστωθεί εάν έχει επιλεγεί το πλαίσιο ελέγχου. Αν η επιστροφή της συνάρτησης είναι true τότε έχει διαλεχθεί το πλαίσιο ελέγχου ενώ αν είναι false δεν έχει επιλεγεί το συγκεκριμένο πλαίσιο ελέγχου.

- public void setSelected(boolean b)
  Η μέθοδος αυτή χρησιμοποιείται για να οριστεί το πλαίσιο ελέγχου ως επιλεγμένο ή ως μη επιλεγμένο. Εάν η παράμετρος b τεθεί ως true τότε το πλαίσιο ελέγχου ορίζεται ως επιλεγμένο ενώ όταν τεθεί ως false το πλαίσιο ελέγχου ορίζεται ως μη επιλεγμένο.

Για να κατασκευάσουμε ραδιοπλήκτρα χρησιμοποιούμε την κλάση JRadioButton. Ετούτη η κλάση φανερώνει στην οθόνη μία μικρή κυκλική περιοχή που όταν την πατήσουμε εμφανίζεται μία μικρή τελεία, ενδεικτικό ότι έχει επιλεχθεί. Χρησιμοποιούμε αυτό το είδος της τάξης κυρίως όταν επιδιώκουμε να συντάξουμε πολλαπλές επιλογές, οι οποίες είναι αλληλοαποκλειόμενες, δηλαδή η επιλογή μίας εξ' αυτών αυτομάτως αναιρεί την επιλογή της προηγούμενης. Ωστόσο για να γίνουν οι επιλογές με τα ραδιοπλήκτρα αλληλοαποκλειόμενες επιβάλλεται να κάνουμε χρήση της κλάσης ButtonGroup η οποία βάζει τα ραδιοπλήκτρα σε μία ομάδα. Οι τάξεις JRadioButton και ButtonGroup ανήκουν στο πακέτο javax.swing ενώ έχουν τους ακόλουθους βασικούς κατασκευαστές:

- public JRadioButton()
  Ετούτος ο κατασκευαστής δημιουργεί ένα ραδιοπλήκτρο το οποίο δεν είναι αρχικά επιλεγμένο και δεν περιέχει συμβολοσειρά ή εικόνα.

- public JRadioButton(Icon a,boolean b)
  Αυτός ο κατασκευαστής παράγει ένα ραδιοπλήκτρο που εμπεριέχει μία εικόνα a και είναι αρχικά επιλεγμένο αν η παράμετρος b έχει την τιμή true. Εάν η παράμετρος b έχει την τιμή false τότε το ραδιοπλήκτρο δεν είναι αρχικά επιλεγμένο.

- public JRadioButton(String s,boolean b)
  Ετούτος ο κατασκευαστής σχηματίζει ένα ραδιοπλήκτρο το οποίο περιέχει μία συμβολοσειρά s και είναι αρχικά επιλεγμένο αν η παράμετρος b έχει την τιμή true. Εάν η παράμετρος b έχει την τιμή false τότε το ραδιοπλήκτρο δεν είναι αρχικά επιλεγμένο.

- public JRadioButton(String s,Icon a,boolean b)
  Αυτός ο κατασκευαστής δημιουργεί ένα ραδιοπλήκτρο που περιέχει τη συμβολοσειρά s και την εικόνα a. Αν η παράμετρος b έχει την τιμή true το ραδιοπλήκτρο είναι αρχικά επιλεγμένο ενώ στην περίπτωση όπου η παράμετρος b είναι false τότε το ραδιοπλήκτρο δεν είναι αρχικά επιλεγμένο.

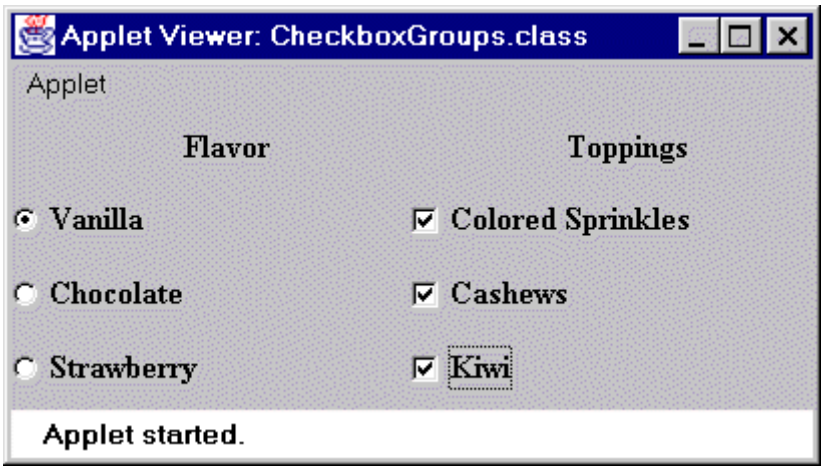
- public JRadioButton(String s)
  Ετούτος ο κατασκευαστής σχηματίζει ένα ραδιοπλήκτρο το οποίο περιέχει τη συμβολοσειρά s δίχως να είναι αρχικά επιλεγμένο.

- public ButtonGroup()
  Ο κατασκευαστής ετούτος είναι ο μοναδικός κατασκευαστής της τάξης ButtonGroup και φυσικά παράγει αντικείμενα της εν λόγω κλάσης.

Η τάξη JRadioButton κληρονομεί από την αφηρημένη κλάση AbstractButton (όπως και η τάξη JCheckBox) τις σπουδαίες μεθόδους isSelected() και setSelected(boolean b) οι οποίες περιγράφηκαν προηγουμένως. Δύο από τις πιο ουσιώδεις συναρτήσεις της κλάσης ButtonGroup είναι οι εξής:

- public void add(AbstractButton a)
  Η μέθοδος ετούτη συνδέει ένα αντικείμενο a της τάξης JRadioButton δηλαδή ένα ραδιοπλήκτρο a στην ομάδα. Εδώ εύλογα θα αναρωτιόταν κάποιος επειδή η συνάρτηση έχει εντός της αντικείμενο της κλάσης AbstractButton και όχι της JRadioButton. Όμως η τάξη JRadioButton είναι υποκλάση της τάξης AbstractButton και συνεπώς μέσα στη μέθοδο δύναται να εισχωρήσει αντικείμενο της JRadioButton.

- public void remove(AbstractButton a)
  Η συνάρτηση αυτή αποσυνδέει το αντικείμενο a της κλάσης JRadioButton τουτέστιν το ραδιοπλήκτρο a από την ομάδα.



Σχήμα 2.23 : Στην εικόνα φαίνεται ένα Applet μέσα στο οποίο στα αριστερά του υπάρχουν τρία ραδιοπλήκτρα με κείμενο και αλληλοαποκλειόμενες επιλογές. Στα δεξιά του Applet διακρίνονται τρία πλαίσια ελέγχου με κείμενο χωρίς όμως αυτές οι πολλαπλές επιλογές να είναι αλληλοαποκλειόμενες.

Η τάξη Graphics μας επιτρέπει να σχεδιάζουμε σχήματα σε διάφορες συσκευές εμφάνισης εικόνων. Για λόγους φορητότητας η κλάση αυτή είναι αφηρημένη καθόσον ο σχεδιασμός των σχημάτων επιτυγχάνεται με διαφορετικό τρόπο σε κάθε πλατφόρμα που υποστηρίζει τη Java. Έτσι όταν η Java υλοποιείται σε κάποια πλατφόρμα δημιουργείται μία νέα τάξη η οποία προέρχεται από την κλάση Graphics. Ετούτη η νέα τάξη είναι αυτή ουσιαστικά που στην πραγματικότητα παρέχει τις ικανότητες σχεδίασης. Η συγκεκριμένη υλοποίηση δεν είναι ορατή στον προγραμματιστή κι έτσι αυτός δύναται να χρησιμοποιήσει τις μεθόδους της κλάσης Graphics για να σχεδιάσει με τρόπο ο οποίος είναι ανεξάρτητος από την εκάστοτε πλατφόρμα υλισμικού και λογισμικού.

Η συνάρτηση paint της τάξης Component λαμβάνει ως παράμετρο ένα αντικείμενο

της κλάσης Graphics που στην πραγματικότητα αποτελεί μία αναφορά ενός αντικειμένου της τάξης που προέρχεται από την κλάση Graphics. Όταν οποιοδήποτε συστατικό γραφικών διεπαφών χρήστη πρόκειται να παρουσιαστεί στην οθόνη τότε το σύστημα αυτομάτως καλεί τη μέθοδο paint. Αντί για αυτή συνήθως καλείται η συνάρτηση repaint. Η τελευταία ζητά να κληθεί η μέθοδος update της κλάσης Component όσο γίνεται ταχύτερα. Η συνάρτηση update καθαρίζει το παρασκήνιο του στοιχείου από οτιδήποτε εμπεριέχει και καλεί άμεσα τη μέθοδο paint. Η μέθοδος update όπως και η συνάρτηση paint παίρνει και αυτή ως παράμετρο ένα αντικείμενο τύπου Graphics το οποίο στην ουσία είναι μία αναφορά αντικειμένου της τάξης που προέρχεται από την κλάση Graphics. Ετούτο το αντικείμενο παρέχεται αυτόματα από το σύστημα στη μέθοδο update. Ένα αντικείμενο τύπου Graphics έχει διάφορες ιδιότητες οι οποίες απαιτούνται για την εμφάνιση του. Μερικές από τις εν λόγω ιδιότητες είναι:

- Το αντικείμενο - συστατικό πάνω στο οποίο θα σχεδιαστεί.
- Τις συντεταγμένες.
- Το τρέχον χρώμα του.
- Το τρέχον font του.
- Και άλλα.

Ορισμένες από τις μεθόδους της τάξης Graphics δίνονται παρακάτω. Όπου δεν αναφέρεται ρητά οι συναρτήσεις αυτές χρησιμοποιούν το τρέχον χρώμα των γραφικών συμφραζομένων. Οι παράμετροι width (πλάτος) και height (ύψος) εκφράζουν εικονοστοιχεία ενώ επιβάλλεται να έχουν μη αρνητικές τιμές, ειδάλλως το σχήμα δε θα εμφανιστεί. Οι συντεταγμένες αυτές μετριούνται από το πάνω αριστερά σημείο του παραθύρου του υποδοχέα. Για παράδειγμα οι συντεταγμένες (0,0) αντιστοιχούν στην άνω αριστερά γωνία του παραθύρου του υποδοχέα. Οι μέθοδοι που παρουσιάζουμε είναι οι εξής:

- public abstract void drawString(String a,int x,int y)
  Ετούτη η συνάρτηση σχεδιάζει τη συμβολοσειρά a ξεκινώντας από τη θέση (x,y).
- public abstract void drawLine(int x1,int y1,int x2,int y2)
  Αυτή η συνάρτηση σχεδιάζει μία γραμμή μεταξύ των σημείων (x1,y1) και (x2,y2).
- public void drawRect(int x,int y,int k,int z)
  Ετούτη η μέθοδος σχεδιάζει ένα ορθογώνιο παραλληλόγραμμο που έχει πλάτος k εικονοστοιχεία και ύψος z εικονοστοιχεία. Η πάνω αριστερή γωνία του ορθογωνίου έχει συντεταγμένες (x,y).
- public abstract void fillRect(int x,int y,int k,int z)
  Αυτή η μέθοδος κάνει ότι και η αμέσως προηγούμενη με τη διαφορά πως το ορθογώνιο παραλληλόγραμμο είναι γεμάτο με το τρέχον χρώμα.
- public abstract void clearRect(int x,int y,int k,int z)
  Ετούτη η συνάρτηση λειτουργεί όπως και η αμέσως προηγούμενη της με τη διαφορά ότι το ορθογώνιο έχει το τρέχον παρασκηνιακό χρώμα της τρέχουσας επιφάνειας σχεδίασης.
- public abstract void drawRoundRect(int x,int y,int k,int z,int q,int p)
  Ετούτη η συνάρτηση σχεδιάζει ένα ορθογώνιο παραλληλόγραμμο που έχει πλάτος k εικονοστοιχεία, ύψος z εικονοστοιχεία και στρογγυλές γωνίες οι οποίες διαθέτουν πλάτος q εικονοστοιχεία και ύψος p εικονοστοιχεία. Η

πάνω αριστερή γωνία του ορθογωνίου δίνεται από τις συντεταγμένες (x,y).

- public abstract void fillRoundRect(int x,int y,int k,int z,int q,int p)
  Αυτή η μέθοδος είναι ίδια με την προηγούμενη με τη διαφορά πως το ορθογώνιο παραλληλόγραμμο με τις στρογγυλές γωνίες είναι γεμάτο με το τρέχον χρώμα.

- public abstract void drawOval(int x,int y,int k,int z)
  Ετούτη η μέθοδος σχεδιάζει μία έλλειψη η οποία έχει πλάτος k εικονοστοιχεία και ύψος z εικονοστοιχεία. Η πάνω αριστερή γωνία του ορθογωνίου παραλληλογράμμου που περικλείει την έλλειψη έχει συντεταγμένες (x,y). Η έλλειψη αγγίζει και τις τέσσερις πλευρές αυτού του ορθογωνίου στο κέντρο τους.

- public abstract void fillOval(int x,int y,int k,int z)
  Αυτή η συνάρτηση λειτουργεί όπως και η αμέσως προηγούμενη της με τη διαφορά ότι η έλλειψη είναι χρωματισμένη με το τρέχον χρώμα.

- public abstract void drawArc(int x,int y,int k,int z,int q,int p)
  Ετούτη η συνάρτηση σχεδιάζει ένα τόξο το οποίο έχει γωνία τόξου p όπου p ακέραιος αριθμός και αναπαριστά μοίρες. Η γωνία του τόξου ξεκινά από την αρχική γωνία q όπου q ακέραιος αριθμός και εκφράζει επίσης μοίρες. Αρνητικές τιμές της παραμέτρου p σημαίνουν σχεδίαση του τόξου με φορά ίδια με αυτήν των δεικτών του ρολογιού ενώ θετικές τιμές σημαίνουν σχεδίαση τόξου με φορά αντίθετη από εκείνη των δεικτών του ρολογιού. Η τιμή 0 της παραμέτρου q αντιστοιχεί στο δεξιότερο σημείο της έλλειψης στην οποία ανήκει το τόξο. Οι παράμετροι x, y προσδιορίζουν το πάνω αριστερό σημείο του ορθογωνίου παραλληλογράμμου στο οποίο περικλείεται η έλλειψη ενώ οι παράμετροι k και z το πλάτος και το ύψος του ορθογωνίου ετούτου αντιστοίχως.

- public abstract void fillArc(int x,int y,int z,int k,int q,int p)
  Αυτή η μέθοδος είναι ίδια με την αμέσως προηγούμενη με τη διαφορά πως το τόξο είναι χρωματισμένο με το τρέχον χρώμα.

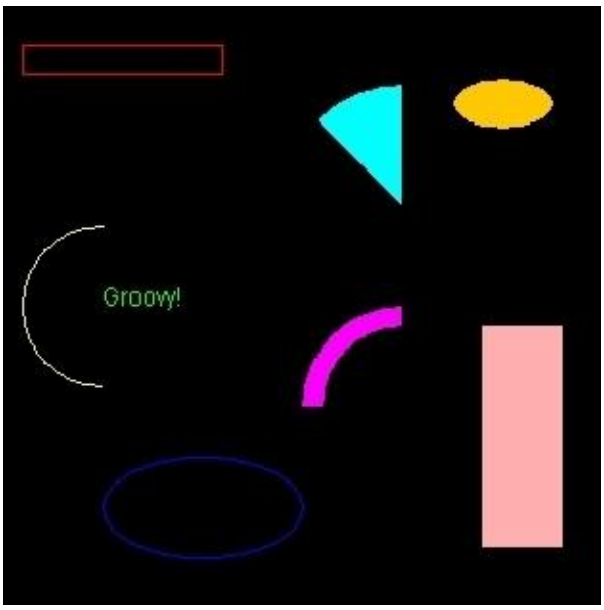
- public abstract Color getColor()
  Ετούτη η μέθοδος επιστρέφει το τρέχον χρώμα των γραφικών συμφραζομένων.

- public abstract void setColor(Color c)
  Αυτή η συνάρτηση θέτει το τρέχον χρώμα ετούτων των γραφικών συμφραζομένων με το χρώμα c.

- public abstract Font getFont()
  Αυτή η μέθοδος επιστρέφει το τρέχον Font τουτέστιν το στυλ και το μέγεθος.

- public abstract void setFont(Font f)
  Ετούτη η μέθοδος θέτει το τρέχον Font δηλαδή το στυλ και το μέγεθος όπως τα προσδιορίζει η παράμετρος f.

- public abstract void translate(int x,int y)
  Αυτή η συνάρτηση μεταφράζει τις προκαταρκτικές συντεταγμένες των γραφικών συμφραζομένων στο σημείο (x,y) του τρέχοντος συστήματος συντεταγμένων. Τροποποιεί τα γραφικά συμφραζόμενα έτσι ώστε η νέα τους αρχή να αντιστοιχεί στο σημείο (x,y) στο προκαταρκτικό σύστημα συντεταγμένων των γραφικών αυτών συμφραζομένων. Όλες οι συντεταγμένες που χρησιμοποιούνται ακολούθως για τα συμφραζόμενα ετούτα θα είναι σχετικές ως προς τη νέα αρχή.



Σχήμα 2.24: Στην εικόνα φαίνεται ένα Applet μέσα στο οποίο υπάρχουν διάφοροι σχεδιασμοί με μεθόδους της κλάσης Graphics.

Η τάξη Color ορίζει σταθερές και μεθόδους για την επεξεργασία χρωμάτων σε προγράμματα της Java. Η κλάση Color ανήκει στο πακέτο java.awt. Οποιοδήποτε χρώμα μπορεί να αναπαραχθεί από συνδυασμό των χρωμάτων κόκκινο (red), πράσινο (green) και μπλε (blue). Οι αναλογίες των χρωμάτων αυτών είναι γνωστές ως τιμές RGB και οι τιμές ετούτες μπορεί να είναι ακέραιες στην περιοχή από 0 εώς 255 ή κινητής υποδιαστολής στην περιοχή 0.0F ως 1.0F. Η πρώτη τιμή προσδιορίζει το ποσοστό του κόκκινου, η δεύτερη το ποσοστό του πράσινου και η τρίτη το ποσοστό του μπλε. Η Java επιτρέπει στον προγραμματιστή να επιλέξει μεταξύ 256x256x256 χρώματα δηλαδή περίπου 16.7 εκατομμύρια διαφορετικά χρώματα. Οι δημόσιες στατικές σταθερές τύπου Color που ορίζει η τάξη αυτή είναι:

Σταθερά	**Χρώμα**	**Τιμή RGB**
black (BLACK)	Μαύρο	0,0,0
blue (BLUE)	Μπλε	0,0,255
cyan (CYAN)	Κυανό	0,255,255
darkGray (DARK_GRAY)	Σκούρο γκρίζο	64,64,64
gray (GRAY)	Γκρίζο	128,128,128
green (GREEN)	Πράσινο	0,255,0
lightGray (LIGHT_GRAY)	Ελαφρύ γκρίζο	192,192,192
magenta (MAGENTA)	Βαθύ κόκκινο	255,0,255
orange (ORANGE)	Πορτοκαλί	255,200,0
pink (PINK)	Ροζ	255,175,175
red (RED)	Κόκκινο	255,0,0
white (WHITE)	Λευκό	255,255,255
yellow (YELLOW)	Κίτρινο	255,255,0

Ορισμένοι από τους κατασκευαστές της κλάσης Color είναι οι ακόλουθοι:

- public Color (int r,int g,int b,int a)
  Ετούτος ο κατασκευαστής δημιουργεί ένα χρώμα το οποίο βασίζεται στις τιμές από 0 εώς 255 των r (κόκκινο), g (πράσινο) και b (μπλε). Η παράμετρος a προσδιορίζει τη διαφάνεια του χρώματος. Αν η τιμή του a είναι 255 το χρώμα είναι τελείως αδιαφανές ενώ εάν η τιμή του είναι ίση με 0 τότε το χρώμα είναι παντελώς

διαφανές.

- public Color (int r,int g,int b)
  Αυτός ο κατασκευαστής λειτουργεί περίπου όπως και ο αμέσως προηγούμενος δηλαδή παράγει ένα χρώμα που στηρίζεται στις τιμές από 0 μέχρι 255 των r (κόκκινο) , g (πράσινο) και b (μπλε). Είναι ισοδύναμος με τον προηγούμενο κατασκευαστή αν η τιμή του a είναι 255.

- public Color (float r,float g,float b,float a)
  Ετούτος ο κατασκευαστής σχηματίζει ένα χρώμα το οποίο βασίζεται στις τιμές από 0.0f εώς 1.0f των r (κόκκινο), g (πράσινο) και b (μπλε) ενώ οι παράμετροι όλοι εδώ είναι τύπου κινητής υποδιαστολής. Η παράμετρος a προδιαγράφει τη διαφάνεια του χρώματος. Εάν το a ισούται με 1.0f τότε το χρώμα είναι ολοκληρωτικά αδιαφανές ενώ σε περίπτωση που ισούται με 0.0f το χρώμα είναι εντελώς διαφανές.

- public Color (float r,float g,float b)
  Αυτός ο κατασκευαστής μοιάζει πολύ με τον αμέσως προηγούμενο. Δημιουργεί ένα χρώμα που στηρίζεται στις τιμές από 0.0f μέχρι 1.0f των r (κόκκινο), g (πράσινο) και b (μπλε). Είναι ισοδύναμος με τον προηγούμενο κατασκευαστή στην περίπτωση όπου το a λάβει την τιμή 1.0f.

Μερικές από τις συναρτήσεις της τάξης Color είναι οι εξής:

- public int getRed()
  Η μέθοδος αυτή επιστρέφει μία τιμή από 0 εώς 255 που εκφράζει το ποσοστό του κόκκινου στο χρώμα.

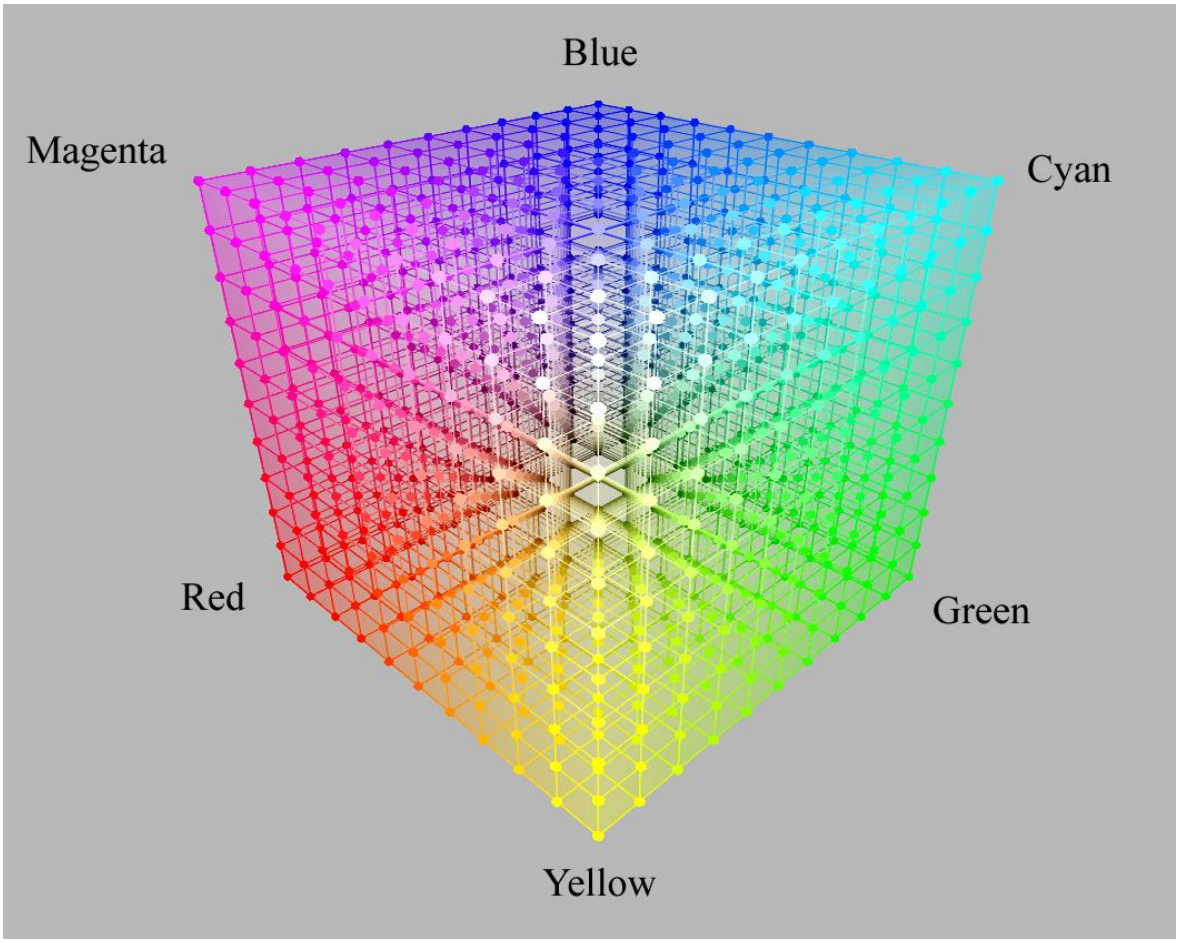
- public int getGreen()
  Η μέθοδος ετούτη επιστρέφει μία τιμή από 0 εώς 255 η οποία παριστάνει το ποσοστό του πράσινου στο χρώμα.

- public int getBlue()
  Η συνάρτηση αυτή επιστρέφει μία τιμή από 0 μέχρι 255 που απεικονίζει το ποσοστό του μπλε στο χρώμα.

- public Color brighter()
  Η συνάρτηση ετούτη επιστρέφει ένα χρώμα το οποίο είναι περισσότερο φωτεινό από αυτό το χρώμα.

- public Color darker()
  Η μέθοδος ετούτη επιστρέφει ένα χρώμα που είναι πιο σκούρο από αυτό το χρώμα.

- public String toString()
  Η μέθοδος ετούτη επιστρέφει μία συμβολοσειρά η οποία αντιπροσωπεύει αυτό το χρώμα. Για παράδειγμα εάν είχαμε θέσει το χρώμα Color (204,204,204) η συνάρτηση θα επέστρεφε τη συμβολοσειρά: java.awt.Color[r=204,g=204,b=204].



Σχήμα 2.25: Στην εικόνα φαίνεται ο διάσημος χρωματικός κύβος RGB

Ένα αντικείμενο το οποίο αποτελεί στιγμιότυπο της κλάσης Font περιέχει πληροφορίες σχετικές με τη γραμματοσειρά. Κάθε γραμματοσειρά έχει όνομα, στυλ και μέγεθος. Τα ονόματα είναι τα ίδια που γνωρίζουμε από την επεξεργασία κειμένου. Υπάρχουν πάρα πολλές γραμματοσειρές και το πλήθος τους σε κάθε υπολογιστή εξαρτάται από το τι έχει εγκατασταθεί στον υπολογιστή αυτόν. Για να υπάρχει κάποια κοινή βάση ορίζονται πέντε συμβολικά ονόματα τα οποία αντιστοιχούν πάντα σε κάποια από τις υπάρχουσες φυσικές γραμματοσειρές. Τα πέντε αυτά ονόματα είναι:

- SansSerif
- Serif
- Monospaced
- Dialog
- DialogInput

Το μέγεθος της γραμματοσειράς ορίζεται σε στιγμές - points. Πρόκειται για μία μονάδα μέτρησης η οποία προέρχεται από την τυπογραφία και δεν αντιστοιχεί στον αριθμό των pixels - εικονοστοιχείων. Το μέγεθος των γραμμάτων μπορούμε να το εντοπίσουμε χρησιμοποιώντας την κλάση FontMetrics. Η κλάση ετούτη όπως και η τάξη Font ανήκουν στο πακέτο java.awt. Η κλάση Font ορίζει σταθερές και

μεθόδους για τον έλεγχο των γραμματοσειρών των χαρακτήρων που παρουσιάζονται στην οθόνη. Μερικές από τις σταθερές που ορίζει η τάξη ετούτη είναι:

- public **Font.BOLD**
  Η συγκεκριμένη σταθερά παριστάνει το τονισμένο στυλ της γραμματοσειράς.

- public Font.*ITALIC*
  Η εν λόγω σταθερά περιγράφει το πλάγιο στυλ της γραμματοσειράς.

- public Font.PLAIN
  Ετούτη η σταθερά απεικονίζει το απλό στυλ της γραμματοσειράς.

Ένας από τους κατασκευαστές της τάξης Font είναι ο επόμενος:

- public Font(String s,int a,int b)
  Αυτός ο κατασκευαστής σχηματίζει ένα αντικείμενο Font στο οποίο η συμβολοσειρά s εκφράζει το όνομα της γραμματοσειράς. Το όνομα της γραμματοσειράς μπορεί να είναι είτε κάποιο από τα πέντε συμβολικά ονόματα που προαναφέρθηκαν είτε το όνομα το οποίο γνωρίζουμε ότι θα υπάρχει στον υπολογιστή που εκτελείται το πρόγραμμα. Αν το όνομα δεν είναι έγκυρο δεν αποστέλλεται μήνυμα σφάλματος. Η παράμετρος a είναι κάποια από τις προειρημένες σταθερές ή κάποιος συνδυασμός αυτών. Ο συνδυασμός μπορεί να πραγματοποιηθεί με τη χρήση του συμβόλου '|' ή '+' όπως για παράδειγμα Font.BOLD | Font.ITALIC. Δηλαδή η παράμετρος a καθορίζει το στυλ της γραμματοσειράς. Τέλος η παράμετρος b διατυπώνει το μέγεθος της γραμματοσειράς το οποίο όπως προαναφέρθηκε ορίζεται σε στιγμές - points.

Ορισμένες από τις συναρτήσεις που παρέχει η τάξη Font είναι οι ακόλουθες:

- public int getStyle()
  Η μέθοδος αυτή επιστρέφει μία ακέραιη τιμή που περιγράφει το τρέχον στυλ της γραμματοσειράς. Η ακέραιη ετούτη τιμή μπορεί να είναι PLAIN, ITALIC, BOLD ή BOLD και ITALIC.

- public int getSize()
  Η συνάρτηση αυτή επιστρέφει μία ακέραιη τιμή η οποία απεικονίζει το τρέχον μέγεθος της γραμματοσειράς. Το μέγεθος αφού είναι σε στιγμές έχει στρογγυλοποιηθεί σε ακέραιο αριθμό.

- public String getName()
  Η μέθοδος ετούτη επιστρέφει το όνομα της γραμματοσειράς ως συμβολοσειρά.

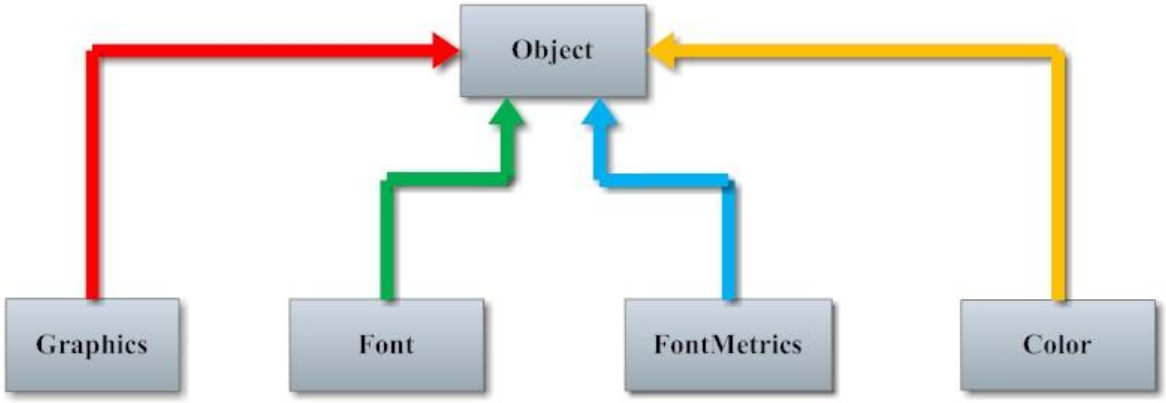
- public String getFamily()
  Η συνάρτηση ετούτη επιστρέφει το τρέχον όνομα της οικογένειας της γραμματοσειράς ως συμβολοσειρά.

- public boolean isPlain()
  Η μέθοδος αυτή επιστρέφει την τιμή true αν το στυλ της γραμματοσειράς είναι απλό ειδάλλως επιστρέφει την τιμή false.

- public boolean isBold()
  Η μέθοδος ετούτη επιστρέφει την τιμή true αν το στυλ της γραμματοσειράς είναι τονισμένο αλλιώς επιστρέφει την τιμή false.

- public boolean isItalic()
  Η συνάρτηση αυτή επιστρέφει την τιμή true εάν το στυλ της γραμματοσειράς είναι πλάγιο ειδάλλως επιστρέφει την τιμή false.

- public float getSize2D()
  Η συνάρτηση ετούτη επιστρέφει μία τιμή κινητής υποδιαστολής η οποία δηλώνει το τρέχον μέγεθος της γραμματοσειράς σε στιγμές.



Σχήμα 2.26 : Στην εικόνα φαίνεται η ιεραρχία των κλάσεων Object, Graphics, Font, FontMetrics και Color. Η υπερκλάση είναι το κουτί της εικόνας στο οποίο πάει το βέλος και η υποκλάση είναι το κουτί της εικόνας από όπου προέρχεται το βέλος.

Σε περίπτωση όπου έχουμε μεγάλο αριθμό πολλαπλών επιλογών αντί να χρησιμοποιήσουμε τα πλαίσια ελέγχου ή τα ραδιοπλήκτρα μπορούμε να κάνουμε χρήση μίας λίστας η οποία είναι εξαιρετικά ευέλικτη. Στη λίστα μπορεί να υπάρχουν κείμενα από τα οποία ο χρήστης θα δύναται να επιλέξει ένα ή περισσότερα. Για να δημιουργήσουμε μία λίστα επιβάλλεται να χρησιμοποιήσουμε την κλάση JList. Η τάξη JList ανήκει στο πακέτο javax.swing ενώ αν πρόκειται να μεταβάλλουμε τα περιεχόμενα της λίστας πρέπει να κάνουμε χρήση της κλάσης DefaultListModel που επίσης ανήκει στο πακέτο javax.swing. Η κλάση DefaultListModel τηρεί μία λίστα με τα περιεχόμενα των δεδομένων ενός στοιχείου της JList. Δύο από τους κατασκευαστές της τάξης JList είναι οι παρακάτω:

- public JList()
  Ο κατασκευαστής αυτός σχηματίζει μία λίστα με κενό μοντέλο. Μοντέλο σημαίνει τα δεδομένα της λίστας.

- public JList(Object data[])
  Ο κατασκευαστής ετούτος παράγει μία λίστα της οποίας τα δεδομένα είναι τύπου Object. Δηλαδή τα δεδομένα δύνανται να είναι εκτός από συμβολοσειρές και κάτι διαφορετικό από τις συμβολοσειρές.

Ορισμένες από τις μεθόδους της κλάσης JList είναι οι επόμενες:

- public void addListSelectionListener(ListSelectionListener a)
  Αυτή η συνάρτηση προσθέτει τον ακροατή a ο οποίος αφυπνίζεται κάθε φορά που υπάρχει αλλαγή στις επιλογές της λίστας.

- public void setSelectionMode(int a)
  Ετούτη η συνάρτηση αποφασίζει ποιον τύπο επιλογών έχει τη δυνατότητα να κάνει ο χρήστης. Το όρισμα a πρέπει να είναι ένα από τα παρακάτω τα οποία επιβάλλεται να προσδιορίζονται με το όνομα της διασύνδεσης της κλάσης

ListSelectionModel η οποία τάξη ανήκει στο πακέτο javax.swing:

- SINGLE_SELECTION
  Αυτό το όρισμα σημαίνει πως ο χρήστης μπορεί να επιλέγει μόνο ένα δεδομένο της λίστας κάθε φορά. Όταν ο χρήστης κάνει την επιλογή του, οποιαδήποτε προηγούμενη επιλογή του μετασχηματίζεται πρώτα σε μη επιλεγμένη.

- SINGLE_INTERVAL_SELECTION
  Ετούτο το όρισμα σημαίνει ότι χρήστης δύναται να επιλέγει πολλαπλά και γειτονικά δεδομένα της λίστας. Όταν ο χρήστης κάνει την επιλογή του, οποιεσδήποτε προηγούμενες επιλογές του μετατρέπονται πρώτα σε μη επιλεγμένες.

- MULTIPLE_INTERVAL_SELECTION
  Αυτό το όρισμα σημαίνει πως ο χρήστης μπορεί να διαλέξει τα δεδομένα της λίστας με οποιονδήποτε συνδυασμό. Επίσης ο χρήστης πρέπει ρητά να κάνει τα δεδομένα ως μη επιλεγμένα ενώ το συγκεκριμένο όρισμα αποτελεί την προεπιλογή σε περίπτωση που δε θα δηλωθεί κάτι άλλο.

- public void clearSelection()
  Ετούτη η μέθοδος καθαρίζει τις επιλογές οι οποίες έγιναν. Σε όλους τους καταχωρισμένους ακροατές διαβιβάζεται η συνάρτηση valueChanged. Δηλαδή αν θυμηθούμε πως φτιάχνεται μία εσωτερική κλάση η οποία διασυνδέεται με την τάξη ActionListener και μέσα της έχει τη μέθοδο actionPerformed το αντίστοιχο γίνεται κι εδώ. Η διαφορά έγκειται τώρα στο ότι η εσωτερική κλάση του ακροατή ListListener διασυνδέεται με την τάξη ListSelectionListener και μέσα της έχει τη συνάρτηση valueChanged. Η κλάση ListSelectionListener ανήκει στο πακέτο javax.swing.event.

- public int getSelectedIndex()
  Αυτή η μέθοδος επιστρέφει τον αριθμοδείκτη της επιλογής η οποία έγινε. Αν πραγματοποιήθηκε πολλαπλή επιλογή τότε επιστρέφει τον αριθμοδείκτη της πρώτης επιλογής. Σε περίπτωση που δεν έγινε επιλογή η συνάρτηση επιστρέφει την τιμή -1.

- public int [] getSelectedIndices()
  Ετούτη η συνάρτηση επιστρέφει έναν πίνακα με τους αριθμοδείκτες των επιλογών που έγιναν. Οι αριθμοδείκτες είναι σε αύξουσα σειρά.

- public Object setSelectedValue()
  Αυτή η συνάρτηση επιστρέφει μία αναφορά στην επιλογή η οποία έγινε. Εάν δεν έχει συμβεί επιλογή τότε επιστρέφεται η τιμή null.

- public Object [] getSelectedValues()
  Ετούτη η μέθοδος επιστρέφει έναν πίνακα αναφορών προς τις επιλογές που πραγματοποιήθηκαν.

- public boolean isSelectedIndex(int a)
  Αυτή η μέθοδος επιστρέφει true εάν έχει επιλεγεί η γραμμή με τον αριθμοδείκτη a. Ειδάλλως επιστρέφει false.

- public boolean isSelectionEmpty()
  Ετούτη η συνάρτηση επιστρέφει true αν δεν έχει πραγματοποιηθεί επιλογή. Αλλιώς επιστρέφει false.

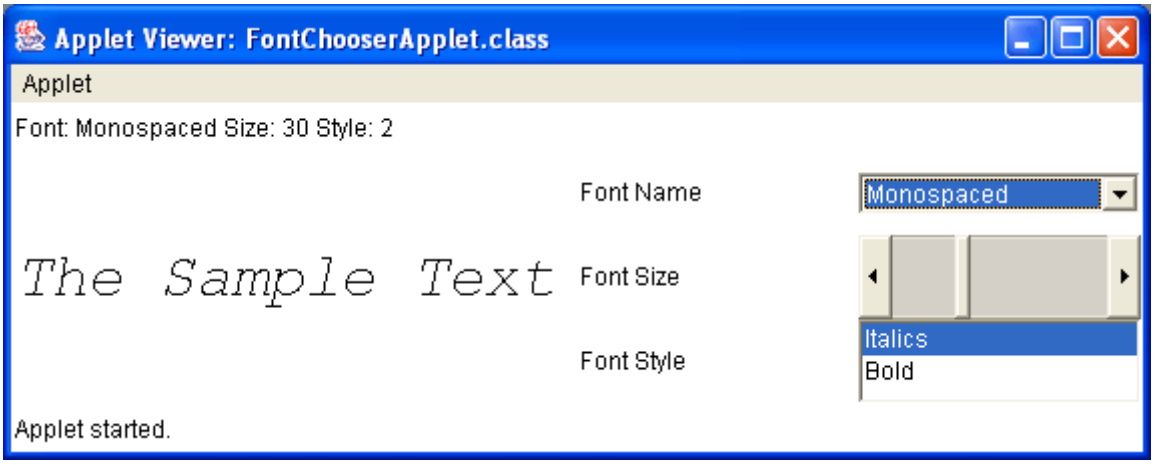
- public void setSelectedIndex(int a)
  Αυτή η συνάρτηση επιτρέπει την ενεργοποίηση μίας επιλογής με αριθμοδείκτη a.
- public void setSelectedIndices(int a [])
  Ετούτη η μέθοδος επιτρέπει την ενεργοποίηση των επιλογών με αριθμοδείκτες οι οποίοι βρίσκονται μέσα στον πίνακα a.

- public void setSelectedValue(Object a,boolean b)
  Αυτή η μέθοδος επιτρέπει να οριστεί το συγκεκριμένο αντικείμενο a ως επιλεγμένο. Χρησιμοποιούμε στη θέση της παραμέτρου b την τιμή true όταν η λίστα πρέπει να κυλάει ώστε να εμφανίζεται η επιλογή. Αλλιώς βάζουμε στη θέση του b την τιμή false.

- public boolean getValueIsAdjusting()
  Κάθε επιλογή διαβιβάζει πολλά μηνύματα από τη συνάρτηση valueChanged στο αντικείμενο του ακροατή. Σε περίπτωση που είναι σημαντικό για εμάς να χειριζόμαστε μόνο ένα από αυτά μπορούμε να χρησιμοποιήσουμε τη συνάρτηση αυτή. Ετούτη η μέθοδος επιστρέφει true για όλα τα μηνύματα εκτός από το τελευταίο μίας επιλογής, για το οποίο επιστρέφεται η τιμή false.

Η εσωτερική κλάση του ακροατή που δημιουργείται και έχει διασύνδεση με την τάξη ListSelectionListener προαναφέρθηκε ότι περιέχει τη μέθοδο:

- public void valueChanged(ListSelectionEvent a)
  Με τη συνάρτηση αυτή διαβιβάζονται τα μηνύματα στα αντικείμενα των ακροατών όταν γίνονται επιλογές στη λίστα. Η επιλογή δύναται να συμβεί είτε από το χρήστη είτε για παράδειγμα με τη μέθοδο setSelectedIndex. Όσον αφορά την κλάση ListSelectionEvent ανήκει στο πακέτο javax.swing.event.



Σχήμα 2.27 : Στην εικόνα φαίνεται ένα Applet στο οποίο διακρίνουμε τη συμβολοσειρά The Sample Text. Αυτή η συμβολοσειρά σύμφωνα με τις πληροφορίες που αντλούμε από το Applet έχει όνομα γραμματοσειράς Monospaced, στυλ γραμματοσειράς πλάγιο και μέγεθος γραμματοσειράς 30 στιγμές.

Η τάξη DefaultListModel έχει τον ακόλουθο κατασκευαστή:

- public DefaultListModel()
  Ο κατασκευαστής ετούτος παράγει ένα αντικείμενο της κλάσης DefaultListModel.

Ορισμένες από τις μεθόδους της κλάσης DefaultListModel είναι οι εξής:

- public void add(int a,Object b)
  Αυτή η συνάρτηση δημιουργεί χώρο για μία νέα αναφορά b μετακινώντας όλες τις αναφορές μία θέση προς τα κάτω αρχίζοντας από τη θέση a. Η νέα αναφορά b τοποθετείται στη θέση a.

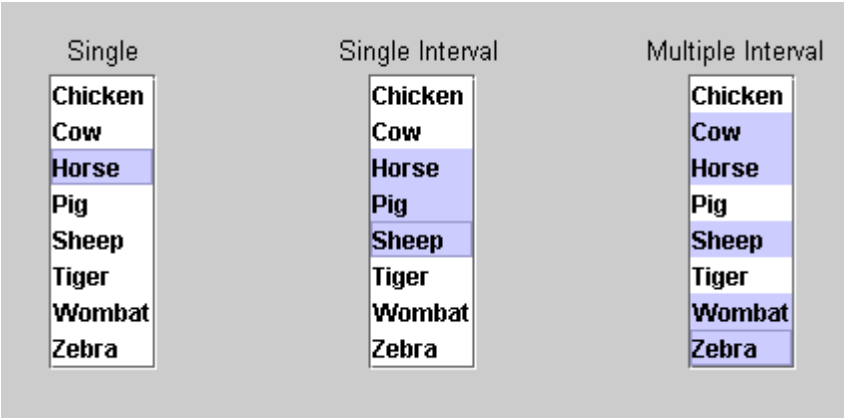
- public void addElement(Object a)
  Ετούτη η συνάρτηση εισάγει μία αναφορά a προς ένα αντικείμενο στο τέλος της λίστας.

- public Object get(int a)
  Αυτή η μέθοδος επιστρέφει την αναφορά με τον καθορισμένο αριθμοδείκτη a.

- public Object remove(int a)
  Ετούτη η μέθοδος αφαιρεί την αναφορά με αριθμοδείκτη a από τη λίστα. Τα στοιχεία μεταφέρονται κατά μία θέση πλησιέστερα στην αρχή της λίστας, ξεκινώντας από τη θέση όπου υπήρχε κενό διάστημα και προχωρώντας από εκεί. Η συνάρτηση επιστρέφει την αναφορά.

- public Object set(int a,Object b)
  Αυτή η μέθοδος αντικαθιστά την αναφορά στον καθορισμένο αριθμοδείκτη a με την αναφορά b.

- public int size()
  Ετούτη η συνάρτηση επιστρέφει το πλήθος των στοιχείων της λίστας.



Σχήμα 2.28 : Στην εικόνα φαίνονται τρία αντικείμενα της τάξης JList δηλαδή τρεις λίστες. Στα αριστερά της εικόνας βρίσκεται μία λίστα η οποία χρησιμοποιεί τη μέθοδο setSelection(int a) και στη θέση του a έχει μπει το όρισμα SINGLE_SELECTION τουτέστιν μπορεί να επιλεχθεί μόνο ένα δεδομένο κάθε φορά. Στη μέση της εικόνας στη θέση του a έχει εισχωρήσει το όρισμα SINGLE_INTERVAL_SELECTION πράγμα που σημαίνει ότι δύνανται να επιλεχθούν ένα ή περισσότερα δεδομένα αλλά με την προϋπόθεση πως είναι γειτονικά. Στα δεξιά της εικόνας στη θέση του a έχει παρεισδύσει το όρισμα MULTIPLE_INTERVAL_SELECTION το οποίο σημαίνει ότι ο χρήστης μπορεί να επιλέξει δεδομένα με οποιονδήποτε συνδυασμό.

Στη γλώσσα Java πολύ συχνά απαιτείται να είμαστε σε θέση να αντιμετωπίσουμε διάφορα σφάλματα στη συγγραφή του κώδικα μας όπως στην περίπτωση της διαίρεσης με το μηδέν και σε άλλες δύσκολες ή παράξενες καταστάσεις. Η αντιμετώπιση τέτοιων καταστάσεων δύναται να πραγματοποιηθεί στα σημεία του προγράμματος όπου αυτές μπορεί να συμβούν. Το πλεονέκτημα ετούτου του τρόπου αντιμετώπισης των

παράξενων καταστάσεων είναι πως μπορούμε να διαπιστώσουμε στην άμεση γειτονία των αποσπασμάτων του προγράμματος όπου δύναται να συμβούν τέτοιες καταστάσεις, αν έχει υλοποιηθεί επαρκώς ο έλεγχος των σφαλμάτων που μπορεί να γίνουν.

Το μειονέκτημα είναι ότι το πρόγραμμα μας καταλήγει να έχει πάρα πολλές εντολές αντιμετώπισης σφαλμάτων, με αποτέλεσμα να είναι δύσκολη η κατανόηση και η συντήρηση του. Επιπρόσθετα υπάρχουν περιπτώσεις στις οποίες μία τάξη δεν είναι σε θέση να αντιμετωπίσει όλα τα σφάλματα που είναι πιθανό να προκύψουν από τη χρήση της, με τρόπο ο οποίος να ικανοποιεί κάθε απόσπασμα προγράμματος (κάθε πελάτη) που τη χρησιμοποιεί. Για την αντιμετώπιση τέτοιων παράξενων καταστάσεων η γλώσσα Java παρέχει το μηχανισμό χειρισμού εξαιρέσεων - exception handling με τον οποίο δυνάμεθα να παγιδεύσουμε - trap και να συλλάβουμε - catch τέτοιου είδους καταστάσεις και να τις χειριστούμε σε ένα ή περισσότερα προσδιορισμένα σημεία του προγράμματος μας με ενιαίο τρόπο.
Εάν μία μέθοδος έχει τη δυνατότητα να αντιμετωπίσει τοπικά τα σφάλματα της, τότε είναι προτιμότερο να το πραγματοποιήσει η ίδια, αποφεύγοντας έτσι την επιβάρυνση που προκαλεί ο μηχανισμός εξαιρέσεων της γλώσσας Java. Ο μηχανισμός εξαιρέσεων πρέπει να χρησιμοποιείται στις περιπτώσεις όπου τα σφάλματα μας επιβάλλεται να τύχουν χειρισμού από μία συνάρτηση διαφορετική από εκείνη που συμβαίνουν, αν γίνουν τέτοια σφάλματα. Οι καταστάσεις οι οποίες προξενούν εξαιρέσεις είναι σύγχρονες με το πρόγραμμα του χρήστη δηλαδή όταν αυτό εκτελείται προκαλούνται άμεσα από ετούτο και συμβαίνουν πάντοτε στα ίδια σημεία του. Οι καταστάσεις αυτές περιλαμβάνουν:

- Αριθμητικές Συνθήκες - Arithmetic Conditions
  Τέτοιου τύπου συνθήκες είναι για παράδειγμα η διαίρεση με το μηδέν.

- Προστασία - Protection
  Σε τέτοιες καταστάσεις γίνεται προσπάθεια του προγράμματος να μεταβάλλει ή απλώς να προσπελάσει ένα μέρος της μνήμης που δεν του ανήκει. Σχετικά παραδείγματα περιλαμβάνουν την υπέρβαση των ορίων ενός πίνακα ή μίας συμβολοσειράς και την υπερχείλιση της στοίβας του χρόνου εκτέλεσης του προγράμματος.

- Εξαιρέσεις Χρήστη - User Exceptions
  Η Java επιτρέπει στους χρήστες όχι μόνο να προσδιορίζουν πως πρέπει να αντιμετωπιστούν οι αιτίες οι οποίες προκαλούν εξαιρέσεις αλλά και να προξενούν οι ίδιοι τις δικές τους εξαιρέσεις.

Ο μηχανισμός διαχείρισης εξαιρέσεων της Java μας επιτρέπει να συλλάβουμε όλες τις εξαιρέσεις ενός ορισμένου τύπου ή όλες τις εξαιρέσεις συναφών τύπων ή όλους τους τύπους των εξαιρέσεων που μπορεί να προκύψουν κατά την εκτέλεση ενός προγράμματος. Για να χειριστούμε τις εξαιρέσεις πρέπει να γράψουμε try { } και μέσα στα άγκιστρα να τοποθετήσουμε τις εντολές οι οποίες πιθανότατα θα πετάξουν κάποιο exception. Κατόπιν επιβάλλεται να γράψουμε catch(Exception e) { } όπου μέσα στα άγκιστρα λέμε στον compiler τι να κάνει μετά την εμφάνιση της εξαίρεσης. Υπάρχουν πολλών ειδών εξαιρέσεις και μπορούμε να χειριστούμε τον κάθε τύπο εξαίρεσης ξεχωριστά. Ο απλούστερος όμως τρόπος για το χειρισμό των εξαιρέσεων είναι αυτός που μόλις περιγράφηκε δηλαδή να θέσουμε μέσα στο catch την κλάση Exception. Πράττοντας τοιουτοτρόπως μετά την εξαίρεση η λειτουργία του προγράμματος συνεχίζεται κανονικά χωρίς να διακοπεί. Η τάξη Exception ανήκει στο πακέτο java.lang ενώ δύο από τους κατασκευαστές της είναι οι ακόλουθοι:

- public Exception()
  Ετούτος ο κατασκευαστής απλά σχηματίζει ένα αντικείμενο της κλάσης Exception.

- public Exception(String s)
  Αυτός ο κατασκευαστής δημιουργεί ένα αντικείμενο της τάξης Exception με παράμετρο τη συμβολοσειρά s. Η παράμετρος s μεταβιβάζεται σε κάθε αντικείμενο της κλάσης Exception και είναι πολύ χρήσιμη όταν απαιτείται να προσδιοριστεί η πηγή της εξαίρεσης μέσω της συνάρτησης getMessage.

Η μέθοδος getMessage ανήκει στην τάξη Throwable την οποία κλάση κληρονομεί η τάξη Exception. Η κλάση Throwable ανήκει στο πακέτο java.lang ενώ η συνάρτηση getMessage περιγράφεται παρακάτω:

- public String getMessage()
  Ετούτη η μέθοδος επιστρέφει μία συμβολοσειρά η οποία προσδιορίζει την πηγή της εξαίρεσης.

Συνοψίζοντας για τις εξαιρέσεις το έργο του σχεδιασμού και της ανάπτυξης του κώδικα ενός προγράμματος θα μπορούσε να υποδιαιρεθεί σε δύο ανεξάρτητα μέρη. Την περίπτωση που δε συμβαίνει κάτι ασυνήθιστο και την περίπτωση στην οποία δύναται να συμβεί οτιδήποτε περίεργο. Η πιο σημαντική χρήση των μηχανισμών χειρισμού των εξαιρέσεων είναι για την αντιμετώπιση μεθόδων στις οποίες μπορεί να συμβεί κάποια ειδική περίπτωση που ο χειρισμός είναι πιθανό να διαφέρει ανάλογα με τον τρόπο που χρησιμοποιείται η μέθοδος. Για παράδειγμα εάν σε μία συνάρτηση υπάρχει διαίρεση με το μηδέν τότε αποδεικνύεται πως για κάποιες κλήσεις της συνάρτησης το πρόγραμμα θα πρέπει να τερματίζεται ενώ για κάποιες άλλες κλήσεις της συνάρτησης θα πρέπει να γίνεται κάτι διαφορετικό. Η μέθοδος ετούτη δύναται να οριστεί τοιουτοτρόπως ώστε αν παρουσιαστεί μία ειδική περίπτωση να παραχθεί ένα σήμα το οποίο με τη σειρά του θα πυροδοτήσει το μηχανισμό που θα χειριστεί την ειδική περίπτωση σε κάποιο άλλο σημείο του προγράμματος το οποίο βρίσκεται έξω από τη συνάρτηση. Έτσι δίνεται η δυνατότητα στον προγραμματιστή να χειρίζεται διαφορετικά τις ειδικές περιπτώσεις που μπορεί να προκύψουν για διαφορετικές κλήσεις της μεθόδου. Στη Java ο χειρισμός των εξαιρέσεων εξελίσσεται είτε ως κάποιο λογισμικό από τις βιβλιοθήκες της Java είτε ως μέρος του κώδικα του προγραμματιστή το οποίο παρέχει ένα μηχανισμό που δημιουργεί ένα σήμα μόλις συμβεί κάτι ασυνήθιστο. Τέλος επιβάλλεται να μνημονευτεί ότι ο χειρισμός των εξαιρέσεων προορίζεται για φειδωλή χρήση δηλαδή κυρίως για περιπτώσεις οι οποίες είναι πολύπλοκες και όχι για περιπτώσεις υπερβολικά απλοϊκές.

Σχήμα 2.29 : Η γλώσσα Java αναμφισβήτητα αποτελεί σήμερα την καλύτερη γλώσσα προγραμματισμού σε ολόκληρη την οικουμένη. Το γεγονός αυτό οφείλεται στο ότι γράφτηκε πάνω στις σπουδαίες γλώσσες C και C++ αλλά αφαιρέθηκαν οι δείκτες και οι μνήμες. Έτσι διατήρησε τις δυνατότητες των γλωσσών C και C++ αλλά έγινε ευκολότερη στην εκμάθηση ενώ επιπρόσθετα εμπλουτίστηκε με ακόμη περισσότερες δυνατότητες. Το μόνο κόστος της συγγραφής της γλώσσας Java πάνω στις γλώσσες C και C++ είναι ότι έχει μικρότερη ταχύτητα από ετούτες. Ωστόσο η ταχύτητα της Java δεν είναι απαγορευτικά αργή και έτσι λοιπόν πολύ δίκαια κατέληξε να θεωρείται η πιο αξιόλογη γλώσσα προγραμματισμού στον κόσμο.

Στα γραφικά της γλώσσας Java μπορούμε να δημιουργήσουμε μενού με την κλάση JMenu η οποία ανήκει στο πακέτο javax.swing. Τα μενού δύνανται να έχουν υπομενού. Κάθε ξεχωριστό στοιχείο μενού είναι αντικείμενο της τάξης JMenuItem που επίσης ανήκει στο πακέτο javax.swing και μπορεί να έχει κείμενο ή κείμενο και εικονίδιο μαζί. Ακόμη το ξεχωριστό στοιχείο μενού δύναται να αποτελείται από ένα ραδιοπλήκτρο ή ένα πλαίσιο ελέγχου με συνοδευτικό κείμενο. Διακρίνουμε πτυσσόμενα μενού (dropdown menus) και αναδυόμενα μενού (pop - up menus). Αναδυόμενο μενού είναι ένα μενού το οποίο για παράδειγμα εμφανίζεται όταν ο χρήστης πατήσει το δεξιό πλήκτρο του ποντικιού. Επιπρόσθετα το μενού πρέπει να βρίσκεται σε μία γραμμή μενού που σχηματίζεται από την κλάση JMenuBar η οποία ομοίως ανήκει στο πακέτο javax.swing. Ένας από τους κατασκευαστές της κλάσης JMenu είναι ο εξής:

- public JMenu(String a)
  Αυτός ο κατασκευαστής σχηματίζει ένα μενού το οποίο έχει ως κείμενο τη συμβολοσειρά a.

Τρεις από τις μεθόδους της τάξης JMenu είναι οι εξής:

- public JMenuItem add(JMenuItem a)
  Ετούτη η μέθοδος προσθέτει στο μενού ένα νέο στοιχείο μενού a και επιστρέφει αυτό το νέο στοιχείο μενού a.

- public JMenuItem add(String s)
  Ετούτη η συνάρτηση προσθέτει στο μενού ένα νέο στοιχείο μενού το οποίο έχει τη συμβολοσειρά s και επιστρέφει αυτό το νέο στοιχείο μενού. Στην ουσία η μέθοδος ετούτη δημιουργεί ένα στιγμιότυπο της τάξης JMenuItem και αποτελεί μία σύντομη οδό παραγωγής νέων καταχωρίσεων για τα μενού.

- public void addSeparator()
  Αυτή η συνάρτηση προσαρτεί μία διαχωριστική γραμμή μεταξύ των καταχωρίσεων του μενού.

Ο μοναδικός κατασκευαστής της κλάσης JMenuBar είναι ο ακόλουθος:

- public JMenuBar()
  Ετούτος ο κατασκευαστής δημιουργεί μία γραμμή μενού όπου μπαίνουν τα μενού.

Μία από τις μεθόδους της τάξης JMenuBar είναι η επόμενη:

- public JMenu add(JMenu a)
  Η συνάρτηση αυτή προσθέτει ένα νέο πτυσσόμενο μενού a στη γραμμή μενού και επιστρέφει το νέο πτυσσόμενο μενού a.

Ορισμένοι από τους κατασκευαστές της κλάσης JMenuItem είναι οι εξής:

- public JMenuItem(Icon a)
  Ετούτος ο κατασκευαστής παράγει ένα στοιχείο μενού το οποίο αποτελείται από την εικόνα a.

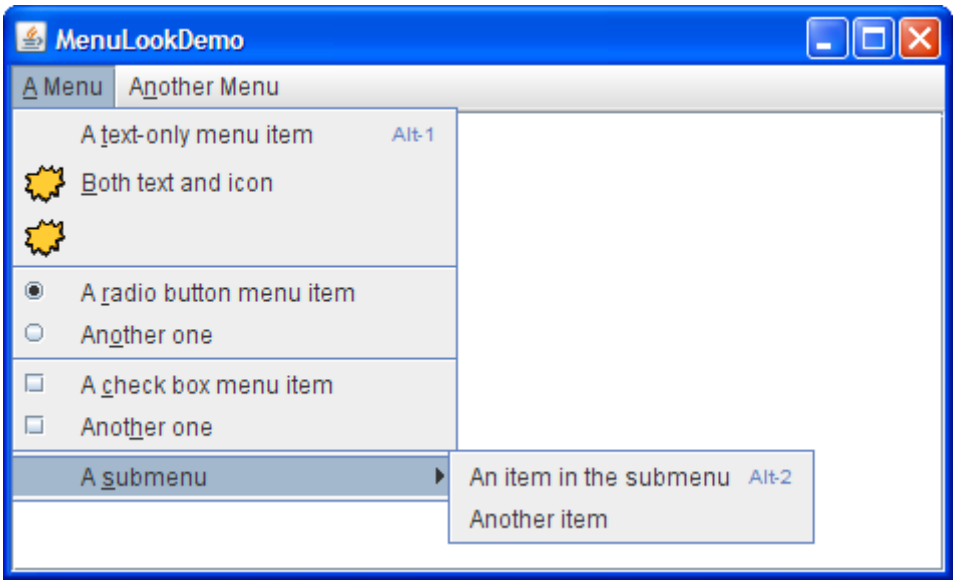
- public JMenuItem(String s)
  Αυτός ο κατασκευαστής σχηματίζει ένα στοιχείο μενού που αποτελείται από τη συμβολοσειρά s.

- public JMenuItem(String s,Icon a)
  Ετούτος ο κατασκευαστής δημιουργεί ένα στοιχείο μενού το οποίο αποτελείται από τη συμβολοσειρά s και το εικονίδιο a.

- public JMenuItem(String s,int a)
  Αυτός ο κατασκευαστής παράγει ένα στοιχείο μενού που αποτελείται από τη συμβολοσειρά s. Επιπλέον γίνεται προσάρτηση ενός μνημονικού χαρακτήρα a. Τουτέστιν ορίζεται πλήκτρο συντόμευσης που συνοδεύει την καταχώριση στο μενού. Στο μενού ο εν λόγω χαρακτήρας παρουσιάζεται ως υπογραμμισμένος.

Σε αυτό το σημείο επιβάλλεται να γίνει αναφορά σε μία μέθοδο της τάξης JApplet που είναι πανομοιότυπη με μία συνάρτηση της κλάσης JFrame. Η μέθοδος είναι η ακόλουθη:

- public void setJMenuBar(JMenuBar a)
  Ετούτη η συνάρτηση τοποθετεί τη γραμμή μενού a στη μικροεφαρμογή.



Σχήμα 2.30 : Στο επάνω μέρος της εικόνας του συγκεκριμένου παραθύρου που έχει δημιουργηθεί από την τάξη JFrame διακρίνεται η γραμμή μενού η οποία εμπεριέχει δύο μενού. Το πρώτο μενού περιλαμβάνει διάφορα στοιχεία μενού καθώς και ένα υπομενού. Το πρώτο στοιχείο μενού καθώς και το πρώτο στοιχείο υπομενού περιέχουν τους μνημονικούς χαρακτήρες Alt-1 και Alt-2 αντιστοίχως. Ομοίως το υπομενού του πρώτου μενού έχει το μνημονικό χαρακτήρα s. Ακόμη φαίνονται διαχωριστικές γραμμές μεταξύ των καταχωρίσεων του πρώτου μενού.

Η κλάση JToolBar μοιάζει πάρα πολύ με την τάξη JMenuBar αλλά χρησιμοποιείται στο αντί να περιλαμβάνει μενού, να περιέχει κυρίως κουμπιά με εικόνες ή κουμπιά με κείμενο ή κουμπιά με εικόνες και κείμενο. Δηλαδή η τάξη JToolBar δημιουργεί μία γραμμή κουμπιών που ονομάζεται και γραμμή εργαλείων. Φυσικά το πως θα είναι τα κουμπιά της κλάσης JButton προδιαγράφεται από τον κατασκευαστή του εκάστοτε κουμπιού. Τα κουμπιά δύνανται να τοποθετηθούν οριζόντια ή κάθετα μέσα στη γραμμή εργαλείων και κάθε κουμπί επιβάλλεται να συνδεθεί ξεχωριστά με τη γραμμή εργαλείων. Η τάξη JToolBar ανήκει στο πακέτο javax.swing ενώ ένας από τους κατασκευαστές της είναι ο επόμενος:

- public JToolBar()
  Ο κατασκευαστής ετούτος σχηματίζει ένα αντικείμενο γραμμής εργαλείων με οριζόντιο προσανατολισμό ο οποίος είναι και ο προκαθορισμένος.

Μερικές από τις μεθόδους της κλάσης JToolBar είναι οι ακόλουθες:

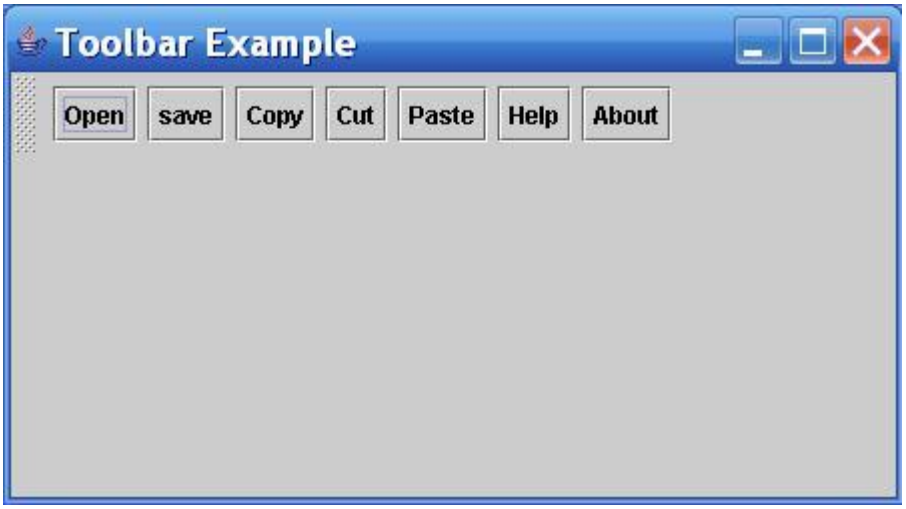
- public void setFloatable(boolean b)
  Σε αυτήν τη συνάρτηση εάν η παράμετρος b είναι true τότε η γραμμή εργαλείων είναι κινητή δηλαδή ο χρήστης μπορεί να τη μετακινήσει. Σε ετούτη την περίπτωση η γραμμή εργαλείων ουσιαστικά μετατρέπεται σε παράθυρο. Αν η παράμετρος b είναι false τότε η γραμμή εργαλείων είναι ακίνητη. Η προεπιλογή για τις γραμμές εργαλείων είναι να είναι κινητές.

- public void addSeparator()
  Ετούτη η μέθοδος προσθέτει στη γραμμή εργαλείων ένα διαχωριστικό.

- public void setBorderPainted(boolean b)
  Σε ετούτη τη συνάρτηση αν η παράμετρος b είναι true τότε το περίγραμμα έχει τη δυνατότητα να χρωματιστεί. Ειδάλλως αν η παράμετρος b είναι false τότε το περίγραμμα δε μπορεί να χρωματιστεί. Η προεπιλογή είναι να έχει η παράμετρος b την τιμή true.

- public boolean isBorderPainted()
  Αυτή η μέθοδος επιστρέφει false αν στην αμέσως προηγούμενη συνάρτηση η παράμετρος b έχει τεθεί ως false. Αλλιώς επιστρέφει true.

- public boolean isFloatable()
  Ετούτη η μέθοδος επιστρέφει true εάν τη γραμμή εργαλείων ο χρήστης δύναται να τη μετακινήσει. Στην αντίθετη περίπτωση επιστρέφει false.



Σχήμα 2.31 : Στην εικόνα φαίνεται ένα παράθυρο το οποίο έχει δημιουργηθεί από την τάξη JFrame. Στο πάνω μέρος της εικόνας διακρίνεται μία γραμμή εργαλείων που αποτελείται από κουμπιά.

Οι αλληλεπιδράσεις με τα προγράμματα των γραφικών της γλώσσας Java επιτυγχάνονται μέσω διαλόγων. Σε ένα διάλογο εισόδου ο χρήστης δύναται να πληκτρολογήσει μία συμβολοσειρά για είσοδο στο πρόγραμμα του ενώ σε ένα διάλογο εξόδου μπορεί να πιέσει το κουμπί OK για να κλείσει το κουτί που περιέχει τα αποτελέσματα αυτού. Παρόλο που τέτοιοι διάλογοι διευκολύνουν την είδοδο και την έξοδο ενός προγράμματος παρέχουν περιορισμένες δυνατότητες. Ένας διάλογος εισόδου δύναται να πάρει μία τιμή κάθε φορά από το χρήστη κι ένας διάλογος εξόδου μπορεί να παρουσιάσει ένα μόνο μήνυμα έστω και αν αυτό αποτελείται από πολλές γραμμές.

Στην πλειονότητα των εφαρμογών ή των μικροεφαρμογών είναι επιθυμητό να λαμβάνουμε περισσότερες της μίας τιμής δεδομένων και να εμφανίζουμε περισσότερα από ένα σύνολα αποτελεσμάτων ταυτόχρονα. Για να το κατορθώσουμε ετούτο μπορούμε να υλοποιήσουμε γραφικές διεπαφές χρήστη οι οποίες οδηγούνται από γεγονότα δηλαδή παράγουν γεγονότα (events) όταν ο χρήστης του προγράμματος αλληλεπιδρά με αυτές. Τυπικές αλληλεπιδράσεις του χρήστη περιλαμβάνουν το πάτημα κάποιου ή κάποιων πλήκτρων, τη μετακίνηση του ποντικιού, το πάτημα ενός κουμπιού, την επιλογή ενός στοιχείου κάποιου μενού, το κλείσιμο ενός παραθύρου και γενικά κάποια αλληλεπίδραση του χρήστη με οποιοδήποτε συστατικό της γραφικής διεπαφής του.

Όταν ο χρήστης αλληλεπιδρά με ένα συστατικό μίας τέτοιας γραφικής διεπαφής χρήστη

το πρόγραμμα του ειδοποιείται ότι έχει συμβεί ένα γεγονός - event και πρέπει να καλέσει μία μέθοδο για να αντιμετωπίσει το γεγονός αυτό. Οι συναρτήσεις που καλούνται μόλις συμβαίνουν γεγονότα και τα οποία επιβάλλεται να τα αντιμετωπίσουν είναι γνωστές ως μέθοδοι χειρισμού γεγονότων (event handling methods) ή χειριστές γεγονότων (event handlers). Ετούτος ο τύπος προγραμματισμού είναι γνωστός σαν προγραμματισμός οδηγούμενος από γεγονότα ή αλλιώς ως γεγονοστρεφής προγραμματισμός (event - driven programming). Δηλαδή όταν ο χρήστης αλληλεπιδρά με ένα στοιχείο μίας γραφικής διεπαφής χρήστη, το πρόγραμμα του οδηγείται από το γεγονός αυτό, τουτέστιν ειδοποιείται ότι έχει πραγματοποιηθεί ένα γεγονός και πρέπει να το χειριστεί καταλλήλως.
Τα γεγονότα της γλώσσας Java διαιρούνται σε δύο κατηγορίες. Στα γεγονότα χαμηλού επιπέδου και στα σημασιολογικά γεγονότα. Τα γεγονότα χαμηλού επιπέδου - low level events συμπεριλαμβάνουν τα γεγονότα του πληκτρολογίου - key events και τα γεγονότα του ποντικιού - mouse events. Άλλα επιπλέον γεγονότα χαμηλού επιπέδου περιέχουν γεγονότα συστατικών - component events, γεγονότα υποδοχέων - container events, γεγονότα εστίασης - focus events και γεγονότα παραθύρων - window events. Τα σημασιολογικά γεγονότα - semantic events συγκαταλέγουν τα γεγονότα ενέργειας - action events. Επιπρόσθετα τα σημασιολογικά γεγονότα περικλείουν τα γεγονότα αντικειμένων - item events και τα γεγονότα επιλογής από λίστα - list selection events. Η αιτία που προξενεί ένα σημασιολογικό γεγονός μπορεί να διαφέρει ανάλογα με το είδος του συστατικού της γραφικής διεπαφής χρήστη με το οποίο αλληλεπιδρά ο χρήστης. Για παράδειγμα ένα αντικείμενο τύπου JTextField προκαλεί ένα γεγονός τύπου ActionEvent όταν ο χρήστης πιέσει το πλήκτρο ENTER μέσα σε αυτό ενώ ένα αντικείμενο της τάξης JButton προκαλεί ένα γεγονός του ίδιου τύπου μόλις ο χρήστης το πατήσει με το ποντίκι. Παρακάτω φαίνεται η ιεραρχία μερικών από τις κλάσεις των γεγονότων του πακέτου java.awt.event. Με έντονα γράμματα έχουν σημειωθεί οι τάξεις που θα περιγράψουμε.

- java.lang.Object
  - java.util.EventObject
    - **java.awt.AWTEvent**
      - java.awt.ComponentEvent → Γεγονότα χαμηλού επιπέδου.
        - java.awt.ContainerEvent
        - java.awt.FocusEvent
        - java.awt.PaintEvent
        - java.awt.WindowEvent
        - **java.awt.InputEvent**
          - **java.awt.KeyEvent**

- **java.awt.MouseEvent**
- java.awt.MouseWheelEvent

- **java.awt.ActionEvent** → Σημασιολογικά γεγονότα.
- java.awt.AdjustmentEvent
- java.awt.ItemEvent

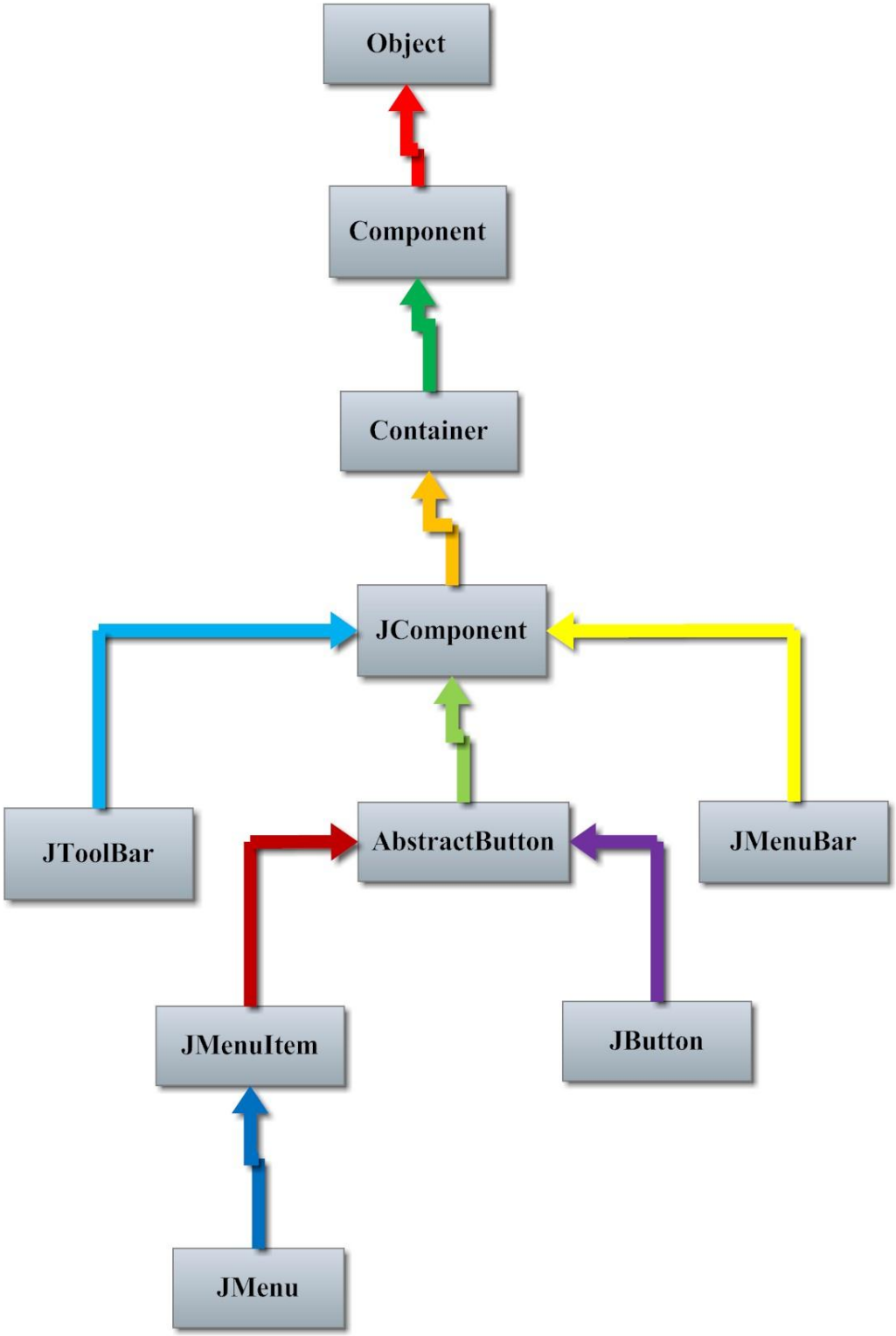
Για λόγους φορητότητας, όποτε είναι δυνατόν, επιβάλλεται να κάνουμε χρήση των σημασιολογικών γεγονότων αντί για των γεγονότων χαμηλού επιπέδου. Για παράδειγμα, ο χειρισμός γεγονότων ενέργειας κάποιου κουμπιού, αντί γεγονότων του ποντικιού, σημαίνει ότι το κουμπί μας θα ανταποκριθεί κατάλληλα στην περίπτωση που ο χρήστης προσπαθήσει να το ενεργοποιήσει με κάποιον εναλλακτικό τρόπο. Ως υπόδειγμα ο χρήστης θα είχε τη δυνατότητα να ενεργοποιήσει το κουμπί λέγοντας μία ορισμένη φράση (audio look and feel).

Για να αντιμετωπιστεί οποιοδήποτε γεγονός πρέπει να προσδιοριστεί η μέθοδος χειρισμού του γεγονότος αυτού. Έτσι ο μηχανισμός χειρισμού γεγονότων περιλαμβάνει τρία μέρη. Την πηγή του γεγονότος, το αντικείμενο του γεγονότος και τον ακροατή του γεγονότος. Η πηγή του γεγονότος (event source) είναι το συστατικό της γραφικής διεπαφής χρήστη με το οποίο ο χρήστης αλληλεπιδρά. Το αντικείμενο του γεγονότος (event object) εμπεριέχει τις πληροφορίες που αφορούν στο γεγονός το οποίο συνέβη. Οι πληροφορίες ετούτες συγκαταλέγουν μία αναφορά προς την πηγή του γεγονότος καθώς και οτιδήποτε άλλο απαιτείται από τον ακροατή γεγονότος για να χειριστεί το εν λόγω γεγονός. Ο ακροατής γεγονότος (event listener) είναι ένα αντικείμενο το οποίο ειδοποιείται όταν συμβεί ένα γεγονός. Μόλις ειδοποιηθεί, ο ακροατής γεγονότος παραλαμβάνει ένα αντικείμενο - γεγονός και το χρησιμοποιεί για να καλέσει τον κατάλληλο χειριστή για το συγκεκριμένο γεγονός.

Όταν πραγματοποιηθεί κάποιο γεγονός ειδοποιούνται μόνο οι ακροατές γεγονότων που έχουν εγγραφεί στην πηγή του γεγονότος και που αφουγκράζονται γεγονότα του ίδιου τύπου με αυτό το οποίο συνέβη. Για παράδειγμα εάν ένας ακροατής γεγονότων του ποντικιού έχει εγγραφεί με κάποιο συστατικό μίας γραφικής διεπαφής χρήστη και το στοιχείο δεν έχει ακροατές άλλων γεγονότων τότε το συστατικό αυτό μπορεί να αντιδράσει μονάχα σε γεγονότα του ποντικιού. Κάθε πηγή γεγονότος δύναται να έχει περισσότερους από έναν ακροατές γεγονότων. Ο διερμηνευτής της Java επιβάλλεται να διατηρεί μία λίστα από τους εγγεγραμμένους ακροατές κάθε πηγής γεγονότων και να τους ειδοποιεί μόλις συμβεί οποιοδήποτε γεγονός. Πρέπει επίσης να παρέχει μεθόδους που επιτρέπουν στους ακροατές μίας πηγής γεγονότων να εγγράφονται και να ακυρώνουν την εγγραφή τους με αυτήν. Ένας ακροατής γεγονότων μπορεί ακόμη να είναι εγγεγραμμένος με περισσότερες από μία πηγές γεγονότων όπως για παράδειγμα με δύο ή περισσότερα κουμπιά. Τέλος μία ενέργεια δύναται να προκαλέσει περισσότερα του ενός γεγονότα. Για παράδειγμα η πίεση του πλήκτρου k μέσα σε ένα πεδίο κειμένου προξενεί τρία γεγονότα. Ένα γεγονός πατήματος πλήκτρου, ένα γεγονός εισαγωγής χαρακτήρα και ένα γεγονός απελευθέρωσης πλήκτρου.

Όταν ένα γεγονός δημιουργείται από την αλληλεπίδραση του χρήστη με ένα συστατικό μίας γραφικής διεπαφής χρήστη τότε δίνεται στο γεγονός ετούτο ένα μοναδικό

αναγνωριστικό γεγονότος - event identifier που το προσδιορίζει μονοσήμαντα. Οι πληροφορίες οι σχετικές με ένα γεγονός μίας γραφικής διεπαφής χρήστη αποθηκεύονται σε ένα αντικείμενο μίας τάξης η οποία επεκτείνει την κλάση AWTEvent. Τα αντικείμενα της τάξης αυτής περιέχουν πληροφορίες σχετικές με το γεγονός το οποίο συνέβη καθώς και την ταυτότητα της πηγής του. Το αντικείμενο - γεγονός χρησιμοποιείται για να προσδιορίσει τον τύπο του ακροατή προς τον οποίο επιβάλλεται να διανεμηθεί το αντίστοιχο γεγονός. Ο τελευταίος με τη σειρά του καθορίζει τη μέθοδο - χειριστή που πρέπει να κληθεί.



Σχήμα 2.32 : Στην εικόνα φαίνεται ένα απόσπασμα της ιεραρχίας των βασικών κλάσεων που χρησιμοποιούνται στα Java Swings και τα Java Applets ενώ συγκαταλέγεται στην

εικόνα και η κλάση JToolBar. Η υπερκλάση είναι το κουτί της εικόνας στο οποίο πάει το βέλος και η υποκλάση είναι το κουτί της εικόνας από όπου προέρχεται το βέλος.

Ένας ακροατής γεγονότων ακούει ή αφουγκράζεται ορισμένα είδη γεγονότων τα οποία προέρχονται από πηγές γεγονότων μέσα σε ένα πρόγραμμα και καλεί την αρμόδια μέθοδο χειρισμού του κάθε γεγονότος που έχει συμβεί. Η χρήση ακροατών γεγονότων για την αντιμετώπιση αυτών είναι γνωστή ως μοντέλο ανάθεσης γεγονότων - delegation event model . Η αντιμετώπιση ενός γεγονότος ανατίθεται σε ένα ειδικό αντικείμενο τον ακροατή γεγονότος . Μόλις πραγματοποιείται ένα γεγονός, το γεγονός ετούτο διανέμεται μόνο στους ακροατές γεγονότων οι οποίοι αντιστοιχούν στον τύπο του. Η διανομή - dispatching ενός γεγονότος έχει ως αποτέλεσμα την κλήση της ανάλογης μεθόδου χειρισμού κάθε εγγεγραμμένου ακροατή γεγονότος για τον τύπο του εν λόγω γεγονότος.

Ένας ακροατής κάποιου γεγονότος γραφικής διεπαφής χρήστη είναι ένα αντικείμενο μίας κλάσης η οποία υλοποιεί μία ή περισσότερες διεπαφές ακροατών - γεγονότων από το πακέτο java.awt.event και το πακέτο javax.swing.event. Κάθε διεπαφή ακροατή γεγονότων προσδιορίζει μία ή περισσότερες μεθόδους χειρισμού γεγονότων που πρέπει να είναι ορισμένες σε μία τάξη η οποία υλοποιεί τη διεπαφή αυτή. Παρακάτω φαίνεται η ιεραρχία μερικών από τις διεπαφές ακροατών γεγονότων του πακέτου java.awt.event. Με έντονα γράμματα διακρίνονται οι διεπαφές τις οποίες θα περιγράψουμε. (Η διεπαφή ActionListener έχει ήδη περιγραφεί).

- **java.util.EventListener**
  - **java.awt.event.KeyListener**
  - **java.awt.event.MouseListener**
  - **java.awt.event.MouseMotionListener**
  - java.awt.event.MouseWheelListener
  - **java.awt.event.ActionListener**
  - java.awt.event.AdjustmentListener
  - java.awt.event.ComponentListener
  - java.awt.event.ContainerListener
  - java.awt.event.FocusListener
  - java.awt.event.ItemListener
  - java.awt.event.TextListener
  - java.awt.event.WindowListener

Κάθε τύπος γεγονότος έχει μία αντίστοιχη διεπαφή ακροατή γεγονότος. Για παράδειγμα τα γεγονότα τύπου keyEvent τυγχάνουν χειρισμού από αντικείμενα τύπου keyListener. Τα γεγονότα τύπου MouseEvent τυγχάνουν χειρισμού από αντικείμενα τύπου MousetListener και MouseMotionListener ενώ τα γεγονότα τύπου ActionEvent τυγχάνουν χειρισμού από αντικείμενα ActionListener. Στην περίπτωση ενός γεγονότος τύπου KeyEvent το γεγονός διανέμεται σε κάθε εγγεγραμμένο ακροατή τύπου KeyListener. Το αναγνωριστικό του γεγονότος τύπου keyEvent προσδιορίζει ποια μέθοδος - χειριστής των γεγονότων του πληκτρολογίου καλείται αυτόματα.

Στην περίπτωση ενός γεγονότος τύπου MouseEvent το γεγονός διανέμεται σε κάθε εγγεγραμμένο ακροατή MouseListener ή MouseMotionListener, ανάλογα με το είδος του γεγονότος το οποίο συμβαίνει. Το αναγνωριστικό του γεγονότος τύπου MouseEvent προσδιορίζει ποια μέθοδος - χειριστής (από τις επτά διαφορετικές μεθόδους χειρισμού γεγονότων του ποντικιού) καλείται αυτομάτως. Τέλος στην περίπτωση κάποιου γεγονότος τύπου ActionEvent το γεγονός διανέμεται στη μέθοδο - χειριστή actionPerformed κάθε εγγεγραμμένου ακροατή ActionListener. Η actionPerformed όπως έχει ήδη προαναφερθεί είναι η μόνη μέθοδος - χειριστής που υπάρχει στη διεπαφή ActionListener . Για να επεξεργαστεί ο προγραμματιστής κάποιο γεγονός πρέπει να κάνει τρία πράγματα:

- Να δηλώσει ότι η κλάση χειρισμού του γεγονότος ή των γεγονότων είτε υλοποιεί μία διεπαφή ακροατή γεγονότων είτε επεκτείνει (κληρονομεί) μία τάξη που υλοποιεί μία διεπαφή ακροατή γεγονότων.

- Να εγγράψει έναν ή περισσότερους ακροατές στο στοιχείο ή στα στοιχεία της γραφικής διεπαφής χρήστη τα οποία αναμένεται να δημιουργήσουν γεγονότα.

- Να υλοποιήσει μία μέθοδο ή ένα σύνολο μεθόδων χειρισμού του γεγονότος ή των γεγονότων αυτών.

Η αφηρημένη τάξη InputEvent ανήκει στο πακέτο java.awt.event και είναι υπερτάξη των κλάσεων KeyEvent και MouseEvent. Η τάξη InputEvent παρέχει τις μεθόδους isAltDown, isControlDown, isMetaDown και isShiftDown τις οποίες κληροδοτεί στις κλάσεις KeyEvent και MouseEvent. Όλες ετούτες οι συναρτήσεις είναι δημόσιες, τύπου boolean και στην περίπτωση του πληκτρολογίου επιστρέφουν την τιμή true αν το πλήκτρο Alt, Ctrl, Meta ή Shift αντίστοιχα ήταν πατημένο όταν συνέβη το γεγονός ειδάλλως επιστρέφουν την τιμή false . Η τάξη InputEvent παρέχει επίσης και την εξής μέθοδο:

- public int getModifiers()
  Ετούτη η συνάρτηση επιστρέφει τη σημαία τροποποίησης (modifiers flag) του γεγονότος που πραγματοποιήθηκε. Πιο συγκεκριμένα επιστρέφει 1 εάν το πλήκτρο Shift ήταν πατημένο μόλις συνέβη το γεγονός, 2 αν το πλήκτρο Ctrl ήταν πατημένο, 8 εάν το πλήκτρο Alt ήταν πατημένο, 3 αν τα πλήκτρα Ctrl και Shift ήταν πατημένα κ.τ.λ.π.

Τα γεγονότα του πληκτρολογίου δημιουργούνται όταν πλήκτρα αυτού πατιούνται ή αφήνονται. Ένα γεγονός τύπου KeyEvent είναι χαμηλού επιπέδου και παράγεται από κάποιο συστατικό (όπως για παράδειγμα ένα αντικείμενο της τάξης JTextField), όταν μέσα στα πλαίσια του στοιχείου αυτού ένα πλήκτρο πιέζεται ή αφήνεται ή όταν ένας χαρακτήρας του κώδικα UNICODE εισάγεται. Ένα τέτοιο γεγονός περνιέται σε όλα τα αντικείμενα τύπου KeyListener ή KeyAdapter (που υλοποιεί τη διεπαφή KeyListener) τα οποία είναι εγγεγραμμένα για να δέχονται γεγονότα αυτού του τύπου με χρήση της μεθόδου addKeyListener του συστατικού. Με τον τρόπο ετούτο κάθε τέτοιο αντικείμενο ακροατής λαμβάνει το γεγονός, όταν αυτό συμβαίνει. Η κλάση KeyEvent παρέχει μία ακέραιη σταθερά, τον κώδικα εικονικού πλήκτρου - virtual key code, για να αναπαραστήσει το κάθε πλήκτρο του πληκτρολογίου. Τα αναγνωριστικά όλων αυτών των σταθερών αρχίζουν με τους χαρακτήρες VK_, όπως για παράδειγμα VK_J ή VK_HOME. Ορισμένες από τις συναρτήσεις της τάξης KeyEvent είναι οι επόμενες:

- public int getKeyCode()
  Αυτή η μέθοδος για ένα γεγονός πίεσης ή απελευθέρωσης πλήκτρου επιστρέφει τον ακέραιο κωδικό του πλήκτρου αυτού ενώ για ένα γεγονός εμφάνισης χαρακτήρα επιστρέφει την τιμή VK_UNDEFINED.

- public char getKeyChar()
  Ετούτη η μέθοδος επιστρέφει το χαρακτήρα του κώδικα UNICODE ο οποίος αντιστοιχεί στο πλήκτρο που δημιούργησε το εν λόγω γεγονός ή την τιμή CHAR_UNDEFINED στην περίπτωση όπου δεν υπάρχει αντίστοιχος έγκυρος χαρακτήρας. Για παράδειγμα η πληκτρολόγηση του πλήκτρου a έχοντας πατημένο το πλήκτρο Shift επιστρέφει την τιμή A.

- public static String getKeyText(int a)
  Αυτή η συνάρτηση επιστρέφει μία συμβολοσειρά η οποία περιγράφει τον κώδικα a του πλήκτρου που προκάλεσε το γεγονός. Για παράδειγμα οι συμβολοσειρές θα μπορούσαν να είναι End, F7 ή C.

- public static String getKeyModifiersText(int a)
  Ετούτη η συνάρτηση επιστρέφει μία συμβολοσειρά η οποία περιγράφει τη σημαία τροποποίησης ανάλογα με την τιμή του a. Για παράδειγμα για a=1 επιστρέφεται η συμβολοσειρά Shift ενώ για a=3 επιστρέφεται η συμβολοσειρά Ctrl+Shift.

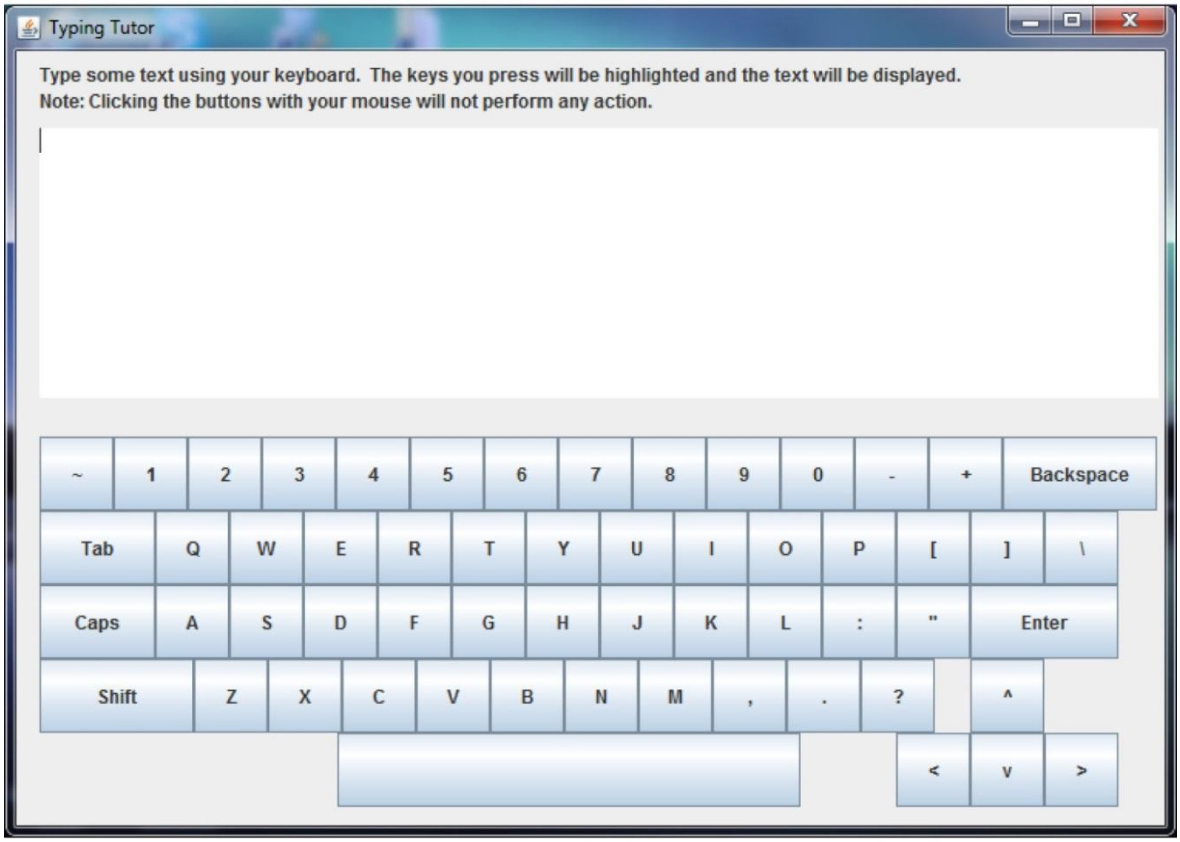
- public boolean isActionKey()
  Αυτή η μέθοδος επιστρέφει την τιμή true εάν το πλήκτρο που προξένησε το γεγονός είναι πλήκτρο ενέργειας - action key. Για παράδειγμα αν το πλήκτρο είναι βελάκι, Home, End, Page Up, Page Down, F1 - F12, Num Lock, Scroll Lock, Caps Lock και Pause. Ειδάλλως επιστρέφει την τιμή false.

Κάθε τάξη η οποία επιθυμεί να επεξεργαστεί γεγονότα του πληκτρολογίου πρέπει είτε να υλοποιεί τη διεπαφή KeyListener (και όλες τις μεθόδους της) είτε να επεκτείνει την αφηρημένη κλάση KeyAdapter. Το αντικείμενο - ακροατής που δημιουργείται από μία τέτοια τάξη εγγράφεται ακολούθως με ένα συστατικό με τη χρήση της μεθόδου addKeyListener του συστατικού ετούτου. Για κάθε γεγονός του πληκτρολογίου καλείται η ανάλογη μέθοδος - χειριστής της διεπαφής αυτής. Η διεπαφή KeyListener έχει τις παρακάτω συναρτήσεις, κάθε μία εκ των οποίων έχει ως παράμετρο ένα αντικείμενο τύπου KeyEvent:

- public void keyPressed(KeyEvent a)
  Αυτή η μέθοδος καλείται όταν πατηθεί κάποιο πλήκτρο του πληκτρολογίου.

- public void keyTyped(KeyEvent a)
  Ετούτη η μέθοδος καλείται μόλις πατηθεί ένα πλήκτρο του πληκτρολογίου που εμφανίζει κάποιο χαρακτήρα. Για παράδειγμα το πλήκτρο j αλλά όχι το πλήκτρο Shift ή το πλήκτρο F1.

- public void keyReleased(KeyEvent a)
  Αυτή η συνάρτηση καλείται όταν αφεθεί κάποιο πλήκτρο μετά από ένα γεγονός πίεσης ή εμφάνισης ενός χαρακτήρα.



Σχήμα 2.33 : Στην εικόνα φαίνεται ένα παράθυρο το οποίο έχει σχηματιστεί από την κλάση JFrame. Ετούτο το παράθυρο είναι ένα εικονικό πληκτρολόγιο. Όταν ο χρήστης πατάει πλήκτρα τότε παράγεται κείμενο στο επάνω μέρος της εικόνας όπως γίνεται και με το πληκτρολόγιο του υπολογιστή. Αν επιλεχθεί κάποιο πλήκτρο με το ποντίκι τότε δεν εμφανίζεται τίποτα στο επάνω μέρος του παραθύρου.
Τα γεγονότα του ποντικιού δύνανται να συμβούν σε οποιοδήποτε συστατικό γραφικής διεπαφής χρήστη το οποίο προέρχεται από την κλάση java.awt.Component εάν ο δρομέας του ποντικιού βρίσκεται μέσα στα φανερά όρια που έχει το συστατικό αυτό πάνω στην οθόνη. Τα γεγονότα του ποντικιού είναι χαμηλού επιπέδου ενώ είναι αντικείμενα της τάξης MouseEvent καθώς και της κλάσης MouseWheelEvent. Ωστόσο θα περιγράψουμε μόνο την κλάση MouseEvent. Ένα γεγονός του ποντικιού - mouse event παράγεται από κάποιο συστατικό όταν:

- Πιέζεται ένα κουμπί του ποντικιού.
- Αφήνεται ένα κουμπί του ποντικιού.
- Πιέζεται και αφήνεται ένα κουμπί του ποντικιού.
- Ο δείκτης του ποντικιού μπαίνει στο φανερό μέρος της γεωμετρίας ενός στοιχείου.
- Ο δείκτης του ποντικιού βγαίνει από το φανερό μέρος της γεωμετρίας ενός συστατικού.

Ένα γεγονός κίνησης του ποντικιού - mouse motion event παράγεται από κάποιο συστατικό όταν:

- Το ποντίκι κινείται.
- Το ποντίκι σέρνεται.

Ένα γεγονός της πρώτης κατηγορίας περνιέται σε όλα τα αντικείμενα τύπου MouseListener ή MouseAdapter (τα οποία υλοποιούν τη διεπαφή MouseListener) που είναι εγγεγραμμένα σε κάποιο συστατικό για να δέχονται τα γεγονότα του τύπου αυτού με χρήση της μεθόδου addMouseListener του στοιχείου. Ένα γεγονός της δεύτερης κατηγορίας περνιέται σε όλα τα αντικείμενα τύπου MouseMotionListener ή MouseMotionAdapter (τα οποία υλοποιούν τη διεπαφή MouseMotionListener) που είναι εγγεγραμμένα για να δέχονται γεγονότα του τύπου αυτού με χρήση της μεθόδου addMouseListener του συστατικού. Τοιουτοτρόπως κάθε αντικείμενο ακροατής λαμβάνει το γεγονός μόλις ετούτο συμβαίνει. Κάθε μέθοδος χειρισμού ενός γεγονότος του ποντικιού παίρνει ως παράμετρο ένα αντικείμενο τύπου MouseEvent το οποίο εμπεριέχει πληροφορίες σχετικές με το είδος του γεγονότος που πραγματοποιήθηκε. Η τάξη MouseEvent κληρονομεί από την κλάση InputEvent μεταξύ άλλων και τις παρακάτω συναρτήσεις:

- public boolean isAltDown()
  Αυτή η μέθοδος επιστρέφει true όταν ο χρήστης πατήσει το μεσαίο πλήκτρο ενός ποντικιού με τρία πλήκτρα. Για να το πετύχει ετούτο με ένα ποντίκι που έχει μόνο ένα ή δύο πλήκτρα, ο χρήστης μπορεί να πιέσει το πλήκτρο Alt του πληκτρολογίου και ταυτοχρόνως το αριστερό πλήκτρο του ποντικιού.

- public boolean isMetaDown()
  Ετούτη η μέθοδος επιστρέφει true στην περίπτωση που ο χρήστης πατήσει το δεξιό πλήκτρο ενός ποντικιού με τρία πλήκτρα. Για να το πετύχει αυτό με ένα ποντίκι το οποίο έχει μόνο ένα πλήκτρο, ο χρήστης μπορεί να πιέσει το πλήκτρο Meta του πληκτρολογίου και συγχρόνως να πιέσει το πλήκτρο του ποντικιού.

Μερικές από τις συναρτήσεις της τάξης MouseEvent είναι οι εξής:

- pubic int getButton()
  Αυτή η συνάρτηση επιστρέφει μία από τις στατικές σταθερές τύπου int NOBUTTON, BUTTON1, BUTTON2, BUTTON3 οι οποίες σταθερές είναι δηλωμένες στην κλάση MouseEvent, ανάλογα με το αν και με ποιο κουμπί του ποντικιού άλλαξε κατάσταση.

- public int getClickCount()
  Ετούτη η μέθοδος επιστρέφει το πλήθος των φορών που πατήθηκε και αφέθηκε το κουμπί του ποντικιού σε αυτό το γεγονός. Για να επιστρέψει η συνάρτηση ακέραιο μεγαλύτερο του 1 πρέπει η χρονική απόσταση μεταξύ των διαδοχικών φορών να είναι σχετικά μικρή.

- public int getX()
  Αυτή η μέθοδος επιστρέφει την οριζόντια συντεταγμένη x της θέσης του ποντικιού σχετικά με την αρχή του συστατικού.

- public int getY()
  Ετούτη η συνάρτηση επιστρέφει την κατακόρυφη συντεταγμένη y της θέσης του ποντικιού σχετικά με την αρχή του στοιχείου.

Κάθε τάξη η οποία επιθυμεί να επεξεργαστεί γεγονότα του ποντικιού (γεγονότα πρώτης κατηγορίας) επιβάλλεται είτε να υλοποιεί τη διεπαφή MouseListener και όλες τις μεθόδους της είτε να επεκτείνει την αφηρημένη κλάση MouseAdapter. Το αντικείμενο - ακροατής το οποίο δημιουργείται από μία τέτοια τάξη εγγράφεται ακολούθως με ένα συστατικό με τη χρήση της συνάρτησης addMouseListener του στοιχείου ετούτου. Για κάθε γεγονός τέτοιου είδους γίνεται κλήση της ανάλογης

μεθόδου - χειρισμού της διεπαφής αυτής. Η διεπαφή MouseListener έχει τις παρακάτω συναρτήσεις, κάθε μία εκ των οποίων έχει ως παράμετρο ένα αντικείμενο τύπου MouseEvent:

- public void mousePressed(MouseEvent a)
  Αυτή η συνάρτηση καλείται μόλις πατηθεί ένα πλήκτρο του ποντικιού με το δείκτη του ποντικιού να βρίσκεται πάνω στο συστατικό.

- public void mouseClicked(MouseEvent a)
  Ετούτη η συνάρτηση καλείται όταν πατηθεί και αφεθεί ελεύθερο ένα πλήκτρο του ποντικιού πάνω στο στοιχείο δίχως να έχει κινηθεί ο δείκτης του ποντικιού.

- public void mouseReleased(MouseEvent a)
  Αυτή η μέθοδος καλείται μόλις αφήνεται ένα κουμπί του ποντικιού ενώ ο δείκτης του ποντικιού βρίσκεται επάνω στο συστατικό. Το γεγονός αυτό ακολουθεί πάντα ένα γεγονός που συμβαίνει όταν πατηθεί κάποιο πλήκτρο του ποντικιού είτε έχει κινηθεί ο δείκτης του ποντικιού είτε όχι.

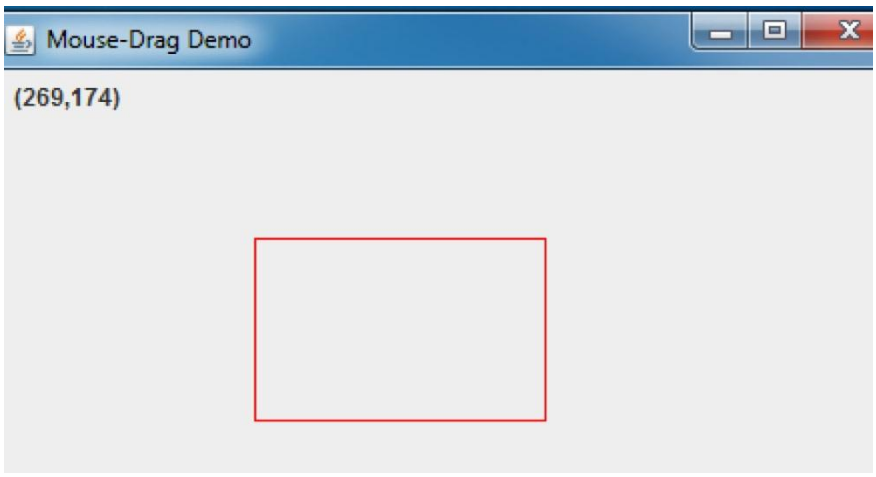
- public void mouseEntered(MouseEvent a)
  Ετούτη η μέθοδος καλείται όταν ο δρομέας του ποντικιού μπαίνει στα φυσικά όρια του στοιχείου.

- public void mouseExited(MouseEvent a)
  Αυτή η συνάρτηση καλείται μόλις ο δρομέας του ποντικιού αφήνει δηλαδή βγαίνει από τα φυσικά όρια του συστατικού. Εδώ πρέπει να επισημανθεί πως το συστατικό που αναφέρεται σε όλες ετούτες τις μεθόδους είναι εκείνο στο οποίο ο συγκεκριμένος ακροατής είναι εγγεγραμμένος.

Κάθε τάξη που επιθυμεί να επεξεργαστεί γεγονότα κίνησης του ποντικιού (γεγονότα δεύτερης κατηγορίας) πρέπει είτε να υλοποιεί τη διεπαφή MouseMotionListener και όλες τις μεθόδους της είτε να επεκτείνει την αφηρημένη κλάση MouseMotionAdapter. Το αντικείμενο - ακροατής το οποίο δημιουργείται από μία τέτοια τάξη εγγράφεται ακολούθως με ένα στοιχείο με χρήση της συνάρτησης addMouseMotionListener του συστατικού ετούτου. Για κάθε γεγονός τέτοιου είδους καλείται η ανάλογη μέθοδος της διεπαφής αυτής. Η διεπαφή MouseMotionListener έχει τις επόμενες μεθόδους, κάθε μία εκ των οποίων έχει ως παράμετρο ένα αντικείμενο τύπου MouseEvent:

- public void mouseMoved(MouseEvent a)
  Αυτή η συνάρτηση καλείται μόλις το ποντίκι κινηθεί με το δείκτη του να βρίσκεται πάνω στο στοιχείο και χωρίς να πατηθεί οποιοδήποτε από τα κουμπιά του.

- public void mouseDragged(MouseEvent a)
  Αυτή η μέθοδος καλείται όταν πατιέται το αριστερό κουμπί του ποντικιού ενώ ο δείκτης του βρίσκεται πάνω στο συστατικό και στη συνέχεια ενώ κρατιέται πατημένο το ποντίκι κινείται. Η διαδικασία ετούτη είναι γνωστή και ως σύρσιμο (dragging) του ποντικιού. Τέτοια γεγονότα παραδίδονται επαναληπτικά στο στοιχείο από το οποίο ξεκίνησε το γεγονός μέχρι τη στιγμή που θα αφεθεί το κουμπί του ποντικιού το οποίο ήταν πατημένο άσχετα με το αν αυτό θα συμβεί εντός ή εκτός των ορίων του συστατικού.



Σχήμα 2.34 : Στην εικόνα φαίνεται ένα παράθυρο το οποίο έχει σχηματιστεί από την κλάση JFrame. Ο χρήστης με το ποντίκι δύναται εύκολα να δημιουργήσει ένα ορθογώνιο παραλληλόγραμμο απλά βάζοντας το ποντίκι πάνω στο παράθυρο και πιέζοντας το αριστερό κουμπί του ποντικιού. Τότε παράγεται το ορθογώνιο παραλληλόγραμμο και το ποντίκι σέρνει τη μία από τις τέσσερις γωνίες του παραλληλογράμμου. Όσο σέρνεται η εν λόγω γωνία τόσο μεταβάλλονται και οι διαστάσεις του ορθογωνίου. Στο πάνω αριστερά μέρος του παραθύρου διακρίνονται οι συντεταγμένες ετούτης της γωνίας που επίσης μεταβάλλονται όσο σέρνεται δηλαδή αλλάζει θέση η γωνία.

Πολλές από τις διεπαφές ακροατών γεγονότων όπως για παράδειγμα οι MouseListener και MouseMotionListener εμπεριέχουν περισσότερες από μία μεθόδους. Είναι φανερό πως δεν είναι πάντα επιθυμητό να ορίζουμε όλες τις μεθόδους που περιέχει μία διεπαφή ακροατών γεγονότων. Για παράδειγμα ένα πρόγραμμα μπορεί να χρειάζεται να χρησιμοποιήσει μόνο τη συνάρτηση mouseClicked από τη διεπαφή MouseListener ή τη μέθοδο mouseDragged από τη διεπαφή MouseMotionListener. Στην περίπτωση ετούτη θα πρέπει να υλοποιήσει στοιχειωδώς δηλαδή απλά γράφοντας την επικεφαλίδα και ένα κενό σώμα για την κάθε μία καθώς και τις υπόλοιπες συναρτήσεις της αντίστοιχης διεπαφής πράγμα το οποίο είναι, αν μη τι άλλο, χρονοβόρο. Για το λόγο αυτό τα πακέτα java.awt.event και javax.swing.event παρέχουν τάξεις προσαρμογείς ακροατών γεγονότων.

Μία τάξη προσαρμογέας ή προσαρμοστική κλάση (adapter class) υλοποιεί μία διεπαφή και παρέχει μία εξ ορισμού υλοποίηση τουτέστιν ένα κενό σώμα για την κάθε συνάρτηση μίας διεπαφής. Ο προγραμματιστής δύναται να επεκτείνει μία προσαρμοστική τάξη για να κληρονομήσει την εξ ορισμού δηλαδή με κενό σώμα υλοποίηση της κάθε μεθόδου και να χρησιμοποιήσει μόνο τη συνάρτηση ή τις συναρτήσεις τις οποίες χρειάζεται για το χειρισμό γεγονότων. Οι κλάσεις προσαρμογείς που παρέχει το πακέτο java.awt.event καθώς και οι διεπαφές τις οποίες υλοποιούν διακρίνονται παρακάτω. Με έντονα γράμματα έχουν σημειωθεί οι τάξεις και οι διεπαφές που μας ενδιαφέρουν περισσότερο.

Τάξη προσαρμογέας γεγονότων	**Υλοποιούμενη διεπαφή**
ComponentAdapter	ComponentListener
ContainerAdapter	ContainerListener
FocusAdapter	FocusListener
KeyAdapter	**KeyListener**
MouseAdapter	**MouseListener**
MouseMotionAdapter	**MouseMotionListener**
WindowAdapter	WindowListener

Μία κλάση η οποία υλοποιεί μία διεπαφή διαθέτει μία σχέση IS-A με την εν λόγω διεπαφή. Όλες οι άμεσες ή έμμεσες υποτάξεις της κλάσης ετούτης κληρονομούν τη σχέση αυτή. Τοιουτοτρόπως ένα αντικείμενο μίας τάξης το οποίο κληρονομεί μία κλάση προσαρμογέα γεγονότων αποτελεί ένα αντικείμενο του αντίστοιχου τύπου ακροατή γεγονότων. Για παράδειγμα ένα αντικείμενο μίας υποκλάσης της τάξης MouseAdapter είναι ένα αντικείμενο τύπου MouseListener. Για να χρησιμοποιήσουμε μία κλάση - προσαρμογέα επιβάλλεται να δημιουργήσουμε μία υποτάξη της αντί να υλοποιήσουμε άμεσα μία διεπαφή ακροατή γεγονότων. Εδώ τίθεται το ερώτημα τι μπορούμε να κάνουμε στην περίπτωση που επιθυμούμε να επεκτείνουμε και κάποια άλλη τάξη εκτός της κλάσης του προσαρμογέα. Για παράδειγμα ας υποθέσουμε ότι γράφουμε μία εφαρμογή η οποία επεκτείνει την τάξη JFrame ή μία μικροεφαρμογή που επεκτείνει την κλάση JApplet. Έστω ότι επιθυμούμε η εφαρμογή ή η μικροεφαρμογή μας να περιέχει εντολές οι οποίες χειρίζονται γεγονότα του ποντικιού. Εφόσον η γλώσσα Java δεν επιτρέπει πολλαπλή κληρονομικότητα η τάξη μας δε μπορεί συγχρόνως να επεκτείνει την κλάση JFrame ή JApplet και την τάξη MouseAdapter. Η λύση είναι να χρησιμοποιήσουμε εντός της κλάσης JFrame ή JApplet μία εσωτερική τάξη η οποία να κληρονομεί την κλάση MouseAdapter. Η χρήση ανώνυμων εσωτερικών τάξεων είναι καλύτερη από τη χρήση των επωνύμων καθόσον μας επιτρέπει να γράφουμε την υλοποίηση των ακροατών των γεγονότων μας στη γειτονία των εντολών που εγγράφουν αυτούς στα συστατικά των προγραμμάτων μας, ιδιαίτερα αν οι υλοποιήσεις ετούτες περιέχουν λίγες εντολές. Η χρήση των εσωτερικών κλάσεων επιπλέον λύνει το πρόβλημα το οποίο προκύπτει στην περίπτωση όπου οι χειριστές των γεγονότων μας χρειάζονται να προσπελάσουν ιδιωτικές υποστασιακές μεταβλητές της περιβάλλουσας τουτέστιν της εξωτερικής τάξης. Εάν δε δηλώσουμε την εσωτερική μας κλάση ως τύπου static τότε αυτή μπορεί να αναφέρεται στις υποστασιακές μεταβλητές της περιβάλλουσας τάξης σαν οι εντολές της να βρίσκονταν μέσα στην τελευταία.

Ένα σημασιολογικό γεγονός σηματοδοτεί ότι έχει πραγματοποιηθεί μία ενέργεια πάνω σε ένα συστατικό μίας γραφικής διεπαφής χρήστη. Τα σημασιολογικά γεγονότα δημιουργούνται από στοιχεία όπως για παράδειγμα αντικείμενα τύπου JButton μόλις κάποια σχετική με αυτά ενέργεια λάβει χώρα, ως υπόδειγμα ότι πατιούνται. Ένα τέτοιο γεγονός περνιέται σε όλα τα αντικείμενα τύπου ActionListener τα οποία είναι εγγεγραμμένα για να δέχονται γεγονότα ετούτου του τύπου με χρήση της μεθόδου addActionListener του συστατικού. Τοιουτοτρόπως κάθε τέτοιο αντικείμενο ακροατής λαμβάνει το γεγονός όταν αυτό συμβαίνει και δεν απαιτείται να γνωρίζει τις λεπτομέρειες επεξεργασίας των γεγονότων του ποντικιού. Έτσι δύναται να επεξεργαστεί γεγονότα τα οποία έχουν σημασία (σημασιολογικά) όπως το πάτημα του κουμπιού. Τα σημασιολογικά γεγονότα είναι αντικείμενα της κλάσης ActionEvent. Η τάξη ετούτη κληρονομεί από την υπερκλάση της τη java.util.EventObject την ακόλουθη συνάρτηση:

- public Object getSource()
  Αυτή η μέθοδος επιστρέφει το αντικείμενο στο οποίο δημιουργήθηκε αρχικά το γεγονός.

Σχήμα 2.35 : Στην εικόνα φαίνεται ένα παράθυρο το οποίο έχει σχηματιστεί από την κλάση JFrame. Μόλις ο χρήστης πατήσει το κόκκινο κουμπί όλο το παράθυρο πλην του χώρου που καταλαμβάνουν τα τρία κουμπιά παίρνει κόκκινο χρώμα. Όταν ο χρήστης πατήσει το πράσινο κουμπί ο ίδιος χώρος γεμίζει με πράσινο χρώμα ενώ μόλις πατήσει το μπλε κουμπί γεμίζει με μπλε χρώμα. Αυτό είναι ένα παράδειγμα μίας τάξης η οποία χειρίζεται σημασιολογικά γεγονότα υλοποιώντας τη διεπαφή ActionListener και τη μοναδική της μέθοδο actionPerformed.

Σε αντίθεση με τον έλεγχο του προγράμματος μίας εφαρμογής που μεταφέρεται αρχικά στη μέθοδο main αυτής, ο έλεγχος του προγράμματος μίας μικροεφαρμογής μεταφέρεται σε αυτήν ή την εγκαταλείπει αυτόματα μέσω των παρακάτω συναρτήσεων. Ετούτες οι μέθοδοι έχουν οριστεί στη Διεπαφή Προγραμματισμού Εφαρμογών (Applications Programming Interface, API) της Java, με τέτοιο τρόπο ώστε να μην κάνουν τίποτα. Έτσι αν ένας προγραμματιστής επιθυμεί να τις χρησιμοποιήσει με διαφορετικό τρόπο πρέπει να δημιουργήσει τις δικές του εκδόσεις. Οι συναρτήσεις είναι οι επόμενες:

- public void init()
  Ετούτη η μέθοδος καλείται αυτομάτως κάθε φορά που φορτώνεται η μικροεφαρμογή από το φυλλομετρητή για εκτέλεση. Σκοπός της είναι να αρχικοποιήσει τη μικροεφαρμογή. Η συνάρτηση αυτή μπορεί να χρησιμοποιηθεί για να δώσει αρχικές τιμές σε τελικές μεταβλητές, δομές δεδομένων τις οποίες θα χρειαστεί η μέθοδος paint, μεταβλητές υπόστασης των συστατικών γραφικής διεπαφής χρήστη της μικροεφαρμογής, να φορτώσει ήχους για παίξιμο ή εικόνες για εμφάνιση, να δημιουργήσει νήματα ελέγχου και άλλα.

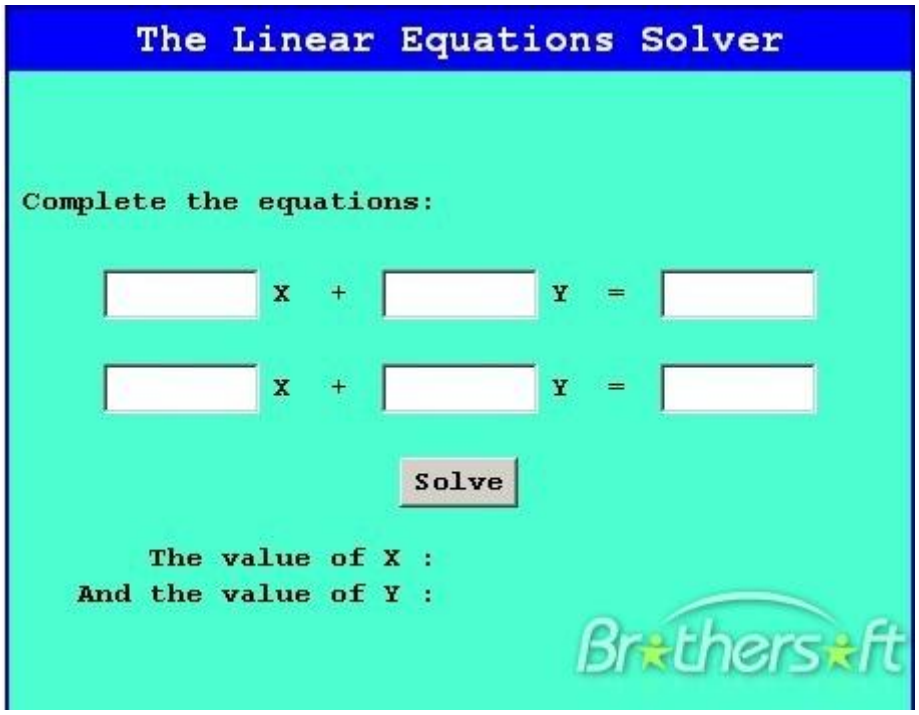
- public void start()
  Αυτή η μέθοδος καλείται μετά τον τερματισμό της εκτέλεσης της συνάρτησης init καθώς και κάθε φορά που ο χρήστης του φυλλομετρητή επιστρέφει στο παράθυρο της μικροεφαρμογής. Η μέθοδος ετούτη μπορεί να χρησιμοποιηθεί για να εκτελέσει όλες τις ενέργειες οι οποίες απαιτούνται να γίνουν στις περιπτώσεις αυτές. Τυπικές τέτοιες ενέργειες συμπεριλαμβάνουν το ξεκίνημα κινούμενων γραφικών (animation), το ξεκίνημα άλλων νημάτων και άλλα.

- public void stop()
  Ετούτη η συνάρτηση καλείται αυτομάτως όταν η μικροεφαρμογή πρέπει να σταματήσει την εκτέλεση της, συνήθως μόλις ο φυλλομετρητής

εγκαταλείπει το παράθυρο της μικροεφαρμογής. Χρησιμοποιείται για να αντιστρέψει τις ενέργειες τις οποίες έκανε η μέθοδος start. Η συνάρτηση αυτή μπορεί να διατεθεί με στόχο να εκτελέσει τις ενέργειες που απαιτούνται για την αναστολή της εκτέλεσης της μικροεφαρμογής. Τέτοιες ενέργειες συγκαταλέγουν τον τερματισμό κινούμενων γραφικών, νημάτων και άλλα.

- public void destroy()
  Ετούτη η μέθοδος καλείται αυτόματα μόλις διεκπεραιώνεται η εκτέλεση της μικροεφαρμογής, δηλαδή όταν ο χρήστης του φυλλομετρητή εγκαταλείπει τη σύνοδο με αυτόν και ξεφορτώνεται η μικροεφαρμογή από τη μνήμη. Χρησιμοποιείται συνήθως για να αντιστρέψει όσες ενέργειες έκανε η init, τουτέστιν για να απελευθερώσει τους πόρους του συστήματος οι οποίοι είχαν δεσμευτεί για την εκτέλεση της μικροεφαρμογής.

Έχει προαναφερθεί πως σε μία μικροεφαρμογή μπορούμε να ενσωματώσουμε μία εφαρμογή. Ομοίως και σε μία εφαρμογή έχουμε τη δυνατότητα να κάνουμε προσάρτηση μίας μικροεφαρμογής. Είναι πολλές φορές επιθυμητό να σχεδιάζουμε ένα πρόγραμμα της Java το οποίο να δύναται να εκτελείται ως εφαρμογή που να τρέχει στο δικό μας υπολογιστή και ως μικροεφαρμογή διαθέσιμη να κατεβαίνει μέσω του ιστού σε άλλους υπολογιστές - πελάτες. Αυτό μπορεί να επιτευχθεί στην περίπτωση που η μικροεφαρμογή μας περιέχει εκτός των συνηθισμένων μεθόδων της και τη συνάρτηση main. Όταν το πρόγραμμα εκτελείται ως μικροεφαρμογή η μέθοδος main δεν εκτελείται καθόλου. Αντιθέτως εάν το πρόγραμμα εκτελείται ως εφαρμογή ο έλεγχος του προγράμματος μεταφέρεται αυτόματα στη μέθοδο αυτή από την εικονική μηχανή της Java. Η συνάρτηση main δημιουργεί τότε ένα αντικείμενο - μικροεφαρμογή. Στη συνέχεια κατασκευάζει ένα πλαίσιο στον υαλοπίνακα του οποίου προσθέτει το αντικείμενο ετούτο. Η main επιβάλλεται να καλεί άμεσα τη συνάρτηση init της μικροεφαρμογής και τη μέθοδο start αν χρειάζεται. Η main μπορεί να αποφύγει την κλήση της init εάν η τάξη παρέχει μία μέθοδο κατασκευαστή η οποία να κάνει ακριβώς τις ίδιες ενέργειες με την init.

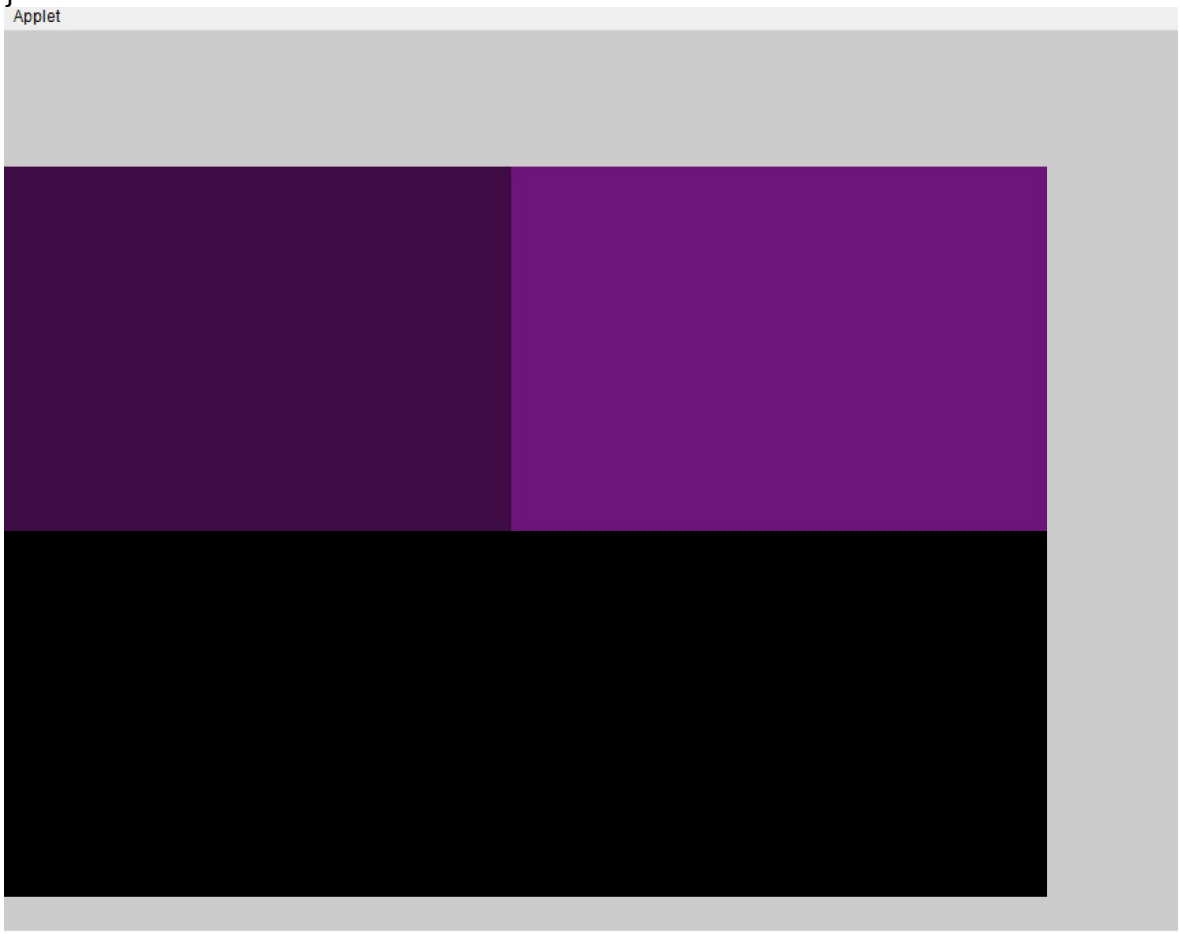


Σχήμα 2.36 : Στην εικόνα διακρίνεται ένα Applet το οποίο επιλύει ένα σύστημα γραμμικών εξισώσεων με δύο άγνωστες μεταβλητές.

Έχει προαναφερθεί πως τα μηχανολογικά σχέδια στα Applets ετούτης της διπλωματικής εργασίας δεν είναι μόνο εικόνες αλλά και τρισδιάστατα σχέδια. Τα τρισδιάστατα σχέδια βρίσκονται σε αρχεία με τη μορφή της γλώσσας VRML (Virtual Reality Modeling Language) δηλαδή έχουν κατάληξη .wrl. Τουτέστιν σε αυτήν την εργασία πραγματοποιήθηκε το ανάποδο από ότι γίνεται στα τρισδιάστατα γραφικά. Στη σχεδίαση τρισδιάστατων γραφικών με τη γλώσσα VRML, χρησιμοποιείται η γλώσσα Java ή η γλώσσα JavaScript για να δημιουργήσουν συμπεριφορές στα γραφικά. Εδώ χρησιμοποιούμε μέσα στο σχεδιασμό προγράμματος με τη γλώσσα Java, αρχεία της γλώσσας VRML ώστε να παρουσιάσουμε τα μηχανολογικά σχέδια σε τρισδιάστατη μορφή. Για να φορτώσουμε ένα αρχείο της γλώσσας VRML στο Eclipse, πρέπει να γράψουμε τον εξής κώδικα:

```
public class test_4 extends JApplet
{
private Color xroma;
public void init ()
{
Container john=getContentPane();
john.setLayout (null);
xroma=new Color(204,204,204);
john.setBackground(xroma);
Canvas3D c = new Canvas3D (SimpleUniverse.getPreferredConfiguration ());
c.setBounds(0,100,765,535);
john.add(c);
BranchGroup scene = createSceneGraph ();
SimpleUniverse u = new SimpleUniverse (c);
u.getViewingPlatform (). setNominalViewingTransform ();
u.addBranchGraph (scene);
}

public BranchGroup createSceneGraph ()
{

BranchGroup objRoot = new BranchGroup ();
TransformGroup objScale = new TransformGroup ();
Transform3D t3d = new Transform3D ();
t3d.setScale (0.3);
objScale.setTransform (t3d);
objRoot.addChild (objScale);
Color3f light1Color = new Color3f (1.0f, 1.0f, 0.9f);
Vector3f light1Direction = new Vector3f (4.0f,-7.0f,-12.0f);
BoundingSphere bounds =     new BoundingSphere (new Point3d (0.0,0.0,0.0), 100.0);
DirectionalLight light1 = new DirectionalLight (light1Color, light1Direction);
light1.setInfluencingBounds (bounds);
objRoot.addChild (light1);
TransformGroup objTrans = new TransformGroup ();
objTrans.setCapability (TransformGroup.ALLOW_TRANSFORM_WRITE);
objScale.addChild (objTrans);

VrmlLoader loader = new VrmlLoader ();
Scene s = null;

try
{
s = loader.load ("www.WRL");
}
catch (Exception e)
```

```
{
}
objTrans.addChild (s.getSceneGroup ());
Transform3D yAxis = new Transform3D ();
Alpha rotationAlpha = new Alpha (-1, Alpha.INCREASING_ENABLE,0,0,4000,0,0,0,0,0);
RotationInterpolator rotator = new RotationInterpolator
(rotationAlpha,objTrans,yAxis,0.0f,(float)Math.PI *2.0f);
rotator.setSchedulingBounds (bounds);
objTrans.addChild (rotator);
objRoot.compile ();
return objRoot;
}
}
```



Σχήμα 2.37 : Στην εικόνα φαίνεται το applet που δημιουργείται με βάση τον κώδικα που αναφέρθηκε προηγουμένως. Το αρχείο VRML έχει φορτωθεί στο applet και το τρισδιάστατο σχέδιο γυρίζει συνεχώς οριζόντια γύρω από τον εαυτό του με φορά αντίθετη από αυτήν του ρολογιού.

Η κλάση Canvas3D αποτελεί μία κλάση η οποία ανήκει στο πακέτο javax.media.j3d. Δημιουργεί ένα πανό μέσα στο οποίο μπορούν να σχηματιστούν τρισδιάστατα γραφικά. Η τάξη BranchGroup ανήκει ομοίως στο πακέτο javax.media.j3d. Λειτουργεί ως δείκτης στη ρίζα του γραφήματος της σκηνής διακλάδωσης. Η κλάση SimpleUniverse ανήκει στο πακέτο com.sun.j3d.utils.universe και δημιουργεί όλα τα απαραίτητα αντικείμενα της ορατής πλευράς του γραφήματος της σκηνής. Πιο συγκεκριμένα διαμορφώνει μία τοπική πλατφόρμα παρακολούθησης και ένα αντικείμενο που παρακολουθεί. Μερικοί κατασκευαστές των τριών αυτών κλάσεων είναι οι εξής:

- public Canvas3D (java.awt.GraphicsConfiguration graphicsConfiguration)

- public BranchGroup ()
- public SimpleUniverse (Canvas3D canvas)

Ορισμένες μέθοδοι των κλάσεων BranchGroup και SimpleUniverse είναι οι επόμενες:

- public ViewingPlatform getViewingPlatform ()
  Ετούτη η συνάρτηση της κλάσης SimpleUniverse επιστρέφει ένα αντικείμενο της τάξης ViewingPlatform το οποίο συσχετίζεται με το γράφημα της σκηνής.
- public void addBranchGraph (BranchGroup a)
  Αυτή η συνάρτηση της κλάσης SimpleUniverse προσθέτει κόμβους στην πλευρά γεωμετρίας του γραφήματος της σκηνής.
- public void addChild (Node child)
  Ετούτη η μέθοδος της τάξης BranchGroup προσθέτει το αντικείμενο της κλάσης Node ή το αντικείμενο της υποκλάσης αυτής στη λίστα των παιδιών της ομάδας των κόμβων.
- public void compile ()
  Αυτή η συνάρτηση της τάξης BranchGroup μεταγλωττίζει την πηγή BranchGroup που συσχετίζεται με αυτό το αντικείμενο και δημιουργεί ένα μεταγλωττισμένο γράφημα σκηνής.

Η κλάση ViewingPlatform χρησιμοποιείται στο να θέτει την ορατή πλευρά ενός τρισδιάστατου γραφήματος σκηνής της γλώσσας Java και ανήκει στο πακέτο com.sun.j3d.utils.universe. Η τάξη TransformGroup καθορίζει το μετασχηματισμό στο χώρο διά μέσου ενός αντικειμένου της κλάσης Transform3D και ανήκει στο πακέτο javax.media.j3d. Η τάξη Transform3D χρησιμοποιείται στο να εμφανίζει αποτελέσματα μετατροπών, περιστροφών και γραμμικών μετασχηματισμών ενώ ανήκει ομοίως στο πακέτο javax.media.j3d. Η κλάση Color3f σχηματίζει ένα χρώμα που έχει τρεις τιμές κινητής υποδιαστολής οι οποίες δηλώνουν το ποσοστό του κόκκινου, πράσινου και μπλε χρώματος. Η τάξη ετούτη ανήκει στο πακέτο javax.vecmath. Η κλάση Vector3f παράγει ένα διάνυσμα που απεικονίζεται από τρεις τιμές κινητής υποδιαστολής οι οποίες δηλώνουν συντεταγμένες. Η τάξη αυτή ανήκει στο ίδιο πακέτο με την προηγούμενη κλάση τουτέστιν στο πακέτο javax.vecmath. Η κλάση BoundingSphere δημιουργεί μία σφαιρική περιοχή η οποία καθορίζεται από ένα σημείο στο κέντρο και μία ακτίνα. Η τάξη αυτή ανήκει στο πακέτο javax.media.j3d. Η κλάση DirectionalLight προσδιορίζει ένα προσανατολισμένο φως με αρχή στο άπειρο. Και ετούτη η τάξη ανήκει στο πακέτο javax.media.j3d. Η κλάση Scene χρησιμοποιείται στο να εξάγει πληροφορίες τρισδιάστατων γραφημάτων σκηνής της γλώσσας Java με φόρτωση του κατάλληλου αρχείου. Η τάξη ετούτη ανήκει στο πακέτο com.sun.j3d.loaders. Η κλάση VrmlLoader δημιουργεί ένα αντικείμενο το οποίο μπορεί να χρησιμοποιηθεί για να φορτωθεί ένα αρχείο της γλώσσας VRML. Η κλάση αυτή ανήκει στο πακέτο org.jdesktop.j3d.loaders. vrml97. Η τάξη Alpha ανήκει στο πακέτο javax.media.j3d και παρέχει συναρτήσεις μετατροπής τιμών χρόνου σε τιμές alpha δηλαδή μία τιμή από το 0 έως το 1. Η κλάση RotationInterpolator ανήκει ομοίως στο πακέτο javax.media.j3d και προδιαγράφει τη συμπεριφορά που τροποποιεί το στοιχείο το οποίο περιστρέφεται από το στόχο του TransformGroup κάνοντας γραμμική παρεμβολή μεταξύ ενός ζεύγους συγκεκριμένων γωνιών. Η γωνία που παρεμβάλλεται παράγει ένα μετασχηματισμό περιστροφής γύρω από τον άξονα y. Η τάξη MouseRotate δίνει τη δυνατότητα στο χρήστη να ελέγχει την περιστροφή του τρισδιάστατου αντικειμένου μέσω του

ποντικιού και ανήκει στο πακέτο com.sun.j3d.utils.behaviors. Μερικοί κατασκευαστές και μέθοδοι ετούτων των κλάσεων είναι οι εξής:

- public ViewingPlatform ()
  Αυτός είναι ένας από τους κατασκευαστές της τάξης ViewingPlatform.

- public TransformGroup ()
  Ετούτος είναι ένας από τους κατασκευαστές της κλάσης TransformGroup.

- public Transform3D ()
  Αυτός είναι ένας από τους κατασκευαστές της τάξης Transform3D.

- public Color3f (float x, float y, float z)
  Ετούτος ο κατασκευαστής δημιουργεί ένα αντικείμενο της κλάσης Color3f με χρώμα που αποτελείται από ποσοστά x κόκκινου, y πράσινου και z μπλε χρώματος.

- public Vector3f (float x, float y, float z)
  Αυτός ο κατασκευαστής σχηματίζει ένα αντικείμενο της κλάσης Vector3f τουτέστιν ένα διάνυσμα με συντεταγμένες x, y και z.

- public BoundingSphere (Point3d center, double radius)
  Ετούτος ο κατασκευαστής παράγει ένα αντικείμενο της τάξης BoundingSphere με κέντρο center και ακτίνα radius. Η κλάση Point3d ανήκει στο πακέτο javax.vecmath και δημιουργεί ένα σημείο με τρεις συντεταγμένες. Οι συντεταγμένες είναι κατά τους άξονες x, y και z.

- public DirectionalLight (Color3f color, Vector3f direction)
  Αυτός ο κατασκευαστής σχηματίζει ένα αντικείμενο της τάξης DirectionalLight με χρώμα color και προσανατολισμό από το διάνυσμα direction.

- public VrmlLoader ()
  Ετούτος ο κατασκευαστής παράγει ένα αντικείμενο το οποίο το αξιοποιούμε με το να χρησιμοποιήσουμε με αυτό τη μέθοδο load για να φορτώσουμε το αρχείο της γλώσσας VRML.

- public Alpha (int loopCount, int mode, long triggerTime, long phaseDelayDuration, long increasingAlphaDuration, long increasingAlphaRampDuration, long alphaAtOneDuration, long decreasingAlphaDuration, long decreasingAlphaRampDuration, long alphaAtZeroDuration)
  Αυτός ο κατασκευαστής παράγει αντικείμενο το οποίο παίρνει όλες τις παραμέτρους της τάξης Alpha.

- public RotationInterpolator (Alpha alpha, TransformGroup target, Transform3D axisOfTransform, float minimumAngle, float maximumAngle)
  Ετούτος ο κατασκευαστής αποτελεί έναν από τους δύο κατασκευαστές της τάξης RotationInterpolator και σχηματίζει ένα αντικείμενο της εν λόγω κλάσης.

- public void setNominalViewingTransform ()
  Αυτή η μέθοδος της κλάσης ViewingPlatform θέτει την απόσταση θέασης στο μετασχηματισμό της ViewingPlatform βασισμένη στο τρέχον πεδίο θέασης.

- public void setTransform (Transform3D t1)
  Ετούτη η μέθοδος είναι μία συνάρτηση της τάξης TransformGroup και θέτει το στοιχείο του μετασχηματισμού με t1.

- public final void setScale (double scale)

Ετούτη η συνάρτηση είναι μία μέθοδος της κλάσης Transform3D και θέτει το βαθμό κλίμακας του τρέχοντος μετασχηματισμού.

- public void setInfluencingBounds (Bounds region)
  Αυτή η συνάρτηση της τάξης DirectionalLight θέτει την περιοχή ενέργειας σε συγκεκριμένα όρια. Η κλάση Bounds είναι μία αφηρημένη τάξη η οποία ανήκει στο πακέτο javax.media.j3d.

- public final void setCapability (int bit)
  Ετούτη η μέθοδος είναι μία συνάρτηση της τάξης TransformGroup και θέτει το συγκεκριμένο bit δυνατότητας.

- public BranchGroup getSceneGroup ()
  Αυτή η μέθοδος είναι μία συνάρτηση της κλάσης Scene και επιστρέφει ένα αντικείμενο της τάξης BranchGroup το οποίο περιέχει τη συνολική σκηνή που έχει φορτωθεί από το φορτωτή.

- public void setSchedulingBounds (Bounds region)
  Ετούτη η συνάρτηση της τάξης RotationInterpolator θέτει σε συγκεκριμένα όρια την περιοχή της συμπεριφοράς του αντικειμένου της εν λόγω κλάσης.

- public MouseRotate ()
  Αυτός ο κατασκευαστής είναι ένας από τους έξι κατασκευαστές της κλάσης MouseRotate.

- public void setTransformGroup (TransformGroup transformGroup)
  Αυτή η συνάρτηση της τάξης MouseRotate αντικαθιστά το αντικείμενο της κλάσης TransformGroup με το νέο αντικείμενο transformGroup. Αυτό επιτρέπει στους χρήστες να χειρίζονται διαφορετικά αντικείμενα της τάξης TransformGroup.

Για να παρουσιαστεί ένα Applet στον ιστό έχει ήδη αναφερθεί πως επιβάλλεται να εισχωρήσει μέσα σε ένα αρχείο της γλώσσας HTML. Συνεπώς ο προγραμματιστής του Applet πρέπει να είναι εξοικειωμένος τουλάχιστον με τα βασικά της γλώσσας HTML. Η γλώσσα HTML (HyperText Markup Language) είναι μία απλή γλώσσα περιγραφής υπερκειμένων. 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.

Η HTML μας δίνει τη δυνατότητα να έχουμε στο ίδιο έγγραφο που εμφανίζεται ως ιστοσελίδα (web page) κείμενο, εικόνες, φωτογραφίες, ήχους, καθώς και συνδέσμους για άλλα έγγραφα ή διευθύνσεις του Παγκόσμιου Ιστού. Τα έγγραφα HTML είναι απλά αρχεία κειμένου, επομένως μπορούν να γραφούν σε οποιονδήποτε υπολογιστή χρησιμοποιώντας έναν επεξεργαστή κειμένου (π.χ. MS NotePad, MS WordPad). Τα αρχεία έχουν κατάληξη ".html" ή ".htm". Επομένως, η αποθήκευση ενός αρχείου με εντολές στη γλώσσα HTML θα πρέπει να γίνει με κατάληξη ".html" ή ".htm", έτσι ώστε να μπορεί να το διαβάσει ένας διαφυλλιστής του Παγκόσμιου Ιστού (WWW browser). Για τη συγγραφή της γλώσσας HTML πρέπει να χρησιμοποιηθεί ένας απλός επεξεργαστής κειμένου ή ένας κειμενογράφος. Οι κειμενογράφοι χωρίζονται σε τρεις μεγάλες κατηγορίες:

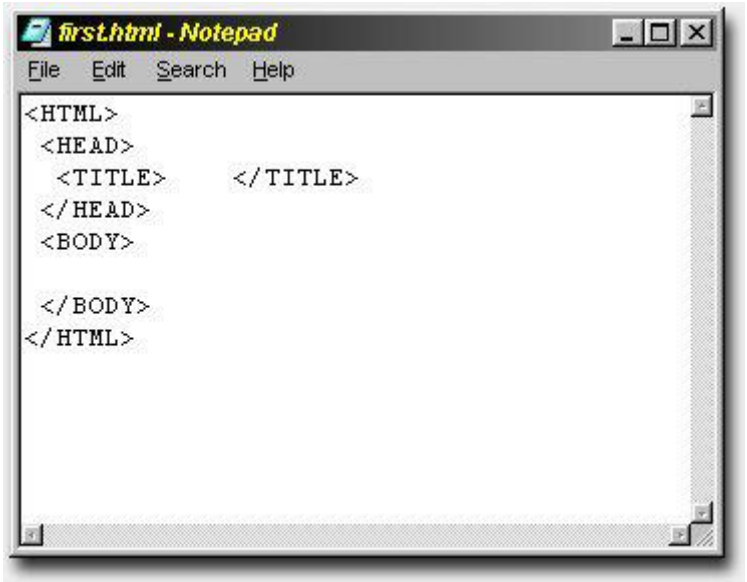
- **Κειμενογράφοι ετικετών** (tag editors), όπως το HTML Assistant Pro, το Hot Dog κ.ά., με τους οποίους ο χρήστης εύκολα γράφει κώδικα σε HTML χρησιμοποιώντας "μπάρες" με ετικέτες ή εργαλειοθήκες. Ο χρήστης βλέπει το κείμενο που επεξεργάζεται σε γλώσσα HTML. Για να δει το αποτέλεσμα ως

ιστοσελίδα πρέπει να επιλέξει την εντολή προεπισκόπησης, οπότε εμφανίζεται το αποτέλεσμα της επεξεργασίας σε διαφυλλιστή.

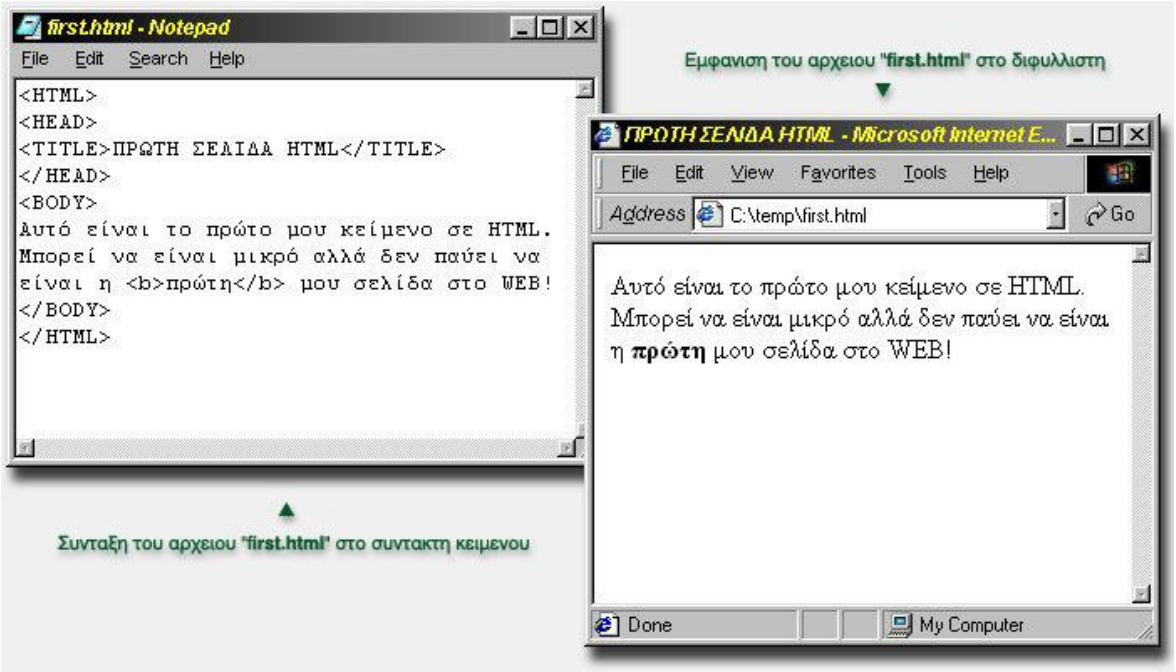
- **WYSIWYG** (What You See Is What You Get), όπως ο Netscape Composer, MS FrontPage κ.ά., τους οποίους μπορούμε να παρομοιάσουμε με MS Word®. Αυτό γίνεται γιατί έχουν ειδικές "μπάρες" που αυτοματοποιούν την εισαγωγή ετικετών, τη μορφοποίηση κειμένου κ.τ.λ.π. Ο χρήστης δε βλέπει τον κώδικα σε γλώσσα HTML αλλά μόνο το παραγόμενο αποτέλεσμα, αυτό που θα φαίνεται στο διαφυλλιστή. Πολλοί από τους κειμενογράφους αυτούς επιτρέπουν στο χρήστη να μεταπηδήσει σε ένα άλλο παράθυρο όπου εμφανίζεται ο κώδικας HTML.

- **Μετατροπείς** (converters), τα οποία είναι προγράμματα που εύκολα, χωρίς κόπο και ιδιαίτερες γνώσεις, επιτρέπουν να μετατραπεί σε ιστοσελίδα ένα κείμενο που έχει ήδη επεξεργαστεί με κάποιο άλλο εργαλείο, π.χ. MS Word, MS Excel, MS Powerpoint κ.ά. Οι μετατροπείς είναι πολύ βολικοί, αφού για τη χρήση τους δεν απαιτείται η παραμικρή γνώση HTML, πολλές φορές όμως το αποτέλεσμα της μετατροπής δεν είναι το επιθυμητό. Σε αυτή την περίπτωση πρέπει να γίνει επέμβαση στον κώδικα HTML και να πραγματοποιηθούν τροποποιήσεις, πράγμα όχι πάντα εύκολο.

Ένα έγγραφο HTML χωρίζεται σε καθορισμένες ενότητες. Οι ενότητες αυτές οριοθετούνται με τη χρήση συγκεκριμένων ετικετών οι οποίες είναι απαραίτητες σε κάθε έγγραφο HTML. Η διπλή ετικέτα **<HTML> - </HTML>** αποτελεί την πρώτη και τελευταία, αντίστοιχα, ετικέτα ενός εγγράφου HTML. Κάθε έγγραφο HTML αποτελείται από δύο κυρίως ενότητες την **επικεφαλίδα** και το **κυρίως σώμα**. Η διπλή ετικέτα **<HEAD> - </HEAD>** οριοθετεί την ενότητα της επικεφαλίδας μέσα στην οποία μπορούμε να εισάγουμε πληροφορίες για το κείμενο που ακολουθεί. Η διπλή ετικέτα **<TITLE> - </TITLE>** περιέχεται στην επικεφαλίδα του HTML εγγράφου και ό,τι γράψουμε μέσα σε αυτή θα εμφανιστεί ως τίτλος της σελίδας στο πάνω μέρος του παραθύρου του διαφυλλιστή. Η διπλή ετικέτα **<BODY> - </BODY>** αποτελεί το κύριο σώμα του HTML εγγράφου. Είναι το μεγαλύτερο τμήμα ενός HTML εγγράφου, διότι αυτό περιέχει το κείμενο και όλα τα συστατικά στοιχεία ενός υπερκειμένου. Συνοψίζοντας τα παραπάνω, πριν αρχίσουμε να εισάγουμε το κείμενο σε ένα έγγραφο HTML, αυτό θα πρέπει να έχει τη μορφή του παρακάτω σχήματος:



Και πλέον είναι όλα έτοιμα για να δημιουργηθεί μία απλή σελίδα HTML όπως φαίνεται παρακάτω:



Η HTML διαθέτει πληθώρα ετικετών (tags) για την επεξεργασία κειμένου, οι οποίες χωρίζονται σε δύο κατηγορίες. Αυτές που χρησιμοποιούμε για τη μορφοποίηση των γραμματοσειρών και των χαρακτήρων και αυτές οι οποίες καθορίζουν την οργάνωση, τη δομή και την ταξινόμηση του κειμένου. Ο καθορισμός της γραμματοσειράς που θα χρησιμοποιήσουμε στο έγγραφο μας, καθώς και η μορφοποίηση της, πραγματοποιείται με την ετικέτα **<FONT>**. Οι πιο συνηθισμένες παράμετροι της ετικέτας <FONT> είναι οι παρακάτω:

- **FACE**="αλφαριθμητικό [,αλφαριθμητικό] [, ..]"
  Η παράμετρος αυτή δέχεται ως όρισμα αλφαριθμητικό που εκφράζει το όνομα της γραμματοσειράς στην οποία θέλουμε να γραφεί το κείμενο που ακολουθεί την ετικέτα <FONT>. Η γραμματοσειρά που επιλέγουμε πρέπει να είναι εγκατεστημένη στον υπολογιστή μας, αλλιώς το κείμενο θα γραφεί στην προεπιλεγμένη γραμματοσειρά του διαφυλλιστή. Για το λόγο αυτό μπορούμε να εισάγουμε περισσότερα από ένα ονόματα γραμματοσειρών, τα οποία χωρίζονται μεταξύ τους με κόμμα.

- **SIZE**=ακέραιος
  Η παράμετρος αυτή καθορίζει το μέγεθος των χαρακτήρων της γραμματοσειράς.
  Το όρισμά της είναι ένας ακέραιος ανάμεσα στο 1 και το 7. Το μικρότερο μέγεθος ορίζεται με το 1 και το μεγαλύτερο με το 7, αντίστοιχα. Εάν δεν οριστεί, τότε το προεπιλεγμένο μέγεθος είναι το 3.

- **COLOR**="#RRGGBB"
  Η παράμετρος αυτή καθορίζει το χρώμα της γραμματοσειράς. Ο διαφυλλιστής έχει τη δυνατότητα να απεικονίσει περίπου 16 εκατομμύρια χρώματα (εάν το επιτρέπει η κάρτα γραφικών του υπολογιστή), τα οποία μπορούν να αναλυθούν σε 256 αποχρώσεις του κόκκινου, του μπλε και του πράσινου, αντίστοιχα. Η παράμετρος COLOR δέχεται σαν όρισμα τρία ζεύγη δεκαεξαδικών αριθμών (το σύμβολο "#" δηλώνει ότι ακολουθεί δεκαεξαδικός αριθμός). Το πρώτο ζευγάρι αντιστοιχεί στην απόχρωση του κόκκινου (Red), το δεύτερο στην απόχρωση του πράσινου (Green) και το τρίτο στην απόχρωση του μπλε (Blue), που συνθέτουν το τελικό χρώμα. Τα παρακάτω παραδείγματα δείχνουν τη χρήση της ετικέτας <FONT>.

Πηγαίος κώδικας

```
<FONT FACE="MS SANS SERIF" SIZE=2 COLOR="#6F006F"> Ο Ήλιος είναι μεγαλύτερος πλανήτης από τη Γη </FONT>
<FONT SIZE=5> Ο Ήλιος είναι μεγαλύτερος πλανήτης </FONT SIZE> <FONT SIZE=4> από τη Γη </FONT SIZE>
<FONT FACE="ARIAL, MS SANS SERIF" COLOR="#00A090"> Ο Ήλιος είναι μεγαλύτερος πλανήτης από τη Γη </FONT>
<FONT FACE="TAHOMA" SIZE=7 COLOR="#0000F0"> Ο Ήλιος είναι μεγαλύτερος πλανήτης από τη Γη </FONT>
```

Εμφάνιση στο διαφυλλιστή

Ο Ήλιος είναι μεγαλύτερος πλανήτης από τη Γη
Ο Ήλιος είναι μεγαλύτερος πλανήτης από τη Γη
Ο Ήλιος είναι μεγαλύτερος πλανήτης από τη Γη

Ο Ήλιος είναι μεγαλύτερος πλανήτης από τη Γη

Πολλές φορές είναι χρήσιμο να ορίζουμε το προεπιλεγμένο μέγεθος της γραμματοσειράς που θα χρησιμοποιήσουμε σε ένα έγγραφο HTML. Για το σκοπό αυτό χρησιμοποιούμε την ετικέτα **<BASEFONT>** με την παράμετρο **SIZE=**ακέραιος η οποία συντάσσεται όπως ακριβώς και η αντίστοιχη παράμετρος της ετικέτας <FONT>. Η μόνη διαφορά είναι ότι η ετικέτα <BASEFONT> πρέπει να εισαχθεί στην επικεφαλίδα του εγγράφου ανάμεσα στη διπλή ετικέτα <HEAD>. Η ετικέτα <BASEFONT SIZE=...> μας παρέχει και την ευκολία να μη χρησιμοποιούμε απόλυτες τιμές κατά την εισαγωγή της ετικέτας <FONT SIZE=... > αλλά σχετικές. Για να ορίσουμε σχετικές τιμές εισάγουμε τους χαρακτήρες " + " και " - " πριν από έναν ακέραιο που τώρα θα έχει εύρος από 1 έως 3. Το τελικό μέγεθος της γραμματοσειράς θα προκύπτει από το άθροισμα των τιμών των δύο παραμέτρων (SIZE). Τα παρακάτω παραδείγματα δείχνουν τη συνδυασμένη χρήση των ετικετών <BASEFONT> και <FONT>.

Πηγαίος κώδικας

```
<HTML>
<HEAD>
<BASEFONT SIZE=4>

</HEAD>
<BODY>
 Η Αθήνα είναι η πρωτεύουσα της Ελλάδας
<FONT SIZE=2>Η Αθήνα είναι η πρωτεύουσα της Ελλάδας </FONT>
<FONT SIZE=+1>Η Αθήνα είναι η πρωτεύουσα της Ελλάδας </FONT>
<FONT SIZE=-1>Η Αθήνα είναι η πρωτεύουσα της Ελλάδας </FONT>
<FONT SIZE=4> Η Αθήνα είναι η πρωτεύουσα της Ελλάδας </FONT>
</BODY>
</HTML>
```

Εμφάνιση στο διαφυλλιστή

Η Αθήνα είναι η πρωτεύουσα της Ελλάδας
Η Αθήνα είναι η πρωτεύουσα της Ελλάδας
Η Αθήνα είναι η πρωτεύουσα της Ελλάδας
Η Αθήνα είναι η πρωτεύουσα της Ελλάδας

Η HTML παρέχει ετικέτες για τη μορφοποίηση χαρακτήρων (προτάσεων ή και παραγράφων), που τις χρησιμοποιούμε όταν θέλουμε να δώσουμε έμφαση σε συγκεκριμένα σημεία του κειμένου. Η διπλή ετικέτα **<B> </B>** μετατρέπει τους χαρακτήρες που περικλείει σε έντονη (bold) γραφή. Η διπλή ετικέτα **<I> </I>** μετατρέπει τους χαρακτήρες που περικλείει σε πλάγια (italic) γραφή. Η διπλή ετικέτα **<U> </U>** μετατρέπει τους χαρακτήρες που περικλείει σε υπογραμμισμένους (underlined). Τα παρακάτω παραδείγματα δείχνουν τη χρήση των ετικετών <B>, <I> και <U>.

Πηγαίος κώδικας

```
Η Αθήνα είναι η πρωτεύουσα της Ελλάδας
<B>Η Αθήνα είναι η πρωτεύουσα της Ελλάδας </B>
<I>Η Αθήνα είναι η πρωτεύουσα της Ελλάδας </I>
<U>Η Αθήνα είναι η πρωτεύουσα της Ελλάδας </U>
```

Εμφάνιση στο διαφυλλιστή

Η	Αθήνα	είναι	η	πρωτεύουσα	της	Ελλάδας
Η	**Αθήνα**	**είναι**	**η**	**πρωτεύουσα**	**της**	**Ελλάδας**
Η	*Αθήνα*	*είναι*	*η*	*πρωτεύουσα*	*της*	*Ελλάδας*
Η	Αθήνα	είναι	η	πρωτεύουσα	της	Ελλάδας

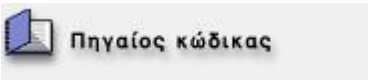
Πολλές φορές, σε ένα έγγραφο θέλουμε να μετατρέψουμε κάποιους χαρακτήρες σε δείκτες ή εκθέτες. Για τέτοιου είδους μετατροπές, οι αντίστοιχες ετικέτες στη γλώσσα HTML είναι οι **** και ****. Η διπλή ετικέτα **** μετασχηματίζει τους χαρακτήρες που περικλείει σε εκθέτες - superscript. Η διπλή ετικέτα **** μετατρέπει τους χαρακτήρες τους οποίους περικλείει σε δείκτες - subscript. Ακόμη για να εισάγουμε κείμενο το οποίο είναι στοιχισμένο στο κέντρο της σελίδας χρησιμοποιούμε τη διπλή ετικέτα **<CENTER> </CENTER>**. Τα παρακάτω παραδείγματα δείχνουν τη χρήση των ετικετών <SUP>, <SUB> και <CENTER>.

Πηγαίος κώδικας

```
Μία ταυτότητα: (x + y) <SUP> 2 </SUP> = x <SUP> 2 </SUP> + y <SUP> 2 </SUP> +
2xy
Ο χημικός τύπος του θειικού οξέος: H<SUB>2</SUB>SO<SUB>4</SUB>
```

Εμφάνιση στο διαφυλλιστή

Μία ταυτότητα: $(x + y)^2 = x^2 + y^2 + 2xy$
Ο χημικός τύπος του θειικού οξέος: H_2SO_4




```
Αυτή η σειρά είναι στοιχισμένη αριστερά.
<CENTER> Ενώ αυτή είναι στοιχισμένη στο κέντρο. </CENTER>
Αυτή η σειρά είναι στοιχισμένη αριστερά.
```

Αυτή η σειρά είναι στοιχισμένη αριστερά.
Ενώ αυτή είναι στοιχισμένη στο κέντρο.
Αυτή η σειρά είναι στοιχισμένη αριστερά.

Μία από τις ετικέτες οι οποίες οργανώνουν το κείμενο είναι η ετικέτα **<H>**. Ετούτη την ετικέτα τη χρησιμοποιούμε στην περίπτωση που επιθυμούμε να εισάγουμε επικεφαλίδες στο κείμενο μας. Η ετικέτα <H> συντάσσεται ως <Hn>, όπου n ένας ακέραιος από το 1 έως το 6, που ορίζει το μέγεθος των χαρακτήρων της επικεφαλίδας. Αντιστοιχεί 1 για το μεγαλύτερο και 6 για το μικρότερο. Το κείμενο ανάμεσα στη διπλή ετικέτα <Hn> </Hn> εμφανίζεται σε έντονη γραφή και ο διαφυλλιστής αφήνει μία κενή γραμμή πάνω και κάτω από αυτό. Η ετικέτα <H> έχει την ακόλουθη παράμετρο:

- **ALIGN**=left | center | right
  Η παράμετρος αυτή ελέγχει τη στοίχιση του κειμένου της επικεφαλίδας, η οποία μπορεί να είναι αριστερή (left), στο κέντρο (center) ή δεξιά (right). Αν δεν τη χρησιμοποιήσουμε, τότε το συγκεκριμένο κείμενο θα στοιχηθεί αριστερά. Τα παρακάτω παραδείγματα εμφανίζουν όλες τις δυνατές επικεφαλίδες οι οποίες δημιουργούνται με τη χρήση της ετικέτας <Hn>.

```
<H1 ALIGN=LEFT> Επικεφαλίδα ύψους 1 </H1>
<H2 ALIGN=RIGHT> Επικεφαλίδα ύψους 2 </H2>
<H3 ALIGN=CENTER> Επικεφαλίδα ύψους 3 </H3>
<H4> Επικεφαλίδα ύψους 4 </H4>
<H5 ALIGN=RIGHT> Επικεφαλίδα ύψους 5 </H5>
<H6 ALIGN=CENTER> Επικεφαλίδα ύψους 6 </H6>
```

Επικεφαλίδα ύψους 1

Επικεφαλίδα ύψους 2

Επικεφαλίδα ύψους 3

Επικεφαλίδα ύψους 4

Επικεφαλίδα ύψους 5

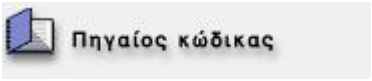
Επικεφαλίδα ύψους 6

Αντίθετα με τους επεξεργαστές κειμένου, ο διαφυλλιστής αγνοεί κάθε αλλαγή παραγράφου στο έγγραφο που συντάσσουμε. Στην περίπτωση όπου ο χρήστης θέλει

να κάνει αλλαγή παραγράφου, τότε πρέπει να χρησιμοποιήσει τη διπλή ετικέτα **<P> </P>**. Ο διαφυλλιστής θα αφήσει μία κενή γραμμή πάνω και κάτω από το κείμενο που βρίσκεται μέσα στην ετικέτα <P> και έτσι δημιουργείται μία παράγραφος. Η διπλή ετικέτα **<DIV> </DIV>** χρησιμοποιείται για να διαιρέσει το υπερκείμενο σε ενότητες (divisions). Η ετικέτα <DIV> δεν παρέχει κάποια μορφοποίηση στο υπερκείμενο αλλά κύριος σκοπός της είναι η διαμόρφωση μεγάλων τμημάτων κειμένου ομαδοποιώντας τις ετικέτες (<P> <CENTER> <H> κ.ο.κ.) σε ενιαία τμήματα υπερκειμένου. Το γεγονός αυτό καθιστά όχι μόνο ευδιάκριτες τις μορφοποιήσεις σε μία ιστοσελίδα αλλά και τη συγγραφή κώδικα HTML πιο εύκολη από το συντάκτη. Οι ετικέτες <P> και <DIV> έχουν τις εξής παραμέτρους:

- **ALIGN**=left | center | right
  Η παράμετρος αυτή της ετικέτας <P> ελέγχει τη στοίχιση του κειμένου της παραγράφου, η οποία μπορεί να είναι αριστερά (left), στο κέντρο (center) ή δεξιά (right). Αν δεν τη χρησιμοποιήσουμε, τότε το κείμενο θα στοιχηθεί αριστερά.

- **ALIGN**=left | center | right
  Η παράμετρος αυτή της ετικέτας <DIV> ελέγχει τη στοίχιση των περιεχομένων που ομαδοποιεί η εν λόγω ετικέτα. Η στοίχιση μπορεί να είναι αριστερά (left), στο κέντρο (center) ή δεξιά (right). Αν δεν τη χρησιμοποιήσουμε, τότε τα περιεχόμενα της στοιχίζονται αριστερά. Παρακάτω φαίνεται ένα παράδειγμα δημιουργίας παραγράφων με τη γλώσσα HTML.




```
Η γραμμή αυτή βρίσκεται πριν την καινούρια παράγραφο.
<P ALIGN=CENTER>
Η γραμμή αυτή ξεκινά μία καινούρια παράγραφο. Το κείμενο που βρίσκεται εδώ θα απέχει μία γραμμή από τις προηγούμενες και τις επόμενες γραμμές του κειμένου και θα είναι στοιχισμένο στο κέντρο.
</P>
Η γραμμή αυτή βρίσκεται μετά την καινούρια παράγραφο.
```

Η γραμμή αυτή βρίσκεται πριν την καινούρια παράγραφο.

Η γραμμή αυτή ξεκινά μία καινούρια παράγραφο. Το κείμενο που βρίσκεται εδώ θα απέχει μία γραμμή από τις προηγούμενες και τις επόμενες γραμμές του κειμένου και είναι θα στοιχισμένο στο κέντρο.

Η γραμμή αυτή βρίσκεται μετά την καινούρια παράγραφο.

Επειδή ο διαφυλλιστής αγνοεί κάθε αλλαγή γραμμής στο κείμενο που συντάσσουμε, όταν θέλουμε να εισάγουμε κείμενο που ξεκινά από μία καινούρια γραμμή, τότε χρησιμοποιούμε την ετικέτα **
. Χρησιμοποιώντας πολλές φορές την ετικέτα <BR>, αφήνουμε κενές γραμμές στην οθόνη του διαφυλλιστή. Όταν θέλουμε να εισάγουμε μία οριζόντια γραμμή στο υπερκείμενο μας, χρησιμοποιούμε την ετικέτα **<HR>. Η οριζόντια γραμμή που θα εμφανιστεί στο διαφυλλιστή, καταλαμβάνει μία σειρά και αποκλείει την ύπαρξη κειμένου αριστερά ή δεξιά της. Στο κάτω μέρος της γραμμής θα εμφανιστεί μία ανεπαίσθητη σκιά που δίνει μία τρισδιάστατη προοπτική στη γραμμή. Οι παράμετροι της ετικέτας <HR> είναι οι παρακάτω:

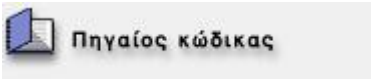
- **SIZE**=ακέραιος
  Η παράμετρος αυτή δέχεται σαν όρισμα έναν ακέραιο που καθορίζει το πάχος της οριζόντιας γραμμής σε εικονοστοιχεία (pixels). Εάν δεν καθοριστεί, τότε θα εμφανιστεί μόνο η σκιά της γραμμής.

- **WIDTH**=ακέραιος | "ποσοστό%"
  Η παράμετρος αυτή καθορίζει το μήκος της γραμμής. Το όρισμά της είναι είτε ένας ακέραιος που αντιπροσωπεύει το μήκος της σε εικονοστοιχεία είτε ένα ποσοστό σε σχέση με το μήκος της οθόνης του διαφυλλιστή.

- **ALIGN**=left | center | right
  Η παράμετρος αυτή ελέγχει τη στοίχιση της γραμμής, η οποία μπορεί να είναι αριστερή (left), στο κέντρο (center) ή δεξιά (right). Αν δεν τη χρησιμοποιήσουμε, τότε η γραμμή θα στοιχηθεί στο κέντρο.

- **COLOR**="#RRGGBB"
  Η παράμετρος αυτή καθορίζει το χρώμα της γραμμής στη μορφή RGB που εξηγήσαμε παραπάνω.

- **NOSHADE**
  Η παράμετρος αυτή δεν εμφανίζει τη σκιά που υπάρχει στο κάτω μέρος της γραμμής. Παρακάτω έχουμε παραδείγματα αλλαγής γραμμής με τη χρήση της ετικέτας
 και οριζόντιων γραμμών με τη χρήση της ετικέτας <HR>.

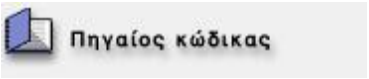


Αυτή είναι η πρώτη σειρά.
Αυτή η σειρά θα εμφανιστεί στην επόμενη γραμμή.

Ενώ αυτή η σειρά θα εμφανιστεί τρεις γραμμές πιο κάτω.

Αυτή είναι η πρώτη σειρά.
Αυτή η σειρά θα εμφανιστεί στην επόμενη γραμμή.

Ενώ αυτή η σειρά θα εμφανιστεί τρεις γραμμές πιο κάτω.




```
<HR SIZE=4>
<HR SIZE=4 NOSHADE>
<HR ALIGN=LEFT SIZE=4 COLOR="000090">
<HR WIDTH="50%" ALIGN=RIGHT COLOR="#000000" SIZE=30>
<HR ALIGN=CENTER WIDTH=350 SIZE=3 COLOR="#909090">
```

Η ετικέτα **<META>** χρησιμοποιείται με σκοπό να συμπεριληφθούν πληροφορίες για την περιγραφή μίας ιστοσελίδας, οι οποίες δεν εμφανίζονται στην οθόνη. Συνήθως

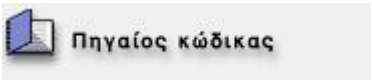
περιέχει μία μικρή περίληψη του περιεχομένου της ιστοσελίδας αλλά και λέξεις - κλειδιά (keywords), με τις οποίες μία μηχανή αναζήτησης θα την εντοπίσει πιο γρήγορα. Η ετικέτα <META> χρησιμοποιείται μόνο στο μέρος της επικεφαλίδας ενός υπερκειμένου, δηλαδή ανάμεσα στις ετικέτες <HEAD> </HEAD>. Η σύνταξη της έχει τη μορφή: <META NAME= "τύπος πληροφορίας" CONTENT=" πληροφορία">. Οι παράμετροι της ετικέτας <META> μπορούν να λάβουν τις παρακάτω τιμές:

- **NAME**="description" **CONTENT**="Περιγραφή της ιστοσελίδας"
  Χρησιμοποιώντας την τιμή description στην παράμετρο NAME προσδιορίζουμε ότι η παράμετρος CONTENT θα περιέχει μία περιγραφή της ιστοσελίδας. Το κείμενο της περιγραφής (δηλαδή το μέγεθος της τιμής της παραμέτρου CONTENT) δε μπορεί να ξεπερνά τους 200 χαρακτήρες.

- **NAME**="comments" **CONTENT**="λέξη, λέξη, λέξη, ..."
  Χρησιμοποιώντας την τιμή comments στην παράμετρο NAME προσδιορίζουμε ότι η παράμετρος CONTENT θα περιέχει μία λίστα λέξεων (που χωρίζονται μεταξύ τους με κόμμα) οι οποίες "χαρακτηρίζουν" το περιεχόμενο της ιστοσελίδας και μπορούν να χρησιμοποιηθούν από μία μηχανή αναζήτησης για τον εντοπισμό της. Η λίστα με τις λέξεις - κλειδιά (δηλαδή το μέγεθος της τιμής της παραμέτρου CONTENT) δε μπορεί να ξεπερνά τους 100 χαρακτήρες. Στο παράδειγμα που ακολουθεί χρησιμοποιούμε την ετικέτα με τα ζεύγη των παραμέτρων που αναλύσαμε.

```
<HEAD>
<TITLE> ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ HTML </TITLE>
<META
  NAME="description"
  CONTENT="Ιστοσελίδα για το πρώτο κεφάλαιο των μαθημάτων της γλώσσας HTML,
  με μια εισαγωγή στη χρήση της γλώσσας και τη σύνταξη υπερκειμένων">
<META
  NAME="comments"
  CONTENT="μαθήματα HTML, εισαγωγή, ιστοσελίδα" >
</HEAD>
```

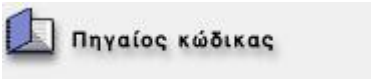
Μία άλλη επιπλέον χρήση της ετικέτας <META> με τις κατάλληλες παραμέτρους, είναι η εξομοίωση της επιλογής του κωδικού του αλφαβήτου της γλώσσας, π.χ. Ελληνικά, Λατινικά κ.ο.κ., με την αντίστοιχη επιλογή από τον κατάλογο επιλογών (menu) του διαφυλλιστή. Ο κωδικός για την εμφάνιση ελληνικών χαρακτήρων είναι ο "ISO-8859-7" ο οποίος έχει καθοριστεί από το **Διεθνή Οργανισμό Προτυποποίησης** (International Organisation for Standardization - ISO). Επίσης, για τους χρήστες του λειτουργικού συστήματος MS - Windows® ισχύει και ο κωδικός "windows - 1253". Πιο κάτω φαίνεται ένα παράδειγμα της ετικέτας <META> με τις παραμέτρους για την επιλογή του αλφαβήτου της ελληνικής γλώσσας.




```
<HTML>
 <HEAD>
   <TITLE> ..... </TITLE>
   <META http-equiv="Content-Type" content="text/html; charset=ISO-8859-7">
 </HEAD>
.
```

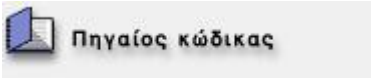
.
.
</HTML>
Τέλος, μία άλλη χρήση της ετικέτας <META>, με τις κατάλληλες παραμέτρους, είναι η αυτόματη μεταφορά του αναγνώστη της τρέχουσας ιστοσελίδας σε μία άλλη τοποθεσία. Αυτή η χρήση της ετικέτας <META> είναι πολύ βολική όταν η URL διεύθυνση μίας ιστοσελίδας έχει αλλάξει / μεταφερθεί σε άλλη διεύθυνση και θέλουμε να υπάρχει αυτόματη μετάβαση του χρήστη σε αυτή. Στην περίπτωση αυτή η ετικέτα <META> χρησιμοποιείται μέσα στο κυρίως σώμα του υπερκειμένου (ετικέτες <BODY> - </BODY>). Η σύνταξη της ετικέτας γίνεται ως εξής:
<META
HTTP-EQUIV="Refresh"
CONTENT="αριθμός δευτερολέπτων πριν την αυτόματη μετάβαση; **URL**=νέα URL διεύθυνση" >

Σε ετούτο το παράδειγμα φαίνεται η αυτόματη μετάβαση σε άλλη τοποθεσία με τη χρήση της ετικέτας <META>.




```
<HTML>
<HEAD>
<TITLE> ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ HTML </TITLE>
</HEAD>

<BODY>
<META HTTP-EQUIV="Refresh" CONTENT="3; URL=http://www.newsite/">

Η ιστοσελίδα του 1ου Κεφαλαίου έχει μεταφερθεί. <BR>
Περιμένετε 3 δευτερόλεπτα για να μεταφερθείτε στη νέα διεύθυνση. <BR>
Παρακαλούμε να σημειώσετε τη νέα διεύθυνση που είναι: http://www.newsite/

</BODY>
</HTML>
```

Όταν θέλουμε κάποια τμήματα του εγγράφου μας (ετικέτες ή απλό κείμενο) να αγνοηθούν από το διαφυλλιστή τότε τα περικλείουμε με τα σύμβολα **<!-- , -->**. Η χρήση των παραπάνω συμβόλων είναι χρήσιμη για την εισαγωγή σχολίων στο έγγραφο μας. Υπάρχουν μερικά ειδικά σύμβολα που έχουν ιδιαίτερη σημασία σε ένα έγγραφο HTML και χρησιμοποιούνται ήδη για τη σύνταξη των ετικετών. Άρα δε μπορούμε να τα συμπεριλάβουμε σε ένα έγγραφο, αφού θα εκτελεστούν ως εντολές από το διαφυλλιστή αντί να εμφανιστούν στην ιστοσελίδα. Για το λόγο αυτό, η γλώσσα HTML παρέχει περιφραστικούς τρόπους για να εισάγουμε αυτούς τους χαρακτήρες σε μία ιστοσελίδα. Ο χαρακτήρας < εκφράζει το "μικρότερο από" και εισάγεται στην ιστοσελίδα με την έκφραση **<**. Ο χαρακτήρας > εκφράζει το "μεγαλύτερο από" και εισάγεται στην ιστοσελίδα με την έκφραση **>**. Το κενό διάστημα " " δημιουργείται με τη χρήση του πλήκτρου spacebar και εισάγεται στην ιστοσελίδα με την έκφραση ** **. Τα διπλά εισαγωγικά (" , ") εισάγονται στην ιστοσελίδα με την έκφραση **"**. Ο χαρακτήρας **&** εκφράζει το "και" σε συντομογραφία και εισάγεται στην ιστοσελίδα με την έκφραση **&**. Παρακάτω εμφανίζεται ένα παράδειγμα εισαγωγής ειδικών χαρακτήρων.

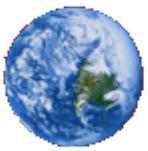



```
<P>
Ισχύει: 5 &lt; 6 και το 6 &gt; 3. <BR>
```

Βρίσκομαι στο γραφείο Δευτέρα & Τρίτη
Το καλύτερο βιβλίο που έχω διαβάσει είναι: " Το Όνομα του Ρόδου ".
Εδώ και ...   πιο πέρα!!!
</P>

Ισχύει: 5 < 6 και το 6 > 3.
Βρίσκομαι στο γραφείο Δευτέρα & Τρίτη
Το καλύτερο βιβλίο που έχω διαβάσει είναι: " Το Όνομα του Ρόδου ".
Εδώ και ... πιο πέρα!!!

Σε μία ιστοσελίδα μπορούμε να ενσωματώσουμε εικόνες. Οι περισσότεροι διαφυλλιστές υποστηρίζουν εικόνες που υπακούουν στα πρότυπα **JPEG** (ή JPG) και **GIF**, ενώ σπανιότερα αυτές που ακολουθούν το πρότυπο **BMP**. Οι εικόνες GIF μπορεί να είναι στατικές ή να περιέχουν κίνηση (Animated GIF). Οι εικόνες GIF υποστηρίζουν επίσης διαφάνεια ως προς το υπόβαθρο της εικόνας. Μία διαφανή εικόνα έχει αόρατο φόντο (background). Η τοποθέτησή της μέσα σε ένα υπερκείμενο την κάνει να μοιάζει ότι επιπλέει, αφού το χρώμα που αποκτά το φόντο της είναι αυτό της ιστοσελίδας. Δείτε τη διαφορά μεταξύ της ίδιας εικόνας όπου αριστερά έχει επεξεργαστεί ώστε να είναι διάφανη και δεξιά που έχει μαύρο υπόβαθρο. Η εισαγωγή εικόνων σε ένα έγγραφο γίνεται με τη χρήση της ετικέτας **<IMG>**. Οι παράμετροι της ετικέτας <IMG> είναι οι παρακάτω:



- **SRC**="URL διεύθυνση"
  Η παράμετρος SRC καθορίζει τη URL διεύθυνση (Source) του αρχείου εικόνας που θέλουμε να εισάγουμε. Η παράμετρος αυτή είναι βασική και πρέπει να δηλώνεται πάντα, γιατί χωρίς αυτή η ετικέτα <IMG> δεν έχει κανένα αποτέλεσμα.

- **ALIGN**=left | right | top | texttop | middle | absmiddle | baseline | bottom | absbottom
  Η παράμετρος αυτή καθορίζει τη στοίχιση της εικόνας σε σχέση με το κείμενο που βρίσκεται γύρω από αυτή ή καθορίζει τη θέση της σε σχέση με όλη τη σελίδα. Πιο συγκεκριμένα:

  - **ALIGN=left**: Η εικόνα θα τοποθετηθεί στο αριστερό περιθώριο (στο επόμενο διαθέσιμο κενό αριστερά) αναγκάζοντας το κείμενο να εμφανιστεί στη δεξιά πλευρά της.
  - **ALIGN=right**: Η εικόνα θα τοποθετηθεί στο δεξιό περιθώριο αναγκάζοντας το κείμενο να εμφανιστεί στην αριστερή πλευρά της.
  - **ALIGN=top**: Η εικόνα θα τοποθετηθεί στο ύψος του υψηλότερου στοιχείου που υπάρχει στη γραμμή.
  - **ALIGN=texttop**: Η επάνω πλευρά της εικόνας θα τοποθετηθεί στο ύψος του υψηλότερου στοιχείου που υπάρχει στη γραμμή.
  - **ALIGN=middle**: Η επάνω πλευρά της εικόνας θα τοποθετηθεί στο μέσο της γραμμής, στο ίδιο ύψος με τα υπόλοιπα στοιχεία της.
  - **ALIGN=absmiddle**: Η εικόνα θα τοποθετηθεί στο μέσο της γραμμής, τόσο σε οριζόντιο όσο και σε κατακόρυφο άξονα σε σχέση με τα υπόλοιπα στοιχεία της.
  - **ALIGN=baseline**: Η κάτω πλευρά της εικόνας τοποθετείται στο ίδιο ύψος με το χαμηλότερο όριο της τρέχουσας γραμμής.
  - **ALIGN=bottom**: Η κάτω πλευρά της εικόνας τοποθετείται στο ίδιο ύψος με τα υπόλοιπα στοιχεία της γραμμής.

- **ALIGN=absbottom**: Η κάτω πλευρά της εικόνας τοποθετείται στο ίδιο ύψος με το χαμηλότερο στοιχείο της γραμμής.

HSPACE=ακέραιος - **VSPACE**=ακέραιος
Οι δύο αυτές παράμετροι καθορίζουν την οριζόντια ή κάθετη, αντίστοιχα, απόσταση που θα έχει η εικόνα σε σχέση με το κείμενο που την περιβάλλει. Χρησιμοποιούμε τις παραπάνω παραμέτρους μαζί με την παράμετρο ALIGN για ορίσουμε ακριβώς τη θέση εικόνας και κειμένου. Οι ακέραιοι αριθμοί που δέχονται σαν ορίσματα, αντιστοιχούν σε εικονοστοιχεία (pixels).

WIDTH=ακέραιος - **HEIGHT**=ακέραιος
Οι δύο αυτές παράμετροι καθορίζουν σε εικονοστοιχεία (pixels) το πλάτος και το ύψος της εικόνας που πρόκειται να εμφανιστεί στη σελίδα. Ο σκοπός των παραμέτρων HEIGHT και WIDTH δεν είναι να αυξομειώσουν το μέγεθος της εικόνας κατά βούληση, αλλά να καθοριστεί ο χώρος που θα πρέπει να δεσμευθεί από το διαφυλλιστή στη σελίδα για την προβολή της εικόνας. Με αυτόν τον τρόπο επιταχύνεται η διαδικασία φόρτωσης των εικόνων. Για αυτό, όπου είναι εφικτό, συνιστάται να δηλώνεται το μέγεθος της εικόνας.

BORDER=ακέραιος - **BORDERCOLOR**="#RRGGBB"
Οι δύο αυτές παράμετροι καθορίζουν το μέγεθος και το χρώμα του περιθωρίου που δημιουργεί ο διαφυλλιστής γύρω από την εικόνα. Η παράμετρος BORDER δέχεται όρισμα σε εικονοστοιχεία (pixels), ενώ η παράμετρος BORDERCOLOR δέχεται ως όρισμα ένα χρώμα σε μορφή RGB.

ALT="αλφαριθμητικό"
Η παράμετρος ALT δέχεται ως όρισμα ένα αλφαριθμητικό, το οποίο εμφανίζεται στη θέση της εικόνας, στην οθόνη του διαφυλλιστή, όταν για οποιοδήποτε λόγο η εμφάνιση της αποτύχει. Πολλές φορές επίσης θέλουμε να γράψουμε κάποιο κείμενο το οποίο περιέχει επεξηγήσεις ή σχόλια που αφορούν μία εικόνα, τα οποία όμως δε θέλουμε να περιέχονται στο κείμενο της σελίδας. Η παράμετρος ALT μας παρέχει και αυτή την ευκολία. Το κείμενο με το οποίο συντάσσεται, εμφανίζεται σαν μία μικρή επεξήγηση όταν και όσο ο δείκτης του ποντικιού βρίσκεται μέσα στα όρια της εικόνας.

Σε μία ιστοσελίδα έχουμε τη δυνατότητα να καθορίσουμε τα χρώματα τόσο του φόντου όσο και του κειμένου. Για την επίτευξη του σκοπού ετούτου δεν υπάρχουν ειδικές ετικέτες, αλλά η γλώσσα HTML παροχετεύει μία ομάδα παραμέτρων οι οποίες ενσωματώνονται στην ετικέτα **<BODY>**. Αυτές οι παράμετροι έχουν καθολικό και όχι τοπικό χαρακτήρα όπως συμβαίνει σε ετικέτες που έχουμε ήδη γνωρίσει (π.χ. <FONT COLOR=...>, <HR COLOR=...> κ.ά.). Δηλαδή οι παράμετροι ετούτοι επιδρούν σε ολόκληρη την ιστοσελίδα. Αναλυτικότερα οι παράμετροι είναι:

BACKGROUND="URL διεύθυνση"
Η παράμετρος BACKGROUND δέχεται σαν όρισμα τη URL διεύθυνση ενός αρχείου εικόνας. Η εικόνα αυτή επαναλαμβάνεται με τέτοιο τρόπο ώστε να καλύψει όλη την επιφάνεια της σελίδας και να δημιουργήσει έτσι ένα ομοιογενές φόντο.

BGCOLOR="#RRGGBB"
Η παράμετρος αυτή καθορίζει το χρώμα της σελίδας στη μορφή RGB, όταν δε χρησιμοποιούμε την παράμετρο BACKGROUND.

TEXT="#RRGGBB"
Η παράμετρος αυτή καθορίζει το χρώμα του κειμένου στη μορφή RGB.

Οι υπερσύνδεσμοι αποτελούν αναμφισβήτητα μία από τις σημαντικότερες εφαρμογές του Παγκόσμιου Ιστού. Μας παρέχουν τη δυνατότητα να πλοηγηθούμε στις αμέτρητες περιοχές που περιλαμβάνει και να χρησιμοποιήσουμε τις υπηρεσίες του. Οι **σύνδεσμοι** (links) δεν είναι τίποτε άλλο από λέξεις / φράσεις "κλειδιά" ή εικόνες, μέσω των οποίων μπορούμε να συνδεθούμε και να οδηγηθούμε είτε σε κάποια άλλη σελίδα οπουδήποτε στο Διαδίκτυο είτε σε κάποιο συγκεκριμένο σημείο της τρέχουσας ή άλλης σελίδας. Οι σύνδεσμοι επίσης μπορούν να ικανοποιήσουν τις απαιτήσεις του χρήστη για κάποια από τις υπηρεσίες του Διαδικτύου, όπως την ηλεκτρονική αλληλογραφία ή τη μεταφορά αρχείων. Η γλώσσα HTML υποστηρίζει τη δημιουργία συνδέσμων οι οποίοι μπορεί να είναι απλές λέξεις, φράσεις, ακόμη και εικόνες. Σε μία ιστοσελίδα οι σύνδεσμοι (που αποτελούνται από χαρακτήρες) εμφανίζονται υπογραμμισμένοι (underlined). Το χρώμα των χαρακτήρων μεταβάλλεται ανάλογα με το αν έχουμε ή όχι χρησιμοποιήσει το σύνδεσμο. Επίσης ο δείκτης του ποντικιού αλλάζει εμφάνιση κάθε φορά που περνάει πάνω από ένα σύνδεσμο. Η δημιουργία ενός συνδέσμου στην HTML πραγματοποιείται με τη χρήση της διπλής ετικέτας **<A> - </A>**. Το κείμενο ή η εικόνα (με τη χρήση της ετικέτας <IMG>) που θα αποτελέσει το σύνδεσμο, περικλείεται στη διπλή ετικέτα <A>. Οι παράμετροι της ετικέτας <A> είναι οι παρακάτω:

- **NAME**="αλφαριθμητικό"
  Η παράμετρος NAME χρησιμοποιείται όταν θέλουμε να δημιουργήσουμε ένα "στόχο" (anchor - αγκύρωση) για ένα σύνδεσμο σε μία σελίδα. Το αλφαριθμητικό δηλώνει ένα "συμβολικό όνομα" που οριοθετεί τη συγκεκριμένη γραμμή της σελίδας όπου δηλώνεται η εντολή <A NAME="..."> </A> και χρησιμοποιείται για να την εντοπίζουμε σαν στόχο.

- **HREF**="URL διεύθυνση"
  Η παράμετρος αυτή στην ουσία υλοποιεί το σύνδεσμο και σαν όρισμα της περιέχει το αρχείο στο οποίο θέλουμε να μεταβούμε. Το κείμενο ή η εικόνα που βρίσκεται ανάμεσα στις ετικέτες <A HREF=" ... "> - </A> θα εμφανιστεί όπως περιγράψαμε παραπάνω. Η παράμετρος HREF συντάσεται με την πλήρη **URL** του πόρου (αρχείου) που στοχεύουμε. Το URL (UNIFORM SOURCE LOCATOR) αποτελεί έναν ενιαίο (ομοιόμορφο) τρόπο εντοπισμού πόρων στο Διαδίκτυο. Με τη χρήση του URL μπορούμε να συνδεθούμε με άλλους εξυπηρετητές (servers) σε όλο τον κόσμο, μέσω διαφορετικών πρωτοκόλλων και υπηρεσιών. Μία συνοπτική περιγραφή του URL γίνεται με την παρακάτω σύνταξη:

  υπηρεσία ή πρωτόκολλο://όνομα εξυπηρετητή/διαδρομή/όνομα αρχείου
  π.χ.:
  http://www.adobe.com/supportservice/custsupport/download.html

  Πιο αναλυτικά, το URL μπορεί να δεχθεί τα παρακάτω πρωτόκολλα ή υπηρεσίες:
  - **"file://..."**: Χρησιμοποιείται για την ανάγνωση αρχείων από τον τοπικό δίσκο.
  - **"ftp://..."**: (File Transfer Protocol) Πρωτόκολλο που χρησιμοποιείται για τη σύνδεση με έναν εξυπηρετητή μεταφοράς αρχείων (FTP server) για μεταφορά αρχείων.
  - **"http://..."**: (HyperText Transfer Protocol) Πρωτόκολλο που χρησιμοποιείται για τη σύνδεση με έναν εξυπηρετητή (WWW server).
  - **"gopher://..."**: Πρωτόκολλο που χρησιμοποιείται για τη σύνδεση με έναν εξυπηρετητή Gopher (Gopher server).
  - **"telnet://..."**: Πρωτόκολλο που μετατρέπει τον υπολογιστή σε τερματικό του εξυπηρετητή που δηλώνεται.
  - **"mailto:..."**: Υπηρεσία αποστολής μηνυμάτων διά μέσου του ηλεκτρονικού ταχυδρομείου.

- **"news:..."**: Υπηρεσία για την ανάγνωση μηνυμάτων από ένα newsgroup που δηλώνεται.

Εάν θέλουμε να παραπέμψουμε το διαφυλλιστή σε ένα συγκεκριμένο σημείο της σελίδας, τότε σαν τμήμα της URL διεύθυνσης έπειτα από το όνομα του αρχείου (path / filename), χρησιμοποιούμε το χαρακτήρα #, ο οποίος ακολουθείται από αλφαριθμητικό που εκφράζει το "συμβολικό όνομα" του σημείου εκείνου. Το σημείο αυτό πρέπει να έχει ήδη καθοριστεί σαν αγκύρωση, με τη χρήση της παραμέτρου <A NAME=" ... "> - </A>.

Παράδειγμα: Αν στο αρχείο "myfile.html" έχω δημιουργήσει σε ένα σημείο ένα "στόχο" με το όνομα "TOP", χρησιμοποιώντας τη σύνταξη <A NAME="TOP"> Στόχος </A>, τότε για να μεταβώ στη σελίδα αυτή θα χρησιμοποιήσω τη σύνταξη <A HREF = "myfile.html" ... ενώ για να μεταβώ στο συγκεκριμένο στόχο που έχω καθορίσει στη σελίδα αυτή θα χρησιμοποιήσω τη σύνταξη <A HREF="myfile.html#TOP" ...

Επισήμανση: Οι δύο παράμετροι NAME και HREF επιτελούν διαφορετικές λειτουργίες μέσα στην ετικέτα <A>, για αυτό δε μπορεί να υπάρχουν και οι δύο ταυτόχρονα στην ίδια ετικέτα. Πρέπει όμως να υπάρχει τουλάχιστον μία από αυτές, αλλιώς η ετικέτα <A> δεν έχει νόημα.

- **TARGET**="αλφαριθμητικό"
  Μία σελίδα μπορεί να είναι χωρισμένη σε πλαίσια (frames). Η παράμετρος TARGET χρησιμοποιείται κυρίως με τη σύνταξη <A HREF= ... για να καθορίσει σε ποιο από τα πλαίσια θα εμφανιστεί η σελίδα που θέλουμε να μεταβούμε μέσω του συνδέσμου. Το αλφαριθμητικό με το οποίο συντάσσεται αποτελεί ένα "συμβολικό όνομα" που χαρακτηρίζει το συγκεκριμένο πλαίσιο.

Ο χρωματισμός των συνδέσμων γίνεται από τρεις διαφορετικές παραμέτρους οι οποίοι αντιστοιχούν στις διαφορετικές μορφές ενός συνδέσμου. Αυτές οι τρεις παράμετροι ενσωματώνονται στην ετικέτα <BODY> ως επιπλέον παράμετροι. Οι παράμετροι ετούτοι καθορίζουν το χρώμα που έχουν οι σύνδεσμοι πριν τη χρησιμοποίηση τους τουτέστιν πριν πατηθούν οι σύνδεσμοι, μετά τη χρησιμοποίηση τους (μετά το πάτημα των συνδέσμων) και κατά τη διάρκεια της χρησιμοποίησης τους (τη στιγμή που γίνεται το πάτημα των συνδέσμων). Αναλυτικότερα οι παράμετροι είναι:

- **LINK**="#RRGGBB"
  Η παράμετρος αυτή καθορίζει (στη μορφή RGB) το χρώμα του συνδέσμου όταν δεν τον έχουμε χρησιμοποιήσει.

- **ALINK**="#RRGGBB"
  Η παράμετρος αυτή καθορίζει (στη μορφή RGB) το χρώμα που θα έχει ο σύνδεσμος τη στιγμή που τον ενεργοποιούμε.

- **VLINK**="#RRGGBB"
  Η παράμετρος αυτή καθορίζει (στη μορφή RGB) το χρώμα του συνδέσμου αφού τον έχουμε χρησιμοποιήσει.

Οι **διατεταγμένοι κατάλογοι** (ordered lists) παρουσιάζουν τα στοιχεία τους σε αριθμητική διάταξη, έτσι ώστε το κάθε ένα να ξεχωρίζει από τα υπόλοιπα. Για να δημιουργήσουμε μία διατεταγμένη λίστα στη γλώσσα HTML, χρησιμοποιούμε τη διπλή ετικέτα **<OL> - </OL>**. Η ετικέτα <OL> ορίζει την αρχή, το τέλος και τον τρόπο αρίθμησης της διατεταγμένης λίστας. Για να εισάγουμε όμως στοιχεία στη λίστα, χρησιμοποιούμε την ετικέτα **<LI>** (List Item) ανάμεσα στη διπλή ετικέτα <OL>. Οι

παράμετροι της ετικέτας <OL> είναι οι παρακάτω ενώ μετά από αυτούς υπάρχουν παραδείγματα κατασκευής διατεταγμένων καταλόγων:

- **TYPE**="1 | A | a | I | i"
  Η παράμετρος αυτή καθορίζει τον τύπο των στοιχείων που θα χρησιμοποιηθούν για την αρίθμηση των περιεχομένων της λίστας. Πιο συγκεκριμένα:

  - **TYPE=1**: Για την αρίθμηση θα χρησιμοποιηθούν ακέραιοι αριθμοί.
  - **TYPE=A**: Για την αρίθμηση θα χρησιμοποιηθούν κεφαλαία γράμματα του λατινικού αλφαβήτου.
  - **TYPE=a**: Για την αρίθμηση θα χρησιμοποιηθούν τα μικρά γράμματα του λατινικού αλφαβήτου.
  - **TYPE=I**: Για την αρίθμηση θα χρησιμοποιηθεί ο ρωμαϊκός τρόπος μέτρησης, με κεφαλαίους λατινικούς χαρακτήρες.
  - **TYPE=i**: Για την αρίθμηση θα χρησιμοποιηθεί ο ρωμαϊκός τρόπος μέτρησης, με μικρούς λατινικούς χαρακτήρες.

Εάν δεν καθοριστεί κάποιος από τους παραπάνω τύπους, τότε ο πρώτος τύπος, αυτός δηλαδή που χρησιμοποιεί ακέραιους αριθμούς, είναι ο προκαθορισμένος.

- **START**=ακέραιος
  Η παράμετρος αυτή χρησιμοποιείται για να καθορίσει την τιμή από την οποία ξεκινά η μέτρηση. Δηλαδή τον αριθμό του πρώτου στοιχείου της λίστας στην περίπτωση που δε θέλουμε να είναι το 1 ή το Α. Ανταποκρίνεται σε καθέναν από τους πιο πάνω τύπους μέτρησης.

Πηγαίος κώδικας

```
<B> Φεβρουάριος 2000 </B>
<OL START=14>
 <LI> Δευτέρα
 <LI> Τρίτη
 <LI> Τετάρτη
 <LI> Πέμπτη
 <LI> Παρασκευή
 <LI> Σάββατο
 <LI> Κυριακή
</OL>
```

Εμφάνιση στο διαφυλλιστή

Φεβρουάριος 2000

14. Δευτέρα
15. Τρίτη
16. Τετάρτη
17. Πέμπτη
18. Παρασκευή
19. Σάββατο
20. Κυριακή

Πηγαίος κώδικας

```
ΟΔΗΓΙΕΣ ΠΡΟΣ ΥΠΟΨΗΦΙΟΥΣ
<OL TYPE=I>
 <LI> Διαβάστε καλά όλα τα θέματα
 <LI> Απαντήστε σε όσα από αυτά ξέρετε
 <LI> <b> Βαθμολογική Αξία θεμάτων: </b>
 <OL TYPE=A>
        <LI> Θέμα 30%
        <LI> Θέμα 20%
        <LI> Θέμα 40%
        <LI> Θέμα 10%
                </OL>
</OL>
```

Εμφάνιση στο διαφυλλιστή

ΟΔΗΓΙΕΣ ΠΡΟΣ ΥΠΟΨΗΦΙΟΥΣ

I. Διαβάστε καλά όλα τα θέματα
II. Απαντήστε σε όσα από αυτά ξέρετε
II. **Βαθμολογική Αξία θεμάτων:**
   A. Θέμα 30%
   B. Θέμα 20%
   C. Θέμα 40%
   D. Θέμα 10%

Οι **μη διατεταγμένοι κατάλογοι** (unordered lists - κατάλογοι χωρίς διάταξη) δεν εμφανίζουν τα στοιχεία τους αριθμημένα, αλλά στο αριστερό μέρος τους εμφανίζεται ένα συγκεκριμένο σύμβολο που διαχωρίζει τις εισαγωγές στοιχείων στον κατάλογο. Για να δημιουργήσουμε μία τέτοια λίστα στην HTML, χρησιμοποιούμε τη διπλή ετικέτα **<UL> - </UL>**. Για να εισάγουμε στη μη διατεταγμένη λίστα ένα στοιχείο, χρησιμοποιούμε και εδώ την ετικέτα **<LI>** (List Item). Η παράμετρος της ετικέτας <UL> είναι η επόμενη ενώ έπειτα από αυτή υπάρχει ένα παράδειγμα κατασκευής μη διατεταγμένου καταλόγου:

TYPE="disc | circle | square"
Η παράμετρος αυτή καθορίζει τον τύπο του συμβόλου που χρησιμοποιείται για τη διάκριση των στοιχείων μίας μη διατεταγμένης λίστας. Πιο συγκεκριμένα:

- **TYPE=disc**: Το σύμβολο έχει τη μορφή ενός συμπαγούς κυκλικού δίσκου.
- **TYPE=circle**: Το σύμβολο έχει τη μορφή ενός κύκλου.
- **TYPE=square**: Το σύμβολο έχει τη μορφή ενός συμπαγούς τετραγώνου.

Εάν δεν καθοριστεί κάποιος από τους παραπάνω τύπους, τότε ο πρώτος είναι ο προκαθορισμένος.

Πηγαίος κώδικας

```
<B>Απαραίτητα Συνοδευτικά Έγγραφα </B>
<UL TYPE=CIRCLE>
  <LI> Δελτίο Αστυνομικής Ταυτότητας
  <LI> Απολυτήριο Λυκείου ή Τεχνικής Σχολής
  <LI> Απολυτήριο Στρατού
  <LI> Σύντομο Βιογραφικό
</UL>
```

Εμφάνιση στο διαφυλλιστή

Απαραίτητα Συνοδευτικά Έγγραφα

- Δελτίο Αστυνομικής Ταυτότητας
- Απολυτήριο Λυκείου ή Τεχνικής Σχολής
- Απολυτήριο Στρατού
- Σύντομο Βιογραφικό

Οι **ερμηνευτικοί κατάλογοι** (definition lists) χρησιμοποιούνται όταν επιθυμούμε να παραστήσουμε κείμενο με "λεξικογραφικό" τρόπο, δηλαδή κάθε εγγραφή στον κατάλογο να περιέχει ένα στοιχείο και να ακολουθεί μία επεξήγηση - ορισμός του. Η δημιουργία ενός ερμηνευτικού καταλόγου στην HTML γίνεται με τη χρήση της διπλής ετικέτας **<DL> - </DL>**. Όταν θέλουμε να εισάγουμε σε ερμηνευτικό κατάλογο έναν όρο ο οποίος θα επεξηγηθεί στη συνέχεια, χρησιμοποιούμε την ετικέτα **<DT>** (Definition Term), ενώ για να εισάγουμε την ερμηνεία ενός όρου χρησιμοποιούμε την ετικέτα **<DD>** (Definition Description). Παρακάτω φαίνεται ένα παράδειγμα κατασκευής ερμηνευτικού καταλόγου:

Πηγαίος κώδικας

```
<B>ΙΣΧΥΣ ΣΤΗΝ ΗΜΙΤΟΝΙΚΗ ΜΟΝΙΜΗ ΚΑΤΑΣΤΑΣΗ </B>
<DL>
  <DT>Στιγμιαία Ισχύς
    <DD> Η στιγμιαία ισχύς στην ημιτονική μόνιμη κατάσταση προσδιορίζεται από το γινόμενο της στιγμιαίας τάσης και του στιγμιαίου ρεύματος.
  <DT> Μέση Ισχύς
    <DD> Η μέση ισχύς είναι ίση με το γινόμενο της ενεργούς τιμής της τάσης επί την ενεργό τιμή της έντασης, επί το συνημίτονο της διαφοράς φάσης μεταξύ ρεύματος και τάσης.
  <DT> Φαινόμενη Ισχύς
```

<DD> Η φαινόμενη ισχύς είναι ίση με το γινόμενο της ενεργούς τιμής της τάσης επί την ενεργό τιμή της έντασης.
</DL>

ΙΣΧΥΣ ΣΤΗΝ ΗΜΙΤΟΝΙΚΗ ΜΟΝΙΜΗ ΚΑΤΑΣΤΑΣΗ

Στιγμιαία Ισχύς
: Η στιγμιαία ισχύς στην ημιτονική μόνιμη κατάσταση προσδιορίζεται από το γινόμενο της στιγμιαίας τάσης και του στιγμιαίου ρεύματος.

Μέση Ισχύς
: Η μέση ισχύς είναι ίση με το γινόμενο της ενεργούς τιμής της τάσης επί την ενεργό τιμή της έντασης, επί το συνημίτονο της διαφοράς φάσης μεταξύ ρεύματος και τάσης.

Φαινόμενη Ισχύς
: Η φαινόμενη ισχύς είναι ίση με το γινόμενο της ενεργούς τιμής της τάσης επί την ενεργό τιμή της έντασης.

Σε αυτήν τη διπλωματική εργασία οι ιστοσελίδες δεν είναι γραμμένες με τη γλώσσα HTML αλλά με τη γλώσσα XHTML (Extensible HyperText Markup Language). Ωστόσο οι γλώσσες HTML και XHTML είναι κατά 95 % πανομοιότυπες. Η γλώσσα XHTML είναι απλώς γλώσσα HTML αλλά γραμμένη χρησιμοποιώντας τη γλώσσα XML (Extensible Markup Language). Δηλαδή η γλώσσα XHTML είναι στην πραγματικότητα μία διάλεκτος της γλώσσας XML, με προκαθορισμένα στοιχεία ενσωματωμένα. Αλλά αυτά τα προκαθορισμένα στοιχεία της XML έχουν τα ίδια ονόματα με τα στοιχεία της HTML. Υπάρχει ένα στοιχείο XHTML με όνομα <html> για παράδειγμα και ένα άλλο με όνομα <body> και άλλο ένα με όνομα <head> κ.ο.κ. Έτσι σε ένα πρόγραμμα περιήγησης ένα έγγραφο XHTML μοιάζει σαν ένα απλό έγγραφο HTML πράγμα το οποίο σημαίνει ότι η XHTML θα λειτουργεί με τα σύγχρονα HTML προγράμματα περιήγησης. Παρακάτω φαίνεται το πως πρέπει να γράφεται ο κώδικας σε ένα έγγραφο της γλώσσας XHTML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<META http-equiv="Content-Type" content="text/html; charset=windows-1253">
<title> </title>
</head>
<body>
</body>
</html>
```

Οι πρώτες γραμμές του κώδικα χρησιμεύουν ως ένας τυπικός κώδικας ο οποίος επιβάλλεται να συμπεριλαμβάνεται σε όλα τα έγγραφα της γλώσσας XHTML. Αυτές οι πρώτες γραμμές του κώδικα στην πραγματικότητα προσδιορίζουν τη σελίδα ως ένα έγγραφο της γλώσσας XHTML 1.1 το οποίο σημαίνει τεχνικά πως η ιστοσελίδα είναι μία σελίδα της γλώσσας XHTML. Στη διπλωματική ετούτη εργασία χρησιμοποιήθηκε η γλώσσα XHTML αντί της γλώσσας HTML καθόσον με τη χρήση της XHTML αναπτύσσονται ιστοσελίδες που ακολουθούν τα τελευταία Web πρότυπα. Και αυτό είναι κάτι καλό.

Η γλώσσα HTML έχει συγκεκριμένες ετικέτες και παραμέτρους για την παρουσίαση του περιεχομένου ενός υπερκειμένου. Πολλές φορές ο συγγραφέας χρειάζεται να τροποποιήσει τον τυποποιημένο και δεδομένο αυτό τρόπο παρουσίασης και να το φέρει "στα μέτρα του", δημιουργώντας νέες παρουσιαστικές ιδιότητες που να τον εξυπηρετούν περισσότερο. Τα **φύλλα στυλ** (style sheets) είναι ένας εύχρηστος μηχανισμός για μία προκαθορισμένη και τυποποιημένη από το σχεδιαστή - συγγραφέα διαμόρφωση του τρόπου παρουσίασης ενός υπερκειμένου. Για παράδειγμα, ο συγγραφέας επιθυμεί το κείμενο της επικεφαλίδας 1 (H1) να εμφανίζεται με μπλε πλάγιους (italics) χαρακτήρες, μεγέθους 14 στιγμών. Για αυτό, πρέπει να χρησιμοποιεί τις ιδιότητες "color", "font-size" και "font-style", με τις κατάλληλες τιμές. Και αυτό μπορεί να το δηλώσει ως εξής:

```
H1
{
color:blue;
font-size:14pt;
font-style:italics
}
```

Ανάλογη δήλωση μπορεί να γίνει για την τυποποίηση του στυλ μίας παραγράφου κ.ο.κ. Για να δηλωθούν περισσότερες από μία ιδιότητες σε μία ετικέτα πρέπει να χρησιμοποιηθεί το ελληνικό ερωτηματικό ";" για το διαχωρισμό τους. Ο τρόπος παρουσίασης των φύλλων στυλ αποτελεί μέρος του βασικού μηχανισμού **Cascading Style Sheets** (CSS), δηλαδή των διαδοχικών φύλλων στυλ. Ο μηχανισμός αυτός επιτρέπει στο συγγραφέα μίας ιστοσελίδας να καθορίσει ένα βασικό φύλλο στυλ αλλά και κατά βούληση να μπορεί να το αντικαταστήσει με ένα άλλο αναθέτοντας άλλες ιδιότητες οι οποίες να ικανοποιούν τις ανάγκες του. Η δημιουργία στυλ σε ένα υπερκείμενο γίνεται με τέσσερις τρόπους:

- Χρησιμοποιώντας τη διπλή ετικέτα **<STYLE>** - **</STYLE>** ανάμεσα στις ετικέτες <HEAD> </HEAD>. Για παράδειγμα, εάν θέλουμε να δηλώσουμε ότι οι επικεφαλίδες H1 θα έχουν χρώμα κόκκινο, τότε το δηλώνουμε εσωτερικά στο αρχείο HTML κάνοντας χρήση της ιδιότητας color ως εξής:

```
<HEAD>
<TITLE> ΦΥΛΛΑ ΣΤΥΛ </TITLE>
<STYLE>
H1 {color : red}
</STYLE>
</HEAD>
```

- Χρησιμοποιώντας την ετικέτα **<LINK>** για σύνδεση με τη διεύθυνση ενός εξωτερικού φύλλου στυλ. Η χρήση της ετικέτας γίνεται ανάμεσα στην περιοχή <HEAD> </HEAD> του υπερκειμένου. Τα αρχεία με φύλλα στυλ έχουν κατάληξη ***.css** (*Cascading Style Sheets*) και είναι απλά αρχεία κειμένου (text files). Αυτός είναι ο ιδανικός τρόπος εάν επιθυμούμε να δημιουργήσουμε μία μήτρα (template) για τις ιστοσελίδες μας και να πετύχουμε έτσι μία ομοιομορφία στον τρόπο εμφάνισης. Η σύνταξη της ετικέτας <LINK> είναι:

```
<LINK REL="stylesheet"
 TYPE="text/css"
 HREF="όνομα_αρχείου_στυλ.css">
```

Η παράμετρος **REL** (από το RELationship) δηλώνει ότι ο σύνδεσμος οδηγεί σε ένα αρχείο φύλλου στυλ (stylesheet). Η παράμετρος **TYPE** καθορίζει τον τύπο MIME (MIME type) του αρχείου που περιγράφει τα στυλ και δέχεται την τυποποιημένη τιμή "text/css". Αυτό σημαίνει ότι το συγκεκριμένο αρχείο περιγράφει Cascading Style

Sheet (κατάληξη css) ενώ στην ουσία αποτελεί ένα αρχείο κειμένου (text). Η τιμή της παραμέτρου **HREF** είναι η URL διεύθυνση του αρχείου με το συγκεκριμένο φύλλο στυλ.

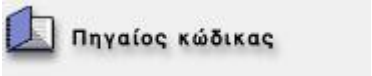
- Μπορούμε επίσης να χρησιμοποιήσουμε το μηχανισμό εισαγωγής **@import** που μοιάζει αρκετά με τον παραπάνω τρόπο. Η οδηγία @import έχει μία παράμετρο, τη **url**, η οποία δέχεται ως τιμή τη URL διεύθυνση του αρχείου με το φύλλο στυλ. Η εισαγωγή ενός αρχείου στυλ που βρίσκεται στη διεύθυνση *"http://www.style.org/ formal_style.css"* γίνεται ως εξής:

```
<HTML>
<HEAD>
<TITLE>Στο αρχείο αυτό χρησιμοποιείται ένα φύλλο στυλ</TITLE>
<STYLE
 TYPE="text/css">
 @import url("http://www.style.org/formal_syle.css")
</STYLE>
</HEAD>
<BODY>
 Το περιεχόμενο της ιστοσελίδας
</BODY>
</HTML>
```

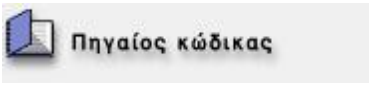
- Τέλος, για την εισαγωγή στυλ έχει πλέον δημιουργηθεί η παράμετρος **STYLE** σε κάθε ετικέτα HTML η οποία μπορεί να μορφοποιεί το περιεχόμενο της. Τοιουτοτρόπως, μπορούμε να δηλώσουμε κάποιο στυλ μορφοποίησης μίας ετικέτας κάνοντας χρήση της παραμέτρου STYLE ως εξής:

```
<H1 STYLE="color: red"> Επικεφαλίδα 1 </H1>
```

Ένας άλλος τρόπος, εκτός από το γενικό κανόνα ο οποίος φαίνεται στον πρώτο τρόπο δημιουργίας στυλ σε ένα υπερκείμενο, για τη δημιουργία συνόλων ιδιοτήτων μορφοποίησης στυλ και εφαρμογής τους κατά βούληση σε οποιαδήποτε ετικέτα του υπερκειμένου δίνεται μέσα από το μηχανισμό διαδοχικών φύλλων στυλ με τη χρήση της κλάσης **CLASS** και του αναγνωριστικού **ID**. Η κλάση **CLASS** είναι μία ομάδα ιδιοτήτων στυλ που μπορούν να αποδοθούν σε μία ετικέτα. Στην περίπτωση εσωτερικής δήλωσης ενός στυλ, η κλάση με τις ιδιότητες μίας ετικέτας περιγράφεται ανάμεσα στην ετικέτα <STYLE>-</STYLE> στην περιοχή <HEAD> ενός υπερκειμένου. Έστω ότι θέλουμε να δηλώσουμε ότι το κείμενο σε μία παράγραφο θα έχει χρώμα κόκκινο και γράμματα πλάγια. Τότε η κλάση "**redtext_italics**" για την ετικέτα <P> δηλώνεται ως εξής:

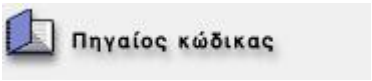



```
<HTML>
<HEAD>
<TITLE> ΦΥΛΛΑ ΣΤΥΛ </TITLE>
<STYLE>
 P.redtext_italics {color:red; font-style:italic}
</STYLE>
</HEAD>
<BODY>
<P> κείμενο χωρίς μορφοποίηση σύμφωνα με την κλάση redtext_italics </P>
<P CLASS="redtext_italics"> κείμενο παραγράφου με το στυλ της κλάσης redtext_italics
</P>
```



<P> κείμενο χωρίς μορφοποίηση σύμφωνα με την κλάση redtext_italics </P>
</BODY>
</HTML>

Πολλές φορές δεν επιθυμούμε μία κλάση να συνδέεται άμεσα με μία ετικέτα αλλά να είναι διαθέσιμη και για διαφορετικές ετικέτες. Στην περίπτωση ετούτη η δήλωση της κλάσης πραγματοποιείται χρησιμοποιώντας μονάχα την τελεία (.) και το όνομα της κλάσης, απορρίπτοντας το όνομα της ετικέτας. Στο παρακάτω παράδειγμα δηλώνουμε μία ανεξάρτητη κλάση η οποία καλείται στη συνέχεια από τις ετικέτες <H1>, <P> και <ADDRESS> και διαμορφώνει το χρώμα και το στυλ των γραμμάτων (τα εμφανίζει υπογραμμισμένα με μπλε χρώμα).



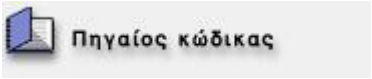
<HTML>
<HEAD>
<TITLE> ΦΥΛΛΑ ΣΤΥΛ </TITLE>
<STYLE>
.blue_underlined {color:blue; text-decoration:underline}
</STYLE>
</HEAD>
<BODY>
<H1 **CLASS="blue_underlined"**> Επικεφαλίδα με μορφοποίηση </H1>
<P **CLASS="blue_underlined"**> Παράγραφος με μορφοποίηση </P>
<P> Παράγραφος χωρίς μορφοποίηση </P>
<ADDRESS **CLASS="blue_underlined"**> Ετικέτα Address με μορφοποίηση
</ADDRESS>
</BODY>
</HTML>

Αντί για τη δήλωση κλάσεων, έχουμε τη δυνατότητα να δηλώσουμε κάποια μοναδικά χαρακτηριστικά **ID** (unique identifiers). Η δήλωση ενός χαρακτηριστικού γίνεται με τη χρήση του ειδικού χαρακτήρα "#" πριν το όνομα του, όπως αντίστοιχα χρησιμοποιήσαμε την τελεία "." για τις κλάσεις. Η χρήση του αναγνωριστικού μίας ετικέτας γίνεται στη δήλωση της, χρησιμοποιώντας την παράμετρο ID συνοδευόμενη με το όνομα του συγκεκριμένου αναγνωριστικού. Παρακάτω φαίνεται η δημιουργία του αναγνωριστικού redtext_italics και η χρήση του:

<HTML>
<HEAD>
<TITLE> ΦΥΛΛΑ ΣΤΥΛ </TITLE>
<STYLE>
#redtext_italics {color:red; font-style:italic}
</STYLE>
</HEAD>
<BODY>
<P> κείμενο χωρίς μορφοποίηση </P>
<P **ID="redtext_italics"**> κείμενο παραγράφου διαμορφωμένο με το αναγνωριστικό redtext_italics </P>
<P> κείμενο χωρίς μορφοποίηση </P>
</BODY>
</HTML>

Μία άλλη χρήση του αναγνωριστικού ID είναι η δημιουργία εξαιρέσεων σε σχέση με

μία προδηλωμένη κλάση. Αυτό μπορεί να εφαρμοστεί γιατί ένα αναγνωριστικό έχει αρκετή ισχύ ώστε να ακυρώνει με την παρουσία του τις όποιες ιδιότητες είχε επιβάλλει προηγουμένως η χρήση κάποιας κλάσης. Με τον τρόπο αυτό δε χρειάζεται να ξαναγράψουμε κλάσεις οι οποίες μπορεί να έχουν το ίδιο σώμα και να διαφέρουν σε λίγες μόνο ιδιότητες, αλλά γράφουμε μία κλάση που περιέχει τα περισσότερα κοινά χαρακτηριστικά και έπειτα δημιουργούμε αναγνωριστικά (IDs) που περιγράφουν τις ιδιότητες που θέλουμε να διαφοροποιήσουμε. Στο παρακάτω παράδειγμα επιδιώκουμε όλες οι παράγραφοι να εμφανίζονται με γκρίζο φόντο και πλάγια γραφή, αλλά δε θέλουμε όλες να έχουν το ίδιο χρώμα χαρακτήρων, που όπως δηλώνεται στην κύρια κλάση, είναι το μαύρο.




```
<HTML>
<HEAD>
<TITLE> ΦΥΛΛΑ ΣΤΥΛ </TITLE>
<STYLE>
.main
{
   display:block;
   background-color:#D0D0D0;
   font-style:italic;
   color:black;
   padding: 5 5 5 5;
    }
#blue_color {color:blue;}
#red_color {color:red;}
#white_color {color:white;}
</STYLE>
</HEAD>
<BODY>
<P> Παράγραφος χωρίς μορφοποίηση </P>
<P CLASS="main">
Παράγραφος με την κύρια μορφοποίηση της κλάσης main </P>
<P CLASS="main" ID="blue_color">
Παράγραφος με τη μορφοποίηση της κλάσης main και του αναγνωριστικού blue_color που εμφανίζει τους χαρακτήρες μπλε. </P>
<P CLASS="main" ID="red_color">
Παράγραφος με τη μορφοποίηση της κλάσης main και του αναγνωριστικού red_color που εμφανίζει τους χαρακτήρες κόκκινους. </P>
<P CLASS="main" ID="white_color">
Παράγραφος με τη μορφοποίηση της κλάσης main και του αναγνωριστικού white_color που εμφανίζει τους χαρακτήρες λευκούς. </P>
<P ID="blue_color">
Παράγραφος με τη μορφοποίηση του αναγνωριστικού blue_color που εμφανίζει τους χαρακτήρες μπλε. </P>
</BODY>
</HTML>
```

Πολλές φορές είναι απαραίτητο να εφαρμοστεί ένα στυλ μορφοποίησης σε ένα τμήμα του περιεχομένου μίας ετικέτας, π.χ. σε ορισμένες λέξεις ή ακόμα και γραμμές κειμένου. Η διπλή ετικέτα **<SPAN>** παρέχει τη λειτουργικότητα να μπορεί να εφαρμοστεί στυλ μορφοποίησης σε τμήμα περιεχομένου για το οποίο δεν υπάρχει τρόπος να οριοθετηθεί. Η ετικέτα <SPAN> έχει σχεδιαστεί για το σκοπό αυτό και μόνο, δηλαδή να εισάγει κάποιο στυλ μορφοποίησης στο τμήμα του περιεχομένου που περικλείει, το οποίο

μπορεί να είναι υποσύνολο του περιεχομένου μίας άλλης ετικέτας. Η απόδοση χαρακτηριστικών διαμόρφωσης στυλ στην ετικέτα <SPAN> - </SPAN> γίνεται με τη χρήση κάποιων από τους τρόπους που έχουν περιγραφεί. Για παράδειγμα, εάν θέλουμε να τονίσουμε με κόκκινο χρώμα τους χαρακτήρες μίας λέξης σε μία επικεφαλίδα ή μερικές λέξεις σε μία παράγραφο, τότε χρησιμοποιούμε την ετικέτα SPAN ως εξής:

```
<HTML>
<HEAD>
<TITLE> ΧΡΗΣΗ ΤΗΣ ΕΤΙΚΕΤΑΣ SPAN </TITLE>
<STYLE>
  .redtext {color:red}
</STYLE>
</HEAD>
<BODY>
<H2> Η επικεφαλίδα με μία λέξη σε <SPAN CLASS="redtext"> κόκκινο </SPAN> χρώμα
</H2>
<P>
 Παράγραφος με <SPAN CLASS="redtext"> κόκκινους χαρακτήρες </SPAN>
</P>
<P> Υπόλοιπο κείμενο </P>
</BODY>
</HTML>
```

Οι ιδιότητες μορφοποίησης στυλ είναι οι εξής:

- height
  Χρήση: Ορίζει το ύψος ενός στοιχείου.
  Τιμές: auto, length, %.

- line-height
  Χρήση: Ορίζει την απόσταση μεταξύ γραμμών και στοιχείων.
  Τιμές: normal, length, %.

- max-height
  Χρήση: Ορίζει το μέγιστο ύψος ενός στοιχείου.
  Τιμές: none, length, %.

- min-height
  Χρήση: Ορίζει το ελάχιστο ύψος ενός στοιχείου.
  Τιμές: length, %.

- max-width
  Χρήση: Ορίζει το μέγιστο πλάτος ενός στοιχείου.
  Τιμές: none, length, %.

- min-width
  Χρήση: Ορίζει το ελάχιστο πλάτος ενός στοιχείου.
  Τιμές: length, %.

- width
  Χρήση: Ορίζει το πλάτος ενός στοιχείου.
  Τιμές: auto, length, %.

- color
  Χρήση: Ορίζει το χρώμα του κειμένου.
  Τιμές: color.

- direction
  Χρήση: Ορίζει την κατεύθυνση του κειμένου όπως από αριστερά προς τα δεξιά ή από τα δεξιά προς τα αριστερά.
  Τιμές: ltr, rtl.

- font
  Χρήση: Μία συντόμευση ιδιότητας που μας επιτρέπει να ορίζουμε όλες τις ιδιότητες γραμματοσειρών σε μία δήλωση.
  Τιμές: font-style, font-variant, font-weight, font-size / line-height, font-family.

- font-family
  Χρήση: Μία λίστα προτεραιοτήτων με ονόματα οικογενειών γραμματοσειρών ή / και γενικά ονόματα για ένα στοιχείο.
  Τιμές: family-name, generic-family.
- font-size
  Χρήση: Ορίζει το μέγεθος μίας γραμματοσειράς.
  Τιμές: xx-small, x-small, small, medium, large, xlarge, xx-large, smaller, larger, length, %.

- font-style
  Χρήση: Ορίζει το στυλ της γραμματοσειράς.
  Τιμές: normal, italic, oblique.

- font-variant
  Χρήση: Εμφανίζει κείμενο σε μία γραμματοσειρά μικρών κεφαλαίων ή σε μία κανονική γραμματοσειρά.
  Τιμές: normal, small-caps.

- font-weight
  Χρήση: Ορίζει το πάχος μίας γραμματοσειράς.
  Τιμές: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900.

- letter-spacing
  Χρήση: Αυξάνει ή μειώνει το διάστημα μεταξύ χαρακτήρων του κειμένου.
  Τιμές: normal, length.

- text-align
  Χρήση: Στοιχίζει το κείμενο μέσα σε ένα στοιχείο.
  Τιμές: left, right, center, justify.

- text-decoration
  Χρήση: Εφαρμόζει μία διακόσμηση σε κείμενο.
  Τιμές: none, underline, overline, line-through, blink.

- text-indent
  Χρήση: Βάζει σε εσοχή την πρώτη γραμμή του κειμένου ενός στοιχείου.
  Τιμές: length, %.

- text-transform
  Χρήση: Ελέγχει τα κεφαλαία - πεζά των γραμμάτων του κειμένου.
  Τιμές: none, capitalize, uppercase, lowercase.

- white-space
  Χρήση: Ορίζει το χειρισμό των κενών μέσα σε ένα στοιχείο.
  Τιμές: normal, pre, nowrap.

- word-spacing
  Χρήση: Αυξάνει ή μειώνει το χώρο μεταξύ λέξεων.
  Τιμές: normal, length.

- background
  Χρήση: Μία συντομευμένη ιδιότητα που μας επιτρέπει να ορίζουμε όλες τις ιδιότητες του φόντου σε μία δήλωση.
  Τιμές: background-color, background - image, background - repeat, background-attachment, background-position.

- background-attachment
  Χρήση: Προσδιορίζει εάν μία εικόνα φόντου είναι σταθερή ή κυλά μαζί με τη σελίδα.
  Τιμές: scroll, fixed.

- background-color
  Χρήση: Ορίζει το χρώμα του φόντου ενός στοιχείου.
  Τιμές: color-rgb, color-hex, color-name, transparent.

- background-image
  Χρήση: Ορίζει μία εικόνα ως φόντο.
  Τιμές: url, none.

- background-position
  Χρήση: Ορίζει την αρχική θέση μίας εικόνας φόντου.
  Τιμές: top left, top center, top right, center left, center center, center right, bottom left, bottom center, bottom right, x-%, y-%, x-pos, y-pos.

- background-repeat
  Χρήση: Ορίζει εάν και πως επαναλαμβάνεται μία εικόνα φόντου.
  Τιμές: repeat, repeat-x, repeat-y, no-repeat.
- border
  Χρήση: Μία συντομευμένη ιδιότητα που μας επιτρέπει να ορίσουμε όλες τις ιδιότητες για τα τέσσερα περιγράμματα σε μία δήλωση.
  Τιμές: border-width, border-style, border-color.

- border-bottom
  Χρήση: Μία συντομευμένη ιδιότητα που μας επιτρέπει να ορίσουμε όλες τις ιδιότητες για το κάτω περίγραμμα σε μία δήλωση.
  Τιμές: border-bottom-width, border-bottom-style, border-bottom-color.

- border-bottom-style
  Χρήση: Ορίζει το στυλ του κάτω περιγράμματος.
  Τιμές: border-color.

- border-bottom-width
  Χρήση: Ορίζει το πλάτος του κάτω περιγράμματος.
  Τιμές: thin, medium, thick, length.

- border-color
  Χρήση: Ορίζει το χρώμα των τεσσάρων περιγραμμάτων.
  Τιμές: color.
  Σημειώσεις: Μπορεί να καθοριστεί χρησιμοποιώντας από ένα έως τέσσερα

χρώματα.

- border-left
  Χρήση: Μία συντομευμένη ιδιότητα που μας επιτρέπει να ορίσουμε όλες τις ιδιότητες του αριστερού περιγράμματος σε μία δήλωση.
  Τιμές: border-left-width, border-left-style, border-left-color.

- border-left-width
  Χρήση: Ορίζει το πλάτος του αριστερού περιγράμματος.
  Τιμές: thin, medium, thick, length.

- border-left-style
  Χρήση: Ορίζει το στυλ του αριστερού περιγράμματος.
  Τιμές: border-style.

- border-left-color
  Χρήση: Ορίζει το χρώμα του αριστερού περιγράμματος.
  Τιμές: border-color.

- border-right
  Χρήση: Μία συντομευμένη ιδιότητα που μας επιτρέπει να ορίσουμε όλες τις ιδιότητες του δεξιού περιγράμματος σε μία δήλωση.
  Τιμές: border-right-width, border-right-style, border-right-color.

- border-right-color
  Χρήση: Ορίζει το χρώμα του δεξιού περιγράμματος.
  Τιμές: border-color.

- border-right-style
  Χρήση: Ορίζει το στυλ του δεξιού περιγράμματος.
  Τιμές: border-style.

- border-right-width
  Χρήση: Ορίζει το πλάτος του δεξιού περιγράμματος.
  Τιμές: thin, medium, thick, length.

- border-style
  Χρήση: Ορίζει το στυλ των τεσσάρων περιγραμμάτων.
  Τιμές: none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset.
  Σημειώσεις: Μπορεί να καθοριστεί χρησιμοποιώντας από ένα έως τέσσερα στυλ.

- border-top
  Χρήση: Μία συντομευμένη ιδιότητα που μας επιτρέπει να ορίσουμε όλες τις ιδιότητες του πάνω περιγράμματος σε μία δήλωση.
  Τιμές: border-top-width, border-top-style, border-top-color.

- border-top-color
  Χρήση: Ορίζει το χρώμα του πάνω περιγράμματος.
  Τιμές: border-color.

- border-top-style
  Χρήση: Ορίζει το στυλ του πάνω περιγράμματος.
  Τιμές: border-syle.

- border-top-width
  Χρήση: Ορίζει το πλάτος του πάνω περιγράμματος.
  Τιμές: thin, medium, thick, length.

- border-width
  Χρήση: Μία συντομευμένη ιδιότητα για τον ορισμό του πλάτους των τεσσάρων περιγραμμάτων σε μία δήλωση.
  Τιμές: thin, medium, thick, length.
  Σημειώσεις: Μπορεί να καθοριστεί χρησιμοποιώντας ένα έως τέσσερα πλάτη.

- margin
  Χρήση: Μία συντομευμένη ιδιότητα που μας επιτρέπει να ορίσουμε όλες τις ιδιότητες των περιθωρίων σε μία δήλωση.
  Τιμές: margin-top, margin-right, margin-bottom, margin-left.

- margin-bottom
  Χρήση: Ορίζει το κάτω περιθώριο ενός στοιχείου.
  Τιμές: auto, length, %.

- margin-left
  Χρήση: Ορίζει το αριστερό περιθώριο ενός στοιχείου.
  Τιμές: auto, length, %.
- margin-right
  Χρήση: Ορίζει το δεξιό περιθώριο ενός στοιχείου.
  Τιμές: auto, length, %.

- margin-top
  Χρήση: Ορίζει το πάνω περιθώριο ενός στοιχείου.
  Τιμές: auto, length, %.

- padding
  Χρήση: Μία συντομευμένη ιδιότητα που μας επιτρέπει να ορίσουμε όλες τις ιδιότητες αναπλήρωσης σε μία δήλωση.
  Τιμές: padding-top, padding-right, padding-bottom, padding-left.

- padding-bottom
  Χρήση: Ορίζει την κάτω αναπλήρωση ενός στοιχείου.
  Τιμές: length, %.

- padding-left
  Χρήση: Ορίζει την αριστερή αναπλήρωση ενός στοιχείου.
  Τιμές: length, %.

- padding-right
  Χρήση: Ορίζει τη δεξιά αναπλήρωση ενός στοιχείου.
  Τιμές: length, %.

- padding-top
  Χρήση: Ορίζει την πάνω αναπλήρωση ενός στοιχείου.
  Τιμές: length, %.

- bottom
  Χρήση: Ορίζει τη μετατόπιση μεταξύ της κάτω άκρης του στοιχείου και της κάτω άκρης του γονέα του στοιχείου.
  Τιμές: auto, length, %.

- clear
  Χρήση: Προσδιορίζει τις πλευρές ενός στοιχείου εκεί όπου δεν επιτρέπονται άλλα κινητά στοιχεία.
  Τιμές: left, right, both, none.

- clip
  Χρήση: Ορίζει το σχήμα ενός στοιχείου.
  Τιμές: auto, shape.
  Σημειώσεις: Το στοιχείο αποκόπτεται σε αυτό το σχήμα όταν εμφανίζεται.

- cursor
  Χρήση: Καθορίζει τον τύπο του δρομέα του ποντικιού που θα εμφανίζεται.
  Τιμές: url, auto, crosshair, default, pointer, move, e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize, text, wait, help.

- display
  Χρήση: Ορίζει εάν και πως εμφανίζεται ένα στοιχείο.
  Τιμές: none, inline, block, list-item, run-in, compact, marker, table, inline-table, table - row - group, table - header - group, table - footer - group, table - row, table-column-group, table-column, table-cell, table-caption.

- float
  Χρήση: Ορίζει που θα εμφανιστεί μία εικόνα ή ένα κείμενο σε σχέση με ένα άλλο στοιχείο.
  Τιμές: left, right, none.

- left
  Χρήση: Ορίζει τη μετατόπιση μεταξύ της αριστερής άκρης του στοιχείου και της αριστερής άκρης του γονικού στοιχείου του.
  Τιμές: auto, length, %.

- overflow
  Χρήση: Προσδιορίζει τι συμβαίνει εάν τα περιεχόμενα ενός στοιχείου υπερχειλίσουν στην περιοχή τους.
  Τιμές: auto, visible, hidden, scroll.

- position
  Χρήση: Καθορίζει τη διάταξη ενός στοιχείου ως προς το αν χρησιμοποιεί στατική, σχετική, απόλυτη ή σταθερή θέση.
  Τιμές: static, relative, absolute, fixed.

- right
  Χρήση: Ορίζει τη μετατόπιση μεταξύ της δεξιάς άκρης του στοιχείου και της δεξιάς άκρης του γονικού στοιχείου του.
  Τιμές: auto, length, %.

- top
  Χρήση: Ορίζει τη μετατόπιση μεταξύ της πάνω άκρης του στοιχείου και της πάνω άκρης του γονικού του στοιχείου.
  Τιμές: auto, length, %.

- vertical-align
  Χρήση: Ορίζει την κάθετη στοίχιση ενός στοιχείου.
  Τιμές: baseline, sub, super, top, text-top, middle, bottom, text-bottom, length, %.

- visibility
  Χρήση: Προσδιορίζει εάν ένα στοιχείο θα πρέπει να φαίνεται (ορατό) ή να είναι κρυφό (αόρατο).
  Τιμές: visible, hidden, collapse.

- z-δείκτης
  Χρήση: Ορίζει τον z-δείκτη (σειρά στη στοίβα) ενός στοιχείου.

Τιμές: auto, number.

- list-style
  Χρήση: Μία συντομευμένη ιδιότητα που μας επιτρέπει να ορίσουμε όλες τις ιδιότητες μίας λίστας σε μία δήλωση.
  Τιμές: list-style-type, list-style-position, list-style-image.

- list-style-image
  Χρήση: Ορίζει μία εικόνα ως το δείκτη των στοιχείων μίας λίστας (κουκκίδα) για τη λίστα.
  Τιμές: none, url.

- list-style-position
  Χρήση: Ορίζει που θα τοποθετηθεί ο δείκτης των στοιχείων μίας λίστας (κουκκίδα) στη λίστα.
  Τιμές: inside, outside.

- list-style-type
  Χρήση: Ορίζει τον τύπο του δείκτη των στοιχείων μίας λίστας (κουκκίδα).
  Τιμές: none, disc, circle, square, decimal, decimal - leading - zero, lower - roman, upper-roman, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, hebrew, armenian, georgian, cjk - ideographic, hiragana, katakana, hiragana - iroha, katakana-iroha.

- border-collapse
  Χρήση: Ορίζει το προτυπο μοντέλο περιγράμματος ενός πίνακα.
  Τιμές: collapse, separate.

- border-spacing
  Χρήση: Ορίζει την απόσταση μεταξύ των περιγραμμάτων γειτονικών κελιών.
  Τιμές: length length.

- caption-side
  Χρήση: Ορίζει τη θέση της λεζάντας σε σχέση με τον πίνακα.
  Τιμές: top, bottom, left, right.

- empty-cells
  Χρήση: Προσδιορίζει εάν τα κελιά χωρίς ορατά περιεχόμενα θα πρέπει να έχουν περιγράμματα.
  Τιμές: show, hide.

- table-layout
  Χρήση: Προσδιορίζει πως θα διαταχθεί ο πίνακας.
  Τιμές: auto, fixed.
  Σημειώσεις: Επιταχύνει την απόδοση του browser για σταθερού μεγέθους πίνακες εάν το ορίσουμε σε fixed.

3. ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ ΤΟΥ APPLET

Σε ετούτο το κεφάλαιο θα αναλυθεί ο σχεδιασμός του προγράμματος για ένα κομμάτι ενός Applet της διπλωματικής αυτής εργασίας. Πιο συγκεκριμένα θα αναφερθούν όλοι οι πιθανοί τρόποι με τους οποίους ο χρήστης μπορεί να μεταβεί από μία κατάσταση ερωτήσεων για ένα μηχανολογικό σχέδιο στην επόμενη κατάσταση επερωτήσεων για το επόμενο μηχανολογικό σχέδιο. Θα παρουσιαστεί ο κώδικας του προγράμματος για ετούτο το απόσπασμα του κώδικα του Applet ενώ τα σχόλια θα βρίσκονται μέσα στο σύμβολο '//'. Με το εν λόγω σύμβολο ο προγραμματιστής δύναται να γράψει σχόλια μίας γραμμής μέσα στον κώδικα της γλώσσας Java. Αν τα σχόλια είναι περισσότερα από μία γραμμές τότε ο προγραμματιστής μπορεί να συμπεριλάβει τα σχόλια ανάμεσα στα σύμβολα '/*' και '*/'. Παρακάτω φαίνεται όλος ο κώδικας αυτού του Applet εκτός από τον κώδικα της συνάρτησης sunartisi:

```
package firstApplet;

import javax.swing.*;                                   // Εδώ δηλώνονται τα πακέτα.
import java.awt.*;
import java.awt.event.*;
import javax.swing.border.*;
import javax.swing.border.LineBorder.*;

public class firstApplet extends JApplet
{                              /* Εδώ δηλώνονται αντικείμενα κλάσεων καθώς και οι ακέραιοι i, j.*/
private Font nikos1,nikos2,nikos3,nikos4,nikos5;
private JLabel lable1,lable2,lable3,lable4,lable5,lable6,lable7,lable8;
private JLabel lable9,lable10,lable11,lable12,lable13,lable14,lable15,lable16;
private JLabel lable17,lable18,lable19,lable20,lable21,lable22,lable23;
private JLabel lable24,lable25,lable26,lable27,lable28,lable29,lable30,lable31,lable32;
private JLabel lable33,lable34,lable35,lable36,lable37,lable38,lable39,lable40,lable41;
private Color kostas,kostas2,kostas3,kostas4;
private                                                                    ImageIcon
picture1,picture2,picture3,picture4,picture5,picture6,picture7,picture8,picture9;
private ImageIcon picture10,picture11,picture12,picture13,picture14,picture15,picture16;
private ButtonGroup group;
private JRadioButton rbutton1,rbutton2,rbutton3,rbutton4,rbutton5,rbutton6;
private JButton button1,button2,button3,button4,button5;
private LineBorder thickBorder,thickBorder2,thickBorder3,thickBorder4;
private buttonListener blistener;
private int i,j;

public void init()
{
Container john=getContentPane();
john.setLayout(null);
kostas=new Color(204,204,204);          // Εδώ δηλώνεται το χρώμα που θα έχει ολόκληρη
john.setBackground(kostas);                                          // η μικροεφαρμογή.
j=0;
i=0;
picture1=new ImageIcon("q1.jpg");                          // Εδώ φορτώνονται οι εικόνες
picture3=new ImageIcon("q2.jpg");                                  // της μικροεφαρμογής
picture4=new ImageIcon("q3.jpg");                        // που είναι μηχανολογικά σχέδια.
picture5=new ImageIcon("q4.jpg");
picture6=new ImageIcon("q5.jpg");
picture7=new ImageIcon("q6.jpg");
picture8=new ImageIcon("q7.jpg");
picture9=new ImageIcon("q8.jpg");
```

```
picture10=new ImageIcon("q9.jpg");
picture11=new ImageIcon("q10.jpg");
picture12=new ImageIcon("q11.jpg");
picture13=new ImageIcon("q12.jpg");
picture14=new ImageIcon("q13.jpg");
picture15=new ImageIcon("q14.jpg");
picture16=new ImageIcon("q15.jpg");
lable1=new JLabel(picture1); // Εδώ αρχικοποιούμε το αντικείμενο lable1 της τάξης JLabel
lable1.setBounds(-2,100,765,535);                    // να περιέχει μία από τις εικόνες
                                                     // που είναι μηχανολογικά σχέδια και
john.add(lable1);                    // τοποθετείται σε ένα συγκεκριμένο σημείο του Applet.
picture2=new ImageIcon("M3.gif");
lable2=new JLabel(picture2); // Εδώ αρχικοποιούμε το αντικείμενο lable2 της τάξης JLabel
lable2.setBounds(30,20,290,60);                      // να περιέχει μία εικόνα η
john.add(lable2);          // οποία δείχνει το σήμα του εργαστηρίου Μ3 και τοποθετείται σε
                                                     // συγκεκριμένο σημείο της μικροεφαρμογής.
lable3=new JLabel("Draw/m3");                        // Εδώ αρχικοποιούμε το αντικείμενο
nikos1=new Font("SansSerif",Font.BOLD,34);       // lable3 της κλάσης JLabel να περιέχει
lable3.setFont(nikos1);       // μία συμβολοσειρά με συγκεκριμένο όνομα, στυλ και μέγεθος
lable3.setBounds(500,15,150,40);          // γραμματοσειράς. Το αντικείμενο lable3 ομοίως
john.add(lable3);                // τοποθετείται σε συγκεκριμένο σημείο της μικροεφαρμογής.
lable4=new JLabel("- Έλεγχος γνώσεων Μηχανολογικής Σχεδίασης");
nikos2=new Font("SansSerif",Font.PLAIN,24);
lable4.setFont(nikos2);
lable4.setBounds(650,15,500,40);
john.add(lable4);
lable5=new JLabel("Κεφάλαιο 2:");
lable5.setFont(nikos2);
lable5.setBounds(935,45,200,40);
john.add(lable5);
lable6=new JLabel("ΟΨΕΙΣ");         // Εδώ αρχικοποιούμε το αντικείμενο lable6 της τάξης
nikos3=new Font("SansSerif",Font.BOLD,24);           // JLabel να έχει μία
lable6.setFont(nikos3);                  // συμβολοσειρά με συγκεκριμένο όνομα, στυλ
kostas2=Color.red;                                   // και μέγεθος γραμματοσειράς. Ακόμη
lable6.setForeground(kostas2);         // για αυτήν τη συμβολοσειρά θέτουμε το χρώμα της
lable6.setBounds(1075,45,200,40);                    // να είναι το κόκκινο.
john.add(lable6);
lable7=new JLabel("Απάντηση:");
lable7.setFont(nikos3);
lable7.setBounds(794,119,200,50);
john.add(lable7);
blistener=new buttonListener();          // Εδώ αρχικοποιούμε τον ακροατή των κουμπιών.
group=new ButtonGroup();
rbutton1=new JRadioButton("",false);             // Εδώ αρχικοποιούμε ένα ραδιοπλήκτρο
rbutton1.setBounds(833,180,25,35);      // να μην περιέχει καμία συμβολοσειρά και να μην
group.add(rbutton1);                // είναι αρχικά επιλεγμένο. Το ραδιοπλήκτρο το βάζουμε
μέσα rbutton1.setBackground(kostas);             // σε μία ομάδα και το τοποθετούμε σε μία
rbutton1.setFocusable(false);                // συγκεκριμένη θέση στη μικροεφαρμογή ενώ
θέτουμε rbutton1.addActionListener(blistener);  // το παρασκηνιακό χρώμα του κουμπιού
john.add(rbutton1);                                  // ίδιο με το χρώμα που έχουμε θέσει για
                                                     // ολόκληρη τη μικροεφαρμογή. Επίσης έχουμε αφαιρέσει την
                                                     // εστίαση από το ραδιοπλήκτρο και το έχουμε συνδέσει με τον ακροατή
                                                     // του κουμπιού. Τα ίδια ακριβώς έχουν γίνει και για τα
                                                     // άλλα 5 ραδιοπλήκτρα.
lable32=new JLabel("Πρόοψη"); // Εδώ αρχικοποιούμε ένα αντικείμενο lable32 της κλάσης
lable32.setFont(nikos2);            // JLabel να περιέχει μία συμβολοσειρά με συγκεκριμένο
lable32.setBounds(861,180,245,35);  // όνομα, στυλ και μέγεθος γραμματοσειράς. Αυτό το
```

```
john.add(lable32);              // αντικείμενο τοποθετείται δεξιά από το πρώτο ραδιοπλήκτρο.
rbutton2=new JRadioButton("",false);     // Τα ίδια ισχύουν και για τα αντικείμενα της τάξης
rbutton2.setBounds(833,220,25,35);          // JLabel με ονόματα lable33, lable34, lable35,
group.add(rbutton2);   // lable36 και lable37. Το αντικείμενο lable33 τοποθετείται δεξιά από
rbutton2.setBackground(kostas);     // το δεύτερο ραδιοπλήκτρο, το αντικείμενο lable34
rbutton2.setFocusable(false);                    // δεξιά από το τρίτο ραδιοπλήκτρο κ.ο.κ.
rbutton2.addActionListener(blistener);
john.add(rbutton2);
lable33=new JLabel("Κάτοψη");
lable33.setFont(nikos2);
lable33.setBounds(861,220,245,35);
john.add(lable33);
rbutton3=new JRadioButton("",false);
rbutton3.setBounds(833,260,25,35);
group.add(rbutton3);
rbutton3.setBackground(kostas);
rbutton3.setFocusable(false);
rbutton3.addActionListener(blistener);
john.add(rbutton3);
lable34=new JLabel("Αριστερή πλάγια όψη");
lable34.setFont(nikos2);
lable34.setBounds(861,260,245,35);
john.add(lable34);
rbutton4=new JRadioButton("",false);
rbutton4.setBounds(833,300,25,35);
group.add(rbutton4);
rbutton4.setBackground(kostas);
rbutton4.setFocusable(false);
rbutton4.addActionListener(blistener);
john.add(rbutton4);
lable35=new JLabel("Δεξιά πλάγια όψη");
lable35.setFont(nikos2);
lable35.setBounds(861,300,245,35);
john.add(lable35);
rbutton5=new JRadioButton("",false);
rbutton5.setBounds(833,340,25,35);
group.add(rbutton5);
rbutton5.setBackground(kostas);
rbutton5.setFocusable(false);
rbutton5.addActionListener(blistener);
john.add(rbutton5);
lable36=new JLabel("Άνοψη");
lable36.setFont(nikos2);
lable36.setBounds(861,340,245,35);
john.add(lable36);
rbutton6=new JRadioButton("",false);
rbutton6.setBounds(833,380,25,35);
group.add(rbutton6);
rbutton6.setBackground(kostas);
rbutton6.setFocusable(false);
rbutton6.addActionListener(blistener);
john.add(rbutton6);
lable37=new JLabel("Πίσω όψη");
lable37.setFont(nikos2);
lable37.setBounds(861,380,245,35);
john.add(lable37);
kostas3=Color.white;
```

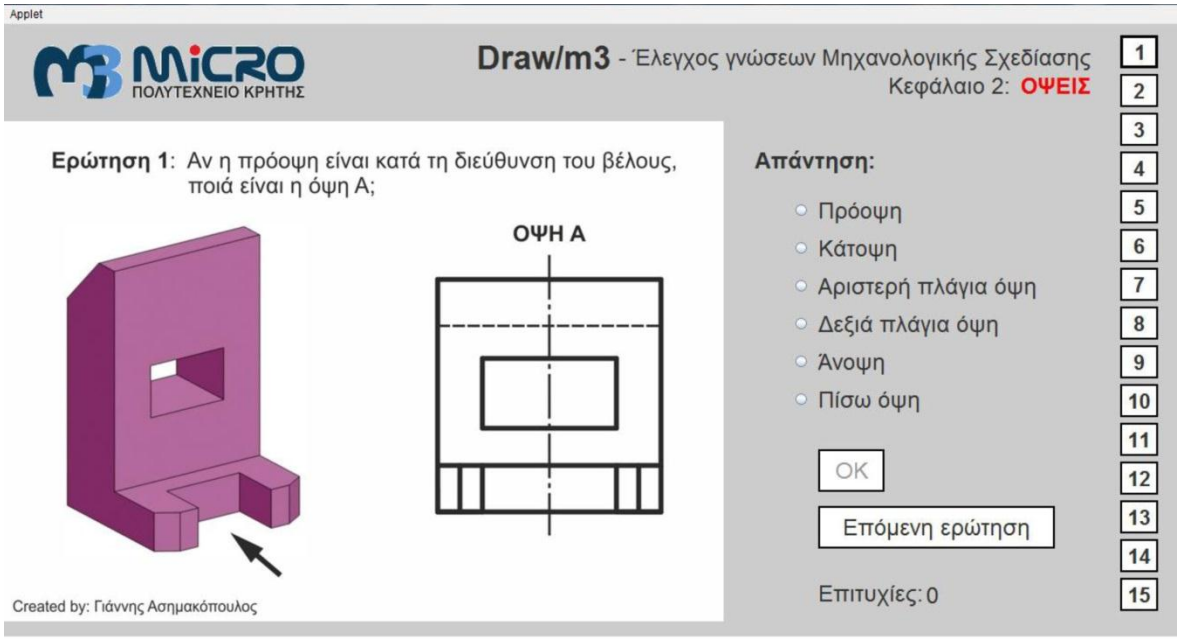
```
thickBorder = new LineBorder(Color.black,2);          // Εδώ αρχικοποιούμε ένα αντικείμενο
      // thickBorder της τάξης LineBorder η οποία ανήκει στο πακέτο javax.swing.border και
         // είναι υποκλάση της τάξης Border. Στην αρχικοποίηση ετούτου του αντικειμένου το
                                  // χρώμα του είναι το μαύρο και έχει κάποιο συγκεκριμένο πάχος.
button1=new JButton("OK");          // Εδώ αρχικοποιούμε το αντικείμενο button1 της τάξης
button1.setBounds(861,450,70,45);                    // JButton να περιέχει μία συμβολοσειρά
button1.setFont(nikos2);       // με συγκεκριμένο όνομα, στυλ και μέγεθος γραμματοσειράς.
button1.setBackground(kostas3);                              // Το παρασκηνιακό χρώμα ετούτου
του button1.setBorder(thickBorder);          // παραλληλόγραμμου κουμπιού είναι το άσπρο.
button1.setFocusable(false);    // Ακόμη έχουμε αφαιρέσει την εστίαση από το κουμπί ενώ
button1.setEnabled(false);                    // το έχουμε απενεργοποιημένο και συνδεδεμένο
button1.addActionListener(blistener); // με τον ακροατή των κουμπιών. Το εν λόγω κουμπί
john.add(button1);                 // τοποθετείται σε συγκεκριμένο σημείο της μικροεφαρμογής.
                 // Επιπρόσθετα στο κουμπί θέτουμε ένα συγκεκριμένο πάχος περιγράμματος.
button2=new JButton("Επόμενη ερώτηση");              // Τα ίδια ακριβώς ισχύουν και για τα
button2.setBounds(861,510,250,45);   // αντικείμενα της κλάσης JButton button2, button3,
button2.setFont(nikos2);        // button4 και button5. Η μόνη διαφορά είναι ότι τα κουμπιά
button2.setBackground(kostas3);                 // button3,button4 και button5 δεν είναι ορατά
button2.setBorder(thickBorder);                                    // ούτε απενεργοποιημένα και
button2.setEnabled(true);                                  // ότι το button2 είναι ενεργοποιημένο.
button2.setFocusable(false);
button2.addActionListener(blistener);
john.add(button2);
button3=new JButton("Ξανά");
button3.setBounds(1021,580,90,45);
button3.setFont(nikos2);
button3.setBackground(kostas3);
button3.setBorder(thickBorder);
button3.setFocusable(false);
button3.addActionListener(blistener);
button3.setVisible(false);
john.add(button3);
button4=new JButton("Συνέχεια για το τέλος");
button4.setBounds(861,510,250,45);
button4.setFont(nikos2);
button4.setBackground(kostas3);
button4.setBorder(thickBorder);
button4.setFocusable(false);
button4.addActionListener(blistener);
button4.setVisible(false);
john.add(button4);
button5=new JButton("Πάλι από την αρχή");
button5.setBounds(861,510,250,45);
button5.setFont(nikos2);
button5.setBackground(kostas3);
button5.setBorder(thickBorder);
button5.setFocusable(false);
button5.addActionListener(blistener);
button5.setVisible(false);
john.add(button5);
lable8=new JLabel("Επιτυχίες:");
lable8.setFont(nikos2);
lable8.setBounds(861,580,200,45);
john.add(lable8);
kostas4=new Color(0,200,0);
thickBorder2 = new LineBorder(Color.black,3);
thickBorder3 = new LineBorder(kostas2,2);
```

```
thickBorder4 = new LineBorder(kostas2,3);
nikos4=new Font("SansSerif",Font.BOLD,22);
lable9=new JLabel("  1 ");      // Εδώ αρχικοποιούμε το αντικείμενο lable9 της τάξης JLabel
lable9.setBounds(1180,12,40,35);          // να περιέχει μία συμβολοσειρά με συγκεκριμένο
lable9.setBackground(kostas3);                 // όνομα, στυλ και μέγεθος. Επίσης θέτουμε τα
lable9.setFont(nikos4);      // παρασκηνιακό χρώμα να είναι το άσπρο και το αντικείμενο να
lable9.setBorder(thickBorder2);           // έχει ένα ορισμένο πάχος περιγράμματος. Τα ίδια
lable9.setOpaque(true);   // ακριβώς ισχύουν και για τα αντικείμενα της κλάσης JLabel
john.add(lable9);               // lable10, lable11, lable12, lable13, lable14, lable15, lable16,
lable10=new JLabel("  2 ");       // lable17, lable18, lable19, lable20, lable21, lable22 και
lable10.setBounds(1180,53,40,35);                // lable23 με τη μόνη διαφορά πως αυτά τα
lable10.setBackground(kostas3);  // αντικείμενα έχουν πιο λεπτό πάχος περιγράμματος
lable10.setFont(nikos4);                                                  // από το αντικείμενο lable9.
lable10.setBorder(thickBorder);
lable10.setOpaque(true);
john.add(lable10);
lable11=new JLabel("  3 ");
lable11.setBounds(1180,94,40,35);
lable11.setBackground(kostas3);
lable11.setFont(nikos4);
lable11.setBorder(thickBorder);
lable11.setOpaque(true);
john.add(lable11);
lable12=new JLabel("  4 ");
lable12.setBounds(1180,135,40,35);
lable12.setBackground(kostas3);
lable12.setFont(nikos4);
lable12.setBorder(thickBorder);
lable12.setOpaque(true);
john.add(lable12);
lable13=new JLabel("  5 ");
lable13.setBounds(1180,176,40,35);
lable13.setBackground(kostas3);
lable13.setFont(nikos4);
lable13.setBorder(thickBorder);
lable13.setOpaque(true);
john.add(lable13);
lable14=new JLabel("  6 ");
lable14.setBounds(1180,217,40,35);
lable14.setBackground(kostas3);
lable14.setFont(nikos4);
lable14.setBorder(thickBorder);
lable14.setOpaque(true);
john.add(lable14);
lable15=new JLabel("  7 ");
lable15.setBounds(1180,258,40,35);
lable15.setBackground(kostas3);
lable15.setFont(nikos4);
lable15.setBorder(thickBorder);
lable15.setOpaque(true);
john.add(lable15);
lable16=new JLabel("  8 ");
lable16.setBounds(1180,299,40,35);
lable16.setBackground(kostas3);
lable16.setFont(nikos4);
lable16.setBorder(thickBorder);
lable16.setOpaque(true);
```

```
john.add(lable16);
lable17=new JLabel("  9 ");
lable17.setBounds(1180,340,40,35);
lable17.setBackground(kostas3);
lable17.setFont(nikos4);
lable17.setBorder(thickBorder);
lable17.setOpaque(true);
john.add(lable17);
lable18=new JLabel(" 10");
lable18.setBounds(1180,381,40,35);
lable18.setBackground(kostas3);
lable18.setFont(nikos4);
lable18.setBorder(thickBorder);
lable18.setOpaque(true);
john.add(lable18);
lable19=new JLabel(" 11");
lable19.setBounds(1180,422,40,35);
lable19.setBackground(kostas3);
lable19.setFont(nikos4);
lable19.setBorder(thickBorder);
lable19.setOpaque(true);
john.add(lable19);
lable20=new JLabel(" 12");
lable20.setBounds(1180,463,40,35);
lable20.setBackground(kostas3);
lable20.setFont(nikos4);
lable20.setBorder(thickBorder);
lable20.setOpaque(true);
john.add(lable20);
lable21=new JLabel(" 13");
lable21.setBounds(1180,504,40,35);
lable21.setBackground(kostas3);
lable21.setFont(nikos4);
lable21.setBorder(thickBorder);
lable21.setOpaque(true);
john.add(lable21);
lable22=new JLabel(" 14");
lable22.setBounds(1180,545,40,35);
lable22.setBackground(kostas3);
lable22.setFont(nikos4);
lable22.setBorder(thickBorder);
lable22.setOpaque(true);
john.add(lable22);
lable23=new JLabel(" 15");
lable23.setBounds(1180,586,40,35);
lable23.setBackground(kostas3);
lable23.setFont(nikos4);
lable23.setBorder(thickBorder);
lable23.setOpaque(true);
john.add(lable23);
lable24=new JLabel("");
lable24.setFont(nikos1);
add(lable24);
lable25=new JLabel("0");
lable25.setFont(nikos2);
lable25.setBounds(975,582,200,45);
john.add(lable25);
```

```
lable26=new JLabel("Η σωστή απάντηση είναι:");     // Εδώ αρχικοποιούμε ένα αντικείμενο
lable26.setFont(nikos2);              // lable26 της τάξης JLabel να περιέχει μία συμβολοσειρά
lable26.setBounds(768,75,300,40);                  // με συγκεκριμένο όνομα, στυλ και μέγεθος
lable26.setVisible(false);    // γραμματοσειράς. Επίσης θέτουμε το αντικείμενο αυτό να μην
john.add(lable26);              // είναι ορατό. Τα ίδια ισχύουν και για τα αντικείμενα της κλάσης
nikos5=new Font("SansSerif",Font.BOLD+Font.ITALIC,24);        // JLabel lable27, lable28,
lable27=new JLabel("Πρόοψη");         // lable29, lable30, lable31, lable38, lable39, lable40,
lable27.setFont(nikos5);                                                              // και lable41.
lable27.setBounds(1054,75,200,40);
lable27.setVisible(false);
john.add(lable27);
lable28=new JLabel("Αριστερή");
lable28.setFont(nikos5);
lable28.setBounds(1043,75,200,40);
lable28.setVisible(false);
john.add(lable28);
lable29=new JLabel("πλάγια όψη");
lable29.setFont(nikos5);
lable29.setBounds(1013,100,200,40);
lable29.setVisible(false);
john.add(lable29);
lable30=new JLabel("Κάτοψη");
lable30.setFont(nikos5);
lable30.setBounds(1058,75,200,40);
lable30.setVisible(false);
john.add(lable30);
lable31=new JLabel("Η σωστή απάντηση ήταν:");
lable31.setFont(nikos2);
lable31.setBounds(768,75,300,40);
lable31.setVisible(false);
john.add(lable31);
lable38=new JLabel("Άνοψη");
lable38.setFont(nikos5);
lable38.setBounds(1070,75,200,40);
lable38.setVisible(false);
john.add(lable38);
lable39=new JLabel("Δεξιά");
lable39.setFont(nikos5);
lable39.setBounds(1087,75,200,40);
lable39.setVisible(false);
john.add(lable39);
lable40=new JLabel("Πίσω");
lable40.setFont(nikos5);
lable40.setBounds(1087,75,200,40);
lable40.setVisible(false);
john.add(lable40);
lable41=new JLabel("όψη");
lable41.setFont(nikos5);
lable41.setBounds(1100,100,200,40);
lable41.setVisible(false);
john.add(lable41);
}
private class buttonListener implements ActionListener
{
        public void actionPerformed(ActionEvent event)
        {
        String buttonb=event.getActionCommand();
```

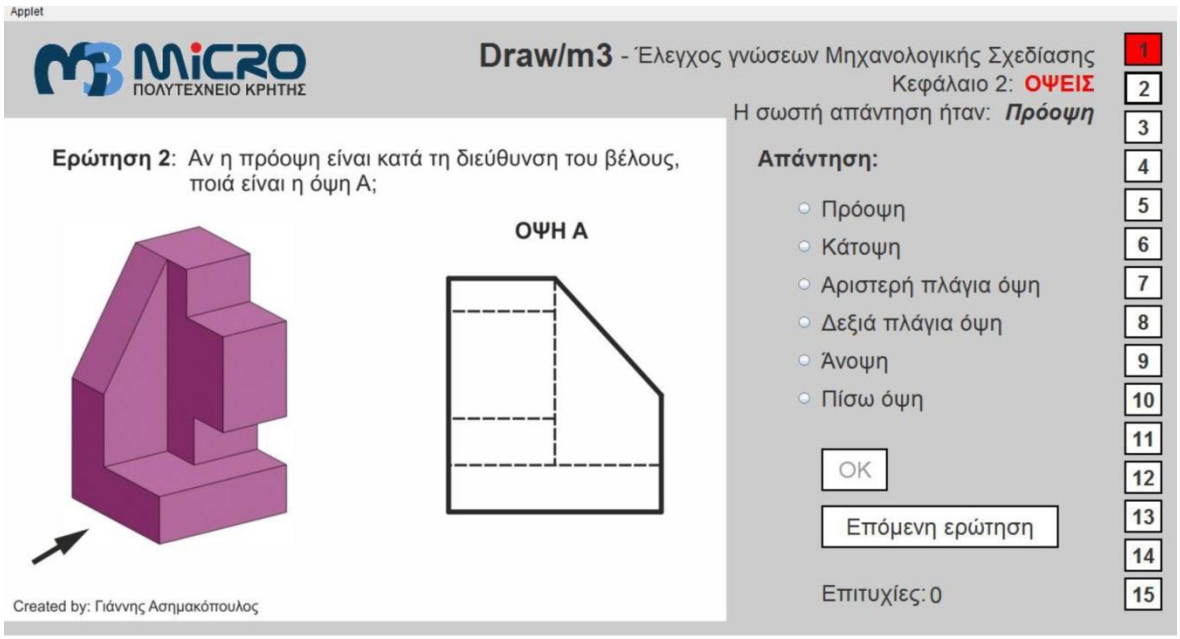
```
        sunartisi(buttonb);
        }
}
```



Σχήμα 3.1 : Αυτή η εικόνα είναι το πρώτο πράγμα που θα δει ο χρήστης όταν τρέξει τη συγκεκριμένη μικροεφαρμογή.

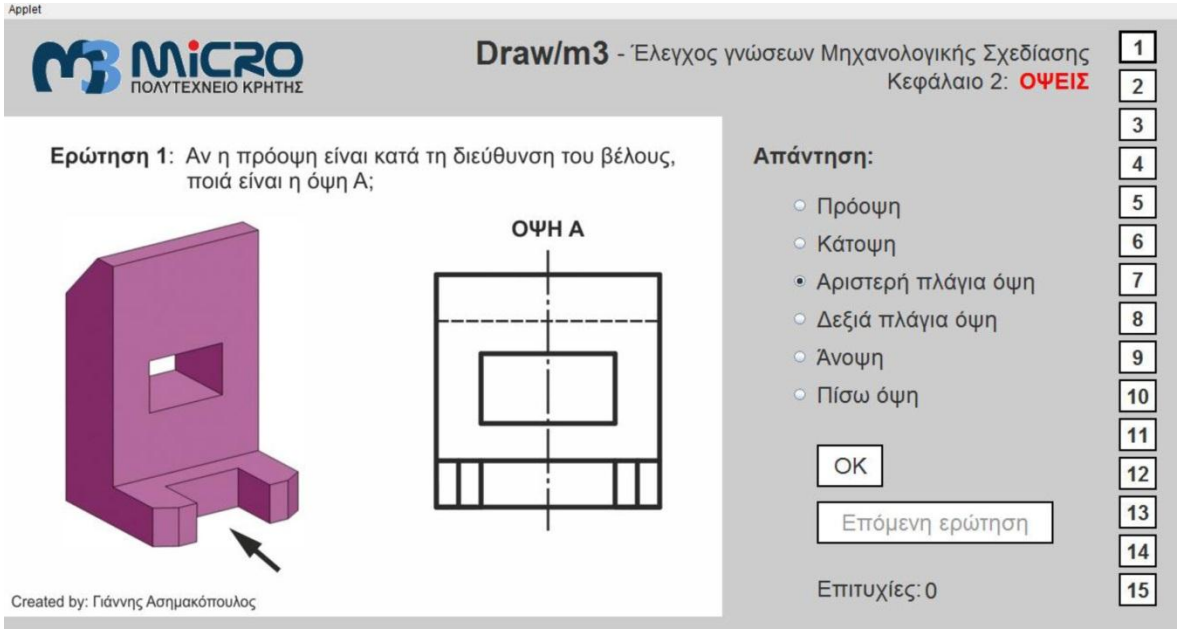
Εδώ είναι το απόσπασμα του κώδικα της συνάρτησης sunartisi:

```
public void sunartisi(String buttonb)
{
if((!rbutton1.isSelected()&&!rbutton2.isSelected()&&!rbutton3.isSelected()&&!rbutton4.isSelected()&&!rbutton5.isSelected()&&!rbutton6.isSelected())&&buttonb.equals("Επόμενη ερώτηση")&&j==0)
{           // Σε ετούτη τη συνθήκη του if ελέγχουμε εάν το j είναι 0 και ο χρήστης δεν επέλεξε
                    // κανένα ραδιοπλήκτρο ενώ πάτησε το παραλληλόγραμμο κουμπί με τη
                                                    // συμβολοσειρά "Επόμενη ερώτηση".
lable27.setVisible(true);      // Τότε θα εμφανιστούν τα αντικείμενα lable27 και lable31 της
lable31.setVisible(true);                  // τάξης JLabel που είναι η απάντηση της ερώτησης.
lable9.setBackground(kostas2);// Επίσης το παρασκηνιακό χρώμα του αντικειμένου lable9
lable9.setBorder(thickBorder);          // της κλάσης JLabel από άσπρο θα γίνει κόκκινο, το
lable10.setBorder(thickBorder2);  // περίγραμμα του αντικειμένου lable9 θα γίνει πιο λεπτό
lable1.setIcon(picture3); // ενώ το περίγραμμα του αντικειμένου lable10 της κλάσης JLabel
j=6;                  // θα γίνει πιο παχύ. Ακόμη θα φορτωθεί στο αντικείμενο lable1 της κλάσης
}   // JLabel η επόμενη εικόνα μηχανολογικού σχεδίου και ο ακέραιος j γίνεται ίσος με το 6.
```



Σχήμα 3.2 : Ετούτη η εικόνα αντικατοπτρίζει τι θα συμβεί στην περίπτωση του if που αναφέρθηκε προηγουμένως.

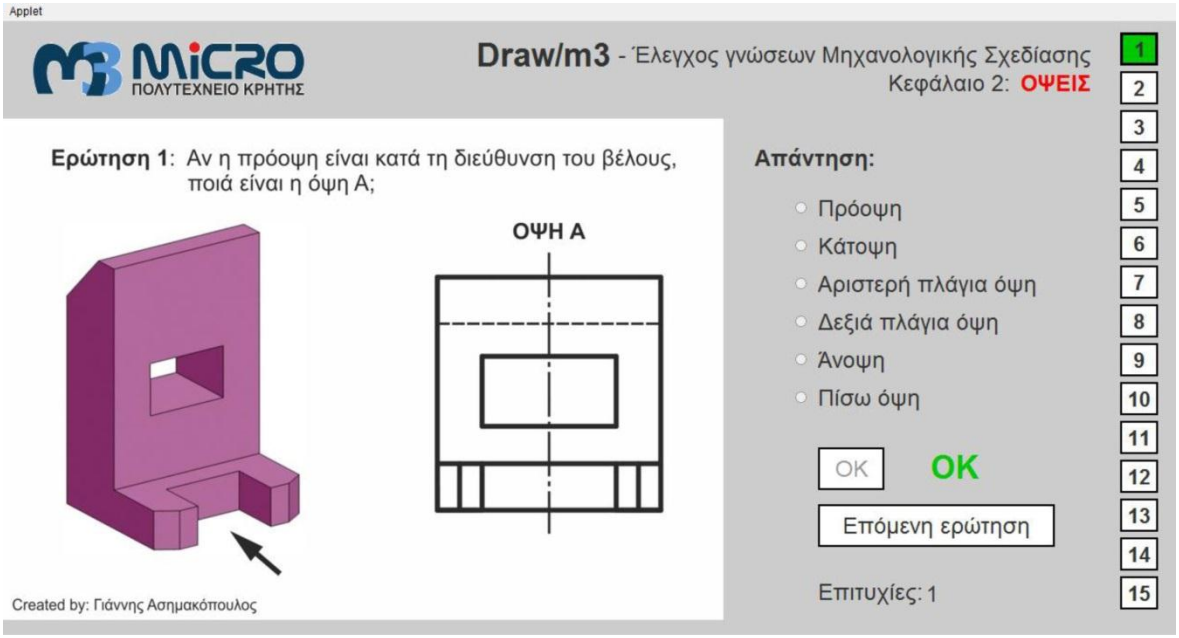
```
else
if((rbutton1.isSelected()||rbutton2.isSelected()||rbutton3.isSelected()||rbutton4.isSelected ()
|| rbutton5.isSelected()||rbutton6.isSelected())&&j==0)
{                       // Με αυτήν τη συνθήκη ελέγχουμε αν ο χρήστης επέλεξε κάποιο από τα
                                                    // ραδιοπλήκτρα και το j είναι 0. Εάν το j είναι 0 και
button1.setEnabled(true);                   // αν επιλέχθηκε κάποιο από τα ραδιοπλήκτρα τότε
                                                                // το αντικείμενο button1 της κλάσης
                                                  //JButton ήτοι το παραλληλόγραμμο κουμπί που
button2.setEnabled(false);          // περιέχει τη συμβολοσειρά "OK" ενεργοποιείται ενώ το
                                  // αντικείμενο button2 της ίδιας τάξης δηλαδή το παραλληλόγραμμο
j++;                                    // κουμπί που περιέχει τη συμβολοσειρά "Επόμενη ερώτηση"
                                                  // απενεργοποιείται. Επίσης η  τιμή του j αυξάνει κατά 1
            // δηλαδή γίνεται 1 αφού στη μέθοδο init τα i, j είχαν αρχικοποιηθεί με την τιμή 0.
}
```



Σχήμα 3.3 : Αυτή η εικόνα δείχνει τι θα συμβεί στην περίπτωση την οποία μόλις περιγράψαμε.

```
else if(buttonb.equals("OK")&&j==1&&rbutton1.isSelected())
{                                          // Με αυτήν τη συνθήκη ελέγχουμε εάν το j είναι 1,
                                                  // αν πατήθηκε το παραλληλόγραμμο
button1.setEnabled(false);          // κουμπί που περιέχει τη συμβολοσειρά "OK" και αν έχει
button2.setEnabled(true);                  // επιλεχθεί το πρώτο ραδιοπλήκτρο.
                                                  // Τότε το κουμπί που περιέχει
lable9.setBackground(kostas4);  // τη συμβολοσειρά "OK" απενεργοποιείται ενώ αυτό που
lable24.setText("OK");      // περιέχει τη συμβολοσειρά "Επόμενη ερώτηση" ενεργοποιείται.
lable24.setForeground(kostas4);          // Επίσης το αντικείμενο lable9 της τάξης JLabel
lable24.setBounds(980,450,200,40);          // γίνεται πράσινο ενώ εμφανίζεται με πράσινα
lable25.setText("1");                                  // γράμματα η λέξη OK.
                                          // Ακόμη δίπλα στη λέξη Επιτυχίες εμφανίζεται
το 1. if(rbutton1.isSelected()&&j==1)          // Εδώ εξετάζουμε αν επιλέχθηκε το
                                                  // πρώτο ραδιοπλήκτρο
{                  // και αν το j είναι ίσο με 1. Αν ισχύουν αυτά τότε βγαίνει από την ομάδα των
group.remove(rbutton1);  // ραδιοπλήκτρων το πρώτο ραδιοπλήκτρο, αφαιρείται η επιλογή
                                  // και προσθέτεται το ραδιοπλήκτρο πάλι στην ομάδα.
rbutton1.setSelected(false);          // Τα ίδια ακριβώς γίνονται και στις αμέσως
                                                  // επόμενες συνθήκες
group.add(rbutton1);                          // για τα υπόλοιπα 5 ραδιοπλήκτρα.
}

else if(rbutton2.isSelected()&&j==1)
{
group.remove(rbutton2);
rbutton2.setSelected(false);
group.add(rbutton2);
}

else if(rbutton3.isSelected()&&j==1)
{
group.remove(rbutton3);
rbutton3.setSelected(false);
```

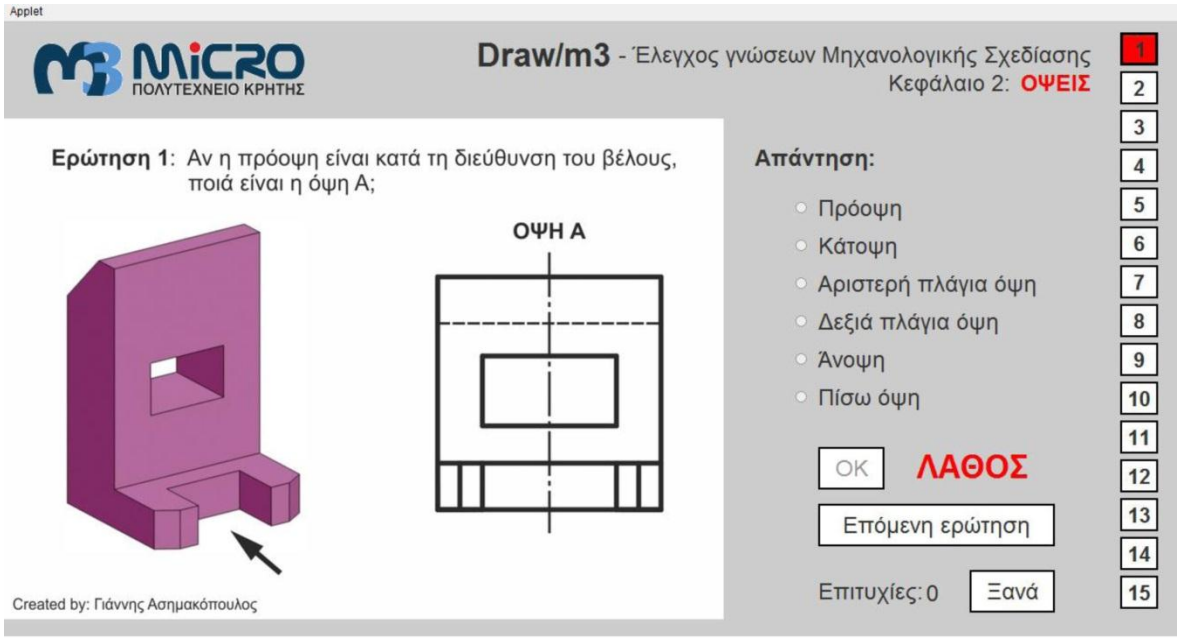
```
group.add(rbutton3);
}

else if(rbutton4.isSelected()&&j==1)
{
group.remove(rbutton4);
rbutton4.setSelected(false);
group.add(rbutton4);
}

else if(rbutton5.isSelected()&&j==1)
{
group.remove(rbutton5);
rbutton5.setSelected(false);
group.add(rbutton5);
}

else if(rbutton6.isSelected()&&j==1)
{
group.remove(rbutton6);
rbutton6.setSelected(false);
group.add(rbutton6);
}

rbutton1.setEnabled(false);                    // Μετά όλα τα ραδιοπλήκτρα απενεργοποιούνται.

rbutton2.setEnabled(false);

rbutton3.setEnabled(false);

rbutton4.setEnabled(false);

rbutton5.setEnabled(false);

rbutton6.setEnabled(false);

i=2;

j=5;                              // Τέλος για αυτήν την περίπτωση το i παίρνει την τιμή 2 ενώ το j
                                                                                    // λαμβάνει την τιμή 5.
}
```



Σχήμα 3.4 : Ετούτη η εικόνα δείχνει τι θα συμβεί στην περίπτωση που μόλις περιγράψαμε.

```
else if(buttonb.equals("OK")&&j==1&&!rbutton1.isSelected())
{    // Εδώ ελέγχουμε αν το j είναι 1, αν πατήθηκε το κουμπί που περιέχει τη συμβολοσειρά
button1.setEnabled(false);      // "OK" και το εάν δεν έχει επιλεγεί το πρώτο ραδιοπλήκτρο.
button2.setEnabled(true);                  // Τότε το κουμπί που περιέχει τη συμβολοσειρά "OK"
button3.setVisible(true); // απενεργοποιείται ενώ το κουμπί που περιέχει τη συμβολοσειρά
lable9.setBackground(kostas2); // "Επόμενη ερώτηση" ενεργοποιείται. Επίσης εμφανίζεται
lable24.setText("ΛΑΘΟΣ");              // ένα άλλο παραλληλόγραμμο κουμπί ενεργοποιημένο
lable24.setForeground(kostas2);    // το οποίο περιέχει τη συμβολοσειρά "Ξανά". Ακόμη το
lable24.setBounds(965,450,200,40);           // αντικείμενο lable9 της κλάσης JLabel γίνεται
                  // από άσπρο κόκκινο ενώ εμφανίζεται με κόκκινα γράμματα η λέξη ΛΑΘΟΣ.
if(rbutton1.isSelected()&&j==1)  // Εδώ εξετάζουμε αν επιλέχθηκε το πρώτο ραδιοπλήκτρο
{                // και εάν το j είναι 1. Αν ναι τότε βγαίνει από την ομάδα των ραδιοπλήκτρων το
group.remove(rbutton1);  // πρώτο ραδιοπλήκτρο, αφαιρείται η επιλογή και προσθέτεται το
rbutton1.setSelected(false);                                  // ραδιοπλήκτρο πάλι στην ομάδα.
group.add(rbutton1);          // Τα ίδια ακριβώς γίνονται και στις αμέσως επόμενες συνθήκες
}                                                             // για τα υπόλοιπα 5 ραδιοπλήκτρα.
else if(rbutton2.isSelected()&&j==1)
{
group.remove(rbutton2);
rbutton2.setSelected(false);
group.add(rbutton2);
}
else if(rbutton3.isSelected()&&j==1)
{
group.remove(rbutton3);
rbutton3.setSelected(false);
group.add(rbutton3);
}
else if(rbutton4.isSelected()&&j==1)
{
group.remove(rbutton4);
rbutton4.setSelected(false);
group.add(rbutton4);
}
```

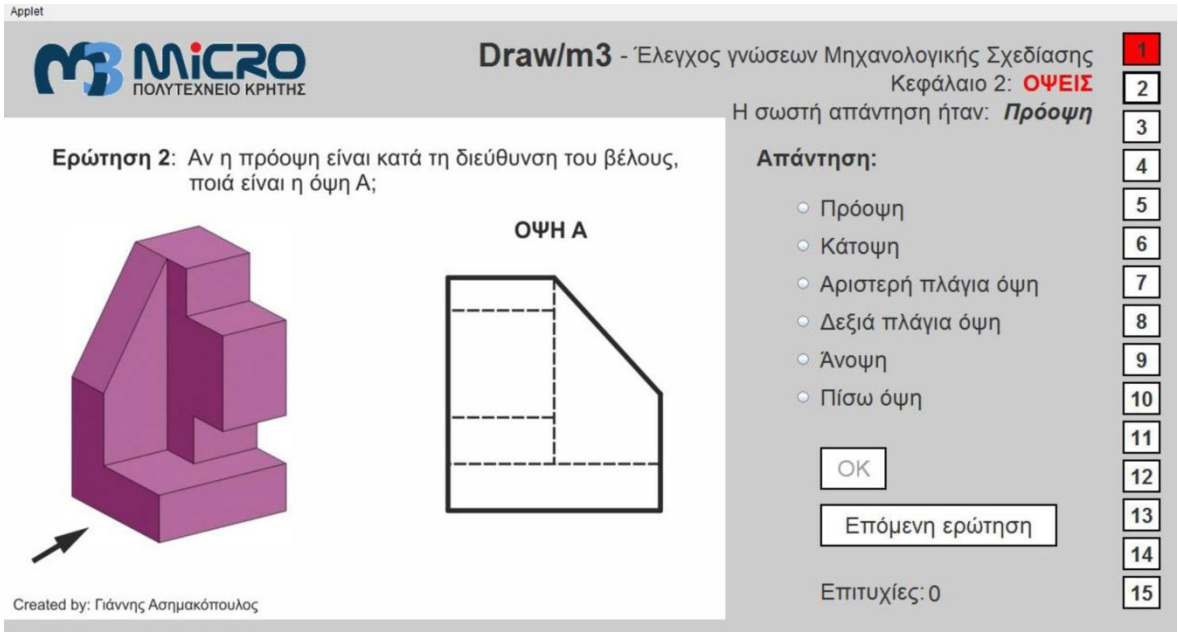
```
else if(rbutton5.isSelected()&&j==1)
{
group.remove(rbutton5);
rbutton5.setSelected(false);
group.add(rbutton5);
}
else if(rbutton6.isSelected()&&j==1)
{
group.remove(rbutton6);
rbutton6.setSelected(false);
group.add(rbutton6);
}
rbutton1.setEnabled(false);                    // Μετά όλα τα ραδιοπλήκτρα απενεργοποιούνται.
rbutton2.setEnabled(false);
rbutton3.setEnabled(false);
rbutton4.setEnabled(false);
rbutton5.setEnabled(false);
rbutton6.setEnabled(false);
j++;                                                    // Το j αυξάνει κατά 1 δηλαδή γίνεται 2.
}
```



Σχήμα 3.5 : Αυτή η εικόνα δείχνει τι θα συμβεί στην περίπτωση που μόλις περιγράψαμε.

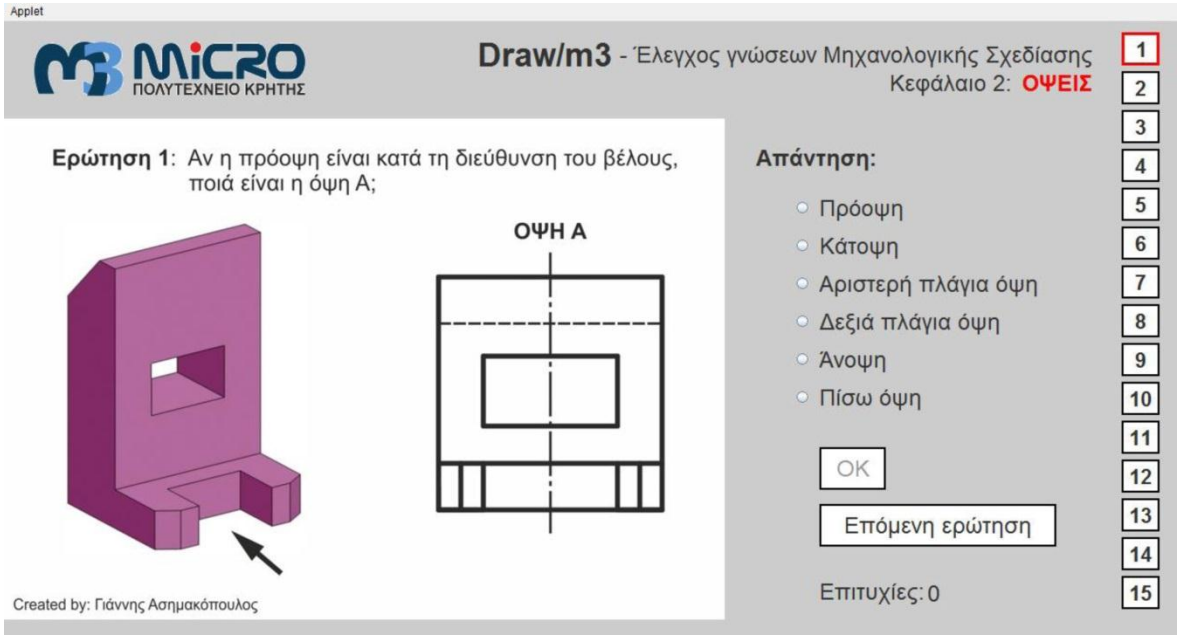
```
else if(buttonb.equals("Επόμενη ερώτηση")&&j==2)
{       // Με ετούτη τη συνθήκη ελέγχουμε αν το j ισούται με 2 και αν ο χρήστης έχει πατήσει
button3.setVisible(false);  // το κουμπί που περιέχει τη συμβολοσειρά "Επόμενη ερώτηση".
lable24.setText("");                // Τότε σβήνεται η συμβολοσειρά που περιέχει το αντικείμενο
lable27.setVisible(true);          // lable24 της τάξης JLabel και εμφανίζονται τα αντικείμενα
                // lable27 και lable31 της ίδιας κλάσης δηλαδή φανερώνεται η σωστή απάντηση.
lable31.setVisible(true);             // Επίσης το περίγραμμα του αντικειμένου lable9 της τάξης
lable9.setBorder(thickBorder);               // JLabel γίνεται πιο λεπτό ενώ το περίγραμμα του
lable10.setBorder(thickBorder2);          // αντικειμένου lable10 της ίδιας κλάσης γίνεται πιο
lable1.setIcon(picture3);                    // παχύ. Ακόμη φορτώνεται στο αντικείμενο lable1
                                                                        // η επόμενη εικόνα
rbutton1.setEnabled(true);                  // μηχανολογικού σχεδίου και όλα τα ραδιοπλήκτρα
```

```
rbutton2.setEnabled(true);                    // ενεργοποιούνται. Επιπρόσθετα
το rbutton3.setEnabled(true);                 // παραλληλόγραμμο κουμπί που περιέχει τη
rbutton4.setEnabled(true);                    // συμβολοσειρά "Ξανά" γίνεται αόρατο ενώ
rbutton5.setEnabled(true);                    // το j γίνεται ίσο με 6.
rbutton6.setEnabled(true);
j=6;
}
```



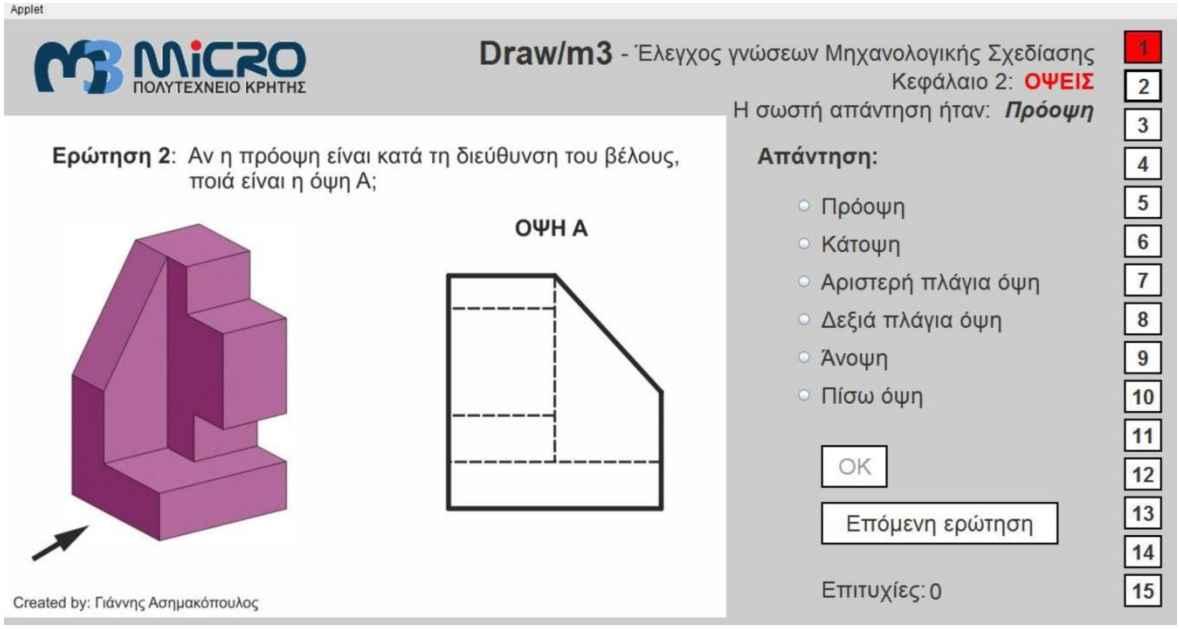
Σχήμα 3.6 : Ετούτη η εικόνα αντικατοπτρίζει τι θα συμβεί στην περίπτωση την οποία μόλις περιγράψαμε.

```
else if(buttonb.equals("Ξανά")&&j==2)
{              // Εδώ ελέγχουμε αν το j ισούται με το 2 και αν ο χρήστης έχει πατήσει το κουμπί
button3.setVisible(false);                  // που περιέχει τη συμβολοσειρά "Ξανά". Σε αυτήν την
lable24.setText("");                  // περίπτωση το κουμπί το οποίο περιέχει τη συμβολοσειρά
lable9.setBorder(thickBorder4); // "Ξανά" γίνεται μη ορατό, η λέξη του αντικειμένου lable24
lable9.setBackground(kostas3);     // της τάξης JLabel αφαιρείται ενώ το αντικείμενο lable9
rbutton1.setEnabled(true);     // της ίδιας κλάσης γεμίζει με άσπρο χρώμα ενώ πρώτα ήταν
rbutton2.setEnabled(true); // κόκκινο και το περίγραμμα του γίνεται κόκκινο. Επίσης όλα τα
rbutton3.setEnabled(true);                  // ραδιοπλήκτρα ενεργοποιούνται ενώ το j αυξάνεται
rbutton4.setEnabled(true);                                    // κατά 1 δηλαδή γίνεται 3.
rbutton5.setEnabled(true);
rbutton6.setEnabled(true);
j++;
}
```



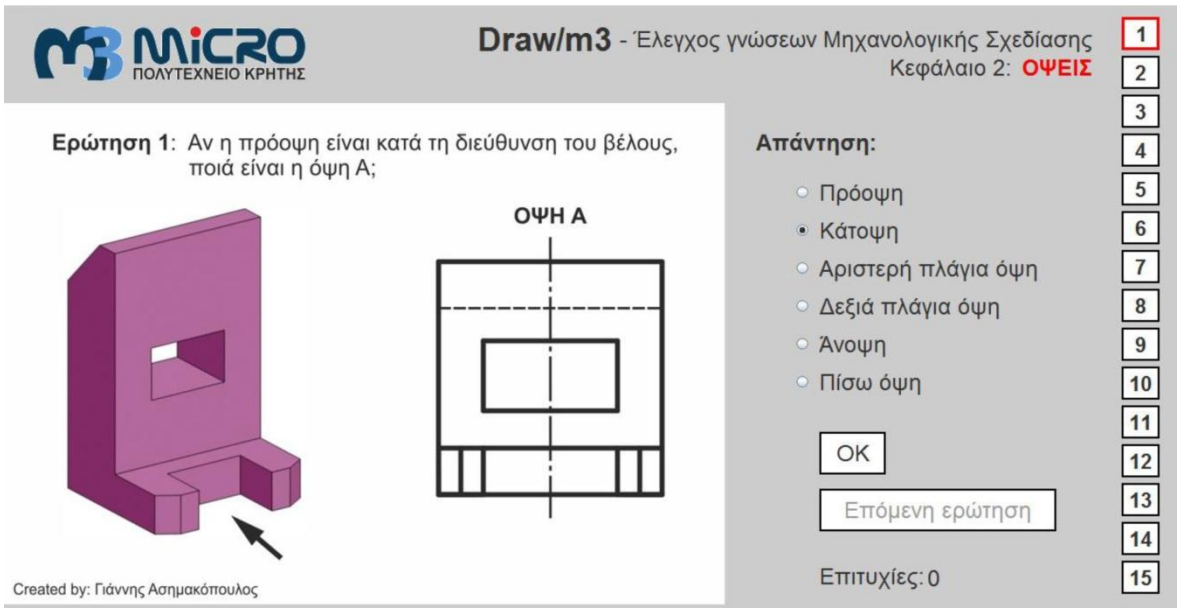
Σχήμα 3.7 : Ετούτη η εικόνα αντικατοπτρίζει τι θα συμβεί στην περίπτωση την οποία μόλις περιγράψαμε.

```
else
if((!rbutton1.isSelected()&&!rbutton2.isSelected()&&!rbutton3.isSelected()&&!rbutton4.is
Selected()&&!rbutton5.isSelected()&&!rbutton6.isSelected())&&buttonb.equals("Επόμενη
ερώτηση")&&j==3)
{                              // Εδώ ελέγχουμε αν το j ισούται με 3 και αν έχει πατήσει ο χρήστης
                                                                    // το κουμπί που περιέχει
lable27.setVisible(true);                    // τη συμβολοσειρά "Επόμενη ερώτηση" ενώ δεν
                                                                    //έχει επιλέξει κανένα
lable31.setVisible(true); // από τα ραδιοπλήκτρα. Τότε εμφανίζονται τα αντικείμενα lable27
lable9.setBackground(kostas2);   // και lable31 της κλάσης JLabel δηλαδή η απάντηση της
lable9.setBorder(thickBorder);     // ερώτησης. Επίσης το αντικείμενο lable9 της ίδιας τάξης
lable10.setBorder(thickBorder2);          // γεμίζει με χρώμα κόκκινο ενώ πρώτα είχε χρώμα
                                                                    // άσπρο
lable1.setIcon(picture3);                    // και το περίγραμμα του γίνεται πιο λεπτό και
                                                                    // παίρνει μαύρο χρώμα
                                                  // ενώ πρώτα ήταν κόκκινο. Ακόμη το περίγραμμα
                                                    // του αντικειμένου lable10 της τάξης JLabel
j=6;                           // γίνεται πιο παχύ, το j γίνεται 6 ενώ φορτώνεται στο αντικείμενο
                                                                    // lable1 της κλάσης
}                                  // JLabel η εικόνα του επόμενου μηχανολογικού σχεδίου.
```



Σχήμα 3.8 : Αυτή η εικόνα δείχνει τι θα συμβεί στην περίπτωση που μόλις περιγράψαμε.

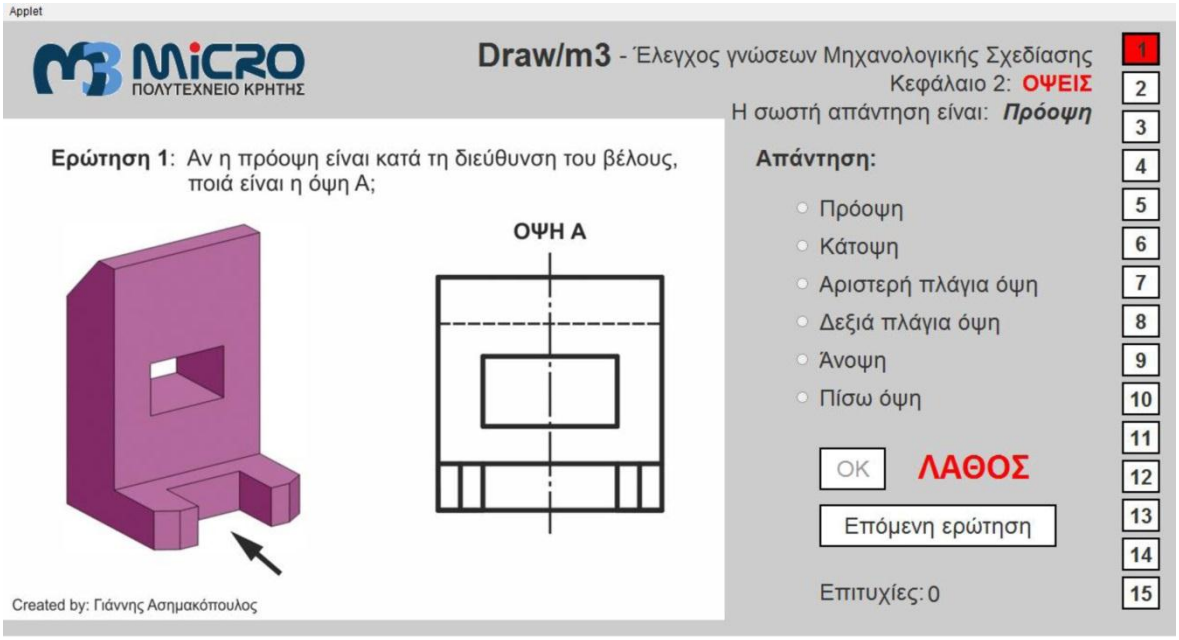
```
else
if((rbutton1.isSelected()||rbutton2.isSelected()||rbutton3.isSelected()||rbutton4.isSelected ()
|| rbutton5.isSelected()||rbutton6.isSelected())&&j==3)
{                    // Εδώ ελέγχουμε αν το j ισούται με 3 και αν ο χρήστης έχει επιλέξει κάποιο
button1.setEnabled(true);  // ραδιοπλήκτρο. Τότε το κουμπί που περιέχει τη συμβολοσειρά
button2.setEnabled(false);                    // "OK" ενεργοποιείται, το κουμπί που εμπεριέχει τη
j++; }        // συμβολοσειρά "Επόμενη ερώτηση" απενεργοποιείται ενώ το j αυξάνει και
γίνεται 4.
```



Σχήμα 3.9 : Ετούτη η εικόνα δείχνει τι θα συμβεί στην περίπτωση που μόλις περιγράψαμε.

```
else if(buttonb.equals("OK")&&j==4&&!rbutton1.isSelected())
{                    // Εδώ ελέγχουμε αν το j ισούται με 4 και αν ο χρήστης έχει πατήσει το κουμπί
button1.setEnabled(false);                    // το οποίο περιέχει τη συμβολοσειρά "OK" ενώ δεν έχει
```

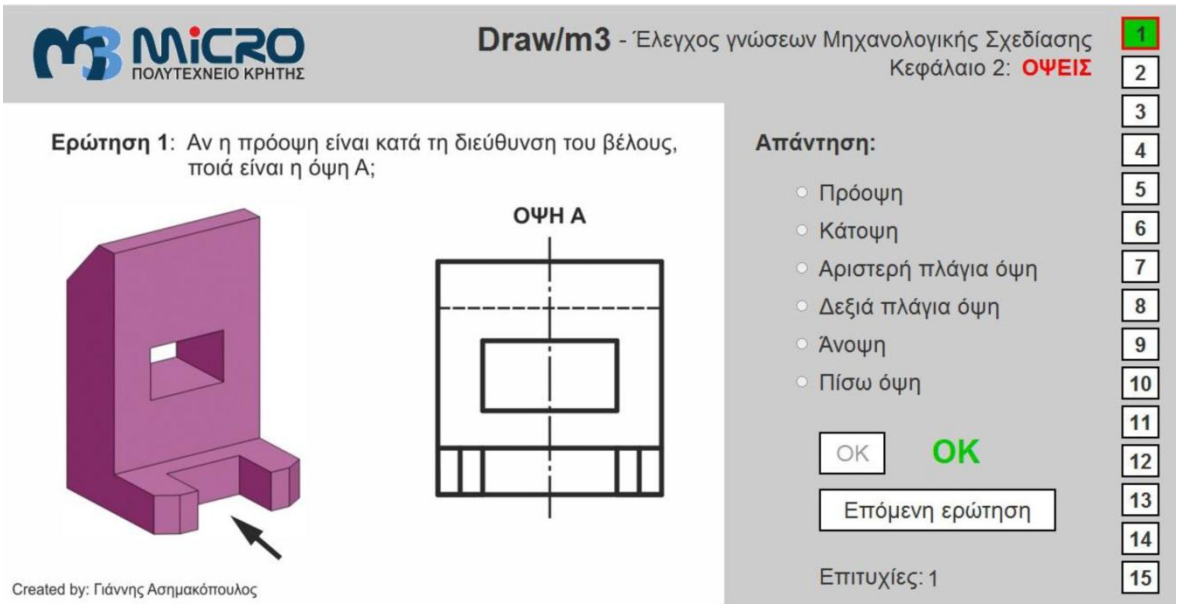
```
button2.setEnabled(true);       // επιλέξει το πρώτο ραδιοπλήκτρο. Τότε το κουμπί το οποίο
lable9.setBackground(kostas2);      // περιέχει τη συμβολοσειρά "OK" απενεργοποιείται, το
lable9.setBorder(thickBorder2);                         // κουμπί που περιέχει τη συμβολοσειρά
lable24.setText("ΛΑΘΟΣ");           // "Επόμενη ερώτηση" ενεργοποιείται ενώ το αντικείμενο
lable24.setForeground(kostas2);   // lable9 της τάξης JLabel γεμίζει με κόκκινο χρώμα ενώ
lable24.setBounds(965,450,200,40);   // πρώτα ήταν άσπρο και το περίγραμμα του γίνεται
lable26.setVisible(true);        // μαύρο ενώ πρώτα ήταν κόκκινο. Επίσης εμφανίζεται η λέξη
lable27.setVisible(true); // ΛΑΘΟΣ με κόκκινα γράμματα ενώ γίνονται ορατά τα αντικείμενα
                                                // lable26 και lable27 της κλάσης JLabel δηλαδή
                                                                // η απάντηση της ερώτησης.
if(rbutton1.isSelected()&&j==4)  // Εδώ εξετάζουμε αν επιλέχθηκε το πρώτο ραδιοπλήκτρο
{       // και αν το j είναι 4. Αν ναι τότε βγαίνει από την ομάδα των ραδιοπλήκτρων το πρώτο
group.remove(rbutton1);             // ραδιοπλήκτρο, αφαιρείται η επιλογή και προσθέτεται το
rbutton1.setSelected(false);                                // ραδιοπλήκτρο πάλι στην ομάδα.
group.add(rbutton1);          // Τα ίδια ακριβώς γίνονται και στις αμέσως επόμενες συνθήκες
}                                                           // για τα υπόλοιπα 5 ραδιοπλήκτρα.
else if(rbutton2.isSelected()&&j==4)
{
group.remove(rbutton2);
rbutton2.setSelected(false);
group.add(rbutton2);
}
else if(rbutton3.isSelected()&&j==4)
{
group.remove(rbutton3);
rbutton3.setSelected(false);
group.add(rbutton3);
}
else if(rbutton4.isSelected()&&j==4)
{
group.remove(rbutton4);
rbutton4.setSelected(false);
group.add(rbutton4);
}
else if(rbutton5.isSelected()&&j==4)
{
group.remove(rbutton5);
rbutton5.setSelected(false);
group.add(rbutton5);
}
else if(rbutton6.isSelected()&&j==4)
{
group.remove(rbutton6);
rbutton6.setSelected(false);
group.add(rbutton6);
}
rbutton1.setEnabled(false);                 // Μετά όλα τα ραδιοπλήκτρα απενεργοποιούνται.
rbutton2.setEnabled(false);
rbutton3.setEnabled(false);
rbutton4.setEnabled(false);
rbutton5.setEnabled(false);
rbutton6.setEnabled(false);
i--;                                                // Τέλος το i μειώνεται κατά 1 και γίνεται -1 ενώ
j++;                                                        // το j αυξάνει κατά 1 δηλαδή γίνεται 5.
}
```



Σχήμα 3.10 : Ετούτη η εικόνα αντικατοπτρίζει τι θα συμβεί στην περίπτωση την οποία μόλις περιγράψαμε.

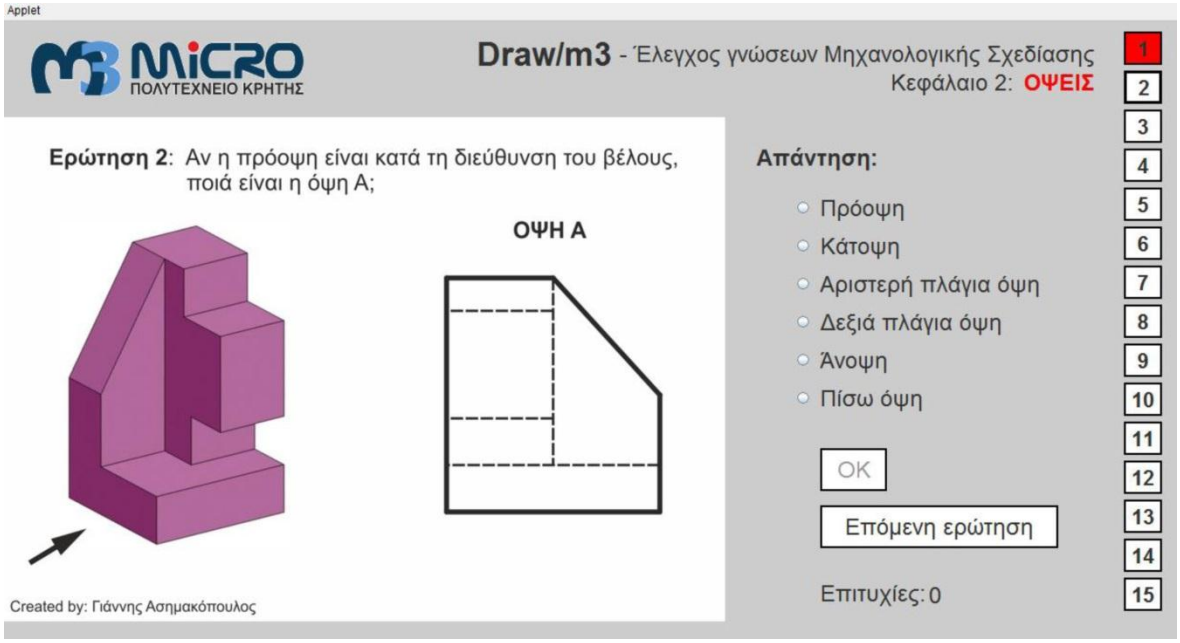
```
else if(buttonb.equals("OK")&&j==4&&rbutton1.isSelected())
{               // Εδώ ελέγχουμε αν το j ισούται με 4, αν έχει επιλεγεί το πρώτο ραδιοπλήκτρο
button1.setEnabled(false);                  // και αν ο χρήστης έχει πατήσει του κουμπί το οποίο
button2.setEnabled(true);               // εμπεριέχει τη συμβολοσειρά "OK". Αν ισχύουν ετούτα
lable9.setBackground(kostas4);                  // τότε το κουμπί που περιέχει τη συμβολοσειρά
lable24.setText("OK");                  // "OK" απενεργοποιείται, το κουμπί το οποίο περιέχει
lable24.setForeground(kostas4);                     // τη συμβολοσειρά "Επόμενη ερώτηση"
lable24.setBounds(980,450,200,40);                  // ενεργοποιείται και εμφανίζεται η λέξη OK
lable25.setText("1");          // με πράσινα γράμματα. Επίσης το αντικείμενο lable9 της τάξης
                                // JLabel γεμίζει με πράσινο χρώμα ενώ πρώτα είχε χρώμα άσπρο
                                                // ενώ εμφανίζεται δίπλα από τη λέξη Επιτυχίες το 1.
if(rbutton1.isSelected()&&j==4)  // Εδώ εξετάζουμε αν επιλέχθηκε το πρώτο ραδιοπλήκτρο
{       // και αν το j είναι 4. Αν ναι τότε βγαίνει από την ομάδα των ραδιοπλήκτρων το πρώτο
group.remove(rbutton1);             // ραδιοπλήκτρο, αφαιρείται η επιλογή και προσθέτεται το
rbutton1.setSelected(false);                                    // ραδιοπλήκτρο πάλι στην ομάδα.
group.add(rbutton1);          // Τα ίδια ακριβώς γίνονται και στις αμέσως επόμενες συνθήκες
}                                                               // για τα υπόλοιπα 5 ραδιοπλήκτρα.
else if(rbutton2.isSelected()&&j==4)
{group.remove(rbutton2);
rbutton2.setSelected(false);
group.add(rbutton2); }
else if(rbutton3.isSelected()&&j==4)
{group.remove(rbutton3);
rbutton3.setSelected(false);
group.add(rbutton3);  }
else if(rbutton4.isSelected()&&j==4)
{group.remove(rbutton4);
rbutton4.setSelected(false);
group.add(rbutton4);  }
else if(rbutton5.isSelected()&&j==4)
{group.remove(rbutton5);
rbutton5.setSelected(false);
group.add(rbutton5);  }
```

```
else if(rbutton6.isSelected()&&j==4)
{group.remove(rbutton6);
rbutton6.setSelected(false);
group.add(rbutton6);  }
rbutton1.setEnabled(false);                  // Μετά όλα τα ραδιοπλήκτρα απενεργοποιούνται.
rbutton2.setEnabled(false);
rbutton3.setEnabled(false);
rbutton4.setEnabled(false);
rbutton5.setEnabled(false);
rbutton6.setEnabled(false);
i++;                                                  // Τέλος το i αυξάνεται κατά 1 και γίνεται 1 ενώ
j++;                                                 // το j επίσης αυξάνεται κατά 1 δηλαδή γίνεται 5.
}
```



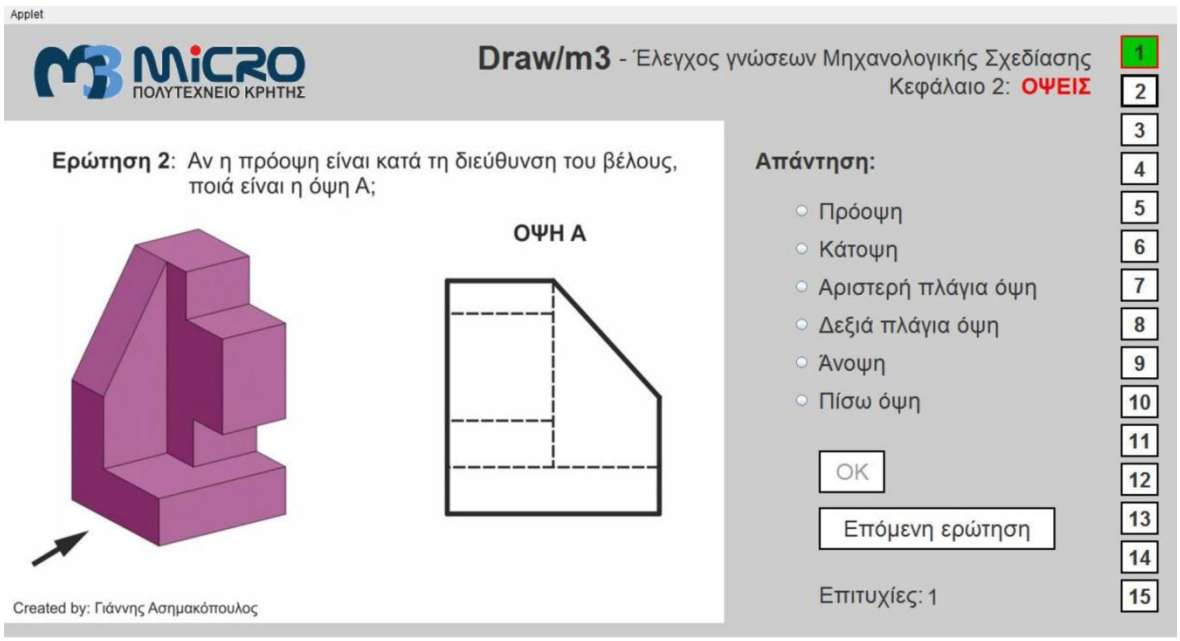
Σχήμα 3.11 : Ετούτη η εικόνα αντικατοπτρίζει τι θα συμβεί στην περίπτωση την οποία μόλις περιγράψαμε.

```
else if(buttonb.equals("Επόμενη ερώτηση")&&j==5)
{                                  // Εδώ ελέγχουμε εάν ο χρήστης πάτησε το κουμπί που περιέχει τη
lable24.setText("");                    // συμβολοσειρά "Επόμενη ερώτηση" και το j είναι 5. Τότε
lable10.setBorder(thickBorder2);      // η λέξη την οποία περιέχει το αντικείμενο lable24 της
lable1.setIcon(picture3);                    // τάξης JLabel αφαιρείται ενώ το περίγραμμα του
          // αντικειμένου lable10 της ίδιας κλάσης γίνεται πιο παχύ. Επίσης φορτώνεται στο
                                  // στο αντικείμενο lable1 της κλάσης JLabel η εικόνα του επόμενου
            // μηχανολογικού σχεδίου. Ακόμη όλα τα ραδιοπλήκτρα ενεργοποιούνται και το j
                                                                  // αυξάνεται κατά 1 τουτέστιν γίνεται 6.
if(i==-1)                                                  // Σε αυτήν την υποπερίπτωση ελέγχουμε εάν
{                                  // το i ισούται με το -1. Αν ναι τότε το περίγραμμα του αντικειμένου
                                                                  // lable9 της κλάσης JLabel
lable9.setBorder(thickBorder);               // γίνεται πιο λεπτό. Ακόμη τα αντικείμενα lable26
lable26.setVisible(false);                   // και lable27 της τάξης JLabel γίνονται μη ορατά.
lable27.setVisible(false);
}
```



Σχήμα 3.12 : Ετούτη η εικόνα αντικατοπτρίζει τι θα συμβεί στην πρώτη υποπερίπτωση της περίπτωσης την οποία μόλις περιγράψαμε.

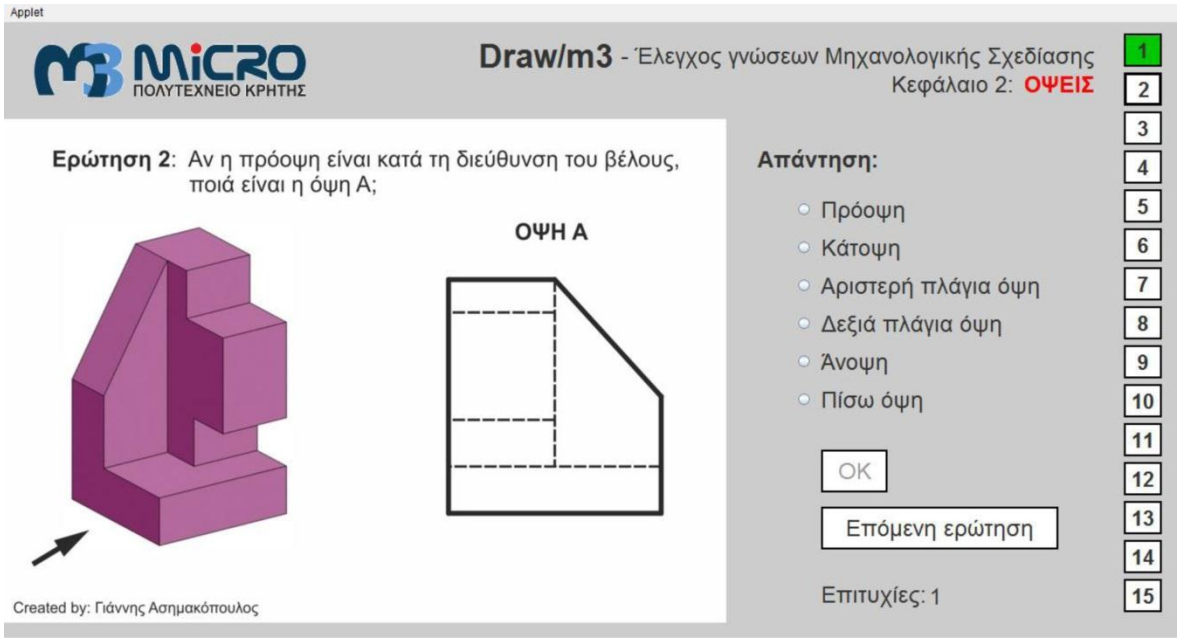
```
else if(i==1)
{                              // Σε αυτήν την υποπερίπτωση ελέγχουμε αν το i ισούται με 1. Αν ναι
                                    // τότε το περίγραμμα του αντικειμένου lable9 της κλάσης JLabel
lable9.setBorder(thickBorder3);          // γίνεται πιο λεπτό αλλά διατηρεί το κόκκινο
                                                                    // χρώμα του.
}
```



Σχήμα 3.13 : Ετούτη η εικόνα αντικατοπτρίζει τι θα συμβεί στη δεύτερη υποπερίπτωση της περίπτωσης την οποία μόλις περιγράψαμε.

```
else if(i==2)                                  // Εδώ ελέγχεται αν το i ισούται με 2.
                                        // Αν ναι τότε το περίγραμμα του αντικειμένου lable9
{                                       // της κλάσης JLabel γίνεται πιο λεπτό.
```

```
lable9.setBorder(thickBorder);
}

rbutton1.setEnabled(true);

rbutton2.setEnabled(true);

rbutton3.setEnabled(true);

rbutton4.setEnabled(true);

rbutton5.setEnabled(true);

rbutton6.setEnabled(true);

j++;

/* Εδώ όπως προαναφέρθηκε όλα τα ραδιοπλήκτρα ενεργοποιούνται ενώ το j αυξάνει κατά 1 δηλαδή γίνεται 6. */\

}
```



Σχήμα 3.14 : Αυτή η εικόνα αντικατοπτρίζει τι θα συμβεί στην Τρίτη υποπερίπτωση της περίπτωσης την οποία μόλις περιγράψαμε.

4. ΣΥΝΟΨΗ

Στην παρούσα διπλωματική εργασία κατασκευάστηκε ένα λογισμικό με τη χρήση Java Applets που σκοπό έχει την εξοικείωση του χρήστη με τις όψεις των μηχανολογικών σχεδίων. Το λογισμικό αυτό μπορεί να τοποθετηθεί στο διαδίκτυο προσφέροντας εύκολη πρόσβαση στον οποιονδήποτε χρήστη.

Η εφαρμογή αυτή παρουσιάζει μια σειρά ερωτήσεων που καλείται να απαντήσει ο χρήστης. Κάθε ερώτηση συνοδεύεται από ένα τρισδιάστατο σχήμα το οποίο αφορά και ταυτόχρονα προβάλλονται οι προτεινόμενες απαντήσεις που μπορούν να δοθούν. Αφού ο χρήστης επιλέξει την απάντηση που θεωρεί σωστή μπορεί να την επιβεβαιώσει πατώντας το κουμπί ‘OK’. Σε περίπτωση που δεν είναι σίγουρος για την απάντηση δίνεται και η επιλογή να αφήσει την ερώτηση για το τέλος.

Ανάλογα με το αν η απάντηση είναι σωστή ή λάθος εμφανίζεται η αντίστοιχη ένδειξη και ο αντίστοιχος χρωματισμός στον αριθμό της ερώτησης. Όταν η ερώτηση είναι λάθος εμφανίζεται και η σωστή απάντηση στην οθόνη του χρήστη. Στο κάτω μέρος της εφαρμογής εμφανίζεται ο συνολικός αριθμός των επιτυχιών.

5. ΒΙΒΛΙΟΓΡΑΦΙΑ

- Κάβουρας, Ιωάννης Κ., Προγραμματισμός με Java.
- Οδυσσέας Ευαγγέλου, Σχολικός Σύμβουλος, Διαδίκτυο και διαπολιτισμική εκπαίδευση: Δυνατότητες και περιορισμοί.
- Γιώργος Επιτήδειος, Internet & Εκπαίδευση.
- Ιωάννης Ι. Κεκές, Πανεπιστήμιο Αθηνών.
- Ηρώ Μυλωνάκου - Κεκέ, Παιδαγωγικό Ινστιτούτο.
- Διαδίκτυο (Internet) και μάθηση: Οι στρατηγικές για την «πλοήγηση» και η διδακτική τους αξία.
- Κασούτσας Αιμιλιανός, Ελληνικό Ανοικτό Πανεπιστήμιο, Σχολή Θετικών Επιστημών και Τεχνολογίας, MSc. Μαθηματικός, Εκπαιδευτικός.
- Τα Mathlets στη Μαθηματική Εκπαίδευση.
- Δρ. Ρετάλης Σ. - Τσέλιος Γ., Μαθήματα HTML: από το απλό στο σύνθετο.
- Savitch, Walter, Η Απόλυτη Java.
- Lervik, Else, Java με UML, Αντικειμενοστρεφής σχεδίαση και προγραμματισμός.
- Steven Holzner, ΟΔΗΓΟΣ ΤΗΣ XML.
- Internet, Java API.