



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εκπαιδευτικό λογισμικό για την εκμάθηση της MATLAB
Τίτλος Διατριβής Αγγλικά	Interactive educational software for learning the basic elements of the programming language MATLAB
Όνοματεπώνυμο Φοιτητή	Κατερίνα Λουρέντζου
Πατρώνυμο	Διονύσιος
Αριθμός Μητρώου	ΜΠΠΛ/ 10040
Επιβλέπων	Μαρία Βίρβου, Καθηγήτρια

Ημερομηνία Παράδοσης **Οκτώβριος 2013**

Πανεπιστήμιο Πειραιώς

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Βίρβου Μαρία
Καθηγήτρια

Γεώργιος Α. Τσιχριντζής
Καθηγητής

Θεμιστοκλής Παναγιωτόπουλος
Καθηγητής

ΠΕΡΙΛΗΨΗ

Η παρούσα μεταπτυχιακή διατριβή πραγματοποιήθηκε στα πλαίσια διπλωματικής του Μεταπτυχιακού Πληροφορική του Πανεπιστημίου Πειραιώς. Θέμα της διατριβής είναι η σχεδίαση και υλοποίηση ενός εύχρηστου και αλληλεπιδραστικού εκπαιδευτικού λογισμικού για την εκμάθηση των βασικών στοιχείων της γλώσσας προγραμματισμού MATLAB. Στόχος του είναι η υποστήριξη ενός συστήματος εξ αποστάσεως εκμάθησης και εξάσκησης στα βασικά στοιχεία της MATLAB, αλλά και η δημιουργία ενός συστήματος προσωπικής εμβάθυνσης στη MATLAB μέσα από ένα ολοκληρωμένο σύστημα διδασκαλίας. Το συγκεκριμένο λογισμικό έχει τη δυνατότητα, έπειτα από κατάλληλη επεξεργασία της βάσης δεδομένων, να χρησιμοποιηθεί και για τη διδασκαλία άλλων μαθημάτων.

ABSTRACT

The present Master Thesis has been completed under the Master diploma Informatics of University of Piraeus. It presents the design and development of a practical and interactive educational software for learning the basic elements of the programming language MATLAB. The objective is to support a system of remote learning and practice in the basics elements of MATLAB and the creation of a comprehensive self-teaching system. This software has the ability, after the appropriate processing of the database, to be used for other courses also.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	3
ABSTRACT	3
[1]ΕΙΣΑΓΩΓΗ	8
[2]ΑΝΑΣΚΟΠΗΣΗ ΠΕΔΙΟΥ	15
ΜΕΡΟΣ Α: Η ΕΞΕΛΙΞΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ Η ΧΡΗΣΗ ΤΗΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ	15
2.1. Η ΕΞΕΛΙΞΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ	15
2.2. Ο ΟΡΟΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΣ (ΤΠΕ) ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ Η ΣΗΜΑΣΙΑ ΤΟΥ ...	18
2.2.1. Χρονολογικές φάσεις ένταξης των ΤΠΕ στην Εκπαίδευση	19
2.2.2. Χρονολογικές φάσεις εισαγωγής και ανάπτυξης της τεχνολογίας στην Εκπαίδευση	20
2.2.3. Μοντέλα Ένταξης των Νέων Τεχνολογιών στην Εκπαίδευση	22
2.2.4. Χαρακτηριστικά των ΤΠΕ σχετικά με την εκπαίδευση	24
2.3. ΟΙ ΕΦΑΡΜΟΓΕΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ	25
2.4. ΣΥΓΧΡΟΝΕΣ ΘΕΩΡΙΕΣ ΜΑΘΗΣΗΣ ΚΑΙ ΤΠΕ	26
2.4.1. Συμπεριφορισμός (behaviorism)	26
2.4.2. Οικοδομισμός ή Δομητισμός (Constructivism)	27
2.4.3. Κοινωνικοπολιτισμικές θεωρίες - Θεωρία της δραστηριότητας (activity theory)	28
2.5. ΝΕΟΣ ΡΟΛΟΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΚΑΙ ΕΚΠΑΙΔΕΥΟΜΕΝΟΥ	28
ΜΕΡΟΣ Β: ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ	29
2.6. ΣΤΟΧΟΣ ΕΛ (ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΛΟΓΙΣΜΙΚΩΝ)	29
2.7. ΚΑΤΗΓΟΡΙΕΣ ΕΛ	30
2.7.1. Κατηγοριοποίηση των ΕΛ με βάση τη θεωρία μάθησης που έχουν σχεδιαστεί	30
2.7.2. Κατηγοριοποίηση των ΕΛ με βάση τη χρήση τους στην εκπαιδευτική διαδικασία	31
2.7.3. Κατηγοριοποίηση των ΕΛ με βάση το βαθμό αλληλεπίδρασης	36
2.7.4. Κατηγοριοποίηση των ΕΛ με βάση την παιδαγωγική προσέγγιση	36
2.8. ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ	36
2.9. ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ	37
2.10. ΣΧΕΔΙΑΣΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ	37
2.10.1. Βασικές Αρχές Σχεδίασης	38
2.10.2. Στρατηγικές Σχεδίασης	38
2.10.3. Βασικοί παράγοντες επιτυχημένης σχεδίασης	39
2.11. ΠΟΙΟΤΗΤΑ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ	39
2.11.1. Ιδιότητες που χαρακτηρίζουν την ποιότητα ενός εκπαιδευτικού λογισμικού	41
2.12. ΜΕΘΟΔΟΙ ΑΞΙΟΛΟΓΗΣΗΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΛΟΓΙΣΜΙΚΟΥ	43
2.12.1. Αξιολόγηση με πειραματικές μεθόδους - εμπειρικές μέθοδοι αξιολόγησης	44
2.12.2. Αξιολόγηση με λίστες αξιολόγησης (check lists)	44
2.12.3. Αξιολόγηση από ειδικούς – μέθοδοι επιθεώρησης	46
2.12.4. Μοντελοποιημένες τεχνικές και αναλυτικές μέθοδοι αξιολόγησης	47
2.12.5. Αξιολόγηση με βάση τις θεωρίες μάθησης	47
2.12.6. Μέθοδοι αξιολόγησης με τη συμμετοχή χρηστών	48
ΜΕΡΟΣ Γ: ΆΛΛΑ ΕΚΠΑΙΔΕΥΤΙΚΑ ΛΟΓΙΣΜΙΚΑ	49
ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ «ΠΛΗΡΟΦΟΡΙΚΗ Α' Β' Γ' ΓΥΜΝΑΣΙΟΥ»	49
ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ «ΑΛΓΟΡΙΘΜΙΚΗ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ»	52

ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ «ΤΑΞΙΔΙ ΣΕ ΕΝΑ ΔΙΚΤΥΟ»	54
[3]ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ	56
3.1. Σχεδιασμός βάσης δεδομένων λογισμικού	56
3.2. Εργαλεία και Τεχνολογίες που χρησιμοποιήθηκαν	58
[4]ΠΑΡΟΥΣΙΑΣΗ ΚΑΙ ΧΡΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	60
[5]ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	92
[6]ΒΙΒΛΙΟΓΡΑΦΙΑ.....	93
[7]ΠΑΡΑΡΤΗΜΑΤΑ.....	94
ΠΑΡΑΡΤΗΜΑ Α: ΕΝΟΤΗΤΕΣ ΘΕΩΡΙΑΣ ΒΑΣΙΚΩΝ ΣΤΟΙΧΕΙΩΝ MATLAB	94
ΠΑΡΑΡΤΗΜΑ Α: ΑΣΚΗΣΕΙΣ ΒΑΣΙΚΩΝ ΕΝΟΤΗΤΩΝ	183

Πανεπιστήμιο Πειραιώς

Εικόνα: 1 - ENIAC ο πρώτος υπολογιστής	16
Εικόνα: 2 - Σύγχρονος φορητός ηλεκτρονικός υπολογιστής (laptop)	18
Εικόνα: 3 Οι ΤΠΕ και η Πληροφορική στην Εκπαίδευση	20
Εικόνα: 4 Ο κώνος των εμπειριών του E.Dale	25
Εικόνα: 5 Εικόνα από το παιχνίδι Age Of Empires III	32
Εικόνα: 6 Interactive Physics - Προσομοίωση πειράματος Φυσικής	34
Εικόνα: 7 Προβολικό σύστημα εικονικής πραγματικότητας	35
Εικόνα: 8 Αρχική οθόνη λογισμικού	50
Εικόνα: 9 Εσωτερικό Α' τάξης	50
Εικόνα: 10 Αεροδιάδρομος με όλες τις δραστηριότητες	51
Εικόνα: 11 Αεροδιάδρομος με μία αχνή δραστηριότητα	51
Εικόνα: 12 Αρχική οθόνη λογισμικού	53
Εικόνα: 13 Δημιουργία διαγραμμάτων μέσω του λογισμικού	54
Εικόνα: 14 Αρχική οθόνη λογισμικού	55
Εικόνα: 15 Προβολή θεωρίας εκπαιδευτική ενότητα λογισμικού	55
Εικόνα: 16 Περιβάλλον ασκήσεων λογισμικού	55
Εικόνα: 17 Βάση Δεδομένων	56
Εικόνα: 18 Σχεσιακό Μοντέλο Βάσης Δεδομένων	58
Εικόνα: 19 Οθόνη έναρξης Λογισμικού	60
Εικόνα: 20 Σύνδεσμος μετάβασης στη σελίδα επαναπροστολής στοιχείων	61
Εικόνα: 21 Σελίδα επαναπροστολής στοιχείων	61
Εικόνα: 22 Σελίδα επαναπροστολής στοιχείων	62
Εικόνα: 23 Κουμπί μετάβασης στη σελίδα εγγραφής εκπαιδευμένου	62
Εικόνα: 24 Σελίδα εγγραφής εκπαιδευμένου	63
Εικόνα: 25 Κουμπί επαναφοράς πεδίων	63
Εικόνα: 26 Κουμπί επιστροφής στην αρχική σελίδα	64
Εικόνα: 27 Κουμπί ολοκλήρωσης εγγραφής	64
Εικόνα: 28 Εμφάνιση μηνύματος για λάθος email χρήστη	65
Εικόνα: 29 Εμφάνιση μηνύματος για λάθος στοιχεία χρήστη	65
Εικόνα: 30 Εμφάνιση μηνύματος επιτυχής εγγραφής χρήστη	66
Εικόνα: 31 Σελίδα σύνδεσης χρήστη	66
Εικόνα: 32 Μήνυμα λάθος στοιχείων χρήστη	67
Εικόνα: 33 Κεντρική σελίδα εκπαιδευμένου	67
Εικόνα: 34 Επιλογή προβολής θεωρίας πρώτου Κεφαλαίου	68
Εικόνα: 35 Κουμπί μετάβασης χρήστη στο διαγώνισμα του κεφαλαίου που ολοκλήρωσε	69
Εικόνα: 36 Ενημερωτική σελίδα χρήστη πριν το διαγώνισμα	69
Εικόνα: 37 Προβολή πρώτης ερώτησης διαγωνίσματος	70
Εικόνα: 38 Προβολή δεύτερης ερώτησης διαγωνίσματος	70
Εικόνα: 39 Προβολή αποτελεσμάτων διαγωνίσματος	71
Εικόνα: 40 Προβολή προφίλ εκπαιδευμένου	72
Εικόνα: 41 Αλλαγή στοιχείων προφίλ εκπαιδευμένου	72
Εικόνα: 42 Αλλαγή προσωπικού κωδικού εκπαιδευμένου	73
Εικόνα: 43 Μήνυμα λάθος εισαγωγής κωδικού εκπαιδευμένου	73
Εικόνα: 44 Μήνυμα επιτυχής αλλαγή κωδικού πρόσβασης εκπαιδευμένου	74
Εικόνα: 45 Κουμπί μετάβασης στη σελίδα εγγραφής εκπαιδευτή	74
Εικόνα: 46 Σελίδα εγγραφής εκπαιδευτή	75
Εικόνα: 47 Κουμπί επαναφοράς πεδίων	75

Εικόνα: 48	Κουμπί επιστροφής στην αρχική σελίδα	76
Εικόνα: 49	Κουμπί ολοκλήρωσης εγγραφής	76
Εικόνα: 50	Εμφάνιση μηνύματος για λάθος email εκπαιδευτή	77
Εικόνα: 51	Εμφάνιση μηνύματος για λάθος στοιχεία εκπαιδευτή	77
Εικόνα: 52	Εμφάνιση μηνύματος επιτυχής εγγραφής εκπαιδευτή	78
Εικόνα: 53	Σελίδα σύνδεσης εκπαιδευτή	78
Εικόνα: 54	Σελίδα σύνδεσης χρήστη	79
Εικόνα: 55	Κεντρική οθόνη εκπαιδευτή	79
Εικόνα: 56	Κύριο μενού προβολής θεωρίας	80
Εικόνα: 57	Προβολή θεωρίας κεφαλαίου “Επιτρεπτά μεγέθη στη Matlab”	80
Εικόνα: 58	Εισαγωγή νέας θεωρίας	81
Εικόνα: 59	Εισαγωγή νέου κεφαλαίου	81
Εικόνα: 60	Προβολή εισαγωγής νέου κεφαλαίου	82
Εικόνα: 61	Επιτυχής εισαγωγή κεφαλαίου	82
Εικόνα: 62	Εισαγωγή υποκεφαλαίου Β επιπέδου	83
Εικόνα: 63	Τροποποίηση υπάρχουσας θεωρίας	83
Εικόνα: 64	Τροποποίηση υπάρχοντος κεφαλαίου	84
Εικόνα: 65	Τροποποίηση προτεραιότητας κεφαλαίων	84
Εικόνα: 66	Μήνυμα λάθους επιλογής τροποποίησης κεφαλαίου	85
Εικόνα: 67	Τροποποίηση περιεχομένου κεφαλαίου	85
Εικόνα: 68	Προβολή καταχωρημένων ασκήσεων	86
Εικόνα: 69	Επιλογή προβολής ασκήσεων ενός κεφαλαίου	86
Εικόνα: 70	Τροποποίηση άσκησης πολλαπλής επιλογής	87
Εικόνα: 71	Επιλογή μετάβασης στην τροποποίηση ασκήσεων	87
Εικόνα: 72	Βήματα τροποποίηση άσκησης	88
Εικόνα: 73	Προβολή μηνύματος λάθους- τροποποίηση ασκήσεων	88
Εικόνα: 74	Εισαγωγή νέας άσκησης	89
Εικόνα: 75	Εισαγωγή νέας άσκησης σωστού-λάθους	89
Εικόνα: 76	Εισαγωγή στοιχείων άσκησης σωστού-λάθους	90
Εικόνα: 77	Επιτυχής εισαγωγή νέας άσκησης	90
Εικόνα: 78	Μήνυμα λάθους εισαγωγής νέας άσκησης	91
Εικόνα: 79	Προβολή στοιχείων μαθητών του εκπαιδευτή	91

[1] ΕΙΣΑΓΩΓΗ

Η ραγδαία εξέλιξη της επιστήμης των υπολογιστών και της επιστήμης της Πληροφορικής τα τελευταία χρόνια έχει επιφέρει σημαντικές αλλαγές σε πολλούς τομείς της ζωής μας. Ο τομέας της εκπαίδευσης είναι ένας από τους τομείς αυτούς που έχουν αλλάξει σε μεγάλο βαθμό, κυρίως τον τρόπο μάθησης και τη μορφή διδασκαλίας. Η εκπαίδευση μέσω υπολογιστή και του διαδικτύου χρησιμοποιείται ολοένα και περισσότερο ως ένα εναλλακτικό τρόπο μάθησης που εστιάζει στις ανάγκες του κάθε εκπαιδευόμενου.

Σκοπός της παρούσας διατριβής είναι η σχεδίαση και κατασκευή ενός εκπαιδευτικού λογισμικού, το οποίο θα χρησιμοποιηθεί για την εκμάθηση των βασικών στοιχείων της γλώσσας προγραμματισμού MATLAB. Αν και με το πέρασμα του χρόνου έχουν δημιουργηθεί στην Ελλάδα πολλά εκπαιδευτικά λογισμικά, τα οποία συνοδεύουν το πρόγραμμα διδασκαλίας της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, παρατηρείται έλλειψη εκπαιδευτικών λογισμικών τα οποία μπορούν να χρησιμοποιηθούν και από τις πανεπιστημιακές κοινότητες. Το λογισμικό αυτό είναι εστιασμένο στην υποστήριξη του εκπαιδευτικού και του εκπαιδευόμενου κατά τη διάρκεια της διδασκαλίας της MATLAB με κύριο στόχο την εξοικονόμηση όσο το δυνατόν περισσότερου διδακτικού χρόνου. Αποτελεί ουσιαστικά ένα βοηθητικό μέσο του εκπαιδευτικού για την ποιοτική αναβάθμιση της διδασκαλίας ενός τμήματος. Επιπλέον, αποτελεί ένα βοήθημα προς τον εκπαιδευόμενο, ο οποίος επιθυμεί να λάβει τις απαραίτητες γνώσεις πάνω στη MATLAB, αλλά ταυτόχρονα χρειάζεται ένα αποτελεσματικότερο σύστημα εκπαίδευσης, προσαρμοσμένο στις ατομικές του ανάγκες.

Το εκπαιδευτικό λογισμικό διαιρείται σε δύο κύρια μέρη. Το ένα μέρος αφορά την υποστήριξη των βασικών λειτουργιών των εκπαιδευτών και το άλλο μέρος αφορά την υποστήριξη της διδασκαλίας των εκπαιδευόμενων. Οι εκπαιδευτές ύστερα από την επιτυχή εγγραφή τους μπορούν να εισάγουν στη βάση δεδομένων του λογισμικού όλες τις απαραίτητες ενότητες θεωρίας, τις οποίες μπορούν να ταξινομήσουν σε κεφάλαια, αλλά και υποκεφάλαια Α και Β επιπέδου. Μέσω του κεντρικού μενού, δίνεται η δυνατότητα επίσης στους εκπαιδευτές να εισάγουν σε κάθε κεφάλαιο ασκήσεις πολλαπλής επιλογής, σωστού-λάθους και συμπλήρωσης κενού. Όλες οι ενότητες θεωρίας, όπως και οι ασκήσεις, μπορούν να τροποποιηθούν ως προς το περιεχόμενό τους, αλλά και ως προς την προτεραιότητα που κατέχουν. Οι αλλαγές πραγματοποιούνται άμεσα στη βάση δεδομένων και έτσι η ενημέρωση των εκπαιδευόμενων όσων αφορά την ύλη είναι ταχύτατη. Για την εποπτεία της εκπαιδευτικής πορείας των μαθητών, το λογισμικό δίνει τη δυνατότητα στον εκπαιδευτή να επιβλέψει την πρόοδο του κάθε μαθητή ξεχωριστά στα διαγωνίσματα των κεφαλαίων.

Όπως ακριβώς και στους εκπαιδευτές, έτσι και στους εκπαιδευόμενους, το λογισμικό τους δίνει τη δυνατότητα ύστερα από την επιτυχή τους εγγραφή, να εισέλθουν στην κεντρική σελίδα των μαθητών από όπου μπορούν να αναγνώσουν την απαραίτητη θεωρία ταξινομημένη σε κεφάλαια και υποκεφάλαια Α και Β επιπέδου. Ο κάθε εκπαιδευόμενος μπορεί να προβάλλει τη θεωρία ενός κεφαλαίου από την αρχή, επιλέγοντας τον τίτλο του κεφαλαίου, ή να προβεί στην άμεση προβολή των στοιχείων της θεωρίας ενός συγκεκριμένου υποκεφαλαίου. Για την ουσιαστικότερη αποστήθιση της γνώσης, ύστερα από κάθε κεφάλαιο διατίθεται ένα διαγώνισμα, το οποίο περιλαμβάνει ασκήσεις εξάσκησης και πρακτικής τριών διαφορετικών κατηγοριών (πολλαπλής επιλογής, σωστού-λάθους και συμπλήρωσης κενού) επιλεγμένες κάθε φορά με τυχαία σειρά εμφάνισης. Ύστερα από την ολοκλήρωση κάθε διαγώνισματος, προβάλλονται στον εκπαιδευόμενο το πλήθος των σωστών και λανθασμένων απαντήσεων, το ποσοστό επιτυχίας που έλαβε στο διαγώνισμα, ένα μήνυμα που πληροφορεί τον εκπαιδευόμενο αν έχει ολοκληρώσει επιτυχώς το διαγώνισμα ή όχι, οι απαντήσεις που έδωσε καθώς επίσης και η σωστή απάντηση των ερωτήσεων, στην περίπτωση όπου ο εκπαιδευόμενος είχε απαντήσει λανθασμένα. Προβιβάσιμος βαθμός θεωρείται οποιοσδήποτε άνω του πενήντα τοις εκατό (>50%). Στην περίπτωση όπου ο χρήστης έχει ολοκληρώσει το διαγώνισμα με ποσοστό άνω

του πενήντα, τότε απενεργοποιείται η επιλογή έναρξης του συγκεκριμένου διαγωνίσματος και επισημαίνεται ως ολοκληρωμένο, ενώ στην περίπτωση όπου δε λάβει το κατώτατο όριο πρόσβασης το λογισμικό προτρέπει το χρήστη να ξαναδιαβάσει το κεφάλαιο από την αρχή. Αν ο χρήστης ολοκληρώσει επιτυχώς το διαγώνισμα, μπορεί να αναγνώσει τη θεωρία και τις ασκήσεις της επόμενης ενότητας. Επιπρόσθετα το λογισμικό παρέχει τη δυνατότητα προβολής της προόδου του σε κάθε κεφάλαιο ξεχωριστά, αλλά και της συνολικής επίδοσης του συγκριτικά με τους υπόλοιπους εκπαιδευομένους.

Η εφαρμογή επίσης παρέχει ένα τελικό διαγώνισμα στο τέλος όλων των κεφαλαίων, το οποίο εμφανίζεται στο χρήστη αφού έχει ολοκληρώσει με επιτυχία όλα τα προηγούμενα διαγώνισματά κάθε κεφαλαίου. Το διαγώνισμα αυτό αποτελεί το τελικό στάδιο του Εκπαιδευτικού Λογισμικού και αποσκοπεί στον προσδιορισμό του επιπέδου των γνώσεων που έχει αποκτήσει ο μαθητευόμενος.

Το λογισμικό ονομάζεται «Μάθετε τη MATLAB εύκολα και γρήγορα» και αναλύει τις εξής βασικές ενότητες:

1. ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

Γενικά Στοιχεία

Ιστορικά Στοιχεία

Παράθυρο Έναρξης της MATLAB

Παράθυρο εντολών (Command Window)

Παράθυρο τρέχοντος καταλόγου (Current directory)

Παράθυρο χώρου εργασίας (Workspace)

Παράθυρο ιστορικού εντολών (Command History)

Παράθυρο σύνταξης αρχείων (Editor Window)

Παράθυρο γραφημάτων (Figure Window)

Παράθυρο βοήθειας (Help)

Πλήτρο Έναρξης και Αλλαγή Τρέχοντα Φακέλου της MATLAB

Έξοδος από το περιβάλλον της MATLAB

Βασικές Οδηγίες για το Παράθυρο Εντολών "COMMAND WINDOW"

2. ΚΕΦΑΛΑΙΟ 2: ΒΟΗΘΕΙΑ ΤΟΥ MATLAB – ΠΛΗΡΟΦΟΡΙΕΣ ΕΚΔΟΣΗΣ

Εντολή HELP

Εντολές VERSION, VER

3. ΚΕΦΑΛΑΙΟ 3: ΕΠΙΤΡΕΠΤΑ ΜΕΓΕΘΗ ΣΤΗ MATLAB

Αριθμοί

Ακέραιοι Αριθμοί

Δεκαδικοί Αριθμοί

Κλασματικοί Αριθμοί

Μιγαδικοί Αριθμοί

Μονοδιάστατοι Πίνακες – Διανύσματα

Διδιάστατοι Πίνακες

Ακολουθίες Χαρακτήρων

4. ΚΕΦΑΛΑΙΟ 4: ΒΑΣΙΚΕΣ ΠΡΑΞΕΙΣ

Αριθμητικές Πράξεις
Συγκριτικές Πράξεις
Λογικές Πράξεις
Πράξεις Χαρακτήρων
Προτεραιότητα των πράξεων

5. ΚΕΦΑΛΑΙΟ 5: ΜΕΤΑΒΛΗΤΕΣ

Δημιουργία Μεταβλητών – Καταχώριση τιμής σε μεταβλητές
Χειρισμός Μεταβλητών
Εντολή Deal
Εντολή Who
Εντολή Whos
Διαγραφή Μεταβλητών
Απόκρυψη Περιεχομένου Μεταβλητών

6. ΚΕΦΑΛΑΙΟ 8: ΔΙΑΧΕΙΡΙΣΗ ΠΙΝΑΚΩΝ ΚΑΙ ΔΙΑΝΥΣΜΑΤΩΝ

Εμφάνιση στοιχείου πίνακα
Τροποίηση στοιχείου υπάρχον πίνακα
Διαγραφή γραμμών ή στηλών πίνακα
Πράξεις πινάκων
Ανάστροφος πίνακας
Ενσωματωμένες συναρτήσεις δημιουργίας πινάκων
Συνάρτηση zeros(m,n)
Συνάρτηση ones(m,n)
Συνάρτηση eye(m,n)
Συνάρτηση rand(m,n)
Συνάρτηση randn(m,n)
Συνάρτηση magic(m,n)
Ενσωματωμένες συναρτήσεις επεξεργασίας πινάκων
Συνάρτηση size(table)
Συνάρτηση length(table)
Συνάρτηση numel (table)
Συνάρτηση max (table) και min(table)

7. ΚΕΦΑΛΑΙΟ 7: ΕΠΙΤΡΕΠΤΑ ΜΕΓΕΘΗ (ΣΥΝΕΧΕΙΑ)

Πολυδιάστατοι πίνακες
Δομές Δεδομένων (Structures)
Εμφάνιση Εγγραφής
Εμφάνιση τιμών ενός πεδίου
Προσθήκη νέας εγγραφής
Προσθήκη νέων πεδίων σε υπάρχουσα Δομή Δεδομένων
Χειρισμός Δομής Δεδομένων
Εντολή fieldnames

Εντολή isfield
Εντολή rmfield
Πίνακες κελιών (Cell Arrays)
Ανάθεση τιμών σε Πίνακα Κελιών
Εμφάνιση τιμών Κελιού
Προσθήκη νέου Κελιού
Χειρισμός Πίνακα Κελιών
Εντολή cell
Εντολή celldisp
Εντολή cellplot

8. ΚΕΦΑΛΑΙΟ 8: ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

Βασικές μαθηματικές συναρτήσεις
Τριγωνομετρικές συναρτήσεις
Εκθετικές και λογαριθμικές συναρτήσεις

9. ΚΕΦΑΛΑΙΟ 9: ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΤΟΥ ΧΩΡΟΥ ΕΡΓΑΣΙΑΣ

Εντολή save
Η Εντολή diary
Η Εντολή load
Η Εντολή clc
Οι Εντολές quit και exit

10. ΚΕΦΑΛΑΙΟ 10: ΕΙΣΟΔΟΣ ΚΑΙ ΕΞΟΔΟΣ ΔΕΔΟΜΕΝΩΝ

Εντολή input
Η Εντολή disp
Η Εντολή pause
Η Εντολή format

11. ΚΕΦΑΛΑΙΟ 11: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΟ MATLAB

Προγράμματα – Αρχεία Κειμένου (script m-files)
Προγράμματα – Αρχεία Συναρτήσεων (function m-files)
Επικεφαλίδα(header)
Σχόλια(comments)
Εντολές (statements)
Υποσυναρτήσεις (subfunctions)
Εντολή τερματισμού συνάρτησης
Παράδειγμα
Δομές Ελέγχου Ροής Προγράμματος
Η Εντολή if
Η Εντολή for
Η Εντολή while
Η Εντολή switch

12. ΚΕΦΑΛΑΙΟ 12: ΓΡΑΦΙΚΑ ΣΤΟ MATLAB

Παράθυρο γραφικών παραστάσεων - Figure

Συνάρτηση plot

Συνάρτηση stem

Συνάρτηση bar και barh

Μορφοποίηση γραφικών παραστάσεων

Εντολές xlabel και ylabel

Εντολή title

Εντολή legend

Εντολή axis

Εντολή grid ή grid on

Εντολή hold ή hold on

Πολλαπλά γραφήματα σε ένα παράθυρο γραφικών

Αποθήκευση γραφικών παραστάσεων

Τέλος, η εφαρμογή είναι σχεδιασμένη με τέτοιο τρόπο ώστε η πλοήγηση των χρηστών να είναι ευχάριστη και ξεκούραστη. Η εμφάνιση των πληροφοριών προς τους χρήστες γίνεται επίσης με ιδιαίτερη προσοχή προκειμένου να εξασφαλίζεται η σαφής και γρήγορη πληροφόρησή του, χωρίς να είναι απαραίτητη η εξοικείωσή του με τη χρήση παρόμοιων εκπαιδευτικών λογισμικών στον παρελθόν. Τέλος, με τη χρήση κατάλληλων εικόνων και αρχείων βίντεο επιτυγχάνεται η συμπύκνωση των πληροφοριών αποφεύγοντας έτσι την μακροσκελή ανάλυσή τους με τη χρήση κειμένων

Με βάση τα παραπάνω προκύπτουν οι λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος οι οποίες παρουσιάζονται παρακάτω:

Οι Λειτουργικές απαιτήσεις του εκπαιδευτή είναι οι εξής:

- ✓ Δημιουργία νέου λογαριασμού καθηγητή
- ✓ Αίτηση επαναποστολής στοιχείων εγγραφής στο σύστημα
- ✓ Προβολή θεωρίας και ασκήσεων των ήδη υπαρχόντων Κεφαλαίων
- ✓ Δημιουργία νέου κεφαλαίου
- ✓ Δημιουργία νέου υποκεφαλαίου Α επιπέδου για υπάρχον κεφάλαιο
- ✓ Δημιουργία νέου υποκεφαλαίου Β επιπέδου για υπάρχον υποκεφάλαιο Α επιπέδου
- ✓ Δημιουργία άσκησης πολλαπλής επιλογής
- ✓ Δημιουργία άσκησης σωστού - λάθους
- ✓ Δημιουργία άσκησης συμπλήρωσης κενού
- ✓ Τροποποίηση θεωρίας υπάρχοντος κεφαλαίου
- ✓ Τροποποίηση θεωρίας υπάρχοντος υποκεφαλαίου Α επιπέδου
- ✓ Τροποποίηση θεωρίας υπάρχοντος υποκεφαλαίου Β επιπέδου
- ✓ Τροποποίηση προτεραιότητας κεφαλαίων
- ✓ Τροποποίηση προτεραιότητας υποκεφαλαίων
- ✓ Τροποποίηση άσκησης πολλαπλής επιλογής
- ✓ Τροποποίηση άσκησης σωστού - λάθους
- ✓ Τροποποίηση άσκησης συμπλήρωσης κενού
- ✓ Προβολή στοιχείων εκπαιδευομένων

- ✓ Προβολή προόδου εκπαιδευομένων
- ✓ Διαγραφή εκπαιδευομένων

Οι λειτουργικές απαιτήσεις του εκπαιδευόμενου είναι οι εξής:

- ✓ Δημιουργία νέου λογαριασμού εκπαιδευομένου
- ✓ Προβολή λογαριασμού εκπαιδευόμενου
- ✓ Τροποποίηση λογαριασμού εκπαιδευομένου
- ✓ Αίτηση επαναποστολής στοιχείων εγγραφής στο σύστημα
- ✓ Προβολή θεωρίας κεφαλαίων και υποκεφαλαίων
- ✓ Εκπόνηση ασκήσεων κεφαλαίων
- ✓ Προβολή επιδόσεων εκπαιδευομένου στις ασκήσεις
- ✓ Προβολή προόδου επιδόσεων εκπαιδευομένου ανά κεφάλαιο

Οι μη λειτουργικές απαιτήσεις ενός συστήματος, περιγράφουν τις υπηρεσίες που δεν αφορούν άμεσα τις συγκεκριμένες λειτουργίες που παρέχονται από το σύστημα. Πρόκειται για χαρακτηριστικά και περιορισμούς στις υπηρεσίες ή τις λειτουργίες που προσφέρει το σύστημα, όπως χρονικοί περιορισμοί, περιορισμοί της διαδικασίας ανάπτυξης, πρότυπα, κ.ά.

Οι μη λειτουργικές απαιτήσεις του συστήματος είναι οι εξής:

- ✓ Είναι απαραίτητη η ύπαρξη μιας βάσης δεδομένων για την αποθήκευση των χρηστών (εκπαιδευτών – εκπαιδευομένων), των ενοτήτων θεωρίας και ασκήσεων και των επιδόσεων των εκπαιδευομένων.
- ✓ Τα στοιχεία κάθε εκπαιδευτή/εκπαιδευομένου μπορούν να τροποποιηθούν μόνο από τον ίδιο το χρήστη.
- ✓ Ο κωδικός εισόδου στο σύστημα κάθε χρήστη είναι απολύτως προσωπικός και δεν επιτρέπεται να τον δουν άλλοι χρήστες.
- ✓ Μόνο οι εκπαιδευτές έχουν τη δυνατότητα εισαγωγής και τροποποίησης της θεωρίας και των ασκήσεων.
- ✓ Μόνο οι εκπαιδευτές έχουν τη δυνατότητα διαγραφής του λογαριασμού ενός εκπαιδευομένου.
- ✓ Το λογισμικό οφείλει να είναι γρήγορο κατά την προβολή της θεωρίας, των ασκήσεων, των επιδόσεων και των στοιχείων ενός χρήστη.
- ✓ Το λογισμικό οφείλει να είναι αξιόπιστο κατά την αποθήκευση των διαφόρων δεδομένων στη βάση δεδομένων.
- ✓ Η λειτουργία του λογισμικού οφείλει να είναι σταθερή και συνεχής, προκειμένου να είναι ικανοποιητική η χρήση του από τους χρήστες.
- ✓ Το ύψος της παρουσίασης του περιεχομένου και η γραμματοσειρά γραφής οφείλει να διατηρούνται σταθερά σε όλη την εφαρμογή.
- ✓ Η πλοήγηση του χρήστη στο περιβάλλον διεπαφής του λογισμικού οφείλει να είναι εύκολη, ώστε να μην μπερδεύεται ή να χάνεται στις διάφορες επιλογές.
- ✓ Το λογισμικό οφείλει να είναι ικανό να αναγνωρίζει την εισαγωγή μη έγκυρων δεδομένων ή λανθασμένων επιλογών και να εμφανίζει τα αντίστοιχα μηνύματα στο χρήστη.
- ✓ Το λογισμικό οφείλει να καταναλώνει όσο το δυνατόν λιγότερους πόρους του συστήματος.
- ✓ Το λογισμικό οφείλει να είναι ευέλικτο σε περίπτωση αλλαγής των απαιτήσεων.
- ✓ Το λογισμικό οφείλει να είναι εύκολα τροποποιήσιμο ώστε να μπορεί να χρησιμοποιηθεί και για άλλα μαθήματα.

- ✓ Το λογισμικό, εφόσον πρόκειται για μία διαδικτυακή ιστοσελίδα, οφείλει να εκτελείται επιτυχώς σε διαφορετικούς φυλλομετρητές, αλλά και σε διαφορετικά λειτουργικά συστήματα.

Πανεπιστήμιο Πειραιώς

[2] ΑΝΑΣΚΟΠΗΣΗ ΠΕΔΙΟΥ

Στο πρώτο μέρος περιγράφεται η εξέλιξη της Πληροφορικής και η χρήση της στον τομέα της εκπαίδευσης. Παρουσιάζονται οι χρονολογικές φάσεις ένταξης των Τεχνολογιών Πληροφοριών και Επικοινωνίας στην εκπαίδευση (ΤΠΕ) και οι σύγχρονες θεωρίες μάθησης με βάση τις οποίες κατασκευάζονται τα εκπαιδευτικά λογισμικά, αναλύεται ο νέος ρόλος του εκπαιδευτικού και του εκπαιδευόμενου, καθώς ο επηρεασμός τους από την ένταξη των ΤΠΕ στην εκπαίδευση ήταν αναπόφευκτος.

Στο δεύτερο μέρος αναλύονται οι βασικές κατηγορίες των εκπαιδευτικών λογισμικών, ο στόχος, τα πλεονεκτήματα και τα μειονεκτήματα που ενδεχομένως να έχουν, καθώς επίσης και οι βασικές αρχές σχεδίασης ενός ποιοτικού εκπαιδευτικού λογισμικού. Επιπλέον παρουσιάζονται οι κυριότερες έξι μέθοδοι αξιολόγησης των εκπαιδευτικών λογισμικών, με βάση τις θεωρίες μάθησης, τη χρήση πειραματικών μεθόδων, των λιστών αξιολόγησης, των μεθόδων επιθεώρησης, τις μοντελοποιημένες τεχνικές και αναλυτικές μέθοδοι και τη συμμετοχή χρηστών.

Στο τρίτο μέρος παρατίθενται ορισμένα από τα εκπαιδευτικά λογισμικά που χρησιμοποιούνται στο ελληνικό εκπαιδευτικό σύστημα και βρίσκονται στη σελίδα της Δικτυακής Εκπαιδευτικής Πύλης του Υπουργείου Παιδείας και Θρησκευμάτων.

ΜΕΡΟΣ Α: Η ΕΞΕΛΙΞΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ Η ΧΡΗΣΗ ΤΗΣ ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ

2.1. Η εξέλιξη της Πληροφορικής

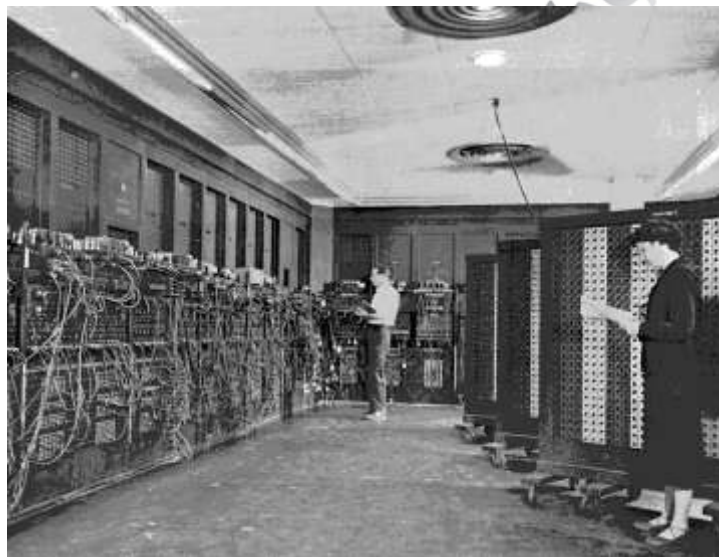
Η Πληροφορική είναι η επιστήμη που ασχολείται και ερευνά την ανάλυση, τη συλλογή, την ταξινόμηση, το χειρισμό, την αποθήκευση, την ανάκτηση, τη μετακίνηση, τη διάδοση, την κωδικοποίηση, και τη μετάδοση συμβολικών αναπαραστάσεων πληροφοριών. Επίσης εξετάζει τη σχεδίαση, υλοποίηση και βελτιστοποίηση αυτοματοποιημένων διατάξεων, συσκευών, υπηρεσιών και συστημάτων συλλογής, αποθήκευσης, επεξεργασίας, εξόρυξης και ανταλλαγής των εν λόγω αναπαραστάσεων. Ως επιστημονικό πεδίο, η Πληροφορική περίπου ταυτίζεται με την Επιστήμη των Υπολογιστών.

Κάθε υπολογιστικό σύστημα, ανεξαρτήτου μεγέθους, αποτελείται από το υλικό μέρος (hardware) και το λογισμικό (software). Τα βασικά στοιχεία του υλικού μέρους του υπολογιστή είναι η κεντρική μονάδα επεξεργασίας (CPU), η κεντρική μνήμη (RAM&ROM), οι μονάδες εισόδου-εξόδου (πληκτρολόγιο, ποντίκι, οθόνη κτλ.), οι περιφερειακές συσκευές (σκληρός δίσκος, δισκέτα, CD-ROM, εκτυπωτής, σαρωτής, μόντεμ κ.τ.λ.). Υπάρχουν διάφορες κατηγορίες ηλεκτρονικών υπολογιστών, οι οποίοι διαθέτουν διαφορετικά χαρακτηριστικά σχετικά με το μέγεθος, τις δυνατότητες (όπως η επεξεργαστική ισχύς) και την αρχιτεκτονική τους, δηλαδή τον τρόπο που τα βασικά τους μέρη συνδέονται και συνεργάζονται μεταξύ τους.

Οι μικροϋπολογιστές αποτελούν την πιο διαδεδομένη κατηγορία υπολογιστών. Στους μικροϋπολογιστές τα βασικά εξαρτήματα, όπως ο επεξεργαστής, η μνήμη κ.ά. βρίσκονται τοποθετημένα σ' ένα τυπωμένο κύκλωμα που ονομάζεται μητρική κάρτα. Εκτός από τον επεξεργαστή και τη μνήμη, πάνω στη μητρική βρίσκονται οι θέσεις επέκτασης στις οποίες τοποθετούνται οι διάφορες κάρτες (γραφικών, ήχου κ.ά.). Στη μητρική επίσης βρίσκονται υποδοχές για τη σύνδεση διαφόρων άλλων συσκευών.

Το λογισμικό του υπολογιστή περιλαμβάνει τα απαραίτητα προγράμματα που δίνουν τις κατάλληλες εντολές για να λειτουργεί το υλικό μέρος. Συνίσταται δε από το λειτουργικό σύστημα (το βασικό πρόγραμμα για τη λειτουργία του ηλεκτρονικού υπολογιστή καθώς και για την επικοινωνία του με τον άνθρωπο) και το λογισμικό εφαρμογών (πακέτα εφαρμογών, γλώσσες προγραμματισμού, εκπαιδευτικό λογισμικό, προγράμματα-εργαλεία κ.ά.).

Ο [ENIAC](#) (Electronic Numerical Integrator and Computer - Ηλεκτρονικός αριθμητικός ολοκληρωτής και υπολογιστής) ήταν ο πρώτος ηλεκτρονικός ψηφιακός υπολογιστής γενικής χρήσης στον κόσμο, ικανός να λύσει ένα πλήρες εύρος υπολογιστικών προβλημάτων. Σχεδιάστηκε και κατασκευάστηκε υπό την εποπτεία των John Mauchly, Καθηγητή Φυσικής και John Presper Eckert, έναν από τους μεταπτυχιακούς φοιτητές του στο Πανεπιστήμιο της Πενσυλβάνια για την έγκαιρη και με ακρίβεια σύνταξη των πινάκων εμβέλειας και τροχιάς για βολές των νέων όπλων από το Εργαστήριο Βαλλιστικής Έρευνας του στρατού των Η.Π.Α., κατά τον Β΄ Παγκόσμιο Πόλεμο (1937-1945). Αν και είχαν ήδη κατασκευαστεί από τα προηγούμενα χρόνια αρκετοί υπολογιστές με ορισμένες από αυτές τις ιδιότητες, όπως ο Z3 του Konrad Zuse, ο ENIAC ήταν ο πρώτος καθαρά ηλεκτρονικός επαναπρογραμματιζόμενος υπολογιστής που μπορούσε να προσομοιώσει μια [Καθολική Μηχανή Τούρινγκ](#). Ο ENIAC έκλεισε στις 9 Νοεμβρίου 1946 για αναβάθμιση μνήμης και συντήρηση και μεταφέρθηκε στο Aberdeen Proving Ground, στην πολιτεία Μέριλαντ, το 1947. Εκεί στις 29 Ιουλίου του ίδιου χρόνου, τέθηκε σε λειτουργία και συνέχισε να λειτουργεί αδιαλείπτως μέχρι την 11:45 μμ στις 2 Οκτωβρίου του 1955.



Εικόνα: 1 - ENIAC ο πρώτος υπολογιστής

Οι υπολογιστές χωρίστηκαν σε τέσσερις γενιές με βάση την τεχνολογία που χρησιμοποιήθηκε για την κατασκευή τους. Στην πρώτη γενιά (1945-1959) χρησιμοποιήθηκαν ηλεκτρονικές λυχνίες κενού, στη δεύτερη (1960-1969) τρανζίστορ, ενώ οι υπολογιστές της τρίτης (1970-1979) και τέταρτης γενιάς (1980 και μετά) βασίστηκαν σε ολοκληρωμένα κυκλώματα μεγάλης κλίμακας (LSI) και πολύ μεγάλης κλίμακας (VLSI) αντίστοιχα. Με την κάθε νέα γενιά υπήρχε μια εξέλιξη και βελτίωση του υλικού (hardware), η οποία είχε ως αποτέλεσμα την αύξηση της ταχύτητας των υπολογισμών και της αποθηκευτικής ικανότητας, όπως και τη μείωση του βάρους και του όγκου τους. Επιπλέον, η βελτίωση του υλικού συνοδεύτηκε από αντίστοιχη ή και μεγαλύτερη βελτίωση του λογισμικού (software), δηλ. των προγραμμάτων τα οποία εκτελεί ο ηλεκτρονικός υπολογιστής.

Οι ψηφιακοί υπολογιστές πέμπτης γενιάς που κυριάρχησαν στην έρευνα και εισέβαλαν κυριολεκτικά στην επιστήμη, στην εκπαίδευση, στη διοίκηση, στη βιομηχανία, στο εμπόριο και στην ιδιωτική κατοικία δρομολογούνται από το 1980. Πρόκειται για κατηγορία υπολογιστών η οποία διαφοροποιείται από τους κλασικούς υπολογιστές στον τρόπο κατασκευής, στην ύπαρξη φιλικής αλληλεπίδρασης με το χρήστη (γραφικά, φωνή κ.ά), στη δυνατότητα καταμεμημένων υπολογισμών και σε άλλα στοιχεία. Από το 1981 το Υπουργείο Διεθνούς Εμπορίου και

Βιομηχανίας της Ιαπωνίας ανακοίνωσε την έναρξη προγράμματος που είχε στόχο την κατασκευή αυτών των υπολογιστικών μηχανημάτων που θα βασίζονταν στην έννοια του λογικού προγραμματισμού. Η καινοτομία αυτού του είδους υπολογιστών είναι το ότι για την επίλυση ενός προβλήματος δεν απαιτείται η διαμόρφωση του προγράμματος σε μορφή που να είναι κατανοητή από τον υπολογιστή, δηλ. δεν χρειάζεται ο εφοδιασμός του υπολογιστή με μια σειρά εντολών που πρέπει να εκτελέσει, αλλά αρκεί η περιγραφή του προβλήματος με βάση γνωστά δεδομένα και σχέσεις που αφορούν το πρόβλημα. Έτσι, ο υπολογιστής προχωρεί στην επίλυση του προβλήματος χρησιμοποιώντας τις ήδη υπάρχουσες πληροφορίες, αλλά και νέες που συνάγει από συγκρίσεις των δεδομένων του.

Οι υπολογιστές πέμπτης γενιάς βρίσκονται ακόμη σε πειραματικό στάδιο. Η έρευνα που διεξάγεται για αυτή την κατηγορία υπολογιστών περιλαμβάνει προσπάθειες για την εξέλιξη των τομέων της κατανόησης της φωνής (speech understanding), της κατανόησης σχημάτων (image understanding), της ανάλυσης κειμένου (text analysis) και από τους υπολογιστές. Ένα μέρος της έρευνας αυτής διεξάγεται στο Τεχνολογικό Ινστιτούτο της Μασαχουσέτης (M.I.T.), όπου και αναπτύχθηκε το ICONIC, το οποίο είναι ένα λογισμικό (software), με συνδυασμό άλλων προγραμμάτων, όπου δίνει τη δυνατότητα στον ηλεκτρονικό υπολογιστή να επεξεργαστεί εντολές που δίνει ο χρήστης μέσω των κινήσεων των χεριών του. Η κατανόηση των χειρονομιών του χρήστη από τον ηλεκτρονικό υπολογιστή επιτυγχάνεται μέσω διάφορων μεθόδων, οι οποίες στηρίζονται στη βιντεοσκόπηση. Από το 1994 η εταιρία Fujitsu πραγματοποιεί έρευνες με στόχο την κατασκευή υπολογιστών, οι οποίοι θα διαβάζουν τη σκέψη του χρήστη και θα δέχονται εντολές μέσω ενός ειδικά κατασκευασμένου κράνους που θα δέχεται τα ηλεκτρικά κύματα του εγκεφάλου του χρήστη και θα τα μεταφέρει στον υπολογιστή. Αν και οι παραπάνω προσπάθειες βρίσκονται σε αρχικό στάδιο, οι προοπτικές για την επίτευξη των παραπάνω στόχων είναι ελπιδοφόρες.

Οι φορητοί ηλεκτρονικοί υπολογιστές που θεωρούνται πλέον αναπόσπαστο κομμάτι της καθημερινής ζωής του σύγχρονου ανθρώπου είναι αποτέλεσμα της εξέλιξης των ολοκληρωμένων κυκλωμάτων. Πρόκειται για υπολογιστές των οποίων τα βασικά μέρη, από τα οποία αποτελούνται, είναι όμοια με αυτά ενός τυπικού συστήματος ηλεκτρονικού υπολογιστή, αλλά ο όγκος τους είναι πολύ μικρότερος και η μεταφορά τους ευκολότερη. Οι φορητοί υπολογιστές γνώρισαν μεγάλη ανάπτυξη στις αρχές της δεκαετίας του 1980, όπου μέχρι εκείνη την εποχή χρησιμοποιούνταν μόνο στα πανεπιστήμια και τα διάφορα ερευνητικά κέντρα. Όλα τα μοντέλα είναι εφοδιασμένα με μπαταρίες, έτσι ώστε να εξασφαλίζεται η αυτονομία τους. Για την αποθήκευση των δεδομένων χρησιμοποιούνται οι σκληροί και οι εύκαμπτοι δίσκοι. Τα πρώτα μοντέλα των φορητών ηλεκτρονικών υπολογιστών που παράχθηκαν ήταν εφοδιασμένα με οθόνες καθοδικής λυχνίας (CRT), με αποτέλεσμα να είναι μεγάλοι σε όγκο και να καταναλώνουν πολλή ενέργεια. Στο μέλλον όμως αντικαταστάθηκαν από τις οθόνες υγρών κρυστάλλων (LCD), οι οποίες έχουν μικρότερο μέγεθος και μπορούν να λειτουργήσουν με ελάχιστη κατανάλωση ηλεκτρικού ρεύματος. Στην πλειοψηφία τους οι οθόνες είναι μονόχρωμες και, κατά συνέπεια, έχουν χαμηλή ευκρίνεια. Όπως ήταν αναμενόμενο, η ανάπτυξη των φορητών υπολογιστών επέφερε και αντίστοιχη ανάπτυξη του λογισμικού που τους συνοδεύει. Επίσης, οι φορητοί ηλεκτρονικοί υπολογιστές μπορούν να συνδέονται και να επικοινωνούν με άλλους υπολογιστές μέσω δικτύων, με την προϋπόθεση να υπάρχει συμβατότητα του λογισμικού και του υλικού που υφίσταται μεταξύ των επικοινωνούντων υπολογιστών.



Εικόνα: 2 - Σύγχρονος φορητός ηλεκτρονικός υπολογιστής (laptop)

Μερικές από τις συχνότερες εφαρμογές των ηλεκτρονικών υπολογιστών είναι η μηχανοργάνωση, μέσω της οποίας επιτυγχάνεται η αποθήκευση και ο χειρισμός βάσεων δεδομένων, η οργάνωση ιατρείων, η οργάνωση και διαχείριση νοσοκομείων και τραπεζών, η καταγραφή και η εξαγωγή των αποτελεσμάτων των εκλογών, η συνεργασία ομάδων εργασίας που βρίσκονται σε διαφορετικούς χώρους, η καταπολέμηση της εγκληματικότητας μέσω της επεξεργασίας των δακτυλικών αποτυπωμάτων, η τεχνολογική υποστήριξη ιατρείων, στρατού, χρηματιστηρίου κ.λπ.

Αναμενόμενη, λοιπόν, είναι και η εισαγωγή των υπολογιστών στον τομέα της εκπαίδευσης. Ήδη από το 1983 οι υπολογιστές αποτελούν εργαλείο στη διδασκαλία διαφόρων μαθημάτων. Εδώ και χρόνια το μάθημα της πληροφορικής έχει ενταχθεί στο βασικό πρόγραμμα σπουδών και η χρήση ηλεκτρονικών υπολογιστών κατά τη διάρκεια της εκπαίδευσης ολοένα και αυξάνεται. Στα πλαίσια της εκπαίδευσης μέσω ηλεκτρονικών υπολογιστών, ο μαθητής διδάσκεται τις βασικές γνώσεις για τους ηλεκτρονικούς υπολογιστές, ενώ του δίνεται η δυνατότητα να έχει πρόσβαση σε μεγάλο όγκο βιβλιογραφικού και εκπαιδευτικού υλικού.

2.2. Ο όρος της Τεχνολογίας Πληροφοριών και Επικοινωνίας (ΤΠΕ) στην εκπαίδευση και η σημασία του

Συνήθως αντί του όρου Πληροφορική χρησιμοποιούνται πλέον σε ευρεία κλίμακα ο όρος Τεχνολογίες της Πληροφορίας και των Επικοινωνιών (Information and Communication Technology (ICT)). Η εξέλιξη της Πληροφορικής και των Τηλεπικοινωνιών συντέλεσαν στην εισαγωγή της Πληροφορικής στην εκπαίδευση, όπως προαναφέρθηκε, δεδομένου ότι μαθητές λαμβάνουν όλες τις απαιτούμενες δεξιότητες και τεχνολογικές γνώσεις που θα τους επιτρέψουν να ανταποκριθούν στις απαιτήσεις της σύγχρονης κοινωνίας.

Ως Τεχνολογίες Πληροφορίας και Επικοινωνιών (ΤΠΕ) εννοούμε τις μεθόδους, τις εφαρμογές και τα προϊόντα της σύγχρονης επιστήμης και τεχνολογίας, που αφορούν στη συγκέντρωση και ηλεκτρονική κωδικοποίηση, επεξεργασία, ταξινόμηση, επιλεκτική και συνδυαστική ανάσυρση, διακίνηση και διάχυση, γνωστοποίηση και μελέτη της οποίας πληροφορίας σε κάθε της μορφή (μέσο), ήτοι κειμένου, αριθμού, γραφήματος, ήχου, εικόνας, βίντεο. (Παπασταματίου, 2008).

Βασικό στοιχείο των ΤΠΕ αποτελεί ο ηλεκτρονικός υπολογιστής, ο οποίος είναι εμπλουτισμένος, στις μέρες μας, με δυνατότητες πολυμέσων, τα οποία επιτρέπουν την επεξεργασία και αποθήκευση κειμένου, ήχου, βίντεο καθώς επίσης και το μεταξύ τους συνδυασμό. Επιπλέον, εκτός από τις δυνατότητες πολυμέσων, ο ηλεκτρονικός υπολογιστής διαθέτει και δυνατότητες υπερμέσων, δηλαδή τη διασύνδεση του υπερκειμένου και των

πολυμέσων μέσω συνδέσμων (links), τα οποία βελτιώνουν έτσι την αλληλεπίδραση με το χρήστη.

2.2.1. Χρονολογικές φάσεις ένταξης των ΤΠΕ στην Εκπαίδευση

Η εξέλιξη της εισαγωγής και της ένταξης της Πληροφορικής στο εκπαιδευτικό σύστημα υπήρξε αρκετά γρήγορη τα τελευταία 25-30 χρόνια. Παρόλα αυτά δεν έλειψαν οι διαφορετικές προσεγγίσεις αλλά και οι πολλαπλές θεωρήσεις για τη θέση των ΤΠΕ στην εκπαίδευση. Το βέβαιο είναι ότι οι χρήσεις αυτών των μέσων στην εκπαιδευτική διαδικασία έχουν επιτρέψει σημαντικές εκπαιδευτικές εφαρμογές τόσο στη διδακτική και μαθησιακή διαδικασία όσο και στη διαχείριση του σχολικού περιβάλλοντος (Κόμης, 2004).

πριν 1970	Εκπαιδευτική τεχνολογία και διδακτικές μηχανές
1970-1980	<p>Πληροφορική (τεχνοκεντρική) προσέγγιση</p> <p>Η Πληροφορική ως αυτόνομο γνωστικό αντικείμενο που μπορεί να ενταχθεί στο πρόγραμμα σπουδών και να διδαχθεί σε διάφορες βαθμίδες της εκπαίδευσης.</p> <p>Προσανατολίζεται στη διδασκαλία προγραμματισμού.</p> <p>Βασίζεται στις απόψεις της θεωρίας της συμπεριφοράς.</p> <p>Απομονωμένη <u>τεχνική προσέγγιση ή κάθετη</u>.</p>
1980-1989	<p>Ολοκληρωμένη προσέγγιση</p> <p>Η Πληροφορική και οι ΤΠΕ ως μέσο γνώσης, έρευνας και μάθησης που διαπερνά όλα τα γνωστικά αντικείμενα. Ως έκφραση μιας ολιστικής, διαθεματικής προσέγγισης της μάθησης (οριζόντια)</p>
1990-σήμερα	<p>Πραγματολογικό μοντέλο ή προσέγγιση</p> <p>Ως συνδυασμός των δύο προηγούμενων προσεγγίσεων.</p> <p>Η πληροφορική και ο ΤΠΕ ως στοιχείο της γενικής κουλτούρας αλλά και κοινωνικό φαινόμενο. (Μακράκης, Κοντογιαννοπούλου -Πολυδωρίδη), 1995</p> <p>Εφικτή ή μεικτή προσέγγιση.</p>

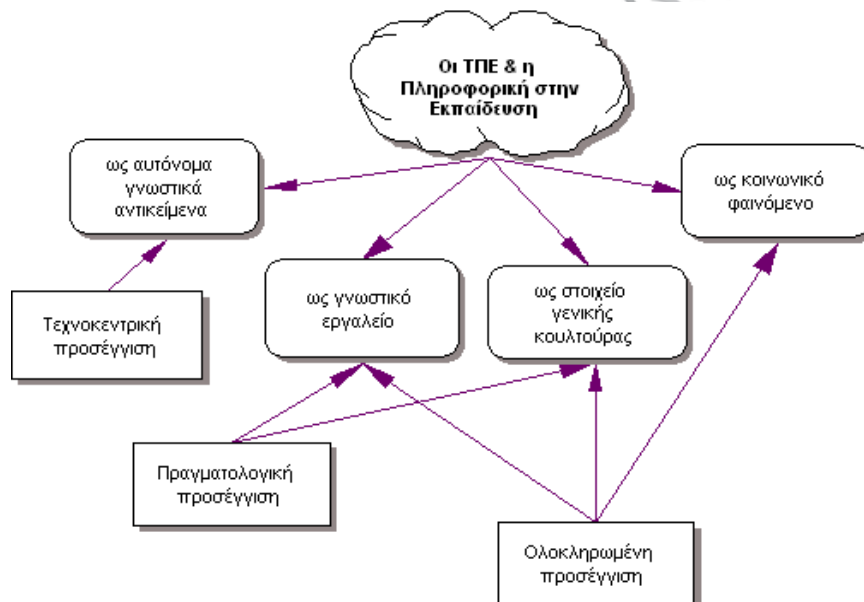
Πίνακας 1 - Χρονολογικές φάσεις ένταξης των ΤΠΕ στην εκπαίδευση

Στα αρχικά στάδια ένταξης των ΤΠΕ στην εκπαίδευση προηγήθηκε μία περίοδος, η οποία χαρακτηρίζεται από την προσπάθεια εισαγωγής και ένταξης των διαφόρων πολυμέσων και τεχνολογιών στην εκπαίδευση. Στη συνέχεια, το 1970 υπήρξε μια σημαντική χρονιά για την εισαγωγή της πληροφορικής στο εκπαιδευτικό σύστημα. Διοργανώνονται παγκόσμιες συναντήσεις και συνέδρια με θέμα την εισαγωγή των υπολογιστών στην εκπαίδευση. Παρότι οι θέσεις που εκφράστηκαν σε αυτά τα σεμινάρια υιοθετούσαν την άποψη της εισαγωγής των

ηλεκτρονικών υπολογιστών σε όλο το φάσμα της εκπαιδευτικής διαδικασίας, το πρακτικό αποτέλεσμα όλης της φάσης αυτής χαρακτηρίζεται κυρίως από τη μάθηση για τους υπολογιστές παρά τη μάθηση με τους υπολογιστές.

Στη συνέχεια, κατά την περίοδο 1970-1980 οι ΤΠΕ αναγνωρίζονται και χρησιμοποιούνται ως γνωστικό αντικείμενο το οποίο μπορεί να διδαχθεί σε όλες τις βαθμίδες της εκπαίδευσης. Η προσέγγιση αυτή είναι γνωστή ως τεχνοκρατική. Τη δεκαετία του 1980, η Πληροφορική αντιμετωπίζεται ως ένα μέσο προσέγγισης όλων των μαθημάτων.

Σύμφωνα με την ολοκληρωμένη προσέγγιση, την περίοδο 1980-1989, οι μαθητές μούονται στην έρευνα, στην απόκτηση γνώσης και στη μάθηση μέσω των ΤΠΕ. Η τελευταία περίοδος ξεκινά από το 1990 και συνεχίζει ακόμα και στις μέρες μας. Η προσέγγιση αυτή ονομάζεται πραγματολογική και σύμφωνα με αυτήν, η πληροφορική και οι ΤΠΕ αποτελούν κοινωνικό φαινόμενο και σημαντικό στοιχείο της εκπαίδευσης και της γενικότερης κουλτούρας. Έτσι, οι δεκαετίες του 1980 και του 1990 χαρακτηρίζονται από τη μαζική εισαγωγή και ένταξη της πληροφορικής και των τεχνολογιών στην εκπαίδευση και σε κάθε πτυχή της ζωής μας γενικότερα. Όμως, λίγα λογισμικά ήταν άξια λόγου προκειμένου να χρησιμοποιηθούν αποτελεσματικά μέσα στην τάξη, ενώ η πλειονότητά τους ήταν τύπου ερωτήσεων πολλαπλών επιλογών (multiple choice) (Παπαδόπουλος Γ., 1999).



Εικόνα: 3 Οι ΤΠΕ και η Πληροφορική στην Εκπαίδευση

2.2.2. Χρονολογικές φάσεις εισαγωγής και ανάπτυξης της τεχνολογίας στην Εκπαίδευση

Κατά τη δεκαετία του 1990 παρατηρείται μείωση του κόστους των συσκευών του υπολογιστή, το διαδίκτυο εξελίσσεται ραγδαία και αναπτύσσονται τα πολυμέσα. Οι ΤΠΕ έχουν ενταχθεί στις διάφορες βαθμίδες της εκπαίδευσης και χρησιμοποιούνται ως μέσα για την επίτευξη της διδασκαλίας.

Περίπου δέκα χρόνια μετά την εισαγωγή της πληροφορικής στην εκπαίδευση, εξετάζεται έντονα το ενδεχόμενο της πληροφορικής ως αυτόνομο διδακτικό αντικείμενο στην πρωτοβάθμια και δευτεροβάθμια εκπαίδευση. Ωστόσο, τα αποτελέσματα των ερευνών πάνω σε αυτό το ερώτημα είναι περισσότερο αμφισβητήσιμα παρά ξεκάθαρα.

	<u>Πρώτη φάση</u>	<u>Δεύτερη Φάση</u>	<u>Τρίτη Φάση</u>	<u>Τέταρτη Φάση</u>
Χαρακτηριστικά	(Media και τεχνολογίες) Πριν από το 1970	(Η πληροφορική προσέγγιση) 1970-1980	(Μέσο – Αντικείμενο εκπαίδευσης) 1980 - 1990	(Τεχνολογίες ως μέσο) Μετά το 1990
Επίπεδο	Γυμνάσια - Λύκεια	Λύκειο	Δημοτικά, Γυμνάσια, Λύκεια	Όλα τα επίπεδα
Τύποι Δράσης	Πειραματισμοί	Έρευνες	Ανάπτυξη προωθούμενη από το κράτος	Τοπική δράση
Προσανατολισμοί	Οπτικοακουστικά μέσα / προγραμματισμένη διδασκαλία	Πληροφορική = Τρόπος Σκέψης	Πληροφορική : Αντικείμενο ή μέσο	Πληροφορική Μέσο – Πολυμέσα
Κατάρτιση εκπαιδευτικών	-	Μακράς διάρκειας κατάρτιση	Μακράς διάρκειας κατάρτιση	Σύντομη κατάρτιση
Λογισμικό	-	Λογισμικό «Παιδαγωγικής Έρευνας»	Λογισμικό παραγωγή της πολιτείας	Λογική της αγοράς
Εξοπλισμός	Οπτικό-ακουστικός εξοπλισμός	Κάποιοι μικροϋπολογιστές	Διάφοροι Τύποι υπολογιστών	Συγκέντρωση γύρω από το PC

Πίνακας 2 - Φάσεις εισαγωγής και ανάπτυξης της τεχνολογίας στην Εκπαίδευση

Σύμφωνα με G. – L. Baron, E. Bruillard, 1996

Όπως συμπεραίνεται, η πρώτη περίοδος εισαγωγής των νέων τεχνολογιών στην εκπαίδευση, χαρακτηρίζεται από την προσπάθεια της συγκρότησης ανάμεσα στην τεχνολογική ανάπτυξη και σε συγκεκριμένους στόχους και ανάγκες, ενώ όλα τα συμπεράσματα εμπεριέχουν μια σειρά από ερωτήματα τα οποία παρουσιάζουν ιδιαίτερο ερευνητικό ενδιαφέρον. Ορισμένα από τα ερωτήματα αυτά είναι το αν η εκπαίδευση είναι έτοιμη να δεχθεί τις αναγκαίες αλλαγές από την εισαγωγή της πληροφορικής, αλλά και το αν μπορεί ο ρόλος του εκπαιδευτικού να μετασχηματιστεί στις νέες συνθήκες μάθησης, αφού δεν είναι πλέον ο μόνος κάτοχος και φορέας αυτής της γνώσης.

2.2.3. Μοντέλα Ένταξης των Νέων Τεχνολογιών στην Εκπαίδευση

Τεχνοκεντρική προσέγγιση

Το πρότυπο αυτό χαρακτηρίζεται από τεχνοκρατικό ντετερμινισμό και έχει ως βασική επιδίωξη την απόκτηση γνώσεων πάνω στη λειτουργία των υπολογιστών και την εισαγωγή στον προγραμματισμό τους. Σκοπός δηλαδή της εισαγωγής των νέων τεχνολογιών στην εκπαίδευση είναι ο τεχνολογικός αλφαριθμητισμός, η εκμάθηση του χειρισμού, της λειτουργίας και του προγραμματισμού του Η/Υ. Η Πληροφορική στα πλαίσια αυτά θεωρείται ως αυτοτελές γνωστικό αντικείμενο, και στη διεθνή βιβλιογραφία απαντάται με τον όρο απομονωμένη τεχνική προσέγγιση ή κάθετη προσέγγιση. Εφαρμόζεται στη Δευτεροβάθμια Εκπαίδευση και διδάσκεται ως ξεχωριστό μάθημα. Είναι μονοδιάστατο καθώς στοχεύει μόνο στον τεχνολογικό αλφαριθμητισμό. Η έλλειψη παιδαγωγικής λειτουργίας καθιστά το μοντέλο αυτό ακατάλληλο για την Πρωτοβάθμια Εκπαίδευση. Περιλαμβάνει προγράμματα εξάσκησης και πρακτικής εφαρμογής (drill & practice).

Ολοκληρωμένο - Ενσωματωμένο (Integrated) Πρότυπο

Ένταξη και ενσωμάτωση των νέων τεχνολογιών μέσα σε όλα τα μαθήματα ως έκφραση μιας ολιστικής, διαθεματικής προσέγγισης της μάθησης (ολοκληρωμένη προσέγγιση). Οι νέες δηλαδή τεχνολογίες θεωρούνται ένα διαθεματικό εργαλείο.

Το πρότυπο αυτό εμφανίστηκε σχετικά πρόσφατα και χαρακτηρίζεται από το ότι η διδασκαλία της χρήσης των νέων τεχνολογιών και η χρήση τους ενσωματώνεται στα επιμέρους γνωστικά αντικείμενα του προγράμματος σπουδών (αποδίδεται με τον όρο οριζόντια ή ολιστική προσέγγιση).

Σύμφωνα με την προσέγγιση αυτή, τα θέματα που αφορούν στους υπολογιστές και στις ΤΠΕ γενικότερα, διδάσκονται μέσα από όλα τα γνωστικά αντικείμενα του σχολείου και δεν συνιστούν ιδιαίτερο γνωστικό αντικείμενο. Ο υπολογιστής χρησιμοποιείται ως εργαλείο αναζήτησης και ανεύρεσης πληροφοριών και ως εργαλείο επικοινωνίας και διεκπεραίωσης καθημερινών εργασιών. Στο πλαίσιο αυτό διδάσκεται και η χρήση του. Οι υποστηρικτές αυτής της προσέγγισης πιστεύουν ότι η διάσπαρά της διδασκαλίας και της χρήσης της πληροφορικής σε όλο το φάσμα του προγράμματος σπουδών και όχι η ένταξή του σε ένα ιδιαίτερο αντικείμενο, μπορεί να βοηθήσει την ουσιαστική και από κοινού δημιουργική συμμετοχή εκπαιδευτικών και μαθητών στην εκπαιδευτική διαδικασία.

Η προσέγγιση αυτή προϋποθέτει σημαντικά διαφορετικές εκπαιδευτικές αντιλήψεις, τόσο στην επιλογή της γνώσης και της διδακτικής πρακτικής όσο και στην εκπαίδευση και την κατάρτιση των εκπαιδευτικών και στην υλικοτεχνική υποδομή.

Πραγματολογικό πρότυπο

Η πραγματολογική προσέγγιση, στην τρέχουσα τουλάχιστον εκδοχή της, συνιστά συνδυασμό των προηγούμενων προσεγγίσεων (τεχνοκεντρικής -ολοκληρωμένης). Συνιστά μια μεταβατική, "εφικτή" λύση, απαραίτητη για ένα τουλάχιστον χρονικό διάστημα μέχρι την πλήρη ένταξη των τεχνολογιών σε όλο το αναλυτικό πρόγραμμα. Το πρότυπο αυτό, χαρακτηρίζεται από τη διδασκαλία ενός αμιγούς μαθήματος γενικών γνώσεων πληροφορικής και την προοδευτική ένταξη της χρήσης των νέων τεχνολογιών ως μέσο στήριξης της μαθησιακής διαδικασίας σε όλα τα γνωστικά αντικείμενα του προγράμματος σπουδών.

Στη βιβλιογραφία αποδίδεται και με τον όρο εφικτή ή μικτή προσέγγιση. Η έμφαση στα πλαίσια αυτής της προσέγγισης, δίνεται στις γνωστικές και τις κοινωνικές διαστάσεις της χρήσης

της πληροφορικής στην εκπαιδευτική διαδικασία. Συνδυάζει τα παιδαγωγικά πλεονεκτήματα της ολοκληρωμένης προσέγγισης με την ανάγκη για τεχνολογικό αλφαριθμητισμό.

Η έκθεση του J. C. Simon (δεκαετία του 80)

Η έκθεση αυτή, προτείνει την κατάρτιση για όλους στην Πληροφορική, συνιστώνται την συνέχιση προηγούμενων εμπειριών στο επίπεδο γυμνασίου και λυκείου.

Τα πληροφοριακά μέσα για εκπαιδευτική χρήση πρέπει να αποτελέσουν αντικείμενο διεπιστημονικών ερευνών και προσαρμογών. Ως συμπέρασμα, η έκθεση θεωρεί ότι, με τη μέχρι τότε κατάσταση των ερευνών, τα πληροφορικά εργαλεία δεν μπορούν να θεωρηθούν σαν καθολικό παιδαγωγικό μέσο, εντούτοις παρουσιάζουν εξαιρετικό παιδαγωγικό ενδιαφέρον σε ποικίλες και ιδιαίτερες περιπτώσεις.

Κάτω από το πρίσμα αυτό, προτείνονται δύο δρόμοι ερευνών: Η Διδασκαλία με τη βοήθεια υπολογιστή (Δι.Β.Υ.) και η γλώσσα LOGO.

Ένα τυπικό πρόγραμμα Δι.Β.Υ. περιέχει:

- την παρουσίαση ενός διδακτικού αντικειμένου και την κατάλληλη ερώτηση,
- την απάντηση από τον μαθητή,
- την αντίδραση του προγράμματος που μπορεί να είναι γραμμική ή με διακλαδώσεις, ανάλογα με την απάντηση που έχει δοθεί και να περιέχει συμπληρωματικές πληροφορίες που καλύπτουν το μαθησιακό κενό.

Έκθεση Schwartz (1981)

Η έκθεση αυτή προσδιορίζει τους στόχους της πληροφορικής στην πρωτοβάθμια εκπαίδευση. Οι στόχοι αυτοί προσανατολίζονται προς δύο κύριες κατευθύνσεις: ο υπολογιστής ως εργαλείο μάθησης και ως στοιχείο της γενικής κουλτούρας.

Όσον αφορά την εφαρμογή των παραπάνω στόχων, η έκθεση κάνει τις ακόλουθες προτάσεις:

Η πληροφορική, ως παιδαγωγικό μέσο συνεπάγεται την κατάρτιση με την βοήθεια της πληροφορικής (το παιδαγωγικό έρεισμα οδηγεί επιπλέον στην ανάπτυξη της πολιτισμικής διάστασης των πληροφορικών μέσων).

Η πληροφορική, ως παιδαγωγικός στόχος συνεπάγεται την κατάρτιση στην πληροφορική. Συγκεκριμένα, η έκθεση προτείνει, αρχίζοντας από την τρίτη τάξη του δημοτικού σχολείου, να χρησιμοποιούν οι μαθητές κατά ομάδες των δύο ατόμων, ως ισοδύναμο μισής εκπαιδευτικής ώρας, κάθε μέρα τον υπολογιστή.

Η εποχή της γενικευμένης εισαγωγής

Η ένταξη των ΤΠΕ, αποτελεί γενικευμένη ευρωπαϊκή εκπαιδευτική πολιτική στην εκπαιδευτική μαθησιακή και διδακτική διαδικασία. Ο ρόλος των εκπαιδευτικών θεωρείται καθοριστικής σημασίας για την ένταξη των ΤΠΕ. Ατομικά χαρακτηριστικά του εκπαιδευτικού, όπως η αυτοαποτελεσματικότητα (self-efficacy) ως προς τους υπολογιστές (Paraskeva, Boufa & Paragianni, 2008), οι στάσεις προς την τεχνολογία (Bullock 2004) και το φύλο του εκπαιδευτικού (Sharpa & Ferrari, 2003) φαίνεται ότι συνδέονται με τη χρήση των ΤΠΕ στη διδασκαλία. Όλο και περισσότερες χώρες ασχολούνται με την εισαγωγή και ενσωμάτωση των ΤΠΕ στα εκπαιδευτικά τους συστήματα, γεγονός που δείχνει το σημαντικό ρόλο που μπορούν να διαδραματίσουν στην εκπαίδευση.

Η ενσωμάτωση των Τεχνολογιών Πληροφορίας και Επικοινωνίας (ΤΠΕ) στην εκπαίδευση αποτελεί θέμα υψηλού ερευνητικού ενδιαφέροντος. Τις τελευταίες δεκαετίες όλες οι οικονομικά ανεπτυγμένες χώρες επιχειρούν - παράλληλα με το μάθημα της πληροφορικής- να ενσωματώσουν τη χρήση των νέων τεχνολογιών στη διδασκαλία των διαφόρων γνωστικών αντικειμένων του αναλυτικού προγράμματος. Ωστόσο, η ένταξη των ΤΠΕ στην εκπαιδευτική

πράξη δεν έχει επιτευχθεί κατά τρόπο αποτελεσματικό. Μέχρι και σήμερα πραγματοποιούνται ειδικά συνέδρια με θέμα την ένταξη και χρήση των ΤΠΕ στην εκπαιδευτική διαδικασία δίνοντας την ευκαιρία σε ενεργούς εκπαιδευτικούς και ερευνητές, με ενδιαφέρον για τις ΤΠΕ στην Εκπαίδευση, να παρουσιάσουν μέσα από ένα επιστημονικά αξιόπιστο πρίσμα τις θεωρητικές, αλλά κυρίως τις έμπρακτες προτάσεις τους. Σκοπός τους είναι η ανάδειξη σημαντικών θετικών και αρνητικών σημείων διαφόρων μεθοδολογιών, εργαλείων αλλά και πρακτικών που ενσωματώνουν τις ΤΠΕ στην Εκπαίδευση.

Τα εκπαιδευτικά συστήματα των ανεπτυγμένων χωρών έχουν σημαντικά επηρεασθεί από τις ραγδαίες τεχνολογικές εξελίξεις των τελευταίων χρόνων. Από τα πανεπιστήμια στα λύκεια, τις σχολές αρχικής επαγγελματικής κατάρτισης στο δημοτικό, κανένας χώρος δεν έμεινε χωρίς να εντάξει - στον ένα ή στον άλλο βαθμό - τα νέα εργαλεία αναπαράστασης και σκέψης. Το κίνημα αυτό, που επιταχύνεται από την εμφάνιση όλο και πιο φιλικών συστημάτων επικοινωνίας χρήστη - μηχανής και λογισμικού, καθώς και από τη γενίκευση της συζήτησης γύρω από τα πληροφορικά μέσα, αντλεί την έμπνευση και το δυναμισμό του μέσα από κοινωνικές, οικονομικές, πολιτικές, παιδαγωγικές και πολιτισμικές ανησυχίες, οι οποίες μπορούν να ιεραρχηθούν σε πολλαπλά επίπεδα.

2.2.4. Χαρακτηριστικά των ΤΠΕ σχετικά με την εκπαίδευση

Όπως χαρακτηριστικά ανέφερε ο Μπαμπινιώτης σε ένα άρθρο του, το 2000, χωρίς τις τεχνολογίες, χωρίς την πληροφορική και τις ποικίλες εφαρμογές της στην «κοινωνία των πληροφοριών όπου ζούμε», ιδίως στην κοινωνία του 21^{ου} αιώνα, δεν μπορεί να νοηθεί ανάπτυξη της παιδείας. Η αξιοποίηση των ΤΠΕ ενδυναμώνει τη διάσταση της μάθησης και αποτελεί πλέον σημαντικό ρόλο στην εκπαίδευση. Οι ΤΠΕ μας παρέχουν μία νέα γενιά γνωστικών εργαλείων, τα οποία έχουν τη δυνατότητα να ενισχύσουν τα εκπαιδευτικά προγράμματα χάρη στα χαρακτηριστικά τους (Κωνσταντίνου, 2005).

Το πρώτο χαρακτηριστικό των ΤΠΕ που σχετίζεται με τον τομέα της εκπαίδευσης είναι τα κίνητρα για μάθηση. Όταν τα περιβάλλοντα μάθησης υποστηρίζονται από τις ΤΠΕ τότε προκαλούν μεγαλύτερο ενδιαφέρον και αυξάνουν τα κίνητρα για μάθηση σε σύγκριση με τα παραδοσιακά περιβάλλοντα μάθησης. Αν και οι απόψεις των ερευνητών διαφέρουν ως προς την εξήγηση αυτού του φαινομένου, αρκετοί φαίνεται να συμφωνούν στο γεγονός ότι σημαντικό πλεονέκτημα αποτελεί η καινοτομία και η κοινωνική ελκυστικότητα των υπολογιστών.

Ένα από τα πιο σημαντικά χαρακτηριστικά των ΤΠΕ είναι ο συνδυασμός εικόνων και κειμένου. Η δυνατότητα που δίνει η χρήση των ΤΠΕ στην εκπαίδευση ώστε να συνδυαστούν κείμενα, εικόνες, ηχητικό υλικό και βίντεο αποτελεί σημαντικό πλεονέκτημα, αφού ενεργοποιούνται και συνεπώς αξιοποιούνται σχεδόν όλες οι αισθήσεις του μαθητευόμενου με αποτέλεσμα την ουσιαστικότερη αφομοίωση της ύλης.

Ανάλογα με το εκπαιδευτικό λογισμικό, αλλά και τον εκπαιδευτικό, οι ΤΠΕ μπορούν να συμβάλλουν στην ενθάρρυνση της ενεργητικής μάθησης. Σύμφωνα με τον κώνο των εμπειριών του E. Dale μετά από δύο εβδομάδες τείνουμε να θυμηθούμε 10% απ' ό,τι διαβάσαμε, 20% απ' ό,τι ακούσαμε, 30% απ' ό,τι είδαμε, 50% απ' ό,τι είδαμε και ακούσαμε και 90% αυτών που είπαμε και κάναμε. Συνεπώς η ενεργητική μάθηση είναι αποτελεσματικότερη μέθοδος σε σχέση με την παθητική μάθηση, την οποία ακολουθεί ένα παραδοσιακό περιβάλλον μάθησης που δεν αξιοποιεί τις ΤΠΕ.

Εξίσου σημαντικό χαρακτηριστικό των ΤΠΕ αποτελεί και ο έλεγχος της μάθησης. Γενικά, είναι ευκολότερα αντιληπτό το τί πρέπει να αφομιώσουμε σε ένα σύγχρονο περιβάλλον μάθησης που χρησιμοποιεί τις ΤΠΕ, σε σχέση με ένα παραδοσιακό περιβάλλον όπου ο καθηγητής αποτελεί τη μοναδική πηγή πληροφοριών. Οι εκπαιδευτικοί πρέπει να υποστηρίζουν και να κατευθύνουν τις αυτόνομες δραστηριότητες των μαθητών, καθιστώντας τους ικανούς να μπορούν να σκέπτονται και να λειτουργούν αυτόνομα, δημιουργώντας τις προϋποθέσεις για «δια βίου μάθηση» (Βοσνιάδου, 2006).



Εικόνα: 4 Ο κώνος των εμπειριών του E.Dale

Με τη χρήση των ΤΠΕ είναι εφικτή η δημιουργία περιβαλλόντων μάθησης που αντικατοπτρίζουν την πραγματική ζωή με τρόπους που δύσκολα υλοποιούνται στις πιο παραδοσιακές μεθόδους διδασκαλίας. Έρευνες έχουν αποδείξει ότι οι άνθρωποι μαθαίνουν καλύτερα όταν συμμετέχουν οι ίδιοι σε δραστηριότητες που θεωρούν χρήσιμες για την πραγματική ζωή και έχουν σχέση με την κουλτούρα τους. Εξίσου σημαντικό είναι το υλικό να έχει οργανωθεί γύρω από γενικές αρχές και εξηγήσεις, αντί να βασίζεται στην απομνημόνευση απομονωμένων στοιχείων και διαδικασιών.

Σημαντικό χαρακτηριστικό των ΤΠΕ στην εκπαίδευση είναι το γεγονός ότι ενθαρρύνουν τη συνεργατική μάθηση. Με την ευρύτερή της έννοια, η συνεργατική μάθηση μπορεί να οριστεί ως η από κοινού εργασία πάνω σε ένα συγκεκριμένο θέμα με τρόπο τέτοιο ώστε να προωθείται η ατομική μάθηση μέσω των συνεργατικών διεργασιών. Στη συνεργατική μάθηση, η γνώση δεν είναι κάτι που παραδίδεται στους μαθητές, παρά κάτι που προκύπτει από τον ενεργό διάλογο μεταξύ αυτών που προσπαθούν να κατανοήσουν και να χρησιμοποιήσουν έννοιες και τεχνικές. Επιπλέον, η συνεργασία έχει ως αποτέλεσμα ένα επίπεδο γνώσης της κοινότητας που είναι μεγαλύτερη από το άθροισμα των γνώσεων του κάθε μέλους ξεχωριστά: «Οι συνεργατικές δραστηριότητες οδηγούν στην ανερχόμενη γνώση, που είναι το αποτέλεσμα της διάδρασης μεταξύ (όχι της συνάθροισης) των γνώσεων και απόψεων όλων όσων συμμετέχουν στο σχηματισμό της» [Whipple 1987]. Έτσι, η αξιοποίηση των ΤΠΕ ενδυναμώνει την κοινωνική διάσταση της μάθησης και ενισχύει την αλληλεπίδραση των μαθητών. Οι χρήσεις των ΤΠΕ που θα μπορούσαν να υποστηρίξουν τη συνεργατική μάθηση είναι τα Περιβάλλοντα Ανοιχτής Μάθησης, τα Συστήματα Καθοδηγούμενης Ανακάλυψης, τα διάφορα εργαλεία Δόμησης και Υποστήριξης της Γνώσης αλλά και τα Εργαλεία Παροχής και Μετάδοσης Πληροφοριών και Επικοινωνίας.

2.3. Οι εφαρμογές της Πληροφορικής στην Εκπαίδευση

Οι ΤΠΕ και γενικότερα η χρήση του ηλεκτρονικού υπολογιστή έχουν καθοριστικό ρόλο στην αναμόρφωση των εκπαιδευτικών συστημάτων. Ο ηλεκτρονικός υπολογιστής παρέχει τη δυνατότητα συνδυασμού μάθησης και διασκέδασης, προσφέροντας ταυτόχρονα μια πληθώρα εφαρμογών όπως το ηλεκτρονικό ταχυδρομείο, τις μηχανές αναζήτησης, αλλά και τα παιχνίδια. Σημαντική λοιπόν παράμετρος της πληροφορικής στην εκπαίδευση είναι η χρησιμοποίησή του κατάλληλου εκπαιδευτικού λογισμικού ως μαθησιακό εργαλείο, ενώ ιδιαίτερη έμφαση και προσοχή είναι απαραίτητο να δοθεί στο σχεδιασμό αυτού του εκπαιδευτικού λογισμικού που θα αναπτυχθεί. Η δυνατότητα παρέμβασης και αλληλεπίδρασης του χρήστη, τόσο με τη διεπιφάνεια χρήσης, όσο και με τους διδάσκοντες, τους συνομήλικες και άλλους έμπειρους ειδικούς είναι ένα από τα χαρακτηριστικά που χρειάζονται προσοχή.

Οι εφαρμογές της πληροφορικής στην εκπαίδευση είναι:

- Πολυμεσικές εκπαιδευτικές εφαρμογές (Computer Based Training): Εκπαιδευτικές εφαρμογές, όπου γίνεται συνδυασμός κειμένου, ήχου, εικόνας (ακίνητης και κινητής) και βίντεο.
- Συστήματα εκπαίδευσης εξ αποστάσεως ή τηλεεκπαίδευσης (distance learning) όπου εκπαιδευτές και εκπαιδευόμενοι επικοινωνούν και συνεργάζονται μέσω δικτύου υπολογιστών.
- Συστήματα προσομοίωσης (simulators) που χρησιμοποιούνται στην εκπαίδευση στο χειρισμό πολύπλοκων συστημάτων.

2.4. Σύγχρονες Θεωρίες Μάθησης και ΤΠΕ

Η εισαγωγή των ΤΠΕ στην εκπαίδευση έχει επηρεαστεί από τις θεωρίες μάθησης. Ο σχεδιασμός και η υλοποίηση ενός εκπαιδευτικού λογισμικού προαπαιτούν την κατάλληλη επιλογή και εφαρμογή μιας θεωρίας μάθησης προκειμένου να απαντηθούν ερωτήματα σχετικά με το τί και πώς έχει ανάγκη ο εκπαιδευόμενος να μάθει, σε ποιο περιβάλλον μάθησης, ποιός θα είναι ο ρόλος του εκπαιδευτικού και του μαθητή κ.ά. Απαραίτητη, λοιπόν είναι η συνοπτική ανασκόπηση των βασικότερων σύγχρονων θεωριών μάθησης, στις οποίες στηρίζεται και ο σχεδιασμός των εκπαιδευτικών λογισμικών.

2.4.1. Συμπεριφορισμός (behaviorism)

Σύμφωνα με τη θεωρία του συμπεριφορισμού, σκοπός της επιστήμης είναι τόσο η περιγραφή όσο και η πρόβλεψη της ανθρώπινης συμπεριφοράς ώστε, μέσα από την εφαρμογή τεχνικών μάθησης, να επιτευχθεί η τροποποίησή της (Leahey, 1997). Η μάθηση ορίζεται ως μία αλλαγή στη συμπεριφορά του ατόμου που προκύπτει μέσω των εμπειριών και των ασκήσεων που τίθενται από τον εκπαιδευτή. Σημαντικός παράγοντας στη διαδικασία της μάθησης η ενίσχυση της συμπεριφοράς του εκπαιδευόμενου, είτε μέσω της αμοιβής (θετική ενίσχυση) είτε μέσω της διόρθωσης ή και τιμωρίας (αρνητική ενίσχυση). Κεντρικός ρόλο επομένως κατέχει ο ρόλος του εκπαιδευτή, ο οποίος είναι και ο μεταδότης της γνώσης και βασικός παράγοντας στην εκπαιδευτική διαδικασία, αφού ενισχύει την επιθυμητή συμπεριφορά. Αρχές μάθησης του Συμπεριφορισμού αποτελούν η ενεργός συμμετοχή του μαθητή, η δόμηση της διδασκαλίας ύλης σε σύντομες διδακτικές ενότητες, η παρουσίαση της ύλης σύμφωνα με τους ρυθμούς του μαθητή, η ενίσχυση των προσπαθειών του μαθητή και άμεση επαλήθευση της απάντησής του και η επιβράβευση της σωστής απάντησης. Χαρακτηριστικά μοντέλα συμπεριφοριστικής διδασκαλίας είναι η προγραμματισμένη διδασκαλία του Skinner (1954), το μοντέλο των διακλαδώσεων του N. Crowder (Nwana, 1990) και το μοντέλο του διδακτικού σχεδιασμού.

Με βάση το συμπεριφοριστικό πρότυπο αναπτύχθηκαν τα λογισμικά καθοδήγησης (tutorials) και εξάσκησης και πρακτικής (drill and practice). Στόχος τους είναι η απόκτηση μίας συγκεκριμένης γνώσης και η αξιολόγησή της μέσω του υπολογιστή, ο οποίος φαίνεται να παίρνει το ρόλο του εκπαιδευτή. Η διαδικασία της μάθησης είναι αποτέλεσμα της θετικής ενίσχυσης, που δέχεται ο μαθητής κατά τη διάρκεια της διδασκαλίας, αλλά και του εμπλουτισμού του μαθήματος με εικόνες, ήχους, βίντεο και αναπαραστάσεων. Η χρήση του λογισμικού εξαρτάται από το επίπεδο γνώσεων του εκπαιδευόμενου, ενώ η οργάνωση της διδασκαλίας είναι αυστηρά δομημένη με το ρόλο του εκπαιδευτή ως καθαρά συντονιστικό.

Η ολοκλήρωση της διαδικασίας μάθησης και άρα η μετάβαση του εκπαιδευόμενου στο επόμενο γνωστικό επίπεδο γίνεται έπειτα από την αξιολόγησή του σε γνώσεις του προηγούμενου γνωστικού επιπέδου μέσω ενός συνόλου ασκήσεων τύπου «σωστό - λάθος» ή

πολλαπλών επιλογών. Σε κάθε στάδιο της αξιολόγησης, ο εκπαιδευόμενος εισπράττει από το λογισμικό την αντίστοιχη με το αποτέλεσμα ενίσχυση, θετική ή αρνητική. Τέλος, πολλά εκπαιδευτικά λογισμικά συμπεριφοριστικού τύπου διαθέτουν πολυμεσικές εφαρμογές, οι οποίες είναι εμπλουτισμένες με εικόνα, ήχο και βίντεο ώστε να προσελκύσουν την προσοχή του εκπαιδευόμενου.

2.4.2. Οικοδομισμός ή Δομητισμός (Constructivism)

Η θεωρία του οικοδομισμού αναγνωρίζει το γεγονός πως τα παιδιά, πριν ακόμα ξεκινήσουν το σχολείο, διαθέτουν κάποιες γνώσεις και έτσι το σχολείο πρέπει να βοηθήσει να οικοδομηθούν οι νέες γνώσεις πάνω σε αυτές που ήδη κατέχουν. Έτσι, γενικά η μάθηση είναι μία υποκειμενική και εσωτερική διαδικασία οικοδόμησης νοημάτων και θεωρείται το αποτέλεσμα οργάνωσης και προσαρμογής των νέων πληροφοριών σε ήδη υπάρχουσες γνώσεις. Σε αντίθεση με τη θεωρία του συμπεριφορισμού, ο οικοδομισμός τονίζει την αξία της κατασκευής της γνώσης κι όχι την απλή αναπαραγωγή της (Jonassen, Peck, & Wilson, 1999). Κεντρικό ρόλο εδώ έχει ο εκπαιδευόμενος, ο οποίος έχει χαρακτήρα ενεργητικό και ερευνητικό, αφού επιλέγει εκείνος αυτά που θεωρεί σημαντικά, οικοδομώντας έτσι τη γνώση του. Ο ρόλος του εκπαιδευτή σταματά πλέον να είναι στο επίκεντρο και αναλαμβάνει έναν υποστηρικτικό και συμβουλευτικό ρόλο στη δραστηριότητα των εκπαιδευόμενων. Γνωστές θεωρίες οικοδομισμού είναι ο γνωστικός εποικοδομισμός του Piaget, η ανακαλυπτική μάθηση του Bruner, η θεωρία της επεξεργασίας της πληροφορίας και η θεωρία του συνδυασμού.

Η θεωρία του οικοδομισμού είχε καθοριστική σημασία στο σχεδιασμό εκπαιδευτικών λογισμικών. Στα τέλη της δεκαετίας του 1960 στο Τεχνολογικό Ινστιτούτο Μασαχουσέτης (M.I.T.), μία ομάδα ερευνητών στην Τεχνητή Νοημοσύνη με επικεφαλής τον Seymour Papert, βασιζόμενοι στο γνωστικό εποικοδομισμό του Piaget, ανέπτυξαν τη γνωστή γλώσσα προγραμματισμού LOGO, σύμφωνα με την οποία δίνοντας απλές εντολές στο μικρόκοσμο μιας χελώνας στον υπολογιστή, μπορούν να σχεδιάσουν απλά γεωμετρικά σχήματα, τα οποία σταδιακά γίνονται συνθετότερα, χρησιμοποιώντας κάθε φορά αλγόριθμους τους οποίους οι ίδιοι κατασκεύασαν. Το όνομά της οφείλεται στην Ελληνική λέξη «λόγος» (λογικό) και βασίζεται σε δύο κύρια επιχειρήματα του εμπνευστή της:

- Η εμπειρία με την LOGO οδηγεί στην απόκτηση γνωστικών δεξιοτήτων επίλυσης προβλημάτων, δεξιότητες που μπορούν να μεταφερθούν και σε άλλα μαθήματα.
- Η LOGO συνιστά έναν ιδανικό χώρο για τη μάθηση μαθηματικών εννοιών όπως η μεταβλητή, η αναδρομικότητα κ.ά.

Οι βασικές αρχές για τον σχεδιασμό μαθησιακών περιβαλλόντων με υπολογιστή, σύμφωνα με τον οικοδομισμό, είναι η παροχή εμπειριών που αφορούν με την διαδικασία οικοδόμησης της γνώσης, η παροχή εμπειριών πολλαπλών προοπτικών, η ενσωμάτωση της μάθησης σε ρεαλιστικά περιβάλλοντα που σχετίζονται με τον πραγματικό κόσμο και η εμπέδωση της μάθησης μέσω κοινωνικής εμπειρίας, η ενθάρρυνση της χρήσης πολλαπλών μορφών αναπαράστασης και τέλος η ενθάρρυνση της έκφρασης και της προσωπικής εμπλοκής.

2.4.3. Κοινωνικοπολιτισμικές θεωρίες - Θεωρία της δραστηριότητας (activity theory)

Σύμφωνα με τις κοινωνικοπολιτισμικές θεωρίες, η μάθηση θεωρείται ως μία διαδικασία κοινωνικής αλληλεπίδρασης. Οι θεωρητικοί των κοινωνικοπολιτιστικών προσεγγίσεων υποστηρίζουν ότι το άτομο αναπτύσσει ικανότητες και δεξιότητες μέσα από τη συνεργασία με άλλα άτομα και έτσι η γνώση είναι ένας φορέας κοινωνικής ταυτότητας και πολιτισμού, ο οποίος αναπτύσσεται στα πλαίσια της αλληλεπίδρασης ατόμου-κοινωνίας ή ατόμου-κοινότητας. Κεντρικό ρόλο κατέχει και εδώ ο ρόλος του μαθητή, ο οποίος προκειμένου να κατανοήσει τις πληροφορίες και να αναπτύξει γνώση, δημιουργεί απτές αναπαραστάσεις, που σχετίζονται με την εκτέλεση δράσεων, αλλά και εικονικές αναπαραστάσεις, που αντιστοιχούν σε δομές χώρου και είναι ανεξάρτητες της δράσης. Ο ρόλος του εκπαιδευτή είναι περισσότερο εμπυχωτικός, οργανωτικός και συντονιστικός κατά τη διαδικασία μάθησης. Χαρακτηριστικές θεωρίες σχετικές με τις κοινωνικοπολιτισμικές θεωρίες, είναι η κοινωνικοπολιτική θεωρία του Vygotsky, η θεωρία της δραστηριότητας καθώς και η θεωρία που υποστηρίζει το μοντέλο της εγκαθιδρυμένης μάθησης.

Η εκπαίδευση με τη χρήση των ΤΠΕ επηρεάστηκε σε πολύ μεγάλο βαθμό από τις κοινωνικοπολιτισμικές θεωρίες. Ο παγκόσμιος ιστός (World Wide Web) είναι αποτέλεσμα της κοινωνικής διάστασης της μάθησης και της σημασίας αλληλεπίδρασης μεταξύ των ατόμων σε συνδυασμό με τη ραγδαία τεχνολογική εξέλιξη. Σύμφωνα με το μοντέλο της θεωρίας της δραστηριότητας δημιουργήθηκαν οι πρώτες διαδικτυακές κοινότητες για την ανάπτυξη λογισμικών ανοιχτού κώδικα (Open Source Code), χρησιμοποιώντας τον παγκόσμιο ιστό. Επίσης, οι κοινωνικοπολιτισμικές θεωρίες επηρέασαν καταλυτικά την παιδαγωγική θεώρηση και μεθοδολογία της διδασκαλίας, με τη χρήση των ΤΠΕ, καθώς και τον τρόπο διαχείρισης του ευρύτερου σχολικού οικοσυστήματος (Κόμης, 2004). Οποιοδήποτε εκπαιδευτικό λογισμικό δίνει τη δυνατότητα ενός ομαδοσυνεργατικού σεναρίου, όπου οι εκπαιδευόμενοι ερευνούν, συζητούν, συμφωνούν και εξάγουν από κοινού ένα αποτέλεσμα, θεωρείται ότι δίνει στη μαθησιακή διαδικασία ένα κοινωνικοπολιτισμικό προσανατολισμό.

2.5. Νέος ρόλος εκπαιδευτικού και εκπαιδευόμενου

Τα σύγχρονα περιβάλλοντα εργασίας και τα εκπαιδευτικά λογισμικά παρέχουν όλο και περισσότερες δυνατότητες για επικοινωνία και συνεργασία, άμεση, σύγχρονη επικοινωνία (με ήχο, εικόνα, γραπτό κείμενο), συνεπτεξεργασία κειμένων και εικόνων, από κοινού επίλυση προβλημάτων, διαμοίραση ψηφιακών πόρων, συμμετοχή σε ψηφιακές κοινότητες, ιστολόγια και wikis κ.ά. Όλο και περισσότερο η ύπαρξη των ΤΠΕ στη διδασκαλία καταργεί ορισμένες παραδοσιακές μεθόδους διδασκαλίας, καθώς διανοίγει νέες δυνατότητες, που δεν είναι δυνατόν να αγνοηθούν. Η πλοήγηση στο διαδίκτυο έχει συμβάλει στην ευκολότερη πρόσβαση των ατόμων σε ένα τεράστιο όγκο πληροφοριών για οποιοδήποτε θέμα. Οι παραδοσιακές μέθοδοι διδασκαλίας αλλάζουν ριζικά με αποτέλεσμα τη δημιουργία ενός νέου ρόλου του εκπαιδευτικού αλλά και του εκπαιδευόμενου.

Ο εκπαιδευτής πλέον σταματάει να είναι ο μοναδικός μεταδότης πληροφοριών και έτσι μετατρέπεται σε συντονιστής, οργανωτής και ενισχυτής των προσπαθειών των εκπαιδευομένων του. Καθώς λοιπόν το εύρος των νέων τεχνολογιών είναι απεριόριστο και ο εκπαιδευτικός δεν είναι δυνατόν να γνωρίζει τα πάντα, διευκολύνει το σχηματισμό ομάδων συνεργασίας, ενθαρρύνει το μεταξύ τους διάλογο ενισχύει αλλά και υποστηρίζει μαθητικές πρωτοβουλίες. Προκειμένου να προωθηθεί η αυτονομία και η αλληλεπίδραση, σχεδιάζει το περιβάλλον μάθησης, εντάσσει την αποκτηθείσα γνώση σε ένα ρεαλιστικό πλαίσιο, χρησιμοποιεί ποικίλες μεθόδους και στρατηγικές διδασκαλίας για να καλύψει όλα τα ενδιαφέροντα και να είναι πιο

ελκυστική η μάθηση και αξιολογεί τη διαδικασία της δικής του διδασκαλίας. Επίσης κάνει χρήση διαφόρων εργαλείων για την παρακολούθηση της πορείας μάθησης.

Επειδή όμως ο ρόλος του είναι ουσιαστικός πρέπει να πληρούνται ορισμένες προϋποθέσεις ώστε να μπορέσει να ανταποκριθεί σε αυτά τα καινούρια δεδομένα. Από τη μία μεριά, ο εκπαιδευτικός πρέπει να κατέχει την κατάλληλη εκπαίδευση και κατάρτιση στα νέα αντικείμενα διδασκαλίας αλλά και στο συνδυασμό τους με τα ήδη υπάρχοντα. Επίσης θα πρέπει να ανακαλύψει κίνητρα για τη συνέχιση της προσπάθειας των εκπαιδευομένων του, τα οποία μπορεί να είναι βραβεύσεις, δημοσιεύσεις, οικονομικά κίνητρα κ.ά. Από την άλλη μεριά, ο εξοπλισμός όπου εφαρμαζόζεται το λογισμικό θα πρέπει να είναι άρτιος και να παρέχεται ανά πάσα στιγμή τεχνική υποστήριξη.

Η ύπαρξη των ΤΠΕ στην εκπαίδευση δε θα μπορούσε να είχε αφήσει ανεπηρέαστο το ρόλο του εκπαιδευόμενου, ο οποίος αποτελεί πλέον μέλος μιας ομάδας με κοινό στόχο, αναλαμβάνει την αποπεράτωση έργου και αναπτύσσει την ανάλογη δράση. Ενδιαφέρον παρουσιάζει επίσης το γεγονός ότι αλληλεπιδρά με συμμαθητές της ίδιας ομάδας, για την εκπλήρωση του στόχου, και αναζητά τη σύνδεση της γνώσης με την πραγματικότητα. Τα λάθη του πλέον σταματούν να έχουν μόνο αρνητικές σημασίες καθώς γίνονται μέρος της διαδικασίας της μάθησης. Ο εκπαιδευόμενος δεν κρατάει απαθή στάση απέναντι σε κάποιο λάθος του, αλλά προσπαθεί να αναζητήσει και να εκτελέσει λύσεις για την σωστή επίλυσή του. Έτσι, βελτιώνει τις δεξιότητές του, τις οποίες μπορεί να χρησιμοποιήσει αργότερα σε όλα τα γνωστικά πεδία. Επιβλέπωντας τη διαδικασία μάθησής του μπορεί να οδηγείται στην αυτοαξιολόγηση και έτσι να αναδομήσει την προσπάθεια που κατέβαλε σχετικά με μία εργασία. Τέλος, παρακολουθώντας καθημερινά τη διαδικασία της διδασκαλίας του αλλά και τα αποτελέσματά της, τον καθιστά ικανότερο να λάβει τις κατάλληλες αποφάσεις για τη μελλοντική του δράση.

ΜΕΡΟΣ Β: ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ

2.6. Στόχος ΕΛ (Εκπαιδευτικών Λογισμικών)

Ο όρος "Εκπαιδευτικό Λογισμικό" αναφέρεται σε μία εφαρμογή ηλεκτρονικού υπολογιστή μέσω της οποίας επιδιώκεται η αξιοποίηση των δυνατοτήτων που προσφέρουν οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας (ΤΠΕ) στον τομέα της εκπαίδευσης. Κάθε εκπαιδευτικό λογισμικό εμπεριέχει διδακτικούς στόχους, ολοκληρωμένα διδακτικά σενάρια, προσομοιώσεις με παιδαγωγική σημασία και επιφέρει συγκεκριμένα διδακτικά και μαθησιακά αποτελέσματα. Σκοπός του είναι η ποιοτική βελτίωση της διαδικασίας διδασκαλίας και μάθησης μέσα από ένα πλούσιο, ελκυστικό και προκλητικό μαθησιακό περιβάλλον που θα ευνοεί τη διερευνητική, την ενεργητική και τη δημιουργική μάθηση. Η μάθηση μετατρέπεται από μία διαδικασία όπου επικρατούσε η απομνημόνευση και η πειθαρχία σε μία διαδικασία ανακάλυψης και ενθουσιασμού.

Έτσι, με την κατάλληλη χρήση του εκπαιδευτικού λογισμικού επιτυγχάνεται η φιλικότερη και ελκυστικότερη παρουσίαση της ύλης καθώς ο ήχος, η εικόνα και το χρώμα προκαλεί το ενδιαφέρον των εκπαιδευομένων. Επίσης, ενθαρρύνεται η ενεργή συμμετοχή μέσα από δημιουργικές δραστηριότητες, πειραματισμό και διερεύνηση με σκοπό την περαιτέρω ανάπτυξη και προσέγγιση των διαφόρων θεμάτων. Η μαθησιακή διαδικασία γίνεται πιο ξεκούραστη, μέσω των οπτικοακουστικών μηνυμάτων τα οποία έχουν μεγάλη περιεκτικότητα πληροφορίας, χωρίς να είναι απαραίτητη η χρήση πολλών μακροσκελών κειμένων. Σε ορισμένες περιπτώσεις επίσης έχει αποδειχθεί πως η χρήση ενός εκπαιδευτικού λογισμικού μπορεί να μειώσει αισθητά το χρόνο και τον κόπο που καταβάλλεται για την αφομοίωση της ύλης.

Στις αρχές το εκπαιδευτικό λογισμικό εξυπηρετούσε περισσότερο τη διδασκαλία των μαθημάτων που σχετίζονταν με την τεχνολογία και την πληροφορική. Με το πέρασμα του χρόνου όμως άρχισαν να εμφανίζονται λογισμικά και για άλλες εκπαιδευτικές θεματικές ενότητες, όπως τα μαθηματικά, η φυσική η χημεία, οι ξένες γλώσσες, κ.α. Η χρήση εκπαιδευτικών λογισμικών εξυπηρετεί πολλές και διαφορετικές ανάγκες. Για παράδειγμα, ορισμένα λογισμικά ενισχύουν την προσπάθεια του μαθητή, παρέχοντάς του εργαλεία εξάσκησης, πηγές γνώσεις, κ.α. Άλλα χρησιμοποιούνται για το συντονισμό του εκπαιδευτικού έργου, τη συγκέντρωση και οργάνωση του εκπαιδευτικού υλικού, την κατάρτιση και τήρηση του προγράμματος εκπαίδευσης και γενικότερα τη διοίκηση και λειτουργία ενός εκπαιδευτικού οργανισμού.

2.7. Κατηγορίες ΕΛ

Όσον αφορά το εκπαιδευτικό λογισμικό έχουν πραγματοποιηθεί αρκετές αξιολογήσεις προσπάθειες κατηγοριοποίησής του κατά το παρελθόν. Είναι φυσικό μία και μόνο σαφής κατηγοριοποίηση να μην υφίσταται, αφού οι κατηγορίες του λογισμικού εξαρτώνται άμεσα από τα κριτήρια που θέτει κανείς. Τα όρια ανάμεσα στις διάφορες κατηγορίες ορισμένες φορές δεν είναι απολύτως σαφή και έτσι είναι ενδεχόμενο η κατηγοριοποίηση του λογισμικού να είναι αρκετά περίπλοκη. Για παράδειγμα, ένα εκπαιδευτικό λογισμικό μπορεί να χρησιμοποιηθεί με τελείως διαφορετικούς τρόπους και έτσι να υπάγεται ταυτόχρονα σε περισσότερες από μια κατηγορίες.

Τα λογισμικά μπορούν να διακριθούν ως προς τις παιδαγωγικές θεωρίες και τις θεωρίες μάθησης στις οποίες στηρίζεται το καθένα από αυτά, ως προς το ρόλο που έχει ο εκπαιδευτής, ο εκπαιδευόμενος και το λογισμικό κατά τη διάρκεια της μάθησης, ως προς το περιβάλλον διεπαφής του, ως προς τη χρήση του στη μαθησιακή διαδικασία, ως προς το βαθμό αλληλεπίδρασης, ως προς την παιδαγωγική προσέγγιση κι ως προς πολλά άλλα κριτήρια.

Στις σελίδες που ακολουθούν περιγράφεται η κατηγοριοποίηση των εκπαιδευτικών λογισμικών με βάση τη θεωρία μάθησης που έχουν σχεδιαστεί, τη χρήση τους στην εκπαιδευτική διαδικασία, το βαθμό αλληλεπίδρασης αλλά και την παιδαγωγική προσέγγιση.

2.7.1. Κατηγοριοποίηση των ΕΛ με βάση τη θεωρία μάθησης που έχουν σχεδιαστεί

- **Λογισμικά καθοδηγούμενης διδασκαλίας (tutorials), πρακτικής και εξάσκησης (drill and practice)**

Στην κατηγορία αυτή ανήκουν λογισμικά που στηρίζονται κυρίως στις συμπεριφορικές και γνωστικές θεωρίες μάθησης. Έχουν συγκεκριμένο θεματικό περιεχόμενο, το οποίο παραμένει σταθερό κατά τη διάρκεια της χρήσης τους και αντικαθιστούν το ρόλο του δασκάλου, υποστηρίζοντας έτσι τη μεταφορά της γνώσης στο μαθητή. Ο σχεδιασμός τους είναι σχετικά εύκολος και για το λόγο αυτό σ' αυτή την κατηγορία κυκλοφορούν πολλά λογισμικά. Παράλληλα, τα λογισμικά εξάσκησης και πρακτικής επιτρέπουν στο μαθητή να εξασκηθεί σε γνώσεις που ήδη έχει αποκτήσει κατά τη διάρκεια της διδασκαλίας.

- **Λογισμικά καθοδηγούμενης ανακάλυψης και διερεύνησης**

Ο σχεδιασμός των λογισμικών αυτών στηρίζεται κυρίως στις γνωστικές και επικοινωνιακές θεωρίες μάθησης. Τα λογισμικά καθοδηγούμενης ανακάλυψης και διερεύνησης μπορούν να τροποποιηθούν εύκολα από τον εκπαιδευτή και δίνουν τη δυνατότητα υποστήριξης αρκετών δραστηριοτήτων που αποβλέπουν στην καλλιέργεια υψηλού επιπέδου δεξιοτήτων των εκπαιδευομένων. Επιτρέπουν την

εμπλοκή σε δραστηριότητες ευνοώντας την αυτενέργεια, την επίλυση προβληματικών καταστάσεων, τη λήψη αποφάσεων και την καλλιέργεια της κριτικής σκέψης.

- **Λογισμικά έκφρασης, επικοινωνίας και δημιουργικότητας**

Πρόκειται για λογισμικά τα οποία στηρίζονται κυρίως σε επικοδομιστικές και κοινωνικοπολιτισμικές θεωρίες μάθησης. Η χρήση τους καλλιεργεί την έκφραση, τη γλωσσική επικοινωνία και τη δημιουργικότητα. Τέτοια λογισμικά είναι οι ψηφιακές εγκυκλοπαίδειες, τα λεξικά, οι δικτυακές εφαρμογές συνεργασίας και επικοινωνίας κ.ά.

2.7.2. Κατηγοριοποίηση των ΕΛ με βάση τη χρήση τους στην εκπαιδευτική διαδικασία

- **Λογισμικό εξάσκησης και πρακτικής (Drill & Practice)**

Τα προγράμματα αυτά στηρίζονται στη προγραμματισμένη διδασκαλία διαθέτοντας συγκεκριμένη ύλη. Εκτός από τη θεωρητική κάλυψη, παρέχουν ένα σύνολο ασκήσεων διάφορων τύπων όπως σωστό - λάθος, πολλαπλών επιλογών, ανοικτού τύπου και έχουν απλή γραμμική μορφή καλώντας τον χρήστη να απαντήσει σε μία σειρά ερωτήσεων. Έτσι, δίνουν τη δυνατότητα στους εκπαιδευόμενους να εξασκηθούν στην ύλη που ήδη έχουν διδαχθεί και να αξιολογηθούν ανάλογα με την επίδοσή τους μέσω του άμεσου ελέγχου των απαντήσεων. Επίσης, δίνουν τη δυνατότητα και στον εκπαιδευτικό να τα χρησιμοποιήσει για την άμεση αξιολόγηση των εκπαιδευομένων του. Αν και θεωρούνται κατάλληλα για επανάληψη, δεν εμφανίζονται τα τελευταία χρόνια ως αυτόνομο λογισμικό αλλά ενσωματώνονται σε άλλου τύπου λογισμικά.

- **Λογισμικό καθοδηγούμενης διδασκαλίας (Tutorial)**

Αυτού του είδους τα λογισμικά παρέχουν τη δυνατότητα παρουσίασης των κυρίως στοιχείων της θεωρίας με τη χρήση κειμένου, παραδειγμάτων, βίντεο, κινούμενης εικόνας, ερωτήσεων και προβλημάτων. Κάνουν το μάθημα ελκυστικότερο, επιδρώντας θετικά στη μάθηση. Τα προγράμματα αυτά βοηθούν στην ανάπτυξη συγκεκριμένων πρακτικών δεξιοτήτων όπως εκτέλεσης αριθμητικών πράξεων, τυφλό σύστημα, εκμάθηση ξένων γλωσσών κ.ά. Παρέχουν ερωτήσεις αυξανόμενης δυσκολίας καθώς και επεξηγήσεις μέσω οθονών βοήθειας (help screen) για τις ενέργειες που πρέπει να κάνει ο εκπαιδευόμενος. Ο κύκλος πληροφορία-ερώτηση-ανάδραση επαναλαμβάνεται καθ' όλη τη διάρκεια της εφαρμογής. Ένα «ιδανικό» πρόγραμμα παρέχει τη δυνατότητα παρουσίασης του υλικού ανάλογα με τις ανάγκες του κάθε χρήστη. Επιπλέον, ελέγχουν τα αποτελέσματα, μετρούν την απόδοσή του και τέλος αξιολογούν την επίδοσή του.

- **Διδακτικά παιχνίδια (Educational games)**

Πρόκειται για πολύ δελεαστικά περιβάλλοντα τα οποία βοηθούν στην απόκτηση και ανάπτυξη δεξιοτήτων μέσω ενός περιβάλλοντος παιχνιδιού δίνοντας τη δυνατότητα στους εκπαιδευόμενους να διδαχθούν μέσα σε ένα πλαίσιο που αυξάνει την κινητοποίηση και τον ενθουσιασμό αλλά και την προσοχή στη μαθησιακή διαδικασία. Διακρίνονται σε δράσης, περιπέτειας (adventure games), στρατηγικής, ανάπτυξης ικανοτήτων γλωσσικών, μαθηματικών κ.ά. Τα παιχνίδια στρατηγικής (σκάκι, μπριτζ, κ.ά.) είναι συνήθως παιχνίδια ρόλων και προσομοιώνουν καταστάσεις που καλείται να

αντιμετωπίσει ο χρήστης. Τα παιχνίδια ανάπτυξης γλωσσικών ικανοτήτων, εξοικείωσης με τις έννοιες, τις σχέσεις κ.ά. απευθύνονται συχνά σε παιδιά προσχολικής ή σχολικής ηλικίας και αποσκοπούν στο να ωθήσουν τα παιδιά να μάθουν παίζοντας. Το παιχνίδι χρησιμοποιείται ως κίνητρο για τον εκπαιδευόμενο έτσι ώστε αυτός να αποκτήσει συγκεκριμένες δεξιότητες και γνώσεις καθώς περιηγείται στο λογισμικό και ολοκληρώνει τις διαδικασίες του παιχνιδιού. Είναι μάλλον το πιο δύσκολο είδος λογισμικού για αξιολόγηση δεδομένου ότι είναι αμφίβολο αν οι ικανότητες που προωθούνται από το παιχνίδι είναι δυνατόν να μεταφέρουν και έννοιες. Είναι σημαντικό να πούμε πως ένα καλό παιχνίδι πρέπει να έχει μαθησιακούς στόχους συμβατούς με το πρόγραμμα σπουδών.

Χαρακτηριστικός είναι ο τρόπος διδασκαλίας τον οποίο βρήκε ένας καθηγητής ιστορίας. Επιθυμώντας να δώσει μια πιο ξεκάθαρη εικόνα ορισμένων ιστορικών γεγονότων στους μαθητές του, αγόρασε τρία παιχνίδια για τον υπολογιστή, το Age Of Empires, το Age Of Mythology και το 300 Spartans. Αφού τα εγκατέστησε στον υπολογιστή του, τα παρουσίασε την επόμενη μέρα στους μαθητές του. Έτσι τα παιδιά είχαν την ευκαιρία να παρακολουθήσουν μια αναπαράσταση της μάχης των Θερμοπυλών, της ναυμαχίας της Σαλαμίνας, διάφορες μορφές της Ελληνικής και όχι μόνο μυθολογίας καθώς και αρχαίες αυτοκρατορίες όπως η Βυζαντινή και η Ρωμαϊκή.



Εικόνα: 5 Εικόνα από το παιχνίδι Age Of Empires III

- **Λογισμικό προσομοίωσης (Simulation)**

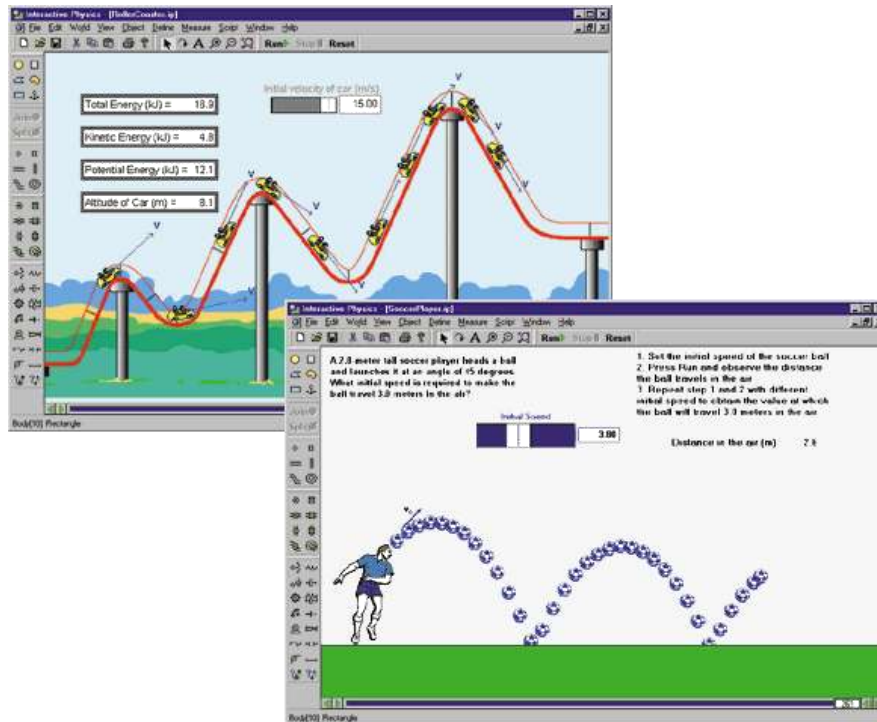
Τα λογισμικά προσομοίωσης ενσωματώνουν συνήθως ένα μαθηματικό μοντέλο, γι' αυτό και αναφέρονται συχνά ως μοντέλα προσομοίωσης. Σύμφωνα με τον Cox (1999) η χρήση προσομοιώσεων (simulations) και λογισμικών μοντελοποίησης προσφέρει τη δυνατότητα στο μαθητή να λειτουργήσει ως ερευνητής, ορίζοντας και κατασκευάζοντας ο ίδιος το μοντέλο ενός προβλήματος, μελετώντας την επίδραση διαφορετικών παραμέτρων σε αυτό αλλά και δοκιμάζοντας διαφορετικές υποθέσεις για τη λειτουργία ενός δεδομένου μοντέλου. Ο μαθητής θα πρέπει σταδιακά να αποκτήσει μια εικόνα για τον τρόπο λειτουργίας του λογισμικού, τα εργαλεία που προσφέρει και τη χρήση τους καθώς και να κατανοήσει το πρόβλημα που πρόκειται να μοντελοποιήσει. Επίσης, οι μαθητές ενεργοποιούνται αξιοποιώντας ένα τέτοιο περιβάλλον, αναλαμβάνοντας εργασίες που τους εμπλέκουν:

- στη δημιουργία και αξιολόγηση των προσωπικών τους ιδεών
- στην αντιπαράθεση της πρότερης γνώσης τους με νέα αντικρουόμενα στοιχεία
- στη σύγκριση των δικών τους μοντέλων για τον κόσμο με πραγματικά δεδομένα/καταστάσεις
- στη συνεργασία σε ομάδες με στόχο την οικοδόμηση πιο πολύπλοκων μοντέλων
- στην επανεξέταση των επιτευγμάτων τους μέσα από την παρουσίαση των δικών τους μοντέλων στην οθόνη.

Αρκετές έτοιμες προσομοιώσεις επιτρέπουν την εικονική αναπαράσταση και μοντελοποίηση ενός φαινομένου ή ενός πραγματικού συστήματος, υπό συνθήκες που προσεγγίζουν τις πραγματικές μέσα σε περιβάλλοντα εικονικών εργαστηρίων. Οι προσομοιώσεις αυτές συνήθως έχουν ένα περιορισμένο αντικείμενο και στόχο και διαμορφώνουν ένα συγκεκριμένο περιβάλλον στο οποίο οι μαθητές μπορούν είτε απλά να παρακολουθήσουν την εξέλιξη ενός φαινομένου είτε να μεταβάλλουν τις συνθήκες στο εικονικό πείραμα, βοηθώντας έτσι την κατανόηση του φαινομένου, με τα εργαλεία που τους παρέχονται. Οι προσομοιώσεις αφορούν διάφορα θέματα όπως Φυσική, Μαθηματικά, Χημεία, Αρχιτεκτονική Υπολογιστών, Μουσεία κ.ά.

Το λογισμικό Interactive Physics είναι ένα δημοφιλές «ανοιχτό» περιβάλλον διερευνητικού χαρακτήρα στη Φυσική. Πρόκειται για ένα πλήρες εργαστήριο κίνησης στον υπολογιστή που συνδυάζει μια απλή διεπαφή (user interface) με μια δυνατή μηχανή που προσομοιώνει τις βασικές αρχές της Κλασικής Μηχανικής του Νεύτωνα. Με το Interactive Physics, ο μαθητής ή ο εκπαιδευτικός μπορεί να δημιουργήσει πλήθος προσομοιώσεων που αντιστοιχούν σε πραγματικές καταστάσεις σωμάτων των οποίων μπορεί να καθορίσει τα χαρακτηριστικά (μάζα, πυκνότητα, θέση, ταχύτητα, ορμή κλπ), μπορεί να παρέμβει και να μεταβάλει τις ιδιότητές τους και τους παράγοντες που επηρεάζουν την κατάστασή τους. Επίσης, μπορεί να δημιουργήσει προσομοιώσεις σχεδιάζοντας αντικείμενα στην οθόνη, ζωντανεύοντάς τα με κίνηση, μέσω μίας ποικιλίας σχηματικών αντικειμένων (ελατήρια, σχοινιά, αποσβεστήρες, μετρητές κ.ά). Κατά τη διάρκεια της εκτέλεσης της προσομοίωσης μπορούν να υπολογιστούν μεγέθη όπως η ταχύτητα, η επιτάχυνση, η ροπή, η κινητική ενέργεια και η τριβή. Οι μετρήσεις αυτές μπορούν να εμφανιστούν με αριθμούς, με γραφικές παραστάσεις ή κινούμενα διανύσματα.

Επιπλέον, στο Διαδίκτυο υπάρχουν και αρκετά ακόμα λογισμικά προσομοιώσεων τα οποία εμπλουτίζονται συνεχώς από Πανεπιστημιακά τμήματα αλλά και διάφορες εκπαιδευτικές κοινότητες ή μέλη τους. Σε αρκετές περιπτώσεις μάλιστα, οι δημιουργοί των λογισμικών αυτών τα διαθέτουν ελεύθερα έτσι ώστε να μην απαιτείται η σύνδεση με το Διαδίκτυο κάθε φορά που εκτελούνται.



Εικόνα: 6 Interactive Physics - Προσομοίωση πειράματος Φυσικής

- **Λογισμικό επίλυσης προβλημάτων (problem solving) και αυτοαξιολόγησης**

Τα λογισμικά επίλυσης προβλημάτων και αυτοαξιολόγησης ζητούν από τους εκπαιδευμένους να επιλύσουν προβλήματα στηριζόμενοι σε γνώσεις που αποκτήθηκαν νωρίτερα. Στα περισσότερα προγράμματα προσομοίωσης και στα παιχνίδια υπάρχουν χαρακτηριστικά problem solving. Μπορούν να αποτελέσουν σπουδαία αφορμή για τη διερευνητική μάθηση δίνοντας τη δυνατότητα εφαρμογής των κλασικών στρατηγικών επίλυσης προβλημάτων. Στο πλαίσιο αυτών των λογισμικών δίνεται η δυνατότητα απόκτησης και βελτίωσης των δεξιοτήτων επίλυσης προβλημάτων, προσομοιώνοντας ένα πραγματικό φαινόμενο.

Ο χρήστης εφαρμόζει αποδεκτές αρχές ή κανόνες προκειμένου να καταλήξει σε κάποιο συμπέρασμα ή λύση, έχοντας τη δυνατότητα να παραμετροποιήσει το πρόβλημα. Έτσι, μαθαίνει να αναπτύσσει διάφορες στρατηγικές επίλυσης, αλλά και να αυτοαξιολογείται, αφού στο τέλος περιλαμβάνονται και ορισμένες εξηγήσεις ή γραφικές αναπαραστάσεις της τελικής κατάστασης από τις απαντήσεις που παράχθηκαν.

Τέλος η χρήση τους αναπτύσσει στον εκπαιδευόμενο μία εκτίμηση και κατανόηση των αλγοριθμικών μεθόδων, αποθαρρύνοντας τις διαδικασίες δοκιμής-λάθους και ενθαρρύνοντας αντίστοιχα αποφάσεις και λύσεις, οι οποίες είναι αποτέλεσμα πνευματικής διαδικασίας. Ένα εξίσου σημαντικό πλεονέκτημα είναι ότι αναδεικνύονται τυχόν παρανοήσεις και αδυναμίες του.

- **Περιβάλλοντα Εικονικής Πραγματικότητας (Virtual Reality)**

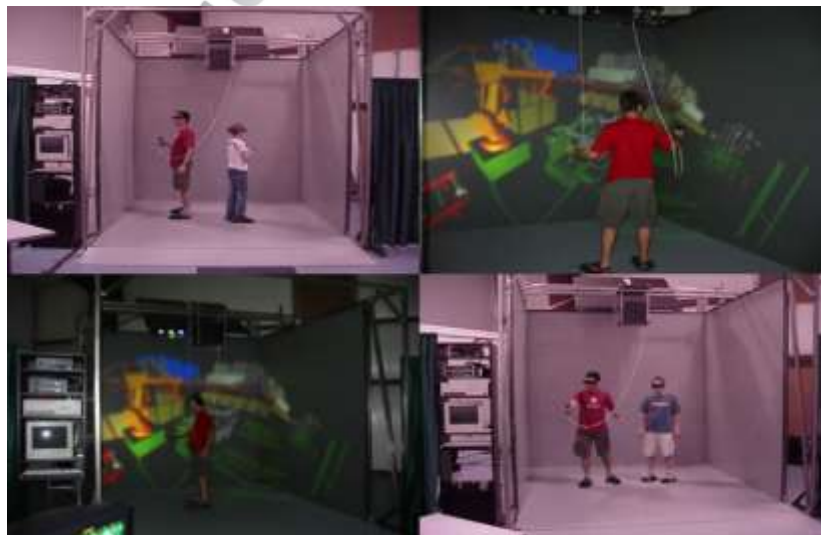
Τα Περιβάλλοντα Εικονικής Πραγματικότητας και γενικότερα οι τεχνολογίες που σχετίζονται με την εικονική πραγματικότητα χρησιμοποιούνται μέχρι στιγμής κυρίως σε ερευνητικό επίπεδο και λιγότερο στην εκπαιδευτική διαδικασία δεδομένου ότι έχουν

πολύ πρόσφατα εμφανιστεί και απαιτούν συνήθως εξειδικευμένη τεχνολογική υποδομή και εξοπλισμό. Ως εικονική πραγματικότητα ορίζεται η αλληλεπίδραση ανθρώπου - υπολογιστή σε τρισδιάστατο χώρο που χρησιμοποιείται για προσομοίωση πραγματικών ή μη καταστάσεων. Ο χρήστης καλείται να αλληλεπιδράσει με το σύστημα μέσω πράξεων, κινήσεων και εκτιμήσεων που μοιάζουν με τις καθημερινές του ενέργειες στο πραγματικό του περιβάλλον.

Η ταξινόμηση των περιβαλλόντων εικονικής πραγματικότητας πραγματοποιείται με βάση τις συσκευές εξόδου (input devices), οι οποίες του παρέχουν τις απεικονίσεις πληροφοριών. Έτσι, ανάλογα με την συσκευή οπτικής απεικόνισης μπορούμε να κατατάξουμε τα συστήματα εικονικής πραγματικότητας στις ακόλουθες κατηγορίες:

- Συστήματα εμβύθισης (immersive VR): Ο χρήστης αποκόπτεται από τον πραγματικό κόσμο με τη χρήση ενός κράνους (HMD - Head Mounted Display), στο οποίο προβάλλονται οι εικόνες του συνθετικού περιβάλλοντος.
- Επιτραπέζια Συστήματα (Desktop VR): Βασίζονται σε προσωπικούς υπολογιστές με δυνατότητα υποστήριξης εξειδικευμένων περιφερειακών πλοήγησης στον τρισδιάστατο εικονικό χώρο και χρήσης στερεοσκοπικών γυαλιών ή κράνους.
- Προβολικά συστήματα (Projection-based VR): χρήση συστήματος προβολής (μονοσκοπικής ή στερεοσκοπικής) από πολλαπλές οθόνες που κυκλώνουν το χρήστη.
- Κατοπτρικοί κόσμοι (Mirror worlds): το σύστημα παρουσιάζει στο χρήστη απεικόνιση του εαυτού του μέσα στο εικονικό περιβάλλον, με την οποία αλληλεπιδρά σε πραγματικό χρόνο.

Δυστυχώς, τα διάφορα εκπαιδευτικά λογισμικά εικονικής πραγματικότητας δεν μπορούν να θεωρηθούν ακόμη πλήρως εμπειρισταωμένα. Ωστόσο έχει αποδειχθεί ότι χαρακτηριστικά της εικονικής πραγματικότητας όπως είναι η ισχυρή αλληλεπίδραση, η άμεση ανταπόκριση του συστήματος στις ενέργειες του εκπαιδευόμενου καθώς επίσης και η ελευθερία κινήσεων σε αυτούς τους εικονικούς κόσμους μπορούν να επηρεάσουν θετικά τη μαθησιακή διαδικασία.



Εικόνα: 7 Προβολικό σύστημα εικονικής πραγματικότητας

2.7.3. Κατηγοριοποίηση των ΕΛ με βάση το βαθμό αλληλεπίδρασης

Όλα τα παραπάνω είδη εκπαιδευτικού λογισμικού που περιγράφηκαν μπορούν να διαχωριστούν σε δύο επιπλέον κατηγορίες με βάση το βαθμό αλληλεπίδρασης. Τα «κλειστά» και τα «ανοικτά» περιβάλλοντα.

➤ **Ανοικτό περιβάλλον**

Ονομάζεται το περιβάλλον μάθησης το οποίο σχεδιάζεται με βάση το γνωσιοθεωρητικό πλαίσιο του εποικοδομισμού. Στα «ανοικτά» μαθησιακά περιβάλλοντα δεν υπάρχουν προκαθορισμένα πλαίσια. Οι δραστηριότητες και οι επιλογές καθορίζονται τόσο από τις ανάγκες του μαθητή για μάθηση όσο και από τις ικανότητές του σε σχέση με τις απαιτούμενες νοητικές διεργασίες. Ως ανοικτά περιβάλλοντα μπορούν να χαρακτηρισθούν οι εφαρμογές υπερκειμένων/υπερμέσων, οι ανοικτές προσομοιώσεις και τα γνωστικά μαθησιακά εργαλεία ή εργαλεία ανάπτυξης νοητικών δεξιοτήτων (γλώσσες προγραμματισμού, μικρόκοσμοι, έμπειρα συστήματα).

➤ **Κλειστό περιβάλλον**

Πρόκειται για τα λογισμικά που υποστηρίζουν το παραδοσιακό δασκαλοκεντρικό μοντέλο διδασκαλίας. Τα περισσότερα «κλειστά» μαθησιακά περιβάλλοντα επιτρέπουν στο μαθητή να εισάγει δεδομένα. Στη διαδικασία αυτή όμως η αντίδραση του συστήματος είναι προκαθορισμένη. Ως κλειστά περιβάλλοντα μπορούν να χαρακτηρισθούν τα λογισμικά παρουσίασης και εξάσκησης, μη αλληλεπιδραστικές προσομοιώσεις καθώς και τα διδαχτικά παιχνίδια.

2.7.4. Κατηγοριοποίηση των ΕΛ με βάση την παιδαγωγική προσέγγιση

➤ **Διερευνητικό λογισμικό**

Τα λογισμικά διερευνητικού χαρακτήρα, στηριζόμενα στην παιδαγωγική αρχή της διερευνητικής μάθησης, εισάγουν και υποστηρίζουν νέους τρόπους διδασκαλίας, βοηθώντας τους εκπαιδευόμενους να αποκτήσουν νέες δεξιότητες, να οξύνουν την αντίληψή τους και να αποκτήσουν κριτικό πνεύμα και αυτενέργεια.

Βασικά χαρακτηριστικά ενός διερευνητικού λογισμικού θεωρούνται η δυνατότητα πειραματισμού και παραμετροποίησης των φαινομένων, η δυνατότητα οικοδόμησης της γνώσης μέσα από τη διαδικασία αναζήτησης της πληροφορίας και της κριτικής αποδοχής ή απόρριψης μιας άποψης, η δυνατότητα διαθεματικής προσέγγισης των εννοιών ώστε η αποκτώμενη γνώση να μην είναι αποσπασματική και τέλος η δυνατότητα συνεργασίας των μαθητών και κοινής οικοδόμησης της γνώσης μέσα από συζήτηση και αντιπαράθεση.

Σε αυτού του είδους τα λογισμικά, ο ρόλος του καθηγητή αλλάζει. Από μοναδικός κυρίαρχος της μαθησιακής διαδικασίας και μοναδικός κάτοχος της γνώσης, την οποία μετέδιδε, μετετρέπεται σε συνεργάτη του εκπαιδευόμενου στη διαδικασία οικοδόμησης της γνώσης. Ο τελικός προορισμός του εκπαιδευτικού είναι η διευκόλυνση του εκπαιδευόμενου με την κατάλληλη υπόδειξη και ενθάρρυνση.

2.8. Πλεονεκτήματα εκπαιδευτικού λογισμικού

Το βασικότερο πλεονέκτημα των εκπαιδευτικών λογισμικών είναι ότι παρέχουν μεγάλη ευκολία στο χρήστη, αφού η εκπαίδευσή του δύναται να γίνει ακόμα και στο προσωπικό του

χώρο μέσω του ηλεκτρονικού του υπολογιστή, αλλά και στο ίδιο το εκπαιδευτικό ίδρυμα, το οποίο παρέχει το λογισμικό. Με τη χρήση των εκπαιδευτικών λογισμικών υποστηρίζεται η διαδικασία της διδασκαλίας και της μάθησης με τη μορφή της τηλε-εκπαίδευσης, γνωστή και ως εξ αποστάσεως εκπαίδευση. Τα τελευταία χρόνια μάλιστα παρατηρείται μεγάλη αύξηση των εκπαιδευτικών λογισμικών που χρησιμοποιούνται για την εκπαίδευση από απόσταση και συχνά χρησιμοποιούν ως ένα μέρος του τίτλου τους την αγγλική λέξη e-learning (ηλεκτρονική μάθηση).

Επίσης, συμβάλλουν στην προώθηση της συνεργατικής αλλά και της διερευνητικής μάθησης. Οι εκπαιδευόμενοι στο πλαίσιο κοινών δραστηριοτήτων που τους ανατίθενται μαθαίνουν να συνεργάζονται, ενώ παράλληλα μέσω της διερευνητικής μάθησης ο κάθε εκπαιδευόμενος μπορεί να ακολουθεί τους δικούς του ρυθμούς μάθησης, προάγοντας ταυτόχρονα και την ανακαλυπτική μάθηση, αλλά και μειώνοντας έτσι το χρόνο μάθησης.

Αντίστοιχα, από την πλευρά του εκπαιδευτή, δίνεται η δυνατότητα συνεχούς ανανέωσης του εκπαιδευτικού υλικού, το οποίο δίνεται άμεσα στους εκπαιδευόμενους του. Τέλος, ο εκπαιδευτής μπορεί ανά πάσα στιγμή να μελετήσει τα στατιστικά των επιδόσεων του κάθε μαθητή αλλά και του συνόλου των μαθητών του, αφού έχουν ολοκληρώσει πρώτα τις ασκήσεις μέσω του εκπαιδευτικού λογισμικού.

Αναμφισβήτητα λοιπόν, το εκπαιδευτικό λογισμικό παρέχει πληθώρα προνομίων τόσο στους εκπαιδευτικούς όσο και στους εκπαιδευόμενους. Η ενσωμάτωση και ταξινόμηση των εκπαιδευτικών λογισμικών στον τομέα της Εκπαίδευσης αποτελεί προτεραιότητα κάθε σύγχρονης κοινωνίας και εξασφαλίζει την ισότιμη συμμετοχή των μαθητών στην ανταγωνιστική Κοινωνία της Πληροφορίας.

2.9. Μειονεκτήματα εκπαιδευτικού λογισμικού

Λογικό είναι το γεγονός ότι εκτός από πλεονεκτήματα, τα εκπαιδευτικά λογισμικά χαρακτηρίζονται και από ορισμένα μειονεκτήματα που συνεπάγονται από τη χρήση της πληροφορικής. Ευτυχώς, τα μειονεκτήματα αυτά είναι ελάχιστα και δεν υπερτερούν των πλεονεκτημάτων τους. Αξίζει όμως να αναφέρουμε ορισμένες από τις δυσκολίες που μπορεί κάποιος ενδεχομένως να συναντήσει κατά τη διάρκεια της χρήσης ενός εκπαιδευτικού λογισμικού. Η δυσκολία χρήσης των εκπαιδευτικών λογισμικών κλιμακώνεται ανάλογα με την εξοικείωση που διαθέτει το άτομο που τα χρησιμοποιεί. Στην περίπτωση όπου κάποιος δεν είναι εξοικειωμένος με τον ηλεκτρονικό υπολογιστή δυσκολεύεται πολύ να χρησιμοποιήσει σωστά το εκπαιδευτικό λογισμικό και έτσι δε μπορεί να φέρει εις πέρας τη διαδικασία μάθησης. Επίσης, σε αρκετές περιπτώσεις, ο τρόπος διδασκαλίας μπορεί να χαρακτηριστεί απρόσωπος και αυτό ενδεχομένως να μπορεί να οδηγήσει το χρήστη στην απομόνωση. Τέλος, υπάρχουν ορισμένα λογισμικά τα οποία, λόγω των υψηλών τεχνικών απαιτήσεων, έχουν υψηλό κόστος κατασκευής και εξοπλισμού.

2.10. Σχεδίαση Εκπαιδευτικού Λογισμικού

«Σχεδίαση λογισμικού» είναι η δραστηριότητα κατά την οποία οι ανθρώπινες απαιτήσεις μετατρέπονται σε λογισμικό ηλεκτρονικού υπολογιστή. Σκοπός της σχεδίασης λογισμικού είναι η χαρτογράφηση των διαφόρων απαιτήσεων σύμφωνα με την οποία θα παραχθεί το κατάλληλο λογισμικό, ώστε να ικανοποιεί τις απαιτήσεις αυτές. Η διαδικασία αυτή, είτε πραγματοποιείται για την εργασία είτε για την εκπαίδευση, περιλαμβάνει τα εξής συνήθως στάδια:

- ανάλυση
- σχεδίαση
- εφαρμογή
- παρατήρηση

- δοκιμή
- αναστοχασμό και επανασχεδίαση λογισμικού (αν κριθεί απαραίτητο)

Σύμφωνα με τον Olson (1988) ο σχεδιαστής ενός εκπαιδευτικού λογισμικού θα πρέπει να έχει ως πρωταρχικό κριτήριο τον ίδιο τον εκπαιδευτικό, αφού εκείνος «γνωρίζει» τις δυνατότητες των μαθητών του, «δημιουργεί» την παιδαγωγική ατμόσφαιρα στην τάξη και «καθοδηγεί» τη διαδικασία μάθησης. Το σύγχρονο καλοσχεδιασμένο εκπαιδευτικό λογισμικό πρέπει επίσης να αξιοποιεί τις προόδους της διδακτικής μεθοδολογίας, τις τεχνολογικές εξελίξεις και τις προόδους στα παρεμφερή με την εκπαίδευση επιστημονικά πεδία.

2.10.1. Βασικές Αρχές Σχεδίασης

Η σχεδίασή του εκπαιδευτικού λογισμικού διαμορφώνεται από ορισμένες βασικές αρχές, οι οποίες ενδεχομένως να μεταβάλλονται ποιοτικά ή ως προς το πλήθος ανάλογα με το είδος του λογισμικού, τον τρόπο λειτουργίας του και τον τρόπο εφαρμογής του. Μερικές σύγχρονες παιδαγωγικές αρχές για τη διδασκαλία και μάθηση και ο τρόπος που αυτές εξειδικεύονται, ώστε να καθοδηγούν τη σχεδίαση ποιοτικού εκπαιδευτικού λογισμικού είναι (Γρηγοριάδου, 1998), (Σολομωνίδου, 2000), (Κορδάκη, 2000):

- Η μελέτη και η ανάλυση των απαιτήσεων που πρέπει να καλύψει το λογισμικό
- Η διατύπωση των παιδαγωγικών στόχων που θέτει
- Η σχεδίαση των διδακτικών σεναρίων που θα συμβάλλουν στην επίτευξη των στόχων και τη δόμηση του περιεχομένου
- Ο καθορισμός του ρόλου και του είδους των αναπαραστάσεων που σχετίζονται με το θέμα
- Η επινόηση και επιλογή δραστηριοτήτων στις οποίες θα εμπλακούν ενεργά οι μαθητές ώστε να επιτευχθούν οι στόχοι που έχουν τεθεί
- Τα εργαλεία που θα διαθέτει το περιβάλλον πρέπει να συμβάλλουν στη κατανόηση του θέματος και όχι να βοηθούν απλά στο τεχνικό μέρος της επίλυσης του προβλήματος
- Το είδος της αλληλεπίδρασης και η ελευθερία εξερεύνησης του περιβάλλοντος από το μαθητή
- Η διαδικασία αξιολόγησης αποτελεσμάτων της διδακτικής αξιοποίησης του εκπαιδευτικού λογισμικού
- Η παροχή κατάλληλης βοήθειας στον εκπαιδευτικό όπου αναλύονται οι γενικοί και οι ειδικοί διδακτικοί στόχοι και δίνονται διευκρινίσεις για την εφαρμογή του λογισμικού στην τάξη.

2.10.2. Στρατηγικές Σχεδίασης

Ο Gagne (1985) διατύπωσε μια από τις πιο διάσημες στρατηγικές σχεδίασης Εκπαιδευτικού Λογισμικού. Πιο συγκεκριμένα, με βάση τη στρατηγική αυτή, το Εκπαιδευτικό Λογισμικό πρέπει να περιλαμβάνει τα εξής:

- προσέλκυση προσοχής
- πληροφόρηση του μαθητή για τους στόχους του μαθήματος και παροχή κινήτρων
- ανάκληση προηγούμενης γνώσης
- παρουσίαση του περιεχομένου
- παροχή καθοδήγησης
- εξαγωγή συμπερασμάτων και αποτελεσμάτων
- παροχή ανατροφοδότησης

- αξιολόγηση συμπερασμάτων και αποτελεσμάτων
- ανάπτυξη της μνήμης και μεταφορά μάθησης.

2.10.3. Βασικοί παράγοντες επιτυχημένης σχεδίασης

Διεπαφή - Σχεδίαση οθονών (Screen Design): Το περιβάλλον διεπαφής, δηλαδή το ενδιάμεσο στοιχείο στην αλληλεπίδραση χρήστη – Η/Υ, πρέπει να λειτουργεί με γραφικά, να λειτουργεί με εργονομία και η αισθητική να αποτελεί βασικό γνώρισμά του. Τα χειριστήρια πρέπει να οργανώνονται και να ομαδοποιούνται κατάλληλα και να κατασκευάζονται έτσι ώστε να λειτουργούν εργονομικά (Σκορδαλάκης, 1991).

Περιεχόμενο (Content): Το περιεχόμενο πρέπει να είναι ευρύ, κατάλληλα διατεταγμένο, ορθά κατακερματισμένο, καλά οργανωμένο, να διαθέτει επιστημονικότητα, αμεροληψία και σταθερό ύψος. Η παρουσίαση του περιεχομένου πρέπει να γίνεται με γλώσσα που ταιριάζει στο υπόβαθρο των χρηστών, να συνδέεται με τις εμπειρίες τους να αναδεικνύει κίνητρα χρήσης και η αναζήτηση της πληροφορίας σ' αυτό να γίνεται εύκολα και γρήγορα.

Πλοήγηση (Navigation), Αλληλεπίδραση (Interaction) και ανάδραση (Feedback): Η πλοήγηση στο εκπαιδευτικό λογισμικό πρέπει να είναι απλή και να δίνεται η δυνατότητα στο χρήστη να χρησιμοποιεί τόσο τα ενεργά στοιχεία πλοήγησης, δηλαδή τους συνδέσμους, όσο και το χάρτη πλοήγησης. Η πρόσβαση στην πληροφορία που αναζητά ο χρήστης, να γίνεται με όσο το δυνατόν λιγότερες κινήσεις.

Δομή της εφαρμογής και έλεγχος από το χρήστη (Learner control): Η δόμηση μιας εκπαιδευτικής εφαρμογής μπορεί να γίνει με πολλούς τρόπους: με απλές σελίδες, με γραμμικό σχεδιασμό σελίδων, με γραμμικό σχεδιασμό με άλματα, με δενδροειδή σχεδιασμό σελίδων, με σχεδιασμό τύπου δικτύου και με σχεδιασμό τύπου αρτηρίας, ο οποίος είναι και ο συνηθέστερος, αλλά και ο προτιμότερος.

Τεχνικά θέματα (Technical issues): Τα τεχνικά θέματα αφορούν την αισθητική παρουσίαση της εφαρμογής και τη λειτουργικότητά της και καθορίζονται από τις γραμματοσειρές, το χρώμα, την πυκνότητα κειμένου, τον ήχο και τα γραφικά.

2.11. Ποιότητα εκπαιδευτικού λογισμικού

Η «Ποιότητα» είναι ένας πολυδιάστατος όρος για τον οποίο δεν υπάρχει ακόμα κάποιος παγκόσμιος κοινός αποδεκτός ορισμός. Αν και μπορεί να αναγνωριστεί σχετικά εύκολα, αποτελεί μια έννοια που παρουσιάζει μεγάλη δυσκολία στο να οριστεί αλλά και να μετρηθεί. Ο Διεθνής Οργανισμός Προτυποποίησης (International Standards Organization, ISO) ορίζει την ποιότητα ως:

«... το σύνολο των γνωρισμάτων και των χαρακτηριστικών του προϊόντος ή της υπηρεσίας που έχουν σχέση με τη δυνατότητα ικανοποίησης και κάλυψης καθορισμένων αναγκών»

Παρ' όλο που υπάρχουν πολλοί ορισμοί της έννοιας «Ποιότητα» παρατηρείται πως το κοινό σημείο όλων των ορισμών είναι ότι η ποιότητα συνεπάγεται την ικανοποίηση των αναγκών του πελάτη (Ζαβλανό, 2003). Στην περίπτωση του Εκπαιδευτικού Λογισμικού, η ποιότητά του εξαρτάται άμεσα από το αν ικανοποιούνται όλες οι απαιτήσεις που είχαν οριστεί

αρχικά κατά το σχεδιασμό του και το λογισμικό χαρακτηρίζεται εύχρηστο και λειτουργικό από τους χρήστες. Ένα λογισμικό κρίνεται ποιοτικό όταν πληρούνται ορισμένες προϋποθέσεις κατά τη δημιουργία του. Μερικές από αυτές τις προϋποθέσεις μπορεί να είναι η δυνατότητα επαναχρησιμοποίησης, η συντηρησιμότητα, η ελεγχιμότητα, η απουσία προγραμματιστικών σφαλμάτων (bugs) αλλά και η μεταφερσιμότητα. Όλα αυτά δηλαδή που στοχεύουν στην όσο το δυνατόν ευκολότερη υλοποίηση αλλαγών στο λογισμικό.

Όπως γίνεται κατανοητό, η έννοια της ποιότητας λογισμικού είναι αρκετά αφηρημένη, με αποτέλεσμα την ανάγκη της διάσπασής της σε επιμέρους χαρακτηριστικά (παράγοντες ποιότητας λογισμικού) και τη δημιουργία μμοντέλων, τα οποία περιγράφουν τη συσχέτιση αυτών. Δύο από τα σημαντικότερα μμοντέλα ποιότητας λογισμικού που χρησιμοποιούνται μέχρι σήμερα είναι το μμοντέλο του McCall και το πρότυπο ISO 9126-1. Το μμοντέλο του McCall εστιάζει στην τμηματοποίηση της ποιότητας σε παράγοντες (actors), τους παράγοντες σε κριτήρια (criteria) και τα κριτήρια σε μετρικές (metrics). Πιο συγκεκριμένα, οι 11 παράγοντες ποιότητας που προτείνει ο McCall είναι οι:

- Ευχρηστία (usability)
- Ακεραιότητα (integrity)
- Αποδοτικότητα (efficiency)
- Ορθότητα (correctness)
- Αξιοπιστία (reliability)
- Συντηρησιμότητα (maintainability)
- Ελεγχιμότητα (testability)
- Ευελιξία (flexibility)
- Επαναχρησιμοποιησιμότητα (reusability)
- Μεταφερσιμότητα (portability)
- Διαλειτουργικότητα (interoperability)

Στο πρότυπο ISO 9126-1 η ποιότητα χωρίζεται σε έξι ποιοτικά χαρακτηριστικά, κάθε ένα από τα οποία διασπάται σε υποχαρακτηριστικά. Κάθε υποχαρακτηριστικό ανήκει σε ένα μόνο παράγοντα ποιότητας χωρίς επικαλύψεις όπως στο μοντέλο του McCall. Τα ποιοτικά χαρακτηριστικά αφορούν τις εγγενείς ιδιότητες ενός προϊόντος που μπορούν να αναλυθούν πριν την διάθεση του προϊόντος στην αγορά και ανεξάρτητα από την κατάσταση χρήσης αυτού. Τα έξι χαρακτηριστικά ποιότητας είναι τα ακόλουθα:

- **Λειτουργικότητα (Functionality):** Το σύνολο των χαρακτηριστικών γνωρισμάτων ενός λογισμικού που αφορούν στην ύπαρξη ενός συνόλου «λειτουργιών» και των ιδιοτήτων τους. Οι λειτουργίες αυτές ικανοποιούν ορισμένες ή συνεπαγόμενες ανάγκες.
- **Αξιοπιστία (Reliability):** Το σύνολο των χαρακτηριστικών γνωρισμάτων ενός λογισμικού που αφορούν στην ικανότητά του να διατηρεί το επίπεδο της απόδοσής του κάτω από συγκεκριμένες συνθήκες για μια καθορισμένη χρονική περίοδο.
- **Ευχρηστία (Usability):** Το σύνολο των χαρακτηριστικών γνωρισμάτων ενός προϊόντος λογισμικού που αφορούν στην προσπάθεια που απαιτείται για τη χρήση (ευκολία εκμάθησης) και την ικανοποίηση του χρήστη.
- **Αποδοτικότητα (Efficiency):** Το σύνολο των χαρακτηριστικών γνωρισμάτων ενός προϊόντος λογισμικού που αφορούν στη σχέση μεταξύ του επιπέδου απόδοσης του λογισμικού και του συνόλου των πόρων που χρησιμοποιήθηκαν, κάτω από συγκεκριμένες συνθήκες.

- **Συντηρησιμότητα (Maintainability):** Το σύνολο των χαρακτηριστικών γνωρισμάτων ενός προϊόντος λογισμικού που αφορούν στην προσπάθεια που απαιτείται για την εφαρμογή συγκεκριμένων τροποποιήσεων. Οι τροποποιήσεις μπορεί να περιλαμβάνουν διορθώσεις, βελτιώσεις ή προσαρμογές του προϊόντος λογισμικού σε αλλαγές του περιβάλλοντος χρήσης.
- **Μεταφερισιμότητα (Portability):** Το σύνολο των χαρακτηριστικών γνωρισμάτων ενός προϊόντος λογισμικού που αφορούν στην ικανότητά του να μεταφερθεί από ένα περιβάλλον σε ένα άλλο. Το περιβάλλον μπορεί να περιλαμβάνει περιβάλλον υλικού, λογισμικού ή οργανωτικό.

2.11.1. Ιδιότητες που χαρακτηρίζουν την ποιότητα ενός εκπαιδευτικού λογισμικού

Οι προδιαγραφές ενός ποιοτικού Εκπαιδευτικού Λογισμικού ταξινομούνται σε τέσσερις στενά συνδεδεμένες κατηγορίες. Τις προδιαγραφές του περιεχομένου, τις προδιαγραφές διδακτικής και παιδαγωγικής μεθοδολογίας, τις τεχνικές προδιαγραφές και τις προδιαγραφές αλληλεπίδρασης και περιβάλλοντος διεπαφής.

I. Προδιαγραφές Περιεχομένου

- Να εναρμονίζεται με το Πρόγραμμα Σπουδών.
- Να μην περιέχει επιστημονικές ανακρίβειες.
- Να παρουσιάζει τις δισταμένες επιστημονικές απόψεις με αντικειμενικότητα.
- Να περιέχει πληροφορία της οποίας η πυκνότητα και η ποσότητα να είναι σε αντιστοιχία με τις απαιτήσεις των μαθητών.

II. Προδιαγραφές Διδακτικής και παιδαγωγικής μεθοδολογίας

- Καθορισμός Στόχων: Θα πρέπει να αναφέρονται οι στόχοι που θα επιτευχθούν από την χρήση του, οι προαπαιτούμενες γνώσεις και δεξιότητες για την αποτελεσματική χρήση του, σε κάθε επιμέρους ενότητα θα υπάρχουν οι επιμέρους στόχοι καθώς και ο μέσος χρόνος εκπαίδευσης.
- Προσέγγιση του μαθησιακού υλικού: Το υλικό πρέπει να είναι δομημένο σε ενότητες, οι εκπαιδευτικές διαδρομές ευέλικτες έτσι ώστε ο κάθε φοιτητής να μπορεί να δημιουργεί τις δικές του εκπαιδευτικές διαδρομές, να προτείνονται εκπαιδευτικές διαδρομές με βάση τα αποτελέσματα των tests.
- Διαδικασία μάθησης: Πρέπει να δίνεται η δυνατότητα στον μαθητή να συμμετέχει ενεργά ενθαρρύνοντας την ενεργητική, τη συνεργατική και τη δημιουργική προσέγγιση της γνώσης.
- Αξιολόγηση: Το λογισμικό πρέπει να προσφέρει ποικιλία διαδικασιών αξιολόγησης και αυτό-αξιολόγησης.

III. Τεχνικές Προδιαγραφές: Αυτές οι προδιαγραφές ταξινομούνται σε τρεις βασικές κατηγορίες:

1. Λειτουργικότητα

Οι άξονες αξιολόγησης της **λειτουργικότητας** του εκπαιδευτικού λογισμικού ώστε αυτό να καλύπτει και να εξυπηρετεί τις ανάγκες αυτών που το χρησιμοποιούν είναι:

- Καταλληλότητα (Suitability), Κάνει αυτό που θέλω;
- Αξιοπιστία (Reliability), Λειτουργεί χωρίς προβλήματα;

Η αξιοπιστία χαρακτηρίζεται από:

- Ωριμότητα (Maturity) δηλαδή οι περιπτώσεις αποτυχίας λόγω σφαλμάτων του ιδίου του λογισμικού πρέπει να είναι ελάχιστες ή ανύπαρκτες.
 - Ανοχή βλαβών (Fault Tolerance) δηλαδή σε περιπτώσεις σφαλμάτων ή «παγώματος» του περιβάλλοντος διεπαφής (interface), να μπορεί να διατηρεί ένα ορισμένο βαθμό απόδοσης.
 - Δυνατότητα Ανάκαμψης (Recoverability) δηλαδή να μπορεί να επανακτά το βαθμό απόδοσής του και να διορθώνει τα δεδομένα που επηρεάστηκαν από τη βλάβη (σε συνδυασμό με το χρόνο και την προσπάθεια που απαιτούνται γι'αυτό).
- Αποδοτικότητα (Efficiency), Είναι ικανοποιητικοί οι χρόνοι απόκρισης;
- Η αποδοτικότητα αφορά :
- στο χρόνο απόκρισης (Time Behavior) όπου πρέπει οι χρόνοι απόκρισης να κυμαίνονται σε «κανονικά» πλαίσια και
 - στη συμπεριφορά πόρων (Resource Behavior) όπου πρέπει οι μέθοδοι υλοποίησης των διαφόρων λειτουργιών να απαιτούν όσο το δυνατό λιγότερους πόρους του συστήματος.
- Χρηστικότητα (Usability), Μπορώ να το χρησιμοποιήσω εύκολα;
- Ασφάλεια (Security), Προστατεύεται από μη εξουσιοδοτημένους χρήστες;
- Συμμόρφωση (Compliance), Συμμορφώνεται με τους κανονισμούς;

2. Υποστήριξη του Λογισμικού

- Αναλυτικότητα (Analyzability). Μπορώ να διαγνώσω ελαττώματα ή τμήματα που πρέπει να αντικατασταθούν χωρίς μεγάλη προσπάθεια.
- Δυνατότητα αλλαγής (Changeability). Η προσπάθεια που απαιτείται για μετατροπή ή για ενδεχόμενες αλλαγές που απαιτούνται όταν π.χ. αναβαθμίζεται το λειτουργικό σύστημα να είναι η μικρότερη δυνατή.
- Σταθερότητα (Stability). Πρέπει να ελαχιστοποιούνται οι κίνδυνοι απρόσμενων αποτελεσμάτων μετά από τροποποιήσεις που έγιναν.
- Δυνατότητα δοκιμών (testability). Πρέπει να ελέγχεται εύκολα η εγκυρότητά του.

3. Συμβατότητα του Λογισμικού

- Δυνατότητα μεταφοράς (Portability). Να μπορεί να εγκατασταθεί σε διαφορετικά εργαστηριακά περιβάλλοντα και περιορισμένη έκδοση του να μπορεί να εκτελείται σε συστήματα χωρίς π.χ. κάρτα ήχου.
- Δυνατότητα Επαναχρησιμοποίησης (Reusability). Μέρος του λογισμικού να μπορεί να χρησιμοποιηθεί και σε άλλη εφαρμογή.
- Διαλειτουργικότητα (Interoperability). Να μπορεί να επικοινωνεί σε επίπεδο ανταλλαγής δεδομένων και με άλλες εφαρμογές (επεξεργαστές κειμένου κ.λπ.). Να ενσωματώνει την δυνατότητα πρόσβασης στον παγκόσμιο ιστό.

IV. Προδιαγραφές Αλληλεπίδρασης και Περιβάλλοντος Διεπαφής

- Γλώσσα – Ορολογία. Γλώσσα απλή και κατανοητή. Δεν πρέπει να χρησιμοποιούνται τεχνικοί όροι χωρίς να επεξηγούνται.

- Δομή. Η σχεδίαση πρέπει να είναι σπονδυλωτή και τα μηνύματα μετάβασης από ενότητα σε ενότητα σαφή και κατανοητά. Καλό είναι να υπάρχει χάρτης περιεχομένων.
- Επίπεδο Αλληλεπίδρασης. Η αλληλεπίδραση πρέπει να έχει τουλάχιστον 2 επίπεδα.
- Δυνατότητα αποθήκευσης – εκτύπωσης. Θα πρέπει να υπάρχει δυνατότητα εκτύπωσης οθόνων, γραφικών, κειμένων και ασκήσεων.
- Βοήθεια. Θα πρέπει να υπάρχει σύστημα άμεσης βοήθειας, σύστημα χάρτη πλοήγησης, καθώς και λεξικό όρων και ονομασιών.
- Επιβεβαίωση ενεργειών. Σε περιπτώσεις που η ενέργεια π.χ. διαγραφή μπορεί να οδηγήσει σε δυσλειτουργία. Πρέπει να επιτρέπει αναιρέσεις ενεργειών.
- Απομνημόνευση πληροφοριών. Η απαίτηση για απομνημόνευση πληροφοριών, που θα χρειαστούν σε επόμενες ενέργειες, πρέπει να περιοριστεί στο ελάχιστο. Κατηγοριοποίηση επιλογών. Οι επιλογές κατηγοριοποιούνται με βάση το περιεχόμενό τους και δίνονται στο χρήστη με σύντομα ενεργητικά ρήματα.
- Παρουσίαση μόνο σχετικών πληροφοριών. Οι πληροφορίες που παρουσιάζονται πρέπει να είναι οι πλέον απαραίτητες και σχετικές με την τρέχουσα δραστηριότητα.
- Παρουσίαση κειμένου, συμβόλων κτλ. Εναλλαγή Γραμματοσειρών, ομοιόμορφη χρήση συμβόλων εικονιδίων κ.λ.π.
- Χρήση πολυμέσων. Η κατάχρηση εντυπωσιακών δυνατοτήτων της τεχνολογίας πολυμέσων εμπεριέχει τον κίνδυνο να μετατραπεί ο φοιτητής σε θεατή και παθητικό δέκτη.
- Παρουσίαση μηνυμάτων Τα μηνύματα θα πρέπει να είναι κατανοητά και να καθοδηγούν τον χρήστη.
- Έλεγχος της πληροφορίας
 - Στην ποσότητα της πληροφορίας. Να έχει την δυνατότητα ελέγχου της ροής της.
 - Στον έλεγχο της κίνησης. Η κίνηση μπρος-πίσω ή η πρόσβαση σε διαφορετικά μέρη της εφαρμογής πρέπει να είναι εύκολη.
 - Στην επανεκκίνηση, πρέπει να υπάρχει δυνατότητα εξόδου από το πρόγραμμα από οποιοδήποτε σημείο καθώς και δυνατότητα επιστροφής στο σημείο όπου έγινε η διακοπή.
 - Στα επίπεδα δυσκολίας. Εναλλακτικές διαδρομές ανάλογα με το γνωστικό του επίπεδο.
- Εναλλακτικές συσκευές. Πρέπει να παρέχονται εναλλακτικοί τρόποι εισαγωγής δεδομένων ποντίκι ή/και πληκτρολόγιο.
- Απαιτούμενες ενέργειες. Η διαδικασία εισαγωγής δεδομένων πρέπει να μην απαιτεί πολύπλοκους χειρισμούς ή χρονοβόρες πληκτρολογήσεις.
- Τεχνικές οδηγίες. Σαφείς οδηγίες για τις απαιτήσεις σε υλικό και λογισμικό καθώς και για την διαδικασία εγκατάστασης και λειτουργίας του λογισμικού.
- Τεκμηρίωση με παραπομπές. Σαφής αναφορά για τις προαπαιτούμενες γνώσεις. Επίσης, πληροφορίες για τον τρόπο ένταξης της εφαρμογής στο ευρύτερο εκπαιδευτικό περιβάλλον.

2.12. Μέθοδοι Αξιολόγησης εκπαιδευτικού λογισμικού

Η ευρέως διαδεδομένη χρήση του Εκπαιδευτικού Λογισμικού στον τομέα της εκπαίδευσης, οι δυνατότητες του καθώς και οι προβληματισμοί της εκπαιδευτικής κοινότητας για την ποιότητα του Εκπαιδευτικού Λογισμικού οδήγησαν στην ανάγκη εύρεσης μεθόδων αξιολόγησής του. Λόγω της πολυπλοκότητας του Εκπαιδευτικού Λογισμικού όμως ο όρος

«αξιολόγηση Εκπαιδευτικού Λογισμικού» αποτελεί μία ευρεία έννοια ακόμα στις μέρες μας με αποτέλεσμα να του δίνονται διάφοροι ορισμοί. Ένας πιθανός ορισμός θα μπορούσε να είναι ο εξής: «Αξιολόγηση Εκπαιδευτικού Λογισμικού είναι η συστηματική συλλογή, ανάλυση και ερμηνεία πληροφοριών για οποιαδήποτε πλευρά του με στόχο τη διαπίστωση της αποτελεσματικότητας και της αποδοτικότητάς του ή την εκτίμηση οποιωνδήποτε άλλων παραμέτρων που σχετίζονται με την εφαρμογή του». Έτσι, η αξιολόγηση Εκπαιδευτικού Λογισμικού είναι μία πολύπλοκη διαδικασία αφού οφείλει να καλύψει ένα ευρύ φάσμα θεμάτων. Πρέπει να βασίζεται σε ένα σύνολο προδιαγραφών καθορισμένων από την αρχή και αναμενόμενων αποτελεσμάτων που προκύπτουν από τη χρήση του προϊόντος που αξιολογείται. (Μικρόπουλος, 2000).

Απαραίτητη προϋπόθεση για την έναρξη μιας αξιολόγησης είναι το να έχει προσδιοριστεί ο λόγος για τον οποίο γίνεται η αξιολόγηση, ποιούς αφορά και το αν έχουν προσδιοριστεί οι ανάγκες και οι απαιτήσεις που πρέπει να πληρούνται από το σύστημα (Wright, 1990). Επίσης, πρέπει να είναι γνωστό εκ των προτέρων το αν θα αξιολογηθούν τα τεχνικά ή εκπαιδευτικά χαρακτηριστικά του λογισμικού και ο τρόπος αλληλεπίδρασης του λογισμικού με τον εκπαιδευόμενο και την αποκτηθείσα γνώση από τη χρήση του λογισμικού. Η αξιολόγηση στηρίζεται σε μια σειρά δεδομένων που έχουν καταγραφεί μέσω μιας ποσοτικής ή ποιοτικής έρευνας ή μέσω εκτίμησης της εμπειρίας και των εντυπώσεων των εκπαιδευτικών που χρησιμοποίησαν το λογισμικό. Τα δεδομένα αυτά λαμβάνονται μέσω ερωτηματολογίων, συνεντεύξεων, παρατηρήσεων, αυτοματοποιημένων μετρήσεων, ψυχομετρικών τεστ, λιστών αξιολόγησης κ.ά. Τα στάδια αξιολόγησης ενός Εκπαιδευτικού Λογισμικού είναι ο καθορισμός των ερευνητικών ερωτημάτων, ο σχεδιασμός και η οργάνωση, η συλλογή δεδομένων, η ανάλυση των δεδομένων, η ερμηνεία των αποτελεσμάτων, η αξιολόγηση των αποτελεσμάτων και τέλος η βελτίωση του λογισμικού. Στις σελίδες που ακολουθούν γίνεται μία περιεκτική παρουσίαση των κυριότερων μεθόδων που χρησιμοποιούνται για την αξιολόγηση των Εκπαιδευτικών Λογισμικών.

2.12.1. Αξιολόγηση με πειραματικές μεθόδους - εμπειρικές μέθοδοι αξιολόγησης

Στην περίπτωση των εμπειρικών μεθόδων αξιολόγησης χρησιμοποιούνται ερωτηματολόγια, τα οποία απαντούν άτομα που συμμετέχουν σε πειραματικές ομάδες και χρησιμοποιούν το εκπαιδευτικό λογισμικό. Πρόκειται για τις πρώτες μεθόδους που χρησιμοποιήθηκαν στις πρώτες έρευνες τη δεκαετία του 1980 προκειμένου να διερευνηθεί το εκπαιδευτικό αποτέλεσμα της χρήσης του υπολογιστή (Σολομωνίδου, 2006). Κατά την περίοδο αυτή πραγματοποιήθηκαν αρκετά πειράματα, τα αποτελέσματα των οποίων είναι αμφιλεγόμενα. Αρκετοί ερευνητές, μεταξύ των οποίων είναι και η κ.Σολομωνίδου (2006), υποστηρίζουν ότι οι πειραματικές μέθοδοι είναι από τους καλύτερους τρόπους αξιολόγησης εκπαιδευτικών λογισμικών. Ωστόσο, άλλοι ερευνητές, όπως οι Hemard & Cushion (2006), διαφωνούν, υποστηρίζοντας ότι η φύση του προβλήματος είναι πολύ πιο σύνθετη για να αντιμετωπιστεί με μία τέτοια προσέγγιση και πως δίνεται μεγαλύτερη έμφαση σε ποσοτικές αναλύσεις σε βάρος της διαδικασίας.

2.12.2. Αξιολόγηση με λίστες αξιολόγησης (check lists)

Οι λίστες αξιολόγησης αποτελούν και αυτές από τους πρώτους μεθόδους αξιολόγησης των εκπαιδευτικών λογισμικών. Πρόκειται για προκαθορισμένες φόρμες οι οποίες περιλαμβάνουν ένα σύνολο ερωτήσεων εστιασμένο πάνω στις απαιτήσεις της αξιολόγησης. Υπάρχουν πολλά και διάφορα είδη από διαθέσιμες φόρμες για την υποβοήθεια της αξιολόγησης λογισμικού. Οι συντάκτες του μπορεί να είναι φορείς,

οργανισμοί, εκπαιδευτικοί, ειδικοί στην πληροφορική και άλλοι. Αυτό είναι ένα στοιχείο που τις διαφοροποιεί μεταξύ τους, μολονότι πολύ συχνά το περιεχόμενό τους επικαλύπτεται (Παναγιωτακόπουλος κ.α., 2003). Τα κριτήρια για την αξιολόγηση του Εκπαιδευτικού Λογισμικού αναφέρονται στην παιδαγωγική και τεχνολογική αξιολόγηση του λογισμικού και διερευνούν την ύπαρξη και το βαθμό στον οποίο υπάρχουν ορισμένα χαρακτηριστικά που κατατάσσονται σε τέσσερις κατηγορίες. Αυτές είναι τα εκπαιδευτικά χαρακτηριστικά, το αναλυτικό πρόγραμμα, η αισθητική και η τεχνική.

Τα εκπαιδευτικά χαρακτηριστικά σχετίζονται άμεσα με την παιδαγωγική προσέγγιση. Πιο συγκεκριμένα ένα Εκπαιδευτικό Λογισμικό οφείλει να:

- παρέχει κίνητρα, δηλαδή οφείλει:
 - να προσελκύει και να διατηρεί την προσοχή
 - να προσδιορίζει την ενίσχυση
 - να προκαλεί, να προωθεί την άμιλλα
 - να ελέγχει την εμπιστοσύνη
 - να κεντρίζει την περιέργεια γενικά καθώς και την γνωστική περιέργεια
 - να κινητοποιεί ειδικά τη φαντασία, να θέτει στόχους
 - να υπολογίζει το βαθμό επίτευξης των στόχων, να είναι χρήσιμο, να ικανοποιεί
 - να έχει κατάλληλη διάρκεια και ποικιλία
- να προωθεί την αλληλεπίδραση και την ανάδραση
- να είναι προσανατολισμένο στο στόχο
- να προσδιορίζει το ρόλο του εκπαιδευτικού
- να χειρίζεται με κατάλληλο τρόπο τα λάθη των μαθητών

Η σχέση του λογισμικού με το αναλυτικό πρόγραμμα ανήκει και αυτή στην παιδαγωγική προσέγγιση. Σύμφωνα με αυτήν το Εκπαιδευτικό Λογισμικό οφείλει:

- να περιέχει αλληλουχίες συμβατές με το αναλυτικό πρόγραμμα
- να επιτρέπει τον πειραματισμό, δηλαδή την εφαρμογή σε πραγματικές καταστάσεις
- να χρησιμοποιεί τις θέσεις της γνωστικής υπερφόρτωσης
- να είναι συμβατό με τα νοητικά σχήματα των μαθητών
- να μπορούν να δουν οι χρήστες του πώς πρέπει να πλοηγούνται και σε ποιο σημείο του λογισμικού βρίσκονται κάθε στιγμή
- να αισθάνονται ότι βρίσκονται σε οικείο περιβάλλον και όχι ότι χάνονται
- να παρέχονται εργαλεία ελέγχου της διαδικασίας από το μαθητή ή ο έλεγχος να γίνεται εξολοκλήρου από το λογισμικό. Οι μαθητές να είναι ελεύθεροι να «ανακαλύψουν» το λογισμικό και η πλοήγηση είναι γραμμική ή διακλαδωμένη
- Η σχέση του λογισμικού με την αισθητική σχετίζεται με την τεχνολογική προσέγγιση αξιολόγησης λογισμικού. Σύμφωνα με αυτήν ένα Εκπαιδευτικό Λογισμικό οφείλει: να έχει ελκυστικά χρώματα που βοηθούν στην κατανόηση
- έχει κατάλληλο τρόπο παρουσίασης των κειμένων
- να γίνεται χρήση κατάλληλων υπερκειμένων, γραφικών, ήχου, κινούμενων σχεδίων και βίντεο
- να παρουσιάζεται η οθόνη του με τρόπο ελκυστικό και κατανοητό
- να παρουσιάζει τις εκπαιδευτικές οδηγίες με τρόπο κατανοητό
- να διαθέτει επαρκείς καταλόγους επιλογών και εικονίδια
- να είναι κατάλληλα σχεδιασμένη η διασύνδεση και να παρέχει πληροφορίες για τον προσανατολισμό, αλλά και κατάλληλα εργαλεία πλοήγησης

Η τέταρτη και τελευταία κατηγορία είναι η σχέση του Εκπαιδευτικού Λογισμικού με την τεχνική. Έτσι, ένα λογισμικό οφείλει:

- να εξασφαλίζει την ατομική του χρήση,
- να καταγράφει δεδομένα από τις απαντήσεις και την επίδοση του εκπαιδευόμενου
- να μπορεί να χρησιμοποιηθεί με ασφάλεια σε περίπτωση περιορισμένης βλάβης

Η αξιολόγηση με λίστες αξιολόγησης είναι γενικά μία κερδοφόρα μέθοδος αξιολόγησης αν λάβουμε υπόψη το χρόνο και το κόστος που χρειάζονται προκειμένου να διεξαχθεί μία αξιολόγηση ενός Εκπαιδευτικού Λογισμικού. Στα πλεονεκτήματα αυτής της μεθόδου αξιολόγησης έγκειται το γεγονός ότι μπορεί να στηριχθεί σε προϋπάρχουσες με παρόμοιο θεματικό αντικείμενο και η ύπαρξη σταθερών στοιχείων από λίστα σε λίστα, καθώς διευκολύνει τη σύγκριση αξιολογήσεων που έχουν γίνει από διαφορετικούς αξιολογητές (Παναγιωτακόπουλος κ.α., 2003). Γενικά, αποτελούν τον πιο διαδεδομένο τρόπο αξιολόγησης, αν και έχουν δεχθεί αυστηρή κριτική από τη διεθνή βιβλιογραφία λογισμικού.

Αρκετοί ερευνητές υποστηρίζουν πως η αξιολόγηση υποβαθμίζεται με τις λίστες αξιολόγησης, αφού τα κριτήρια για την ανάπτυξη των καταλόγων ομαδοποιούνται και ποσοτικοποιούνται χωρίς διάκριση ανάμεσά τους. Έτσι, για παράδειγμα, μπορεί να ένα Εκπαιδευτικό Λογισμικό να βαθμολογηθεί με άριστα ως προς την ποιότητα του γραφιστικού περιβάλλοντος και να διαθέτει χαμηλή βαθμολογία ως προς το αποτέλεσμα του λογισμικού στη μάθηση. Ενώ, λοιπόν, το συγκεκριμένο λογισμικό μπορεί να χαρακτηρίζεται από σχετικά υψηλή βαθμολογία, να κρίνεται απαράδεκτο από πλευρά παιδαγωγικής προσέγγισης. Ως αποτέλεσμα των παραπάνω προκύπτει το γεγονός ότι η προσέγγιση αυτή ενδέχεται να δημιουργήσει μία κατάσταση πλάνης, η οποία αποπροσανατολίζει την εκπαιδευτική κοινότητα. Βασικά επίσης μειονεκτήματα αποτελούν και η έλλειψη αντικειμενικότητας στον καθορισμό «ειδικού βάρους» της κάθε ερώτησης, η αδυναμία να ανταποκριθούν σε σύγκριση Εκπαιδευτικού Λογισμικού το οποίο διέπεται από διαφορετικούς τρόπους διδακτικής προσέγγισης, η έμφαση στις ομοιότητες παρά στις διαφορές κατά τη σύγκριση δύο διαφορετικών εφαρμογών Εκπαιδευτικού Λογισμικού, η επικέντρωσή τους σε τεχνικά παρά σε άλλους είδους θέματα που αφορούν το υπό αξιολόγηση λογισμικό (Παναγιωτακόπουλος κ.α., 2003). Σηματικό, επίσης, μειονέκτημα αυτής της μεθόδου είναι η έκταση της λίστας αξιολόγησης, αφού δημιουργεί κόπωση σε αυτούς που τα συμπληρώνουν, με αποτέλεσμα να υπάρχει ανακρίβεια στις απαντήσεις τους όπως και ασυμπλήρωτες ερωτήσεις.

2.12.3. Αξιολόγηση από ειδικούς – μέθοδοι επιθεώρησης

Βασική παράμετρος σε αυτή τη μέθοδο αξιολόγησης αποτελούν οι αξιολογητές, οι οποίοι γνωρίζουν τους κανόνες και τη μεθοδολογία σχεδιασμού των Εκπαιδευτικών Λογισμικών. Τα άτομα αυτά αξιολογούν το πρόγραμμα χρησιμοποιώντας μεθόδους, οι οποίες προσομοιώνουν την αναμενόμενη τυπική χρήση της εφαρμογής. Οι μέθοδοι αυτοί μπορούν να εφαρμοστούν στα αρχικά στάδια του σχεδιασμού και έτσι έχουν χαμηλότερο κόστος από την παρατήρηση των χρηστών. Οι δύο κυριότεροι μέθοδοι επιθεώρησης είναι:

- η ευρετική αξιολόγηση (Nielsen, 1994) που πραγματοποιείται από πραγματικούς αξιολογητές, οι οποίοι ελέγχουν τη διεπιφάνεια χρήσης έχοντας ως άξονα ένα σύνολο ευρετικών κανόνων. Πρόκειται για μία υποκειμενική μέθοδο, η οποία στηρίζεται στην εφαρμογή ενός μικρού αριθμού γνωστών κανόνων σχεδιασμού διαδραστικών συστημάτων και γίνεται από έμπειρους αξιολογητές ευχρηστίας, οι οποίοι δεν έχουν εμπλακεί στην ανάπτυξη του συστήματος (Τσέλιος, 2007).
- το γνωσιακό προβάδισμα (Lewis et al., 1990) που ακολουθεί μια λεπτομερή διαδικασία προσομοίωσης της εκτέλεσης μιας ορισμένης εργασίας, καθορίζοντας αν οι προσομοιωμένοι στόχοι του χρήστη και η ανάδραση του συστήματος

μπορούν να οδηγήσουν στην επόμενη σωστή κίνηση. Η μέθοδος αυτή χρησιμοποιείται κυρίως για την ανάλυση διαδραστικών συστημάτων στα οποία ο χρήστης μαθαίνει τη χρήση του συστήματος με διερευνητικό τρόπο. Σκοπός της μεθόδου αυτής είναι η ανακάλυψη των σχεδιαστικών ατελειών του συστήματος.

Όπως όλες οι προαναφερθείσες μέθοδοι αξιολόγησης, έτσι και η μέθοδος αξιολόγησης από ειδικούς, αμφισβητείται από ορισμένους ειδικούς, ενώ υποστηρίζεται από άλλους. Αρκετοί θεωρούν ότι η μέθοδος αυτή μπορεί να δώσει σχεδιαστικές λύσεις που αφορούν τη διασύνδεση και την αλληλεπίδραση, ενώ άλλοι υποστηρίζουν πως μόνο σε πραγματικό περιβάλλον χρήσης του μπορεί ένα λογισμικό να αξιολογηθεί ουσιαστικά.

2.12.4. Μοντελοποιημένες τεχνικές και αναλυτικές μέθοδοι αξιολόγησης

Η μέθοδος αυτή αφορά την ανάπτυξη μοντέλων που προβλέπουν ή καταγράφουν τη συμπεριφορά των χρηστών ενός λογισμικού. Σύμφωνα με τις τεχνικές αυτές, περιγράφονται οι στόχοι και οι ενέργειες που εκτελεί ένας τυπικός χρήστης όταν αλληλεπιδρά με το λογισμικό. Διακρίνονται σε προβλεπτικά και καταγραφικά μοντέλα. Τα προβλεπτικά μοντέλα παράγονται από τον σχεδιαστή, ενώ τα καταγραφικά μοντέλα παράγονται από το χρήστη. Η μέθοδος ανάλυσης εργασιών περιλαμβάνει τη μελέτη, τη συλλογή και ανάλυση δεδομένων, τα οποία αφορούν τον τρόπο με τον οποίο οι χρήστες εκτελούν συγκεκριμένες διεργασίες ώστε να προκύψει βαθύτερη κατανόηση των αλληλεπιδράσεων που λαμβάνουν χώρα (Tselios et al, 2002). Πιο συγκεκριμένα μια διαδικασία ανάλυσης στόχου αποτελείται από πέντε γενικά στάδια:

- προκαταρκτική ανάλυση, όπου προσδιορίζεται το φυσικό και κοινωνικό περιβάλλον, όπου αναλύεται το σύστημα
- συλλογή των στοιχείων
- μοντελοποίηση της εργασίας
- ανάλυση μοντέλου διεργασιών
- διατύπωση των αποτελεσμάτων

Πλεονεκτήματα της μεθόδου αυτής είναι το γεγονός ότι μπορεί να χρησιμοποιηθεί ως βάση για τη σχεδίαση της διασύνδεσης, αφού βασίζεται στις εργασίες του χρήστη και το γεγονός ότι μπορεί να γίνει αντιληπτή και από ανθρώπους με διαφορετικό γνωστικό υπόβαθρο από των αναλυτών της διασύνδεσης. Επιπλέον, το μοντέλο μπορεί να αξιολογηθεί ως προς την πληρότητα υποστήριξης των απαιτούμενων εργασιών και το προϊόν ανάλυσης μπορεί να αξιολογηθεί και να αναλυθεί από πλευρά ευχρηστίας της διασύνδεσης.

Από την άλλη μεριά, μειονεκτήματα αυτής της μεθόδου αποτελούν το γεγονός ότι οι πραγματικές συμπεριφορές και τα συναισθήματα των ερωτηθέντων μπορεί να διαφέρουν από τις απαντήσεις τους και ότι οι μέθοδοι αυτοί δε λαμβάνουν υπόψη και άλλους καθοριστικούς παράγοντες, όπως τα επίπεδα δυσκολίας των διαφόρων εργασιών, η έκταση του μαθήματος, οι προσδοκίες των χρηστών, το πολιτιστικό υπόβαθρο κ.ά.

2.12.5. Αξιολόγηση με βάση τις θεωρίες μάθησης

Τα λογισμικά που έχουν αναπτυχθεί με βάση τη θεωρία του συμπεριφορισμού δίνουν περιορισμένο έλεγχο στο χρήστη, ο οποίος αντιμετωπίζεται ως παθητικός αποδέκτης της γνώσης. Το περιβάλλον τους δεν είναι αλληλεπιδραστικό και λεκτική επιβράβευση αποτελεί και την αλληλεπίδραση μεταξύ του ανθρώπου και του υπολογιστή. Αντίθετα, στα λογισμικά που στηρίζονται στη θεωρία του οικοδομισμού, ο μαθητής έχει τον έλεγχο της διαδικασίας

της μάθησης, το περιβάλλον είναι αλληλεπιδραστικό και πολυμεσικό, ενώ η επιβράβευση του μαθητή είναι η ίδια του η εργασία. Αρκετοί υποστηρίζουν πως ένα περιβάλλον που στηρίζεται στη θεωρία του οικοδομισμού οδηγεί σε ουσιαστικότερη μάθηση, αφού πρόκειται για ένα περιβάλλον όπου τα λάθη των χρηστών είναι αναμενόμενα και χρήσιμα. Οι χρήστες μπορούν να τα επανορθώσουν και έτσι να μάθουν μέσα από αυτά.

Στηριζόμενοι λοιπόν στο κατά πόσο ένα Εκπαιδευτικό Λογισμικό χαρακτηρίζεται από χαρακτηριστικά του οικοδομισμού, μπορούμε να το αξιολογήσουμε, αφού θεωρείται καλό και από παιδαγωγική άποψη. Προκειμένου ένα λογισμικό να χαρακτηρίζεται όμως από χαρακτηριστικά του οικοδομισμού πρέπει να έχει προηγηθεί έρευνα για το πως οι εκπαιδευόμενοι οικοδομούν τη γνώση σχετικά με το μαθησιακό θέμα όπου εμπλέκεται το λογισμικό. Κατά την αξιολόγηση λαμβάνονται επίσης υπόψη η ευχρηστία, η μάθηση αλλά και θέματα που συνδέονται και με τις δύο έννοιες ταυτόχρονα.

Γενικά, η αξιοποίηση οποιασδήποτε θεωρίας μάθησης μπορεί να συνεισφέρει στην ανάπτυξη του λογισμικού έχοντας ως γνώμονα τα κύρια χαρακτηριστικά τους. Έτσι, ένα Εκπαιδευτικό Λογισμικό οφείλει να στηρίζεται σε κάποια προτεινόμενη θεωρία μάθησης ώστε να μπορεί να διαπιστωθεί και να αξιολογηθεί μέσω αυτής. Καμία θεωρία μάθησης δεν είναι απορριπτέα στο σχεδιασμό ενός Εκπαιδευτικού Λογισμικού. Όλες μπορούν να εφαρμοστούν ανάλογα με το υπόβαθρο των χρηστών και το εκπαιδευτικό θέμα που αναλύεται, αρκεί ο κατασκευαστής του Εκπαιδευτικού Λογισμικού να γνωρίζει τα ισχυρά και ασθενή σημεία κάθε θεωρίας, ώστε να βελτιστοποιήσει τη χρήση τους με την κατάλληλη εκπαιδευτική στρατηγική. Η εφαρμογή των Θεωριών Συμπεριφοράς απαιτεί μικρό βαθμό επεξεργασίας, η εφαρμογή των Γνωστικών Θεωριών Μάθησης μεγαλύτερο και τέλος η εφαρμογή των Θεωριών Οικοδόμησης της Γνώσης πολύ μεγάλο βαθμό επεξεργασίας από τον άνθρωπο. Έτσι, οι Θεωρίες Συμπεριφοράς είναι περισσότερο εφαρμόσιμες σε απλές προσεγγίσεις θεμάτων που αναφέρονται σε μικρές ηλικίες, οι Γνωστικές Θεωρίες Μάθησης εφαρμόζονται καλύτερα σε λογισμικά προβλημάτων στα οποία δίνεται μεγαλύτερη έμφαση και οι Θεωρίες Οικοδόμησης της Γνώσης ταιριάζουν περισσότερο στις εφαρμογές λύσης προβλημάτων ή στις εφαρμογές που εμπεριέχουν στοιχεία κοινωνικής διαπραγμάτευσης (Παναγιωτακόπουλος κ.α., 2003).

2.12.6. Μέθοδοι αξιολόγησης με τη συμμετοχή χρηστών

Η αξιολόγηση με τη συμμετοχή χρηστών θεωρείται από πολλούς ερευνητές η καταλληλότερη μέθοδος αξιολόγησης του Εκπαιδευτικού Λογισμικού, αφού οι εκπαιδευόμενοι είναι οι τελικοί χρήστες της εφαρμογής και άρα οι βασικότεροι κριτές της. Σύμφωνα με αυτό τον τρόπο αξιολόγησης, χρησιμοποιούνται συνεντεύξεις, ερωτηματολόγια, παρατήρηση, καταγραφή αντιδράσεων, ηχογράφηση απόψεων και συζητήσεων και άλλων διάφορων μεθόδων ύστερα και κατά τη χρήση του Εκπαιδευτικού Λογισμικού, έχοντας ως επίκεντρο τους εκπαιδευόμενους αλλά και τους εκπαιδευτικούς.

Σύμφωνα με τους McDougall & Squires, προτείνεται μία νέα μέθοδος αξιολόγησης Εκπαιδευτικού Λογισμικού, η οποία λαμβάνει υπόψη την αλληλεπίδραση των μαθητών, των εκπαιδευτών και των σχεδιαστών του λογισμικού. Πιο συγκεκριμένα αναλύονται οι τρεις παρακάτω παράμετροι-διαστάσεις ως εξής:

Η προοπτική αλληλεπίδρασης εκπαιδευτή και εκπαιδευόμενου: Στοχεύει στο να αποκτήσουν οι εκπαιδευόμενοι αυτονομία και δεξιότητες μέσα από τις εμπειρίες τους, ενώ παράλληλα να γνωρίσουν τις προσωπικές τους ανάγκες και να αυξήσουν τη συνεργατική μάθηση. Όλα τα παραπάνω έχουν ως αποτέλεσμα την αλλαγή της μαθησιακής κουλτούρας, αφού ο εκπαιδευτής χάνει τον ηγετικό του ρόλο και λειτουργεί περισσότερο ως οργανωτής της γνώσης και υποστηρικτής της προσπάθειας των εκπαιδευόμενων.

Η προοπτική αλληλεπίδρασης σχεδιαστή και εκπαιδευόμενου: Κατά την αξιολόγηση ενός λογισμικού πρέπει να περιλαμβάνεται μια εκτίμηση σχετικά με το πάνω σε ποιά ή ποιές θεωρίες μάθησης έχει στηριχθεί ο σχεδιασμός του. Η αξιολόγησή του πρέπει να αφορά το αν μπορεί να διαπιστωθεί η θεωρία μάθησης στην οποία στηρίζεται η κατασκευή του και το αν η θεωρία που επιλέχθηκε είναι η κατάλληλη. Επίσης η αξιολόγηση αφορά το αν ο σχεδιασμός του λογισμικού είναι σύμφωνος με τη θεωρία μάθησης.

Η προοπτική αλληλεπίδρασης σχεδιαστή και εκπαιδευτή: Το περιεχόμενο ενός Εκπαιδευτικού Λογισμικού και οι διαδικασίες τις οποίες υποστηρίζει πρέπει είναι σύμφωνες με το πρόγραμμα σπουδών. Επομένως, η ουσιαστική αλληλεπίδραση σχεδιαστή και εκπαιδευτή είναι η σχέση που έχει το εκπαιδευτικό λογισμικό με το πρόγραμμα σπουδών.

ΜΕΡΟΣ Γ: ΑΛΛΑ ΕΚΠΑΙΔΕΥΤΙΚΑ ΛΟΓΙΣΜΙΚΑ

Το Υπουργείο Παιδείας και Θρησκευμάτων έχει προμηθευτεί περισσότερους από εκατό τίτλους εκπαιδευτικών λογισμικών την τελευταία δεκαετία. Τα λογισμικά αυτά είναι αναρτημένα στην ιστοσελίδα <http://www.e-yliko.gr/resource/supportmaterial/EduAll.aspx> και στην <http://www.pischools.gr/> όπου και υπάρχει το Εκπαιδευτικό υποστηρικτικό υλικό. Ορισμένα από αυτά είναι ελεύθερα προς όλους τους ενδιαφερόμενους, ενώ άλλα διατίθεται μόνο για τις Σχολικές Μονάδες Πρωτοβάθμιας ή Δευτεροβάθμιας Εκπαίδευσης. Ορισμένα από τα εκπαιδευτικά λογισμικά που έχουν πάρει έγκριση από το παιδαγωγικό ινστιτούτο και πρόκειται να παρουσιαστούν συνοπτικά παρακάτω είναι το «Πληροφορική Α' Β' Γ' Γυμνασίου», «Αλγοριθμική και Προγραμματισμός» και το «Ταξίδι σε ένα Δίκτυο».

ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ «Πληροφορική Α' Β' Γ' Γυμνασίου»

<http://www.e-yliko.gr/Lists/List40/DispForm.aspx?ID=117>

Το λογισμικό υποστηρίζει τις εξής διδακτικές ενότητες του μαθήματος της Πληροφορικής στο Γυμνάσιο:

- i. Γνωρίζω τον υπολογιστή ως ενιαίο σύστημα
- ii. Επικοινωνώ με τον υπολογιστή
- iii. Χρήση εργαλείων έκφρασης, επικοινωνίας, ανακάλυψης και δημιουργίας
- iv. Ο υπολογιστής στο σχολείο και στην καθημερινή ζωή
- v. Διαθεματική ενότητα

Καθεμία από τις διδακτικές ενότητες περιλαμβάνει :

- Διδασκαλία
- Πρακτική εξάσκηση και επίδειξη
- Προσομοιώσεις εφαρμογών και λογισμικού
- Συνεργατικές δραστηριότητες
- Αξιολόγηση

Συνοδεύει το σχολικό εγχειρίδιο του Γυμνασίου για το μάθημα της Πληροφορικής και διατίθεται ελεύθερα για εκπαιδευτική χρήση.



Εικόνα: 8 Αρχική οθόνη λογισμικού

Βασική ιδέα σχεδιασμού του λογισμικού είναι η μεταφορά του χρήστη στο περιβάλλον ενός αεροδρομίου. Σε αυτό το σημείο, ο χρήστης επιλέγει την τάξη που τον ενδιαφέρει και μεταφέρεται στην αντίστοιχη αίθουσα αναχωρήσεων. Στην ακόλουθη εικόνα παρουσιάζονται οι θεματικές ενότητες της Α' Γυμνασίου.



Εικόνα: 9 Εσωτερικό Α' τάξης

Επιλέγοντας μία από τις εκπαιδευτικές ενότητες, ο χρήστης εισέρχεται στο αεροδρόμιο, όπου του εμφανίζεται ένα σύνολο δραστηριοτήτων, σαν αυτοκόλλητα με έντονα χρώματα πάνω σε μία βαλίτσα. Οι δραστηριότητες είναι αριθμημένες με αριθμούς από το ένα ως το έξι και αντιπροσωπεύουν τη Διδασκαλία, τη Πρακτική εξάσκηση και επίδειξη, τις Προσομοιώσεις εφαρμογών και λογισμικού, τις Συνεργατικές δραστηριότητες και την Αξιολόγηση αντίστοιχα. Οι δραστηριότητες που δεν είναι διαθέσιμες εμφανίζονται στο χρήστη αχνά και χωρίς αριθμηση.



Εικόνα: 10 Αεροδιάδρομος με όλες τις δραστηριότητες



Εικόνα: 11 Αεροδιάδρομος με μία αχνή δραστηριότητα



Εικόνα: 12 Πρακτική και εξάσκηση λογισμικού

ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ «Αλγοριθμική και Προγραμματισμός»

<http://www.e-yliko.gr/Lists/List40/DispForm.aspx?ID=178>

Το εκπαιδευτικό λογισμικό αυτό πασκοπεύει, μέσω της εντύφησης με την αλγοριθμική και τον προγραμματισμό, στην καλλιέργεια της αναλυτικής σκέψης και συνθετικής ικανότητας των μαθητών, στην ανάπτυξη ικανοτήτων μεθοδολογικού χαρακτήρα, στην καλλιέργεια της αυστηρότητας της διατύπωσης και της έκφρασης. Στόχος δεν είναι η εκμάθηση κάποιων συγκεκριμένων γλώσσας προγραμματισμού και για το λόγο αυτό το ψηφιακό περιβάλλον δεν αντιστοιχεί σε κάποιο από αυτά των γλωσσών. Κεντρικός άξονας για όλες τις δραστηριότητες είναι οι βασικές έννοιες του προγραμματισμού. Το εκπαιδευτικό πακέτο υποστηρίζει αρχεία δραστηριοτήτων και όχι απλά προγραμμάτων. Δίνει τη δυνατότητα παρακολούθησης και αλληλεπίδρασης κατά την εκτέλεση του αναπτυσσόμενου αλγορίθμου, όπως βήμα προς βήμα εκτέλεση, παρακολούθηση μεταβλητών και πίνακα τιμών.

Το λογισμικό «Αλγοριθμική και προγραμματισμός» αποτελείται από τα παρακάτω στοιχεία:

- Χώρος Δραστηριοτήτων
- Δραστηριότητες για τον μαθητή
- Δραστηριότητες για τον καθηγητή, με εκτενή σχόλια και λύσεις
- Διερμηνευτής της ΓΛΩΣΣΑΣ
- Δημιουργός Διαγραμμάτων Ροής

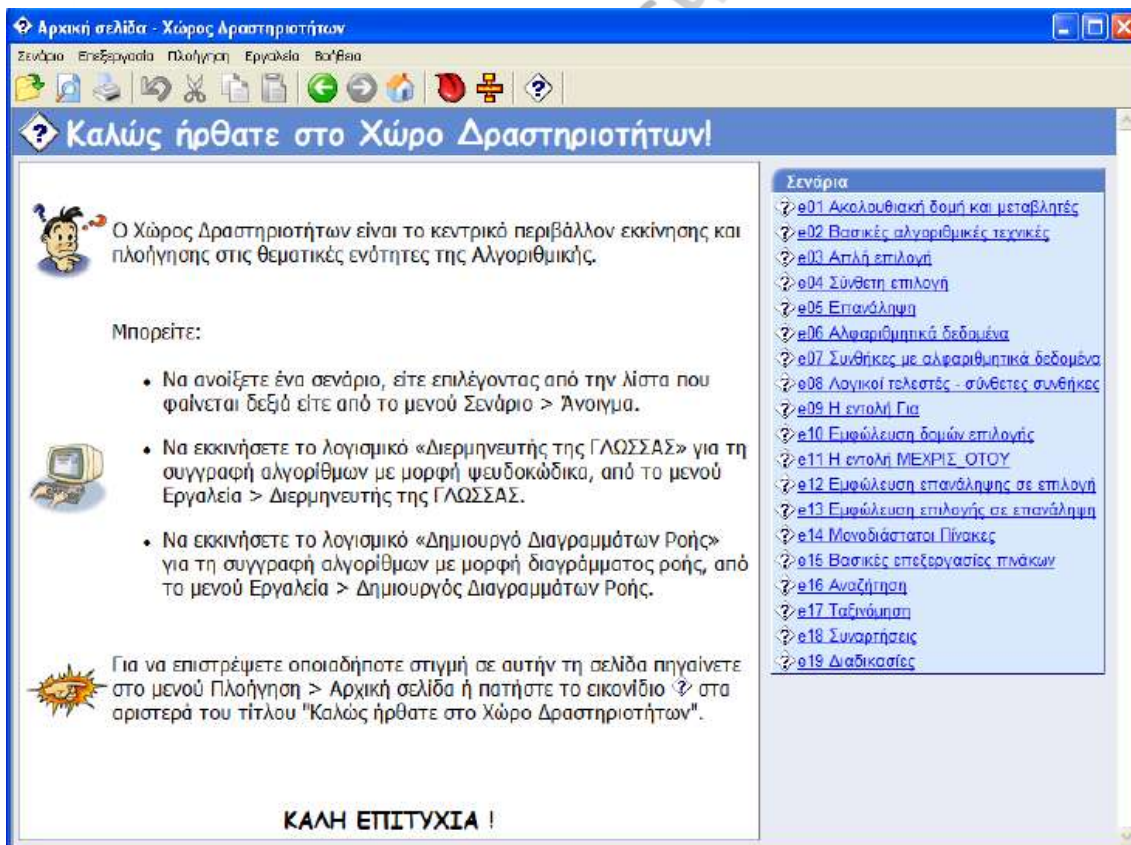
Οι απαιτήσεις σε υλικό είναι οι παρακάτω:

- Οι απαιτήσεις σε υλικό της Αλγοριθμικής είναι ιδιαίτερα χαμηλές:
- Επεξεργαστής 486 συχνότητας 66MHz (προτείνεται Pentium@200MHz)
- Μνήμη RAM 16 Mb (προτείνεται 32 Mb)

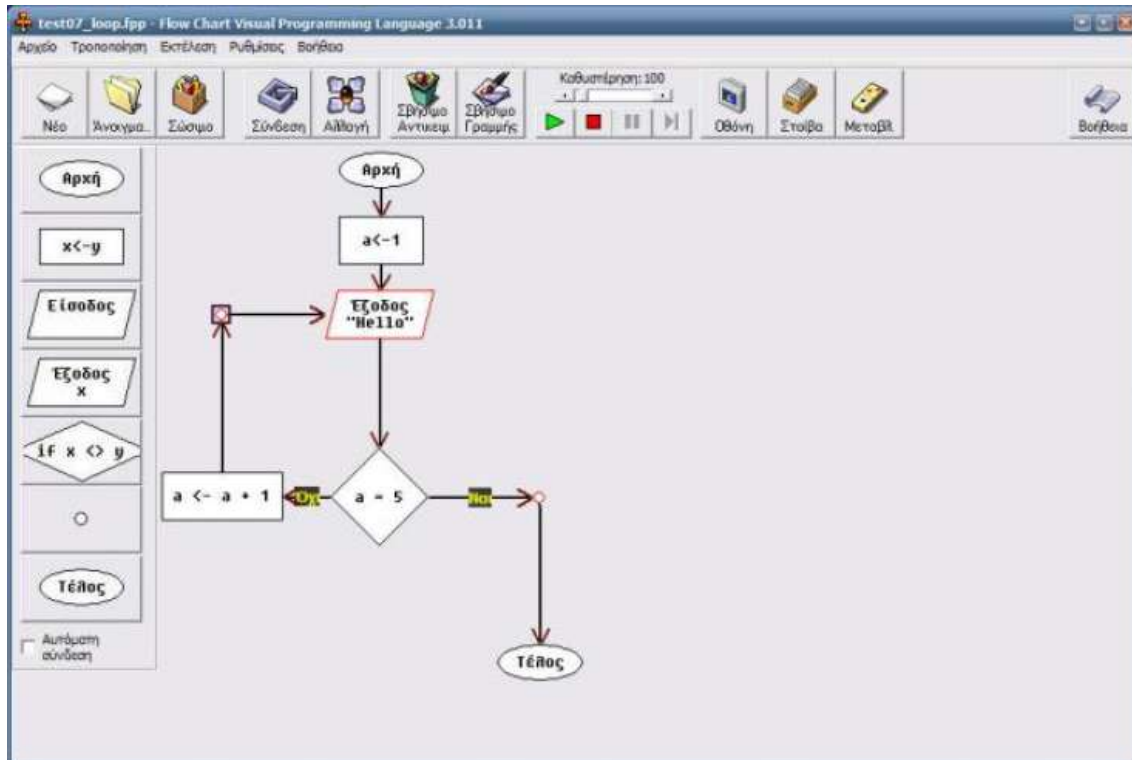
- Οποιαδήποτε κάρτα γραφικών τύπου VGA (προτείνεται ανάλυση τουλάχιστον 800x600 με 256 χρώματα)
- Οποιαδήποτε οθόνη τύπου VGA (προτείνεται έγχρωμη τουλάχιστον 800x600)
- 100 Mb ελεύθερου χώρου στον σκληρό δίσκο
- Μονάδα CD-ROM οποιασδήποτε ταχύτητας ή κοινόχρηστος δικτυακός φάκελος για την εγκατάσταση του CD
- Προτείνεται η ύπαρξη ποντικιού, μάλιστα για ορισμένες δραστηριότητες είναι απαραίτητη.

Απαιτήσεις σε λογισμικό είναι οι εξής:

- Οποιαδήποτε έκδοση των ελληνικών ή αγγλικών Windows (32 bit): 95, 98, Me, NT, 2000, XP, 2003
- Internet Explorer 4.0 ή νεότερος (η εγκατάσταση του I.E. 4.0 στα NT προϋποθέτει SP3 ή νεότερο). Προτείνεται I.E. 5.5 ή νεότερος. Μπορείτε να αναβαθμίσετε δωρεάν τον Internet Explorer σας από συνοδευτικά CD περιοδικών ή από την ιστοσελίδα της Microsoft, <http://www.microsoft.com/windows/ie/default.mspx>.
- Ειδικά για τα Windows 95, για την εμφάνιση των δραστηριοτήτων θα πρέπει να εγκατασταθεί το HTML Help Installation and Update Package το οποίο και εμπεριέχεται στο CD της Αλγοριθμικής. Οι νεότερες εκδόσεις των Windows έχουν προεγκατεστημένο το λογισμικό HTML Help.



Εικόνα: 12 Αρχική οθόνη λογισμικού

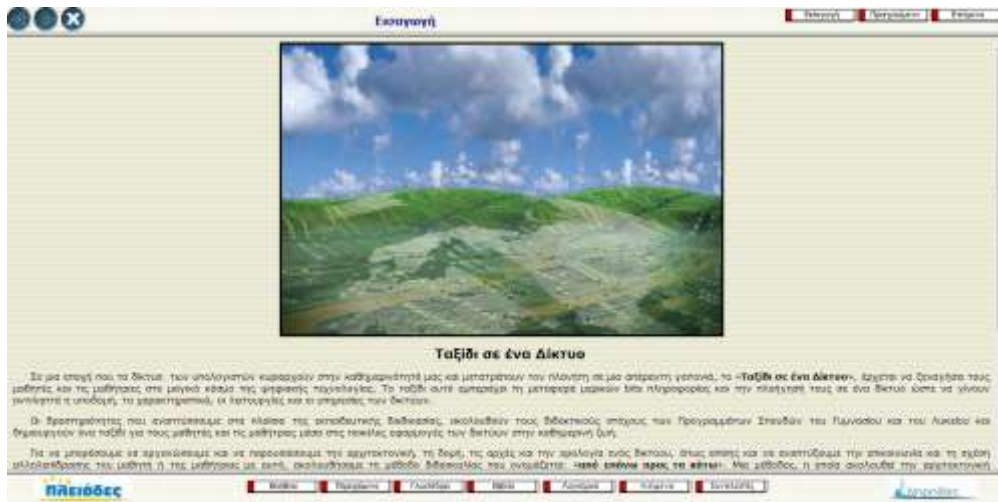


Εικόνα: 13 Δημιουργία διαγραμμάτων μέσω του λογισμικού

ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ «Ταξίδι σε ένα Δίκτυο»

<http://www.e-yliko.gr/Lists/List40/DispForm.aspx?ID=6>

Το εκπαιδευτικό πακέτο με τίτλο «Ταξίδι σ' ένα δίκτυο» έχει ως κεντρικό άξονα για όλες τις δραστηριότητές του τα Δίκτυα Υπολογιστών. Συγκεκριμένα, έχει αναπτυχθεί ένα ψηφιακό ταξίδι στα δίκτυα των υπολογιστών. Η μεταφορά μερικών bits πληροφορίας και η πλοήγησή τους στο δίκτυο αυτό, αποτελεί το «ταξίδι» στην υποδομή, στα χαρακτηριστικά, στις υπηρεσίες κ.ά. των δικτύων. Το εκπαιδευτικό πακέτο στοχεύει στην απόκτηση γνώσεων και στην ανάπτυξη δεξιοτήτων που αφορούν στις επικοινωνίες δεδομένων και στα δίκτυα υπολογιστών (με έμφαση στη δομή, στην ταξινόμηση και στην περιγραφή των προτύπων τους). Οι δραστηριότητες του πακέτου δίνουν τη δυνατότητα στο μαθητή να παρακολουθεί δικτυακές λειτουργίες και τεχνικές, να δημιουργεί τα «δικά του δίκτυα», να παρακολουθεί τον τρόπο λειτουργίας τους και την απόδοσή τους και να επεμβαίνει με τις δικές του επιλογές στη διαχείρισή τους.



Εικόνα: 14 Αρχική οθόνη λογισμικού



Εικόνα: 15 Προβολή θεωρίας εκπαιδευτική ενότητα λογισμικού

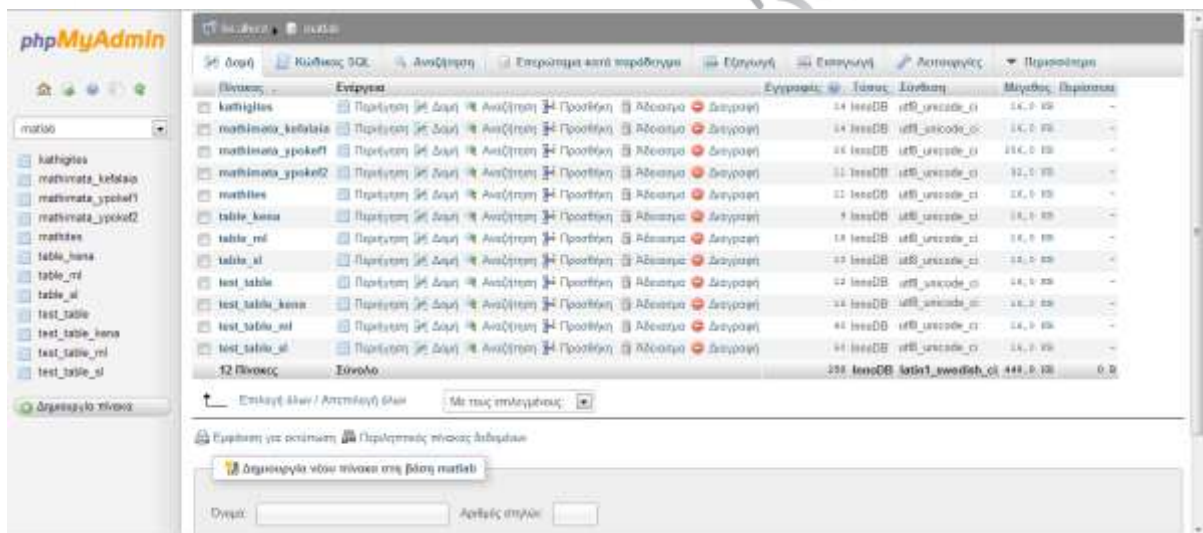


Εικόνα: 16 Περιβάλλον ασκήσεων λογισμικού

[3] ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

3.1. Σχεδιασμός βάσης δεδομένων λογισμικού

Ένα από πιο σημαντικά μέρη κάθε λογισμικού, συμπεριλαμβανομένων και των εκπαιδευτικών λογισμικών, είναι ο σχεδιασμός της βάσης δεδομένων στην οποία καταχωρούνται όλα τα δεδομένα που χρειάζονται για την ομαλή λειτουργία του. Η βάση δεδομένων ενός εκπαιδευτικού λογισμικού περιλαμβάνει όλες τις απαραίτητες πληροφορίες για τους εκπαιδευτές, τους εκπαιδευομένους, τη θεωρία και τις ασκήσεις που αφορούν τις ενότητες των μαθημάτων. Η προσεκτική λοιπόν οργάνωση της βάσης δεδομένων είναι άκρως σημαντική αφού είναι απαραίτητο να διασφαλιστεί η ασφάλεια των δεδομένων, η αποφυγή πλεοναζόντων εγγραφών και η εύκολη αναζήτηση των δεδομένων. Στη παρούσα διατριβή η βάση δεδομένων κατασκευάστηκε σε MySQL στο περιβάλλον του XAMPP Server.



Εικόνα: 17 Βάση Δεδομένων

Η βάση δεδομένων της εφαρμογής αποτελείται από δώδεκα πίνακες οι οποίοι αναλύονται παρακάτω:

Μαθητές (Κωδ-μαθητή, Όνομα-μαθητή, Επώνυμο-μαθητή, Όνομα-πατέρα-μαθητή,

Email-μαθητή, Διεύθυνση-μαθητή, Τηλέφωνο-μαθητή, Όνομα-χρήστη-μαθητή,

Κωδικός-μαθητή)

Καθηγητές (Κωδ-καθηγητή , Όνομα-καθηγητή, Επώνυμο-καθηγητή , Email-καθηγητή,

Όνομα-χρήστη-καθηγητή, Κωδικός- καθηγητή)

Κεφάλαια (Κωδ-κεφαλαίου, Προτεραιότητα-κεφαλαίου, Τίτλος-κεφαλαίου, Κείμενο-κεφαλαίου)

ΥποκεφάλαιαΑ (Κωδ-υποκεφαλαίουΑ, Κωδ-κεφαλαίου, Προτεραιότητα-υποκεφαλαίουΑ, Τίτλος-υποκεφαλαίουΑ, Κείμενο-υποκεφαλαίουΑ)

ΥποκεφάλαιαΒ (Κωδ-υποκεφαλαίουΒ, Κωδ-υποκεφαλαίουΑ, Προτεραιότητα-υποκεφαλαίουΒ, Τίτλος-υποκεφαλαίουΒ, Κείμενο-υποκεφαλαίουΒ)

Ερωτήσεις-κενών (Κωδ-κενών, Κωδ-κεφαλαίου, Ερώτηση-κενά, Απάντηση-κενά)

Ερωτήσεις-πολλαπλής (Κωδ-πολλαπλής, Κωδ-κεφαλαίου, Ερώτηση-πολλαπλής, Απάντηση1-πολλαπλής, Απάντηση2-πολλαπλής, Απάντηση3-πολλαπλής)

Ερωτήσεις-Σ/Λ (Κωδ- Σ/Λ, Κωδ-κεφαλαίου, Ερώτηση- Σ/Λ, Απάντηση- Σ/Λ)

ΤΕΣΤ (Κωδ-τεστ, Κωδ-μαθητή, Κωδ-κεφαλαίου, Ωρα-έναρξης-τεστ, Βαθμός-μαθητή)

ΤΕΣΤ-κενών (Κωδ-τεστ-κενών, Κωδ-τεστ, Κωδ-κενών, Απάντηση-μαθητή)

ΤΕΣΤ-πολλαπλής (Κωδ-τεστ-πολλαπλής, Κωδ-τεστ, Κωδ-πολλαπλής, Απάντηση-μαθητή)

ΤΕΣΤ-Σ/Λ (Κωδ-τεστ-Σ/Λ, Κωδ-τεστ, Κωδ-Σ/Λ, Απάντηση-μαθητή)

ΤΕΛΙΚΟ (Κωδ-τεστ-τελικό, Κωδ-μαθητή, Κωδ-κεφαλαίου, Ωρα-έναρξης-τεστ, Βαθμός-μαθητή)

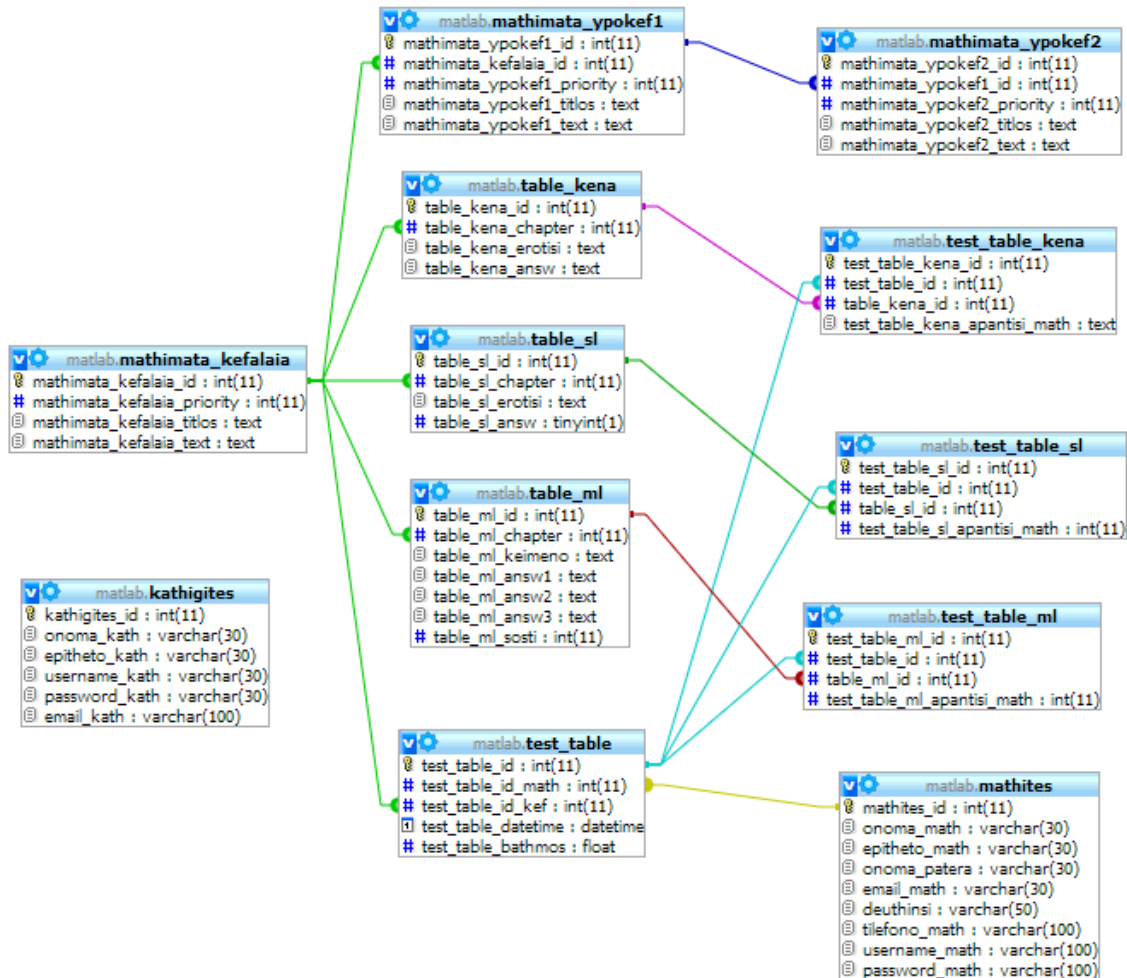
ΤΕΛΙΚΟ -κενών (Κωδ-τεστ-τελικό-κενών, Κωδ-τεστ-τελικό, Κωδ-κενών, Απάντηση-μαθητή)

ΤΕΛΙΚΟ -πολλαπλής (Κωδ-τεστ-τελικό-πολλαπλής, Κωδ-τεστ-τελικό, Κωδ-πολλαπλής, Απάντηση-μαθητή)

ΤΕΛΙΚΟ -Σ/Λ (Κωδ-τεστ-τελικό-Σ/Λ, Κωδ-τεστ-τελικό, Κωδ-Σ/Λ, Απάντηση-μαθητή)

ΛΑΘΟΙ (Κωδ-λαθους, Κωδ-μαθητή , Τύπος-ερώτησης, Κωδ- ερώτησης)

ΒΙΝΤΕΟ (Κωδ-βίντεο, Κωδ-κεφάλαιο , βίντεο-όνομα, βίντεο-μέγεθος, βίντεο-τύπος)



Εικόνα: 18 Σχισιακό Μοντέλο Βάσης Δεδομένων

3.2. Εργαλεία και Τεχνολογίες που χρησιμοποιήθηκαν

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν οι παρακάτω τεχνολογίες και εργαλεία:

HTML (Hyper Text Markup Language): Η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, όπου τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

CSS (Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ): Γλώσσα υπολογιστή που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου, το οποίο έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML και γενικότερα ενός ιστότοπου. Διαμορφώνει χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

PHP: Γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

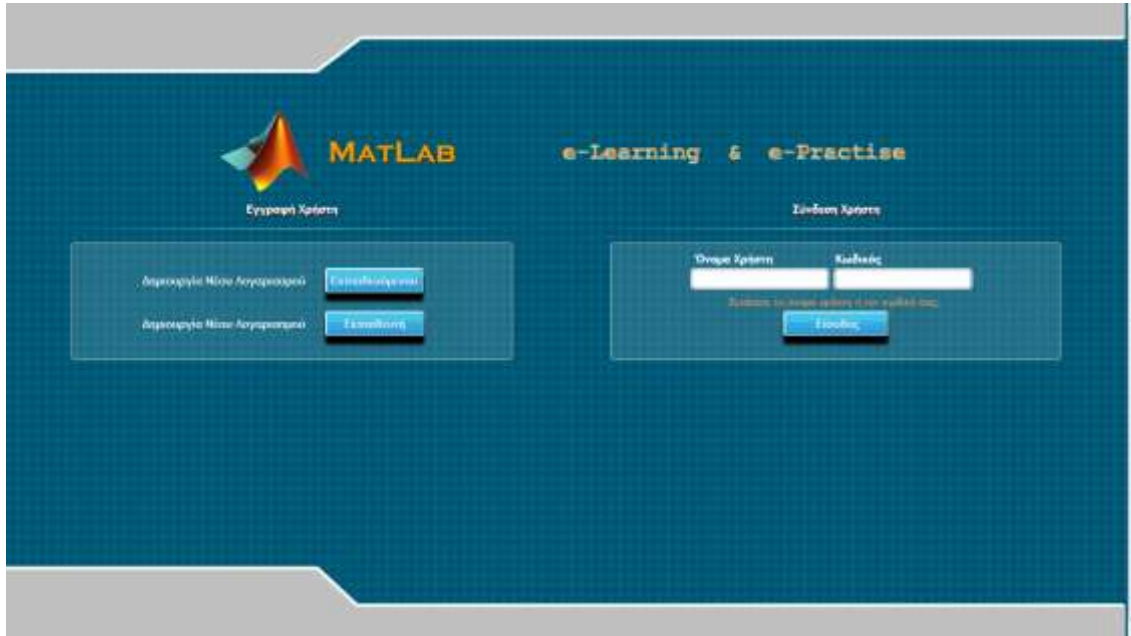
JavaScript: Γλώσσα προγραμματισμού που έχει ως σκοπό την παραγωγή δυναμικού περιεχομένου και την εκτέλεση κώδικα προς την πλευρά του πελάτη (client-side) σε ιστοσελίδες. Η σύνταξή της είναι επηρεασμένη από τη C, αλλά αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java. Ωστόσο, αυτές οι δύο αυτές γλώσσες έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

jQuery: Βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιήσει τη υλοποίηση σεναρίων (scripting) στη πλευρά του πελάτη (client-side) της HTML και να υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig) και χρησιμοποιείται σε πάνω από το 65% των 10.000 ιστοτόπων με τη μεγαλύτερη επισκεψιμότητα. Η jQuery είναι ελεύθερο λογισμικό, με άδεια MIT.

XAMPP: Ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl.

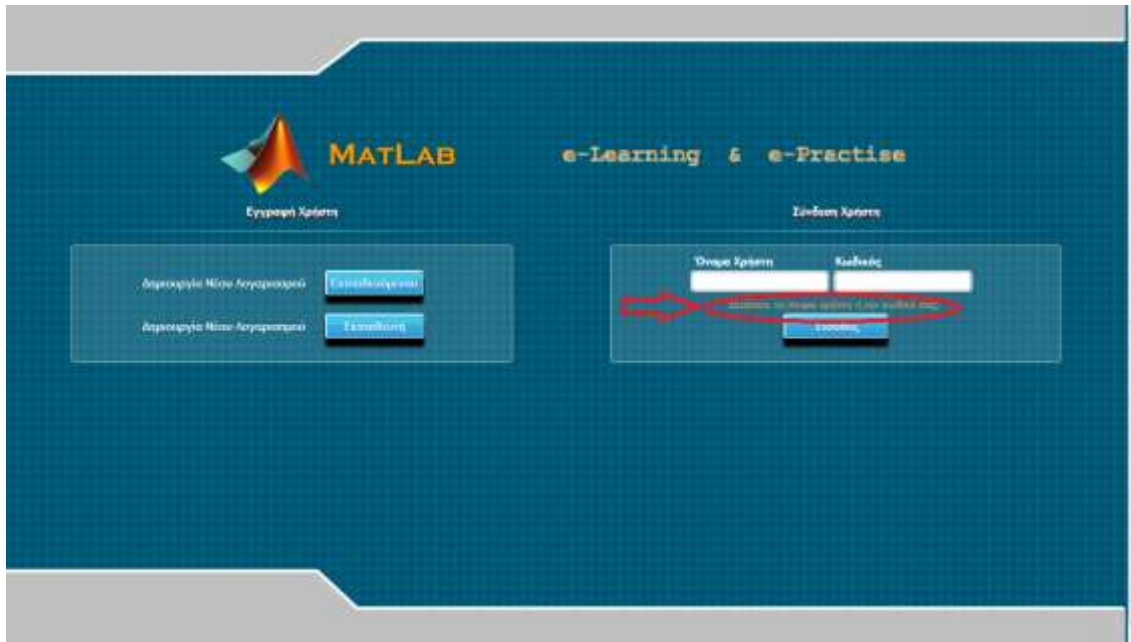
[4] ΠΑΡΟΥΣΙΑΣΗ ΚΑΙ ΧΡΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Σε αυτό το κεφάλαιο θα παρουσιαστούν δύο σενάρια λειτουργίας. Ένα σενάριο λειτουργίας με ρόλο εκπαιδευόμενου και ένα σενάριο λειτουργίας με ρόλο εκπαιδευτή. Την πρώτη φορά που εισέρχεται κάποιος στην εφαρμογή συναντά την παρακάτω εικόνα:



Εικόνα: 19 Οθόνη έναρξης Λογισμικού

Εφόσον ο χρήστης, είτε πρόκειται για εκπαιδευόμενο είτε για εκπαιδευτή, δεν έχει ξαναχρησιμοποιήσει το λογισμικό, πρέπει να δημιουργήσει ένα νέο λογαριασμό εγγραφής για να μπορεί να εισέλθει στο κεντρικό μενού. Η εγγραφή κάθε χρήστη διαφέρει και για αυτό το λόγο θα αναλυθεί στο αντίστοιχο σενάριο λειτουργίας του χρήστη. Προκειμένου ένας χρήστης να συνδεθεί στο σύστημα είναι απαραίτητο να εισάγει το όνομα χρήστη και τον κωδικό, που δήλωσε κατά την εγγραφή του, στα αντίστοιχα πεδία. Στην περίπτωση όπου έχει ξεχάσει το όνομα χρήστη ή/και τον κωδικό μπορεί να επιλέξει μέσα από τον σύνδεσμο «Ξεχάσατε το όνομα χρήστη ή τον κωδικό σας;» να μεταφερθεί στη σελίδα επαναπροστολής των στοιχείων του.



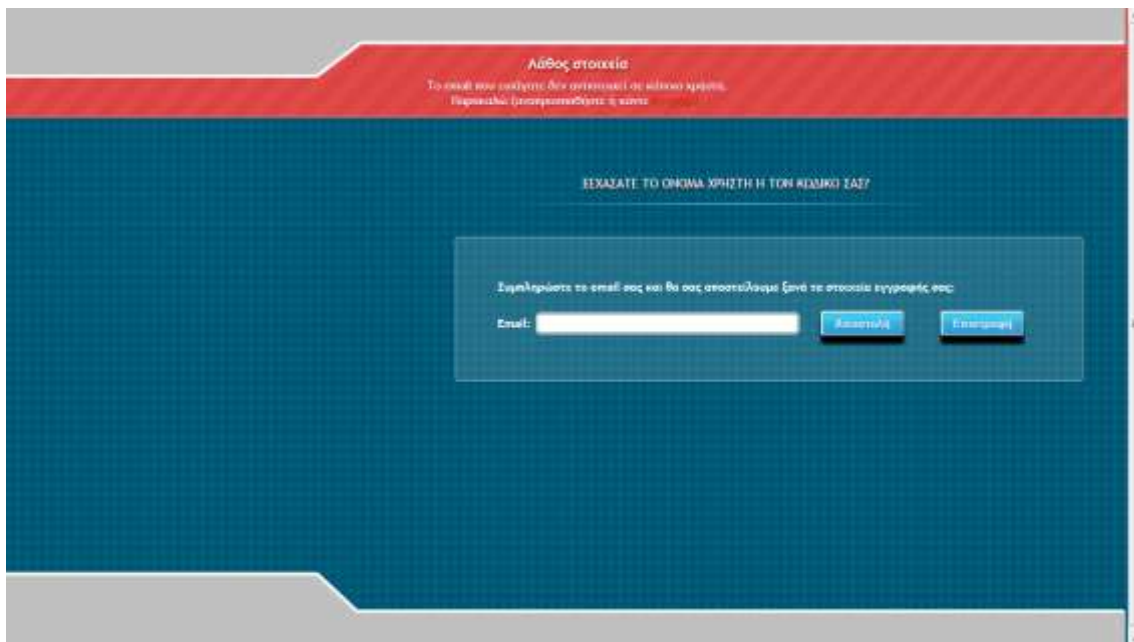
Εικόνα: 20 Σύνδεσμος μετάβασης στη σελίδα επαναποστολής στοιχείων

Στην παρακάτω εικόνα παρουσιάζεται η σελίδα επαναποστολής των στοιχείων ενός χρήστη μέσω ηλεκτρονικού ταχυδρομείου. Ο χρήστης (εκπαιδευτής/εκπαιδευόμενος) μπορεί να εισάγει τη διεύθυνση ηλεκτρονικού ταχυδρομείου (email) του στο πεδίο *Email* και να πατήσει το κουμπί αποστολή. Αν το λογισμικό εντοπίσει κάποιον εγγεγραμμένο χρήστη με αυτό το email, τότε αποστέλλει τα στοιχεία εγγραφής, τα οποία είναι απαραίτητα για τη σύνδεσή του στο λογισμικό. Πιο συγκεκριμένα του αποστέλλει, στο ηλεκτρονικό του ταχυδρομείο, το όνομα χρήστη και τον κωδικό πρόσβασης.



Εικόνα: 21 Σελίδα επαναποστολής στοιχείων

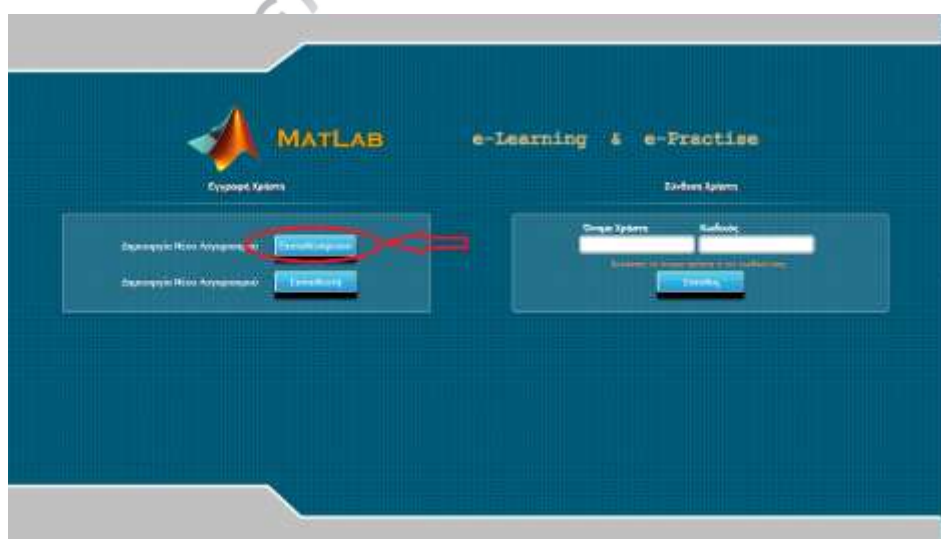
Στην περίπτωση όπου ο χρήστης εισάγει λανθασμένα το email του ή το λογισμικό δεν εντοπίζει κάποιον χρήστη με αυτή τη διεύθυνση ηλεκτρονικού ταχυδρομείου, τότε εμφανίζει το μήνυμα λάθους που φαίνεται στην παρακάτω εικόνα.



Εικόνα: 22 Σελίδα επαναποστολής στοιχείων

Σενάριο λειτουργίας με ρόλο εκπαιδευόμενου

Προκειμένου ένας καινούριος εκπαιδευόμενος, να δημιουργήσει ένα λογαριασμό εισόδου στο σύστημα είναι απαραίτητο να επιλέξει τη μετάβασή του στην αντίστοιχη σελίδα εγγραφής μέσω του κουμπιού που φαίνεται στην παρακάτω εικόνα:



Εικόνα: 23 Κουμπί μετάβασης στη σελίδα εγγραφής εκπαιδευόμενου

Όταν ο εκπαιδευόμενος πατήσει το κουμπί, που αναφέρθηκε παραπάνω, θα μεταβεί στην εξής σελίδα:

Εικόνα: 24 Σελίδα εγγραφής εκπαιδευόμενου

Σε αυτό το σημείο ο εκπαιδευόμενος θα πρέπει να εισάγει το όνομα, το επίθετο, το όνομα του πατέρα του, το email του, το όνομα χρήστη και τον κωδικό που θα χρησιμοποιεί κατά τη σύνδεσή του στο λογισμικό. Προαιρετικά αν το επιθυμεί μπορεί να εισάγει διεύθυνση και ένα τηλέφωνο του.

Στην περίπτωση όπου έχει κάνει κάποιο λάθος κατά την πληκτρολόγηση των στοιχείων του και έτσι επιθυμεί να σβήσει όλα τα πεδία που συμπλήρωσε, μπορεί να πατήσει το κουμπί **Επαναφορά**. Μέσω αυτού του κουμπιού διαγράφονται αυτόματα όλες οι τιμές των πεδίων.

Εικόνα: 25 Κουμπί επαναφοράς πεδίων

Αν ο εκπαιδευόμενος θέλει να επιστρέψει στην αρχική σελίδα χωρίς να εγγραφεί, τότε μπορεί να πατήσει το κουμπί **Επιστροφή**, όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα: 26 Κουμπί επιστροφής στην αρχική σελίδα

Αλλιώς, αν ο χρήστης έχει εισάγει όλα τα απαραίτητα στοιχεία και επιθυμεί να ολοκληρώσει την εγγραφή του, μπορεί να πατήσει το αντίστοιχο κουμπί **Εγγραφή**.



Εικόνα: 27 Κουμπί ολοκλήρωσης εγγραφής

Ωστόσο, στην περίπτωση όπου έχει εισάγει λάθος στοιχεία, η εφαρμογή τον ενημερώνει, εμφανίζοντάς του το αντίστοιχο μήνυμα λάθους. Στην παρακάτω εικόνα φαίνεται η εμφάνιση του μηνύματος λάθους όταν ο χρήστης έχει εισάγει λάθος διεύθυνση ηλεκτρονικού ταχυδρομείου.



Εικόνα: 28 Εμφάνιση μηνύματος για λάθος email χρήστη

Στην περίπτωση όπου ο εκπαιδευόμενος έχει ήδη εγγραφεί, το λογισμικό εμφανίζει στο χρήστη το μήνυμα λάθους που φαίνεται στην παρακάτω εικόνα. Πιο συγκεκριμένα, αν κάποιος χρήστης έχει ξαναχρησιμοποιήσει το ίδιο επίθετο, το ίδιο όνομα πατέρα και την ίδια διεύθυνση ηλεκτρονικού ταχυδρομείου σε προηγούμενη εγγραφή του, τότε το λογισμικό τον ενημερώνει με το ανάλογο μήνυμα.



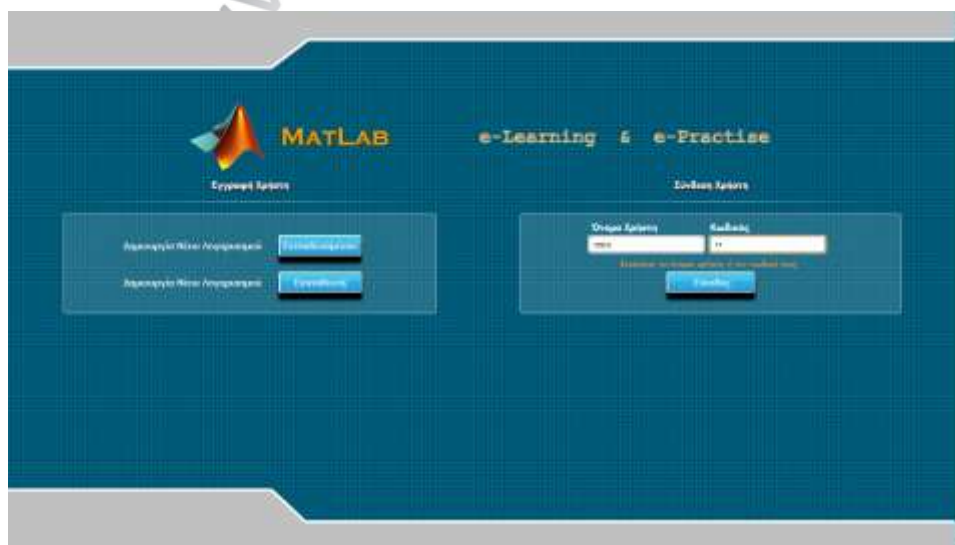
Εικόνα: 29 Εμφάνιση μηνύματος για λάθος στοιχεία χρήστη

Αν αφότου ο χρήστης εισάγει τα στοιχεία του και πατήσει το κουμπί εγγραφής, δε του εμφανιστεί κανένα μήνυμα λάθους, τότε η εφαρμογή θα του εμφανίσει ένα μήνυμα επιτυχούς εγγραφής όπως φαίνεται στην παρακάτω εικόνα. Πατώντας το πλήκτρο **Σύνδεση** το λογισμικό θα τον μεταφέρει στη σελίδα σύνδεσης λογαριασμού, ενώ πατώντας πάνω στο μήνυμα, ο χρήστης θα παραμείνει στη σελίδα εγγραφής νέου χρήστη, για να εγγράψει πιθανότατα κάποιον άλλον εκπαιδευόμενο. Ωστόσο, και στις δύο περιπτώσεις η εγγραφή του εκπαιδευόμενου έχει ολοκληρωθεί επιτυχώς.



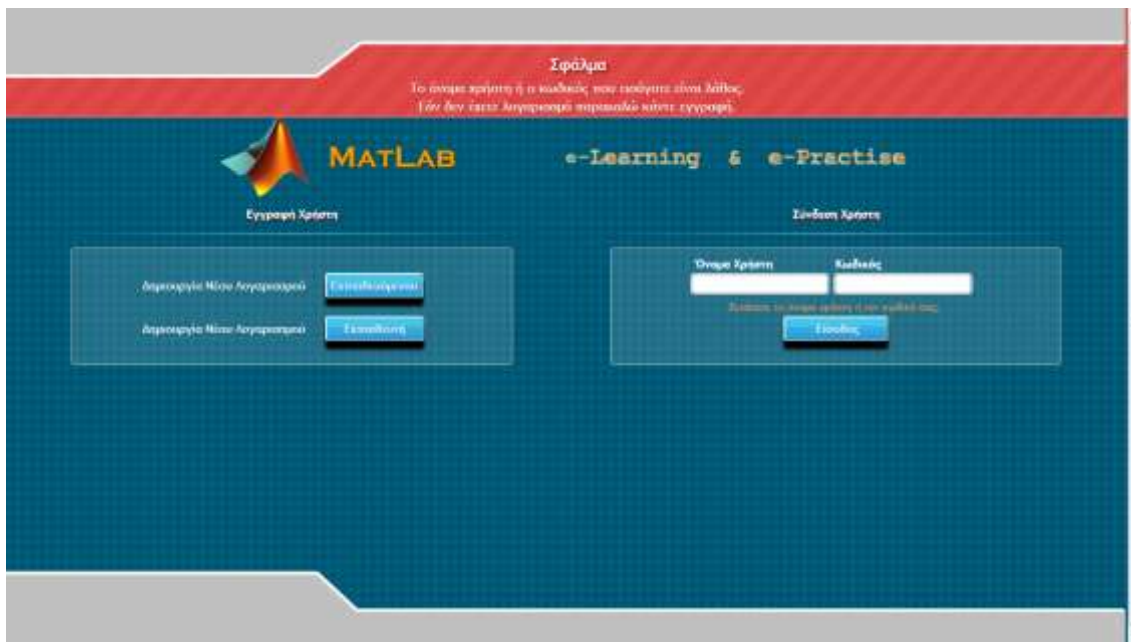
Εικόνα: 30 Εμφάνιση μηνύματος επιτυχής εγγραφής χρήστη

Μετά την επιτυχή εγγραφή του εκπαιδευόμενου, αν ο χρήστης επιλέξει τη μετάβαση του μέσω του συνδέσμου **Σύνδεση**, το λογισμικό τον στέλνει στη σελίδα σύνδεσης όπου μπορεί να εισάγει το όνομα χρήστη και τον κωδικό που δήλωσε κατά την εγγραφή του στα αντίστοιχα πεδία, όπως φαίνεται στην ακόλουθη εικόνα.



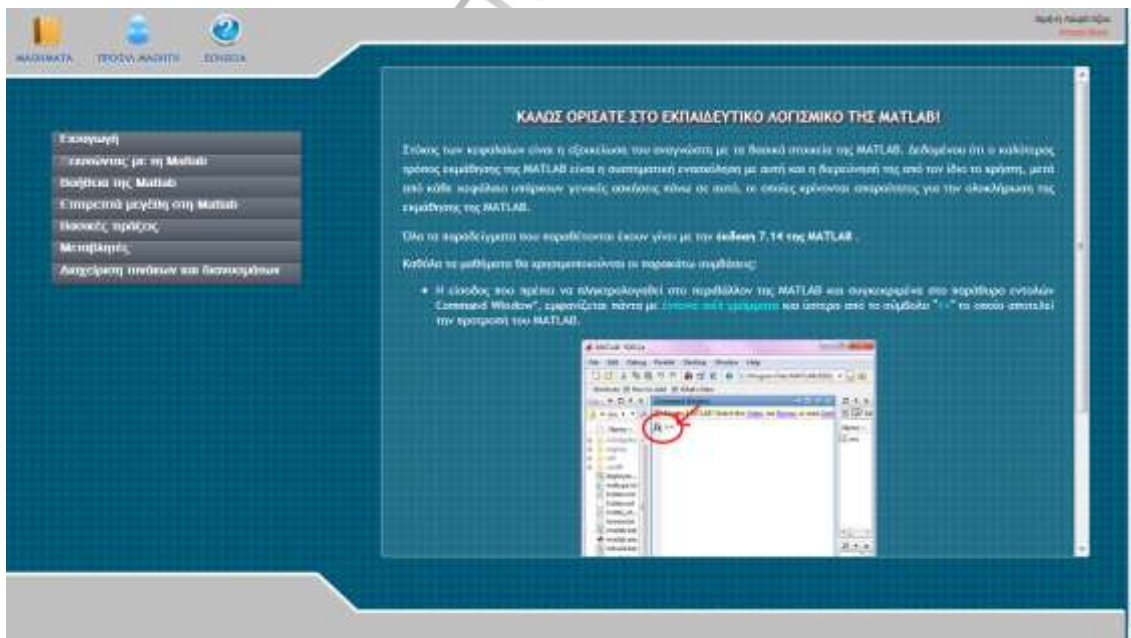
Εικόνα: 31 Σελίδα σύνδεσης χρήστη

Αν ο εκπαιδευόμενος εισάγει λάθος τα στοιχεία του τότε το λογισμικό του εμφανίζει το ακόλουθο μήνυμα λάθους:



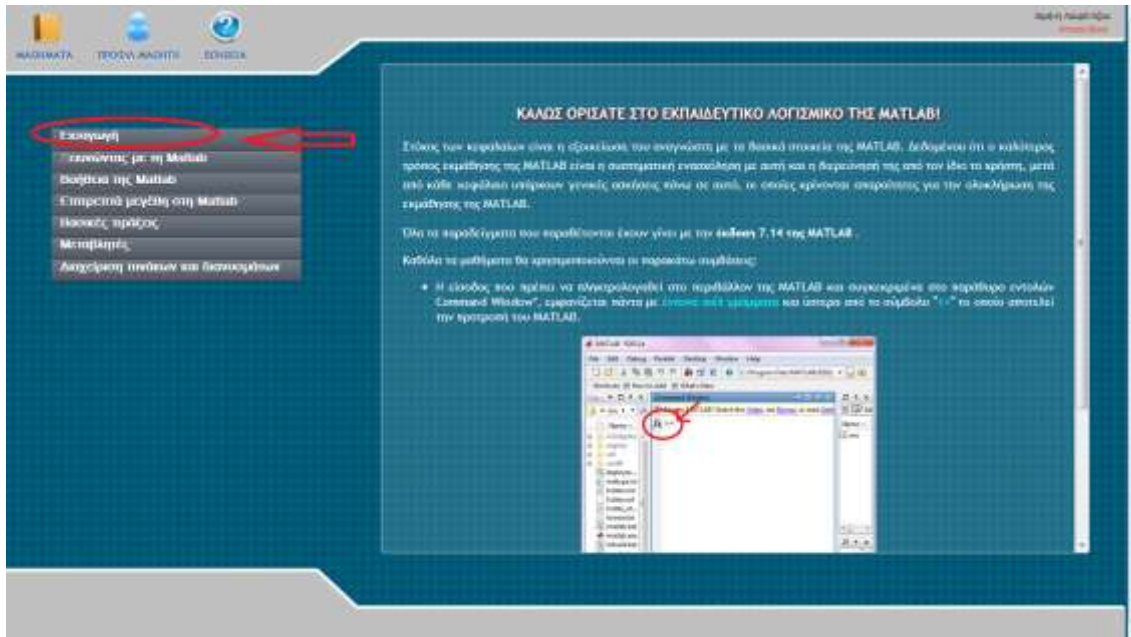
Εικόνα: 32 Μήνυμα λάθους στοιχείων χρήστη

Αν τα στοιχεία σύνδεσης του χρήστη είναι σωστά, τότε μεταφέρεται στην κεντρική σελίδα μαθητή που φαίνεται στην παρακάτω εικόνα. Όπως παρατηρούμε ο εκπαιδευόμενος έχει τρεις κεντρικές επιλογές στο πάνω μέρος της σελίδας όπου και εμφανίζεται το μενού. Η περιοχή εμφάνισης του μενού παραμένει σταθερή καθ' όλη την περιήγηση του χρήστη στο λογισμικό.



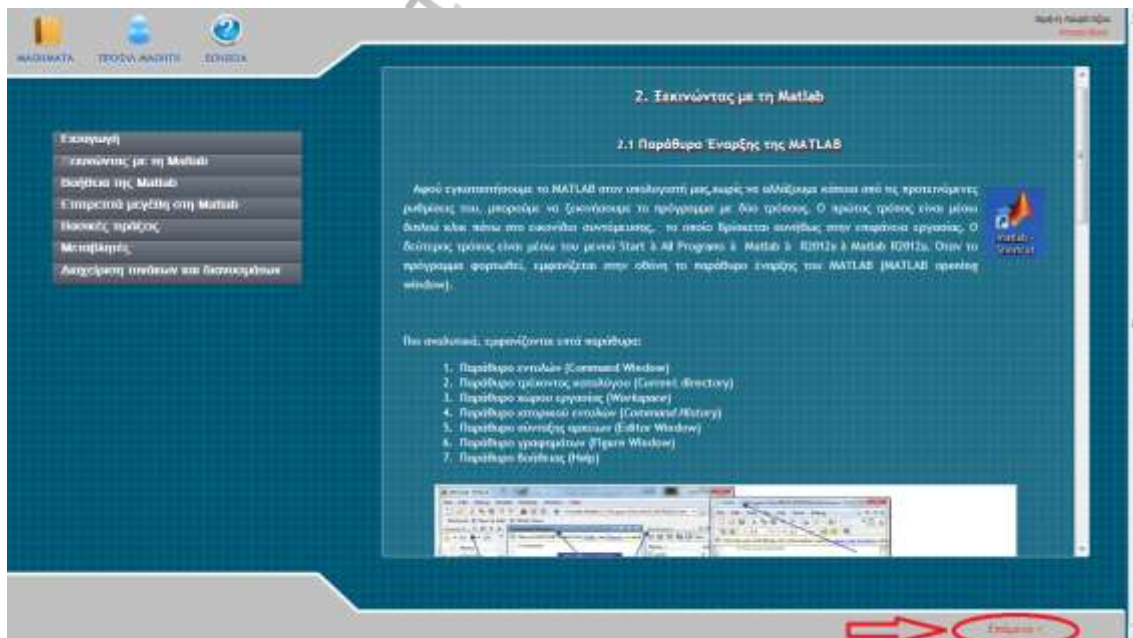
Εικόνα: 33 Κεντρική σελίδα εκπαιδευόμενου

Για την προβολή των μαθημάτων ο εκπαιδευόμενος πατάει επάνω στην ονομασία των κεφαλαίων που τον ενδιαφέρει. Στην περίπτωση όπου επιθυμεί να διαβάσει τη θεωρία από την αρχή επιλέγει το πρώτο κεφάλαιο, ενώ στην περίπτωση όπου έχει διαβάσει ήδη ορισμένα κεφάλαια, τότε μπορεί να επιλέξει απευθείας το κεφάλαιο που επιθυμεί.



Εικόνα: 34 Επιλογή προβολής θεωρίας πρώτου Κεφαλαίου

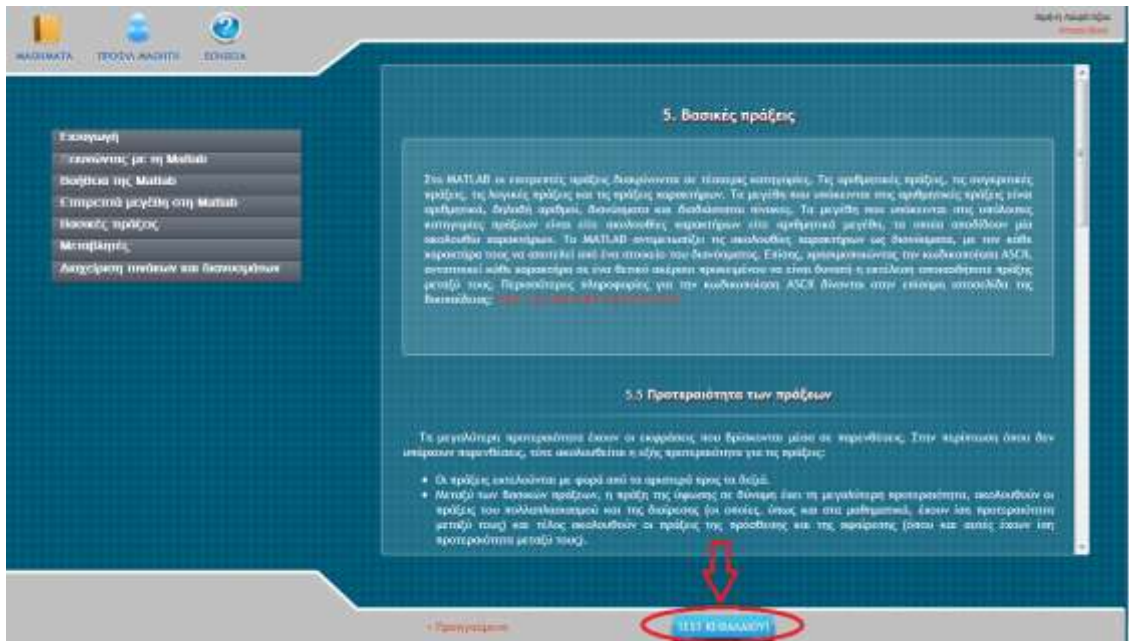
Αμέσως το λογισμικό του προβάλλει τη θεωρία του κεφαλαίου όπως φαίνεται παρακάτω:



Προβολή θεωρίας κεφαλαίου 2. Ξεκινώντας με τη Matlab, υποκεφάλαιο 2.1

Στο τέλος κάθε μαθήματος υπάρχει ένας σύνδεσμος με τίτλο **Επόμενο**. Πατώντας αυτή την επιλογή ο χρήστης μεταφέρεται στη θεωρία του επόμενου μαθήματος. Αντίστοιχη επιλογή

υπάρχει και για να μεταφερθεί ο χρήστης σε προηγούμενο μάθημα. Στο τελευταίο μάθημα κάθε κεφαλαίου το λογισμικό ενημερώνει τον εκπαιδευόμενο ότι έχει ολοκληρώσει τη θεωρία του κεφαλαίου και του παρέχει τη δυνατότητα να ελέγξει τις γνώσεις του κάνοντας το test του αντίστοιχου κεφαλαίου.



Εικόνα: 35 Κουμπί μετάβασης χρήστη στο διαγώνισμα του κεφαλαίου που ολοκλήρωσε

Πριν την έναρξη του διαγωνίσματος, το λογισμικό εμφανίζει την παρακάτω οθόνη στον εκπαιδευόμενο ενημερώνοντάς τον για τους τύπους των ερωτήσεων, το ποσοστό συμμετοχής κάθε ερώτησης και τον κατώτερο βαθμό επιτυχίας του κάθε διαγωνίσματος.



Εικόνα: 36 Ενημερωτική σελίδα χρήστη πριν το διαγώνισμα

Πατώντας το κουμπί **ΕΝΑΡΞΗ ΤΕΣΤ** εμφανίζεται η πρώτη ερώτηση στον εκπαιδευόμενο όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα: 37 Προβολή πρώτης ερώτησης διαγωνίσματος

Αφού ο εκπαιδευόμενος επιλέξει μία απάντηση μπορεί να πατήσει το κουμπί **Επόμενο** προκειμένου να μεταβεί στην επόμενη ερώτηση.



Εικόνα: 38 Προβολή δεύτερης ερώτησης διαγωνίσματος

Όταν ο χρήστης απαντήσει και στην τελευταία ερώτηση το λογισμικό θα του εμφανίσει τα αποτελέσματα:

The image displays two screenshots of a MATLAB exam interface. The top screenshot shows the exam summary with a score of 71.43 and a question about matrix multiplication. The bottom screenshot shows the same question with a score of 12 and a second question about matrix operations.

Top Screenshot: Exam Summary

ΜΟΛΙΣ ΟΛΟΚΛΗΡΩΣΑΤΕ ΤΟ ΤΕΣΤ!

ΣΩΣΤΕΣ	ΛΑΘΕΣ	ΣΥΝΟΛΟ	ΒΑΘΜΟΣ	ΑΠΟΤΕΛΕΣΜΑ
5	2	7	71.43	ΠΡΟΣΒΑΣΗ

Συνεπείστε μόλις ολοκληρώσετε απαντώντας το ΤΕΣΤ!

Παρέχονται φιλάνοι τα αποτελέσματα των απαντήσεων σας:

Ερώτηση: 1

Ποιο αποτέλεσμα εμφανίζεται ως μας απαντάμε η πληκτρολόγηση της εντολής `3*2` στο Περιβάριο εντολών του MATLAB;

Bottom Screenshot: Question 1 and Question 2

Ερώτηση: 1

Ποιο αποτέλεσμα εμφανίζεται ως μας απαντάμε η πληκτρολόγηση της εντολής `3*2` στο Περιβάριο εντολών του MATLAB;

0 4

1

12

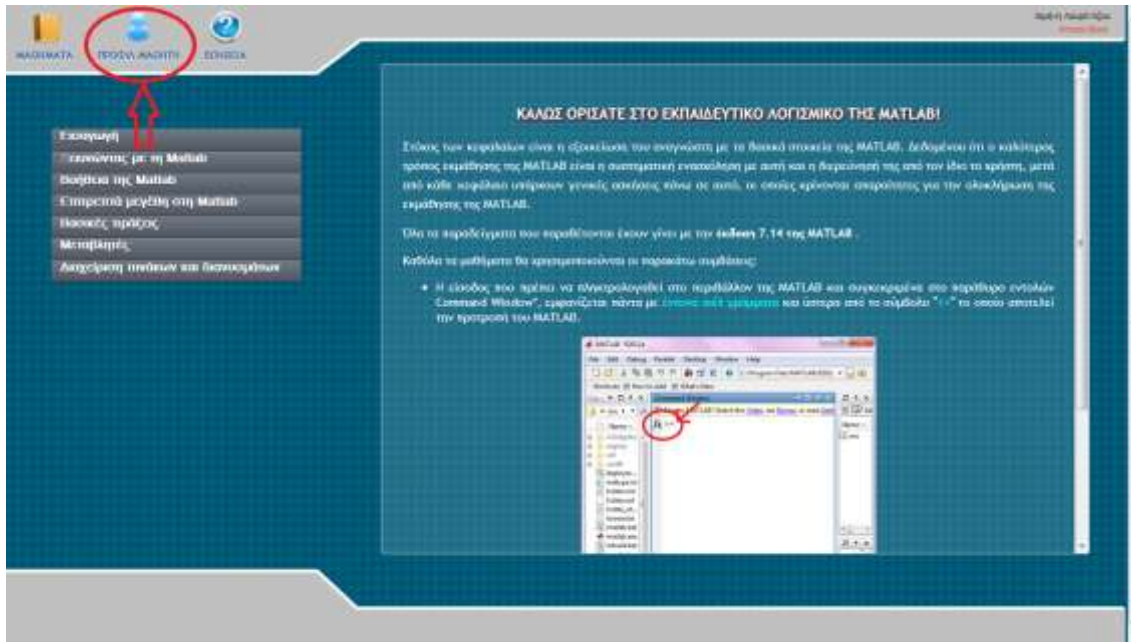
Ερώτηση: 2

Ποιο αποτέλεσμα εμφανίζεται ως μας απαντάμε η πληκτρολόγηση της εντολής `3*(2+0.25)^4` στο Περιβάριο εντολών του MATLAB;

0 2

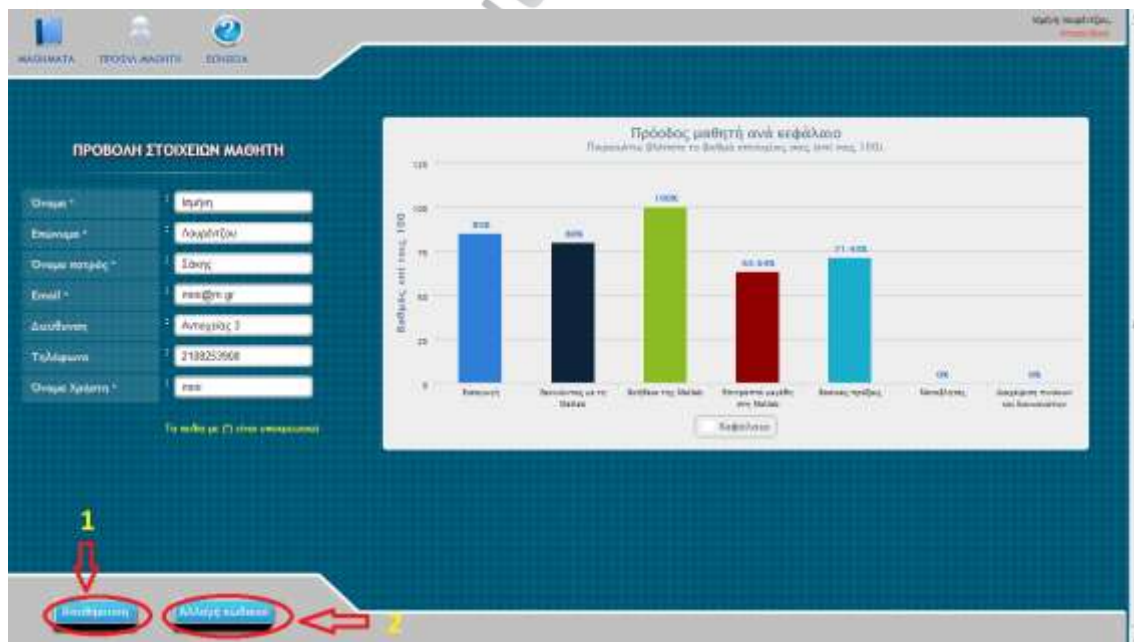
Εικόνα: 39 Προβολή αποτελεσμάτων διαγωνίσματος

Για την προβολή του προφίλ του ο εκπαιδευόμενος πατάει στο κουμπί **ΠΡΟΦΙΛ**.



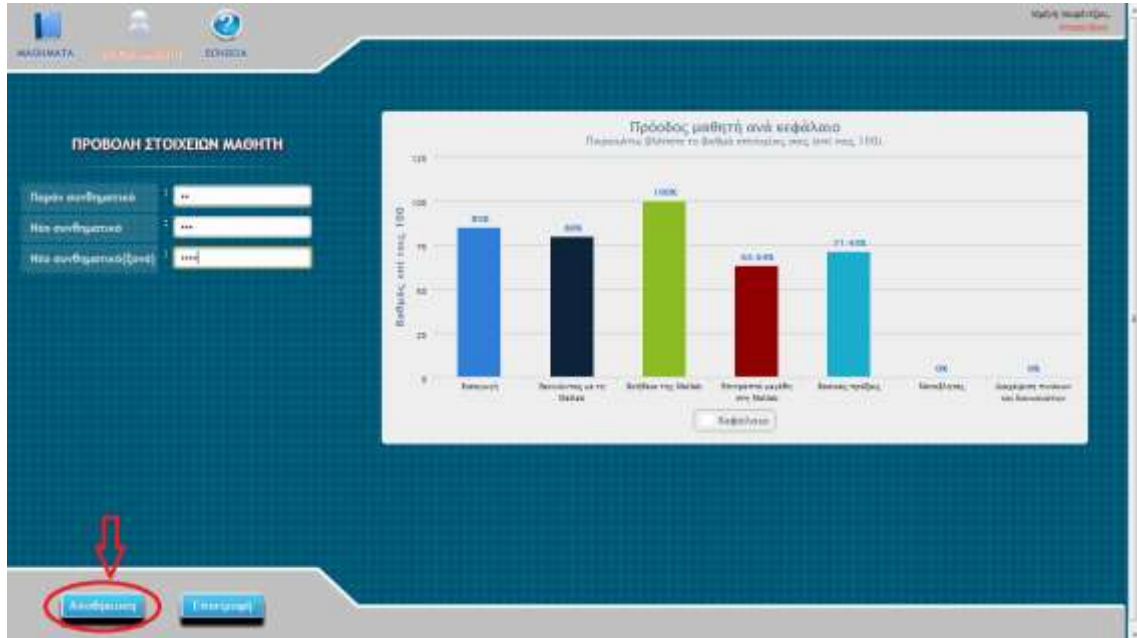
Εικόνα: 40 Προβολή προφίλ εκπαιδευόμενου

Σε αυτό το σημείο ο εκπαιδευόμενος έχει τη δυνατότητα να αλλάξει τα προσωπικά του στοιχεία και να πατήσει το κουμπί **Αποθήκευση** (βλέπε 1 στην παρακάτω εικόνα) ή να πατήσει απευθείας το κουμπί **Αλλαγή κωδικού** (βλέπε 2 στην παρακάτω εικόνα) προκειμένου να μεταβεί στη σελίδα αλλαγής προσωπικού κωδικού όπως φαίνεται στην παρακάτω Εικόνα:



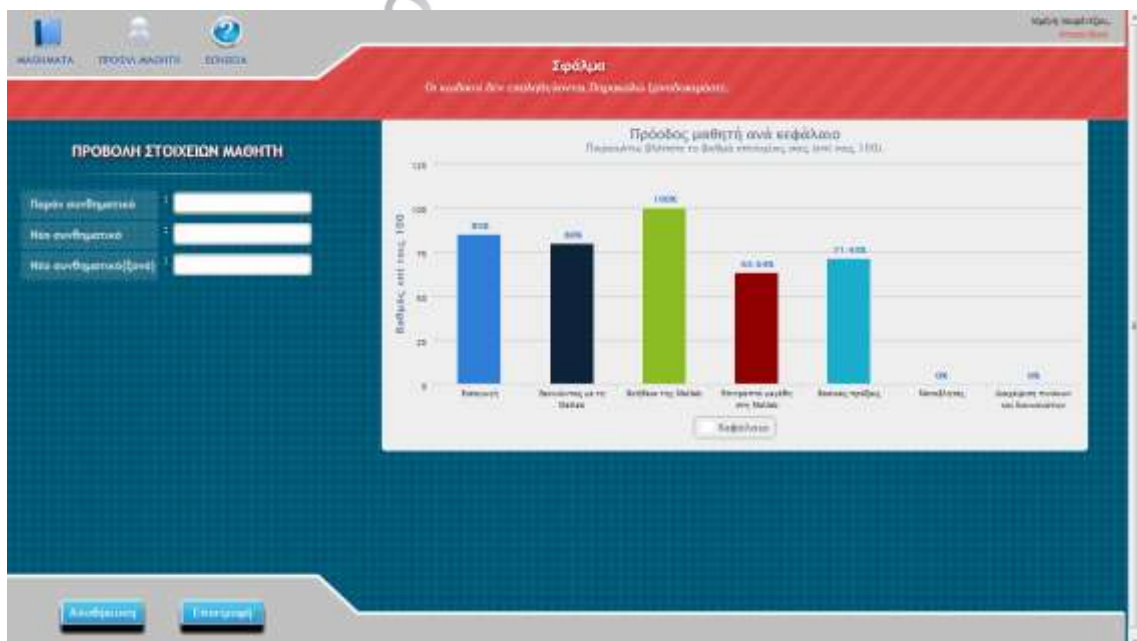
Εικόνα: 41 Αλλαγή στοιχείων προφίλ εκπαιδευόμενου

Πατώντας πάνω στο κουμπί **Αλλαγή κωδικού**, μεταφέρεται στη σελίδα που φαίνεται στην ακόλουθη εικόνα. Εδώ ο εκπαιδευόμενος πληκτρολογεί τον υπάρχοντα κωδικό του πρόσβασης και δύο φορές (για επαλήθευση) το νέο κωδικό που επιθυμεί. Έπειτα προκειμένου να ολοκληρωθεί η αλλαγή του κωδικού και να αποθηκευτεί ο νέος, πατάει το κουμπί **Αποθήκευση**.



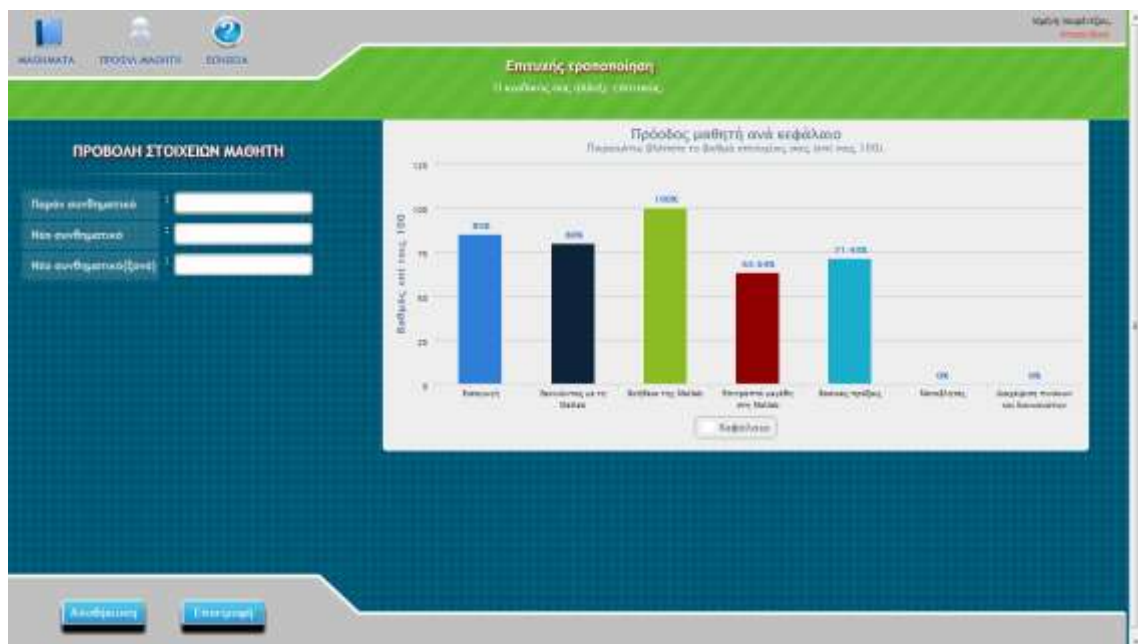
Εικόνα: 42 Αλλαγή προσωπικού κωδικού εκπαιδευόμενου

Στην περίπτωση όπου ο χρήστης δεν έχει εισάγει σωστά το κωδικό του ή τα πεδία όπου έχει πληκτρολογήσει το νέο κωδικό δεν επαληθεύονται, το λογισμικό του εμφανίζει το μήνυμα «Οι κωδικοί δεν επαληθεύονται. Παρακαλώ ξαναπροσπαθήστε», όπως φαίνεται στην ακόλουθη εικόνα:



Εικόνα: 43 Μήνυμα λάθος εισαγωγής κωδικού εκπαιδευόμενου

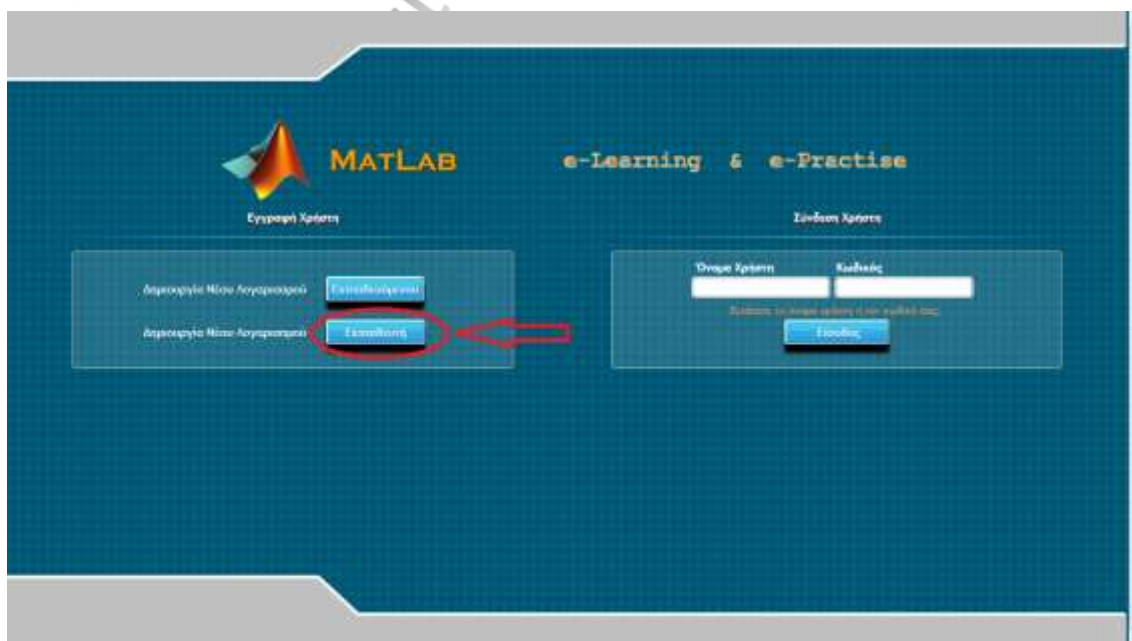
Αντίθετα αν ο χρήστης έχει εισάγει όλους τους κωδικούς επιτυχώς τότε το λογισμικό του εμφανίζει το μήνυμα «Ο κωδικός σας άλλαξε επιτυχώς», όπως φαίνεται στην **Εικόνα 33**.



Εικόνα: 44 Μήνυμα επιτυχής αλλαγή κωδικού πρόσβασης εκπαιδευόμενου

Σενάριο λειτουργίας με ρόλο εκπαιδευτή

Προκειμένου ένας νέος εκπαιδευτής, να δημιουργήσει ένα λογαριασμό εισόδου στο σύστημα είναι απαραίτητο να επιλέξει τη μετάβασή του στην αντίστοιχη σελίδα εγγραφής μέσω του κουμπιού που φαίνεται στην παρακάτω εικόνα:



Εικόνα: 45 Κουμπί μετάβασης στη σελίδα εγγραφής εκπαιδευτή

Όταν ο εκπαιδευτής πατήσει το κουμπί εγγραφής, που αναφέρθηκε παραπάνω, θα μεταβεί στην εξής σελίδα:

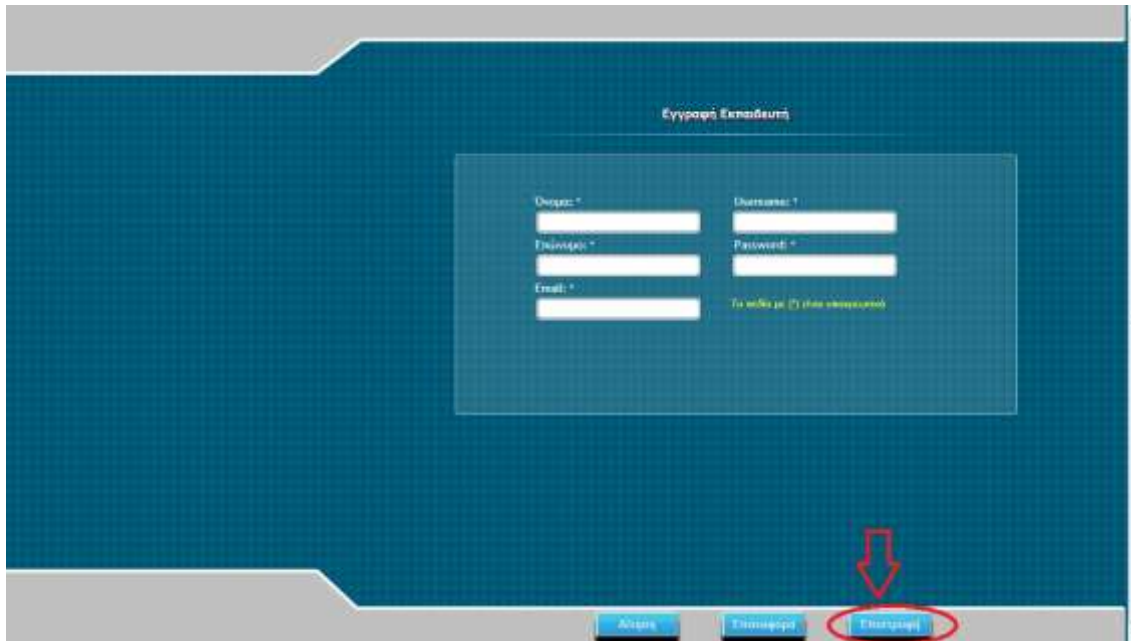
Εικόνα: 46 Σελίδα εγγραφής εκπαιδευτή

Σε αυτό το σημείο ο εκπαιδευτής θα πρέπει να εισάγει το όνομα, το επίθετο, το email του, το όνομα χρήστη και τον κωδικό που θα χρησιμοποιεί κατά τη σύνδεσή του στο λογισμικό.

Στην περίπτωση όπου έχει κάνει κάποιο λάθος κατά την πληκτρολόγηση των στοιχείων του και επομένως επιθυμεί να σβήσει όλα τα πεδία που συμπλήρωσε, μπορεί να πατήσει το κουμπί **Επαναφορά**. Μέσω αυτού του κουμπιού αυτού διαγράφονται αυτόματα όλες οι τιμές των πεδίων.

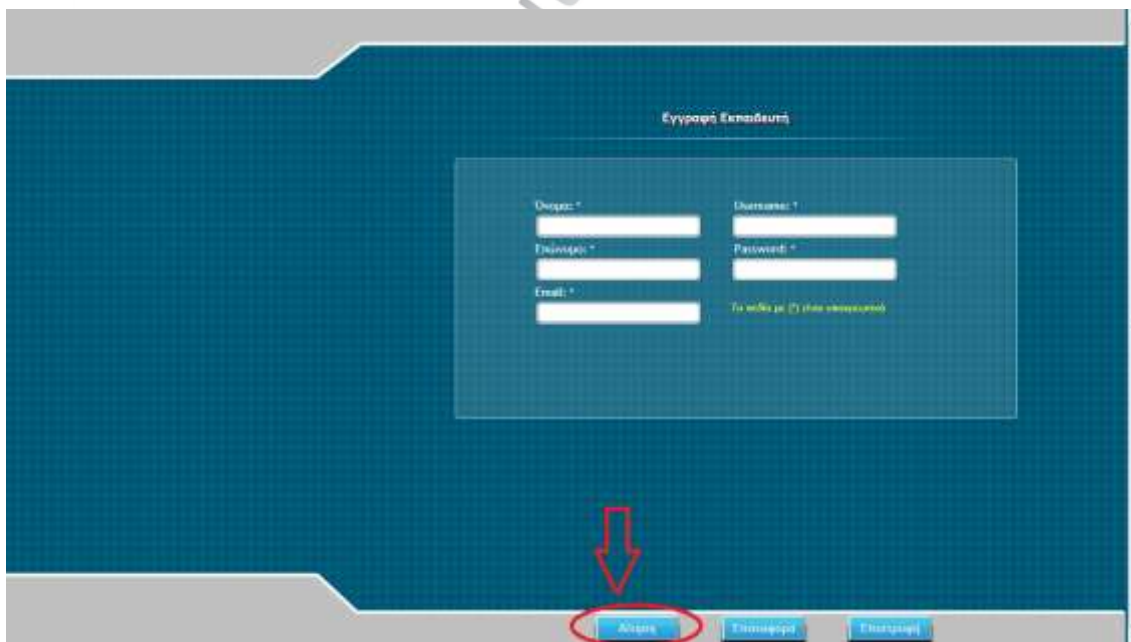
Εικόνα: 47 Κουμπί επαναφοράς πεδίων

Αν ο εκπαιδευτής επιθυμεί να επιστρέψει στην αρχική σελίδα χωρίς να εγγραφεί, τότε μπορεί να πατήσει το κουμπί **Επιστροφή**, όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα: 48 Κουμπί επιστροφής στην αρχική σελίδα

Αλλιώς, αν ο εκπαιδευτής έχει εισάγει όλα τα απαραίτητα στοιχεία και επιθυμεί να ολοκληρώσει την εγγραφή του, μπορεί να πατήσει το αντίστοιχο κουμπί **Αίτηση**.



Εικόνα: 49 Κουμπί ολοκλήρωσης εγγραφής

Ωστόσο, στην περίπτωση όπου έχει εισάγει λάθος στοιχεία, όπως για παράδειγμα τη διεύθυνση ηλεκτρονικού ταχυδρομείου του, η εφαρμογή τον ενημερώνει, εμφανίζοντάς του το αντίστοιχο μήνυμα λάθους, όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα: 50 Εμφάνιση μηνύματος για λάθος email εκπαιδευτή

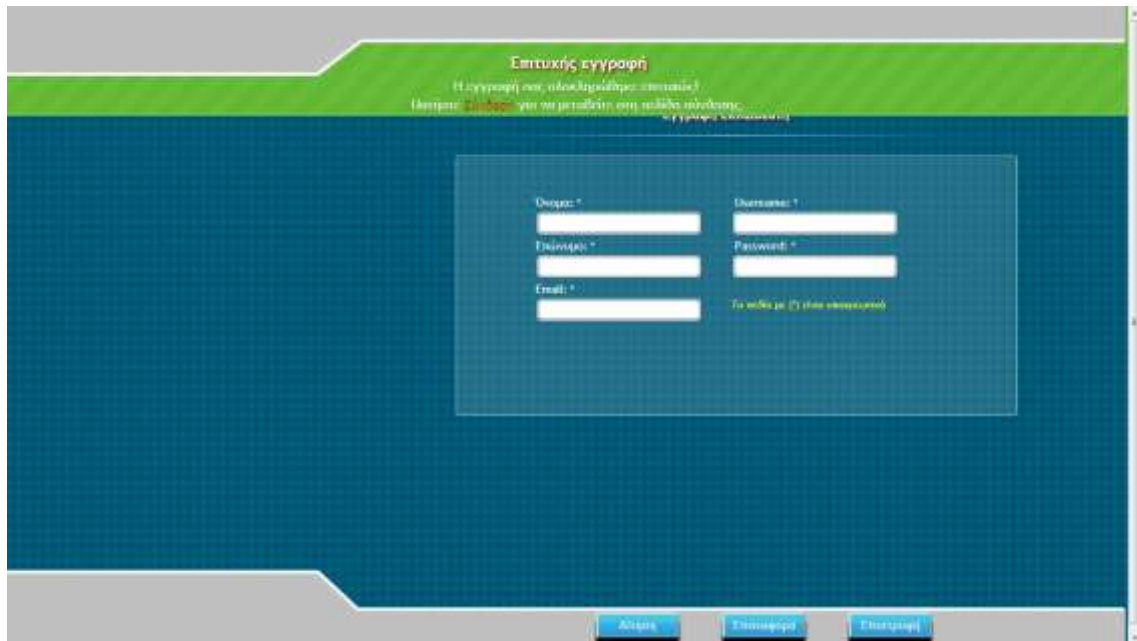
Στην περίπτωση όπου ο εκπαιδευτής έχει ήδη εγγραφεί, χρησιμοποιώντας το ίδιο επίθετο και την ίδια διεύθυνση ηλεκτρονικού ταχυδρομείου το λογισμικό εμφανίζει στο χρήστη το μήνυμα λάθους που φαίνεται στην ακόλουθη εικόνα:



Εικόνα: 51 Εμφάνιση μηνύματος για λάθος στοιχεία εκπαιδευτή

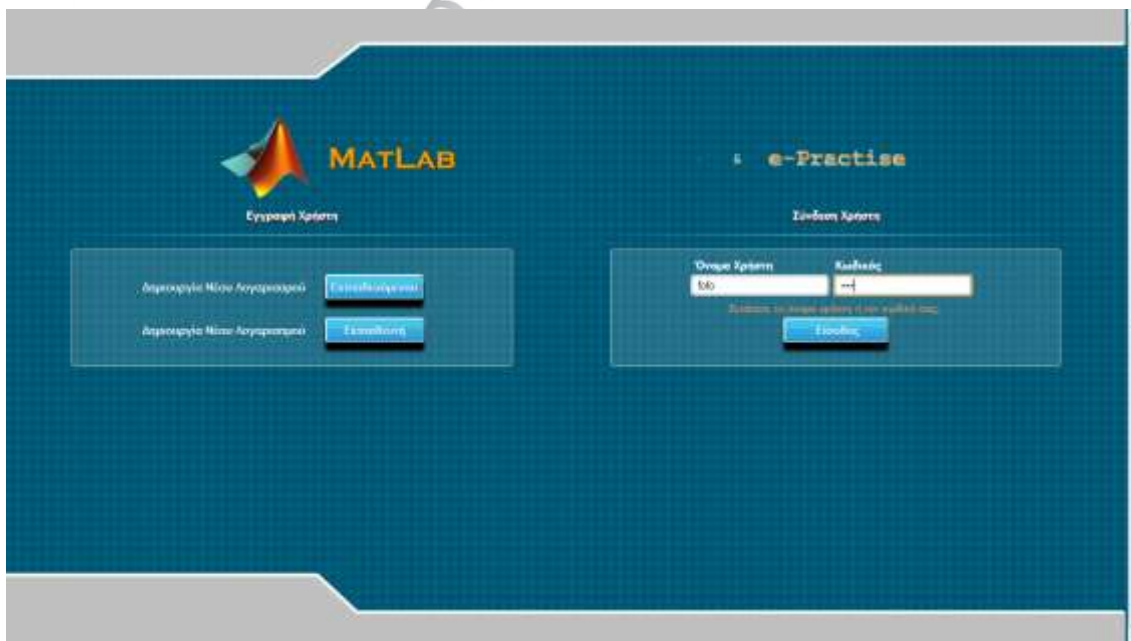
Αν αφότου ο εκπαιδευτής εισάγει τα στοιχεία του και πατήσει το κουμπί εγγραφής, δε του εμφανιστεί κανένα μήνυμα λάθους, τότε η εφαρμογή θα του εμφανίσει ένα μήνυμα επιτυχούς εγγραφής όπως φαίνεται στην παρακάτω εικόνα. Πατώντας το κουμπί **Σύνδεση** το λογισμικό θα τον μεταφέρει στη σελίδα σύνδεσης λογαριασμού, ενώ πατώντας πάνω στο μήνυμα, ο χρήστης

θα παραμείνει στη σελίδα εγγραφής νέου χρήστη, για να εγγράψει πιθανότατα κάποιον άλλον εκπαιδευτή. Ωστόσο, και στις δύο περιπτώσεις η εγγραφή του εκπαιδευτή έχει ολοκληρωθεί επιτυχώς.



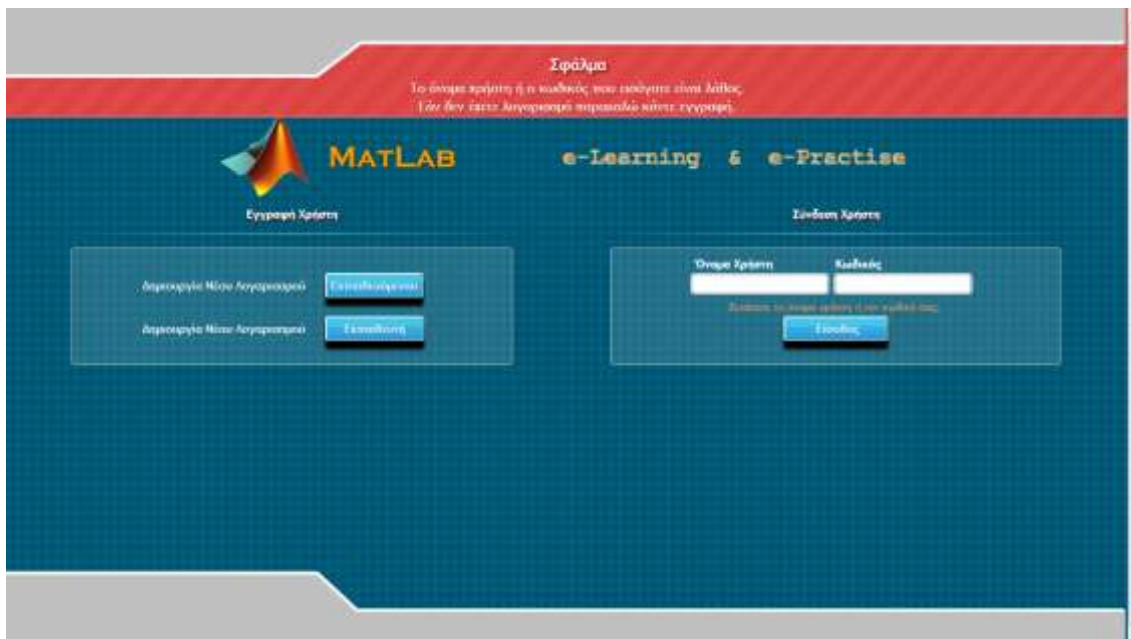
Εικόνα: 52 Εμφάνιση μηνύματος επιτυχής εγγραφής εκπαιδευτή

Μετά την επιτυχή εγγραφή του εκπαιδευτή, αν επιλέξει τη μετάβαση του μέσω του κουμπιού **Σύνδεση**, το λογισμικό τον στέλνει στη σελίδα σύνδεσης όπου μπορεί να εισάγει το όνομα χρήστη και τον κωδικό που δήλωσε κατά την εγγραφή του στα αντίστοιχα πεδία, όπως φαίνεται στην ακόλουθη εικόνα.



Εικόνα: 53 Σελίδα σύνδεσης εκπαιδευτή

Αν ο εκπαιδευτής εισάγει λάθος τα στοιχεία του τότε το λογισμικό του εμφανίζει το ακόλουθο μήνυμα λάθους:



Εικόνα: 54 Σελίδα σύνδεσης χρήστη

Αν τα στοιχεία σύνδεσης του χρήστη είναι σωστά, τότε μεταφέρεται στην κεντρική σελίδα εκπαιδευτή όπως φαίνεται στην παρακάτω εικόνα. Όπως παρατηρούμε ο εκπαιδευτής έχει τέσσερις κεντρικές επιλογές στο πάνω μέρος της σελίδας όπου και εμφανίζεται το μενού. Η περιοχή εμφάνισης του μενού παραμένει σταθερή καθ' όλη την περιήγηση του στο λογισμικό.



Εικόνα: 55 Κεντρική οθόνη εκπαιδευτή

Το λογισμικό εμφανίζει στον εκπαιδευτή πάντα όλα τα κεφάλαια και υποκεφάλαια, σύμφωνα με τη σειρά προτεραιότητας που τα είχε καταχωρήσει, στα αριστερά της οθόνης. Έτσι, για την προβολή της θεωρίας ενός συγκεκριμένου κεφαλαίου πρέπει να επιλέξει το κεφάλαιο ή υποκεφάλαιο που τον ενδιαφέρει. Χαρακτηριστικό σε αυτή τη σελίδα είναι ότι η προβολή των ενότητων της θεωρίας γίνεται με τον ίδιο ακριβώς τρόπο τόσο στον εκπαιδευτή όσο και στον εκπαιδευόμενο.



Εικόνα: 56 Κύριο μενού προβολής θεωρίας



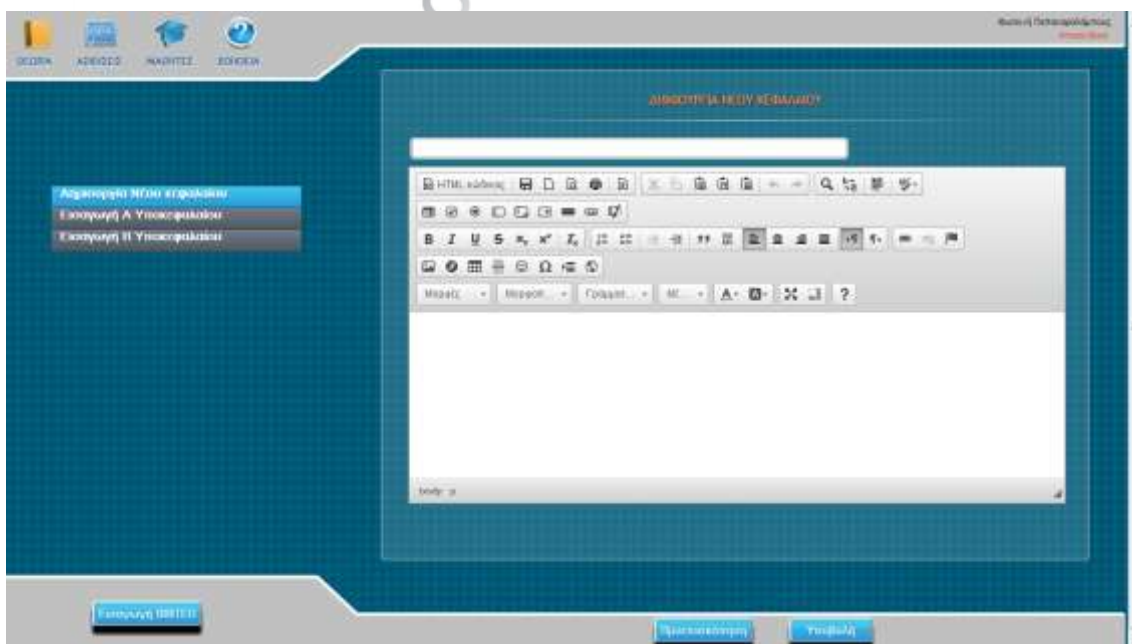
Εικόνα: 57 Προβολή θεωρίας κεφαλαίου “Επιτρεπτά μεγέθη στη Matlab”

Για την εισαγωγή νέας θεωρίας επιλέγει από το κεντρικό μενού τα **ΜΑΘΗΜΑΤΑ** και στη συνέχεια πατάει πάνω στο κουμπί **ΕΙΣΑΓΩΓΗ** που εμφανίζεται στο κάτω αριστερό μέρος της οθόνης.



Εικόνα: 58 Εισαγωγή νέας θεωρίας

Αμέσως το λογισμικό του εμφανίζει την οθόνη δημιουργίας νέου κεφαλαίου, όπως φαίνεται στην παρακάτω εικόνα. Από αυτή τη σελίδα, ο εκπαιδευτής μπορεί να επιλέξει μέσα από τρεις κεντρικές επιλογές του υπομενού. Την εισαγωγή νέου κεφαλαίου, την εισαγωγή υποκεφαλαίου A επιπέδου σε υπάρχον κεφάλαιο και την εισαγωγή νέου υποκεφαλαίου B επιπέδου σε υπάρχον υποκεφάλαιο A επιπέδου.



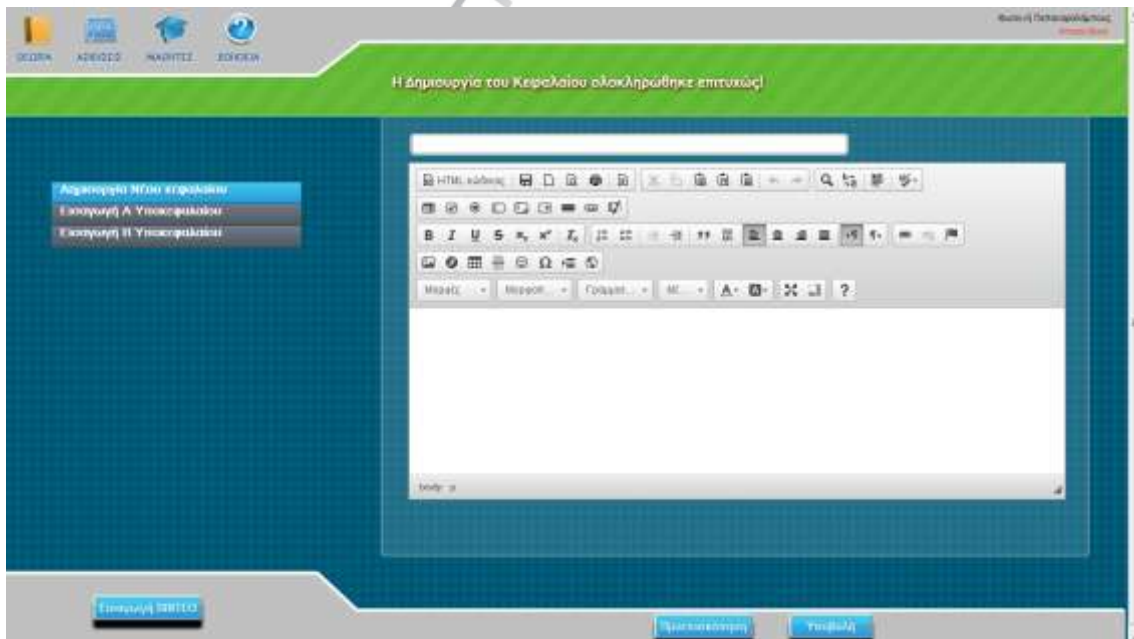
Εικόνα: 59 Εισαγωγή νέου κεφαλαίου

Εισάγοντας τον τίτλο του κεφαλαίου και την αντίστοιχη θεωρία του, ο εκπαιδευτής έχει δύο κύριες επιλογές, την προεπισκόπηση του κεφαλαίου (Εικόνα 3.7.52) που πρόκειται να αποθηκεύσει ή την απευθείας αποθήκευσή του, μέσω του κουμπιού **Υποβολή**.



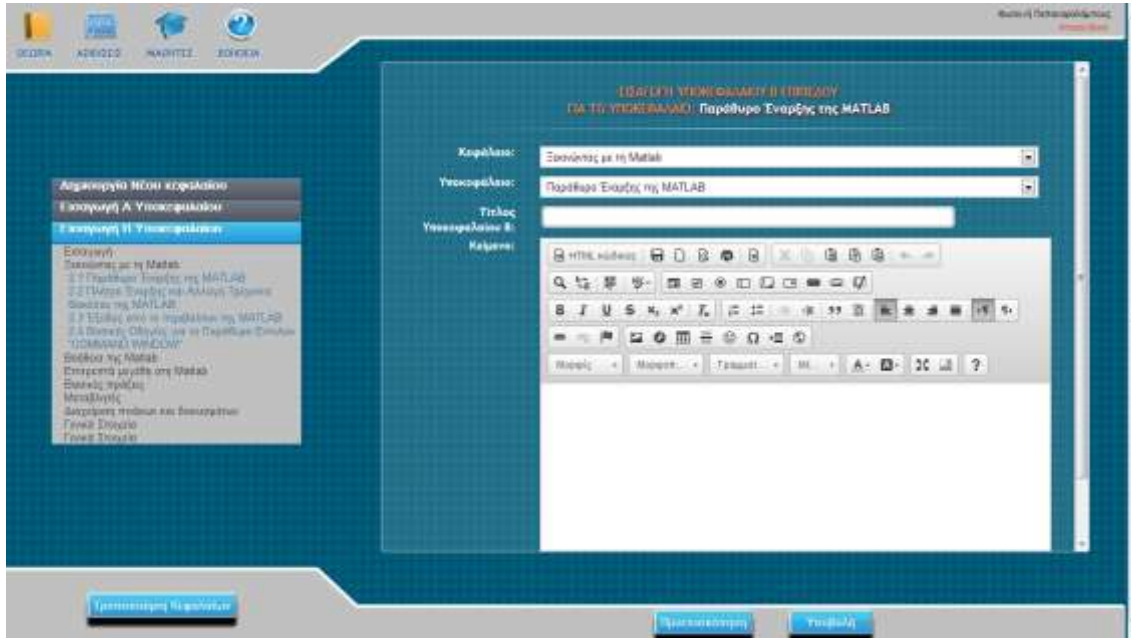
Εικόνα: 60 Προβολή εισαγωγής νέου κεφαλαίου

Στην περίπτωση όπου επιλέξει την απευθείας αποθήκευσή του, το λογισμικό τον ενημερώνει για την επιτυχή αποθήκευση του κεφαλαίου.



Εικόνα: 61 Επιτυχής εισαγωγή κεφαλαίου

Προκειμένου να δημιουργήσει ένα νέο υποκεφάλαιο Β επιπέδου ο εκπαιδευτής πρέπει να επιλέξει το κεφάλαιο και στη συνέχεια το υποκεφάλαιο Α επιπέδου στο οποίο θα καταχωρηθεί το νέο υποκεφάλαιο. Αφού τα επιλέξει του εμφανίζεται η ακόλουθη οθόνη:



Εικόνα: 62 Εισαγωγή υποκεφαλαίου Β επιπέδου

Για την τροποποίηση της θεωρίας ο εκπαιδευτής επιλέγει από το κεντρικό μενού τα **ΜΑΘΗΜΑΤΑ** και στη συνέχεια πατάει πάνω στο κουμπί **ΤΡΟΠΟΠΟΙΗΣΗ** που εμφανίζεται στο κάτω αριστερό μέρος της οθόνης.



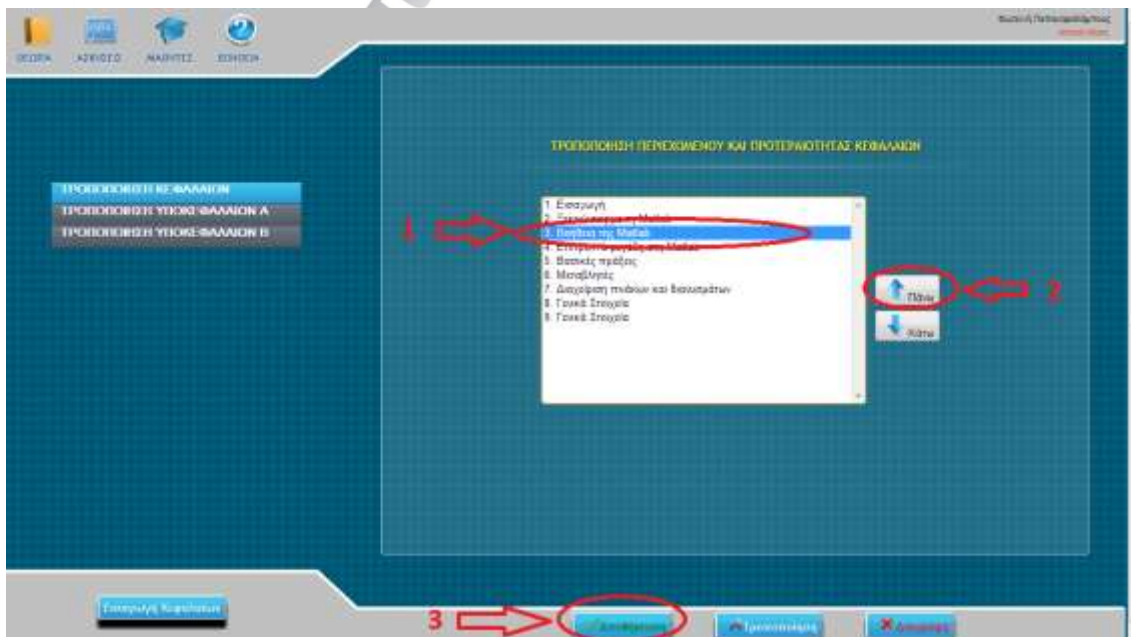
Εικόνα: 63 Τροποποίηση υπάρχουσας θεωρίας

Η εφαρμογή αμέσως εμφανίζει στον εκπαιδευτή την ακόλουθη σελίδα από όπου μπορεί να αλλάξει τη σειρά προτεραιότητας των κεφαλαίων, να επιλέξει ένα κεφάλαιο για να τροποποιήσει το περιεχόμενό του, να διαγράψει ένα κεφάλαιο ή να μεταβεί στην τροποποίηση των υποκεφαλαίων Α και Β επιπέδου, μέσω του υπομενού στα αριστερά της οθόνης.



Εικόνα: 64 Τροποποίηση υπάρχοντος κεφαλαίου

Για την αλλαγή προτεραιότητας ο εκπαιδευτής πρέπει πρώτα να επιλέξει τον τίτλο του κεφαλαίου που τον ενδιαφέρει και στη συνέχεια να το μεταφέρει, μετακινώντας τη θέση του κεφαλαίου με τη χρήση των κουμπιών **Πάνω** ή **Κάτω**. Στην περίπτωση όπου επιθυμεί να μετακινήσει το τρίτο κεφάλαιο στην κορυφή της λίστας, ορίζοντάς το έτσι πρώτο σε προτεραιότητα, είναι απαραίτητη η επιλογή του (Εικόνα 3.7.65) και στη συνέχεια το πάτημα του κουμπιού **Πάνω** για δύο συνεχόμενες φορές. Τελευταία κίνηση θα είναι το πάτημα του κουμπιού **Αποθήκευση** έτσι ώστε να αποθηκεύσει το λογισμικό τη λίστα των κεφαλαίων με τη νέα σειρά προτεραιότητας.



Εικόνα: 65 Τροποποίηση προτεραιότητας κεφαλαίων

Για την τροποποίηση του περιεχομένου ενός κεφαλαίου είναι απαραίτητη η επιλογή του τίτλου του και στη συνέχεια του πάτημα του κουμπιού **Τροποποίηση**, αλλιώς το λογισμικό εμφανίζει το μήνυμα λάθους «Παρακαλώ επιλέξτε κάποιο κεφάλαιο», όπως φαίνεται στην ακόλουθη εικόνα:



Εικόνα: 66 Μήνυμα λάθους επιλογής τροποποίησης κεφαλαίου

Στην περίπτωση όπου ο εκπαιδευτής επιλέξει το κεφάλαιο και πατήσει το κουμπί **Τροποποίηση**, του εμφανίζεται η ακόλουθη οθόνη, όπου μπορεί να βλέπει τον τίτλο του κεφαλαίου και το περιεχόμενό του, αν έχει οριστεί, όπως ακριβώς και στην προβολή του κεφαλαίου. Στο παράδειγμα που ακολουθεί, το κεφάλαιο «**ΞΕΚΙΝΩΝΤΑΣ ΜΕ ΤΗ MATLAB**» δεν έχει κάποιο κείμενο θεωρίας που να το συνοδεύει και γι' αυτό εμφανίζεται η πρόταση «**Δεν έχει οριστεί κείμενο**» στο σημείο όπου θα έπρεπε να εμφανιστεί το περιεχόμενο του κεφαλαίου.



Εικόνα: 67 Τροποποίηση περιεχομένου κεφαλαίου

Για την προβολή των ασκήσεων που έχει καταχωρήσει, επιλέγει **ΑΣΚΗΣΕΙΣ** από το κύριο μενού.



Εικόνα: 68 Προβολή καταχωρημένων ασκήσεων

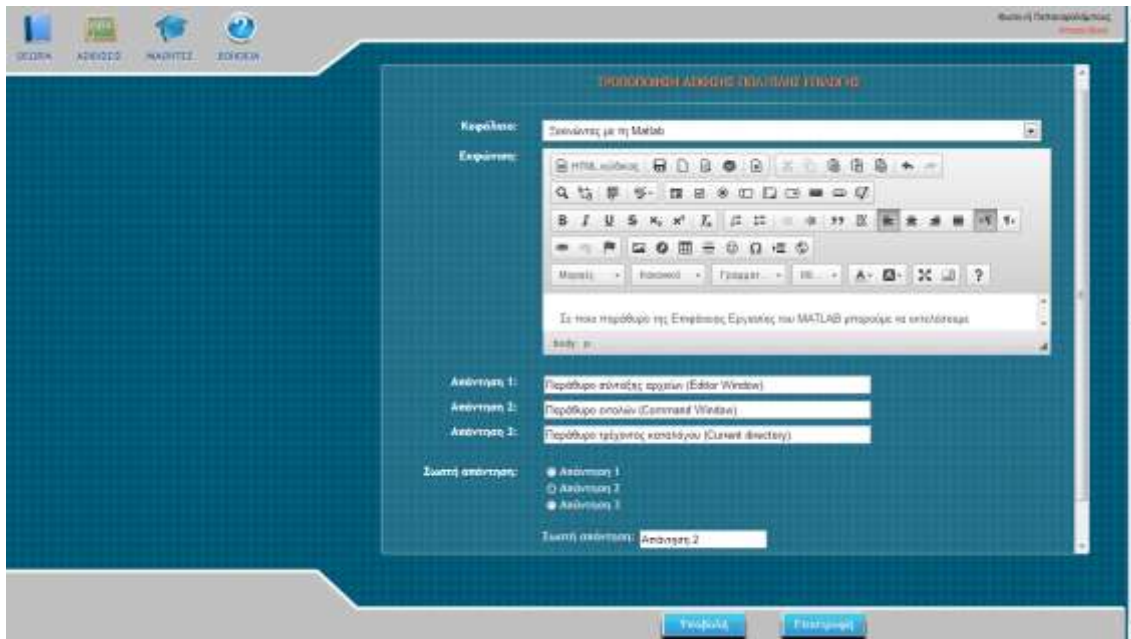
Σε αυτό το σημείο, ο εκπαιδευτής επιλέγει το κεφάλαιο του οποίου επιθυμεί να προβάλει τις ασκήσεις και την κατηγορία των ασκήσεων. Το λογισμικό, θεωρεί ως προεπιλογή του εκπαιδευτή, την προβολή όλων των κατηγοριών των ασκήσεων ενός κεφαλαίου.



Εικόνα: 69 Επιλογή προβολής ασκήσεων ενός κεφαλαίου

Δίπλα από την εκφώνηση κάθε άσκησης, δίνεται η δυνατότητα στον εκπαιδευτή να επιλέξει την τροποποίηση της συγκεκριμένης άσκησης μέσα από το σύνδεσμο **Τροποποίηση**.

Επιλέγοντάς τον, ο χρήστης μεταφέρεται στην οθόνη που βλέπουμε παρακάτω:



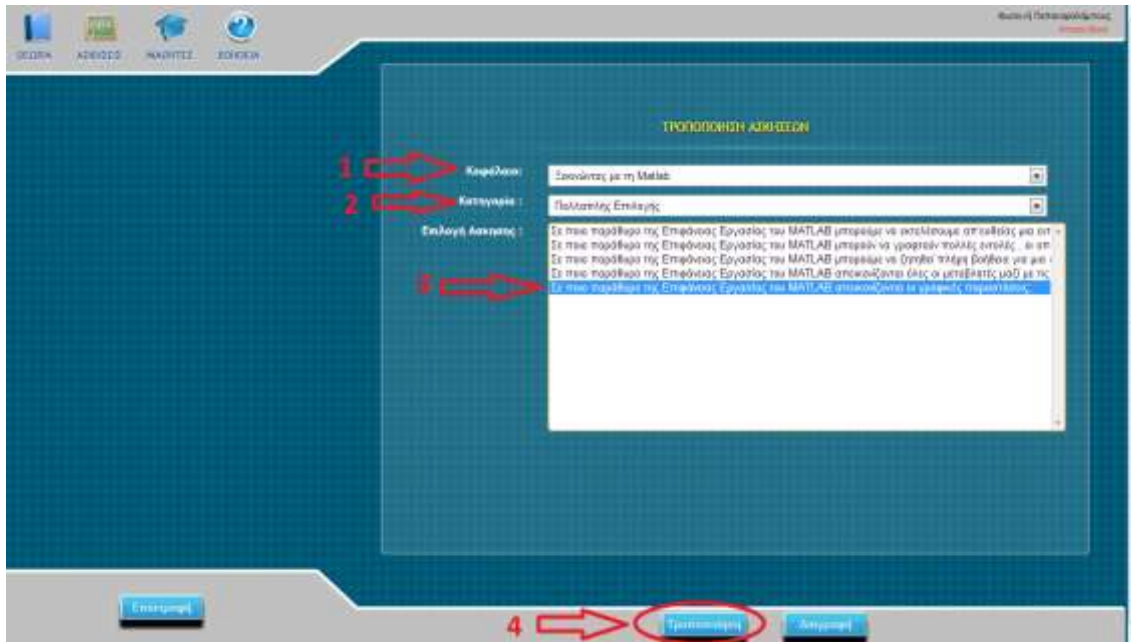
Εικόνα: 70 Τροποποίηση άσκησης πολλαπλής επιλογής

Ωστόσο, στην τροποποίηση ασκήσεων ο εκπαιδευτής μπορεί να μεταβεί και μέσω του κουμπιού Τροποποίηση.



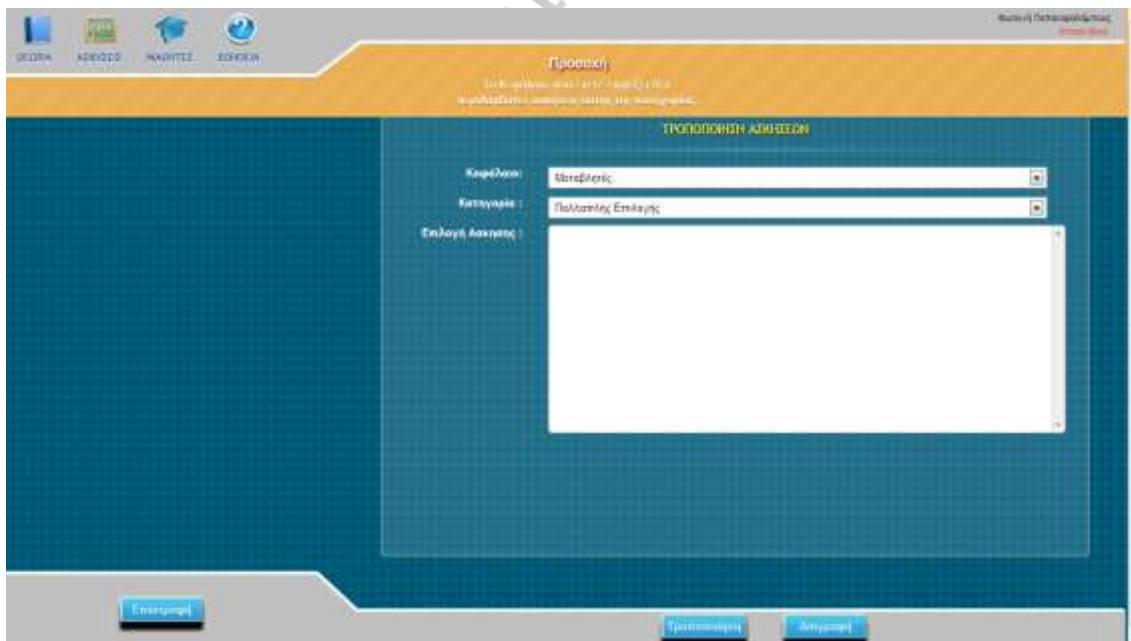
Εικόνα: 71 Επιλογή μετάβασης στην τροποποίηση ασκήσεων

Εδώ, ο εκπαιδευτής καλείται να επιλέξει το κεφάλαιο στο οποίο κατατάσσεται η άσκηση, το είδος της άσκησης και την ερώτηση της άσκησης. Στη συνέχεια αφού έχει επιλέξει την άσκηση πατάει το κουμπί Τροποποίηση.



Εικόνα: 72 Βήματα τροποποίηση άσκησης

Στην περίπτωση όπου επιλέξει ένα κεφάλαιο το οποίο δε περιλαμβάνει ασκήσεις της επιλεγμένης κατηγορίας, το λογισμικό τον ενημερώνει εμφανίζοντάς του το μήνυμα που φαίνεται στην ακόλουθη εικόνα:



Εικόνα: 73 Προβολή μηνύματος λάθους- τροποποίηση ασκήσεων

Για την εισαγωγή νέας άσκησης ο εκπαιδευτής επιλέγει Ασκήσεις από το κεντρικό μενού και στη συνέχεια το κουμπί Εισαγωγή που εμφανίζεται κάτω αριστερά:

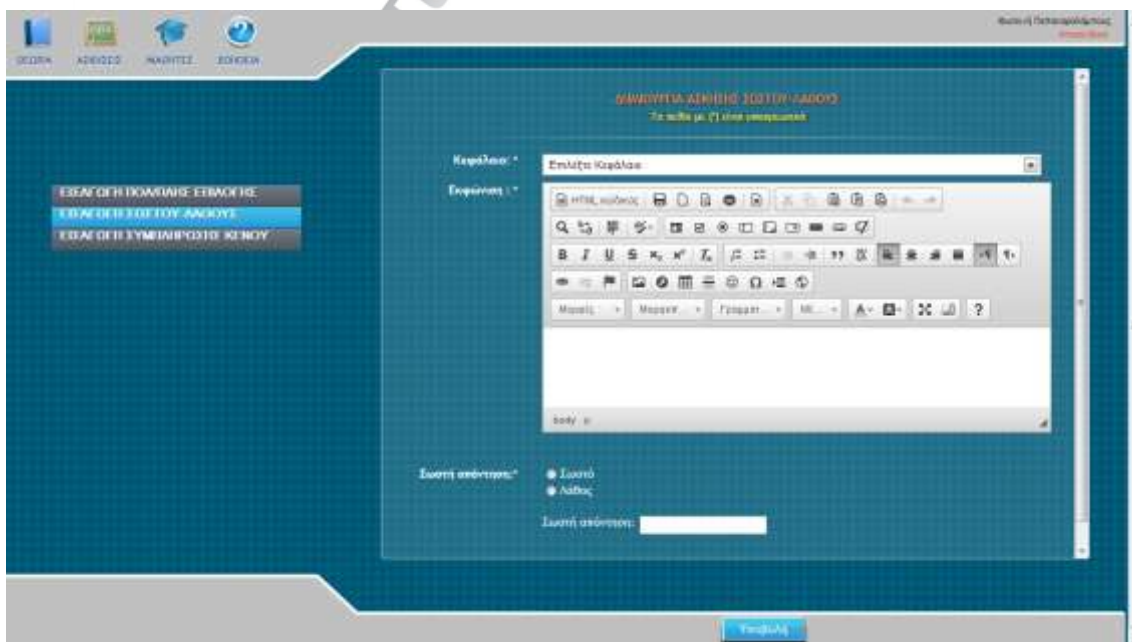


Εικόνα: 74 Εισαγωγή νέας άσκησης

Το λογισμικό του εμφανίζει τη σελίδα που παρουσιάζεται στην ακόλουθη εικόνα, από όπου μπορεί να επιλέξει την κατηγορία της άσκησης μέσω του υπομενού:

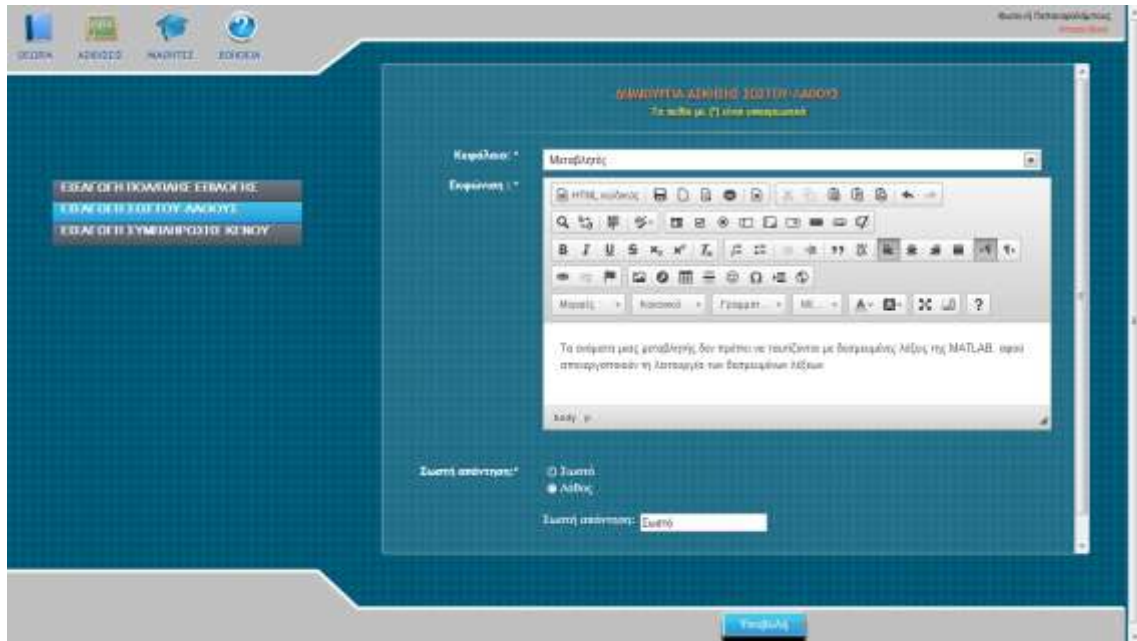
1. ΕΙΣΑΓΩΓΗ ΠΟΛΥΠΛΗΣ ΕΠΙΛΟΓΗΣ
2. ΕΙΣΑΓΩΓΗ ΣΩΣΤΟΥ-ΛΑΘΟΥΣ
3. ΕΙΣΑΓΩΓΗ ΣΥΜΠΛΗΡΩΣΗΣ ΚΕΝΟΥ.

Επιλέγοντας την εισαγωγή άσκησης σωστού-λάθους, το λογισμικό προβάλλει στον εκπαιδευόμενο την ακόλουθη οθόνη:



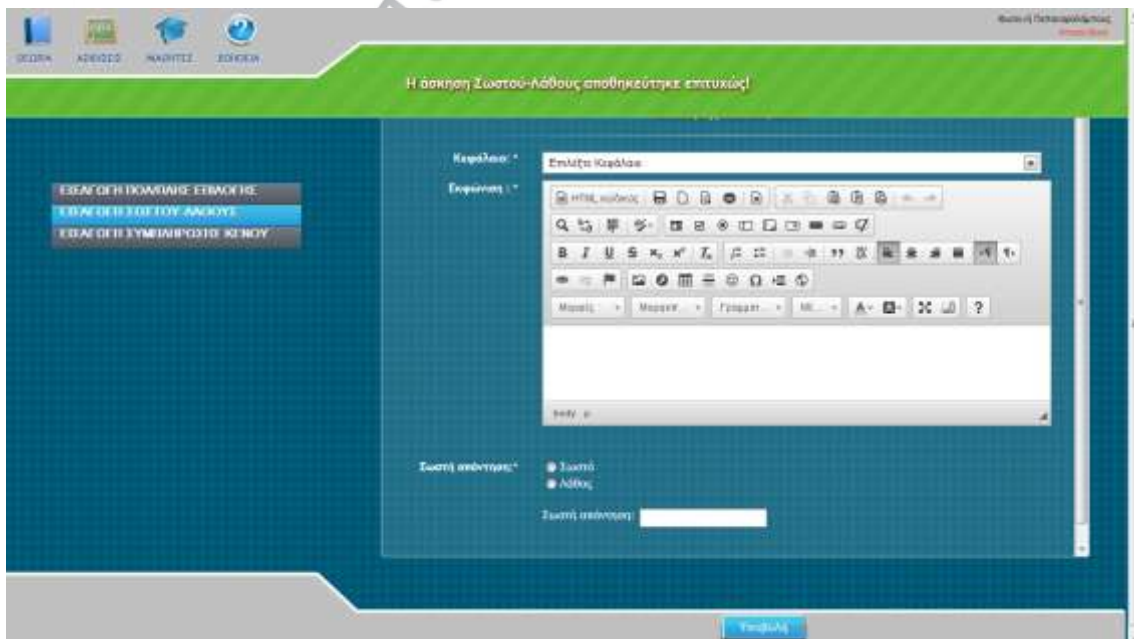
Εικόνα: 75 Εισαγωγή νέας άσκησης σωστού-λάθους

Εδώ ο εκπαιδευτής μπορεί να επιλέξει το κεφάλαιο στο οποίο επιθυμεί να εισάγει την άσκηση και στη συνέχεια να εισάγει την εκφώνηση αλλά και τις απαντήσεις που θα εμφανίζονται στον εκπαιδευόμενο. Τελευταίο βήμα αποτελεί η επιλογή της σωστής απάντησης ώστε να την αποθηκεύσει το λογισμικό προκειμένου να είναι ικανό να διορθώσει τον εκπαιδευόμενο κατά τη διάρκεια ενός διαγωνίσματος.



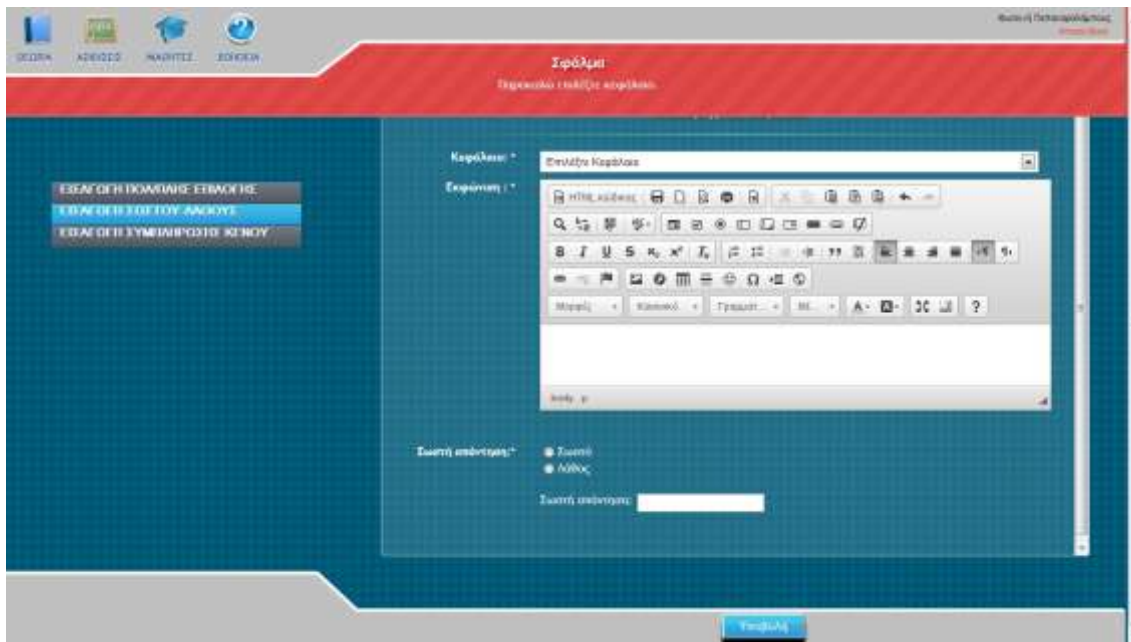
Εικόνα: 76 Εισαγωγή στοιχείων άσκησης σωστού-λάθους

Αφού έχει ολοκληρώσει την εισαγωγή των στοιχείων της άσκησης, ο εκπαιδευτής πατάει το κουμπί **Υποβολή** και το λογισμικό, εφόσον έχει αποθηκεύσει επιτυχώς την άσκηση, του εμφανίζει και το ανάλογο μήνυμα όπως φαίνεται στην παρακάτω εικόνα:



Εικόνα: 77 Επιτυχής εισαγωγή νέας άσκησης

Ωστόσο, όπως είναι γνωστό και από προηγούμενες αναφορές, στην περίπτωση όπου ο χρήστης δεν έχει συμπληρώσει όλα τα απαραίτητα πεδία το λογισμικό του εμφανίζει το αντίστοιχο μήνυμα λάθους:



Εικόνα: 78 Μήνυμα λάθους εισαγωγής νέας άσκησης

Για την προβολή των στοιχείων των εκπαιδευομένων του, ο εκπαιδευτής επιλέγει **ΜΑΘΗΤΕΣ** από το κεντρικό μενού.



Εικόνα: 79 Προβολή στοιχείων μαθητών του εκπαιδευτή

[5] ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Στις μέρες μας, ο παραδοσιακός τρόπος διδασκαλίας αδυνατεί να καλύψει πλήρως τις ανάγκες της σύγχρονης κοινωνίας. Είναι λοιπόν αναπόφευκτη η ανάγκη δημιουργίας εφαρμογών, οι οποίες θα αναβαθμίσουν το επίπεδο της εκπαίδευσης και θα παρέχουν μία σειρά πλεονεκτημάτων κατά τη χρήση τους. Δεδομένου ότι από τις σημαντικότερες εφαρμογές στον τομέα της εκπαίδευσης είναι τα εκπαιδευτικά λογισμικά, στην παρούσα εργασία παρουσιάστηκε ο σχεδιασμός και η υλοποίηση ενός εκπαιδευτικού λογισμικού για τη διδασκαλία των βασικών στοιχείων της MATLAB. Αν και το συγκεκριμένο λογισμικό αποτελεί ένα ολοκληρωμένο πακέτο για τη διδασκαλία της MATLAB, μπορεί να χρησιμοποιηθεί και για την εκμάθηση οποιουδήποτε άλλου μαθήματος, είτε σχετίζεται άμεσα με τη Πληροφορική είτε όχι, με την κατάλληλη παραμετροποίηση της βάσης δεδομένων.

Οι δώδεκα ενότητες θεωρίας και ασκήσεων επιλέχτηκαν έτσι ώστε να καλύπτεται ένα εύλογο μέρος των εισαγωγικών κεφαλαίων της MATLAB. Κάθε ενότητα είναι εμπλουτισμένη με οπτικοακουστικό υλικό και είναι δομημένη με τέτοιο τρόπο, ώστε να αποτελεί τη βάση για τις επόμενες. Ο βαθμός δυσκολίας των ενότητων μεταβάλλεται βαθμιαία προκειμένου ο εκπαιδευόμενος να μπορεί να εστιάσει στην θεωρία της κάθε ενότητας και να μη δυσανασχετεί κατά την προσπάθεια κατανόησής τους. Ο αριθμός των ερωτήσεων κάθε κεφαλαίου είναι ο μικρότερος δυνατός έτσι ώστε να εξετάζεται η απαραίτητη ύλη, αλλά και να μη κουράζει τον εκπαιδευόμενο, κάνοντάς τον να χάσει το ενδιαφέρον του. Η θεωρία παρουσιάζεται με ευχάριστο τρόπο μέσα από ένα εύκολο στη χρήση περιβάλλον διεπαφής, το οποίο είναι εμπλουτισμένο με χρώματα στο ύψος του λογότυπου της MATLAB, εικόνων και οπτικοακουστικού υλικού για τη συμπύκνωση της πληροφορίας.

Μία ενδεχόμενη βελτίωση όπου θα μπορούσε να λάβει το συγκεκριμένο λογισμικό είναι ο εμπλουτισμός του με συνεργατικά πρότυπα, καθώς το μάθημα γίνεται περισσότερο επικοινωνιακό και ενδιαφέρον για τους εκπαιδευόμενους. Η δημιουργία ενός διαδικτυακού τύπου συνάντησης ιδεών και ανθρώπων, όπου θα τους δίνεται η δυνατότητα να ανταλλάσσουν μεταξύ τους γνώσεις και πιθανότατα να βοηθούν ο ένας τον άλλον, θα ήταν ένα ιδιαίτερα σημαντική. Ωστόσο, αυτό προϋποθέτει τη δημιουργία ενός αυτόματου ελεγκτικού μηχανισμού για το περιεχόμενο, το οποίο θα αναρτάται στις εκάστοτε ιστοσελίδες. Σκοπός του διαδικτυακού τύπου συνάντησης είναι η αλληλεπικοινωνία μεταξύ των χρηστών, εκπαιδευομένων και εκπαιδευτών, και όχι η ανάρτηση των απαντήσεων σε κάποιο από τα διαγωνίσματα. Αξιόλογη επίσης βελτίωση του εκπαιδευτικού λογισμικού μπορεί να είναι η δυνατότητα σύνθεσης, ανάρτησης και ελέγχου ομαδικών ασκήσεων, οι οποίες θα ενισχύσουν το συνεργατικό χαρακτήρα του.

[6] ΒΙΒΛΙΟΓΡΑΦΙΑ

ΕΛΛΗΝΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ

- Βίρβου Μαρία**, *Αλληλεπίδραση ανθρώπου-υπολογιστή*, Πανεπιστήμιο Πειραιά.
- Βίρβου Μαρία**, *Τεχνολογία Λογισμικού*, Πανεπιστήμιο Πειραιά.
- Κόμης Β. (2004)**, *Εισαγωγή στις Εφαρμογές των ΤΠΕ στην Εκπαίδευση*, Αθήνα, Εκδόσεις Νέων Τεχνολογιών.
- Παναγιωτόπουλος Χ., Πιερρακάς Χ., Πιντέλας Π. (2003)**, *Το εκπαιδευτικό λογισμικό και η αξιολόγησή του*, Εκδόσεις Μεταίχμιο.
- Πανέτος Σ. (2001)**, *Οι υπολογιστές στην εκπαίδευση*, Εκδόσεις ΙΩΝ.
- Πρέζας Π. (2003)**, *Θεωρίες Μάθησης και Εκπαιδευτικό Λογισμικό*, Εκδόσεις Κλειδάριθμος.
- Ράπτης Αρ. & Ράπτη Αθ. (1997)**, *Πληροφορική και Εκπαίδευση : Συνολική προσέγγιση*. Αθήνα, Εκδόσεις Τελέθριον
- Ελληνική Επιστημονική Ένωση Τεχνολογιών Πληροφορίας και Επικοινωνιών στην Εκπαίδευση (ΕΤΠΕ) (2006)**, *Οι τεχνολογίες της πληροφορίας και της επικοινωνίας στην εκπαίδευση*, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
- Gutenberg - Γιώργος & Κώστας Δαρδανός (2006)**, *Σχεδιάζοντας περιβάλλοντα μάθησης υποστηριζόμενα από τις σύγχρονες τεχνολογίες*.
- Τζιμογιάννης, Α. (2001)**, *Οι ΤΠΕ στην Εκπαίδευση: Πραγματικότητα και προοπτικές*, σ. 4
- Νίκος Παπασταματίου**, φυσικός, επίτιμος Σχολικός Σύμβουλος, Διαφάνειες με θέμα «*Κατηγοριοποίηση, Πιστοποίηση, Αξιολόγηση*», ΕΠΙΜΟΡΦΩΣΗ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΣΤΗ ΧΡΗΣΗ ΚΑΙ ΑΞΙΟΠΟΙΗΣΗ ΤΩΝ ΤΠΕ ΣΤΗΝ ΕΚΠΑΙΔΕΥΤΙΚΗ ΔΙΔΑΚΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ.

ΑΝΑΦΟΡΕΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

<http://el.wikipedia.org>

<http://www.e-yliko.gr/resource/supportmaterial/EduAll.aspx>

<http://www.netschoolbook.gr/epimorfosi/introduction.html>

http://ekped.gr/praktika/gen/10_63k.swf

[7]ΠΑΡΑΡΤΗΜΑΤΑ

Παράρτημα Α: Ενότητες θεωρίας βασικών στοιχείων Matlab

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1. Γενικά Στοιχεία

Το MATLAB είναι ένα σύγχρονο ολοκληρωμένο μαθηματικό λογισμικό πακέτο. Το όνομά του προέρχεται από τα αρχικά γράμματα των λέξεων MATrix LABoratory (εργαστήριο πινάκων). Πρόκειται για ένα περιβάλλον προγραμματισμού με σκοπό την ανάπτυξη αλγορίθμων, την ανάλυση δεδομένων, την οπτικοποίηση, και τους αριθμητικούς υπολογισμούς. Χρησιμοποιώντας το MATLAB, μπορούμε να επιλύσουμε τεχνικά προβλήματα πληροφορικής ταχύτερα από ότι με τις παραδοσιακές γλώσσες προγραμματισμού, όπως η C, C ++, και η Fortran.



Ξεκίνησε ως ένα υπολογιστικό εργαλείο για την εύκολη διαχείριση πινάκων, οι οποίοι αποτελούν και τη βασική δομή δεδομένων του MATLAB. Επιπλέον, είναι εφοδιασμένο με πολλές επιλογές για γραφικά (δηλ. την κατασκευή γραφικών παραστάσεων) και προγράμματα γραμμένα στη ίδια γλώσσα προγραμματισμού για την επίλυση άλλων προβλημάτων, όπως η εύρεση των ριζών μη γραμμικής εξίσωσης, η επίλυση μη γραμμικών συστημάτων κ.α.

1.2. Ιστορικά Στοιχεία

Στα τέλη της δεκαετίας του 1970 ο πρόεδρος του τμήματος Πληροφορικής του Πανεπιστημίου του Νέου Μεξικού, Cleve Moler, ανέπτυξε την πρώτη έκδοση του MATLAB, με σκοπό την πρόσβαση των φοιτητών του σε LINPACK και EISPACK χωρίς να χρειάζεται να μάθουν Fortran. Σύντομα και μέσω συνεδρίων, εξαπλώθηκε σε άλλα πανεπιστήμια και βρήκε έναν ισχυρό ακροατήριο μέσα στην κοινότητα των εφαρμοσμένων μαθηματικών. Τη δεκαετία του 1980, το πρόγραμμα γράφτηκε ξανά σε γλώσσα C με περισσότερες λειτουργίες, βιβλιοθήκες καθώς και δυνατότητα γραφικών και απεικονίσεων από τους Moler, Jack Little και Steve Bangert. Το 1984 δημιουργήθηκε η εταιρεία MathWorks Inc η οποία είναι υπεύθυνη μέχρι σήμερα για τη διανομή, ανάπτυξη, υποστήριξη και πώληση του MATLAB.

Υιοθετήθηκε για πρώτη φορά από τους ερευνητές και τους επαγγελματίες ελέγχου μηχανικής, αλλά γρήγορα εξαπλώθηκε σε πολλούς τομείς. Σήμερα χρησιμοποιείται επίσης στην εκπαίδευση, ιδίως για τη διδασκαλία της γραμμικής άλγεβρας και αριθμητικής ανάλυσης. Τέλος, είναι ιδιαίτερα δημοφιλής μεταξύ των επιστημόνων που ασχολούνται με την επεξεργασία εικόνας.

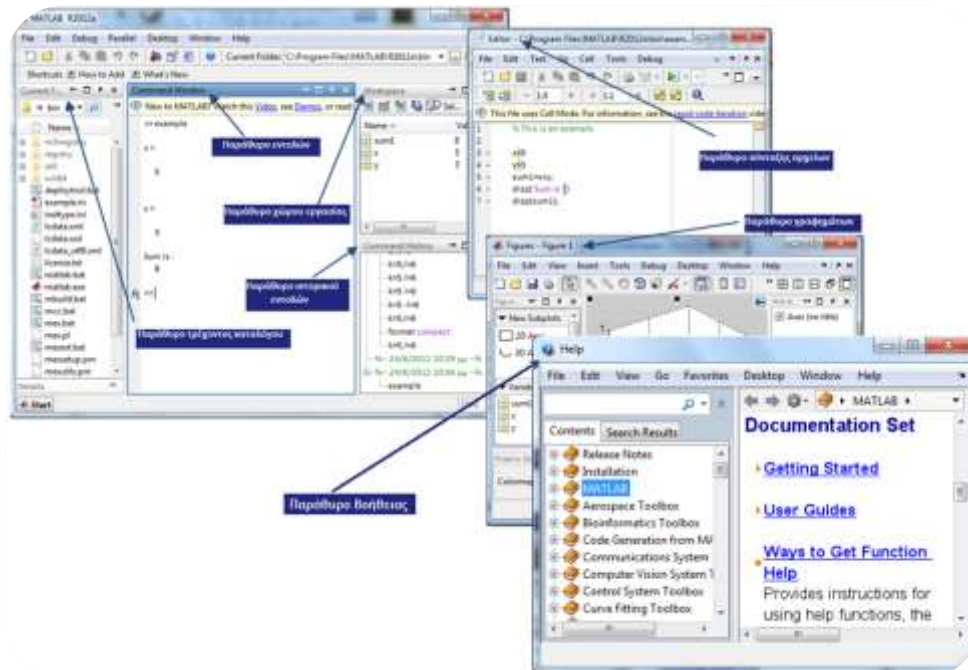
1.3. Παράθυρο Έναρξης της MATLAB

Αφού εγκαταστήσουμε το MATLAB στον υπολογιστή μας, χωρίς να αλλάξουμε κάποια από τις προτεινόμενες ρυθμίσεις του, μπορούμε να ξεκινήσουμε το πρόγραμμα με δύο τρόπους. Ο πρώτος τρόπος είναι μέσω διπλού κλικ πάνω στο εικονίδιο συντόμευσης, το οποίο βρίσκεται συνήθως στην επιφάνεια εργασίας. Ο δεύτερος τρόπος είναι μέσω του μενού Start → All Programs → Matlab → R2012a → Matlab R2012a. Όταν το πρόγραμμα φορτωθεί, εμφανίζεται στην οθόνη το παράθυρο έναρξης του MATLAB (MATLAB opening window).



Πιο αναλυτικά, εμφανίζονται επτά παράθυρα:

1. Παράθυρο εντολών (Command Window)
2. Παράθυρο τρέχοντος καταλόγου (Current directory)
3. Παράθυρο χώρου εργασίας (Workspace)
4. Παράθυρο ιστορικού εντολών (Command History)
5. Παράθυρο σύνταξης αρχείων (Editor Window)
6. Παράθυρο γραφημάτων (Figure Window)
7. Παράθυρο βοήθειας (Help)



Εικόνα 2 - Επιφάνεια εργασίας Matlab

1. **Παράθυρο εντολών (Command Window):** Αυτό είναι το βασικό παράθυρο, συνήθως στα δεξιά. Οι εντολές της MATLAB εισάγονται σ' αυτό το παράθυρο μετά την προτροπή ">>". Τα αποτελέσματα επίσης τυπώνονται στο παράθυρο αυτό.

2. **Παράθυρο τρέχοντος καταλόγου (Current directory):** Είναι ένα μικρό παράθυρο της επιφάνειας εργασίας του MATLAB, συνήθως πάνω αριστερά, όπου αναγράφεται ο τρέχον κατάλογος (directory) του συστήματος και το σύνολο των αρχείων, τα οποία είναι αποθηκευμένα σ' αυτόν. Με τη χρήση του ποντικιού μπορούμε να πλοηγηθούμε στα διάφορα αυτά αρχεία. Κάνοντας δεξί κλικ με το ποντίκι πάνω στο επιλεγμένο αρχείο είναι δυνατή η εκτέλεση διάφορων επιλογών, οι οποίες σχετίζονται με αυτό όπως η μετονομασία αρχείου, η διαγραφή αρχείου και η εκτέλεση M- File. Αν το παράθυρο είναι κρυμμένο, επιλέγουμε Current Directory.

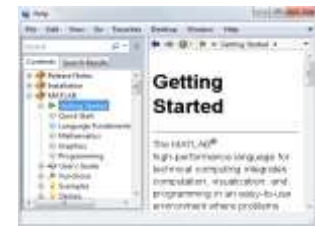
3. **Παράθυρο χώρου εργασίας (Workspace):** Εδώ απεικονίζονται όλες οι μεταβλητές μαζί με πληροφορίες σχετικά με τον τύπο και το μέγεθός τους, οι οποίες εισάγονται και χρησιμοποιούνται στο Παράθυρο εντολών. Ο χρήστης έχει τη δυνατότητα να ανακαλέσει ανά πάσα στιγμή τη μεταβλητή αυτή στο Παράθυρο εντολών, πληκτρολογώντας απλώς το όνομα της. Στις τελευταίες εκδόσεις του MATLAB προσφέρεται η δυνατότητα για κάθε μεταβλητή, η οποία καταγράφεται στο Παράθυρο του χώρου εργασίας, να αναπαρίσταται και μέσω γραφήματος. Το παράθυρο αυτό συνήθως εναλλάσσεται με το παράθυρο τρέχοντα φακέλου ανάλογα με την επιλογή Workspace ή Current Directory από το μενού Desktop. Ωστόσο, υπάρχει η δυνατότητα μόνιμης εμφάνισής του, η οποία είναι συνήθως στο πάνω δεξιό τμήμα της επιφάνειας εργασίας του MATLAB.

4. **Παράθυρο ιστορικού εντολών (Command History):** Είναι ένα μικρό παράθυρο της επιφάνειας εργασίας του MATLAB, συνήθως κάτω αριστερά, όπου αναγράφεται το σύνολο των καταχωρημένων εντολών, οι οποίες έχουν εκτελεστεί στο πλαίσιο πολυσύνθετων τμημάτων, ακόμα και μέρες πριν από την τελευταία εισαγωγή στο σύστημα. Παρέχεται επίσης η δυνατότητα να επιλεγεί από αυτό το παράθυρο μια επιθυμητή εντολή και στη συνέχεια να εκτελεστεί στο παράθυρο εντολών, κάνοντας διπλό κλικ με το ποντίκι πάνω στην εντολή αυτή. Αν δεν εμφανιστεί το παράθυρο αυτό μπορούμε να επιλέξουμε Command History στην επιλογή Window.

5. **Παράθυρο σύνταξης αρχείων (Editor Window):** Είναι το παράθυρο, στο οποίο ο χρήστης μπορεί να αναπτύξει, να επεξεργαστεί, να αποθηκεύσει τα δικά του αρχεία εντολών, τα οποία κυρίως απαρτίζονται από τα M-Files. Τα M-Files αναφέρονται αναλυτικότερα στο κεφάλαιο 12. Αν και είναι δυνατό τα αρχεία αυτά να συνταχθούν μέσω των κλασικών προγραμμάτων σύνταξης (notepad, notepad++ κ.α.), το MATLAB προσφέρει το αντίστοιχο πρόγραμμα, το οποίο είναι ενσωματωμένο στο πακέτο λογισμικού.

6. **Παράθυρο γραφημάτων (Figure Window):** Είναι ένα ξεχωριστό παράθυρο, από την επιφάνεια εργασίας του MATLAB, όπου παρέχεται το αποτέλεσμα των εντολών, οι οποίες σχετίζονται με τα γραφήματα και έχουν εκτελεστεί είτε απευθείας στο Παράθυρο εντολών είτε έμμεσα μέσω ενός αρχείου M-File. Από τις τελευταίες εκδόσεις του MATLAB (MATLAB 7) παρέχεται επίσης αυτό το παράθυρο η επεξεργασία και ο χειρισμός των γραφημάτων.

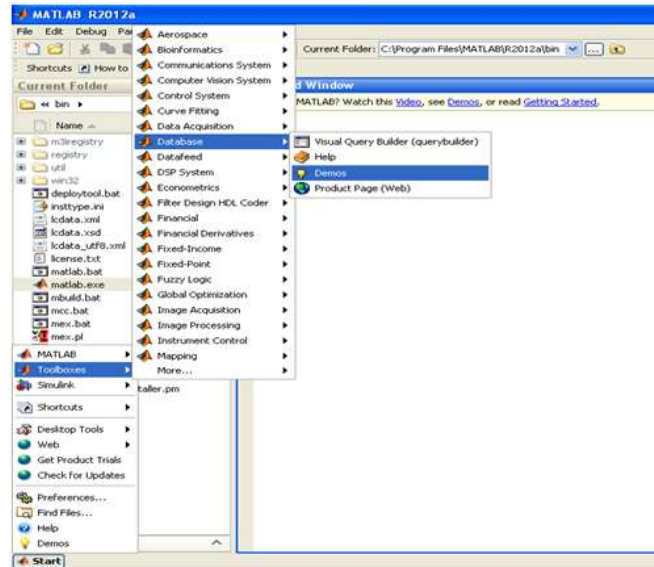
7. **Παράθυρο βοήθειας (Help):** Πρόκειται και αυτό για ένα ξεχωριστό παράθυρο, από την επιφάνεια εργασίας του MATLAB, όπου παρέχεται πλήρης βοήθεια για όλο το περιβάλλον, τις εντολές, τις συναρτήσεις και τα toolbox του MATLAB. Η καρτέλα Contents περιλαμβάνει όλες τις σχετικές κατηγορίες αναζήτησης βοήθειας που μπορεί να χρειαστεί ένας χρήστης, αλλά και την καρτέλα Index, η οποία περιλαμβάνει ένα πεδίο αναζήτησης για ελεύθερη επιλογή του χρήστη. Ιδιαίτερα σημαντικό είναι το τμήμα



Getting Started, μέσω του οποίου κάποιος μπορεί να μάθει εύκολα τις δυνατότητες του MATLAB.

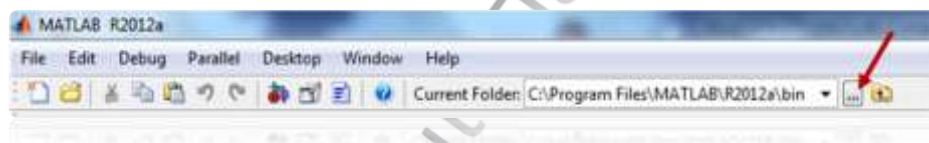
1.4. Πλήκτρο Έναρξης και Αλλαγή Τρέχοντα Φακέλου της MATLAB

Κλείνοντας την ενότητα αυτή, πρέπει να γίνει αναφορά στο **πλήκτρο της Έναρξης (Start)**, το οποίο μπορεί να χρησιμοποιηθεί για επίδειξη συγκεκριμένων εργαλειαθκών του MATLAB, καθώς και για επεξήγηση του τρόπου εφαρμογής αυτών. Στα αριστερά βλέπουμε τη χρήση του πλήκτρου έναρξης για την επίδειξη ενός Demo στην εργαλειαθήκη των βάσεων δεδομένων.



Εικόνα 3 - Matlab Toolboxes

Επίσης, αν επιθυμούμε μπορούμε να αλλάξουμε τον τρέχοντα φάκελο, στον οποίο εργαζόμαστε και να επιλέξουμε τον προσωπικό μας φάκελο, κάνοντας την κατάλληλη επιλογή στο τετραγωνίδιο που δείχνει το κόκκινο βέλος στο ακόλουθο σχήμα.



Εικόνα 4 – Αλλαγή τρέχοντα φακέλου Matlab

1.5. Έξοδος από το περιβάλλον της MATLAB

Η έξοδος από το πρόγραμμα μπορεί να γίνει με τρεις τρόπους:

1. με την εντολή **quit** ή την εντολή **exit** στο παράθυρο εντολών
2. με κλικ στο τετραγωνίδιο **[X]** που βρίσκεται πάνω δεξιά στο παράθυρο του MATLAB και τέλος
3. με την επιλογή **File** → **Exit MATLAB** από το παράθυρο εργασίας.

1.6. Βασικές Οδηγίες για το Παράθυρο Εντολών “COMMAND WINDOW”

- ✓ Η MATLAB διακρίνει τους κεφαλαίους από τους πεζούς χαρακτήρες. Οι μεταβλητές **A** και **a** είναι διαφορετικές μεταξύ τους. Οπότε, είναι απαραίτητη η σωστή και συνεπής πληκτρολόγηση των ονομάτων που χαρακτηρίζουν ενσωματωμένες εντολές, σταθερές και μεταβλητές, αρχεία, συναρτήσεις κλπ.
- ✓ Όταν εισάγουμε το όνομα μιας μεταβλητής στο παράθυρο εντολών, η MATLAB τυπώνει στην οθόνη το αποτέλεσμα της εκτέλεσης και συγκεκριμένα την τιμή της.

π.χ.

```
>> 2
ans =
2
```

- ✓ Αν εισάγουμε το σύμβολο ';', δηλαδή το ελληνικό ερωτηματικό, στο τέλος μιας εντολής, το αποτέλεσμα της δεν τυπώνεται στην οθόνη.
- ✓ Μπορούμε να εισάγουμε μια ακολουθία εντολών της MATLAB σε μια γραμμή χωρίζοντάς τις όμως με κόμματα.

Παράδειγμα 1	Παράδειγμα 2
<code>>> 2,3</code>	<code>>>k=5,l=6</code>
<code>ans =</code>	<code>k =</code>
<code>2</code>	<code>5</code>
<code>ans =</code>	<code>l =</code>
<code>3</code>	<code>6</code>

- ✓ Πατώντας τα πλήκτρα με τα πάνω και κάτω βέλη ([↑] και [↓]) μπορούμε να διατρέξουμε όλες τις προηγούμενες εντολές. Επίσης, μια προηγούμενη εντολή μπορεί να επαναληφθεί αν εισάγουμε τα πρώτα γράμματα και μετά πατήσουμε το πλήκτρο με το πάνω βέλος [↑].

ΚΕΦΑΛΑΙΟ 2: ΒΟΗΘΕΙΑ ΤΟΥ MATLAB – ΠΛΗΡΟΦΟΡΙΕΣ ΕΚΔΟΣΗΣ

Ένα σημαντικό πλεονέκτημά του MATLAB είναι ένα εκτενές σύστημα βοήθειας όπου κάθε εντολή επεξηγείται αναλυτικά και με αντιπροσωπευτικά παραδείγματα. Στην επίσημη ιστοσελίδα της MATLAB, <http://www.mathworks.com>, μπορεί κάποιος να βρει μια πληθώρα πληροφοριών τόσο για αρχάριους όσο και προχωρημένους.

2.1. Εντολή HELP

Εκτός από το Παράθυρο βοήθειας (Help), μέσα από το οποίο μας παρέχεται πληθώρα βοήθειας όσον αφορά το MATLAB, εξίσου σημαντική είναι και η εντολή **help**. Πληκτρολογώντας την εντολή **help** και την εντολή στην οποία χρειαζόμαστε βοήθεια, στο Παράθυρο Εντολών (Command Window), μπορούμε να βρούμε αναλυτικές πληροφορίες για τη σύνταξη της εντολής, τις ιδιότητες της, τις παραμέτρους της (αν υπάρχουν), αλλά και πολλά παραδείγματα που κάνουν ευκολότερη την κατανόηση της.

Για παράδειγμα αν θέλουμε να λάβουμε πληροφορίες σχετικά με τη σύνταξη της συνάρτησης της τετραγωνικής ρίζας, πληκτρολογούμε **help sqrt**. Η MATLAB αμέσως μας εμφανίζει :

π.χ.

```
>> help sqrt
```

sqrt Square root.

sqrt(X) is the square root of the elements of X. Complex results are produced if X is not positive.

See also [sqrtm](#), [realsqrt](#), [hypot](#).

Overloaded methods:

[codistributed/sqrt](#)
Reference page in Help browser
[doc sqrt](#)

Στην περίπτωση που θέλουμε να λάβουμε πληροφορίες σχετικά με τη σύνταξη της συνάρτησης του δεκαδικού λογάριθμου, πληκτρολογούμε **help log10**. Η MATLAB μας εμφανίζει:

π.χ.

```
>> help log10
log10 Common (base 10) logarithm.
log10(X) is the base 10 logarithm of the elements of X.
Complex results are produced if X is not positive.
See also log, log2, exp, logm.
Overloaded methods:
codistributed/log10
fints/log10
Reference page in Help browser
doc log10
```

2.2. Εντολές VERSION, VER

Η πρώτη εντολή είναι η εντολή `version`, με την οποία μπορούμε να μάθουμε την έκδοση της MATLAB που χρησιμοποιούμε. Παρακάτω φαίνεται το αποτέλεσμα της MATLAB όταν αυτή πληκτρολογηθεί στο παράθυρο εντολών.

π.χ.

```
>> version

ans =
7.14.0.739 (R2012a)
```

Με την εντολή `ver` παίρνουμε περισσότερες πληροφορίες, όπως το λειτουργικό σύστημα (operating system), την έκδοση του μεταγλωττιστή (compiler) και τις εκδόσεις των εργαλειοθηκών (toolboxes) του πακέτου:

π.χ.

```
>> ver
7.14.0.739 (R2012a)

>> ver
-----
MATLAB Version: 7.14.0.739 (R2012a)
MATLAB License Number: 161052
Operating System: Microsoft Windows 7 Version 6.1 (Build 7601: Service Pack 1)
Java Version: Java 1.6.0_17-b04 with Sun Microsystems Inc. Java HotSpot(TM) 64-Bit Server VM mixed mode
-----
```

<i>MATLAB</i>	<i>Version 7.14 (R2012a)</i>
<i>Simulink</i>	<i>Version 7.9 (R2012a)</i>
<i>Aerospace Blockset</i>	<i>Version 3.9 (R2012a)</i>
<i>Aerospace Toolbox</i>	<i>Version 2.9 (R2012a)</i>
<i>Bioinformatics Toolbox</i>	<i>Version 4.1 (R2012a)</i>
<i>Communications System Toolbox</i>	<i>Version 5.2 (R2012a)</i>
<i>Computer Vision System Toolbox</i>	<i>Version 5.0 (R2012a)</i>
<i>Control System Toolbox</i>	<i>Version 9.3 (R2012a)</i>
<i>Curve Fitting Toolbox</i>	<i>Version 3.2.1 (R2012a)</i>
<i>DO Qualification Kit</i>	<i>Version 1.6 (R2012a)</i>
<i>DSP System Toolbox</i>	<i>Version 8.2 (R2012a)</i>
<i>Data Acquisition Toolbox</i>	<i>Version 3.1 (R2012a)</i>
<i>Database Toolbox</i>	<i>Version 3.11 (R2012a)</i>
<i>Datafeed Toolbox</i>	<i>Version 4.3 (R2012a)</i>
<i>Econometrics Toolbox</i>	<i>Version 2.1 (R2012a)</i>
<i>Embedded Coder</i>	<i>Version 6.2 (R2012a)</i>
<i>Filter Design HDL Coder</i>	<i>Version 2.9.1 (R2012a)</i>
<i>Financial Derivatives Toolbox</i>	<i>Version 5.9 (R2012a)</i>
<i>Financial Toolbox</i>	<i>Version 4.2 (R2012a)</i>
<i>Fixed-Income Toolbox</i>	<i>Version 2.3 (R2012a)</i>
<i>Fixed-Point Toolbox</i>	<i>Version 3.5 (R2012a)</i>
<i>Fuzzy Logic Toolbox</i>	<i>Version 2.2.15 (R2012a)</i>
<i>Global Optimization Toolbox</i>	<i>Version 3.2.1 (R2012a)</i>
<i>HDL Coder</i>	<i>Version 3.0 (R2012a)</i>
<i>HDL Verifier</i>	<i>Version 4.0 (R2012a)</i>
<i>IEC Certification Kit</i>	<i>Version 2.1 (R2012a)</i>
<i>Image Acquisition Toolbox</i>	<i>Version 4.3 (R2012a)</i>
<i>Image Processing Toolbox</i>	<i>Version 8.0 (R2012a)</i>
<i>Instrument Control Toolbox</i>	<i>Version 3.1 (R2012a)</i>
<i>MATLAB Builder EX</i>	<i>Version 2.2 (R2012a)</i>
<i>MATLAB Builder JA</i>	<i>Version 2.2.4 (R2012a)</i>
<i>MATLAB Builder NE</i>	<i>Version 4.1.1 (R2012a)</i>
<i>MATLAB Coder</i>	<i>Version 2.2 (R2012a)</i>

.....

ΚΕΦΑΛΑΙΟ 3: ΕΠΙΤΡΕΠΤΑ ΜΕΓΕΘΗ ΣΤΗ MATLAB

Τα επιτρεπτά μεγέθη στη MATLAB αποτελούν οι ακέραιοι και δεκαδικοί αριθμοί, οι κλασματικοί αριθμοί, οι μιγαδικοί αριθμοί, οι μονοδιάστατοι και δισδιάστατοι πίνακες καθώς και οι ακολουθίες χαρακτήρων. Εκτός από τα παραπάνω, είναι δυνατόν να οριστούν και άλλα τρία επιτρεπτά μεγέθη μέσω μεταβλητών. Αυτά είναι οι πολυδιάστατοι πίνακες, οι δομές δεδομένων και οι πίνακες κελιών. Δεδομένου ότι η ανάθεσή τους σε μεταβλητές κρίνεται απαραίτητη, για την καλύτερη κατανόησή τους, θα αναφερθούμε σε αυτά ύστερα από το κεφάλαιο 6 «ΜΕΤΑΒΛΗΤΕΣ».

3.1. Αριθμοί

Η MATLAB διαχειρίζεται όλους τους αριθμούς σαν να είναι πραγματικοί διπλής ακρίβειας (float, double precision) είτε είναι ακέραιοι (integer) είτε είναι πραγματικοί (real).

3.1.1. Ακέραιοι Αριθμοί

Για τους θετικούς αριθμούς το θετικό πρόσημο “+” είναι προαιρετικό, ενώ για τους αρνητικούς το αρνητικό πρόσημο “-” είναι υποχρεωτικό. Έτσι οι αριθμοί 3 και +3 ταυτίζονται όταν δοθούν ως είσοδο στο παράθυρο εντολών του MATLAB. Ωστόσο ο αντίθετός τους είναι μόνο ο -3.

3.1.2. Δεκαδικό Αριθμοί

Οι δεκαδικοί αριθμοί δηλώνονται με δύο τρόπους. Είτε σε δεκαδική μορφή με τη χρήση της τελείας στη θέση της υποδιαστολής είτε σε ακέραια ή δεκαδική μορφή πολλαπλασιασμένη με τη δύναμη του δέκα. Έτσι, ο αριθμός 1,254 πρέπει να πληκτρολογηθεί ως 1.254 στο παράθυρο εντολών του MATLAB. Όπως αναφέραμε και προηγουμένως στις βασικές οδηγίες για το παράθυρο εντολών, είναι δυνατή η πληκτρολόγηση πολλών εντολών στην ίδια γραμμή χωρισμένες όμως με κόμμα. Γι’ αυτό το λόγο και οι δεκαδικοί αριθμοί δίνονται με τελεία προς αποφυγή αυτού του λάθους. Στο παρακάτω παράδειγμα φαίνεται η διαφορά του αποτελέσματος στο MATLAB όταν ο αριθμός 1,254 πληκτρολογηθεί με τελεία ή υποδιαστολή.

Παράδειγμα 1	Επεξήγηση
<pre>>> 1.254 ans = 1.2540</pre>	Ο αριθμός δηλώθηκε σωστά ως δεκαδικός και αναγνωρίστηκε επιτυχώς από το MATLAB.
Παράδειγμα 2	Επεξήγηση
<pre>>> 1,254 ans = 1 ans = 254</pre>	Ο αριθμός δε δηλώθηκε σωστά, αφού χρησιμοποιήθηκε η υποδιαστολή. Έτσι, το MATLAB αναγνώρισε δύο διαφορετικούς ακέραιους αριθμούς (το 1 και το 254) .

ΑΚΕΡΑΙΑ Η ΔΕΚΑΔΙΚΗ ΜΟΡΦΗ ΠΟΛΛΑΠΛΑΣΙΑΣΜΕΝΗ ΜΕ ΔΥΝΑΜΗ ΤΟΥ ΔΕΚΑ

Οι πραγματικές τιμές μπορούν να γραφούν είτε με τον συμβολισμό της σταθερής υποδιαστολής (fixed-point notation) είτε με τον επιστημονικό συμβολισμό (scientific notation). Έστω για παράδειγμα ότι επιθυμούμε να διαχειριστούμε δύο πολύ μεγάλους ή πολύ μικρούς αριθμούς, τους αριθμούς 0.000234 και 5.210.000.000. Στον επιστημονικό συμβολισμό γράφουμε τους πιο πάνω αριθμούς ως $234 * 10^{-6}$ και $521 * 10^7$. Στο MATLAB γράφουμε τους δύο αριθμούς αντικαθιστώντας τον εκθέτη του 10 με το χαρακτήρα e. Έτσι οι παραπάνω αριθμοί γράφονται:

$$234 * 10^{-6} \rightarrow 234e-6 \quad \text{και} \\ 521 * 10^7 \rightarrow 521e7$$

! Είναι πολύ σημαντικό να μην υπάρχει κενό μεταξύ του δεκαδικού αριθμού και του e ή μεταξύ του e και του εκθέτη. Αν αφήσουμε κενό η MATLAB θα διαβάσει δύο αριθμούς αντί για ένα!

Παράδειγμα

Έστω ο αριθμός $2.3 \cdot 10^{-5}$. Αν αφήσουμε κενό μεταξύ του δεκαδικού αριθμού και του e παίρνουμε μήνυμα λάθους από το MATLAB.

```
>> 2.3 e-5
??? 2.3 e-5
|
Error: Unexpected MATLAB operator.
```

Το ίδιο συμβαίνει και στην περίπτωση που αφήσουμε κενό μεταξύ του e και του εκθέτη.

```
>> 2.3e -5
??? 2.3e -5
|
Error: Unexpected MATLAB operator.
```

Αν όμως γράψουμε τώρα σωστά τον αριθμό η MATLAB μας επιστρέφει τον αριθμό αποθηκευμένο στη μεταβλητή ans.

```
>> 2.3e-5
ans =
    2.3000e-005
```

3.1.3. Κλασματικοί Αριθμοί

Οι κλασματικοί αριθμοί αποτελούν μία από τις πιο απλές κατηγορίες αριθμών, αφού όλα τα κλάσματα υποδηλώνουν διαίρεση. Έτσι, όλα τα κλάσματα δηλώνονται ως **αριθμητής/παρονομαστής** στο παράθυρο εντολών του MATLAB. Για παράδειγμα, το κλάσμα $\frac{2}{5}$ δηλώνεται με τον εξής τρόπο: `>> 2/5`

Ωστόσο, το αποτέλεσμα που μας επιστρέφεται, ύστερα από την εισαγωγή του στο Παράθυρο εντολών, είναι σε μορφή πραγματικού διπλής ακρίβειας, αφού όλες οι πράξεις στη MATLAB γίνονται με διπλή ακρίβεια (double precision).

3.1.4. Μιγαδικοί Αριθμοί

Στη MATLAB δεν απαιτείται να δηλώνεται ο τύπος των μεταβλητών, με αποτέλεσμα να μη χρειάζεται να δηλωθεί αν μια μεταβλητή θα περιέχει πραγματικό ή μιγαδικό αριθμό. Εξ' ορισμού στο λογισμικό αυτό όλοι οι αριθμοί θεωρούνται μιγαδικοί, με τους πραγματικούς αριθμούς να αποτελούν μια ειδική περίπτωση των μιγαδικών αριθμών. Για τη σύνταξη των μιγαδικών αριθμών μπορούν να χρησιμοποιηθούν είτε ο χαρακτήρας i είτε ο χαρακτήρας j, οι οποίοι εξ' ορισμού παριστάνουν τη φανταστική μονάδα.

Η σύνταξη των μιγαδικών αριθμών πραγματοποιείται ως εξής **a+bi** ή **a+b*i**

Η πρώτη μορφή μεταγλωττίζεται πάντα από τη MATLAB ως μιγαδικός αριθμός, ενώ αντίθετα η τελευταία μορφή μόνο στην περίπτωση κατά την οποία δεν έχει οριστεί προηγούμενα το i ως μια άλλη μεταβλητή. Τα ίδια ακριβώς ισχύουν και το χαρακτήρα j .

Για τη διαχείριση τους μπορούμε να χρησιμοποιήσουμε όλες οι πράξεις των μιγαδικών αριθμών, αλλά και πολλές ενσωματωμένες συναρτήσεις για μιγαδικούς αριθμούς που διαθέτει το MATLAB.

Παρακάτω βλέπουμε παραδείγματα εισαγωγής μιγαδικών αριθμών:

Εντολή	Επεξήγηση
<pre>>> i ans = 0 + 1.0000i</pre>	Το i παριστάνει τη φανταστική μονάδα.
<pre>>> j ans = 0 + 1.0000i</pre>	Το j παριστάνει τη φανταστική μονάδα.
<pre>>> 2-3i ans = 2.0000 - 3.0000i</pre>	Μεταβλητή που περιέχει μιγαδικό αριθμό.
<pre>>> 16-8i ans = 16.0000 - 8.0000i</pre>	Μεταβλητή που περιέχει μιγαδικό αριθμό.
<pre>>> (2-3i) - (16-8i) ans = -14.0000 + 5.0000i</pre>	Αφαίρεση μιγαδικών αριθμών.
<pre>>> real((2-3i) - (16-8i)) ans = -14</pre>	Εμφάνιση του πραγματικού μέρους ενός μιγαδικού αριθμού.
<pre>>> imag((2-3i) - (16-8i)) ans = 5</pre>	Εμφάνιση του φανταστικού μέρους ενός μιγαδικού αριθμού.
<pre>>> 10*(16-8i) ans = 1.6000e+002 -8.0000e+001i</pre>	Πολλαπλασιασμός μιγαδικού αριθμού με πραγματικό αριθμό.

<pre>>> (2-3i)^2 ans = -5.0000 -12.0000i</pre>	Ύψωση ενός μιγαδικού αριθμού σε πραγματικό αριθμό.
---	--

3.2. Μονοδιάστατοι Πίνακες – Διανύσματα

Στο περιβάλλον της MATLAB οι πίνακες είναι ορθογώνια διατεταγμένοι αριθμοί και γράφονται μέσα σε αγκύλες “[]”. Για παράδειγμα ο πίνακας [2 5 3 1] ορίζεται ως εξής:

Παράδειγμα 1	Παράδειγμα 2
<pre>>>[2 5 3 1] ans = 2 5 3 1</pre>	<pre>>> [2,5,3,1] ans = 2 5 3 1</pre>

Παρατηρώντας το παραπάνω παράδειγμα βλέπουμε πως τα στοιχεία μίας γραμμής χωρίζονται είτε με τη χρήση κενών ανάμεσα των στοιχείων είτε με τη χρήση της υποδιαστολής “,”. Το κενό διάνυσμα ορίζεται ως [].

π.χ.

```
>> []

ans =
[]
```

Ένας εναλλακτικός τρόπος δημιουργίας ενός διανύσματος, χωρίς την ανάγκη εισαγωγής ενός προς ενός όλων των στοιχείων του διανύσματος, είναι ο ορισμός στην εντολή της πρώτης τιμής, της τελευταίας δυνατής τιμής και του βήματος μεταβολής των τιμών με την άνω και κάτω τελεία ενδιάμεσά τους. Αν δεν οριστεί συγκεκριμένη τιμή για το βήμα, τότε ορίζεται αυτόματα ίσο με τη μονάδα (βήμα=1). Σημειώνεται ότι η πρώτη τιμή, η τελευταία τιμή και το βήμα μεταβολής των τιμών μπορούν να είναι ακέραιοι ή πραγματικοί αριθμοί, θετικοί ή αρνητικοί. Παρακάτω παρουσιάζονται ορισμένα ενδεικτικά παραδείγματα της περίπτωσης αυτής.

Εντολή	Επεξήγηση
<pre>>> [0:10:30] ans = 0 10 20 30</pre>	δημιουργία διανύσματος πρώτο στοιχείο: 0 βήμα μεταβολής: 10 τελευταίο δυνατό στοιχείο: 30
<pre>>>[100:105] ans = 100 101 102 103 104 105</pre>	δημιουργία διανύσματος πρώτο στοιχείο: 100 βήμα μεταβολής: 1 (εξορισμού) τελευταίο δυνατό στοιχείο: 105
<pre>>>[5:-3:-9] ans = 5 2 -1 -4 -7</pre>	δημιουργία διανύσματος πρώτο στοιχείο: 5 βήμα μεταβολής: -3 τελευταίο δυνατό στοιχείο: -9

Όταν το βήμα είναι θετικό, η τελική τιμή θα πρέπει να είναι μεγαλύτερη ή ίση από την αρχική τιμή. Αντίθετα, όταν το βήμα είναι αρνητικό η τελική τιμή θα πρέπει να είναι μικρότερη ή ίση από την αρχική. Σε αυτή την περίπτωση, η MATLAB εμφανίζει μήνυμα λάθους και δε δημιουργεί το διάνυσμα που δόθηκε στο παράθυρο εντολών.

Παράδειγμα 1	
<pre>>> [2:-2:10] ans = Empty matrix: 1-by-0</pre>	<p>βήμα αρνητικό.</p> <p>λάθος : τελευταίο στοιχείο > πρώτο στοιχείο</p>
Παράδειγμα 2	
<pre>>>[2:2:1] ans = Empty matrix: 1-by-0</pre>	<p>βήμα θετικό.</p> <p>λάθος: τελευταίο στοιχείο < πρώτο στοιχείο</p>

3.3. Διδιάστατοι Πίνακες

Οι διδιάστατοι πίνακες ορίζονται με παρόμοιο τρόπο. Αρχικά ορίζονται τα στοιχεία της πρώτης γραμμής και στη συνέχεια η αλλαγή γραμμής γίνεται με τη χρήση του ελληνικού ερωτηματικού ";" ή με την αλλαγή γραμμής, συνδυάζοντας το Enter και το Shift. Έτσι, ο πίνακας $\begin{bmatrix} 2 & 3 \\ 5 & 1 \end{bmatrix}$ ορίζεται ως εξής:

Παράδειγμα 1	Παράδειγμα 2
<pre>>> [2 3;5 1] ans = 2 3 5 1</pre>	<pre>>> [2 3 5 1] ans = 2 3 5 1</pre>
Παράδειγμα 3	Παράδειγμα 4
<pre>>> [1, 4, 6;7, 1, 3] ans = 1 4 6 7 1 3</pre>	<pre>>> [] ans = []</pre>

! Προσοχή! Οι συνιστώσες ενός πίνακα βρίσκονται ανάμεσα σε αγκύλες και όχι παρενθέσεις!

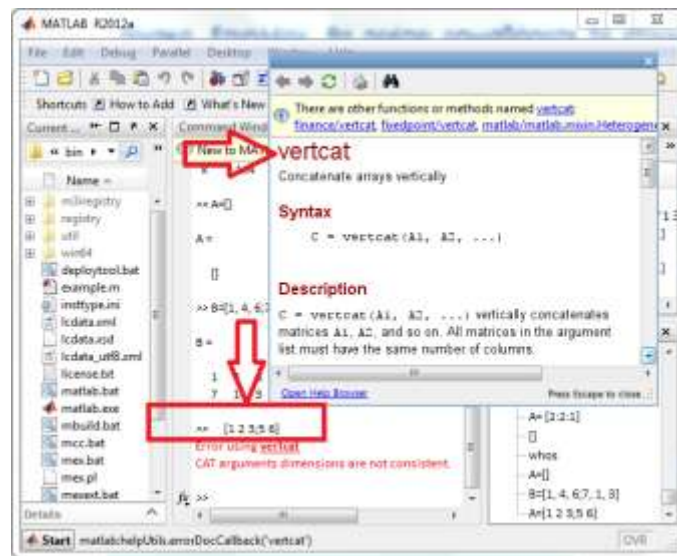
Παρακάτω βλέπουμε το μήνυμα λάθους που παίρνουμε όταν χρησιμοποιήσουμε παρενθέσεις αντί αγκύλες.

Π.χ.

```
>> (-5, 2, 3.5)
(-5, 2, 3.5)
```

Error: Expression or statement is incorrect--possibly unbalanced (, {, or [.

Τα στοιχεία των πινάκων θα πρέπει να είναι ορθώς εισηγμένα. Όπως ισχύει και στα Μαθηματικά, ένας πίνακας οφείλει να διαθέτει τον ίδιο αριθμό στοιχείων σε όλες τις γραμμές. Για παράδειγμα, αν ένας πίνακας έχει τρία στοιχεία στην πρώτη γραμμή, τότε οφείλει να έχει τόσα στοιχεία και σε οποιαδήποτε άλλη γραμμή. Σε αντίθετη περίπτωση, το λογισμικό εμφανίζει μήνυμα λάθους στο χρήστη, όπως φαίνεται στο παράδειγμα της παρακάτω εικόνας:



Εικόνα 5 – Μήνυμα λάθους MATLAB.

3.4. Ακολουθίες Χαρακτήρων

Προκειμένου μία ακολουθία χαρακτήρων να εισαχθεί και να αναγνωριστεί ως τέτοια (αντί για εντολή), πρέπει να περικλείεται από μονά εισαγωγικά. Στα μονά εισαγωγικά μπορεί να περικλείεται ένας χαρακτήρας (ελληνικός, λατινικός, αριθμός, σύμβολο κ.α.) ή και περισσότεροι. Παρακάτω παρατίθενται παραδείγματα εισαγωγής ακολουθιών χαρακτήρων στη MATLAB.

Παράδειγμα 1	Παράδειγμα 2
<pre>>> 'παράδειγμα'</pre> <p>ans = παράδειγμα</p>	<pre>>> 'matlab'</pre> <p>ans = matlab</p>
Παράδειγμα 3	Παράδειγμα 4
<pre>>> ' απλό παράδειγμα'</pre> <p>ans = απλό παράδειγμα</p>	<pre>>> ' Αυτό είναι ένα πιο σύνθετο παράδειγμα!'</pre> <p>ans = Αυτό είναι ένα πιο σύνθετο παράδειγμα!</p>

Παράδειγμα 5	Παράδειγμα 6
<pre>>> 'Αυτό αποτελεί "1" ακολουθία χαρακτήρων.'</pre> <p><i>ans =</i> Αυτό αποτελεί "1" ακολουθία χαρακτήρων.</p>	<pre>>> '**Μου αρέσει πολύ η M@TL@B!**'</pre> <p><i>ans =</i> **Μου αρέσει πολύ η M@TL@B!**</p>

ΚΕΦΑΛΑΙΟ 4: ΒΑΣΙΚΕΣ ΠΡΑΞΕΙΣ

Στο MATLAB οι επιτρεπτές πράξεις διακρίνονται σε τέσσερις κατηγορίες. Τις αριθμητικές πράξεις, τις συγκριτικές πράξεις, τις λογικές πράξεις και τις πράξεις χαρακτήρων. Τα μεγέθη που υπόκεινται στις αριθμητικές πράξεις είναι αριθμητικά, δηλαδή αριθμοί, διανύσματα και δισδιάστατοι πίνακες. Τα μεγέθη που υπόκεινται στις υπόλοιπες κατηγορίες πράξεων είναι είτε ακολουθίες χαρακτήρων είτε αριθμητικά μεγέθη, τα οποία αποδίδουν μία ακολουθία χαρακτήρων. Το MATLAB αντιμετωπίζει τις ακολουθίες χαρακτήρων ως διανύσματα, με τον κάθε χαρακτήρα τους να αποτελεί από ένα στοιχείο του διανύσματος. Επίσης, χρησιμοποιώντας την κωδικοποίηση ASCII, αντιστοιχεί κάθε χαρακτήρα σε ένα θετικό ακέραιο προκειμένου να είναι δυνατή η εκτέλεση οποιασδήποτε πράξης μεταξύ τους. Περισσότερες πληροφορίες για την κωδικοποίηση ASCII δίνονται στην επίσημη ιστοσελίδα της Βικιπαίδειας:

<http://el.wikipedia.org/wiki/ASCII>

4.1. Αριθμητικές Πράξεις

Εφόσον το MATLAB μπορεί να χρησιμοποιηθεί και σαν απλή αριθμομηχανή για τις βασικές αριθμητικές πράξεις χρησιμοποιούνται συγκεκριμένοι τελεστές (σύμβολα), οι οποίοι φαίνονται στον παρακάτω πίνακα:

Αριθμητικοί τελεστές	Πράξη
+	Πρόσθεση
-	Αφαίρεση
*	Πολλαπλασιασμός
/	Δεξιά Διαίρεση
\	Αριστερή Διαίρεση
^	Ύψωση σε δύναμη

Με τους αριθμητικούς τελεστές εκτελούνται οι γνωστές από την αριθμητική πράξεις μεταξύ ενός, δύο ή περισσότερων μεγεθών, τα οποία μπορεί να είναι ομοειδή ή και ετεροειδή. Παρακάτω παρατίθενται παραδείγματα αριθμητικών πράξεων:

Αριθμητικές πράξεις

Παράδειγμα 1	Παράδειγμα 2
<pre>>> 1+2 ans = 3</pre>	<pre>>> 3.14567-3 ans = 0.1457</pre>
Παράδειγμα 3	Παράδειγμα 4
<pre>>> 3*4 ans = 12</pre>	<pre>>> 3^2 ans = 9</pre>
Παράδειγμα 5	Παράδειγμα 6
<pre>>> 8/2 ans = 4</pre>	<pre>>> 8/2.2 ans = 3.6364</pre>
Παράδειγμα 7	Παράδειγμα 8
<pre>>> 8\2 ans = 0.2500</pre>	<pre>>> 2/8 ans = 0.2500</pre>

Στο MATLAB είναι επίσης δυνατό να υπολογισθεί οποιαδήποτε αριθμητική παράσταση, κάνοντας χρήση μιας μόνο εντολής. Η προτεραιότητα των πράξεων είναι ίδια με τις διαδικαστικές (procedural) γλώσσες^[2]. Ωστόσο, η προτεραιότητα των πράξεων επεξηγείται αναλυτικότερα στην ενότητα 4.5

Παράδειγμα 1	Παράδειγμα 2
<pre>>> (15-6)*5+3 ans = 48</pre>	<pre>>> (3-1)^2/(2*6+9) ans = 0.1905</pre>
Παράδειγμα 3	Παράδειγμα 4
<pre>>> (2.3+3.7/2.5)/(3^2.6+1/3) ans = 0.2132</pre>	<pre>>> (2^3.6+1/2)\(5+3.7/1.5) ans = 0.5914</pre>

[2] Διαδικαστικές γλώσσες (procedural) όπου το πρόγραμμα είναι οργανωμένο σε διαδικασίες, που αποτελούνται από σειρές εντολών που περιγράφουν αλγορίθμους. Παραδείγματα γλωσσών που ανήκουν σε αυτή την κατηγορία είναι η Pascal ή η C.

4.2. Συγκριτικές Πράξεις

Οι συγκριτικές πράξεις εφαρμόζονται μεταξύ ζευγών ομοειδών μεγεθών. Η σύγκριση μεταξύ δύο ποσοτήτων γίνεται με τους σχετικούς τελεστές. Το αποτέλεσμα της πράξης παίρνει δύο τιμές. Είτε το ένα όταν η σχέση είναι αληθής, είτε το μηδέν όταν η σχέση είναι ψευδής. Παρακάτω παρουσιάζονται οι τελεστές σύγκρισης καθώς επίσης και παραδείγματα για την καλύτερη κατανόηση τους:

Συγκριτικοί τελεστές	Πράξη
==	ίσον σύγκρισης
<	μικρότερο
<=	μικρότερο ή ίσο
>	μεγαλύτερο
>=	μεγαλύτερο ή ίσο
~=	όχι ίσο

Παράδειγμα 1	Παράδειγμα 2	Παράδειγμα 7	Παράδειγμα 8
<pre>>> 8>5 ans = 1</pre>	<pre>>> 10.65<12 ans = 1</pre>	<pre>>> 'papi'=='mami' ans = 0 1 0 1</pre>	<pre>>> 'papi'>='mami' ans = 1 1 1 1</pre>
Παράδειγμα 3	Παράδειγμα 4	Παράδειγμα 9	Παράδειγμα 10
<pre>>> 8<=2 ans = 0</pre>	<pre>>> 2==8 ans = 0</pre>	<pre>>> 'papi'<='mami' ans = 0 1 0 1</pre>	<pre>>> 'papi'<'mami' ans = 0 0 0 0</pre>
Παράδειγμα 5	Παράδειγμα 6	Παράδειγμα 11	Παράδειγμα 12
<pre>>> 4^2==16 ans = 1</pre>	<pre>>> 3^2_=9 ans = 0</pre>	<pre>>> 'papi'>'mami' ans = 1 0 1 0</pre>	<pre>>> 'papi'~='mami' ans = 1 0 1 0</pre>

Στη σύγκριση μεταξύ χαρακτήρων απαραίτητη προϋπόθεση είναι ισότητα μεταξύ των μηκών τους. Ως αποτέλεσμα το MATLAB εμφανίζει ένα διάνυσμα με τον αριθμό ένα, όπου οι ASCII αριθμοί των αντίστοιχων χαρακτήρων ικανοποιούν τη συνθήκη, αλλιώς με τον αριθμό μηδέν. Σε περίπτωση που τα μήκη των ακολουθιών δεν ταυτίζονται το MATLAB εμφανίζει μήνυμα λάθους, όπως φαίνεται στο παρακάτω παράδειγμα:

```
>> 'papi'~='ma'
Error using ~=
Matrix dimensions must agree.
```

4.3. Λογικές Πράξεις

Οι λογικές πράξεις εφαρμόζονται μεταξύ ομοειδών και ίδιων διαστάσεων μεγεθών. Τα μεγέθη αυτά μπορεί να είναι είτε αριθμητικά είτε ακολουθίες χαρακτήρων. Η λογική πράξη μπορεί να γίνει μεταξύ ενός ή περισσότερων μεγεθών και το αποτέλεσμα της παίρνει δύο τιμές. Είτε το ένα όταν είναι αληθής, είτε το μηδέν όταν είναι ψευδής. Παρακάτω παρουσιάζονται οι τελεστές λογικών πράξεων και μερικά ενδεικτικά παραδείγματα:

Λογικοί Τελεστές	Πράξη	Πίνακες αλήθειας																		
~	λογική πράξη NOT	<table border="1"> <thead> <tr> <th colspan="3">Πίνακας Αληθείας για AND</th> </tr> <tr> <th>A</th> <th>B</th> <th>A & B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	Πίνακας Αληθείας για AND			A	B	A & B	0	0	0	1	1	1	0	1	0	1	0	0
Πίνακας Αληθείας για AND																				
A	B	A & B																		
0	0	0																		
1	1	1																		
0	1	0																		
1	0	0																		
&	λογική πράξη AND	<table border="1"> <thead> <tr> <th colspan="3">Πίνακας Αληθείας για OR</th> </tr> <tr> <th>A</th> <th>B</th> <th>A B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> </tbody> </table>	Πίνακας Αληθείας για OR			A	B	A B	0	0	0	1	1	1	0	1	1	1	0	1
Πίνακας Αληθείας για OR																				
A	B	A B																		
0	0	0																		
1	1	1																		
0	1	1																		
1	0	1																		
	λογική πράξη OR	<table border="1"> <thead> <tr> <th colspan="2">Πίνακας Αληθείας για NOT</th> </tr> <tr> <th>A</th> <th>~A</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	Πίνακας Αληθείας για NOT		A	~A	0	1	1	0										
Πίνακας Αληθείας για NOT																				
A	~A																			
0	1																			
1	0																			

Παράδειγμα 1	Παράδειγμα 2
<pre>>> (7>=6 & 8<=8.49) ans = 1</pre>	<pre>>> (9<6 9>=8) ans = 1</pre>
Παράδειγμα 3	Παράδειγμα 4
<pre>>> ~[1 1 1 1] ans = 0 0 0 0</pre>	<pre>>> ~[0 0 1 0] ans = 1 1 0 1</pre>
Παράδειγμα 5	Παράδειγμα 6
<pre>>> [0 0 1 0] & [1 1 1 1] ans = 0 0 1 0</pre>	<pre>>> [0 1 1 0] & [0 1 1 1] ans = 0 1 1 0</pre>

Παράδειγμα 5	Παράδειγμα 6
<pre>>> [0 1 1 0] [1 1 1 1]</pre> <p>ans =</p> <p>1 1 1 1</p>	<pre>>> [0 1 1 0] [0 1 1 1]</pre> <p>ans =</p> <p>0 1 1 1</p>

4.4. Πράξεις Χαρακτήρων

Οι πράξεις χαρακτήρων εκφράζονται ως εντολές. Με τη χρήση τους μας δίνεται η δυνατότητα να συνενώσουμε ακολουθίες χαρακτήρων, να βρούμε τη θέση μιας ακολουθίας χαρακτήρων, η οποία περιέχεται σε μία άλλη, να μετασχηματίσουμε μία ακολουθία χαρακτήρων σε αριθμούς ASCII ή το αντίστροφο κ.α. Στον παρακάτω πίνακα παρουσιάζονται οι εντολές που χρησιμοποιούνται για τις πράξεις μεταξύ ακολουθιών χαρακτήρων καθώς επίσης και παραδείγματα εκτέλεσής τους για την καλύτερη κατανόησή τους:

Εντολές	Πράξη
<p>strcat('text1','text2','...')</p> <p>π.χ</p> <pre>>> strcat('good','morning!')</pre> <p>ans =</p> <p>goodmorning!</p>	<p>Συνενώνει τις ακολουθίες χαρακτήρων 'text1' και 'text2' και στις επιστρέφει ως μια ενιαία.</p> <p>→ Συννένωσε τις δύο λέξεις σε μία.</p>
<p>strvcat('text1','text2','...')</p> <p>π.χ</p> <pre>>> strvcat('good','morning!')</pre> <p>ans =</p> <p>good</p> <p>morning!</p>	<p>Συνενώνει κάθετα τις ακολουθίες χαρακτήρων 'text1' και 'text2' και τις επιστρέφει ως ένα διάνυσμα στήλης.</p> <p>→ Συννένωσε τις δύο λέξεις σε ένα διάνυσμα στήλης.</p>
<p>findstr('text1','text2')</p> <p>π.χ</p> <pre>>> findstr('good','od')</pre> <p>ans =</p> <p>3</p>	<p>Βρίσκει τις θέσεις της μικρότερης ακολουθίας χαρακτήρων (εκ των δύο) στη μεγαλύτερη και τις επιστρέφει ως ένα διάνυσμα αριθμών.</p> <p>→ Βρήκε τη λέξη "od" στην τρίτη θέση της λέξης "good".</p>
<p>strrep('text1','text2','text3')</p>	<p>Αντικαθιστά τη δεύτερη ακολουθία χαρακτήρων ('text2') με την τρίτη ('text3'),</p>

<p>π.χ >> <code>strrep('good morning','morning','night')</code></p> <p><i>ans =</i> <i>good night</i></p>	<p>όπου αυτή εμφανίζεται στην πρώτη ακολουθία χαρακτήρων ('text1') και τις επιστρέφει ως μία ακολουθία χαρακτήρων.</p> <p>→ Αντικατέστησε τη λέξη "morning" με τη λέξη "night".</p>
<p>char(n)</p> <p>π.χ >> <code>char(85)</code></p> <p><i>ans =</i> <i>U</i></p>	<p>Επιστρέφει το χαρακτήρα που αντιστοιχεί στο θετικό ακέραιο αριθμό ASCII n.</p> <p>→ Ο χαρακτήρας που αντιστοιχεί στον αριθμό 85 είναι το "U".</p>
<p>double('text')</p> <p>π.χ >> <code>double('U')</code></p> <p><i>ans =</i> <i>85</i></p>	<p>Επιστρέφει ένα διάνυσμα θετικών ακέραιων αριθμών όπου το κάθε στοιχείο του αντιστοιχεί σε κάθε ένα από τους χαρακτήρες τις ακολουθίας χαρακτήρων 'text'.</p> <p>→ Ο αριθμός που αντιστοιχεί στον χαρακτήρα "U" είναι το 85.</p>
<p>int2str(n)</p> <p>π.χ >> <code>a=int2str(85)</code></p> <p><i>a =</i> <i>85</i></p>	<p>Μετατρέπει τον ακέραιο αριθμό n σε ακολουθία χαρακτήρων.</p> <p>→ Μετέτρεψε τον ακέραιο αριθμό 85 στον χαρακτήρα 85.</p>
<p>num2str(n)</p> <p>π.χ >> <code>b=num2str(55)</code></p> <p><i>b =</i> <i>55</i></p>	<p>Μετατρέπει τον πραγματικό αριθμό n σε ακολουθία χαρακτήρων.</p> <p>→ Μετέτρεψε τον πραγματικό αριθμό 55 στον χαρακτήρα 55.</p>
<p>num2str(R)</p> <p>π.χ >> <code>M=num2str([5 44 7 8])</code></p> <p><i>M =</i> <i>5 44 7 8</i></p>	<p>Μετατρέπει τον πίνακα πραγματικών αριθμών R σε πίνακα από ακολουθίες χαρακτήρων.</p> <p>→ Μετέτρεψε τον πίνακα πραγματικών αριθμών σε πίνακα ακολουθιών χαρακτήρων. (Τα περιεχόμενα του πίνακα δε μπορούν πλέον να χρησιμοποιηθούν για πράξεις αριθμών)</p>

4.5. Προτεραιότητα των πράξεων

Τη μεγαλύτερη προτεραιότητα έχουν οι εκφράσεις που βρίσκονται μέσα σε παρενθέσεις. Στην περίπτωση όπου δεν υπάρχουν παρενθέσεις, τότε ακολουθείται η εξής προτεραιότητα για τις πράξεις:

- Οι πράξεις εκτελούνται με φορά από τα αριστερά προς τα δεξιά.
- Μεταξύ των βασικών πράξεων, η πράξη της ύψωσης σε δύναμη έχει τη μεγαλύτερη προτεραιότητα, ακολουθούν οι πράξεις του πολλαπλασιασμού και της διαίρεσης (οι οποίες, όπως και στα μαθηματικά, έχουν ίση προτεραιότητα μεταξύ τους) και τέλος ακολουθούν οι πράξεις της πρόσθεσης και της αφαίρεσης (όπου και αυτές έχουν ίση προτεραιότητα μεταξύ τους).

Με βάση την προαναφερθείσα προτεραιότητα των πράξεων, τα παρακάτω παραδείγματα επιστρέφουν το ίδιο αποτέλεσμα τόσο με τη χρήση των παρενεθέσεων όσο και χωρίς αυτές:

Πράξεις με παρενθέσεις	Πράξεις χωρίς παρενθέσεις
$\gg 10+5*4-(20/2)$ <p>ans = 20</p>	$\gg 10+5*4-20/2$ <p>ans = 20</p>
$\gg (10^2)-20+(5*4)$ <p>ans = 100</p>	$\gg 10^2-20+5*4$ <p>ans = 100</p>

Στην περίπτωση όπου χρησιμοποιούνται τελεστές σύγκρισης, λογικών πράξεων ή ακόμα και τελεστές πράξεων πινάκων στοιχείο προς στοιχείο, όπως θα δούμε στο κεφάλαιο 8, η προτεραιότητα των πράξεων φαίνεται στον παρακάτω ολοκληρωμένο πίνακα:

Τελεστές	Προτεραιότητα
()	μέγιστη
~	.
.' ^ ' ^ + -	.
.* ./ \ * / \	.
+ -	.
: < <= > >= == ~=	.
&	ελάχιστη

ΚΕΦΑΛΑΙΟ 5: ΜΕΤΑΒΛΗΤΕΣ

5.1. Δημιουργία Μεταβλητών – Καταχώριση τιμής σε μεταβλητές

Το αποτέλεσμα οποιασδήποτε από τις προαναφερθείσες εντολές ανατίθεται προσωρινά στο **ans**. Η ανάθεση του αποτελέσματος στο **ans** διατηρείται μέχρι την εκτέλεση της επόμενης εντολής. Τα προηγούμενα αποτελέσματα χάνονται και η μόνη δυνατότητα πρόσβασης σε αυτά είναι η επισκόπησή τους από το παράθυρο της MATLAB. Οπότε, η **ans** αποτελεί μια αυτόματη προσωρινή μεταβλητή μνήμης, στην οποία ανατίθεται το πιο πρόσφατο αποτέλεσμα εντολής.

Προκειμένου να διατηρηθούν στο χώρο εργασίας της MATLAB πολλά χρήσιμα και όχι αναγκαστικά πρόσφατα αποτελέσματα γίνεται η χρήση μεταβλητών. Η ανάθεση ενός αποτελέσματος σε μια μεταβλητή γίνεται με τη χρήση του συμβόλου της ισότητας “=”. Η γενική σύνταξη της εντολής αυτής είναι η ακόλουθη:

Όνομα μεταβλητής = τιμή ή μεταβλητή ή αποτέλεσμα πράξεων

Από τη στιγμή αυτή, η μεταβλητή διατηρεί την τιμή που της έχει ανατεθεί μέχρι τη στιγμή επανάθεσής της. Πιο συγκεκριμένα, μία μεταβλητή μπορεί να λάβει διαδοχικά πολλές τιμές, αλλά ανά πάσα στιγμή διατηρεί την τιμή από την τελευταία ανάθεσή της, όπως ακριβώς συμβαίνει και με το **ans**. Σημιώνεται ότι σε περίπτωση ανάθεσης ενός αποτελέσματος σε μία μεταβλητή η **ans** δεν ενεργοποιείται.

Το όνομα που λαμβάνει μια μεταβλητή:

- Χαρακτηρίζεται από ένα έως εξήντα τρεις (63) χαρακτήρες συνήθως αγγλικού αλφαβήτου. Ο πρώτος χαρακτήρας είναι αναγκαστικά γράμμα (κεφαλαίο ή πεζό), ενώ οι επόμενοι μπορεί να είναι γράμματα (κεφαλαία ή πεζά), αριθμοί ή κάτω παύλες (underscore). Άλλοι χαρακτήρες δεν επιτρέπονται, ενώ υπενθυμίζεται εδώ η διάκριση κεφαλαίων και πεζών γραμμάτων. Για παράδειγμα η μεταβλητή **a** δεν είναι η ίδια με την μεταβλητή **A**. Παρακάτω παρατίθενται παραδείγματα έγκυρων και μη έγκυρων μεταβλητών.

Έγκυρες μεταβλητές	Μη έγκυρες μεταβλητές
<ul style="list-style-type: none"> ✓ p ✓ h_123_first ✓ variablename1 ✓ v_ariablename1 ✓ variablename1_ ✓ Variablename1 ✓ vArlaBIEnAmE1 	<ul style="list-style-type: none"> 7p → πρώτος χαρακτήρας αριθμός! my program → περιλαμβάνει μη έγκυρο χαρακτήρα (κενό χαρακτήρα) ! k-r → περιλαμβάνει μη έγκυρο χαρακτήρα (παύλα)! _4583 → πρώτος χαρακτήρας λάθος (κάτω παύλα)!

- Δεν πρέπει να ταυτίζεται με δεσμευμένες λέξεις της MATLAB (όπως τις εντολές **log**, **clear**, **who**, **whos**, **sum**, **exp**, **sqrt**, **cos**, **tan**, **sin** κ.λ.π). Η δημιουργία μεταβλητής με κάποια δεσμευμένη λέξη είναι εφικτή. Ωστόσο, αυτό απενεργοποιεί τη λειτουργία της δεσμευμένης λέξης, αφού αυτή έχει πλέον καταχωρηθεί ως μεταβλητή.

Ας πάρουμε για παράδειγμα την εντολή εύρεσης εκθετικού αριθμού, που είδαμε και στο κεφάλαιο με τα επιτρεπτά μεγέθη της MATLAB. Για να υπολογίσουμε τον αριθμό e^5 θα πληκτρολογήσαμε την εντολή **exp(5)** στο παράθυρο εντολών της MATLAB. Παρακάτω φαίνεται και το αποτέλεσμα που θα μας επέστρεφε η MATLAB.

π.χ.

```
>> exp(5)
```

```
ans =  
148.4132
```

Στην περίπτωση που όμως δημιουργήσουμε μία μεταβλητή με τη δεσμευμένη λέξη `exp`, τότε απενεργοποιούμε τη δυνατότητα που είχαμε να υπολογίσουμε εκθετικούς αριθμούς. Παρακάτω φαίνεται η εσφαλμένη δημιουργία της μεταβλητής `exp`, η οποία περιλαμβάνει τον τυχαίο ακέραιο αριθμό δέκα (10), καθώς επίσης και η επανάληψη της εντολής `exp(5)`, η οποία όμως είναι ανεπιτυχής αυτή τη φορά, αφού η δεσμευμένη λέξη `exp` απενεργοποιήθηκε και αναγνωρίζεται πλέον ως μία απλή μεταβλητή.

π.χ.

```
>> exp=10
```

```
exp =  
10
```

```
>> exp(5)
```

```
Index exceeds matrix dimensions.
```

Οι μεταβλητές στη MATLAB μπορεί να είναι όχι μόνο πραγματικές αλλά και μιγαδικές ή αλφαριθμητικές, δηλ. να έχουν ως τιμές ακολουθίες χαρακτήρων (strings), ή ακόμα και λογικές (logical), δηλ. να παίρνουν τις τιμές `true` (αληθής) και `false` (ψευδής). Παρακάτω παρατίθενται μερικά παραδείγματα αυτών των μεταβλητών.

Εντολή	Επεξήγηση
<pre>>> x=3 x = 3</pre>	Καταχώρηση της τιμής 3 στη μεταβλητή <code>x</code> . Μεταβλητή <code>x</code> → πραγματική
<pre>>> y=-6.5 y = -6.5</pre>	Καταχώρηση της τιμής -6.5 στη μεταβλητή <code>y</code> . Μεταβλητή <code>y</code> → πραγματική
<pre>>> z_4=y z_4 = -6.5</pre>	Καταχώρηση της τιμής που περιέχει η μεταβλητή <code>y</code> στη μεταβλητή <code>z_4</code> Μεταβλητή <code>z_4</code> → πραγματική
<pre>>> A=[2 3;5 1] A = 2 3 5 1</pre>	Καταχώρηση του πίνακα $\begin{bmatrix} 2 & 3 \\ 5 & 1 \end{bmatrix}$ στη μεταβλητή <code>A</code> . Μεταβλητή <code>A</code> → πίνακας

<pre>>> k=16-8i</pre> <pre>k=</pre> <pre>16-8i</pre>	<p>Καταχώρηση της μιγαδικής τιμής 16-8i στη μεταβλητή k.</p> <p>Μεταβλητή k → μιγαδική</p>
<pre>>> learn=true</pre> <pre>learn =</pre> <pre>1</pre>	<p>Καταχώρηση της τιμής true στη μεταβλητή learn.</p> <p>Μεταβλητή learn → λογική</p>

Βέβαια σαν τιμή μιας μεταβλητής μπορεί να είναι και οποιοσδήποτε αλφαριθμητικός χαρακτήρας επιθυμεί ο χρήστης, αρκεί ο χαρακτήρας αυτός, ο οποίος βρίσκεται δεξιά του συμβόλου του (=), να βρίσκεται εντός μονών εισαγωγικών (' ').

Εντολή	Επεξήγηση
<pre>>>a='matlab'</pre> <pre>a =</pre> <pre>matlab</pre>	<p>Καταχώρηση ακολουθίας χαρακτήρων στη μεταβλητή a.</p> <p>Μεταβλητή a → αλφαριθμητική</p>
<pre>>> b='Πρώτο Θέμα'</pre> <pre>b =</pre> <pre>Πρώτο Θέμα</pre>	<p>Καταχώρηση ακολουθίας χαρακτήρων στη μεταβλητή b.</p> <p>Μεταβλητή b → αλφαριθμητική</p>

Επίσης ανά πάσα στιγμή είναι δυνατό να αλλάξει η τιμή μιας μεταβλητής, απλά πληκτρολογώντας τη νέα τιμή της μεταβλητής. Παρακάτω βλέπουμε ότι στη μεταβλητή b πραγματοποιείται δεύτερη ανάθεση με νέα τιμή. Έτσι, η μεταβλητή b έχει πλέον αλλάξει τιμή και έχει κρατήσει την τιμή από την τελευταία ανάθεση που της έχει γίνει.

Εντολή	Επεξήγηση
<pre>>>b='Δεύτερο Θέμα'</pre> <pre>b =</pre> <pre>Δεύτερο Θέμα</pre>	<p>Καταχώρηση νέας ακολουθίας χαρακτήρων στη μεταβλητή b.</p> <p>Μεταβλητή b → αλφαριθμητική</p>

! Προσοχή! Να σημειωθεί ότι πρέπει πάντα από τη δεξιά μεριά του συμβόλου "=" να υπάρχει τιμή ή μεταβλητή. Όταν μια μεταβλητή ανατεθεί σε μια άλλη, πρέπει να έχει ήδη τιμή από προηγούμενη ανάθεση.

Εντολή	Επεξήγηση
<pre>>>z=w Undefined function or variable 'w'.</pre>	<p>Καταχώρηση της τιμής που περιέχει η μεταβλητή w στη μεταβλητή z.</p> <p>Λάθος :</p> <p>Η μεταβλητή w δεν έχει τιμή από προηγούμενη ανάθεση. Οπότε η μεταβλητή z ποιά τιμή πρέπει να πάρει;</p>

Αν πραγματοποιηθεί ανάθεση σε μία νέα μεταβλητή, η οποία ενδεχόμενα κατά λάθος λάβει το όνομα μίας ήδη υπάρχουσας μεταβλητής, τότε η παλιά τιμή της μεταβλητής χάνεται.

Εντολή	Επεξήγηση
<pre>>> b=6.9 b = 6.9</pre>	<p>Καταχώρηση της τιμής 6.9 στη μεταβλητή b.</p> <p>Μεταβλητή b → πραγματική</p> <p>Η μεταβλητή b από αλφαριθμητική μετατράπηκε σε πραγματική ύστερα από τη δεύτερη ανάθεση. Ενδεχομένως κάτι τέτοιο να μη γίνει σκόπιμα, αλλά η μεταβλητή b έχει πλέον χάσει την παλιά της τιμή.</p>

5.2. Χειρισμός Μεταβλητών

5.2.1. Εντολή Deal

Μία πολύ χρήσιμη εντολή, η οποία απλουστεύει το χειρισμό των μεταβλητών είναι η εντολή deal. Με την εντολή αυτή μας δίνεται η δυνατότητα να πραγματοποιήσουμε ανάθεση πολλών ταυτόχρονα μεγεθών (αριθμοί, μονοδιάστατοι και δισδιάστατοι πίνακες, ακολουθία χαρακτήρων κτλ.) σε αντίστοιχα πολλές μεταβλητές ή ακόμα και να πραγματοποιήσουμε ανάθεση ενός μεγέθους σε πολλές ταυτόχρονα μεταβλητές. Η γενική σύνταξη αυτής της εντολής είναι:

$$[Y1, Y2, Y3, \dots] = \text{deal}(X1, X2, X3, \dots) \text{ ή } \text{deal}(X)$$

όπου Y1,Y2,Y3 κ.ο.κ. είναι οι μεταβλητές που θέλουμε να δημιουργήσουμε και X1,X2,X3 κ.ο.κ. είναι τα μεγέθη τα οποία θέλουμε να εισάγουμε σε κάθε μία από αυτές τις μεταβλητές. Η αντιστοιχία των μεγεθών προς τις μεταβλητές είναι ένα προς ένα και άρα το μέγεθος X1 εισάγεται στη μεταβλητή Y1, το μέγεθος X2 εισάγεται στη μεταβλητή Y2 κ.ο.κ. Παρακάτω ακολουθούν μερικά ενδεικτικά παραδείγματα της εντολής deal.

Παράδειγμα 1	Παράδειγμα 3
<pre>>>[A,B,C] = deal(2,5,1) A = 2</pre>	<pre>>> [A,B,C] = deal('Matlab', 'matrix', 'laboratory') A = Matla`b</pre>

<pre>B = 5 C = 1</pre>	<pre>B = Matrix C = Laboratory</pre>
Παράδειγμα 2	Παράδειγμα 4
<pre>>> [A,B,C] = deal(2,5.5,10^2) A = 2 B = 5.5000 C = 100</pre>	<pre>>> [day,Month,Year] = deal('Saturday','July',2012) day = Saturday Month = July Year = 2012</pre>

5.2.2. Εντολή Who

Με την εντολή who η MATLAB μας παραθέτει όλες τις μεταβλητές που υπάρχουν στη μνήμη του χώρου εργασίας (Workspace). Παρακάτω βλέπουμε το αποτέλεσμα που μας επιστρέφει η MATLAB ύστερα από την είσοδο της εντολή who στο παράθυρο εντολών και ύστερα από την εντολή deal, την οποία περιγράψαμε προηγουμένως.

π.χ.

```
>> who
```

Your variables are:

```
A B C Month Year day
```

5.2.3. Εντολή Whos

Η εντολή whos δε μας δίνει πρόσθετες πληροφορίες για τις μεταβλητές που υπάρχουν στη μνήμη του χώρου εργασίας. Αυτό το κενό πληροφοριών καλύπτει η εντολή whos, η οποία μαζί με τον κατάλογο των μεταβλητών του χώρου εργασίας, μας δίνει πληροφορίες για όλες τις ενεργές μεταβλητές.

π.χ.

```
>> whos
```

Name	Size	Bytes	Class	Attributes
A	1x1	8	double	
B	1x1	8	double	
C	1x1	8	double	
Month	1x4	8	char	
Year	1x1	8	double	

day 1x8 16 char

Στην περίπτωση που επιθυμούμε να πάρουμε πληροφορίες για κάποια συγκεκριμένη μεταβλητή, μπορούμε να χρησιμοποιήσουμε τις παραπάνω εντολές για μεμονωμένες μεταβλητές, όπως φαίνεται πιο κάτω:

Παράδειγμα

Για την καλύτερη κατανόηση του παραδείγματος θα εισαγάγουμε τις ακόλουθες μεταβλητές στο παράθυρο εντολών του MATLAB:

- `real = 4,23` (πραγματική μεταβλητή)
- `complex = 2 - 3i` (μιγαδική μεταβλητή)
- `vector = [3 0 2 -1]` (πραγματικό διάνυσμα)
- `matrix = [1 2; 3 4]` (δισδιάστατο πίνακα)
- `string = 'real'` (ακολουθία χαρακτήρων) (είναι το όνομα της πρώτης μεταβλητής μας) και
- `variable_log = true` (λογική μεταβλητή)

π.χ.

```
>> real=4.23;
>> comp=2 -3i;
>> vector=[3 0 2 -1];
>> matrix=[1,2;3,4];
>> string='real';
>> variable_log=true;
```

Εισάγοντας την εντολή `whos`, η MATLAB μας επιστρέφει:

```
>> whos
```

<i>Name</i>	<i>Size</i>	<i>Bytes</i>	<i>Class</i>	<i>Attributes</i>
<i>comp</i>	<i>1x1</i>	<i>16</i>	<i>double</i>	<i>complex</i>
<i>matrix</i>	<i>2x2</i>	<i>32</i>	<i>double</i>	
<i>real</i>	<i>1x1</i>	<i>8</i>	<i>double</i>	
<i>string</i>	<i>1x4</i>	<i>8</i>	<i>char</i>	
<i>variable_log</i>	<i>1x1</i>	<i>1</i>	<i>logical</i>	
<i>vector</i>	<i>1x4</i>	<i>32</i>	<i>double</i>	

Παρατηρώντας το αποτέλεσμα που μας επέστρεψε η MATLAB, βλέπουμε ότι όλες οι μεταβλητές αντιμετωπίζονται σαν πίνακες (arrays). Η πρώτη στήλη περιλαμβάνει το όνομα κάθε μεταβλητής (Name), η δεύτερη τη διάσταση του κάθε πίνακα (Size), η τρίτη το πλήθος των bytes (Bytes) και η τέταρτη την κατηγορία (Class) της μεταβλητής, δηλαδή αν η μεταβλητή είναι διπλής ακρίβειας, λογική (logical), αλφαριθμητική (πίνακας χαρακτήρων, char array), δομή δεδομένων ή πίνακας κελιών (cell). Η προεπιλογή για τους αριθμούς είναι η διπλή ακρίβεια (double precision). Τέλος, στην πέμπτη στήλη, η MATLAB μας προσδιορίζει τις ιδιότητες μιας μεταβλητής (μιγαδική (complex), παγκόσμια (global) κ.α.).

5.3. Διαγραφή Μεταβλητών

Με τη χρήση της εντολής **clear** μας δίνεται η δυνατότητα να διαγράψουμε από τη μνήμη του χώρου εργασίας όσες μεταβλητές επιθυμούμε. Η εντολή αυτή, διαγράφει το όνομα της μεταβλητής, καθώς επίσης και το περιεχόμενο της από το χώρο εργασίας της MATLAB. Στην περίπτωση που επιθυμούμε τη διαγραφή όλων των μεταβλητών που έχουμε δημιουργήσει, χρησιμοποιούμε μόνο την εντολή **clear**, ενώ στην περίπτωση που επιθυμούμε τη διαγραφή μίας μόνο συγκεκριμένης μεταβλητής, εισάγουμε δίπλα από την εντολή **clear** και το όνομα της μεταβλητής αυτής. Για τη διαγραφή δύο ή περισσότερων μεταβλητών η σύνταξη παραμένει ίδια με τη μόνη διαφορά ότι δίπλα από την εντολή **clear** δε θα υπάρχει μόνο ένα όνομα μεταβλητής, αλλά ένα πλήθος μεταβλητών οι οποίες θα χωρίζονται με κόμμα ή κενό. Μερικά ενδεικτικά παραδείγματα διαγραφής μεταβλητών είναι τα παρακάτω:

Εντολή	Επεξήγηση
<code>>> clear</code>	Διαγραφή όλων των μεταβλητών από τη μνήμη του χώρου εργασίας.
<code>>> clear Month, Year</code>	Διαγραφή μόνο των μεταβλητών <code>Month</code> , <code>Year</code> από τη μνήμη του χώρου εργασίας.
<code>>> clear real comp vector matrix</code>	Διαγραφή μόνο των μεταβλητών <code>real</code> , <code>comp</code> , <code>vector</code> , <code>matrix</code> από τη μνήμη του χώρου εργασίας.

5.4. Απόκρυψη Περιεχομένου Μεταβλητών

Πολλές φορές μας ενδιαφέρει να μην εμφανίζεται η τιμή μιας μεταβλητής ύστερα από κάποια ανάθεση. Για να το πετύχουμε αυτό, μπορούμε να εισάγουμε στο τέλος της εντολής το ελληνικό ερωτηματικό ";". Αυτό ισχύει για οποιαδήποτε εντολή εισάγουμε στο περιβάλλον της MATLAB. Εισάγοντας το ελληνικό ερωτηματικό στο τέλος των εντολών, αποτρέπουμε την εμφάνιση των αποτελεσμάτων, ενώ μας δίνεται και η δυνατότητα να πραγματοποιήσουμε επανειλημμένες αναθέσεις στην ίδια σειρά όπως φαίνεται στα παρακάτω παραδείγματα.

Εντολή	Επεξήγηση
<code>>> b=6.9;</code>	Καταχώρηση της τιμής 6.9 στη μεταβλητή <code>b</code> . Μεταβλητή <code>b</code> → πραγματική Το αποτέλεσμα της ανάθεσης δεν εμφανίζεται στο χρήστη. Ωστόσο η μεταβλητή <code>b</code> έχει δημιουργηθεί και περιέχει την τιμή που πληκτρολογήσαμε.
<code>>> time=2; sun='good'; cold='false';</code>	<ul style="list-style-type: none"> - Καταχώρηση της τιμής 2 στη μεταβλητή <code>time</code>. - Καταχώρηση της ακολουθίας χαρακτήρων <code>good</code> στη μεταβλητή <code>sun</code>. - Καταχώρηση της λογικής τιμής <code>false</code> στη μεταβλητή <code>cold</code>. - Μεταβλητή <code>time</code> → πραγματική - Μεταβλητή <code>sun</code> → αλφαριθμητική - Μεταβλητή <code>cold</code> → λογική

Στην πρώτη στήλη του παρακάτω παραδείγματος, η παρουσία του ";" αποτρέπει την εμφάνιση των περιεχομένων των μεταβλητών, ενώ στη δεύτερη στήλη, τα περιεχόμενα των μεταβλητών εμφανίζονται ύστερα από την πληκτρολόγηση της κάθε εντολής.

Εντολή με τη χρήση ";"	Εντολή χωρίς ";"
<code>>> a=2;</code>	<code>>> a=2</code> <code>a =</code> <code>2</code>
<code>>> b=5;</code>	<code>>> b=5</code> <code>b =</code> <code>5</code>
<code>>> k=4.34543;</code>	<code>>> k=4.34543</code> <code>k =</code> <code>4.3454</code>
<code>>> name='matlab';</code>	<code>>> name='matlab'</code> <code>name =</code> <code>matlab</code>

Η δυνατότητα της MATLAB να αλλάζει το είδος του μεγέθους που ανατίθεται σε μία μεταβλητή είναι πρωτοποριακή, όπως είναι και το γεγονός ότι οι μεταβλητές δε χρειάζεται να δηλωθούν εκ των προτέρων.

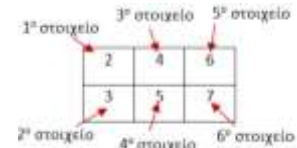
ΚΕΦΑΛΑΙΟ 6: ΔΙΑΧΕΙΡΙΣΗ ΠΙΝΑΚΩΝ ΚΑΙ ΔΙΑΝΥΣΜΑΤΩΝ

6.1. Εμφάνιση στοιχείου πίνακα

Κάθε πίνακας αποτελείται από ένα σύνολο στοιχείων που είναι τοποθετημένα σε γραμμές και στήλες. Όλα τα στοιχεία του χαρακτηρίζονται από δύο δείκτες που δηλώνουν τη γραμμή και τη στήλη στην οποία ανήκει το κάθε στοιχείο. Στην περίπτωση των διανυσμάτων (μονοδιάστατων πινάκων), κάθε στοιχείο χαρακτηρίζεται από ένα δείκτη που είτε δηλώνει τη γραμμή είτε τη στήλη. Η παραπάνω διαδικασία γενικεύεται για πίνακες μεγαλύτερων διαστάσεων.

Στο MATLAB, σε αντίθεση με άλλες γλώσσες προγραμματισμού, όλοι οι πίνακες δεικτοδοτούνται αρχίζοντας από το ένα, ανεξαρτήτων διαστάσεων. Επίσης, οι θέσεις των στοιχείων ορίζονται με παρενθέσεις. Έτσι, για το διάνυσμα $K=[5\ 4\ 3\ 2]$, το πρώτο στοιχείο του είναι το $K(1,1)=5$, ενώ για τον πίνακα $A=\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \end{bmatrix}$ το πρώτο στοιχείο του είναι το $A(1,1)=2$.

Μία επιπλέον δυνατότητα του MATLAB είναι η πρόσβαση των στοιχείων ενός διανύσματος ή πίνακα χωρίς να χρειαστεί να δηλώσουμε ακριβώς και τους δύο δείκτες του. Αν θέλουμε για παράδειγμα να εμφανίσουμε το πρώτο (1^ο), το τέταρτο (4^ο) και το τελευταίο στοιχείο του διανύσματος K πληκτρολογούμε:



```
>> K(1)      >> K(4)      >> K(end)
ans =        ans =        και  ans =
     5         2         2
```

Ενώ για το πρώτο (1^ο), το τέταρτο (4^ο) και το τελευταίο στοιχείο του πίνακα A πληκτρολογούμε:

```
>> A(1)      >> A(4)      >> A(end)
ans =        ans =        και  ans =
     2         5         7
```

Εξίσου σημαντική είναι και η δυνατότητα που μας δίνει το MATLAB, για την εμφάνιση των στοιχείων ολόκληρων των γραμμών ή στηλών ενός πίνακα. Η λειτουργία αυτή πραγματοποιείται με τη χρήση του συμβόλου ":" στη θέση των γραμμών ή στηλών που εκτυπώνονται ολόκληρες. Παρακάτω δίνονται μερικά ενδεικτικά παραδείγματα εμφάνισης ολόκληρων γραμμών ή στηλών με τη χρήση μίας μόνο εντολής:

Εντολή	Επεξήγηση
<pre>>> A(1,:) ans = 2 4 6</pre>	Εμφάνιση της πρώτης γραμμής του πίνακα και όλες τις στήλες.
<pre>>> A(:,1) ans = 2 3</pre>	Εμφάνιση της πρώτης στήλης του πίνακα και όλες τις γραμμές.
<pre>>> A(:,2) ans = 4 5</pre>	Εμφάνιση της δεύτερης στήλης του πίνακα και όλες τις γραμμές.
<pre>>> A(:,end) ans = 6 7</pre>	Εμφάνιση της τελευταίας στήλης του πίνακα και όλες τις γραμμές.
<pre>>> A(2,:) ans = 3 5 7</pre>	Εμφάνιση της δεύτερης γραμμής του πίνακα και όλες τις στήλες.

Επίσης, δηλώνοντας A(i:b, j:c) μπορούμε να απομονώσουμε τα στοιχεία του πίνακα A που ανήκουν στις γραμμές i μέχρι και b και στις στήλες j μέχρι και c.

Εντολή	Επεξήγηση
<pre>>> A(1:2,1:2) ans = 2 4 3 5</pre>	Εμφάνιση στοιχείων από την πρώτη έως τη δεύτερη γραμμή και από την πρώτη έως τη δεύτερη στήλη.
<pre>>> A(1,1:2) ans = 2 4</pre>	Εμφάνιση στοιχείων της πρώτης γραμμής, από την πρώτη έως τη δεύτερη στήλη.

6.2. Τροποποίηση στοιχείου υπάρχον πίνακα

Χάρη στη δεικτοδότηση των πινάκων, το MATLAB μας δίνει τη δυνατότητα να τροποποιήσουμε την τιμή ενός στοιχείου, καλώντας το και αποδίδοντας του νέα τιμή, με την εντολή της ανάθεσης. Η σύνταξη της εντολής είναι:

Όνομα πίνακα (αριθμός στοιχείου) = νέα τιμή

ή

Όνομα πίνακα (i, j) = νέα τιμή, όπου i,j η γραμμή και η στήλη αντίστοιχα του στοιχείου

Με αυτό τον τρόπο, για παράδειγμα, αν θέλουμε να αντικαταστήσουμε το τρίτο στοιχείο του διανύσματος, το οποίο βρίσκεται στην πρώτη γραμμή και δεύτερη στήλη του πίνακα, δίνοντάς του την τιμή 9, μπορούμε να το κάνουμε με δύο τρόπους:

Εντολή α' τρόπος	Εντολή β' τρόπος
<pre>>> A(1,2)=9 A = 2 9 6 3 5 7</pre>	ή <pre>>> A(3)=9 A = 2 9 6 3 5 7</pre>

6.3. Διαγραφή γραμμών ή στηλών πίνακα

Στο MATLAB υπάρχει η δυνατότητα διαγραφής ολόκληρων των γραμμών ή στηλών ενός πίνακα, απλά με την εντολή ανάθεσης του κενού στοιχείου στις γραμμές ή στήλες που επιθυμούμε να διαγράψουμε. Έστω, για παράδειγμα, ότι έπρεπε να διαγράψουμε κάποια

στοιχεία του πίνακα $A = \begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \\ 5 & 46 & 57 \end{bmatrix}$. Αυτό γίνεται με τις παρακάτω εντολές:

Εντολή	Επεξήγηση
<pre>>> A(1,:)=[]</pre> <p>A =</p> <pre> 3 5 7 4 6 78 5 46 57</pre>	Διαγραφή όλων των στοιχείων της πρώτης γραμμής.
<pre>>> A(2:3,:)=[]</pre> <p>A =</p> <pre> 2 4 6 5 46 57</pre>	Διαγραφή όλων των στοιχείων από τη δεύτερη έως και την τρίτη γραμμή.
<pre>>> A(:,2)=[]</pre> <p>A =</p> <pre> 2 6 3 7 4 78 5 57</pre>	Διαγραφή όλων των στοιχείων της δεύτερης στήλης.
<pre>>> A(:,2:3)=[]</pre> <p>A =</p> <pre> 2 3 4 5</pre>	Διαγραφή όλων των στοιχείων από τη δεύτερη έως και την τρίτη στήλη.

6.4. Πράξεις πινάκων

Μεταξύ των πινάκων είναι δυνατό να υλοποιηθούν συνηθισμένες πράξεις αριθμητικής όπως η πρόσθεση, η αφαίρεση, ο πολλαπλασιασμός, η διαίρεση και η ύψωση σε δύναμη. Στον παρακάτω πίνακα παρουσιάζονται οι πράξεις αριθμητικής μεταξύ δύο πινάκων καθώς επίσης και οι περιορισμοί που πρέπει να ισχύουν σε κάθε μία από αυτές:

Αριθμητική πράξη	Αριθμητική παράσταση	Περιορισμοί
+	A+B	A και B ίδιων διαστάσεων
-	A-B	A και B ίδιων διαστάσεων
/ ή \	A/B ή A\B	A και B ίδιων διαστάσεων και τετραγωνικοί

*	A*B	Ο αριθμός των στηλών του A ίσος με τον αριθμό των γραμμών του B
^	A ^{δύναμη}	A τετραγωνικό

Έστω ο πίνακας $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ και ο πίνακας $B = \begin{bmatrix} 10 & 20 & 30 \\ 400 & 560 & 680 \\ 67 & 48 & 49 \end{bmatrix}$, οι οποίοι εισάγονται στο MATLAB με τις εξής εντολές:

```
>> A=[1 2 3 ; 4 5 6 ; 7 8 9]
```

```
>> B=[10 20 30 ; 400 560 680 ; 67 48 49]
```

Παρακάτω δίνονται τα αποτελέσματα των αριθμητικών πράξεων μεταξύ των δύο πινάκων. Για την καλύτερη κατανόηση των αποτελεσμάτων παρατίθενται στη δεξιά στήλη οι αντίστοιχοι τύποι υπολογισμού.

Αριθμητική πράξη	Τύπος υπολογισμού
<pre>>> A+B</pre> <pre>ans =</pre> <pre> 11 22 33</pre> <pre> 404 565 686</pre> <pre> 74 56 58</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{pmatrix}$
<pre>>> A+5</pre> <pre>ans =</pre> <pre> 6 7 8</pre> <pre> 9 10 11</pre> <pre> 12 13 14</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} + c = \begin{pmatrix} a_{11} + c & a_{12} + c \\ a_{21} + c & a_{22} + c \end{pmatrix}$
<pre>>> A-B</pre> <pre>ans =</pre> <pre> -9 -18 -27</pre> <pre>-396 -555 -674</pre> <pre> -60 -40 -40</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} - \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11} - b_{11} & a_{12} - b_{12} \\ a_{21} - b_{21} & a_{22} - b_{22} \end{pmatrix}$
<pre>>> A-5</pre> <pre>ans =</pre> <pre> -4 -3 -2</pre> <pre> -1 0 1</pre> <pre> 2 3 4</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} - c = \begin{pmatrix} a_{11} - c & a_{12} - c \\ a_{21} - c & a_{22} - c \end{pmatrix}$
<pre>>> A*B</pre> <pre>ans =</pre> <pre> 1011 1284 1537</pre> <pre> 2442 3168 3814</pre> <pre> 3873 5052 6091</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix}$

<pre>>> A*10 ans = 10 20 30 40 50 60 70 80 90</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} * c = \begin{pmatrix} a_{11} * c & a_{12} * c \\ a_{21} * c & a_{22} * c \end{pmatrix}$
<pre>>> A/B ans = 0.1000 -0.0000 0.0000 0.0112 0.0073 0.0146 -0.0777 0.0146 0.0291</pre>	$A/B = A*B^{-1}$
<pre>>> B\A ans = 0.3126 0.4090 0.5053 -0.6830 -1.0427 -1.4024 0.3845 0.6255 0.8665</pre>	$B\A = B^{-1}*A$
<pre>>> A/2 ans = 0.5000 1.0000 1.5000 2.0000 2.5000 3.0000 3.5000 4.0000 4.5000</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} / c = \begin{pmatrix} a_{11} / c & a_{12} / c \\ a_{21} / c & a_{22} / c \end{pmatrix}$
<pre>>> A^2 ans = 30 36 42 66 81 96 102 126 150</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} a_{11}a_{11}+a_{12}a_{21} & a_{11}a_{12}+a_{12}a_{22} \\ a_{21}a_{11}+a_{22}a_{21} & a_{21}a_{12}+a_{22}a_{22} \end{pmatrix}$

Εκτός από τις αριθμητικές πράξεις μεταξύ πινάκων, είναι δυνατές οι πράξεις πινάκων στοιχείο προς στοιχείο. Για την επίτευξη των πράξεων είναι απαραίτητη η εισαγωγή της τελείας πριν από το σύμβολο οποιασδήποτε πράξης. Με αυτό τον τρόπο δε γίνεται η πράξη μεταξύ των πινάκων αλλά μεταξύ των στοιχείων τους. Στον παρακάτω πίνακα βλέπουμε τα ίδια παραδείγματα που χρησιμοποιήθηκαν στις αριθμητικές πράξεις μεταξύ πινάκων, με μόνη διαφορά την εισαγωγή της τελείας για τη δήλωση των πράξεων στοιχείο προς στοιχείο.

Υπενθυμίζονται οι πίνακες: πίνακας $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, πίνακας $B = \begin{bmatrix} 10 & 20 & 30 \\ 400 & 560 & 680 \\ 67 & 48 & 49 \end{bmatrix}$.

Αριθμητική πράξη	Τύπος υπολογισμού
<pre>>> A.*B ans = 10 40 90 1600 2800 4080 469 384 441</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{pmatrix}$

<pre>>> A./B ans = 0.1000 0.1000 0.1000 0.0100 0.0089 0.0088 0.1045 0.1667 0.1837</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}/b_{11} & a_{12}/b_{12} \\ a_{21}/b_{21} & a_{22}/b_{22} \end{pmatrix}$
<pre>>> B.\A ans = 0.1000 0.1000 0.1000 0.0100 0.0089 0.0088 0.1045 0.1667 0.1837</pre>	<p>ίδια με τη δεξιά διαίρεση κατά στοιχείο</p>
<pre>>> A.^2 ans = 1 4 9 16 25 36 49 64 81</pre>	$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}^V = \begin{pmatrix} a_{11}^V & a_{12}^V \\ a_{21}^V & a_{22}^V \end{pmatrix}$

6.5. Ανάστροφος πίνακας

Η δυνατότητα που προσφέρει το MATLAB για τη δημιουργία ανάστροφου πίνακα βρίσκει έδαφος σε αρκετές εφαρμογές. Η δημιουργία ανάστροφου πίνακα στο MATLAB γίνεται με τη χρήση του συμβόλου “'” ή “.’” αμέσως μετά τη δήλωση του πίνακα (χωρίς ενδιάμεσα κενά). Παρακάτω βλέπουμε τρία παραδείγματα δημιουργίας ανάστροφου πίνακα.

Διάνυσμα στήλης	Ανάστροφος πίνακας
<pre>>> A = [6;7;8;9;10] A = 6 7 8 9 10</pre>	<pre>>> A = [6;7;8;9;10]' A = 6 7 8 9 10</pre>
<pre>>> A = [6,7,8,9,10] A = 6 7 8 9 10</pre>	<pre>>> A = [6,7,8,9,10]' A = 6 7 8 9 10</pre>

<pre>>> A=[2 3;5 1;6 8] A = 2 3 5 1 6 8</pre>	<pre>>> A=[2 3;5 1;6 8].'</pre> <pre>A = 2 5 6 3 1 8</pre>
---	--

6.6. Ενσωματωμένες συναρτήσεις δημιουργίας πινάκων

Καθώς η διαδικασία δημιουργίας ενός πίνακα είναι χρονοβόρα, όταν πρόκειται για μεγάλο μέγεθος πίνακες, το MATLAB διαθέτει πολλές ενσωματωμένες συναρτήσεις, οι οποίες εφαρμόζονται σε αυτούς, είτε για τη δημιουργία τους είτε για την επεξεργασία τους.

6.6.1. Συνάρτηση `zeros(m,n)`

Η συνάρτηση `zeros` δημιουργεί ένα πίνακα διαστάσεων $m \times n$, όπου όλα τα στοιχεία του είναι ίσα με το μηδέν. Ο τετραγωνικός πίνακας `zeros(n, n)` δημιουργείται και με την εντολή `zeros(n)`.

Παράδειγμα 1	Παράδειγμα 2
<pre>>> zeros(2,3) ans = 0 0 0 0 0 0</pre>	<pre>>> zeros(3) ans = 0 0 0 0 0 0 0 0 0</pre>

6.6.2. Συνάρτηση `ones(m,n)`

Η συνάρτηση `ones` δημιουργεί ένα πίνακα διαστάσεων $m \times n$, όπου όλα τα στοιχεία του είναι ίσα με τη μονάδα. Όπως ακριβώς και στη συνάρτηση `zeros`, ο τετραγωνικός πίνακας `ones(n,n)` δημιουργείται και με την εντολή `ones(n)`.

Παράδειγμα 1	Παράδειγμα 2
<pre>>> ones(2,3) ans = 1 1 1 1 1 1</pre>	<pre>>> ones(3) ans = 1 1 1 1 1 1 1 1 1</pre>

6.6.3. Συνάρτηση `eye(m,n)`

Η συνάρτηση **eye** δημιουργεί ένα μοναδιαίο πίνακα διαστάσεων $m \times n$. Όλα τα στοιχεία της κύριας διαγωνίου είναι ίσα με τη μονάδα, ενώ τα υπόλοιπα είναι ίσα με το μηδέν. Η δημιουργία ενός τετραγωνικού μοναδιαίου πίνακα πραγματοποιείται με την εντολή **eye(n)**.

Παράδειγμα 1	Παράδειγμα 2
<pre>>> eye(2,3) ans = 1 0 0 0 1 0</pre>	<pre>>> eye(3) ans = 1 0 0 0 1 0 0 0 1</pre>

6.6.4. Συνάρτηση `rand(m,n)`

Η συνάρτηση **rand** δημιουργεί ένα $m \times n$ πίνακα με τυχαίους αριθμούς ως στοιχεία, ομοιόμορφα κατανομημένοι στο διάστημα [0-1]. Ο τετραγωνικός πίνακας **rand(n,n)** δημιουργείται και με την εντολή **rand(n)**.

Παράδειγμα 1	Παράδειγμα 2
<pre>>> rand(2,3) ans = 0.8147 0.1270 0.6324 0.9058 0.9134 0.0975</pre>	<pre>>> rand(3) ans = 0.2785 0.9649 0.9572 0.5469 0.1576 0.4854 0.9575 0.9706 0.8003</pre>

6.6.5. Συνάρτηση `randn(m,n)`

Η συνάρτηση **randn** δημιουργεί ένα $m \times n$ πίνακα με τυχαίους αριθμούς ως στοιχεία από κανονική κατανομή ($\mu=0, \sigma^2=1$). Ο τετραγωνικός πίνακας **randn (n,n)** δημιουργείται και με την εντολή **randn (n)**.

Παράδειγμα 1	Παράδειγμα 2
<pre>>> randn(3) ans = 2.7694 0.7254 -0.2050 -1.3499 -0.0631 -0.1241 3.0349 0.7147 1.4897</pre>	<pre>>> randn(2,3) ans = 1.4090 0.6715 0.7172 1.4172 -1.2075 1.6302</pre>

6.6.6. Συνάρτηση *magic(m,n)*

Η συνάρτηση **magic** δημιουργεί ένα τετραγωνικό πίνακα διαστάσεων $n \times n$, ο οποίος έχει την ιδιότητα ότι το άθροισμα των στοιχείων μιας οποιασδήποτε σειράς, στήλης ή των διαγωνίων του είναι το ίδιο.

Παράδειγμα 1	Παράδειγμα 2
<pre>>> magic(3) ans = 8 1 6 3 5 7 4 9 2</pre>	<pre>>> magic(4) ans = 16 2 3 13 5 11 10 8 9 7 6 12 4 14 15 1</pre>

6.7. Ενσωματωμένες συναρτήσεις επεξεργασίας πινάκων

6.7.1. Συνάρτηση *size(table)*

Η συνάρτηση **size** προσδιορίζει τις διαστάσεις ενός πίνακα, δηλαδή προσδιορίζει τις τιμές των στοιχείων m και n που τον περιγράφουν. Στα παρακάτω παραδείγματα βλέπουμε τις

διαστάσεις των πινάκων $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ και $B = \begin{bmatrix} 10 & 20 \\ 30 & 40 \\ 50 & 60 \\ 70 & 80 \end{bmatrix}$ αντίστοιχα. Όπως φαίνεται και στα

παραδείγματα ο πίνακας A αποτελείται από τρεις γραμμές και τρεις στήλες, ενώ ο πίνακας B αποτελείται από τέσσερις γραμμές και δύο στήλες.

Παράδειγμα 1	Παράδειγμα 2
<pre>>> size(A) ans = 3 3</pre>	<pre>>> size(B) ans = 4 2</pre>

Η εντολή **size(table,n)** επιστρέφει το μέγεθος της n -οστής διάστασης του πίνακα. Έτσι, όπως φαίνεται και στο παρακάτω παράδειγμα, για τον πίνακα B , ο οποίος είναι δύο διαστάσεων, η εντολή **size(B,1)** δίνει το πλήθος των γραμμών και η εντολή **size(B,2)** δίνει το πλήθος των στηλών.

Παράδειγμα 3	Παράδειγμα 4
<pre>>> size(B,1) ans = 4</pre>	<pre>>> size(B,2) ans = 2</pre>

6.7.2. Συνάρτηση `length(table)`

Η συνάρτηση **length** προσδιορίζει το μήκος ενός πίνακα. Στο MATLAB, ως μήκος ορίζεται η μεγαλύτερη τιμή από τις διαστάσεις του πίνακα, δηλαδή η μεγαλύτερη τιμή από τα στοιχεία m και n . Οπότε, η συνάρτηση **length** επιστρέφει τη μεγαλύτερη τιμή από τις διαστάσεις του πίνακα. Στα παραδείγματα χρησιμοποιούνται οι πίνακες A και B που δηλώθηκαν και στη συνάρτηση `size`.

Παράδειγμα 1	Παράδειγμα 2
<pre>>> length(A) ans = 3</pre>	<pre>>> length(B) ans = 4</pre>

6.7.3. Συνάρτηση `numel(table)`

Η συνάρτηση **numel** προσδιορίζει το πλήθος των στοιχείων ενός πίνακα. Υπενθυμίζεται ότι χρησιμοποιούνται οι πίνακες A και B που δηλώθηκαν στα παραπάνω παραδείγματα.

Παράδειγμα 1	Παράδειγμα 2
<pre>>> numel(A) ans = 9</pre>	<pre>>> numel(B) ans = 8</pre>

6.7.4. Συνάρτηση `max(table)` και `min(table)`

Η συνάρτηση **max** επιστρέφει τη μέγιστη τιμή από κάθε μία στήλη του πίνακα, ενώ η συνάρτηση **min** επιστρέφει την ελάχιστη τιμή από κάθε μία στήλη του πίνακα. Στο παρακάτω παράδειγμα χρησιμοποιούνται και πάλι οι πίνακες A και B που δηλώθηκαν παραπάνω.

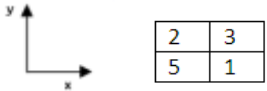
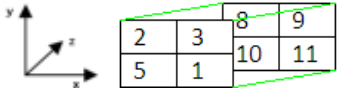
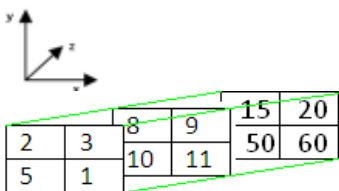
Παράδειγμα 1	Παράδειγμα 2
<pre>>> max(A) ans = 7 8 9</pre>	<pre>>> max(B) ans = 70 80</pre>
<pre>>> min(A) ans = 1 2 3</pre>	<pre>>> min(B) ans = 10 20</pre>

ΚΕΦΑΛΑΙΟ 7: ΕΠΙΤΡΕΠΤΑ ΜΕΓΕΘΗ (ΣΥΝΕΧΕΙΑ ΚΕΦΑΛΑΙΟΥ 3)

Όπως αναφέρθηκε και στο κεφάλαιο 3, είναι δυνατό να οριστούν τρία επιτρεπτά μεγέθη στη MATLAB μέσω μεταβλητών

7.1. ΠΟΛΥΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ

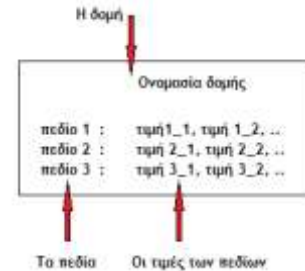
Η δημιουργία πολυδιάστατων πινάκων πραγματοποιείται σταδιακά επεκτείνοντας κάθε φορά τον πίνακα μικρότερων διαστάσεων που επιθυμούμε. Αυτή η δυνατότητα σταδιακής δημιουργίας πολυδιάστατων πινάκων οφείλεται στη δυναμική μνήμη της MATLAB κατά την οποία οι διαστάσεις των μεταβλητών μεταβάλλονται αυτόματα. Παρακάτω παρουσιάζονται αναλυτικότερα ορισμένα παραδείγματα σταδιακής δημιουργίας πολυδιάστατων πινάκων.

Εντολή	Επεξήγηση
<pre>>> A=[2 3 ; 5 1] A = 2 3 5 1</pre>	<p>Δημιουργία δισδιάστατου 2x2 πίνακα</p> $A = \begin{bmatrix} 2 & 3 \\ 5 & 1 \end{bmatrix}$ 
<pre>>> A(:, :, 2)=[8 9 ; 10 11] A(:, :, 1) = 2 3 5 1 A(:, :, 2) = 8 9 10 11</pre>	<p>Επέκταση δισδιάστατου πίνακα σε τρισδιάστατο 2x2x2 πίνακα, προσθέτοντας τα στοιχεία μόνο της τελευταίας διάστασης και αφήνοντας όλα τα άλλα όπως ήταν.</p> 
<pre>>> A(:, :, 3)=[15 20 ; 50 60] A(:, :, 1) = 2 3 5 1 A(:, :, 2) = 8 9 10 11 A(:, :, 3) = 15 20 50 60</pre>	<p>Επέκταση τρισδιάστατου πίνακα σε 2x2x3 πίνακα, προσθέτοντας τα στοιχεία μόνο της τελευταίας διάστασης και αφήνοντας όλα τα άλλα όπως ήταν.</p> 

Παρόλο που παρέχεται η δυνατότητα από τη MATLAB δημιουργίας πολυδιάστατων πινάκων και επέκτασης των διαστάσεών τους, δε συνίσταται η επέκτασή τους σε πολλές διαστάσεις, διότι στη MATLAB, όπως και σε άλλες γλώσσες προγραμματισμού, δεν είναι δυνατή η ολόκληρη εμφάνισή τους στην οθόνη. Αυτό, έχει ως αποτέλεσμα τη δυσκολία χειρισμού τους και έτσι είναι προτιμότερη η δημιουργία πολλών δισδιάστατων πινάκων, οι οποίοι συνθέτουν έναν πολυδιάστατο πίνακα.

7.2. ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ (STRUCTURES)

Στην περίπτωση όπου κρίνεται απαραίτητη η αποδοτική αποθήκευση και οργάνωση των δεδομένων για την καλύτερη διαχείρισή τους, οι δομές δεδομένων είναι προτιμότερες από τους πίνακες. Αποτελούνται από πεδία (fields), καθένα από τα οποία λαμβάνει τιμές.



Η γενική σύνταξη μιας δομής δεδομένων είναι η εξής :

Όνομα δομής = struct('πεδίο1', {'τιμή1_1', 'τιμή1_2', 'τιμή1_3'}, 'πεδίο2', {'τιμή2_1', 'τιμή2_2', 'τιμή2_3'})

Ωστόσο είναι δυνατή και η τμηματική σύνταξη μιας δομής δεδομένων:

Όνομα δομής(1).πεδίο1='τιμή1_1'
 Όνομα δομής(1).πεδίο2='τιμή2_1'
 Όνομα δομής(1).πεδίο3='τιμή3_1' ...

} Η πρώτη εγγραφή με τα πεδία και τις τιμές τους

Όνομα δομής(2).πεδίο1='τιμή1_2'
 Όνομα δομής(2).πεδίο2='τιμή2_2'
 Όνομα δομής(2).πεδίο3='τιμή3_2' ...

} Η δεύτερη εγγραφή με τα πεδία και τις τιμές τους

Έστω ότι για τη διεκπεραίωση μιας διαδικασίας είναι απαραίτητη η αποθήκευση του ονοματεπώνυμου, της ηλικίας και του τηλεφώνου μιας ομάδας φοιτητών. Στο MATLAB ο καλύτερος τρόπος αποθήκευσης αυτών των δεδομένων με σκοπό την εύκολη και γρήγορη διαχείρισή τους είναι η δομή δεδομένων. Στο παρακάτω παράδειγμα βλέπουμε τη χρήση μιας δομής δεδομένων για την αποθήκευση ενδεικτικά έξι φοιτητών και με τους δύο μεθόδους σύνταξης.

Παράδειγμα γενικής σύνταξης εισαγωγής δομής δεδομένων:

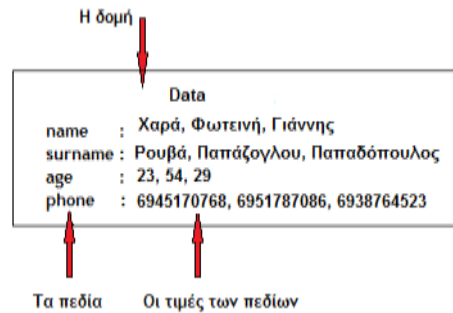
```
>>Data=struct('name',{'Χαρά','Φωτεινή','Γιάννης','Ισμήνη','Σάκης','Μαρία'},'surname',{'Ρο  

  υβά','Παπάζογλου','Παπαδόπουλος','Βασιλάκη','Μαρτίνης','Κολλιοτάση'},'age',{'23','54',  

  '29','27','53','24'},'phone',{'6945170768','6951787086','6938764523','6957149036','69772115  

  87','6984668775'})
```

Data =
1x6 struct array with fields:
name
surname
age
phone



Παράδειγμα τμηματικής σύνταξης εισαγωγής δομής δεδομένων:

```
>>Data (1).name= 'Χαρά'; ή Data.name='Χαρά';
>>Data (1).surname='Ρουβά'; ή Data.surname='Ρουβά';
>>Data (1).age='23'; ή Data.age='23';
>>Data (1).phone ='6945170768'; ή Data.phone ='6945170768';
>>Data (2).name='Φωτεινή';
>>Data (2).surname='Παπάζογλου';
>>Data (2).age='54';
>>Data (2).phone ='6951787086'; ...
```

7.2.1. ΕΜΦΑΝΙΣΗ ΕΓΓΡΑΦΗΣ

Η εμφάνιση όλων των πεδίων μιας εγγραφής πραγματοποιείται πληκτρολογώντας, στο παράθυρο εντολών του MATLAB, το όνομα της δομής δεδομένων και τον αριθμό της εγγραφής σε παρένθεση. Έτσι, για να δούμε την πρώτη εγγραφή της δομής δεδομένων Data, πληκτρολογούμε:

```
>> Data(1)

ans =
    name: 'Χαρά'
  surname: 'Ρουβά'
    age: '23'
   phone: '6945170768'
```

Για να λάβουμε το περιεχόμενο ενός μόνο συγκεκριμένου πεδίου εγγραφής πληκτρολογούμε, στο παράθυρο εντολών του, το όνομα της δομής δεδομένων, τον αριθμό της εγγραφής σε παρένθεση και τέλος μία τελεία ακολουθούμενη από το όνομα του πεδίου της δομής. Έστω ότι αναζητούμε το περιεχόμενο του πεδίου name της τρίτης εγγραφής:

```
>> Data(3).name           ή           >> getfield(Data(3),'name')

ans =                      ans =
  Γιάννης                  Γιάννης
```

7.2.2. ΕΜΦΑΝΙΣΗ ΤΙΜΩΝ ΕΝΟΣ ΠΕΔΙΟΥ

Για να λάβουμε το περιεχόμενο των τιμών όλων των εγγραφών ενός πεδίου πληκτρολογούμε, στο παράθυρο εντολών του MATLAB, το όνομα της δομής και το όνομα του πεδίου χωρισμένα με μία τελεία. Έτσι, για να λάβουμε το περιεχόμενο όλων των εγγραφών του πεδίου `surname` πληκτρολογούμε:

```
>> Data.surname
```

```
ans =
```

```
Ρουβά
```

```
ans =
```

```
Παπάζογλου
```

```
ans =
```

```
Παπαδόπουλος
```

```
ans =
```

```
Βασιλάκη
```

```
ans =
```

```
Μαρτίνης
```

```
ans =
```

```
Κολλι�ιάση
```

7.2.3. ΠΡΟΣΘΗΚΗ ΝΕΑΣ ΕΓΓΡΑΦΗΣ

Η προσθήκη νέας εγγραφής μπορεί να γίνει ανά πάσα στιγμή με ανάθεση σε ένα διαθέσιμο δείκτη. Για το παραπάνω παράδειγμα, εφόσον πραγματοποιήθηκαν εγγραφές έξι φοιτητών, η προσθήκη νέου φοιτητή θα γίνει στο `Data(7)`. Παρακάτω βλέπουμε την εισαγωγή του ονόματος μιας καινούριας φοιτήτριας στο `Data(7)`.

```
>> Data(7).name='Ρούλα'
```

```
>> setfield(Data(7),'name','Ρούλα')
```

```
Data =
```

```
1x7 struct array with fields:
```

```
name
surname
age
phone
```

```
ή
```

```
ans =
```

```
name: 'Ρούλα'
surname: []
age: []
phone: []
gender: []
```

Τα υπόλοιπα πεδία, για την έβδομη εγγραφή, θα παραμείνουν κενά μέχρι την ανάθεσή τους. Έτσι, τα πεδία `surname`, `age` και `phone` εμφανίζονται ως εξής στο παράθυρο εντολών του MATLAB ύστερα από την εντολή `>> Data(7)`, η οποία εμφανίζει όλα τα στοιχεία της έβδομης εγγραφής:

```
ans =  
    name: 'Ρούλα'  
  surname: []  
    age: []  
   phone: []
```

Η πιθανή εισαγωγή μιας εγγραφής σε ένα μεγαλύτερο δείκτη της δομής δεδομένων, για παράδειγμα στο `Data(10)`, δε θα δημιουργούσε κανένα απολύτως πρόβλημα. Η εγγραφή θα ολοκληρωνόταν επιτυχώς, δημιουργώντας τα `Data(7)`, `Data(8)`, `Data(9)` και `Data(10)`, αφήνοντας ωστόσο τα `Data(7)`, `Data(8)` και `Data(9)` να είναι κενά όπως φαίνεται στο παρακάτω παράδειγμα:

```
>> Data(10).surname='Μακρυγιάννης'
```

```
Data =  
1x10 struct array with fields:  
    name  
  surname  
    age  
   phone
```

```
>> Data(8)
```

```
ans =  
    name: []  
  surname: []  
    age: []  
   phone: []
```

```
>> Data(9)
```

```
ans =  
    name: []  
  surname: []  
    age: []  
   phone: []
```

```
>> Data(10)
```

```
ans =  
    name: []  
  surname: 'Μακρυγιάννης'  
    age: []  
   phone: []
```


7.2.4. Προσθήκη νέων πεδίων σε υπάρχουσα Δομή Δεδομένων

Είναι δυνατή η εισαγωγή νέων πεδίων σε υπάρχουσα δομή δεδομένων. Η εισαγωγή του νέου πεδίου μπορεί να πραγματοποιηθεί ανά πάσα στιγμή με ανάθεση τιμής του καινούριου πεδίου σε οποιαδήποτε υπάρχουσα εγγραφή. Έστω ότι για το παραπάνω παράδειγμα κρινόταν απαραίτητη η εισαγωγή του γένους των φοιτητών και άρα η εισαγωγή του πεδίου 'gender' στη δομή δεδομένων. Η προσθήκη του νέου πεδίου μπορεί να γίνει σε οποιαδήποτε εγγραφή επιλέξουμε. Στο παράδειγμα, το νέο πεδίο θα οριστεί στην πρώτη εγγραφή και αυτόματα θα εισαχθεί με κενές τιμές (μέχρι ανάθεσής της) και στις υπόλοιπες εγγραφές της δομής δεδομένων.

```
>> Data(1).gender='Θήλυ'
```

```
Data =
    1x10 struct array with fields:
        name
        surname
        age
        phone
        gender
```

```
>> Data(1).gender
```

```
ans =
    Θήλυ
```

```
>> Data(2).gender
```

```
ans =
    []
```

```
>> Data(3).gender
```

```
ans =
    []
```

```
>> Data(8).gender
```

```
ans =
    []
```

Το γένος των υπόλοιπων εγγραφών δεν έχει οριστεί και άρα είναι κενό!!!

7.2.5. Χειρισμός Δομής Δεδομένων

Για το χειρισμό των δομών δεδομένων υπάρχουν τρεις βασικές εντολές, οι οποίες επεξηγούνται παρακάτω.

ENTOLH fieldnames

Με την εντολή fieldnames εμφανίζουμε όλα τα ονόματα των πεδίων της δομής δεδομένων που έχουμε δημιουργήσει. Η σύνταξη της εντολής είναι:

```
fieldnames(όνομα δομής δεδομένων)
```

Έτσι, όταν πληκτρολογήσουμε την εντολή `>> fieldnames(Data)` η MATLAB μας εμφανίζει:

```
ans =
    'name'
    'surname'
    'age'
    'phone'
    'gender'
```

ΕΝΤΟΛΗ isfield

Η εντολή `isfield` επιστρέφει δύο τιμές στο χρήστη. Εάν το πεδίο, που έχει εισαχθεί στην εντολή ως παράμετρος, αποτελεί πεδίο της δομής δεδομένων, τότε επιστρέφει την τιμή ένα (1), αλλιώς επιστρέφει την τιμή μηδέν (0). Η σύνταξη της εντολής είναι:

isfield(όνομα δομής δεδομένων,'όνομα πεδίου')

Παρακάτω ακολουθούν μερικά ενδεικτικά παραδείγματα της εντολής `isfield`:

Εντολή	Επεξήγηση
<pre>>> isfield(Data,'name') ans = 1</pre>	<p>Η μεταβλητή <code>ans</code> περιέχει την τιμή 1, αφού το πεδίο <code>name</code> είναι έγκυρο πεδίο της δομής δεδομένων <code>Data</code>.</p>
<pre>>> isfield(Data,'Name') ans = 0</pre>	<p>Η μεταβλητή <code>ans</code> περιέχει την τιμή 0, αφού το πεδίο <code>Name</code> δεν είναι πεδίο της δομής δεδομένων <code>Data</code>! Η MATLAB, όπως αναφέραμε και σε προηγούμενα κεφάλαια κάνει διάκριση μεταξύ των πεζών και κεφαλαίων γραμμάτων με αποτέλεσμα το πεδίο <code>name</code> να είναι διαφορετικό από το πεδίο <code>Name</code>!</p>
<pre>>> isfield(Data,'gender') ans = 1</pre>	<p>Η μεταβλητή <code>ans</code> περιέχει την τιμή 1, αφού το πεδίο <code>gender</code> είναι έγκυρο πεδίο της δομής δεδομένων <code>Data</code>.</p>
<pre>>> isfield(Data,'day') ans = 0</pre>	<p>Η μεταβλητή <code>ans</code> περιέχει την τιμή 0, αφού το πεδίο <code>day</code> δεν είναι πεδίο της δομής δεδομένων <code>Data</code>!</p>

! Σημειώνεται πως η σύνταξη της παραπάνω εντολής, αλλά και όλων των εντολών της MATLAB, χρειάζεται ιδιαίτερη προσοχή καθώς μια μικρή διαφοροποίηση μπορεί να προκαλέσει προβλήματα κατά της εκτέλεσής της, όπως φαίνεται στα παρακάτω παραδείγματα:

Εντολή	Επεξήγηση
<pre>>> isfield(Data.'name') isfield(Data.'name') Error: Unexpected MATLAB expression.</pre>	<p>Λάθος: Στην εντολή έχει χρησιμοποιηθεί η τελεία και όχι η υποδιαστολή, όπως ορίζει η σύνταξή της, για το διαχωρισμό του ονόματος της δομής δεδομένων και του ονόματος του πεδίου .</p>
<pre>>> isfield(Data,name) Undefined function or variable 'name'.</pre>	<p>Λάθος: Το όνομα του πεδίου δε περικλείεται σε μονά εισαγωγικά και έτσι το MATLAB δε μπορεί να εκτελέσει επιτυχώς την εντολή.</p>

ΕΝΤΟΛΗ rmfield

Η εντολή rmfield αφαιρεί ένα συγκεκριμένο πεδίο από μία ήδη υπάρχουσα δομή δεδομένων. Η γενική σύνταξη της εντολής είναι:

rmfield(όνομα δομής δεδομένων,'όνομα πεδίου')

Η εντολή rmfield χρησιμοποιείται επίσης για τη δημιουργία μίας νέας δομής δεδομένων χρησιμοποιώντας μία ήδη υπάρχουσα δομή δεδομένων, αφαιρώντας ένα συγκεκριμένο πεδίο. Η σύνταξη της εντολής σε αυτή την περίπτωση γίνεται:

νέα δομή δεδομένων = rmfield(όνομα δομής δεδομένων,'όνομα πεδίου')

Παρακάτω βλέπουμε την εκτέλεση της εντολής rmfield, η οποία δημιουργεί μιας νέας δομής δεδομένων με όνομα Domi_2 χωρίς το πεδίο gender:

```
>> Domi_2 = rmfield(Data,'gender')
```

```
Domi_2 =
```

```
1x10 struct array with fields:
```

```
name
surname
age
phone
```

7.3. Πίνακες κελιών (Cell Arrays)

Εκτός από τις δομές δεδομένων (structures), το MATLAB προσφέρει και έναν ιδιαίτερο τύπο μεταβλητής που λέγεται πίνακας κελιών (cell array). Πρόκειται για μία συλλογή

μεταβλητών οργανωμένων σε μορφή πινάκων, των οποίων τα στοιχεία ονομάζονται «κελιά» και με τη διαφορά ότι επιτρέπεται η ανομοιογένεια (όπως στις δομές). Έτσι, η διαφορά τους από τους πίνακες, είναι ότι περιέχουν μεταβλητές διαφορετικού τύπου, ενώ η διαφορά τους από τις δομές είναι ότι οι επιμέρους μεταβλητές των κελιών χαρακτηρίζονται από δείκτες όπως ακριβώς στους πίνακες και όχι από ονόματα πεδίων, όπως στις δομές. Συνεπώς, οι πίνακες κελιών συνδυάζουν την ευελιξία των διαφορετικών τύπων μεταβλητών και την ευκολία προσπέλασης και σάρωσης των κελιών τους με τη χρήση των δεικτών.

Τα κελιά μπορεί να περιέχουν οποιοδήποτε τύπο μεταβλητής, όπως απλές μεταβλητές, αριθμούς, διανύσματα, πίνακες, ακολουθίες χαρακτήρων, δομές ή ακόμα και άλλους πίνακες κελιών. Η απλούστερη και συνήθης χρήση αυτών των πινάκων είναι ως πίνακες αποθήκευσης ακολουθιών χαρακτήρων.

7.3.1. ΑΝΑΘΕΣΗ τιμών σε Πίνακα Κελιών

Η ανάθεση των τιμών σε κελιά, αλλά και η αναφορά σ' αυτά γίνεται με χρήση των αγκίστρων "{ }" και όχι των αγκυλών όπως τους πίνακες. Διατάσσονται σε γραμμές και στήλες με τον ίδιο τρόπο που γίνεται και στους πίνακες. Τα στοιχεία μιας γραμμής χωρίζονται μεταξύ τους με κενό ή κόμμα, ενώ οι γραμμές χωρίζονται με το ελληνικό ερωτηματικό ";" ή με αλλαγή γραμμής (Enter). Ο κενός πίνακα κελιών ορίζεται μόνο με τη χρήση των αγκίστρων:

Όνομα κενού πίνακα κελιών = { }

Η γενική σύνταξη ενός πίνακα κελιών είναι η εξής:

όνομα πίνακα κελιών = { 'τιμή1', 'τιμή2', 'τιμή3', ... } → κελί μίας γραμμής, πολλών στηλών

όνομα πίνακα κελιών = { 'τιμή1'; 'τιμή2'; 'τιμή3'; ... } → κελί μίας στήλης, πολλών γραμμών

όνομα πίνακα κελιών = { 'τιμή1', 'τιμή2', ... ; 'τιμή3', 'τιμή4', ... } → κελί πολλών γραμμών ,στηλών

Η τμηματική σύνταξη ενός πίνακα κελιών είναι:

όνομα πίνακα κελιών{1,1}='τιμή1' όνομα πίνακα κελιών{1,2}='τιμή2' όνομα πίνακα κελιών{1,3}='τιμή3' ...	} κελί μίας γραμμής, πολλών στηλών
όνομα πίνακα κελιών{1,1}='τιμή1' όνομα πίνακα κελιών{2,1}='τιμή2' όνομα πίνακα κελιών{3,1}='τιμή3' ...	} κελί μίας στήλης, πολλών γραμμών
όνομα πίνακα κελιών{1,1}='τιμή1' όνομα πίνακα κελιών{1,2}='τιμή2' όνομα πίνακα κελιών{2,1}='τιμή3' όνομα πίνακα κελιών{2,2}='τιμή4' ...	} κελί πολλών γραμμών και στηλών

Παρακάτω δίνονται μερικά ενδεικτικά παραδείγματα πίνακα κελιών τόσο με τη μέθοδο της γενικής όσο και με τη μέθοδο της τμηματικής σύνταξης:

Παραδείγματα γενικής σύνταξης εισαγωγής πίνακα κελιών:

>> `Pinakas_kelion = [1 2 ; 4 5; 7 8] , 2+3i , 'Hello world ' , [0:2:10]`

`Pinakas_kelion =`

```
[3x2 double] [2.0000 + 3.0000i] 'Hello world' [1x6 double]
```

```
>> Pinakas_kelion = { [1 2 ; 4 5; 7 8] , 2+3i ; 'Hello world ' , [ 0:2:10] }
```

```
Pinakas_kelion =
```

```
[3x2 double] [2.0000 + 3.0000i]
'Hello world ' [1x6 double]
```

```
>> Pinakas_kelion = { [1 2 ; 4 5; 7 8] ; 2+3i ; 'Hello world ' ; [ 0:2:10] }
```

```
Pinakas_kelion =
```

```
[3x2 double]
[2.0000 + 3.0000i]
'Hello world '
[1x6 double]
```

Παράδειγμα τμηματικής σύνταξης εισαγωγής πίνακα κελιών:

```
>>A(1,1)={ [1 2 ; 4 5; 7 8] }
```

→ πίνακας

```
A =
[3x2 double]
```

```
>>A(1,2)={ 2+3i }
```

→ μιγαδικός

```
A =
[3x2 double] [2.0000 + 3.0000i]
```

```
>>A(2,1)={ 'Hello world' }
```

→ ακολουθία χαρακτήρων

```
A =
[3x2 double] [2.0000 + 3.0000i]
'Hello world' []
```

```
>>A(2,2)={ [ 0:2:10] }
```

→ διάνυσμα με στοιχεία από 0 έως 10 με βήμα 2

```
A =
[3x2 double] [2.0000 + 3.0000i]
'Hello world' [1x6 double]
```

Με τις παραπάνω τέσσερις εντολές κατασκευάστηκε ένας πίνακας τεσσάρων κελιών 2x2, όπου η δήλωση του κάθε κελιού πραγματοποιήθηκε ξεχωριστά. Έτσι, για παράδειγμα, η εντολή `>>A(2,1)={ 'Hello world' }` δηλώνει ότι το κελί που βρίσκεται στη δεύτερη γραμμή και πρώτη στήλη του πίνακα κελιών, θα αποτελείται από μια ακολουθία χαρακτήρων. Επιπλέον, παρατηρώντας το αποτέλεσμα της τελευταίας εντολής, με την οποία ολοκληρώνεται και ο πίνακας κελιών, βλέπουμε το MATLAB δε μας δείχνει τα περιεχόμενα όλων των κελιών. Εμφανίζονται μόνο τα κελιά εκείνα που δεν καταλαμβάνουν σημαντικό χώρο στην οθόνη, όπως το "Hello world" ή ο μιγαδικός αριθμός, ενώ για όλα τα υπόλοιπα κελιά παρουσιάζει απλώς τον τύπο τους.

7.3.2. ΕΜΦΑΝΙΣΗ ΤΙΜΩΝ ΚΕΛΙΟΥ

Αν ο χρήστης θέλει να δει τα περιεχόμενα ενός κελιού, πρέπει να πληκτρολογήσει, στο παράθυρο εντολών του MATLAB, το όνομα του πίνακα κελιών και το κελί που τον ενδιαφέρει σε άγκιστρα. Για παράδειγμα, η εμφάνιση των περιεχομένων του πρώτου κελιού (1,1), δηλαδή πρώτη γραμμή και πρώτη στήλη, πραγματοποιείται με την εξής εντολή:

```
>> A{1,1}
ans =
     1     2
     4     5
     7     8
```

Η εμφάνιση των περιεχομένων του κελιού (2,2) πραγματοποιείται με την εντολή:

```
>> A{2,2}
ans =
     0     2     4     6     8    10
```

7.3.3. Προσθήκη νέου Κελιού

Η προσθήκη ενός κελιού, σε ήδη υπάρχοντα πίνακα κελιών, μπορεί να γίνει ανά πάσα στιγμή με τη δήλωση του κελιού και του περιεχομένου του. Συνεχίζοντας το παραπάνω παράδειγμα, η προσθήκη ενός κελιού, το οποίο θα αποτελείται από έναν ακέραιο αριθμό, στην τρίτη γραμμή και πρώτη στήλη (3,1), γίνεται με την εντολή:

```
>> A(3,1)={145}
A =
 [3x2 double] [2.0000 + 3.0000i]
 'Hello world' [1x6 double]
 [ 145]        []
```

Αντίστοιχα, η προσθήκη ενός κελιού, το οποίο θα αποτελείται από έναν ακέραιο αριθμό, στην πρώτη γραμμή και τρίτη στήλη (1,3), γίνεται με την εντολή:

```
>> A(1,3)={ 145}
A =
 [3x2 double] [2.0000 + 3.0000i] [145]
 'Hello world' [1x6 double]      []
```

Στα δύο παραπάνω παραδείγματα παρατηρούμε πως η δημιουργία ενός πρόσθετου κελιού σε ένα πίνακα κελιών, έχει ως αποτέλεσμα όχι μόνο τη δημιουργία του κελιού που επιθυμούμε, αλλά και τη δημιουργία των υπόλοιπων κελιών, προκειμένου ο πίνακας να κρατήσει τις μέγιστες διαστάσεις που είχε προηγουμένως. Πιο αναλυτικά, ο πίνακας κελιών A,

πριν τη προσθήκη του τελευταίου κελιού, αποτελούνταν από δύο γραμμές και δύο στήλες. Με την εντολή `>> A(3,1)={145}`, προστέθηκε στον πίνακα μία γραμμή. Επομένως, οι διαστάσεις του άλλαξαν σε 3x2 (τρεις γραμμές, δύο στήλες). Επειδή με την παραπάνω εντολή ορίστηκε μόνο το περιεχόμενο του ενός κελιού, ενώ δημιουργήθηκαν δύο, το MATLAB ορίζει αυτόματα τον κενό πίνακα ως προεπιλεγμένη τιμή των κελιών που δεν ορίστηκαν. Αντίστοιχα, το ίδιο συμβαίνει και με την εντολή του δεύτερου παραδείγματος, `>> A(1,3)={ 145}`, όπου οι διαστάσεις του πίνακα κελιών μεταβλήθηκαν από 2x2 σε 2x3. Όπως ακριβώς και στην προηγούμενη περίπτωση έτσι και εδώ, στο νέο κελί (2,3) ορίστηκε ως προεπιλεγμένη τιμή ο κενός πίνακας. Ωστόσο, η τιμή αυτή, μπορεί να αλλάξει με μια ανάθεση στο συγκεκριμένο κελί τα μεγέθη που εμείς επιθυμούμε.

7.3.4. Χειρισμός Πίνακα Κελιών

Μερικές χρήσιμες συναρτήσεις για τους πίνακες κελιών είναι η `cell`, `celldisp` και η `cellplot`, οι οποίες αναλύονται παρακάτω:

ΕΝΤΟΛΗ `cell`

Η εντολή `cell` δημιουργεί ένα κενό πίνακα κελιών με μοναδική παράμετρο τις διαστάσεις. Αν και η ανάθεσή του σε κάποια μεταβλητή δεν είναι υποχρεωτική, είναι απαραίτητη για την ευκολότερη διαχείρισή του. Επομένως η σύνταξη της εντολής `cell` είναι η εξής:

Όνομα πίνακα κελιών = `cell(n)` → δημιουργεί κενό πίνακα κελιών διαστάσεων $n \times n$

Όνομα πίνακα κελιών = `cell(m,n)` → δημιουργεί κενό πίνακα κελιών διαστάσεων $m \times n$

Όνομα πίνακα κελιών = `cell(m,n,z)` → δημιουργεί κενό πίνακα κελιών διαστάσεων $m \times n \times z$...

Παρακάτω δίνονται μερικά ενδεικτικά παραδείγματα δημιουργίας κενού πίνακα κελιών προκαθορισμένων διαστάσεων:

```
>> K=cell(5,2)
```

K =

```

[] []
[] []
[] []
[] []
[] []
```

```
>> L=cell(4)
```

L =

```

[] [] [] []
[] [] [] []
[] [] [] []
[] [] [] []
```

ΕΝΤΟΛΗ `celldisp`

Η εντολή `celldisp` απεικονίζει ένα πίνακα κελιών. Το ιδιαίτερο αυτής της εντολής είναι ότι απεικονίζει όλα τα περιεχόμενα των κελιών του πίνακα χωρίς να χρειάζεται να πληκτρολογήσουμε την εντολή εμφάνισης περιεχομένου ξεχωριστά για το κάθε κελί. Η σύνταξη

της εντολής `celldisp` είναι ιδιαίτερα απλή, καθώς το μόνο που χρειάζεται να εισάγουμε στο παράθυρο εντολών του MATLAB, είναι την εντολή `celldisp` και δίπλα, μέσα σε παρένθεση, το όνομα του πίνακα κελιών, όπως φαίνεται παρακάτω:

```
>>celldisp(A)
```

```
A{1,1} =
     1     2
     4     5
     7     8

A{2,1} =
Hello world

A{1,2} =
 2.0000 + 3.0000i

A{2,2} =
     0     2     4     6     8    10

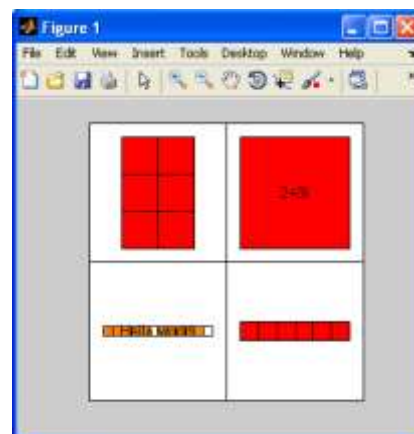
A{1,3} =
 145

A{2,3} =
 []
```

ΕΝΤΟΛΗ `cellplot`

Η εντολή `cellplot` δημιουργεί τη γραφική απεικόνιση του πίνακα κελιών. Εμφανίζεται ένα παράθυρο γραφικών (*figure*), που αναλύονται στο κεφάλαιο 13, με μια γραφική απεικόνιση του πίνακα κελιών. Κάθε τύπος μεταβλητής απεικονίζεται με διαφορετικό τρόπο με σκοπό το γρήγορο οπτικό έλεγχο του είδους των περιεχομένων του. Παρακάτω βλέπουμε ένα παράδειγμα της εντολής `cellplot` με είσοδο τον πίνακα κελιών του κεφαλαίου 7.3.1:

```
>>cellplot(Pinakas_kelion)
Pinakas_kelion =
 [3x2 double] [2.0000 + 3.0000i]
 'Hello world ' [1x6 double]
```



ΚΕΦΑΛΑΙΟ 8: ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

Το MATLAB διαθέτει ένα μεγάλο πλήθος ενσωματωμένων συναρτήσεων, οι οποίες είναι προγραμματισμένες να εκτελούν συγκεκριμένη λειτουργία. Ορισμένες από αυτές τις συναρτήσεις σχετίζονται με τον υπολογισμό των τριγωνομετρικών αριθμών μιας γωνίας (σε rad), άλλες με το δεκαδικό και νεπέριο λογάριθμο ενός αριθμού κ.α. Παρακάτω παρουσιάζονται μερικές από τις βασικές ενσωματωμένες συναρτήσεις. Στα παραδείγματα εκτέλεσής των τριγωνομετρικών συναρτήσεων χρησιμοποιείται η μαθηματική σταθερά $\pi=3,14$, όπου στο MATLAB συμβολίζεται με `pi`.

Βασικές μαθηματικές συναρτήσεις	
<code>sqrt(x)</code>	Υπολογίζει τη τετραγωνική ρίζα του αριθμού x .
<code>abs(x)</code>	Υπολογίζει την απόλυτη τιμή του αριθμού x .
<code>sign(x)</code>	Επιστρέφει το πρόσημο του αριθμού x . (+1 εάν $x \geq 0$, -1 αν $x < 0$)
<code>floor(x)</code>	Επιστρέφει τον αμέσως μικρότερο ακέραιο αριθμό του x .
<code>ceil(x)</code>	Επιστρέφει τον αμέσως μεγαλύτερο ακέραιο αριθμό του x .
<code>fix(x)</code>	Επιστρέφει τον αμέσως κοντινότερο στο μηδέν σε σχέση με τον αριθμό x .
<code>round(x)</code>	Επιστρέφει τον κοντινότερο ακέραιο αριθμό (μεγαλύτερο ή μικρότερο) του x .
Τριγωνομετρικές συναρτήσεις	
<code>sin(x)</code>	Υπολογίζει το ημίτονο της γωνίας x .
<code>cos(x)</code>	Υπολογίζει το συνημίτονο της γωνίας x .
<code>tan(x)</code>	Υπολογίζει την εφαπτομένη της γωνίας x .
<code>cot(x)</code>	Υπολογίζει τη συνεφαπτομένη της γωνίας x .
<code>asin(x)</code>	Υπολογίζει το τόξο ημίτονου του αριθμού x .
<code>acos(x)</code>	Υπολογίζει το τόξο συνημίτονου του αριθμού x .
<code>atan(x)</code>	Υπολογίζει το τόξο εφαπτομένης του αριθμού x .
<code>acot(x)</code>	Υπολογίζει το τόξο συνεφαπτομένης του αριθμού x .
Εκθετικές και λογαριθμικές συναρτήσεις	
<code>exp(x)</code>	Υπολογίζει την παράσταση e^x .
<code>log(x)</code>	Υπολογίζει το λογάριθμο με βάση το e του αριθμού x .
<code>log10(x)</code>	Υπολογίζει το λογάριθμο με βάση το δέκα του αριθμού x .
<code>log2(x)</code>	Υπολογίζει το λογάριθμο με βάση το δύο του αριθμού x .

Εντολή	Επεξήγηση	Εντολή	Επεξήγηση
<code>>> sqrt(4)</code> <code>ans =</code> <code>2</code>	$\sqrt{4} = 2$	<code>>> sin(3*pi/2)</code> <code>ans =</code> <code>-1</code>	$\eta\mu(3\pi/2) = -1$
<code>>> abs(-10)</code> <code>ans =</code> <code>10</code>	$ -10 = 10$	<code>>> sin(0)</code> <code>ans =</code> <code>0</code>	$\eta\mu(0) = 0$
<code>>> sign(-3)</code> <code>ans =</code> <code>-1</code>	Αποτέλεσμα = -1, αφού ο αριθμός είναι αρνητικός	<code>>> cos(pi/4)</code> <code>ans =</code> <code>0.7071</code>	$\sigma\upsilon\nu(\pi/4) = (\sqrt{2})/2 = 0.7071$

<pre>>> floor(5.6) ans = 5</pre>	Μικρότερος ακέραιος κοντά στο 5,6 είναι ο 5.	<pre>>> sin(pi/4) ans = 0.7071</pre>	$\eta\mu(\pi/4) = (\sqrt{2}) / 2 = 0.7071$
<pre>>> ceil(5.6) ans = 6</pre>	Μεγαλύτερος ακέραιος κοντά στο 5,6 είναι ο 6.	<pre>>> sqrt(2)/2 ans = 0.7071</pre>	$(\sqrt{2}) / 2 = 0.7071$
<pre>>> fix(6.324) ans = 6</pre>	Αμέσως κοντινότερος στο 0 του 6,324 είναι ο 6.	<pre>>> exp(0) ans = 1</pre>	$e^0 = 1$
<pre>>> round(5.49) ans = 5</pre>	Κοντινότερος ακέραιος αριθμός του 5,49 είναι ο 5.	<pre>>> exp(log10(1)) ans = 1</pre>	$e^{\log_{10}(1)}$
<pre>>> round(5.50) ans = 6</pre>	Κοντινότερος ακέραιος αριθμός του 5,50 είναι ο 6.	<pre>>> log10(10) ans = 1</pre>	$\log_{10}(10)$
<pre>>> cos(pi) ans = -1</pre>	$\text{c}\omega\sigma(\pi) = -1$	<pre>>> log(1) ans = 0</pre>	$\log(1)=0$
<pre>>> sin(pi/2) ans = 1</pre>	$\eta\mu(\pi/2) = 1$	<pre>>> log(0) ans = -Inf</pre>	$\log(0)=\infty$

ΚΕΦΑΛΑΙΟ 9: ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΤΟΥ ΧΩΡΟΥ ΕΡΓΑΣΙΑΣ

Παρακάτω παρουσιάζονται ορισμένες χρήσιμες εντολές για τη διαχείριση του παραθύρου εργασίας και των μεταβλητών που έχουμε δημιουργήσει στο MATLAB.

9.1. Η εντολή `save`

Με την εντολή `save όνομα αρχείου` ή `save όνομα αρχείου.mat` δημιουργείται ένα δυαδικό αρχείο με το όνομα που ορίσαμε και αποθηκεύονται σε αυτό όλες οι μεταβλητές που είχαμε χρησιμοποιήσει. Οπότε στην περίπτωση που έχουμε δημιουργήσει τις μεταβλητές `x,y,k,t` με τιμές 3,5,7 και 9 αντίστοιχα και επιθυμούμε την αποθήκευσή τους στο αρχείο "metablites_iouniou", δεν έχουμε παρά να πληκρολογήσουμε στο παράθυρο εντολών της MATLAB την εντολή:

```
>> save metablites_iouniou ή save metablites_iouniou .mat
```

Με αυτό τον τρόπο έχουμε αποθηκεύσει όλες τις μεταβλητές που δημιουργήσαμε με τις αντίστοιχες τιμές τους. Στην περίπτωση όπου επιθυμούμε την αποθήκευση μόνο των μεταβλητών `x,y` πληκρολογούμε την εντολή `save` με το όνομα του αρχείου και δίπλα τις μεταβλητές που μας ενδιαφέρουν:

```
>> save metablites_iouniou x y
```

Στην περίπτωση όπου δεν οριστεί κάποιο όνομα αρχείου, τότε το MATLAB αποθηκεύει όλες τις μεταβλητές με τα περιεχόμενά τους στο προκαθορισμένο αρχείο `matlab.mat`.

9.2. Η εντολή `diary`

Με την εντολή `diary` δημιουργείται ένα ASCII αρχείο με το όνομα που ορίσαμε και αποθηκεύονται σε αυτό όλες οι εντολές που εκτελέσαμε καθώς επίσης και τα αποτελέσματα που μας επέστρεψε το MATLAB. Πρόκειται δηλαδή για ένα «ημερολόγιο» εντολών και αποτελεσμάτων από τη στιγμή που εισάγαμε την εντολή `diary`, στο παράθυρο εντολών του MATLAB, έως την έξοδό μας από το πρόγραμμα ή την εντολή `diary off`. Η εντολή `diary off` διακόπτει την αποθήκευση των εντολών και των αποτελεσμάτων δημιουργώντας έτσι το ASCII αρχείο. Στην περίπτωση όπου δεν έχουμε ορίσει το όνομα του αρχείου, τότε το MATLAB εισάγει ως προεπιλεγμένο όνομα το "diary". Η σύνταξη της εντολής είναι :

```
>> diary('όνομα αρχείου.txt')
```

Έστω ότι χρησιμοποιούμε την εντολή `diary` για τη δημιουργία του αρχείου "entoles.txt" και στη συνέχεια πληκρολογούμε τις εντολές που ακολουθούν παρακάτω :

```
>> diary('entoles.txt')
```

```
>> x=4  
x =  
    4
```

```
>> y=5  
y =
```

```

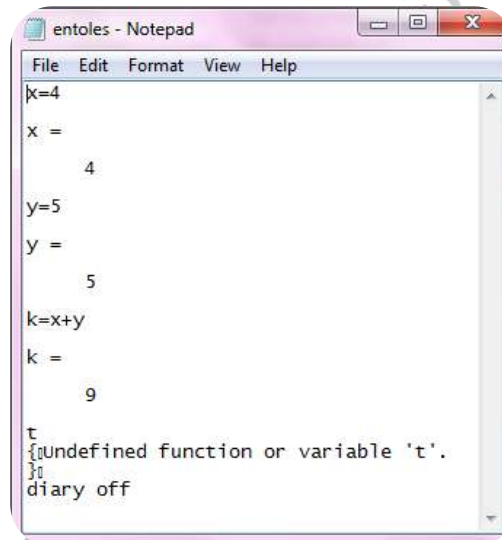
5
>> k=x+y
k =
    9

>> t
Undefined function or variable 't'.

>> diary off

```

Η πληκτρολόγηση της τελευταίας εντολής θα μπορούσε να είχε αντικατασταθεί με το κλείσιμο του MATLAB είτε από το κεντρικό μενού είτε από το δεξί κόκκινο εικονίδιο [X]. Όπως βλέπουμε στην παρακάτω εικόνα, το αρχείο entoles.txt συμπεριλαμβάνει εκτός από τα αποτελέσματα των εντολών που πληκτρολογήσαμε και τα μηνύματα λάθους που μας εμφανίζει το MATLAB. Στη συγκεκριμένη περίπτωση το μήνυμα λάθους προέκυψε, διότι ζητήθηκε να εμφανιστεί το περιεχόμενο της μεταβλητής t, η οποία όμως δεν είχε προηγουμένως οριστεί.



```

entoles - Notepad
File Edit Format View Help
x=4
x =
    4
y=5
y =
    5
k=x+y
k =
    9

t
{undefined function or variable 't'.
}
diary off

```

Εικόνα 6 – Παράδειγμα εντολής diary.

9.3. Η εντολή load

Με την εντολή **load** ανακτούμε όλες τις μεταβλητές που έχουμε αποθηκεύσει με τη βοήθεια της εντολής **save**. Οι μεταβλητές ανακτώνται στη μνήμη του υπολογιστή με σκοπό την περαιτέρω αξιοποίησή τους. Η σύνταξη της εντολής είναι **load όνομα αρχείου**. Στην περίπτωση όπου δε δοθεί κάποιο όνομα αρχείου, τότε το MATLAB θα ανακαλέσει τις εντολές από το προκαθορισμένο αρχείο **matlab.mat**, εφόσον αυτό έχει δημιουργηθεί από την εντολή **save**.

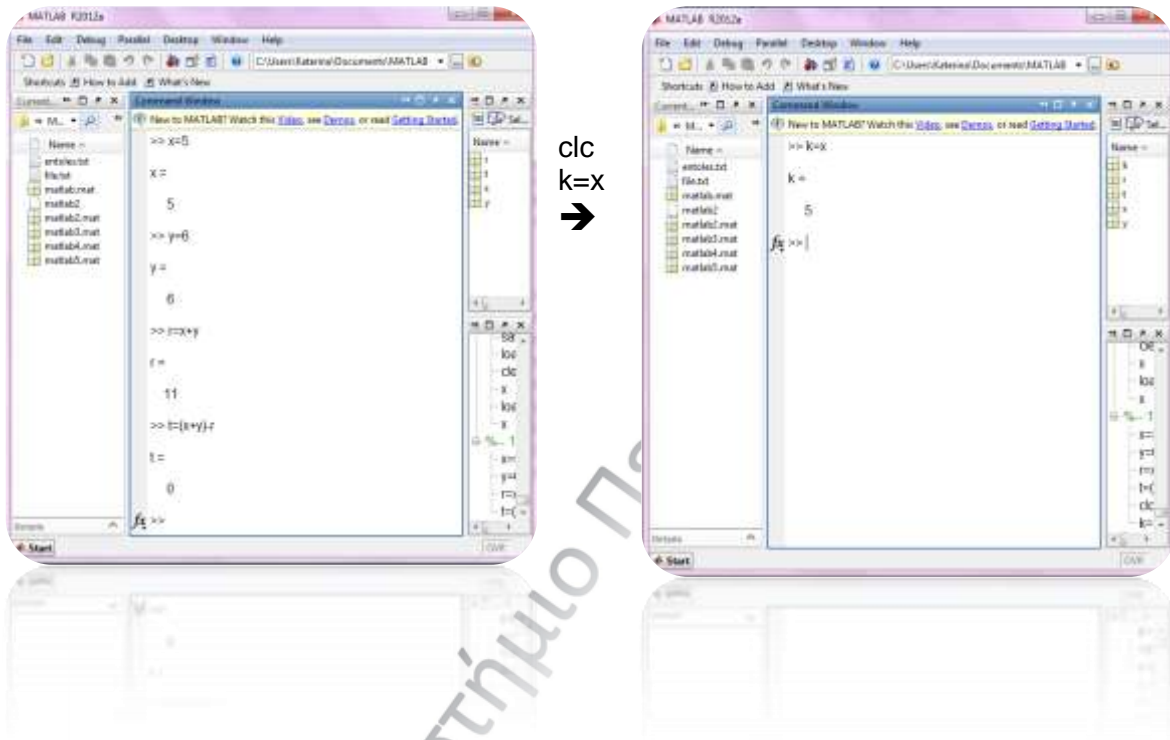
Στα παραδείγματα της εντολής **save** είχαμε δημιουργήσει το αρχείο **metablites_iouniou**, το οποίο περιλαμβάνει τις μεταβλητές $x=3$, $y=5$, $k=7$, $t=9$. Για την ανάκτησή τους χρησιμοποιούμε την εντολή **load metablites_iouniou**, ενώ στην περίπτωση που δεν είχαμε ορίσει το όνομα του αρχείου θα χρησιμοποιούσαμε την εντολή **load** μόνη της.

9.4. Η εντολή clc

Η εντολή **clc** καθαρίζει την οθόνη του Παραθύρου εντολών (Command Window) από τις μεταβλητές εισόδου και εξόδου. Παρόλο που ο χώρος εργασίας φαίνεται κενός, οι μεταβλητές

που έχουν δημιουργηθεί μέχρι τώρα δε διαγράφονται αλλά παραμένουν στη μνήμη του υπολογιστή και έτσι μπορούν να χρησιμοποιηθούν ανά πάσα στιγμή.

Παρακάτω βλέπουμε το Παράθυρο εντολών, το οποίο περιέχει τέσσερις μεταβλητές. Ύστερα από την πληκτρολόγηση της εντολής **clc**, το Παράθυρο εντολών καθαρίζεται και έτσι όταν εισάγουμε μία εντολή θα εμφανιστεί στο άνω μέρος του παραθύρου. Η εντολή η οποία εισάγεται στο παράδειγμα είναι η δημιουργία μίας νέας μεταβλητής *k* στην οποία γίνεται ανάθεση η μεταβλητή *x*, που είχαμε δημιουργήσει προηγουμένως.



Εικόνα 7 – Παράδειγμα εντολής **clc**.

9.5. Οι εντολές **quit** και **exit**

Με τις εντολές **quit** και **exit** πραγματοποιείται η έξοδος από το MATLAB. Ένας εναλλακτικός τρόπος εξόδου είναι η επιλογή του από το κεντρικό μενού (File → Exit MATLAB) ή με τη χρήση της συντόμευσης Ctrl+Q. Επειδή τα δεδομένα χάνονται με την έξοδό μας από το πρόγραμμα, καλό είναι πριν από κάθε τερματισμό της λειτουργίας του MATLAB, να αποθηκεύουμε την εργασία που κάναμε είτε με την εντολή **save** είτε με την εντολή **diary**.

ΚΕΦΑΛΑΙΟ 10: ΕΙΣΟΔΟΣ ΚΑΙ ΕΞΟΔΟΣ ΔΕΔΟΜΕΝΩΝ

Σε αυτό το κεφάλαιο παρουσιάζονται τέσσερις πολύ χρήσιμες εντολές για την είσοδο και την απεικόνιση των μεταβλητών και των τιμών τους από και προς το χρήστη. Με τις παρακάτω εντολές μας δίνεται η δυνατότητα να εμφανίζουμε μηνύματα κειμένου στο χρήστη, να δεχόμαστε την είσοδο δεδομένων από αυτόν κατά τη διάρκεια εκτέλεσης του προγράμματος, αλλά και να σταματάμε την εκτέλεση του προγράμματος για συγκεκριμένο χρονικό διάστημα. Αξιοποιούνται κυρίως στα αρχεία *m-files*, τα οποία παρουσιάζονται αναλυτικά στο κεφάλαιο δώδεκα (12).

10.1. Η Εντολή `input`

Η εντολή `input` εισάγει δεδομένα ύστερα από την προβολή ενός μηνύματος προτροπής στο χρήστη. Τα δεδομένα συνήθως αποθηκεύονται σε μία μεταβλητή, προκειμένου να είναι δυνατή η επεξεργασία τους. Η γενική σύνταξη της εντολής είναι:

όνομα μεταβλητής = `input('μήνυμα εμφάνισης')`

Με την εκτέλεση της εντολής εμφανίζεται στην οθόνη το μήνυμα προτροπής που ορίστηκε και το πρόγραμμα αναμένει την απόδοση μίας τιμής, μέσω του πληκτρολογίου. Στη συνέχεια, αποθηκεύεται στη μεταβλητή το περιεχόμενο που εισήγαγε ο χρήστης.

Στην περίπτωση όπου χρειαζόμαστε, για παράδειγμα, να αποθηκεύσουμε την ηλικία του χρήστη, προκειμένου να την επεξεργαστούμε αργότερα, η εντολή `input` είναι κατάλληλη αφού μέσω του μηνύματος προτροπής ενημερώνουμε το χρήστη για το τί δεδομένα περιμένουμε από αυτόν. Τα παραπάνω επιτυγχάνονται με την εντολή:

```
>> age=input('Please enter your age: ')
```

Σε αυτό το σημείο το MATLAB έχει εμφανίσει το μήνυμα “ *Please enter your age:*” στο χρήστη και περιμένει την απάντησή του. Όταν ο χρήστης εισάγει την ηλικία του, αυτή θα αποθηκευτεί στη μεταβλητή `age`.

Στην περίπτωση, όπου τα δεδομένα που αναμένονται από το χρήστη δεν είναι αριθμητικά (αριθμοί, πίνακες, πίνακες κελιών), αλλά αλφαριθμητικά (ακολουθίες χαρακτήρων), τότε οι τιμές που θα δοθούν από το χρήστη πρέπει να δοθούν μέσα σε μονά εισαγωγικά (`' '`).

```
>> name=input('Please enter your name: ')
```

```
Please enter your name: 'Katerina'
```

```
name =  
Katerina
```

Προκειμένου να αποφευχθεί αυτό, είναι απαραίτητος ο ορισμός μίας επιπλέον παραμέτρου, της `'s'`. Έτσι, η σύνταξη της εντολής γίνεται:

όνομα μεταβλητής = `input('μήνυμα εμφάνισης', 's')`

Οπότε, για την αποθήκευση του ονόματος του χρήστη, για παράδειγμα, με τη βοήθεια της εντολής `input` πληκτρολογούμε:

```
>> name=input('Please enter your name: ','s')
```

```
Please enter your name: Katerina
```

```
name =  
Katerina
```

10.2. Η Εντολή `disp`

Η εντολή `disp` απεικονίζει το περιεχόμενο μιας μεταβλητής ή ένα αλφαριθμητικό στην οθόνη. Δέχεται μόνο ένα όρισμα, το οποίο μπορεί να είναι είτε μία μεταβλητή είτε ένα αλφαριθμητικό, το οποίο περικελείται σε μονά εισαγωγικά (`' '`). Στην περίπτωση της μεταβλητής, το περιεχόμενό της μπορεί να αποτελείται από έναν αριθμό, πίνακα ή μία ακολουθία χαρακτήρων. Στην οθόνη εκτυπώνεται κάθε φορά μόνο το περιεχόμενό της χωρίς το όνομα της. Στην περίπτωση όπου το όρισμα της εντολής είναι ένα αλφαριθμητικό (περικλειόμενο μέσα σε μονά εισαγωγικά), εκτυπώνεται χωρίς τα εισαγωγικά. Η σύνταξη της

εντολής **disp** είναι απλή. Για να τη χρησιμοποιήσουμε πληκτρολογούμε στο παράθυρο εντολών του MATLAB:

- disp('κείμενο')** για να εμφανιστεί το κείμενο στην οθόνη του χρήστη
 ή **disp(μεταβλητή)** για να εμφανιστεί το περιεχόμενο μιας μεταβλητής στην οθόνη του χρήστη.

Παρακάτω παρουσιάζονται μερικά ενδεικτικά παραδείγματα της εντολής **disp**.

Εντολή	Επεξήγηση
<pre>>> x=5 x = 5 ----- >> A=[1 2; 3 4] A = 1 2 3 4 ----- >> text='Matlab!' text = Matlab!</pre>	<p>Πρώτα ορίζονται οι τρεις μεταβλητές, οι οποίες είναι απαραίτητες για τα παραδείγματα της εντολής disp.</p> <p>x → μεταβλητή αριθμού</p> <p>A → μεταβλητή πίνακα αριθμών</p> <p>text → μεταβλητή αλφαριθμητικού</p>
<pre>>> disp(x) 5</pre>	<p>Η εντολή disp δέχτηκε ως όρισμα μία μεταβλητή αριθμού και εκτύπωσε μόνο το περιεχόμενό της.</p>
<pre>>> disp(A) 1 2 3 4</pre>	<p>Η εντολή disp δέχτηκε ως όρισμα μία μεταβλητή πίνακα και εκτύπωσε μόνο το περιεχόμενό της, δηλαδή τον πίνακα.</p>
<pre>>> disp(text) Matlab!</pre>	<p>Η εντολή disp δέχτηκε ως όρισμα μία μεταβλητή αλφαριθμητικού και εκτύπωσε μόνο το περιεχόμενό της, δηλαδή το κείμενο.</p>
<pre>>> disp('MATLAB') MATLAB</pre>	<p>Η εντολή disp δέχτηκε ως όρισμα ένα αλφαριθμητικό και εκτύπωσε το ίδιο το αλφαριθμητικό.</p>

Μερικές φορές χρειάζεται να εμφανίσουμε την τιμή μιας μεταβλητής δίπλα από ένα μήνυμα κειμένου. Έτσι, επειδή η εντολή **disp** δέχεται μόνο ένα όρισμα, είναι αναγκαία η συνένωση του μηνύματος κειμένου με την τιμή της μεταβλητής. Αυτό επιτυγχάνεται με τη χρήση ενός πίνακα, όπου ως πρώτο στοιχείο ορίζεται το αλφαριθμητικό (μήνυμα κειμένου) και ως δεύτερο στοιχείο του ορίζεται η μεταβλητή που επιθυμούμε. Η σύνταξη της εντολής είναι η εξής:

`disp(['μήνυμα κειμένου',μεταβλητή])`

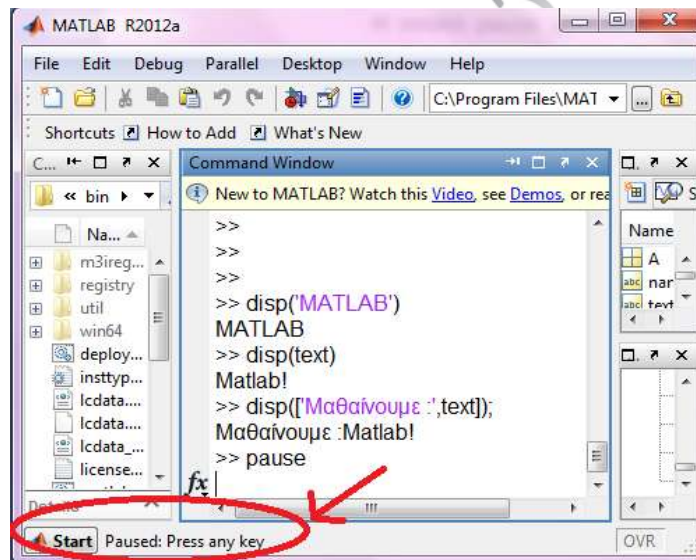
Παράδειγμα

```
>> disp(['Μαθαίνουμε : ',text])
```

Μαθαίνουμε : Matlab!

10.3. Η Εντολή pause

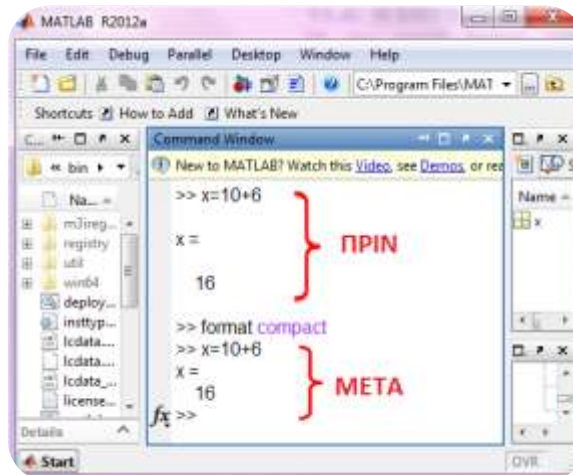
Η εντολή **pause** χρησιμοποιείται για την προσωρινή διακοπή της εκτέλεσης του προγράμματος που τρέχει στο MATLAB. Η διακοπή πραγματοποιείται όταν εκτελείται η εντολή **pause** και έτσι το MATLAB αναμένει κάποια επιπλέον εντολή από το χρήστη προκειμένου να συνεχίσει την εκτέλεση. Η έξοδος από την κατάσταση αναμονής γίνεται με το πάτημα οποιοδήποτε κουμπιού, όπως φαίνεται και στην ενημέρωση του MATLAB στην παρακάτω εικόνα.



Εικόνα 8 – Παράδειγμα εντολής pause.

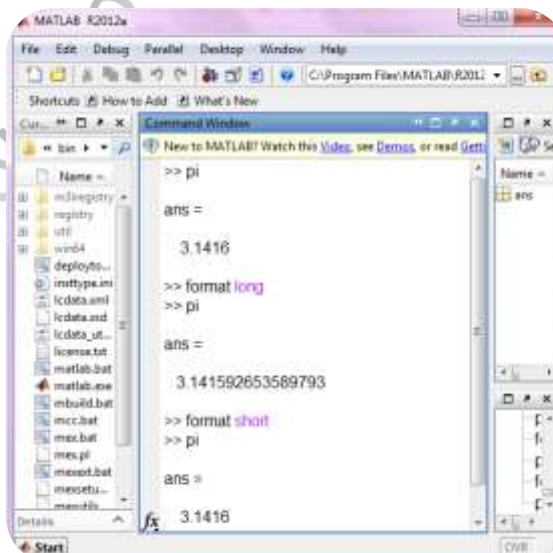
10.4. Η Εντολή format

Με την εντολή **format** ρυθμίζουμε τον τρόπο εμφάνισης των εξαγόμενων αποτελεσμάτων. Μία από τις χρήσεις της εντολής **format** είναι η ρύθμιση των κενών μεταξύ των γραμμών, όπου εμφανίζονται τα αποτελέσματα. Προκειμένου να εξοικονομήσουμε χώρο και άρα να έχουμε μικρή απόσταση μεταξύ των γραμμών χρησιμοποιούμε την εντολή **format compact**. Στην παρακάτω εικόνα παρουσιάζεται ο τρόπος εμφάνισης της εντολής `>>x=10+6` πριν την εντολή **format compact** και μετά. Παρατηρήστε πόσο πυκνά εμφανίζονται τα αποτελέσματα μετά την είσοδο της εντολής **format compact** στο Παράθυρο εντολών (Command Window).

Εικόνα 9 – Παράδειγμα εντολής `format compact`.

Ο εξ' ορισμού τρόπος εμφάνισης των αποτελεσμάτων στο MATLAB είναι αυτός που αντιστοιχεί στην εντολή `format loose`, κατά την οποία η απόσταση μεταξύ των γραμμών είναι αρκετά μεγαλύτερη από αυτή της εντολής `format compact`, προκειμένου να είναι πιο ευανάγνωστη η έξοδος που μας επιστρέφει.

Μία ακόμα χρήση της εντολής `format` είναι η εναλλαγή μεταξύ των διαφορετικών μορφών εμφάνισης των αποτελεσμάτων των αριθμών κινητής υποδιαστολής. Ο προκαθορισμένος τρόπος εμφάνισης των αποτελεσμάτων αντιστοιχεί στην εντολή `format short`, όπου εμφανίζονται μόνο τα τέσσερα (4) πρώτα ψηφία του αποτελέσματος μετά την υποδιαστολή. Στην περίπτωση όπου το αποτέλεσμα, που μας επιστρέφει το MATLAB, δεν έχει επαρκή αριθμό δεκαδικών ψηφίων, προκειμένου να δούμε το πραγματικό αποτέλεσμα, χρησιμοποιούμε την εντολή `format long`. Με την εντολή αυτή εμφανίζονται δεκαπέντε (15) ψηφία μετά την υποδιαστολή και όχι τέσσερα όπως είναι στην προκαθορισμένη μορφή. Στην ακόλουθη εικόνα παρουσιάζονται οι διαφορετικοί τρόποι εμφάνισης του αριθμού π ($\pi=3,14$) με τη χρήση των εντολών `format long` και `short`.

Εικόνα 10 – Παράδειγμα εντολής `format long` και `short`.

Η εντολή `format` χρησιμοποιείται επίσης για την εμφάνιση αριθμών κινητής υποδιαστολής σε μορφή πολλαπλασιασμένη με δύναμη του δέκα με τα ανάλογα ψηφία που επιθυμούμε. Για την εμφάνιση των αριθμών σε αυτή τη μορφή με εμφανή τα τέσσερα πρώτα

δεκαδικά ψηφία, μετά την υποδιαστολή, χρησιμοποιείται η εντολή **format short e**, ενώ για την εμφάνιση των αριθμών σε αυτή τη μορφή με εμφανή τα πρώτα δεκαπέντε ψηφία χρησιμοποιείται η εντολή **format long e**. Μία ακόμα χρήση της εντολής **format** είναι η εμφάνιση του πρόσημου ενός αριθμού και η εμφάνιση μόνο δύο δεκαδικών ψηφίων μετά την υποδιαστολή (κυρίως για νομισματικές αξίες, όπως ευρώ/λεπτά, δολάρια/σεντς). Στον ακόλουθο πίνακα παρουσιάζονται οι διαφορετικοί τρόποι εμφάνισης του δεκαδικού αριθμού 234,2012 ύστερα από την είσοδο των παραπάνω εντολών στο MATLAB:

Εντολή	Επεξήγηση
<pre>>> format short e >> 234.2012 ans = 2.3420e+02</pre>	<p>Ο αριθμός 234.2012 εμφανίζεται σε μορφή πολλαπλασιασμένη με δύναμη του δέκα με τέσσερα εμφανή δεκαδικά ψηφία ως $2.3420 * 10^2$.</p>
<pre>>> format long e >> 234.2012 ans = 2.342012000000000e+02</pre>	<p>Ο αριθμός 234.2012 εμφανίζεται σε μορφή πολλαπλασιασμένη με δύναμη του δέκα με δεκαπέντε εμφανή δεκαδικά ψηφία ως $2.342012000000000 * 10^2$.</p>
<pre>>> format + >> 234.2012 ans = +</pre>	<p>Το πρόσημο του αριθμού 234.2012 είναι +.</p>
<pre>>> format bank >> 234.2012 ans = 234.20</pre>	<p>Ο αριθμός 234.2012 εμφανίζεται σε μορφή σταθερής υποδιαστολής με δύο εμφανή δεκαδικά ψηφία ως 234.20.</p>

Τέλος, αξίζει να σημειωθεί ότι σε όλες τις περιπτώσεις χρήσης της εντολής **format** επηρεάζεται μόνο ο τρόπος εμφάνισης των αποτελεσμάτων στο Παράθυρο εντολών και όχι ο τρόπος υπολογισμού ή αποθήκευσής τους.

ΚΕΦΑΛΑΙΟ 11: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΗ MATLAB

Προκειμένου να εκμεταλλευτούμε πλήρως τις δυνατότητες που μας παρέχει το MATLAB είναι αναγκαία η δημιουργία και η εκτέλεση αρχείων m-files. Με τη χρήση των αρχείων αυτών επιταχύνεται η διαδικασία υπολογισμού των ζητούμενων δεδομένων, αφού δεν απαιτείται η επανέναρξη του MATLAB και η ανασύνταξη των εντολών για τον εκτελεσμό των πράξεων. Επίσης, ορισμένες φορές είναι αναγκαία η επανάληψη της εκτέλεσης μίας ομάδας εντολών, ύστερα από μία πιθανή τροποποίηση παραμέτρων. Τα m-files λύνουν όλα τα παραπάνω

προβλήματα, αφού μέσα σε αυτά αποθηκεύονται οι εντολές του MATLAB. Υπάρχουν δύο ειδών αρχεία m-files, τα αρχεία κειμένου (script files) και τα αρχεία συναρτήσεων (function files). Για τη δημιουργία ενός m-file μπορούμε να χρησιμοποιήσουμε έναν απλό συντάκτη κειμένου (editor), όπως το notepad, το notepad++ ή και το wordpad. Ωστόσο, το MATLAB διαθέτει δικό του συντάκτη, η εκκίνηση του οποίου γίνεται από το μενού **File → New → Script** (για τη δημιουργία αρχείου script), **File → New → Function** (για τη δημιουργία αρχείου συνάρτησης) ή με την εντολή **>> edit** στο παράθυρο εντολών του. Αμέσως, εμφανίζεται ένα παράθυρο μέσα στο οποίο συντάσσονται οι εντολές. Τα m-files πρέπει να αποθηκεύονται στο φάκελο εργασίας (working directory) του MATLAB, ώστε να υπάρχει η δυνατότητα άμεσης εκτέλεσης τους με την κλήση του ονόματός τους. Η ονομασία των αρχείων πρέπει να συνοδεύεται υποχρεωτικά από την προέκταση **.m**. Η κλήση των m-files γίνεται με την πληκτρολόγηση του ονόματός τους μέσα από το παράθυρο εντολών και ισοδυναμεί με την εκτέλεση των εντολών που περιέχουν.


11.1. Προγράμματα – Αρχεία Κειμένου (script m-files)

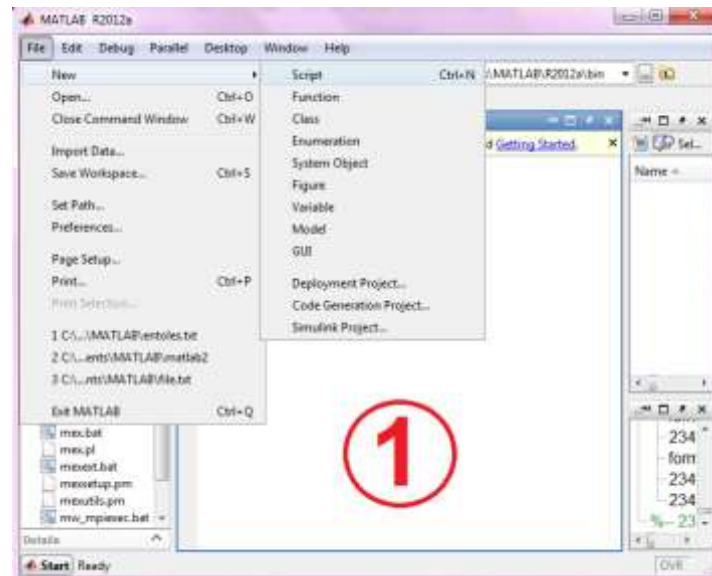
Τα αρχεία κειμένου αποτελούνται από ένα σύνολο εντολών, οι οποίες εκτελούνται από το MATLAB μόνο όταν καλέσουμε το αρχείο με το όνομά του, χωρίς την προέκταση **.m**, μέσα από το παράθυρο εντολών. Δεν έχουν ορίσματα εισόδου και εξόδου αλλά εκτελούν μια ακολουθία εντολών και μπορούν να περιέχουν, εκτός από εντολές, μεταβλητές, αριθμούς και συναρτήσεις.

Γενικά, είναι ιδιαίτερα χρήσιμα στις περιπτώσεις, όπου είναι απαραίτητη η επανάληψη εντολών και πράξεων, οι οποίες αλλάζουν κάθε φορά τιμή, χωρίς όμως να χρειάζεται η πληκτρολόγηση των εντολών ξεχωριστά. Μία ακόμα χρησιμότητα αυτών των αρχείων είναι η δυνατότητα εισαγωγής πολλών πληροφοριών.

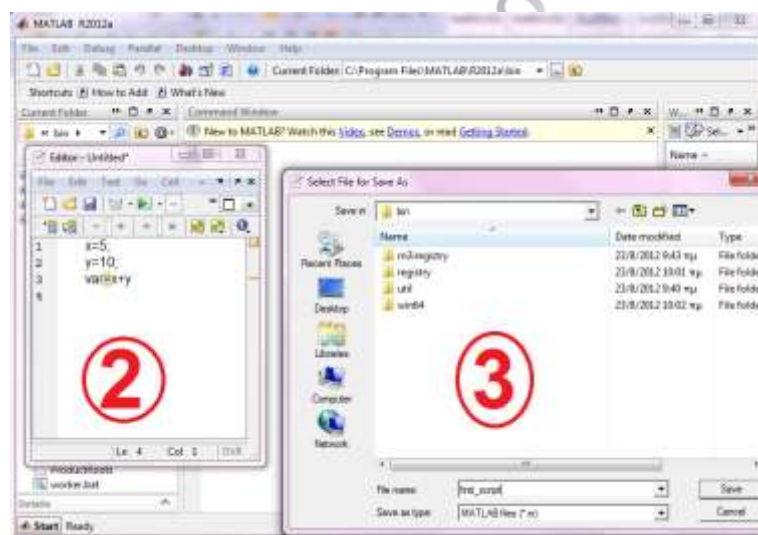
Το όνομα ενός αρχείου m-file δε μπορεί να έχει το ίδιο όνομα με μία μεταβλητή, η οποία υπολογίζεται μέσω του αρχείου, διότι στην περίπτωση αυτή το MATLAB δεν θα καταφέρει να ενεργοποιήσει το αρχείο αυτό. Επιπλέον, το όνομα του αρχείου m-file πρέπει να αρχίζει υποχρεωτικά με ένα χαρακτήρα (a...z, A...Z), ενώ το υπόλοιπο τμήμα του ονόματος μπορεί να περιέχει εκτός από χαρακτήρες, αριθμούς (0...9) όπως επίσης και το σύμβολο της κάτω παύλας (**_**, underscore). Οι περιορισμοί είναι ίδιοι με αυτούς των ονομάτων των μεταβλητών όπως παρουσιάστηκαν στο Κεφάλαιο 6.1.

Στο παρακάτω παράδειγμα βλέπουμε τη σειρά των βημάτων που απαιτούνται για τη δημιουργία και την εκτέλεση ενός πολύ απλού αρχείου κειμένου με το όνομα «**first_script**»:

- File → New → Script
- Είσοδος στο αναδυόμενο παράθυρο των εντολών:
x=5;
y=10;
var=x+y
- File → Save As ή πατώντας στο εικονίδιο .
- Πληκτρολόγηση του ονόματος **first_script** στο πεδίο File Name
- Πάτημα του εικονιδίου Save
- Πληκτρολόγηση του **first_script** στο παράθυρο εντολών: **>> first_script**



Εικόνα 11 – Παράδειγμα δημιουργίας Αρχείου Κειμένου.



Εικόνα 12 – Παράδειγμα αποθήκευσης Αρχείου Κειμένου.

11.2. Προγράμματα – Αρχεία Συναρτήσεων (function m-files)

Τα αρχεία συναρτήσεων αποτελούνται από ένα σύνολο εντολών, οι οποίες εκτελούνται από το MATLAB μόνο όταν καλέσουμε το αρχείο με το όνομά του, χωρίς την προέκταση .m, μέσα από το παράθυρο εντολών ή από άλλα προγράμματα. Σε αντίθεση με τα αρχεία κειμένου, διαθέτουν ορίσματα εισόδου-εξόδου και οι μεταβλητές τους είναι τοπικά ορισμένες. Αυτό σημαίνει πως οι μεταβλητές που ορίζονται στη συνάρτηση παύουν να ισχύουν μετά το τέλος της εκτέλεσής της.

Η δομή των αρχείων συναρτήσεων είναι συγκεκριμένη. Αποτελούνται από την επικεφαλίδα (header), τα σχόλια (comments), τις εντολές (statements), τις υποσυναρτήσεις (subfunctions) και την εντολή τερματισμού της συνάρτησης, τα οποία αναλύονται παρακάτω:

11.2.1. **Επικεφαλίδα(header)**

Η επικεφαλίδα αποτελείται από μία γραμμή όπου ορίζεται η συνάρτηση και καταγράφονται αναλυτικά οι μεταβλητές εισόδου και εξόδου. Χωρίς αυτή τη γραμμή το αρχείο συνάρτησης μετατρέπεται σε αρχείο κειμένου. Η επικεφαλίδα έχει την ακόλουθη μορφή:

function [output1, output2, ...] = function_name (input1, input2, ...)

Το function_name είναι το όνομα της συνάρτησης και πρέπει να ταυτίζεται με το όνομα του αρχείου **m-file** (χωρίς την κατάληξη .m). Για παράδειγμα, αν το όνομα της συνάρτησης είναι count_people, τότε και το όνομα του αρχείου συνάρτησης, το οποίο περιέχει τη συνάρτηση αυτή, πρέπει να ονομαστεί **count_people.m**.

Οι μεταβλητές εισόδου (input1, input2, ...) είναι σε παρενθέσεις, ενώ οι μεταβλητές εξόδου [output1, output2, ...] βρίσκονται σε αγκύλες. Ωστόσο, στην περίπτωση όπου έχουμε μόνο μια μεταβλητή εξόδου δεν είναι υποχρεωτική η χρήση αγκυλών. Για παράδειγμα αντί:

function [output1] = function_name (input1, input2, ...) μπορούμε να γράψουμε:

function output1 = function_name (input1, input2, ...)

Η δήλωση μεταβλητών εισόδου δεν είναι υποχρεωτική και έτσι μια συνάρτηση είναι δυνατόν να μην έχει καθόλου ορίσματα εισόδου. Ωστόσο στην περίπτωση αυτή όλες οι απαραίτητες μεταβλητές για την εξαγωγή κάποιου αποτελέσματος θα πρέπει να έχουν οριστεί εκ των προτέρων μέσα σε αυτή και όχι να ορίζονται κατά την κλήση της συνάρτησης. Προαιρετική είναι και η δήλωση μεταβλητών εξόδου, ωστόσο δε συνιστάται αφού σε αυτή την περίπτωση η κλήση της συνάρτησης δεν εξάγει κάποιο αποτέλεσμα αν δεν υπάρχει κατάλληλη εντολή εξόδου δεδομένων .

Για τη δήλωση και την ονομασία μιας συνάρτησης ισχύουν τρεις κανόνες:

1. Η εντολή function πρέπει να πληκτρολογείται με πεζά γράμματα, αλλιώς το MATLAB αδυνατεί να λάβει τον κώδικα ως συνάρτηση και έτσι δε γίνεται επιτυχής εκτέλεσή του.
2. Το όνομα της συνάρτησης (function_name) ακολουθεί του ίδιους περιορισμούς με αυτούς των ονομάτων των μεταβλητών (Κεφάλαιο 6.1).
3. Το όνομα της συνάρτησης (function_name) δεν πρέπει να συμπίπτει με όνομα συνάρτησης βιβλιοθήκης, όπως οι sin, cos, max, min, exp, size κ.α.

11.2.2. **Σχόλια(comments)**

Τα σχόλια ξεκινούν με το σύμβολο “%” και μπορούν να βρίσκονται οπουδήποτε μέσα στο κώδικά μας. Το κείμενο που ακολουθεί μετά το σύμβολο “%” δε λαμβάνεται υπόψη από το λογισμικό. Αν και δεν είναι υποχρεωτική η ενσωμάτωσή τους μέσα στον κώδικα, συνιστάται η προσθήκη σχολίων, αφού βοηθούν στην κατανόηση του κώδικα. Οι γραμμές των σχολίων, οι οποίες ακολουθούν αμέσως μετά από την επικεφαλίδα, προβάλλονται από το MATLAB, αν πληκτρολογηθεί η εντολή **help function_name**.

11.2.3. **Εντολές (statements)**

Οι εντολές αποτελούν το κύριο μέρος της συνάρτησης, αφού μέσω αυτών υπολογίζονται οι μεταβλητές εξόδου (output1, output2, ...). Συχνά σε αυτό το σημείο, παρατηρείται χρήση των δομών ελέγχου ροής του προγράμματος, οι οποίες είναι ένα σύνολο εντολών που εκτελούνται ή επαναλαμβάνονται όταν ικανοποιείται κάποια συνθήκη. Οι δομές ελέγχου ροής του προγράμματος αναλύονται στο επόμενο κεφάλαιο (κεφάλαιο 12.3).

11.2.4. Υποσυναρτήσεις (subfunctions)

Οι υποσυναρτήσεις είναι εσωτερικές συναρτήσεις (functions), οι οποίες περιέχονται σε ένα αρχικό αρχείο συνάρτησης (function m-file) και βρίσκονται μετά το τέλος της κύριας συνάρτησης. Αναγνωρίζονται μόνο από τις συναρτήσεις που περιέχονται στο αρχείο αυτό και ορίζονται με τον ίδιο ακριβώς τρόπο που ορίζονται και οι συναρτήσεις. Αποτελούνται από την επικεφαλίδα (header), τα σχόλια (comments), τις εντολές (statements), τις υποσυναρτήσεις (subfunctions) και την εντολή τερματισμού της συνάρτησης, τα οποία αναλύθηκαν παραπάνω.

Η χρήση υποσυναρτήσεων είναι πολύ σημαντική, αφού μειώνει τον όγκο του κώδικα της κύριας συνάρτησης και έτσι γίνεται πιο συνεκτική, κάνοντας ευκολότερο το διάβασμα και την απασφαλμάτωσή της. Ένα επίσης σημαντικό πλεονέκτημα των υποσυναρτήσεων είναι ότι περιλαμβάνουν ένα μέρος του κώδικα που χρησιμοποιείται σε πολλά σημεία της κύριας συνάρτησης.

11.2.5. Εντολή τερματισμού συνάρτησης

Ένα αρχείο συνάρτησης τερματίζεται με την εντολή “**end**” στην τελευταία γραμμή του κώδικα. Ωστόσο, η εντολή αυτή είναι υποχρεωτική μόνο στην περίπτωση όπου το αρχείο συνάρτησης περιλαμβάνει υποσυναρτήσεις.

11.2.6. Παραδείγματα

Παράδειγμα 1

Στο παρακάτω παράδειγμα θα ορίσουμε ένα αρχείο συνάρτησης το οποίο θα έχει το όνομα “**BMI**” (BMI=Body Mass Index -Δείκτης Μάζας Σώματος) και θα δέχεται ως ορίσματα εισόδου το βάρος και το ύψος ενός ανθρώπου. Στη συνέχεια με τον υπολογισμό του τύπου $\text{βάρος}/\text{ύψος}^2$ θα τερματίζει η συνάρτηση εμφανίζοντας το δείκτη μάζας σώματος, το οποίο θα αποτελεί και τη μοναδική μεταβλητή εξόδου. Το αρχείο συνάρτησης έχει την ίδια σειρά βημάτων με τη δημιουργία ενός αρχείου script, με μόνη διαφορά ότι αντί για File→New→Script επιλέγουμε File→New→Function:

<code>function [bmi] = BMI(weight,height)</code>	→ Επικεφαλίδα
<code>% Counts the BMI (Body Mass Index) of a person</code> <code>% BMI = weight / (height) ^2</code>	→ Σχόλια(comments)
<code>w=weight;</code> <code>h= (height)^2;</code> <code>bmi = w/h;</code>	→ Εντολές (statements)
<code>end</code>	→ Εντολή τερματισμού συνάρτησης

Εφόσον το όνομα της συνάρτησης είναι “BMI”, το όνομα του αρχείου συνάρτησης πρέπει να είναι BMI.m. Για να εκτελέσουμε το αρχείο, πληκτρολογούμε στο παράθυρο εντολών του MATLAB, το όνομα της συνάρτησης με τις μεταβλητές εισόδου σε παρένθεση. Για το συγκεκριμένο παράδειγμα θα εισάγουμε τα στοιχεία ενός ατόμου με βάρος 53 κιλά και ύψος 1,61 μέτρα:

```
>> BMI(53,1.61)
ans =
    20.4467
```

Το MATLAB αμέσως μας εμφανίζει το δείκτη μάζας σώματος, το οποίο έχει αναθέσει στη μεταβλητή `ans`, χωρίς άλλες πρόσθετες πληροφορίες. Το αποτέλεσμα εμφανίζεται στην προκαθορισμένη μεταβλητή `ans`, διότι δεν του καθορίσαμε κάποια μεταβλητή στην οποία θα αποθηκεύσει το αποτέλεσμα, ύστερα από την εκτέλεση της εντολής `>> BMI(53,1.61)`. Εκτελώντας την ίδια εντολή, αλλά ορίζοντας στην αρχή τη μεταβλητή που θα αποθηκευτεί το αποτέλεσμα, παίρνουμε ακριβώς το αποτέλεσμα που επιθυμούμε:

```
>> bmi=BMI(53,1.61)
```

```
bmi =  
20.4467
```

Παράδειγμα 2

Στο παρακάτω παράδειγμα θα ορίσουμε ένα αρχείο συνάρτησης με το όνομα “**MO**” (Μέσος Όρος). Η συνάρτηση αυτή θα δέχεται ως όρισμα εισόδου ένα διάνυσμα αριθμών και θα επιστρέφει δύο τιμές, τη μέση τιμή και την τυπική απόκλιση του διανύσματος, τα οποία αποτελούν και τα ορίσματα εξόδου. Αρχικά υπολογίζει το μήκος του διανύσματος, το οποίο ισοδυναμεί με το πλήθος των στοιχείων του. Στη συνέχεια υπολογίζει το πηλίκο του αθροίσματος των στοιχείων του προς το πλήθος των στοιχείων του και το αναθέτει στη μεταβλητή της μέσης τιμής. Ο υπολογισμός της τυπικής απόκλισης γίνεται με τη χρήση του τύπου $\sigma =$

$$\sqrt{\frac{\sum(\text{στοιχείο}-\text{μέση τιμή})^2}{\text{πλήθος στοιχείων πίνακα}}}$$

```
function [mean,sd] = MO(pinakas)
```

→ Επικεφαλίδα

```
% It returns the mean and standard deviation of  
an input vector.
```

→ Σχόλια(comments)

```
plithos = length(pinakas);
```

```
mean = sum(pinakas)/plithos;
```

```
sd = sqrt(sum((pinakas-mean).^2/plithos));
```

→ Εντολές (statements)

```
end
```

→ Εντολή τερματισμού
συνάρτησης

Επειδή στην επικεφαλίδα της συνάρτησης έχουν οριστεί δύο ορίσματα εξόδου, η κλήση της πρέπει να γίνει με τη χρήση ενός διανύσματος με δύο μεταβλητές (όσα είναι και τα ορίσματα εξόδου), έτσι ώστε μετά την εκτέλεση της να μας επιστραφούν και τα δύο ορίσματα εξόδου. Σε αντίθετη περίπτωση το MATLAB θα μας επιστρέψει μόνο την τιμή του πρώτου ορίσματος εξόδου.

```
>> [mesi,tipiki_aplokisi]=MO([10,10,40])
```

```
mesi =  
20
```

```
tipiki_aplokisi =  
14.1421
```

11.3. Δομές Ελέγχου Ροής Προγράμματος

Οι δομές ελέγχου ροής του προγράμματος είναι σύνολα εντολών, τα οποία χρησιμοποιούνται συνήθως στις περιπτώσεις όπου ένα κομμάτι του κώδικα εκτελείται συχνά ή

στις περιπτώσεις όπου επαναλαμβάνονται τα ίδια βήματα πολλές φορές. Χωρίζονται σε δύο κατηγορίες, τις δομές επιλογής και τις δομές επανάληψης. Στην πρώτη κατηγορία ανήκει οι εντολές `if` και `switch`, ενώ στη δεύτερη οι εντολές `for` και `while`, που αναλύονται παρακάτω.

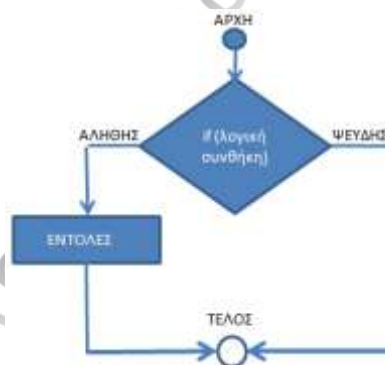
11.4. Η Εντολή `if`

Η **εντολή `if`** μας δίνει τη δυνατότητα να δημιουργήσουμε προγράμματα τα οποία θα αλληλεπιδρούν με το χρήστη και θα παράγουν διαφορετικά δεδομένα εξόδου ανάλογα με τα δεδομένα εισόδου. Αυτό πραγματοποιείται μέσα από μία διαδικασία επιλογής των εκάστοτε εντολών, που εκτελούνται, ύστερα από τον έλεγχο ικανοποίησης μιας συνθήκης, η οποία παίρνει την τιμή αληθής ή ψευδής. Η **εντολή `if`** εμφανίζεται με τρεις διαφορετικές μορφές. Την εντολή `if...end`, την εντολή `if...else...end` και την εντολή `if...elseif...end`. Κοινά χαρακτηριστικά στη σύνταξη των διαφορετικών μορφών εμφάνισης και στις τρεις περιπτώσεις αποτελούν η αρχική και τελική δήλωση των δεσμευμένων λέξεων `if` και `end` αντίστοιχα.

Εντολή `if`

Η απλούστερη μορφή της **εντολής `if`** είναι η πρώτη, όπου ελέγχεται η συνθήκη και στην περίπτωση όπου αυτή ισχύει, δηλαδή είναι αληθής, τότε εκτελούνται οι εντολές που περιλαμβάνει το σύνολο εντολών. Αν η συνθήκη είναι ψευδής, οι εντολές δεν εκτελούνται. Η σύνταξη της είναι η εξής:

```
if συνθήκη
    σύνολο
εντολών;
end
```



Παράδειγμα

Στο παρακάτω παράδειγμα βλέπουμε τη χρήση της **εντολής `if`** για τον έλεγχο της μεταβλητής `x`, η οποία περιέχει έναν πραγματικό αριθμό.

```
>> if x>0
disp('Ο αριθμός είναι θετικός');
end
```

Αν το περιεχόμενο της μεταβλητής `x` είναι θετικός αριθμός τότε εκτυπώνεται το μήνυμα «Ο αριθμός είναι θετικός», ενώ στην αντίθετη περίπτωση δεν εκτυπώνεται τίποτα.

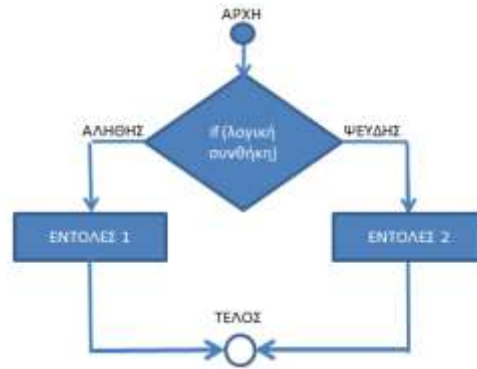
Εντολή `if...else...end`

Στην περίπτωση όπου επιθυμούμε την εκτέλεση κάποιων εντολών όταν η συνθήκη δεν ικανοποιείται, δηλαδή είναι ψευδής, χρησιμοποιούμε τη δεύτερη μορφή της **εντολής `if`**. Με αυτό τον τρόπο εξασφαλίζουμε ότι ο χρήστης θα λάβει το ανάλογο μήνυμα ή θα εκτελεστούν οι ανάλογες εντολές, όποιο και αν είναι το αποτέλεσμα που θα μας επιστρέψει ο έλεγχος της συνθήκης. Στην περίπτωση όπου η συνθήκη ισχύει, τότε εκτελούνται οι εντολές που βρίσκονται στο πρώτο σύνολο εντολών, διαφορετικά εκτελούνται οι εντολές του δεύτερου συνόλου εντολών. Η σύνταξη της εντολής είναι η ακόλουθη:


```

if συνθήκη
    σύνολο εντολών1;
else
    σύνολο εντολών2;
end

```



Παράδειγμα

Κατά αυτόν τον τρόπο το παραπάνω παράδειγμα μετατρέπεται σε:

```

>> if x>0
disp('Ο αριθμός είναι θετικός');
else
disp('Ο αριθμός είναι αρνητικός');
end

```

Εντολή if...elseif...end

Στην περίπτωση όπου οι εναλλακτικές επιλογές είναι περισσότερες από δύο, χρησιμοποιείται η τρίτη μορφή της **εντολής if**. Εδώ χρησιμοποιούνται εμφωλευμένες **εντολές if**, οι οποίες περιέχονται η μία μέσα στην άλλη. Μεγάλη προσοχή χρειάζεται κατά τη γραφή του κώδικα, αφού εδώ παρατηρούνται τα περισσότερα λάθη, κυρίως στην εντολή elseif.

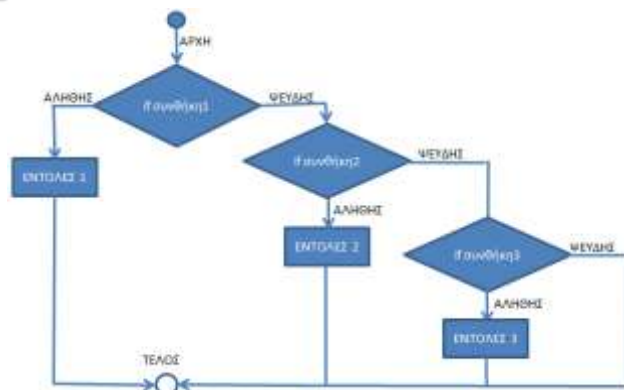
! Η εντολή elseif γράφεται ως μια λέξη, δηλαδή δεν πρέπει να υπάρχει κενό μεταξύ του else και του if.

Σε αντίθετη περίπτωση το MATLAB κατανόει των κώδικα ως δύο απλές **εντολές if** και περιμένει τη δεύτερη δήλωση της δεσμευμένης λέξης **end** προκειμένου να εκτελέσει το σύνολο εντολών. Παρακάτω παρουσιάζεται η σύνταξη της εντολής **if...elseif...end**:

```

if συνθήκη1
    σύνολο εντολών1;
elseif συνθήκη2
    σύνολο εντολών2;
elseif συνθήκη3
    σύνολο εντολών3;
else
    σύνολο εντολών4;
end

```



Παράδειγμα

Αν θέλουμε να βελτιστοποιήσουμε το παράδειγμα με τη μεταβλητή x , λαμβάνοντας υπόψη και την περίπτωση όπου η μεταβλητή x ταυτίζεται με τον αριθμό μηδέν, παρατηρούμε ότι είναι απαραίτητη η χρήση των εμφωλευμένων **εντολών if**. Έτσι, το παράδειγμα ολοκληρώνεται ως εξής:

```
>> if x>0
    disp('Ο αριθμός είναι θετικός');
elseif x==0
    disp('Ο αριθμός είναι το μηδέν');
else
    disp('Ο αριθμός είναι αρνητικός');
end
```

Στην περίπτωση όπου η μεταβλητή x είναι θετικός αριθμός το MATLAB θα μας εμφανίσει το μήνυμα «Ο αριθμός είναι θετικός», αλλιώς στην περίπτωση όπου η μεταβλητή x είναι ο αριθμός μηδέν θα μας εμφανίσει το μήνυμα «Ο αριθμός είναι το μηδέν» και σε κάθε άλλη περίπτωση, δηλαδή εάν ο αριθμός είναι αρνητικός, θα μας εμφανίσει το μήνυμα «Ο αριθμός είναι αρνητικός».

11.5. Εντολή for

Η **εντολή for** είναι μία από τις εντολές ελέγχου ροής προγράμματος που διαθέτει το MATLAB. Με τη χρήση της **εντολής for** μας δίνεται η δυνατότητα να επαναλάβουμε την εκτέλεση ενός συνόλου εντολών συγκεκριμένες φορές. Έτσι δημιουργείται μία κυκλική διαδικασία, η οποία αποκαλείται βρόχος και εκτελείται επαναληπτικά μέχρι να παραβιαστεί το κριτήριο της επανάληψης. Η δομή της **εντολής for** είναι η ακόλουθη:

```
for αρχική τιμή δείκτη : βήμα-μεταβολής δείκτη : τελική τιμή δείκτη
    εντολή;
end
```

Η αρχική τιμή δείκτη αποδίδει σε μία μεταβλητή ελέγχου του βρόχου (συνήθως τη μεταβλητή i) μία αρχική τιμή. Το βήμα- μεταβολής του δείκτη καθορίζει το πώς θα μεταβάλλεται η μεταβλητή ελέγχου του βρόχου ύστερα από κάθε επανάληψη. Η τελική τιμή δείκτη είναι η τελική τιμή που μπορεί να δεχτεί η μεταβλητή ελέγχου του βρόχου. Οι τιμές που μπορούν να δεχτούν η αρχική τιμή δείκτη, το βήμα-μεταβολής δείκτη και η τελική τιμή δείκτη είναι ακέραιοι ή πραγματικοί αριθμοί, θετικοί ή αρνητικοί.

Η μεταβλητή ελέγχου συγκρίνεται στην αρχή και ύστερα από κάθε επανάληψη με την τελική τιμή δείκτη. Εφόσον βρίσκεται εντός των ορίων της αρχικής και τελικής τιμής δείκτη το σύνολο των εντολών που βρίσκεται εντός του βρόχου εκτελείται. Στη συνέχεια, η μεταβλητή ελέγχου μεταβάλλεται σύμφωνα με το βήμα-μεταβολής δείκτη, το οποίο μπορεί να είναι είτε θετικός αριθμός, δηλαδή η μεταβλητή να αυξάνεται, είτε αρνητικός αριθμός, δηλαδή η μεταβλητή ελέγχου να μειώνεται. Στην περίπτωση όπου το βήμα-μεταβολής δείκτη είναι θετικό, η αρχική τιμή δείκτη πρέπει να είναι μικρότερη από την τελική τιμή δείκτη, ενώ στην περίπτωση όπου το βήμα-μεταβολής δείκτη είναι αρνητικό, η αρχική τιμή δείκτη πρέπει να είναι μεγαλύτερη από την τελική τιμή δείκτη. Η επανάληψη των κύκλων σταματά μόλις η μεταβλητή ελέγχου λάβει τιμές εκτός αυτών των ορίων.

Παράδειγμα 1

Στο παρακάτω παράδειγμα χρησιμοποιείται η **εντολή for** για την τη καταχώριση του αριθμού μηδέν στις πρώτες πέντε θέσεις του διανύσματος μίας γραμμής $A=[1\ 2\ 3\ 4\ 5\ 6\ 36\ 88\ 99\ 97\ 5\ 4\ 3]$.

```
>> for i=1:1:5
    A(i)=0
end
```

Με την παραπάνω εντολή δημιουργείται η μεταβλητή ελέγχου i , η οποία δέχεται αρχικά την τιμή ένα. Ελέγχεται αν η τιμή της μεταβλητή ελέγχου i είναι μικρότερη από την τελική τιμή δείκτη (αφού του βήμα-μεταβολής είναι θετικό) και στη συνέχεια εφόσον ο έλεγχος επιστρέψει θετικό αποτέλεσμα, δηλαδή αληθές, καταχωρείται ο αριθμός μηδέν στην πρώτη θέση ($i=1$) του διάνυσματος, με την εντολή $A(i)=0$. Στη συνέχεια, η μεταβλητή ελέγχου i , αυξάνεται κατά μία μονάδα, όπως ορίζει το βήμα-μεταβολής δείκτη και έτσι η μεταβλητή i δέχεται την τιμή δύο ($i=2$). Έπειτα, ελέγχεται η συνθήκη για το αν η τιμή της μεταβλητή ελέγχου i είναι μικρότερη από την τελική τιμή δείκτη και αν το αποτέλεσμα του ελέγχου είναι αληθές τότε καταχωρείται ο αριθμός μηδέν στη δεύτερη θέση ($i=2$) του διάνυσματος. Η μεταβλητή ελέγχου αυξάνεται κατά μία μονάδα και επαναλαμβάνεται η παραπάνω διαδικασία έως ότου η τιμή της μεταβλητή ελέγχου i είναι ξεπεράσει την τελική τιμή δείκτη ($i>5$). Στον παρακάτω πίνακα παρουσιάζονται αναλυτικά όλα τα βήματα της εντολής `for` με τις εκάστοτε τιμές που λαμβάνει η μεταβλητή i και το διάνυσμα A .

i	Συνθήκη ελέγχου	Αποτέλεσμα ελέγχου	Εκτέλεση εντολής	Διάνυσμα A
-	-	-	-	$A=[1\ 2\ 3\ 4\ 5\ 6\ 36\ 88\ 99\ 97\ 5\ 4\ 3]$
1	$1 \leq 5$	Αληθές	$A(1)=0$	$A=[0\ 2\ 3\ 4\ 5\ 6\ 36\ 88\ 99\ 97\ 5\ 4\ 3]$
2	$2 \leq 5$	Αληθές	$A(2)=0$	$A=[0\ 0\ 3\ 4\ 5\ 6\ 36\ 88\ 99\ 97\ 5\ 4\ 3]$
3	$3 \leq 5$	Αληθές	$A(3)=0$	$A=[0\ 0\ 0\ 4\ 5\ 6\ 36\ 88\ 99\ 97\ 5\ 4\ 3]$
4	$4 \leq 5$	Αληθές	$A(4)=0$	$A=[0\ 0\ 0\ 0\ 5\ 6\ 36\ 88\ 99\ 97\ 5\ 4\ 3]$
5	$5 \leq 5$	Αληθές	$A(5)=0$	$A=[0\ 0\ 0\ 0\ 0\ 6\ 36\ 88\ 99\ 97\ 5\ 4\ 3]$
6	$6 < 5$	Ψευδές	καμίας	Δεν αλλάζει

Αξίζει να σημειωθεί πως η μεταβλητή ελέγχου μπορεί να λάβει οποιοδήποτε έγκυρο όνομα μεταβλητής εμείς επιθυμούμε και πως το βήμα-μεταβολής δείκτη μπορεί να παραλειφθεί όταν ισούται με τη μονάδα. Οπότε η πρώτη γραμμή του παραπάνω παραδείγματος θα μπορούσε να γραφεί και ως `>> for i=1:5`.

Παράδειγμα 2

Στο παρακάτω παράδειγμα υπολογίζεται το άθροισμα των πρώτων εκατό ζυγών αριθμών. Για το λόγο αυτό, η αρχική τιμή δείκτη είναι ο αριθμός δύο, αφού αυτός είναι ο πρώτος ζυγός αριθμός, το βήμα-μεταβολής είναι ο ακέραιος αριθμός δύο και η τελική τιμή δείκτη είναι ο αριθμός εκατό. Επίσης ως όνομα της μεταβλητής δείκτη ορίστηκε η λατινική λέξη *zigos_arithmos*.

```
>> sum=0;
for zigos_arithmos=2:2:100
    sum=sum+zigos_arithmos;
end

disp(sum)
```

Το αποτέλεσμα που μας επιστρέφει το MATLAB είναι ο αριθμός 2550. Παρατηρούμε πως για τον υπολογισμό του αθροίσματος ήταν απαραίτητος ο ορισμός μίας ακόμα μεταβλητής με το όνομα *sum*, στην οποία αποθηκεύεται κάθε φορά το άθροισμα των ζυγών αριθμών. Η εκχώρηση του αριθμού μηδέν στη μεταβλητή αυτή γίνεται πριν τον **βρόχο for**, ενώ η εκχώρηση του αποτελέσματος της πρόσθεσης γίνεται μέσα σε αυτόν.

Πιο αναλυτικά, η μεταβλητή *sum* αρχικά λαμβάνει την τιμή μηδέν, εφόσον δεν έχει προστεθεί ακόμα σε αυτήν κάποιος ζυγός αριθμός. Στη συνέχεια η μεταβλητή ελέγχου λαμβάνει την αρχική τιμή δείκτη, η οποία είναι ο πρώτος ζυγός αριθμός, δηλαδή το δύο. Υπολογίζεται το άθροισμα της μεταβλητής *sum* και της μεταβλητής *zigos_arithmos*, δηλαδή του προηγούμενου αθροίσματος και του αριθμού δύο. Έπειτα, η μεταβλητή ελέγχου *zigos_arithmos* λαμβάνει την

τιμή τέσσερα, αφού προστίθεται σε αυτήν το βήμα-μεταβολής δείκτη (το δύο) και ελέγχεται αν είναι μικρότερη από την τελική τιμή δείκτη (το εκατό). Εφόσον το αποτέλεσμα είναι αληθές προβαίνει πάλι στην πρόσθεση του αριθμού τέσσερα στο ήδη υπάρχον άθροισμα. Η επανάληψη των κύκλων σταματά όταν η μεταβλητή ελέγχου λάβει την τιμή εκατόν δύο και ξεπεράσει έτσι την τελική τιμή δείκτη, επιστρέφοντας ψευδές αποτέλεσμα στον έλεγχο που πραγματοποιείται πριν την εκτέλεση των εντολών.

Εμπλουτίζοντας το παραπάνω παράδειγμα προκειμένου να είναι πιο φιλικό προς το χρήστη, θα χρησιμοποιήσουμε την **εντολή disp** από το κεφάλαιο 11.2 και την **εντολή num2str()** από το κεφάλαιο 5.4. Με την **εντολή disp** θα εμφανίσουμε ένα μήνυμα κειμένου στο χρήστη, το οποίο θα συνοδεύει το αποτέλεσμα και θα τον πληροφορεί για την πληροφορία που εμφανίζεται στην οθόνη του. Επειδή η μεταβλητή *sum*, η οποία περιέχει το αποτέλεσμα που υπολογίσαμε, είναι αριθμός δε μπορεί να εισαχθεί σε αυτή τη μορφή ως δεύτερη παράμετρος στην **εντολή disp**. Έτσι, είναι απαραίτητη η μετατροπή της σε ακολουθία χαρακτήρων και για τον ευκολότερο χειρισμό της, η αποθήκευσής της σε μία νέα μεταβλητή με το όνομα *result*. Τέλος, γίνεται χρήση του ελληνικού ερωτηματικού (Κεφάλαιο 2.4) στις κατάλληλες εντολές προκειμένου να αποκρυφθούν οι περιττοί υπολογισμοί και να εμφανιστεί μόνο το τελικό αποτέλεσμα στο χρήστη.

```
>> sum=0;

for zigos_arithmos=2:2:100
    sum=sum+zigos_arithmos;
end

result=num2str(sum);
disp(['Το άθροισμα των εκατό πρώτων ζυγών αριθμών είναι: ',result]);
```

Παρακάτω βλέπουμε το αποτέλεσμα που μας επιστρέφει το MATLAB:

Το άθροισμα των εκατό πρώτων ζυγών αριθμών είναι: 2550

Μερικές φορές για την επίλυση σύνθετων προβλημάτων είναι απαραίτητη η χρήση εμφωλευμένων βρόχων. Σε αυτή την περίπτωση ο ένας βρόχος εσωκλείεται σε έναν άλλο. Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό και δεν επιτρέπεται η χρησιμοποίηση της ίδιας μεταβλητής ως μετρητής σε δύο ή περισσότερους βρόχους όταν ο ένας περιλαμβάνει σε έναν άλλο. Τέλος, ο βρόχος που ξεκινάει τελευταίος πρέπει να ολοκληρώνεται και πρώτος.

Παράδειγμα 3

Στο παρακάτω παράδειγμα γίνεται χρήση εμφωλευμένων βρόχων για την εκτύπωση του περιεχομένου του πίνακα $A = \begin{bmatrix} 2 & 5 & 6 \\ 3 & 1 & 9 \end{bmatrix}$. Αρχικά ορίζεται ο *πίνακας A* με το περιεχόμενό του. Αποθηκεύεται στη μεταβλητή *megehtos_pinaka*, με τη χρήση της **συνάρτησης size** (Κεφάλαιο 8.6.2), το πλήθος των γραμμών και στηλών του πίνακα και με τη βοήθεια αυτής της μεταβλητής, αποθηκεύεται ξεχωριστά το πλήθος των γραμμών και στηλών στις μεταβλητές *grammes_pinaka*, *stiles_pinaka* αντίστοιχα. Στη συνέχεια χρησιμοποιούμε την **εντολή for** για τη δημιουργία μιας μεταβλητής ελέγχου (i), η οποία θα δέχεται κάθε φορά τον αριθμό της γραμμής που εξετάζουμε. Με την εμφωλευμένη **εντολή for** δημιουργείται η δεύτερη μεταβλητή ελέγχου (j), η οποία θα δέχεται κάθε φορά τον αριθμό της στήλης που εξετάζουμε. Οι εντολές *grammi=int2str(i)*; και *stili=int2str(j)*; μετατρέπουν τον αριθμό της γραμμής και στήλης σε ακολουθία χαρακτήρων χωρίς να εμφανίζουν το αποτέλεσμα, προκειμένου να εισαχθούν ως παράμετροι στην εντολή *disp*. Το ίδιο συμβαίνει και με τη μεταβλητή *perioxomeno*, η οποία δέχεται ως είσοδο το περιεχόμενο του κελιού που εξετάζουμε στον πίνακα και το μετατρέπει σε ακολουθία χαρακτήρων. Τέλος, αξίζει να σημειωθεί ότι οι τρεις παραπάνω εντολές θα

μπορούσαν να παραλειφθούν και να εμφανιζόταν απλά το περιεχόμενο του κελιού στο χρήστη χωρίς κάποιο μήνυμα κειμένου στην ίδια γραμμή.

```
A=[2 5 6 ; 3 1 9 ]

megehtos_pinaka=size(A)
grammes_pinaka=megehtos_pinaka(1);
stiles_pinaka=megehtos_pinaka(2);

for i=1:grammes_pinaka
    for j=1:stiles_pinaka

        grammi=int2str(i);
        stili=int2str(j);
        periexomeno=num2str( A(i,j));

        disp(['Θέση (',grammi,',',stili,') =',periexomeno]);
    end
end
```

Το MATLAB θα επιστρέψει στο χρήστη:

```
A =
    2    5    6
    3    1    9

megehtos_pinaka =
    2    3

Θέση (1,1) =2
Θέση (1,2) =5
Θέση (1,3) =6
Θέση (2,1) =3
Θέση (2,2) =1
Θέση (2,3) =9
```

Παρακάτω παρουσιάζονται τα περιεχόμενα των μεταβλητών που υπολογίζονται για τη διευκόλυνση των εμφωλευμένων εντολών for:

```
A= 2 5 6
    3 1 9

megehtos_pinaka= 2 3
grammes_pinaka=2
stiles_pinaka=3
```

Στον ακόλουθο πίνακα παρουσιάζονται αναλυτικά όλα τα βήματα των εμφωλευμένων εντολών for με τις εκάστοτε τιμές που λαμβάνουν οι παραπάνω μεταβλητές και το μήνυμα κειμένου που εμφανίζεται στο χρήστη.

i	j	grammi (γραμμή)	stili (στήλη)	Περιεχόμενο πίνακα A	Μήνυμα κειμένου
1	1	1	1	2	Θέση (1,1) =2
1	2	1	2	5	Θέση (1,2) =5
1	3	1	3	6	Θέση (1,3) =6

2	1	2	1	3	Θέση (2,1) =3
2	2	2	2	1	Θέση (2,2) =1
2	3	2	3	9	Θέση (2,3) =9

Παράδειγμα 4

Στο παράδειγμα που ακολουθεί γίνεται χρήση εμφωλευμένων βρόχων για την εύρεση της μέγιστης και της ελάχιστης τιμής του πίνακα $A = \begin{bmatrix} 2 & 5 & 6 \\ 3 & 1 & 9 \end{bmatrix}$. Απαραίτητη είναι η αρχικοποίηση του ελάχιστου και του μέγιστου στοιχείου του πίνακα. Για το λόγο αυτό, ορίζεται αυθαίρετα το πρώτο στοιχείο του πίνακα ως το ελάχιστο και το μέγιστο ταυτόχρονα. Όπως καταλαβαίνουμε κάτι τέτοιο δεν είναι εφικτό, εκτός της περίπτωσης όπου ο πίνακας περιέχει μόνο ένα στοιχείο. Ωστόσο δε μας επηρεάζει, αφού οι τιμές των μεταβλητών που περιέχουν τη μέγιστη και ελάχιστη τιμή του πίνακα θα αλλάξουν σταδιακά. Καθώς ο έλεγχος πραγματοποιείται για κάθε ένα στοιχείο του πίνακα, εξετάζεται αν αυτό είναι μεγαλύτερο από το περιεχόμενο της μεταβλητής Max η οποία κρατά κάθε φορά το μέγιστο στοιχείο του πίνακα. Αν ο έλεγχος προκύψει αληθής, τότε η μεταβλητή Max αποθηκεύει την τιμή του στοιχείου αυτού ως το μεγαλύτερο. Το ίδιο πραγματοποιείται και την εύρεση του ελάχιστου στοιχείου με μόνη διαφορά τη συνθήκη ελέγχου, η οποία ελέγχει αν το στοιχείο που εξετάζουμε είναι μικρότερο της μεταβλητής Min, η οποία κρατά το ελάχιστο στοιχείο του πίνακα.

```
Min = A(1,1); Max = A(1,1);
```

→ Ορισμός του πρώτου στοιχείου του πίνακα ως το ελάχιστο, αλλά και το μέγιστο.

```
for i=1:size(A,1)
```

→ Αναζήτηση κατά γραμμές.

```
for j=1:size(A,2)
```

→ Αναζήτηση κατά στήλες.

```
if A(i,j)>Max
    Max = A(i,j);
end
```

→ Έλεγχος αν στοιχείο > μεταβλητή Max και ανάθεση στοιχείου στη Max, αν έλεγχος είναι αληθής.

```
if A(i,j)<Min
    Min = A(i,j);
end
```

→ Έλεγχος αν στοιχείο < μεταβλητή Min και ανάθεση στοιχείου στη Min, αν έλεγχος είναι αληθής.

```
end
end
```

```
Max=num2str(Max);
Min=num2str(Min);
```

→ Μετατροπή μεταβλητών Max και Min σε ακολουθία χαρακτήρων.

```
disp(['Το μέγιστο στοιχείο
του πίνακα είναι το',Max]);
```

```
disp(['Το ελάχιστο στοιχείο
του πίνακα είναι το',Min]);
```

→ Εμφάνιση αποτελέσματος.

11.6. Εντολή while

Η **εντολή while** είναι και αυτή μία από τις εντολές βρόχου που διαθέτει το MATLAB. Με τη χρήση αυτής της εντολής μας δίνεται η δυνατότητα να επαναλάβουμε την εκτέλεση ενός συνόλου εντολών όσο ικανοποιείται μια συνθήκη. Η εντολή αυτή χρησιμοποιείται στις περιπτώσεις όπου δεν είναι γνωστός εκ των προτέρων ο ακριβής αριθμός των επαναλήψεων των εντολών και άρα η δομή ελέγχου for δεν μπορεί να χρησιμοποιηθεί. Ένα επίσης σημαντικό χαρακτηριστικό που την ξεχωρίζει από τις άλλες δομές ελέγχου ροής του προγράμματος είναι ότι με τη χρήση της **εντολής while** μπορούν να εκφραστούν όλα προβλήματα επανάληψης. Η γενική σύνταξη της εντολής είναι η εξής:

```
while συνθήκη
    σύνολο εντολών
end
```

Όπως παρατηρούμε η συνθήκη που ελέγχει τον αριθμό των επαναλήψεων βρίσκεται στην αρχή της επανάληψης. Ο βρόχος επαναλαμβάνεται όσο η συνθήκη είναι αληθής και τερματίζεται όταν η συνθήκη γίνει ψευδής. Επομένως υπάρχει η πιθανότητα να μην εκτελεστεί ποτέ το σύνολο εντολών, αν η συνθήκη ελέγχου είναι εξαρχής ψευδής, αλλά και η πιθανότητα ο βρόχος να εκτελείται συνεχώς στην περίπτωση όπου η συνθήκη ήταν αληθής εξαρχής και δε μεταβάλλεται. Στην περίπτωση του ατέρμονα βρόχου, όπως ονομάζεται ο βρόχος όταν εκτελείται επάπειρον, είναι απαραίτητη η εκχώρηση μίας εντολής μέσα στο βρόχο η οποία θα μεταβάλλει την τιμή της συνθήκης (αυξάνει ή μειώνει), προκειμένου να την μετατρέψει σε ψευδή και να τερματιστεί έτσι ο βρόχος. Τέλος, απαραίτητη είναι μία εντολή, η οποία θα αποδίδει μία αρχική τιμή στη μεταβλητή ελέγχου πριν τον βρόχο while.

Παράδειγμα 1

Για την καλύτερη κατανόηση της **εντολής while** ακολουθεί ένα παράδειγμα το οποίο εμφανίζει δέκα φορές το μήνυμα κειμένου «Παρακαλώ δώστε ένα αριθμό:» και στη συνέχεια αφού ο χρήστης έχει εισάγει τον αριθμό που επιθυμεί, του εμφανίζει την εκχώρησή του στην οθόνη ύστερα από το μήνυμα κειμένου «Ο αριθμός που εισάγατε είναι ο:».

```
i=0;
while i<10
    number=input('Παρακαλώ δώστε ένα αριθμό: ');

    aritos= num2str(number);
    disp([' Ο αριθμός που εισάγατε είναι ο : ',aritos]);
    i=i+1;
end
```

Το MATLAB θα επιστρέφει:

```
Παρακαλώ δώστε ένα αριθμό: 1
Ο αριθμός που εισάγατε είναι ο : 1
Παρακαλώ δώστε ένα αριθμό: 2
Ο αριθμός που εισάγατε είναι ο : 2
Παρακαλώ δώστε ένα αριθμό: 3
Ο αριθμός που εισάγατε είναι ο : 3
Παρακαλώ δώστε ένα αριθμό: 4
Ο αριθμός που εισάγατε είναι ο : 4
Παρακαλώ δώστε ένα αριθμό: 5
Ο αριθμός που εισάγατε είναι ο : 5
Παρακαλώ δώστε ένα αριθμό: 10
```

Ο αριθμός που εισάγατε είναι ο : 10
 Παρακαλώ δώστε ένα αριθμό: 11
 Ο αριθμός που εισάγατε είναι ο : 11
 Παρακαλώ δώστε ένα αριθμό: 12
 Ο αριθμός που εισάγατε είναι ο : 12
 Παρακαλώ δώστε ένα αριθμό: 13
 Ο αριθμός που εισάγατε είναι ο : 13
 Παρακαλώ δώστε ένα αριθμό: 15
 Ο αριθμός που εισάγατε είναι ο : 15

Στην πρώτη εντολή δημιουργείται η μεταβλητή ελέγχου i και της αποδίδεται η αρχική τιμή μηδέν. Στη συνέχεια ελέγχεται η συνθήκη $i < 10$ ($0 < 10$), που ισχύει και εκτελείται η εντολή `input` με την οποία εμφανίζεται το μήνυμα κειμένου στο χρήστη και αποθηκεύει την τιμή που έχει δώσει στη μεταβλητή `number`. Όπως έχουμε προαναφέρει, μετατρέπεται η μεταβλητή `number` σε αλφαριθμητικό και εισάγεται ως δεύτερη παράμετρος στην εντολή `disp`, όπου και εμφανίζεται το μήνυμα στο χρήστη. Πριν το τέλος της πρώτης επανάληψης, η μεταβλητή ελέγχου i αυξάνεται και έτσι ξαναελέγχεται η συνθήκη $i < 10$ ($1 < 10$). Η διαδικασία αυτή επαναλαμβάνεται μέχρι η μεταβλητή i να πάρει την τιμή δέκα ($i = 10$) και έτσι ο έλεγχος από την συνθήκη $i < 10$ ($10 < 10$) να επιστρέψει ψευδής.

Παράδειγμα 2

Όπως αναφέραμε και προηγουμένως ένα σημαντικό πλεονέκτημα της εντολής `while` είναι ότι μπορεί να επιλύσει οποιοδήποτε πρόβλημα επανάληψης. Για το λόγο αυτό παρακάτω παρουσιάζεται η αντίστοιχη επίλυση του τρίτου παραδείγματος από την εντολή `for`. Υπενθυμίζεται ο πίνακας $A = \begin{bmatrix} 2 & 5 & 6 \\ 3 & 1 & 9 \end{bmatrix}$ του οποίου εμφανίζονται τα περιεχόμενα στο χρήστη:

```
A=[2 5 6 ; 3 1 9]
```

```
megehtos_pinaka=size(A)
grammes_pinaka=megehtos_pinaka(1);
stiles_pinaka=megehtos_pinaka(2);

i=1;
j=1;
while i<=grammes_pinaka
    while j<=stiles_pinaka
        grammi=int2str(i);
        stili=int2str(j);
        periexomeno=num2str( A(i,j));

        disp(['Θέση (' ,grammi,',',stili,') =',periexomeno]);
        j=j+1;
    end
    i=i+1;
    j=1;
end
```

Η πρώτη αλλαγή του παραπάνω κώδικα σε σχέση με αυτού της εντολής `for` είναι η δήλωση των μεταβλητών i, j και η εκχώριση των αρχικών τιμών τους πριν τους βρόχους `while`. Η δεύτερη αλλαγή είναι η εισαγωγή της εντολής `j=j+1`; στο τέλος του εσωτερικού βρόχου, προκειμένου να αυξάνεται ο αριθμός των στηλών (μέσω της μεταβλητής j) που εξετάζουμε. Έξω από τον εσωτερικό βρόχο και αφού έχουμε επομένως ελέγχει όλες τις στήλες του πίνακα για την γραμμή που εξετάζαμε, αυξάνεται η μεταβλητή i , προκειμένου να αυξηθεί και ο αριθμός των γραμμών αντίστοιχα. Σημαντική είναι η αμέσως επόμενη εντολή `j=1`; η οποία ξανακαταχωρεί τον

αριθμό ένα στη μεταβλητή με την οποία εξετάζουμε τις στήλες. Αν η παραπάνω εντολή είχε παραλειφθεί, το MATLAB θα αδυνατούσε να εμφανίσει τις επόμενες γραμμές του πίνακα, αφού η μεταβλητή *j* θα είχε την τελευταία τιμή που της είχε αποδοθεί, δηλαδή *j*=4. Αυτό θα είχε ως αποτέλεσμα να ξεκινούσε τον έλεγχο από την αμέσως επόμενη γραμμή αλλά από την τέταρτη στήλη, η οποία δεν υπάρχει! Επομένως στο έλεγχο *while j<=stiles_pinaka* η συνθήκη θα ήταν ψευδής και θα τερματιζόταν ο βρόχος.

Ο πίνακας με τις τιμές που λαμβάνουν οι μεταβλητές *i*, *j*, *grammi*, *stili* και το μήνυμα κειμένου που εμφανίζεται κάθε φορά στο χρήστη παραμένει ίδιος:

<i>i</i>	<i>j</i>	<i>grammi</i> (γραμμή)	<i>stili</i> (στήλη)	Περιεχόμενο πίνακα A	Μήνυμα κειμένου
1	1	1	1	2	Θέση (1,1) =2
1	2	1	2	5	Θέση (1,2) =5
1	3	1	3	6	Θέση (1,3) =6
2	1	2	1	3	Θέση (2,1) =3
2	2	2	2	1	Θέση (2,2) =1
2	3	2	3	9	Θέση (2,3) =9

11.7. Εντολή switch

Η εντολή **switch** είναι και αυτή μία εντολή ελέγχου ροής προγράμματος που διαθέτει το MATLAB. Ανήκει στην κατηγορία δομών επιλογής όπως η εντολή **if** και μας επιτρέπει να κατευθύνουμε τη ροή του προγράμματος εκτελώντας ένα συγκεκριμένο σύνολο εντολών, ανάλογα με την τιμή μιας συνθήκης ελέγχου. Η διαφορά τους είναι πως στην εντολή **switch**, οι μεταβλητές ελέγχου παίρνουν τιμές μόνο από ένα σύνολο διακριτών τιμών, ενώ στην εντολή **if** οι μεταβλητές ελέγχου μπορούν να πάρουν τιμές είτε από ένα σύνολο διακριτών τιμών είτε από ένα συνεχές σύνολο τιμών. Η σύνταξη της εντολής είναι η εξής:

```

switch μεταβλητή ελέγχου
  case τιμή1 ή {τιμή11, τιμή12, ...}
    σύνολο εντολών1
  case τιμή2 ή {τιμή21, τιμή22, ...}
    σύνολο εντολών2
  ...
  otherwise
    σύνολο εντολών
end

```

Όπως παρατηρούμε, η σύνταξη της ξεκινάει και τελειώνει με τις δεσμευμένες λέξεις **switch**, **end**, αποτελείται από μία σειρά από **ετικέτες case** και μία προαιρετική περίπτωση την **otherwise**. Η λέξη κλειδί **switch** ακολουθείται από μία μεταβλητή ελέγχου, της οποίας η τιμή συγκρίνεται με κάθε μία τιμή από τις **ετικέτες case**. Οι ετικέτες **case** μπορούν να περιλαμβάνουν είτε μία τιμή είτε ένα κελί με ένα σύνολο πιθανών τιμών. Αν η τιμή της μεταβλητής ελέγχου ταυτίζεται με την τιμή1 ή κάποια από τις τιμές που υπάρχουν στο κελί (τιμή11, τιμή12, ...) της πρώτης ετικέτας **case**, τότε εκτελείται το σύνολο εντολών1, αλλιώς αν ταυτίζεται με την τιμή2 ή κάποια από τις τιμές που υπάρχουν στο κελί (τιμή21, τιμή22, ...) της δεύτερης ετικέτας **case**, τότε εκτελείται το σύνολο εντολών2 κ.ο.κ. Στην περίπτωση όπου δεν ταυτίζεται με καμία από τις ετικέτες **case** τότε εκτελείται το σύνολο εντολών που ακολουθεί την περίπτωση **otherwise** (αν υπάρχει), αλλιώς δεν εκτελείται καμία εντολή. Τέλος, οι εντολές που περιέχονται στην περίπτωση **otherwise** συνήθως αποτελούν ένα μήνυμα λάθους προς το χρήστη.

Παράδειγμα 1

Για την καλύτερη κατανόηση της δομής της switch παρουσιάζεται το παρακάτω παράδειγμα, το οποίο διαβάζει έναν ακέραιο αριθμό από το 1 μέχρι και το 12 και στη συνέχεια εμφανίζει στο χρήστη τον ανάλογο μήνα του έτους, δηλαδή Ιανουάριος για τον αριθμό 1, Φεβρουάριος για τον αριθμό 2, ..., Δεκέμβριος για τον αριθμό 12).

```
minas=input('Παρακαλώ δώστε έναν ακέραιο αριθμό από το 1 έως το 12 για το μήνα που επιθυμείτε: ');
```

```
switch minas
case 1
    disp('Ο μήνας είναι ο : Ιανουάριος');
case 2
    disp('Ο μήνας είναι ο : Φεβρουάριος');
case 3
    disp('Ο μήνας είναι ο : Μάρτιος');
case 4
    disp('Ο μήνας είναι ο : Απρίλιος');
case 5
    disp('Ο μήνας είναι ο : Μάιος');
case 6
    disp('Ο μήνας είναι ο : Ιούνιος');
case 7
    disp('Ο μήνας είναι ο : Ιούλιος');
case 8
    disp('Ο μήνας είναι ο : Αύγουστος');
case 9
    disp('Ο μήνας είναι ο : Σεπτέμβριος');
case 10
    disp('Ο μήνας είναι ο : Οκτώβριος');
case 11
    disp('Ο μήνας είναι ο : Νοέμβριος');
case 12
    disp('Ο μήνας είναι ο : Δεκέμβριος');
otherwise
    disp('Λάθος μήνας! Ο αριθμός πρέπει να είναι ακέραιος από το 1 έως το 12! ');
end
```

Παρακάτω βλέπουμε τα αποτελέσματα που μας επιστρέφει το MATLAB όταν ο χρήστης εισάγει τον αριθμό 12 και 33 για μήνα του έτους:

*Παρακαλώ δώστε έναν ακέραιο αριθμό από το 1 έως το 12 για το μήνα που επιθυμείτε: 12
Ο μήνας είναι ο : Δεκέμβριος*

*Παρακαλώ δώστε έναν ακέραιο αριθμό από το 1 έως το 12 για το μήνα που επιθυμείτε: 33
Λάθος μήνας! Ο αριθμός πρέπει να είναι ακέραιος από το 1 έως το 12.*

Παράδειγμα 2

Το παρακάτω παράδειγμα διαβάζει τα αποτελέσματα των εξετάσεων αγγλικών ενός μαθητή. Αν ο μαθητής πήρε Α εμφανίζει το μήνυμα ΑΡΙΣΤΑ, αν πήρε Β το μήνυμα ΚΑΛΑ, αν πήρε C το μήνυμα ΜΕΤΡΙΑ και αν πήρε F το μήνυμα ΑΠΕΤΥΧΕ:

```
apotelesmata=input('Παρακαλώ δώστε τα αποτελέσματα των αγγλικών του μαθητή: ', 's');
```

```
switch apotelesmata
case 'A'
```

```

disp('Τα αποτελέσματα είναι : ΑΡΙΣΤΑ ');
case 'B'
disp('Τα αποτελέσματα είναι :ΚΑΛΑ ');
case 'C'
disp('Τα αποτελέσματα είναι : ΜΕΤΡΙΑ');
case 'F'
disp('Τα αποτελέσματα είναι : ΑΠΕΤΥΧΕ');
otherwise
disp('Λάθος αποτέλεσμα εξέτασης! ');
end

```

Παρακάτω βλέπουμε τα αποτελέσματα που μας επιστρέφει το MATLAB όταν ο χρήστης εισάγει τον αριθμό ως αποτέλεσμα τα A, C, 4 και K αντίστοιχα:

*>>Παρακαλώ δώστε τα αποτελέσματα των αγγλικών του μαθητή: A
Τα αποτελέσματα είναι : ΑΡΙΣΤΑ*

*>>Παρακαλώ δώστε τα αποτελέσματα των αγγλικών του μαθητή: C
Τα αποτελέσματα είναι : ΜΕΤΡΙΑ*

*>> Παρακαλώ δώστε τα αποτελέσματα των αγγλικών του μαθητή: 4
Λάθος αποτέλεσμα εξέτασης!*

*>>Παρακαλώ δώστε τα αποτελέσματα των αγγλικών του μαθητή: K
Λάθος αποτέλεσμα εξέτασης!*

Παράδειγμα 3

Το παρακάτω παράδειγμα διαβάζει δύο ακέραιους αριθμούς a, b και έναν από τους τρεις χαρακτήρες πρόσθεσης, αφαίρεσης και πολλαπλασιασμού (+, -, *). Ανάλογα με το χαρακτήρα που διαβάζει εκτελεί την αντίστοιχη πράξη (πρόσθεση, αφαίρεση ή πολλαπλασιασμό) και εμφανίζει τα κατάλληλα μηνύματα.

```

a=input('Παρακαλώ εισάγετε τον πρώτο ακέραιο αριθμό : ');
b=input('Παρακαλώ εισάγετε το δεύτερο ακέραιο αριθμό : ');
praksi=input('Παρακαλώ εισάγετε το σύμβολο της πράξης που επιθυμείτε (+, -, * ) : ','s');

switch praksi
case '+'
apotelesma=a+b;
apotelesma=num2str(apotelesma);
disp(['Το αποτέλεσμα της πρόσθεσης είναι : ',apotelesma] );
case '-'
apotelesma=a-b;
apotelesma=num2str(apotelesma);
disp(['Το αποτέλεσμα της αφαίρεσης είναι : ',apotelesma] );
case '*'
apotelesma=a*b;
apotelesma=num2str(apotelesma);
disp(['Το αποτέλεσμα του πολ/σμού είναι : ',apotelesma] );
otherwise
disp('Μη έγκυρη πράξη! ');
end

```

Παρακάτω βλέπουμε τα αποτελέσματα που μας επιστρέφει το MATLAB όταν ο χρήστης εισάγει τους αριθμούς 4,2 με πρόσθεση, 5,2 με αφαίρεση και 5,2 με τον αριθμό 9 στη θέση της πράξης αντίστοιχα:

Παρακαλώ εισάγετε τον πρώτο ακέραιο αριθμό : 4
 Παρακαλώ εισάγετε το δεύτερο ακέραιο αριθμό : 2
 Παρακαλώ εισάγετε το σύμβολο της πράξης που επιθυμείτε (+, -, *) : +
 Το αποτέλεσμα της πρόσθεσης είναι : 6

Παρακαλώ εισάγετε τον πρώτο ακέραιο αριθμό : 5
 Παρακαλώ εισάγετε το δεύτερο ακέραιο αριθμό : 2
 Παρακαλώ εισάγετε το σύμβολο της πράξης που επιθυμείτε (+, -, *) : -
 Το αποτέλεσμα της αφαίρεσης είναι : 3

Παρακαλώ εισάγετε τον πρώτο ακέραιο αριθμό : 5
 Παρακαλώ εισάγετε το δεύτερο ακέραιο αριθμό : 2
 Παρακαλώ εισάγετε το σύμβολο της πράξης που επιθυμείτε (+, -, *) : 9
 Μη έγκυρη πράξη!

ΚΕΦΑΛΑΙΟ 12: ΓΡΑΦΙΚΑ ΣΤΟ MATLAB

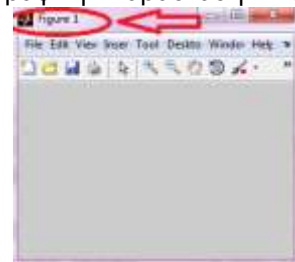
Προκειμένου να βγάλουμε εύκολα συμπεράσματα σχετικά με τα δεδομένα τα οποία εξετάζουμε είναι αναγκαία η χρήση γραφικών παραστάσεων. Μέσω των γραφικών παραστάσεων μας δίνεται η δυνατότητα να κατανοήσουμε τα δεδομένα μας, αφού μπορούμε να κάνουμε εύκολα συγκρίσεις και να δούμε γρήγορα και καθαρά πόσο καλά οι μετρήσεις μας συμφωνούν με κάποια πρόβλεψη που πιθανόν έχουμε κάνει. Το MATLAB μας παρέχει πληθώρα δυνατοτήτων σχετικά με τα γραφικά που μπορούμε να δημιουργήσουμε. Σκοπός αυτού του κεφαλαίου είναι η εξοικείωση με τις βασικές συναρτήσεις για τη διδιάστατη γραφική απεικόνιση των δεδομένων στα ειδικά παράθυρα γραφικών που ονομάζονται *figures*.

12.1. Παράθυρο γραφικών παραστάσεων – Figure

Με την εντολή **figure** δημιουργείται ένα νέο παράθυρο γραφικών το οποίο αναμένει την εκτέλεση κάποιας εντολής προκειμένου να απεικονιστεί μία γραφική παράσταση. Κάθε παράθυρο που δημιουργείται χαρακτηρίζεται από έναν αύξοντα αριθμό, ο οποίος εμφανίζεται πάνω αριστερά ως τίτλος του παραθύρου, όπως φαίνεται στη δεξιά εικόνα.

Η εντολή αυτή μπορεί να παραλειφθεί με την χρήση κάποιας από τις συναρτήσεις γραφικών παραστάσεων, οι οποίες ανοίγουν αυτόματα ένα νέο παράθυρο γραφικών, όταν δεν υπάρχει και αναλύονται παρακάτω (συνάρτηση `plot`, `stem`, `bar` κ.α.). Ωστόσο, στην περίπτωση όπου υπάρχει ήδη τουλάχιστον ένα ανοιχτό παράθυρο γραφικών, η απεικόνιση της νέας γραφικής παράστασης πραγματοποιείται στο τελευταίο χρησιμοποιημένο παράθυρο. Αυτό έχει ως αποτέλεσμα την απώλεια πληροφοριών του διαγράμματος το οποίο καλύφθηκε από τη δημιουργία της τελευταίας γραφικής παράστασης. Για την αποφυγή αυτού του πιθανού λάθους και την απεικόνιση του νέου διαγράμματος, σε ένα ξεχωριστό παράθυρο χωρίς την απώλεια πληροφοριών από τα ήδη υπάρχοντα διαγράμματα, είναι απαραίτητη η επαναπληκτρολόγηση της εντολής `figure` πριν τη χρήση κάποιας συνάρτησης γραφικών του MATLAB. Με αυτό τον τρόπο δημιουργείται ένα νέο παράθυρο γραφικών, το οποίο λαμβάνει τον αμέσως επόμενο αύξοντα αριθμό. Έτσι, στην περίπτωση όπου έχουμε ανοίξει ήδη τέσσερα παράθυρα γραφικών, το επόμενο θα λάβει το όνομα "Figure 5".

Για το κλείσιμο του τρέχοντος `figure` χρησιμοποιείται η εντολή **close** και για το κλείσιμο όλων των παραθύρων χρησιμοποιείται η εντολή **close all**, ενώ στην περίπτωση όπου επιθυμούμε να κλείσουμε ένα συγκεκριμένο παράθυρο γραφικών χρησιμοποιείται η εντολή **close(n)**, όπου `n` είναι ο αύξοντας αριθμός που χαρακτηρίζει το συγκεκριμένο παράθυρο. Σβήσιμο του γραφήματος από το τρέχον παράθυρο γραφικών (`figure`), χωρίς όμως το κλείσιμο του παραθύρου πραγματοποιείται με την εντολή **clf**.



12.2. Συνάρτηση plot

Η βασικότερη συνάρτηση γραφικών είναι η **συνάρτηση plot**, η οποία δημιουργεί γραφικές παραστάσεις καμπυλών, τα σημεία των οποίων είναι αποθηκευμένα σε ισομήκη διανύσματα. Η σύνταξή της είναι η εξής: **plot(x,y,'ιδιότητες')**. Τα δύο πρώτα ορίσματα είναι τα διανύσματα τα οποία αναπαριστούν τις συντεταγμένες (x,y) των σημείων που θα σχεδιαστούν. Το τρίτο όρισμα είναι μία προαιρετική ακολουθία χαρακτήρων που αποτελείται από το πολύ τρεις χαρακτήρες (έναν από κάθε κατηγορία) και σχετίζεται με το χρώμα, τη μορφή και το σημάδι με το οποίο θα αποδοθεί η γραμμή. Δεν είναι υποχρεωτικός ο ορισμός και των τριών κατηγοριών και η σειρά εμφάνισης της κάθε κατηγορίας δεν είναι σημαντική. Στην περίπτωση όπου δεν έχουν οριστεί καθόλου ιδιότητες της γραφικής παράστασης, εμφανίζεται η προκαθορισμένη μορφή μπλε σημείων ενωμένων από μία μπλε εννιαία γραμμή και στην περίπτωση όπου δεν έχουν οριστεί τα σημεία ενός εκ τους δύο αξόνων, το MATLAB ορίζει αυτόματα ως σημεία του δεύτερου άξονα τη δεικτοδότηση των στοιχείων του υπάρχοντος διανύσματος ξεκινώντας όμως από τον αριθμό ένα (σε αντίθεση με άλλες γλώσσες προγραμματισμού). Στον παρακάτω πίνακα εμφανίζονται οι επιτρεπτοί χαρακτήρες για τη δήλωση των ιδιοτήτων της γραφικής παράστασης:

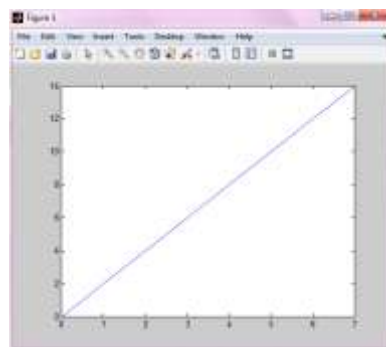
ΧΡΩΜΑ	ΜΟΡΦΗ ΓΡΑΜΜΗΣ	ΣΗΜΕΙΟ ΓΡΑΜΜΗΣ
r κόκκινο	- συνεχής	. τελεία
g πράσινο	-- διακεκομμένη	o κύκλος
b μπλε	: Γραμμή με στίγματα	x σημείο x
w άσπρο	-. Διακεκομμένη γραμμή με στίγματα	+ σημείο +
k μαύρο		* αστερίσκος
y κίτρινο		s τετράγωνο
m ιώδες		d ρόμβος
c γαλάζι		p πεντάγωνο
		h εξάγωνο
		^ βέλος προς τα πάνω
		v βέλος προς τα κάτω
		> βέλος δεξιά
		< βέλος αριστερά

Παράδειγμα 1

Στο παρακάτω παράδειγμα κατασκευάζεται το διάγραμμα των διανυσμάτων x, y, όπου x=[0 1 2 3 4 5 6 7] και y=[0 2 4 6 8 10 12 14] και άρα της πρωτοβάθμιας εξίσωσης $y=a*x$ ($y=2*x$). Επειδή δεν έχει οριστεί κάποια ιδιότητα της γραφικής παράστασης εμφανίζεται με την προκαθορισμένη μορφή εμφάνισης.

```
x=[0 1 2 3 4 5 6 7];
y=[0 2 4 6 8 10 12 14];
```

```
plot(x,y)
```



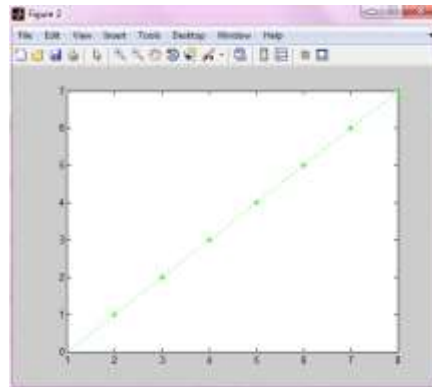
Παράδειγμα 2

Στο παρακάτω παράδειγμα κατασκευάζεται το διάγραμμα του διανύσματος x όπου $x=[0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$ χρησιμοποιώντας πράσινη γραμμή από τελείες με σημεία αστερίσκου. Το διάνυσμα y κατασκευάζεται από μόνο του παίρνοντας κάθε φορά την αρίθμηση του σημείου του διανύσματος x . Άρα εφόσον το διάνυσμα x αποτελείται από οκτώ στοιχεία έτσι και το διάνυσμα y θα περιέχει τους αριθμούς από το 1 μέχρι και το 7, αφού η αρίθμηση ξεκινάει από τον αριθμό ένα ($y=[1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$).

```
x=[0 1 2 3 4 5 6 7];
```



```
plot(x,'g*');
```



Παράδειγμα 3

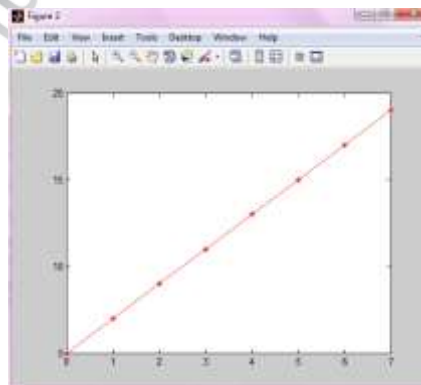
Στο παρακάτω παράδειγμα κατασκευάζεται το διάγραμμα των διανυσμάτων x , y , όπου $x=[0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$ και $y=[5 \ 7 \ 9 \ 11 \ 13 \ 15 \ 17 \ 19]$ και άρα της πρωτοβάθμιας εξίσωσης $y=a*x+b$ ($y=2*x+5$) χρησιμοποιώντας κόκκινη συνεχής γραμμή με σημεία αστερίσκου.

```
x=[0 1 2 3 4 5 6 7];
```

```
y=[5 7 9 11 13 15 17 19];
```



```
plot(x,y)
```

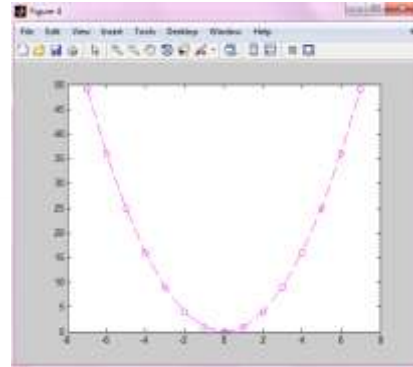


Παράδειγμα 4

Στο παρακάτω παράδειγμα κατασκευάζεται η γραφική παράσταση των διανυσμάτων x , y , όπου $x=[-7 \ -6 \ -5 \ -4 \ -3 \ -2 \ -1 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$ και $y=[49 \ 36 \ 25 \ 16 \ 9 \ 4 \ 1 \ 0 \ 1 \ 4 \ 9 \ 16 \ 25 \ 36 \ 49]$ και άρα της δευτεροβάθμιας εξίσωσης $y=a*x^2$ ($y=1*x^2$) χρησιμοποιώντας ιώδες διακεκομμένη γραμμή με κυκλικά σημεία.

```
x=[-7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7];
y=[49 36 25 16 9 4 1 0 1 4 9 16 25 36
49];
```

```
plot(x,y,'mo--');
```



Σημειώνεται πως ο επακριβής ορισμός των συντεταγμένων του διανύσματος y θα μπορούσε να είχε αποφευχθεί, αν ορίζαμε το διάνυσμα y συναρτήση του x υψώνοντας κάθε στοιχείο του διανύσματος x στο τετράγωνο ως εξής: $y=x.^2$ (Κεφάλαιο 8.4).

! Προσοχή! Τα διανύσματα που περιέχουν τις συνταγμένες των αξόνων x και y πρέπει να είναι ισομήκη, δηλαδή να περιέχουν τον ίδιο αριθμό στοιχείων, αλλιώς το MATLAB δεν μπορεί να δημιουργήσει το επιθυμητό γράφημα.

12.3. Συνάρτηση stem

Μία ακόμα συνάρτηση γραφικών παραστάσεων είναι η **συνάρτηση stem**, η οποία δημιουργεί γραφικές παραστάσεις δύο διανυσμάτων ίδιων διαστάσεων. Για κάθε σημείο που απεικονίζεται πάνω στο διάγραμμα δημιουργεί και το αντίστοιχο ευθύγραμμο τμήμα που αντιστοιχεί στην τιμή της τεταγμένης του σημείου αυτού. Η γενική σύνταξη της εντολής stem είναι η εξής: **stem(x,y,'ιδιότητες')**.

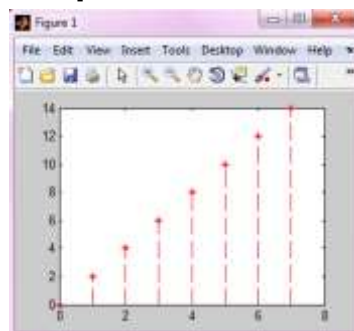
Όπως και στην εντολή plot, έτσι και εδώ δεν είναι υποχρεωτικός ο ορισμός και των δύο διανυσμάτων ή των ιδιοτήτων της γραφικής παράστασης.

Παράδειγμα 1

Στο παρακάτω παράδειγμα χρησιμοποιείται η συνάρτηση stem για την απεικόνιση της γραφικής παράστασης $y=2*x$ όπου $x=[0 1 2 3 4 5 6 7]$.

```
x=[0 1 2 4 5 6 3 7];
y=2.*x
```

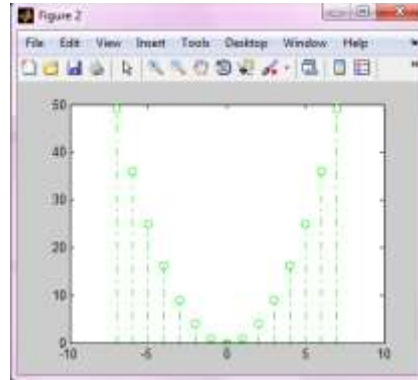
```
stem(x,y,'r--*')
```



Παράδειγμα 2

Στο παρακάτω παράδειγμα δημιουργείται η γραφική παράσταση των διανυσμάτων x , y , όπου $x=[-7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7]$ και $y=a*x^2$ ($y=1*x^2$).

```
x=[-7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7];
y=x.^2
stem(x,y,'g-');
```



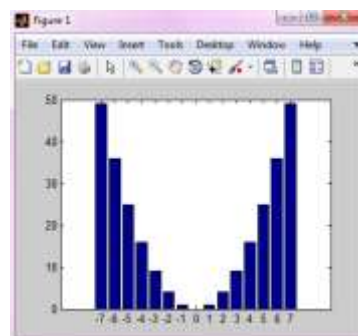
12.4. ΣΥΝΑΡΤΗΣΗ bar και barh

Η **συνάρτηση bar** χρησιμοποιείται για τη δημιουργία κάθετων δισδιάστατων ραβδογραμμάτων, ενώ η συνάρτηση **barh** για τη δημιουργία οριζόντιων δισδιάστατων ραβδογραμμάτων. Η γενική σύνταξη της είναι **bar(x,y)** και **barh(x,y)**. Στην πρώτη περίπτωση, στον οριζόντιο άξονα απεικονίζονται οι τιμές του διανύσματος x και στον κατακόρυφο άξονα οι τιμές του διανύσματος y , ενώ στην περίπτωση της εντολής **barh(x,y)**, οι τιμές του διανύσματος x απεικονίζονται στον κατακόρυφο άξονα. Η προκαθορισμένη μορφή εμφάνισης των ράβδων είναι μπλε χρώματος. Δεν ισχύουν οι ίδιες ιδιότητες διαμόρφωσης γραφικών παραστάσεων που ισχύουν στη συνάρτηση **plot** και **stem** και γι' αυτό το λόγο στην περίπτωση όπου επιθυμούμε την αλλαγή της προκαθορισμένης μορφής του γραφήματος **bar** αρκεί να συμβουλευτούμε το MATLAB κάνοντας χρήση της εντολής **help bar**.

Παράδειγμα 1

Στο παρακάτω παράδειγμα χρησιμοποιείται η συνάρτηση **bar** για την απεικόνιση της γραφικής παράστασης $y=x^2$, όπου $x=[-7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7]$.

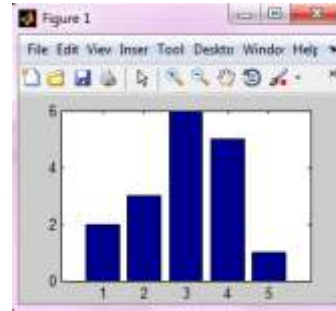
```
x=[-7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7];
y=x.^2;
bar(x,y);
```



Παράδειγμα 2

Στο παρακάτω παράδειγμα χρησιμοποιείται η συνάρτηση **bar** για την απεικόνιση του διανύσματος $y=[2 3 6 5 1]$.

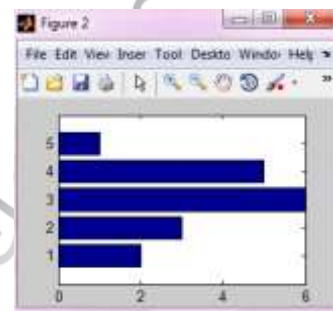

```
y=[2 3 6 5 1];
bar(y)
```



Παράδειγμα 3

Τροποποιώντας το παραπάνω παράδειγμα με τη χρήση της συνάρτησης `barh` για την απεικόνιση του διανύσματος $y=[2\ 3\ 6\ 5\ 1]$ λαμβάνουμε τα εξής δεδομένα:

```
x=[1 2 3 4 5];
y=[2 3 6 5 1];
barh(x,y)
```



12.5. Μορφοποίηση γραφικών παραστάσεων

12.5.1. Εντολές `xlabel` και `ylabel`

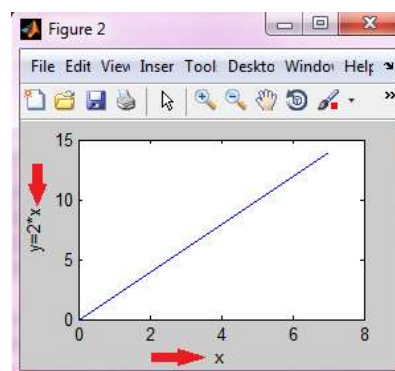
Οι εντολές `xlabel` και `ylabel` χρησιμοποιούνται για την εισαγωγή κειμένου, ως ετικέτας, στον άξονα x και y αντίστοιχα. Η γενική σύνταξή τους είναι: `xlabel('μήνυμα κειμένου')` και `ylabel('μήνυμα κειμένου')`.

Παράδειγμα

Εισάγοντας ετικέτα στους άξονες του πρώτου παραδείγματος, το γράφημα παίρνει την εξής μορφή:

```
x=[0 1 2 3 4 5 6 7];
y=[0 2 4 6 8 10 12 14];
```

```
plot(x,y)
xlabel('x');
ylabel('y=2*x');
```



12.5.2. Εντολή title

Με την **εντολή title** εισάγεται ένα κείμενο στο πάνω μέρος του γραφήματος ως τίτλος της γραφικής παράστασης. Η γενική σύνταξη της εντολής είναι: **title('ονομασία γραφήματος')**.

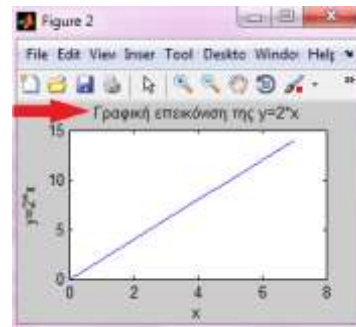
Παράδειγμα

Συνεχίζοντας το παραπάνω παράδειγμα και εισάγωντας τίτλο, το γράφημα μετατρέπεται:

```
x=[0 1 2 3 4 5 6 7];
y=[0 2 4 6 8 10 12 14];
```

```
plot(x,y)
xlabel('x');
ylabel('y=2*x');
```

```
title('Γραφική επεικόνιση της y=2*x');
legend('y=2*x');
```



12.5.3. Εντολή legend

Με την **εντολή legend** εισάγεται ένα πλαίσιο κειμένου το οποίο περιέχει ένα μήνυμα κειμένου, ως υπόμνημα, στο άνω μέρος του γραφήματος για την επεξήγηση της γραφικής παράστασης. Η γενική σύνταξη της εντολής είναι: **legend('μήνυμα κειμένου')**. Παρόλο που η προκαθορισμένη θέση του πλαισίου είναι στο άνω μέρος του γραφήματος, η εντολή **legend** δέχεται και ένα τρίτο όρισμα, το οποίο ορίζει τη θέση του στο διάγραμμα. Πρόκειται για ένα ακέραιο αριθμό από το μείον ένα έως το τέσσερα όπως φαίνεται στον παρακάτω πίνακα:

Αριθμός	Θέσης πλαισίου
-1	δεξιά- έξω από το γράφημα
0	επίλογή εκείνης της θέσης, η οποία αποκρύβει όσο το δυνατόν λιγότερη πληροφορία
1	πάνω δεξιά
2	πάνω αριστερά
3	κάτω αριστερά
4	κάτω δεξιά

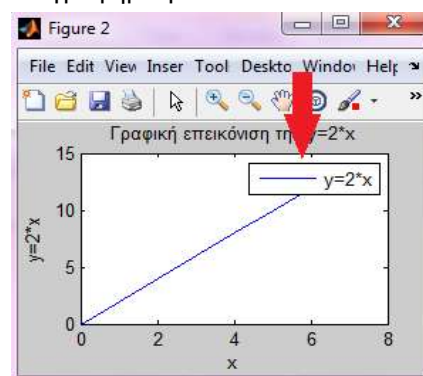
Παράδειγμα

Εισάγωντας ως υπόμνημα το κείμενο $y=2*x$ το γράφημα γίνεται:

```
x=[0 1 2 3 4 5 6 7];
y=[0 2 4 6 8 10 12 14];
```

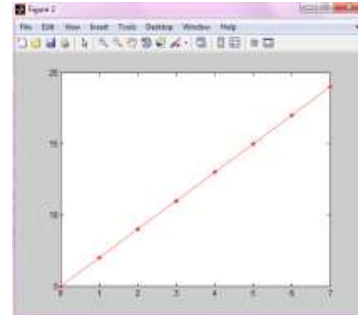
```
plot(x,y)
xlabel('x');
ylabel('y=2*x');
title('Γραφική
επεικόνιση της y=2*x');
```

```
legend('y=2*x',1);
```



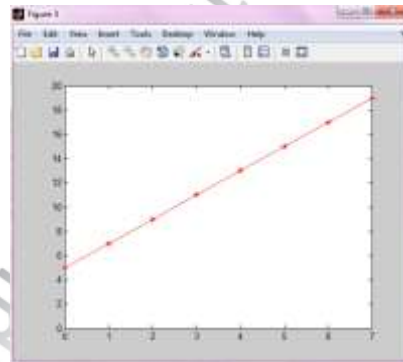
12.5.4. ΕΝΤΟΛΗ *axis*

Στο τρίτο παράδειγμα της συνάρτησης `plot` (Κεφάλαιο 13.2) παρατηρούμε ότι η απεικόνιση της γραφικής παράστασης $y=2*x+5$ ξεκινάει από το σημείο (0,5), αλλά οι άξονες δεν έχουν κοινή αρχή το σημείο (0,0). Για το λόγο αυτό θα προσθέσουμε την εντολή `axis([XMIN XMAX YMIN YMAX])` στο τέλος του κώδικα, η οποία αλλάζει τα όρια των αξόνων ορίζοντας μία αρχική και τελική τιμή για τον άξονα xx' και yy' . Πιο συγκεκριμένα θα ορίσουμε την εντολή `axis([0 7 0 20])` ορίζοντας έτσι ως αρχική και τελική τιμή του άξονα xx' το 0 και το 7 αντίστοιχα και ως αρχική και τελική τιμή του άξονα yy' το 0 και το 20 αντίστοιχα. Έτσι, το διάγραμμα παίρνει την συνηθισμένη μορφή που χρησιμοποιούμε και στα μαθηματικά, όπως φαίνεται παρακάτω:



```
x=[0 1 2 3 4 5 6 7];
y=[5 7 9 11 13 15 17 19];

plot(x,y,'r*-');
axis([0 7 0 20]);
```



Η εμφάνιση των αξόνων πραγματοποιείται με την εντολή `axis on` και είναι η προκαθορισμένη κατάσταση εμφάνισης των αξόνων. Για την απόκρυψη των αξόνων και του άσπρου συνήθως υποβάθρου χρησιμοποιείται η εντολή `axis off`.

12.5.5. Εντολή *grid* ή *grid on*

Με την εντολή `grid` ή `grid on` δημιουργείται ένα πλέγμα στο γράφημα για την ευκολότερη και ταχύτερη ανάγνωσή του. Αντίστοιχα για την απενεργοποίηση του πλέγματος εισάγουμε πάλι την εντολή `grid` ή την εντολή `grid off`.

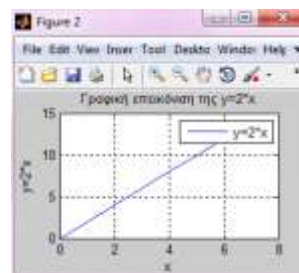
Παράδειγμα

Εισάγωντας πλέγμα, το γράφημα παίρνει την τελική του μορφή:

```
x=[0 1 2 3 4 5 6 7];
y=[0 2 4 6 8 10 12 14];

plot(x,y)
xlabel('x');
ylabel('y=2*x');
title('Γραφική απεικόνιση της y=2*x');

legend('y=2*x');
```



12.5.6. Εντολή hold ή hold on

Η εντολή **hold** ή **hold on** είναι ίσως από τις χρησιμότερες εντολές γραφικών, αφού μας επιτρέπει την ταυτόχρονη απεικόνιση διαφορετικών γραφικών παραστάσεων στο ίδιο παράθυρο γραφικών (figure). Όπως προαναφέρθηκε και στο κεφάλαιο 13.1, στην περίπτωση όπου ήδη υπάρχει ένα ανοιχτό παράθυρο γραφικών, η απεικόνιση της νέας γραφικής παράστασης πραγματοποιείται στο τελευταία χρησιμοποιημένο παράθυρο ή σε καινούριο παράθυρο εφόσον προηγηθεί η εντολή figure. Με τη χρήση της εντολής **hold** ή **hold on** το τρέχον παράθυρο γραφικών «παγώνει» και η καινούρια γραφική παράσταση σχεδιάζεται στο ίδιο γράφημα, μαζί με τις ήδη υπάρχουσες πληροφορίες.

Για την απελευθέρωση του τρέχοντος παραθύρου χρησιμοποιείται η εντολή **hold off**, με αποτέλεσμα τα νέα γραφήματα να σχεδιάζονται σε καινούρια παράθυρα γραφικών.

Παράδειγμα

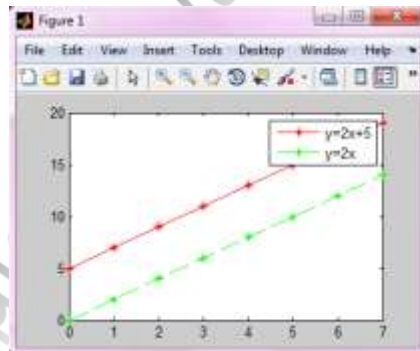
Στο παρακάτω παράδειγμα θα δημιουργήσουμε τη γραφική παράσταση των συναρτήσεων $y=2x+5$ και $y=2x$ με κόκκινη συνεχής γραμμή με αστερίσκους και πράσινη διακεκομμένη γραμμή με αστερίσκους αντίστοιχα. Θα εισάγουμε επίσης και υπόμνημα για την καλύτερη κατανόηση των διαγραμμάτων.

```
x=[0 1 2 3 4 5 6 7];
y=[5 7 9 11 13 15 17 19];
y2=[0 2 4 6 8 10 12 14];
```

→

```
plot(x,y,'r*-');
hold on
plot(x,y2,'g*--');
```

```
axis([0 7 0 20]);
legend('y=2x+5','y=2x')
```

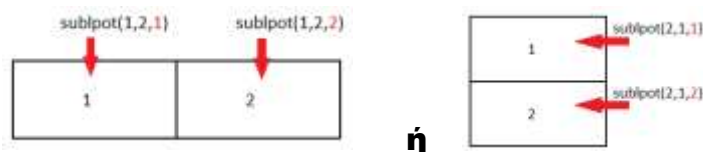


12.6. Πολλαπλά γραφήματα σε ένα παράθυρο γραφικών

Η συνάρτηση subplot μας δίνει τη δυνατότητα να απεικονίσουμε δύο ή περισσότερα γραφήματα στο ίδιο παράθυρο γραφικών. Η σύνταξη της εντολής είναι:

subplot(πλήθος γραμμών, πλήθος στηλών, αύξων αριθμός θέσης γραφήματος)

Με την εντολή αυτή το ενεργό παράθυρο γραφικών χωρίζεται σε τόσα μέρη όσα ορίζουν οι γραμμές και οι στήλες. Ο αύξων αριθμός θέσης ορίζει τη θέση στην οποία θα τοποθετηθεί το κάθε γράφημα. Πιο συγκεκριμένα, στην περίπτωση όπου επιθυμούμε την ταυτόχρονη απεικόνιση δύο διαγραμμάτων σε ένα παράθυρο γραφικών είναι αναγκαίος ο ορισμός μίας γραμμής και δύο στηλών ή δύο γραμμών και μίας στήλης. Εφόσον, το πρώτο γράφημα θα τοποθετηθεί στην πρώτη θέση ο αύξων αριθμός του θα είναι ο ακέραιος αριθμός ένα και αντίστοιχα ο αύξων αριθμός του δεύτερου θα είναι ο ακέραιος αριθμός δύο.



Απαραίτητη προϋπόθεση είναι η εντολή subplot να προηγείται των εντολών δημιουργίας γραφημάτων (plot, stem, bar κ.α)

Παράδειγμα 1

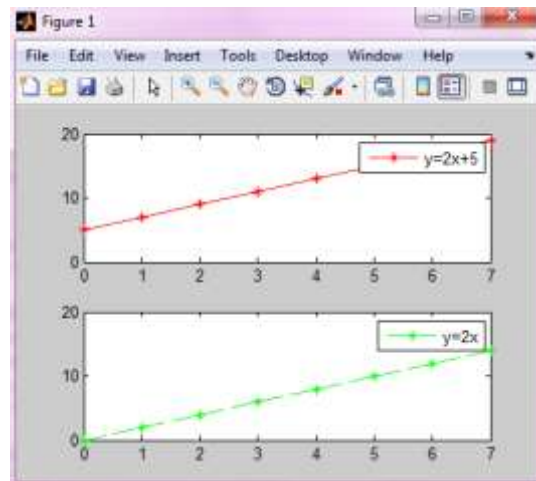
Στο παρακάτω παράδειγμα θα δημιουργήσουμε τη γραφική παράσταση των συναρτήσεων $y=2x+5$ και $y=2x$ με κόκκινη συνεχής γραμμή με αστερίσκους και πράσινη διακεκομμένη γραμμή με αστερίσκους αντίστοιχα. Θα εισάγουμε επίσης και υπόμνημα για την καλύτερη κατανόηση των διαγραμμάτων.

```
x=[0 1 2 3 4 5 6 7];
y=[5 7 9 11 13 15 17 19];
y2=[0 2 4 6 8 10 12 14];
```

```
subplot(2,1,1);
plot(x,y,'r*-');
legend('y=2x+5');
axis([0 7 0 20]);
```

```
hold on
subplot(2,1,2);
plot(x,y2,'g*-');
axis([0 7 0 20]);
legend('y=2x');
```

```
axis([0 7 0 20]);
legend('y=2x');
```



Παράδειγμα 2

Παρακάτω παρουσιάζεται η χρήση της εντολής subplot για την απεικόνιση τεσσάρων γραφημάτων σε ένα παράθυρο γραφικών:

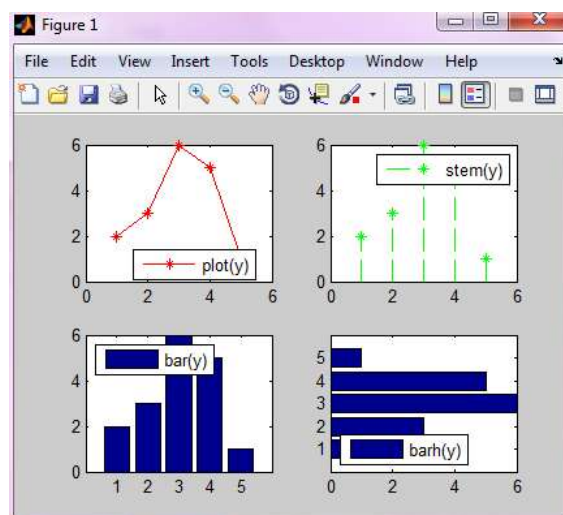
```
y=[2 3 6 5 1];
```

```
subplot(2,2,1);
plot(y,'r*-');
legend('plot(y)',0);
```

```
subplot(2,2,2);
stem(y,'g*-');
legend('stem(y)',1);
```

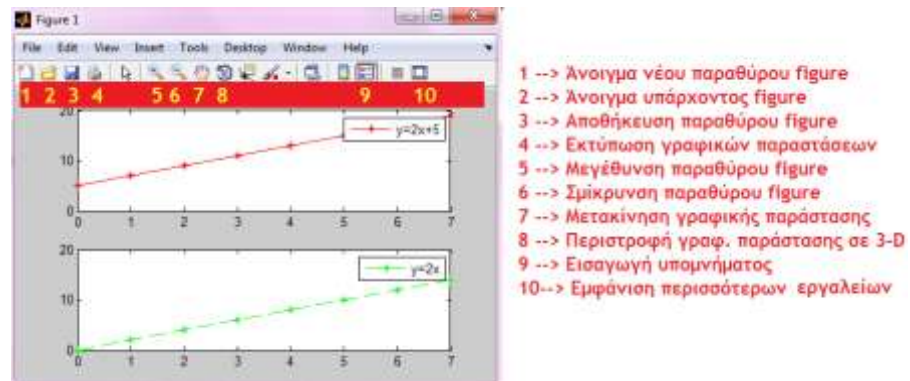
```
subplot(2,2,3);
bar(y,'grouped');
legend('bar(y)',2);
```

```
subplot(2,2,4);
barh(y);
legend('barh(y)',3);
```



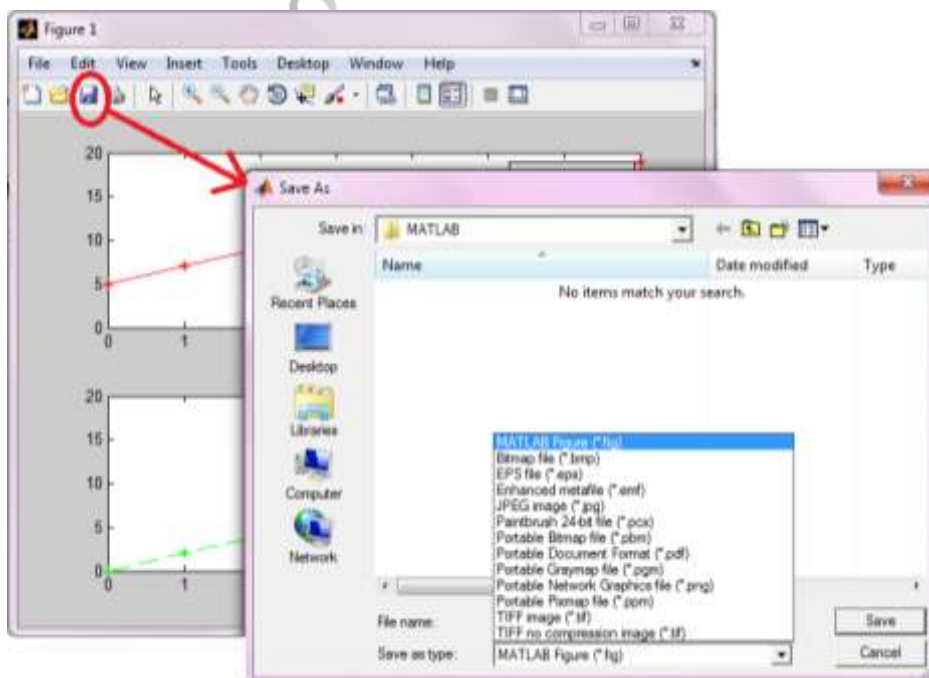
12.7. Αποθήκευση γραφικών παραστάσεων

Το MATLAB μας παρέχει επίσης αρκετά εργαλεία για τη διαχείριση των γραφικών παραστάσεων που έχουμε δημιουργήσει. Μερικές από πιο σημαντικές φαίνονται στην παρακάτω εικόνα. Κάθε αριθμός που εμφανίζεται στο κόκκινο φόντο της εικόνας, με κίτρινα γράμματα, αντιστοιχεί στο εργαλείο ακριβώς από πάνω από αυτόν. Στο αριστερό μέρος της εικόνας βρίσκεται η επεξήγησή του κάθε εργαλείου:



Εικόνα 13 –Εργαλεία γραφικών παραστάσεων.

Όπως παρατηρούμε, μία από τις δυνατότητες που μας δίνει το MATLAB μέσω της εργαλειοθήκης του παραθύρου, όσον αφορά τα γραφικά η αποθήκευσή τους. Στην περίπτωση, λοιπόν, που ο χρήστης επιθυμεί να έχει στη διάθεσή του τα γραφήματα που έχει δημιουργήσει, με σκοπό την επαναχρησιμοποίησή τους, μπορεί να τα αποθηκεύσει ως αρχεία με κατάληξη **.fig**, μία επιλογή την οποία προτείνει το ίδιο το MATLAB. Δυνατή είναι επίσης και η αποθήκευση των γραφημάτων ως αρχεία εικόνων με τις διάφορες γνωστές καταλήξεις αρχείων εικόνων (**.jpeg**, **.png**, **.bmp**, **.gif** κ.α.).



Εικόνα 14 –Παράδειγμα αποθήκευσης γραφικής παράστασης.

Παράρτημα Α: Ασκήσεις βασικών ενοτήτων

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1. Σε ποιο παράθυρο της Επιφάνειας Εργασίας του MATLAB μπορούμε να εκτελέσουμε απ'ευθείας μια εντολή;
 - a) Παράθυρο σύνταξης αρχείων (Editor Window).
 - b) Παράθυρο εντολών (Command Window).
 - c) Παράθυρο τρέχοντος καταλόγου (Current directory).
2. Σε ποιο παράθυρο της Επιφάνειας Εργασίας του MATLAB μπορούν να γραφτούν πολλές εντολές, οι οποίες θα εκτελεστούν αργότερα;
 - a) Παράθυρο σύνταξης αρχείων (Editor Window).
 - b) Παράθυρο εντολών (Command Window).
 - c) Παράθυρο τρέχοντος καταλόγου (Current directory).
3. Σε ποιο παράθυρο της Επιφάνειας Εργασίας του MATLAB μπορεί να ζητηθεί πλήρη βοήθεια για μια εντολή;
 - a) Παράθυρο τρέχοντος καταλόγου (Current directory).
 - b) Παράθυρο χώρου εργασίας (Workspace).
 - c) Παράθυρο βοήθειας (Help).
4. Σε ποιο παράθυρο της Επιφάνειας Εργασίας του MATLAB απεικονίζονται όλες οι μεταβλητές μαζί με τις σχετικές τους πληροφορίες;
 - a) Παράθυρο χώρου εργασίας (Workspace).
 - b) Παράθυρο ιστορικού εντολών (Command History).
 - c) Παράθυρο βοήθειας (Help).
5. Σε ποιο παράθυρο της Επιφάνειας Εργασίας του MATLAB απεικονίζονται οι γραφικές παραστάσεις;
 - a) Παράθυρο σύνταξης αρχείων (Editor Window).
 - b) Παράθυρο γραφημάτων (Figure Window).
 - c) Παράθυρο βοήθειας (Help).
6. Η MATLAB διακρίνει τους κεφαλαίους από τους πεζούς χαρακτήρες;
 - a) Σωστό.
 - b) Λάθος.
7. Δε μπορούμε να εισάγουμε πολλές εντολές στην ίδια γραμμή στη MATLAB.
 - a) Σωστό.
 - b) Λάθος.
8. Για την απόκρυψη του αποτελέσματος μίας εντολής από την οθόνη του χρήστη χρησιμοποιείται το ελληνικό ερωτηματικό.
 - a) Σωστό.
 - b) Λάθος.

ΚΕΦΑΛΑΙΟ 2: ΒΟΗΘΕΙΑ ΤΟΥ MATLAB – ΠΛΗΡΟΦΟΡΙΕΣ ΕΚΔΟΣΗΣ

1. Ποιά εντολή πρέπει να πληκτρολογήσουμε στο Παράθυρο Εντολών του MATLAB για να λάβουμε πληροφορίες για την εντολή plot;
 - a) help me plot
 - b) help for plot
 - c) help plot
2. Ποιά εντολή μας εμφανίζει την έκδοση του MATLAB που χρησιμοποιούμε (πεζούς λατινικούς χαρακτήρες);
Εντολή: _____
3. Για ποιο λόγο η εντολή **helpmin** είναι λάθος και το MATLAB αδυνατεί να μας παρέχει βοήθεια;
 - a) Δε πρέπει να υπάρχει η εντολή help στην εντολή.
 - b) Δεν υπάρχει κενό μεταξύ της εντολής help και της εντολής min.
 - c) Όλα τα παραπάνω.

ΚΕΦΑΛΑΙΟ 3: ΕΠΙΤΡΕΠΤΑ ΜΕΓΕΘΗ ΣΤΗ MATLAB

1. Ποιός είναι ο σωστός τρόπος δήλωσης του δεκαδικού αριθμού 5,637;
 - a) 5,637
 - b) 5.637
 - c) 5;637
2. Ποιός από τους παρακάτω τρόπους αποτελεί το σωστό τρόπο δήλωσης δεκαδικού αριθμού με τη χρήση των δυνάμεων του δέκα (0.0879);
 - a) 879e-4
 - b) 879.10(-4)
 - c) 879 e-4
3. Η φανταστική μονάδα των μιγαδικών αριθμών στο MATLAB παριστάνεται ως είτε με το χαρακτήρα i, είτε με το χαρακτήρα j.
 - a) Σωστό.
 - b) Λάθος.
4. Στην περίπτωση όπου χρησιμοποιηθεί η σύνταξη **a+b*i** για τη δήλωση ενός μιγαδικού αριθμού πρέπει να προσέχουμε να μην έχει οριστεί προηγουμένως το i ως μια άλλη μεταβλητή.
 - a) Σωστό.
 - b) Λάθος.
5. Ποιά από τις παρακάτω ενσωματωμένες συναρτήσεις του MATLAB χρησιμοποιείται για την εμφάνιση του φανταστικού μέρους ενός μιγαδικού αριθμού;
 - a) im(a+bi)
 - b) imagine(a+bi)

- c) $\text{imag}(a+bi)$
6. Ο μονοδιάστατος πίνακας $[5 \ 45 \ 67 \ 3]$ είναι ένα διάνυσμα τεσσάρων γραμμών και μίας στήλης.
- Σωστό.
 - Λάθος.
7. Τα στοιχεία μίας γραμμής ενός πίνακα χωρίζονται είτε με τη χρήση κενών ανάμεσα των στοιχείων είτε με τη χρήση της υποδιαστολής.
- Σωστό.
 - Λάθος.
8. Το διάνυσμα $[10:5:100]$ έχει ως:
- πρώτο στοιχείο το 10, τελευταίο δυνατό στοιχείο το 5 και βήμα μεταβολής το 100.
 - πρώτο στοιχείο το 10, τελευταίο δυνατό στοιχείο το 100 και βήμα μεταβολής το 5.
 - πρώτο στοιχείο το 5, τελευταίο δυνατό στοιχείο το 100 και βήμα μεταβολής το 10.
9. Το βήμα μεταβολής ενός πίνακα μπορεί να παραλειφθεί και να οριστεί αυτόματα ίσο με τη μονάδα.
- Σωστό.
 - Λάθος.
10. Τί ισχύει για την τελική και αρχική τιμή κατά τη δήλωση ενός πίνακα όταν το βήμα είναι αρνητικό; Συμπληρώστε με το κατάλληλο σύμβολο ανισότητας ($<$ ή $>$).
- τελική τιμή _____ ή ίση από την αρχική τιμή
11. Συμπληρώστε τα κατάλληλα πεδία έτσι ώστε να δημιουργηθεί ένα διάνυσμα, το οποίο θα περιέχει τους ακέραιους αριθμούς από το 1 έως και το 10. ($[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$)
- [___ : ___ : ___]
12. Συμπληρώστε τα κατάλληλα πεδία έτσι ώστε να δημιουργηθεί ένα διάνυσμα, το οποίο θα περιέχει τους πρώτους οκτώ ακέραιους μονούς αριθμούς. ($[1, 3, 5, 7, 9, 11, 13, 15]$)
- [___ : ___ : ___]
13. Συμπληρώστε τα κατάλληλα πεδία έτσι ώστε να δημιουργηθεί ένα διάνυσμα, το οποίο θα περιέχει τους πρώτους οκτώ ακέραιους μονούς και αρνητικούς αριθμούς, ξεκινώντας από τον αριθμό -1. ($[-1, -3, -5, -7, -9, -11, -13, -15]$)
- [___ : ___ : ___]
14. Συμπληρώστε τα κατάλληλα πεδία έτσι ώστε να δημιουργηθεί ένα διάνυσμα, το οποίο θα περιέχει την προπαίδεια του πέντε σε φθίνουσα σειρά. ($[50, 45, 40, 35, 30, 25, 20, 15, 10, 5]$)
- [___ : ___ : ___]
15. Η αλλαγή γραμμής στους διδιάστατους πίνακες γίνεται με τη χρήση της υποδιαστολής.
- Σωστό.
 - Λάθος.

16. Οι συνιστώσες ενός διδιάστατου πίνακα βρίσκονται ανάμεσα σε:
- Κενά.
 - Παρενθέσεις.
 - Αγκύλες.
17. Ποιό είναι το λάθος στη δήλωση του διδιάστατου πίνακα: [2 3 4 ; 5 6]
- Δε διαθέτει τον ίδιο αριθμό στοιχείων σε όλες τις γραμμές.
 - Το ερωτηματικό έπρεπε να ήταν πριν τον αριθμό 4.
 - Δεν έπρεπε να είχε ερωτηματικό.
18. Συμπληρώστε τα κατάλληλα πεδία έτσι ώστε να δημιουργηθεί ένας διδιάστατος πίνακας, ο οποίος θα περιέχει τους αριθμούς 5, 10, 15 στην πρώτη γραμμή και τους αριθμούς 2, 4, 6 στη δεύτερη.
- [__ , __ , __ ; __ , __ , __]
19. Πώς ορίζεται ο κενός πίνακας;
- []
 - ()
 - Τίποτα από τα παραπάνω.
20. Πώς θα αναγνωρίσει το MATLAB την εντολή: `>> 'καλημέρα'`
- ως ένα διάνυσμα.
 - ως ένα πίνακα.
 - ως μία ακολουθία χαρακτήρων.
21. Προκειμένου μία ακολουθία χαρακτήρων να εισαχθεί και να αναγνωριστεί ως τέτοια πρέπει πάντα να περικλείεται από μονά εισαγωγικά.
- Σωστό.
 - Λάθος.

ΚΕΦΑΛΑΙΟ 4: ΒΑΣΙΚΕΣ ΠΡΑΞΕΙΣ

1. Ποιό αποτέλεσμα αναμένεται να μας επιστρέψει η πληκτρολόγηση της εντολής `>> 3*2` στο Παράθυρο εντολών του MATLAB;
 - a) 6
 - b) 32
 - c) 3*2
2. Ποιό αποτέλεσμα αναμένεται να μας επιστρέψει η πληκτρολόγηση της εντολής `>> 3^2` στο Παράθυρο εντολών του MATLAB;
 - a) 6
 - b) 9
 - c) 32
3. Ποιό αποτέλεσμα αναμένεται να μας επιστρέψει η πληκτρολόγηση της εντολής `>> 5*(20-10)+6` στο Παράθυρο εντολών του MATLAB;
 - a) 96
 - b) 56
 - c) 46
4. Ποιό αποτέλεσμα αναμένεται να μας επιστρέψει η πληκτρολόγηση της εντολής `>> (0.5^2+0.25)*5` στο Παράθυρο εντολών του MATLAB;
 - a) 2
 - b) 5
 - c) 1.25
5. Τί αποτέλεσμα θα μας επιστρέψει ο συγκριτικός έλεγχος της εντολής `>> 5>7` ;
 - a) 1
 - b) 0
 - c) Κανένα από τα παραπάνω.
6. Συμπληρώστε τα κατάλληλα πεδία με το αποτέλεσμα που θα προκύψει ύστερα από το συγκριτικό έλεγχο της εντολής `>> 'papi' == 'mami'` .
— — — —
7. Τί αποτέλεσμα θα μας επιστρέψει η λογική πράξη `>> (2<6 & 7>=6)` όταν πληκτρολογηθεί στο Παράθυρο εντολών του MATLAB;
 - a) 1
 - b) 0
 - c) Κανένα από τα παραπάνω.
8. Τί αποτέλεσμα θα μας επιστρέψει η λογική πράξη `>> (6<2 | 6>=7)` όταν πληκτρολογηθεί στο Παράθυρο εντολών του MATLAB;
 - a) 1
 - b) 0
 - c) Κανένα από τα παραπάνω.
9. Συμπληρώστε τα κατάλληλα πεδία με το αποτέλεσμα που θα προκύψει ύστερα από τη λογική πράξη `>> ~[0 1 1 0]`
— — — —

10. Ποιά ενσωματωμένη συνάρτηση του MATLAB συνενώνει δύο ακολουθίες χαρακτήρων και στις επιστρέφει ως μια ενιαία;
- a) findstr()
 - b) double()
 - c) strcat()
11. Ποιά ενσωματωμένη συνάρτηση του MATLAB βρίσκει τις θέσεις της μικρότερης ακολουθίας χαρακτήρων (εκ των δύο) στη μεγαλύτερη και τις επιστρέφει ως ένα διάνυσμα αριθμών;
- a) findstr()
 - b) double()
 - c) char()
12. Ποιά ενσωματωμένη συνάρτηση του MATLAB συνενώνει κάθετα δύο ακολουθίες χαρακτήρων και τις επιστρέφει ως ένα διάνυσμα στήλης;
- a) findstr()
 - b) strvcat()
 - c) strcat()
13. Ποιά ενσωματωμένη συνάρτηση του MATLAB επιστρέφει το χαρακτήρα που αντιστοιχεί στο θετικό ακέραιο αριθμό ASCII n;
- a) findstr()
 - b) double()
 - c) char()
14. Ποιά ενσωματωμένη συνάρτηση του MATLAB αντικαθιστά τη δεύτερη ακολουθία χαρακτήρων με την τρίτη, όπου αυτή εμφανίζεται στην πρώτη ακολουθία χαρακτήρων και επιστρέφει ως μία ακολουθία χαρακτήρων;
- a) int2str()
 - b) strrep()
 - d) double()
15. Ποιά ενσωματωμένη συνάρτηση του MATLAB μετατρέπει τον πραγματικό αριθμό n σε ακολουθία χαρακτήρων;
- a) findstr()
 - b) num2str()
 - c) strcat()
16. Ποιά ενσωματωμένη συνάρτηση του MATLAB μετατρέπει τον ακέραιο αριθμό n σε ακολουθία χαρακτήρων;
- a) double()
 - a) num2str()
 - b) int2str()

17. Ποιά ενσωματωμένη συνάρτηση του MATLAB επιστρέφει ένα διάνυσμα θετικών ακέραιων αριθμών όπου το κάθε στοιχείο του αντιστοιχεί σε κάθε ένα από τους χαρακτήρες μίας ακολουθίας χαρακτήρων;
- a) double()
 - b) num2str()
 - c) strcat()
18. Ποιά ενσωματωμένη συνάρτηση του MATLAB μετατρέπει τον πίνακα πραγματικών αριθμών R σε πίνακα από ακολουθίες χαρακτήρων;
- a) int2str()
 - b) num2str()
 - c) strrep()

ΚΕΦΑΛΑΙΟ 5: ΜΕΤΑΒΛΗΤΕΣ

1. Πού ανατίθενται τα αποτελέσματα των εντολών σε περίπτωση όπου δεν έχει οριστεί κάποια μεταβλητή;
- a) μεταβλητή ans.
 - b) μεταβλητή result.
 - c) κανένα από τα παραπάνω.
2. Γιατί αναθέτουμε το αποτέλεσμα των εντολών σε μεταβλητές;
- a) Για να διατηρήσουμε τα αποτελέσματα των εντολών κατά την εκτέλεση του προγράμματος.
 - b) Διότι τα αποτελέσματα θα χαθούν μόλις εκτελεστεί η επόμενη εντολή, αφού η μεταβλητή ans είναι μια αυτόματη προσωρινή μεταβλητή μνήμης, στην οποία ανατίθεται το πιο πρόσφατο αποτέλεσμα εντολής.
 - c) Όλα τα παραπάνω.
3. Ποιό από τα παρακάτω ονόματα μεταβλητής δεν είναι αποδεκτό;
- a) name1_
 - b) name_1
 - c) 1name_
4. Ποιό από τα παρακάτω ονόματα μεταβλητής δεν είναι αποδεκτό και γιατί;
- a) Το name-1, γιατί περιλαμβάνει το μη έγκυρο χαρακτήρα παύλα.
 - b) Το name_, γιατί περιλαμβάνει την κάτω παύλα στο τέλος.
 - c) Το n_ame77, γιατί περιλαμβάνει αριθμούς.
5. Τα ονόματα μιας μεταβλητής δεν πρέπει να ταυτίζονται με δεσμευμένες λέξεις της MATLAB, αφού απενεργοποιούν τη λειτουργία των δεσμευμένων λέξεων.
- a) Σωστό.
 - b) Λάθος.
6. Ποιές θα είναι η τιμή της μεταβλητής x ύστερα από την εκτέλεση της εντολής `>>x = 10+2*5;`
- a) x=20
 - b) x=60
 - c) x=125

7. Ποιές θα είναι η τιμή της μεταβλητής y ύστερα από την εκτέλεση της εντολής
`>> y = 4^(20-(5*2+8)) ;`
a) $y = 6.8719$
b) $y = 64$
c) $y = 16$
8. Ποιές θα είναι οι τιμές των μεταβλητών ύστερα από την εκτέλεση των παρακάτω σειρών εντολών;
`>> a=100;`
`>> b=a/2;`
`>> c=b+100;`
`>> a=c+b;`
`>> c=0;`
a) $a=100, b=50, c=150$
b) $a=200, b=50, c=150$
c) $a=200, b=50, c=0$
9. Ποιά θα είναι η τιμή της μεταβλητής `result` ύστερα από την εκτέλεση της συγκριτικής πράξης
`>> result = 2<3;`
a) `result = 23`
b) `result = 1`
c) `result = 0`
10. Ποιές θα είναι οι τιμές των μεταβλητών ύστερα από την εκτέλεση των συγκριτικών πράξεων;
`>> Var1 = 2<3;`
`>> Var2 = 2>3;`
`>> Var3 = 2>=3;`
`>> Var4 = 2<=3;`
`>> Var5 = 2==3;`
`>> Var6 = 2~3;`
a) $Var1=0, Var2=0, Var3=0, Var4=1, Var5=0, Var6=1$
b) $Var1=1, Var2=0, Var3=0, Var4=1, Var5=0, Var6=1$
c) $Var1=1, Var2=0, Var3=1, Var4=0, Var5=0, Var6=1$
11. Ποιά θα είναι η τιμή της μεταβλητής `logical_var` ύστερα από την εκτέλεση της λογικής πράξης
`>> logical_var = 1 & 0;`
a) `logical_var = 1`
b) `logical_var = 0`
c) Κανένα από τα παραπάνω.
12. Ποιές θα είναι οι τιμές των μεταβλητών ύστερα από την εκτέλεση των παρακάτω λογικών πράξεων;
`>> x1 = (1 & 0)`
`>> x2 = (~ 1) & 0`
`>> x3 = ~ 1 & 0`
`>> x4 = 1 | 0`

```
>> x5 = (1 | 0) & 0
```

- a) x1=1, x2=1, x3=1, x4=0, x5=1
- b) x1=0, x2=0, x3=0, x4=1, x5=0
- c) x1=0, x2=0, x3=0, x4=0, x5=0

13. Ποιές θα είναι οι τιμές των μεταβλητών ύστερα από την εκτέλεση των παρακάτω συγκριτικών και λογικών πράξεων;

```
>> a = 36>4 & 36<4
>> b = a==4 | a>=36
>> c = b+36
>> d = c<=4 & ~(c~=4)
>> e = ~(4>c)
>> f = 40
>> g = (a>0 & f<3) | (a>4 & f<10)
```

- a) a=1, b=0, c=0, d=0, e=1, f=40, g=0
- b) a=0, b=0, c=36, d=0, e=1, f=40, g=0
- c) a=0, b=0, c=36, d=1, e=1, f=40, g=1

14. Με την εντολή **clear** διαγράφονται όλες οι μεταβλητές που έχουμε δημιουργήσει από τη μνήμη του χώρου εργασίας.

- a) Σωστό.
- b) Λάθος.

15. Προκειμένου να μην εμφανιστεί το αποτέλεσμα που προκύπτει από την εκτέλεση μίας εντολής, εισάγουμε στο τέλος αυτής το ελληνικό ερωτηματικό “;”.

- a) Σωστό.
- b) Λάθος.

16. Ποιές μεταβλητές υπάρχουν στη μνήμη του χώρου εργασίας του MATLAB (Workspace) ύστερα από την εκτέλεση των παρακάτω εντολών (Τί θα μας εμφανίζει η εντολή who);

```
>> clear;
>> x = 5;
>> y = x + 2;
>> k = y + 200;
>> x =x+3;
>> who
```

- a) Your variables are: k x y
- b) Your variables are: ans k x
- c) Your variables are: ans k x y

17. Ποιά είναι η διαφορά της εντολής whos εντολή με την who;

- a) Η εντολή whos μας επιστρέφει περισσότερες πληροφορίες για τις μεταβλητές που υπάρχουν στη μνήμη του χώρου εργασίας, σε σχέση με την εντολή whos.
- b) Η εντολή who μας επιστρέφει πληροφορίες για περισσότερες μεταβλητές σε σχέση με την εντολή whos.

- c) Η εντολή `whos` μας επιστρέφει περισσότερες πληροφορίες για τις μεταβλητές που υπάρχουν στη μνήμη του χώρου εργασίας, σε σχέση με την εντολή `who`.

ΚΕΦΑΛΑΙΟ 6: ΔΙΑΧΕΙΡΙΣΗ ΠΙΝΑΚΩΝ ΚΑΙ ΔΙΑΝΥΣΜΑΤΩΝ

1. Στο MATLAB η δεικτοδότηση των πινάκων αρχίζει από το ένα.
 - a) Σωστό.
 - b) Λάθος.
2. Η εντολή `>> A(1,4)` εμφανίζει το στοιχείο που βρίσκεται στη πρώτη γραμμή και τέταρτη στήλη του μονοδιάστατου πίνακα $A=[15\ 67\ 32\ 89]$, δηλαδή τον αριθμό 89.
 - a) Σωστό.
 - b) Λάθος.
3. Εκτός από εντολή `K(1,1)` με την οποία εμφανίζεται το πρώτο στοιχείο ενός πίνακα K , μπορούμε να χρησιμοποιήσουμε και την εντολή `K(1)`.
 - a) Σωστό.
 - b) Λάθος.
4. Η εντολή `K(end)` εμφανίζει:
 - a) το τελευταίο στοιχείο ενός πίνακα.
 - b) το πρώτο στοιχείο ενός πίνακα.
 - c) τη λέξη `end` που βρίσκεται μέσα σε ένα πίνακα.
5. Συμπληρώστε με το κατάλληλο στοιχείο το κενό έτσι ώστε η παρακάτω εντολή να εμφανίζει όλα τα στοιχεία της δεύτερης γραμμής του πίνακα A .


```
>> A(2, __)
```
6. Ποιά από τις παρακάτω εντολές εμφανίζει όλα τα στοιχεία της τελευταίας στήλης του πίνακα A ;
 - a) `A(:,end)`
 - b) `A(end)`
 - c) `A(:,:)`
7. Ποιά από τις εντολές εμφανίζει τα στοιχεία του πίνακα που βρίσκονται στις θέσεις με έντονη κόκκινη γραμματοσειρά; $A=$

7	3	6	9
2	3	4	5
56	78	21	42
84	496	54	98

 - a) `A(:,:)`
 - b) `A(1:3,2:4)`
 - c) `A(2:4,1:3)`
8. Η εντολή `>> A(1,2)=9` είναι μία εντολή ανάθεσης και αποδίδει τον αριθμό εννέα στη θέση (1,2) του πίνακα A .
 - a) Σωστό.
 - b) Λάθος.

9. Η εντολή `>> A(:,2)=[]` :
- Διαγράφει όλα τα στοιχεία της δεύτερης γραμμής του πίνακα A.
 - Διαγράφει όλα τα στοιχεία της δεύτερης στήλης του πίνακα A.
 - Διαγράφει όλα τα στοιχεία του πίνακα A.
10. Οι διαστάσεις των πινάκων πρέπει να είναι ίσες στην πρόσθεση και την αφαίρεση πινάκων.
- Σωστό.
 - Λάθος.
11. Για τη δήλωση των πράξεων μεταξύ των πινάκων στοιχείο προς στοιχείο χρησιμοποιείται η υποδιαστολή πριν από το σύμβολο οποιασδήποτε πράξης. Έτσι το άθροισμα στοιχείο προς στοιχείο του πίνακα A και του B δηλώνεται ως εξής: `>> A,+B`
- Σωστό.
 - Λάθος.
12. Η συνάρτηση **zeros** δημιουργεί ένα πίνακα διαστάσεων $m \times n$, όπου όλα τα στοιχεία του είναι ίσα με το μηδέν.
- Σωστό.
 - Λάθος.
13. Η συνάρτηση **ones** δημιουργεί ένα πίνακα διαστάσεων $m \times n$, όπου όλα τα στοιχεία του είναι ίσα με τον αριθμό έντεκα.
- Σωστό.
 - Λάθος.
14. Ο μοναδιαίος πίνακας $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ δημιουργείται εύκολα από την εντολή:
- `ones(3)`
 - `eye(3)`
 - `rand(3)`
15. Η συνάρτηση **rand** δημιουργεί ένα $m \times n$ πίνακα με τυχαίους αριθμούς ως στοιχεία, ομοιόμορφα κατανομημένοι στο διάστημα $[0,1]$, ενώ η συνάρτηση **randn** δημιουργεί ένα $m \times n$ πίνακα με τυχαίους αριθμούς ως στοιχεία από κανονική κατανομή ($\mu=0, \sigma^2=1$).
- Σωστό.
 - Λάθος.
16. Η συνάρτηση **size** προσδιορίζει τις διαστάσεις ενός πίνακα, δηλαδή το πλήθος των γραμμών και στηλών του.
- Σωστό.
 - Λάθος.
17. Η εντολή **size(table,1)** επιστρέφει το πλήθος των γραμμών ενός πίνακα και η εντολή **size(table,2)** επιστρέφει το πλήθος των στηλών ενός πίνακα.
- Σωστό.
 - Λάθος.
18. Έστω ο πίνακας `Bathmoi=[15.4 18 17.3 12]`. Η εκτέλεση της εντολής **size(Bathmoi,1)** θα επιστρέψει τον αριθμό ____.

19. Έστω ο πίνακας $Bathmoi=[15.4 \ 18 \ 17.3 \ 12]$. Η εκτέλεση της εντολής **size(Bathmoi,2)** θα επιστρέψει:
- τον αριθμό 1.
 - τον αριθμό 4.
 - τον αριθμό 18.
20. Έστω ο πίνακας $thermokrasies= [12 \ 11 ; 12 \ 9 ; 11 \ 9 ; 10 \ 7]$. Η εκτέλεση των εντολών **size(thermokrasies,1)** και **size(thermokrasies,2)** θα επιστρέψουν:
- τον αριθμό 2 και 4 αντίστοιχα.
 - τον αριθμό 4 και 2 αντίστοιχα.
 - τον αριθμό 12 και 11 αντίστοιχα.
21. Η συνάρτηση **length** επιστρέφει τη μικρότερη τιμή από τις διαστάσεις ενός πίνακα.
- Σωστό.
 - Λάθος.
22. Συμπληρώστε το αποτέλεσμα που θα μας επιστρέψει το MATLAB ύστερα από την εκτέλεση της εντολής **>> length(thermokrasies)** , όταν ο πίνακας περιλαμβάνει τις εξής τιμές: $thermokrasies= [12 \ 11 ; 12 \ 9 ; 11 \ 9 ; 10 \ 7]$.
- Αποτέλεσμα: ____
23. Συμπληρώστε το αποτέλεσμα που θα μας επιστρέψει το MATLAB ύστερα από την εκτέλεση της εντολής **length** , σε ένα διάνυσμα τριών στηλών.
- Αποτέλεσμα: ____
24. Το πλήθος των στοιχείων ενός μονοδιάστατου πίνακα A υπολογίζεται με την εντολή:
- length(A)**
 - size(A)**
 - Όλα τα παραπάνω.
25. Η συνάρτηση **numel** προσδιορίζει το πλήθος των στοιχείων ενός πίνακα, ανεξαρτήτου μεγέθους.
- Σωστό.
 - Λάθος.
26. Η συνάρτηση **max(A)**:
- Επιστρέφει τη μέγιστη τιμή από κάθε μία στήλη του πίνακα A.
 - Επιστρέφει τη μέγιστη τιμή από κάθε μία γραμμή του πίνακα A.
 - Επιστρέφει τη μέγιστη τιμή του πίνακα A.
27. Εφόσον η εντολή **size(table,1)** επιστρέφει το μέγεθος της πρώτης διάστασης του πίνακα **table** και η εντολή **size(table,2)** το μέγεθος της δεύτερης διάστασης, το γινόμενο των δύο παραπάνω εντολών, **size(table,1) * size(table,2)**, θα επιστρέψει το πλήθος των στοιχείων του πίνακα.
- Σωστό.
 - Λάθος.

ΚΕΦΑΛΑΙΟ 7: ΕΠΙΤΡΕΠΤΑ ΜΕΓΕΘΗ (ΣΥΝΕΧΕΙΑ ΚΕΦΑΛΑΙΟΥ 4)

1. Το βασικό πλεονέκτημα ενός πίνακα κελιών σε σχέση με τους απλούς πίνακες είναι ότι επιτρέπει την ανομοιογένεια των περιεχομένων του, δηλαδή διαφορετικού τύπου δεδομένα.
 - a) Σωστό.
 - b) Λάθος
2. Στα πεδία μιας δομής αποθηκεύονται μεταβλητές πάντοτε ίδιου τύπου.
 - a) Σωστό.
 - b) Λάθος.
3. Ένας πίνακας δεν είναι δυνατόν να είναι πεδίο μιας δομής.
 - a) Σωστό.
 - b) Λάθος.
4. Ένα πεδίο μιας δομής μπορεί να περιέχει συμβολοσειρά.
 - a) Σωστό.
 - b) Λάθος.
5. Με την παρακάτω εντολή δημιουργείται μία δομή δεδομένων με το όνομα **D** και τρία πεδία τιμών, το name, surname και το age.
`>>D=struct('name',{'Χαρά','Άννα'},'surname',{'Κίρκου','Χατζή'},'age',{'54','29'})`
 - a) Σωστό.
 - b) Λάθος.
6. Η εμφάνιση όλων των πεδίων και των περιεχομένων της δεύτερης εγγραφής μιας δομής δεδομένων με όνομα Foitites πραγματοποιείται με την εντολή:
 - a) Data(2)
 - b) Foitites(2)
 - c) Foitites{2}
7. Η εμφάνιση του περιεχομένου μόνο του πεδίου age της δεύτερης εγγραφής μιας δομής δεδομένων με όνομα Foitites πραγματοποιείται με την εντολή:
 - a) age. Foitites (2)
 - b) Foitites(2).age
 - c) Foitites{2}.age
8. Ποιά από τις παρακάτω εντολές επιστρέφει το ίδιο αποτέλεσμα με την εντολή Foitites(2).age ;
 - a) setfield(Foitites (2),' age ','Ρούλα')
 - b) getfield(Foitites (2),' age ')
 - c) Όλα τα παραπάνω.
9. Η προσθήκη νέας εγγραφής σε μια υπάρχουσα δομή δεδομένων μπορεί να γίνει ανά πάσα στιγμή με ανάθεση σε ένα διαθέσιμο δείκτη.
 - a) Σωστό.
 - b) Λάθος.
10. Η προσθήκη της νέας εγγραφής ενός κυρίου με το όνομα Πελοπίδας στην υπάρχουσα δομή δεδομένων **D**:

```
D=struct('name',{'Χαρά','Άννα'},'surname',{'Κίρκου','Χατζή'},'age',{'54','29'})
```

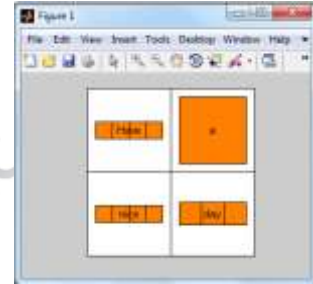
στη θέση 3, πραγματοποιείται με την εντολή:

- a) `D(3).name='Πελοπίδας'`
 - b) `setfield(D(3),'name','Πελοπίδας')`
 - c) Όλα τα παραπάνω.
11. Η προσθήκη μιας νέας εγγραφής σε υπάρχουσα δομή δεδομένων σε κάποια θέση, η οποία περιέχει ήδη δεδομένα είναι εφικτή. Ωστόσο τα δεδομένα που υπάρχουν επικαλύπτονται από τα δεδομένα της τελευταίας ανάθεσης.
- a) Σωστό.
 - b) Λάθος.
12. Η εντολή `>> D(1).address='Ποσειδώνος 15'` δημιουργεί το νέο πεδίο `address` (εφόσον δεν υπάρχει) στην υπάρχουσα δομή δεδομένων `D` και εισάγει την τιμή Ποσειδώνος 15 στην πρώτη εγγραφή.
- a) Σωστό.
 - b) Λάθος.
13. Ποιά εντολή από τις παρακάτω χρησιμοποιείται για την εμφάνιση όλων των πεδίων μίας δομής δεδομένων;
- a) `rmfield(όνομα δομής δεδομένων,'όνομα πεδίου')`
 - b) `isfield(όνομα δομής δεδομένων,'όνομα πεδίου')`
 - c) `fieldnames(όνομα δομής δεδομένων)`
14. Η εντολή `rmfield` χρησιμοποιείται:
- a) Για την αφαίρεση ενός συγκεκριμένου πεδίου από μία υπάρχουσα δομή.
 - b) Για τη δημιουργία μίας νέας δομής δεδομένων χρησιμοποιώντας μία υπάρχουσα δομή δεδομένων, αφαιρώντας ένα συγκεκριμένο πεδίο.
 - c) Όλα τα παραπάνω.
15. Ποιά εντολή από τις παρακάτω χρησιμοποιείται για τον έλεγχο ύπαρξης ενός πεδίου σε μια δομή δεδομένων;
- a) `rmfield(όνομα δομής δεδομένων,'όνομα πεδίου')`
 - b) `isfield(όνομα δομής δεδομένων,'όνομα πεδίου')`
 - c) `fieldnames(όνομα δομής δεδομένων)`
16. Η εντολή `Data ={ [1 2 ;4 5;7 8] , [2 3 4 ; 5 6 7] }` δημιουργεί ένα πίνακα κελιών μίας γραμμής και πέντε στηλών.
- a) Σωστό.
 - b) Λάθος.
17. Η εντολή `>>A(2,1)={ 'Hello world'}` εισάγει σε ένα πίνακα κελιών στη δεύτερη γραμμή και πρώτη στήλη την ακολουθία χαρακτήρων "Hello world". Αν ο πίνακας δεν υπάρχει, τον κατασκευάζει και στη συνέχεια εισάγει την ακολουθία χαρακτήρων. Τις υπόλοιπες θέσεις που δημιουργήθηκαν τις αφήνει κενές.
- a) Σωστό.
 - b) Λάθος.
18. Ποιά από τις παρακάτω εντολές χρησιμοποιείται για την απεικόνιση των περιεχομένων ενός κελιού που βρίσκεται στην πρώτη γραμμή και τρίτη στήλη του πίνακα κελιών με όνομα `Array`;

- a) Array(1,3)
 b) Array{1,3}
 c) Τίποτα από τα παραπάνω.
19. Ποιά από τις παρακάτω εντολές χρησιμοποιείται για την απεικόνιση όλων των περιεχομένων των κελιών ενός πίνακα με το όνομα **Pinakas_Mou**;
- a) `cellplot(Pinakas_Mou)`
 b) `cell(n)`
 c) `celldisp(Pinakas_Mou)`

20. Η παρακάτω εικόνα έχει δημιουργηθεί από την εντολή `cellplot`, η οποία δημιούργησε τη γραφική απεικόνιση του πίνακα κελιών
- $$A = \begin{Bmatrix} \text{'Have' 'a'} \\ \text{'nice' 'day'} \end{Bmatrix}.$$

- a) Σωστό.
 b) Λάθος.



ΚΕΦΑΛΑΙΟ 8: ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

1. Η εκτέλεση της εντολής `abs(-7)` θα επιστρέψει:
- a) τον αριθμό 7.
 b) τον αριθμό -7
 c) τίποτα από τα παραπάνω.
2. Η εντολή `sqrt(x)` υπολογίζει τη τετραγωνική ρίζα του αριθμού x.
- a) Σωστό.
 b) Λάθος.
3. Ποιά από τις παρακάτω εντολές επιστρέφει το πρόσημο του αριθμού x;
- a) `abs(x)`
 b) `sign(x)`
 c) `fix(x)`
4. Η εντολή _____ στρογγυλοποιεί τον αριθμό x και επιστρέφει τον αμέσως μικρότερο ακέραιο αριθμό του.
- a) `ceil(x)`
 b) `floor(x)`
 c) `round(x)`
5. Ποιά από τις παρακάτω εντολές στρογγυλοποιεί τον αριθμό x και επιστρέφει τον αμέσως μεγαλύτερο ακέραιο αριθμό του.
- a) `ceil(x)`
 b) `floor(x)`
 c) `round(x)`

6. Η εντολή **log(x)** υπολογίζει το λογάριθμο με βάση το e του αριθμού x , ενώ η εντολή **log10(x)** υπολογίζει το λογάριθμο με βάση το δέκα του αριθμού x .
- Σωστό.
 - Λάθος.
7. Συμπλώστε το κενό με την κατάλληλη τριγωνομετρική συνάρτηση προκειμένου να υπολογίζεται το συνημίτονο της γωνίας $\pi/6 = 30^\circ$. (Υπενθυμίζεται η μαθηματική σταθερά $\pi=3,14$ στο MATLAB συμβολίζεται με `pi`)
- _____ (`pi/6`)
8. Το αποτέλεσμα που επέστρεψε το MATLAB ύστερα από την εκτέλεση της εντολής `>> log(0)` είναι `ans = -Inf`. Αυτό σημαίνει πως:
- Η παραπάνω εντολή δεν εκτελέστηκε σωστά.
 - Ο λογάριθμος με βάση το e του αριθμού μηδέν είναι το άπειρο.
 - Τίποτα από τα παραπάνω.

ΚΕΦΑΛΑΙΟ 9: ΕΝΤΟΛΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΤΟΥ ΧΩΡΟΥ ΕΡΓΑΣΙΑΣ

- Η διαφορά της εντολής **save** και την εντολή **diary** είναι ότι με την πρώτη αποθηκεύουμε όλες τις μεταβλητές που έχουμε δημιουργήσει ενώ με τη δεύτερη αποθηκεύουμε όλες τις εντολές που έχουμε εκτελέσει, μαζί με τα αποτελέσματά τους.
 - Σωστό.
 - Λάθος.
- Τα μηνύματα λάθους που προκύπτουν από την εκτέλεση λάθος εντολών δεν αποθηκεύονται στο ASCII αρχείο που δημιουργείται από την εντολή **diary**.
 - Σωστό.
 - Λάθος.
- Η εντολή **clc** καθαρίζει την οθόνη του Παραθύρου εντολών (Command Window) από όλα τα δεδομένα, αλλά δε διαγράφει τις μεταβλητές που έχουν δημιουργηθεί.
 - Σωστό.
 - Λάθος.
- Οι εντολές **load** και **exit** χρησιμοποιούνται για την έξοδο από το MATLAB.
 - Σωστό.
 - Λάθος.
- Ποιά από τις παρακάτω εντολές κάνει την ίδια λειτουργία με την εντολή **quit**?
 - `exit`
 - `load`
 - `clc`

ΚΕΦΑΛΑΙΟ 10: ΕΙΣΟΔΟΣ ΚΑΙ ΕΞΟΔΟΣ ΔΕΔΟΜΕΝΩΝ

1. Η εντολή **input** χρησιμοποιείται για την εισαγωγή δεδομένων από το χρήστη μέσω του πληκτρολογίου.
 - a) Σωστό.
 - b) Λάθος.
2. Η εντολή **>> number=input('Please enter your phone number: ')** :
 - a) Εμφανίζει το μήνυμα "Please enter your phone number:" στο χρήστη και περιμένει την απάντησή του για να την αποθηκεύσει στη μεταβλητή number.
 - b) Εμφανίζει απλά το μήνυμα "Please enter your phone number:" στο χρήστη.
 - c) Τίποτα από τα παραπάνω.
3. Η εντολή **>> disp('MATLAB')** εμφανίζει το κείμενο **MATLAB** στην οθόνη του χρήστη.
 - a) Σωστό.
 - b) Λάθος.
4. Εφόσον το όρισμα της εντολής **disp** δεν περικλείεται σε μονά εισαγωγικά, τότε λαμβάνεται ως μεταβλητή και εκτυπώνεται μόνο το περιεχόμενό της.
 - a) Σωστό.
 - b) Λάθος.
5. Επειδή η εντολή **disp** δέχεται μόνο ένα όρισμα, η ανάγκη συνένωσης του μηνύματος κειμένου με την τιμή της μεταβλητής επιτυγχάνεται με:
 - a) τη χρήση της εντολής **strrep()**.
 - b) τη χρήση ενός πίνακα, όπου ως πρώτο στοιχείο ορίζεται το μήνυμα κειμένου και ως δεύτερο στοιχείο του ορίζεται η μεταβλητή που επιθυμούμε.
 - c) Όλα τα παραπάνω.
6. Η εντολή **format** ρυθμίζει:
 - a) τα κενά μεταξύ των γραμμών, όπου εξάγονται τα αποτελέσματα.
 - b) την εναλλαγή μεταξύ των διαφορετικών μορφών εμφάνισης των αποτελεσμάτων των αριθμών κινητής υποδιαστολής.
 - c) Όλα τα παραπάνω.
7. Η εντολή **format** δεν επηρεάζει μόνο τον τρόπο εμφάνισης των αποτελεσμάτων στο Παράθυρο εντολών και τον τρόπο υπολογισμού ή αποθήκευσής τους.
 - a) Σωστό.
 - b) Λάθος.

ΚΕΦΑΛΑΙΟ 11: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΟ MATLAB

1. Τα Αρχεία Κειμένου (script m-files) χρησιμοποιούνται στις περιπτώσεις:
 - a) όπου είναι απαραίτητη η επανάληψη εντολών και πράξεων, οι οποίες αλλάζουν κάθε φορά τιμή, χωρίς όμως να χρειάζεται η πληκτρολόγηση των εντολών ξεχωριστά.
 - b) όπου είναι απαραίτητη η εισαγωγή πολλών πληροφοριών.
 - c) Όλα τα παραπάνω.

2. Τα Αρχεία Κειμένου (script m-files) δεν έχουν ορίσματα εισόδου και εξόδου.
 - a) Σωστό.
 - b) Λάθος.

3. Το όνομα ενός Αρχείου Κειμένου (script m-file):
 - a) μπορεί να έχει το ίδιο όνομα με μία μεταβλητή, η οποία υπολογίζεται μέσω του αρχείου.
 - b) μπορεί να αρχίζει με χαρακτήρα (a...z, A...Z), αριθμό (0...9) ή ακόμα και το σύμβολο της κάτω παύλας (_,underscore)
 - c) Κανένα από τα παραπάνω.

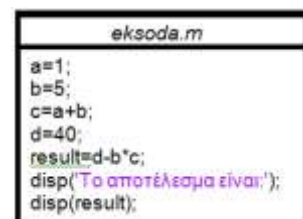
4. Οι περιορισμοί των αρχείων m-files είναι ίδιοι με αυτούς των ονομάτων των μεταβλητών.
 - a) Σωστό.
 - b) Λάθος.

5. Τί πληκτρολογούμε στο Παράθυρο Εντολών του MATLAB προκειμένου να εκτελεστούν οι εντολές ενός Αρχείου Κειμένου (script m-file);
 - a) Το όνομα του Αρχείου Κειμένου χωρίς την κατάληξη .m
 - b) Τις εντολές του Αρχείου Κειμένου.
 - c) Κανένα από τα παραπάνω.

6. Στη δεξιά εικόνα βλέπουμε τις εντολές που βρίσκονται μέσα στο αρχείο eksoda.m. Τί θα εμφανιστεί στην οθόνη του χρήστη όταν εισάγει το όνομα του αρχείου, eksoda, στο Παράθυρο Εντολών του MATLAB;
 - a) `>> eksoda`
Το αποτέλεσμα είναι:
`10`
 - b) `>> eksoda`
Το αποτέλεσμα είναι:
`210`
 - c) Κανένα από τα παραπάνω.

7. Το όνομα της συνάρτησης πρέπει να ταυτίζεται με το όνομα του Αρχείου Συνάρτησης (function m-files) χωρίς την κατάληξη .m
 - a) Σωστό.
 - b) Λάθος.

8. Στα αρχεία συνάρτησης (functions), η επικεφαλίδα αποτελείται από μία γραμμή όπου ορίζεται η συνάρτηση και καταγράφονται αναλυτικά οι μεταβλητές εισόδου και εξόδου. Αν



```
eksoda.m
a=1;
b=5;
c=a+b;
d=40;
result=d-b*c;
disp('Το αποτέλεσμα είναι:');
disp(result);
```


δεν υπάρχει αυτή τη γραμμή τότε το αρχείο συνάρτησης μετατρέπεται σε αρχείο κειμένου (script m-files).

- a) Σωστό.
- b) Λάθος.

9. Στην παρακάτω εικόνα βλέπουμε τις εντολές που βρίσκονται μέσα στο αρχείο `ονομα.m`. Τί θα εμφανίσει στο χρήστη η εκτέλεση του παρακάτω αρχείου m-file όταν εισάγει το όνομα του αρχείου, `ονομα`, στο Παράθυρο Εντολών του MATLAB και στη συνέχεια εισάγει ως όνομα το Πελοπίδα;

```
ονομα.m
name=input('Παρακαλώ εισάγετε το όνομά σας:','s');
disp(['Γειά σου ',name, '!!!']);
```

- a) `>> ονομα`
Παρακαλώ εισάγετε το όνομά σας: Πελοπίδα
Γειά σου Πελοπίδα!!!
 - b) `>> ονομα`
Πελοπίδας
Γειά σου Πελοπίδα!!!
 - c) Κανένα από τα παραπάνω.
10. Ποιό πρέπει να είναι το όνομα του Αρχείου Συνάρτησης (function m-files) εάν περιέχει μία κεντρική συνάρτηση με το όνομα `income_calculation`;
- a) `income_calculation.m`
 - b) Όποιο όνομα θέλουμε.
 - c) Όλα τα παραπάνω.
11. Στα Αρχεία Συνάρτησης (function m-files) οι μεταβλητές εισόδου είναι σε παρενθέσεις, ενώ οι μεταβλητές εξόδου βρίσκονται σε αγκύλες.
- a) Σωστό.
 - b) Λάθος.
12. Ποιός από τους παρακάτω περιορισμούς ισχύει για τα ονόματα των συναρτήσεων;
- a) Το όνομα της συνάρτησης δεν πρέπει να συμπίπτει με όνομα συνάρτησης βιβλιοθήκης.
 - b) Το όνομα της συνάρτησης ακολουθεί του ίδιους περιορισμούς με αυτούς των ονομάτων των μεταβλητών.
 - c) Όλα τα παραπάνω.
13. Τα σχόλια είναι κείμενο, το οποίο δε λαμβάνεται υπόψη από το λογισμικό και ξεκινούν με το σύμβολο `%`.
- a) Σωστό.
 - b) Λάθος.
14. Τα σχόλια μπορούν να βρίσκονται οπουδήποτε μέσα στο κώδικά μας.
- a) Σωστό.
 - b) Λάθος.
15. Η ενσωμάτωση σχολίων μέσα στον κώδικα είναι υποχρεωτική, αφού βοηθούν στην κατανόηση του κώδικα.

- a) Σωστό.
b) Λάθος.
16. Η χρήση υποσυναρτήσεων στα Αρχεία Συνάρτησης δεν είναι υποχρεωτική αλλά είναι πολύ σημαντική, αφού μειώνει τον όγκο του κώδικα της κύριας συνάρτησης.
a) Σωστό.
b) Λάθος.
17. Η παρακάτω σειρά αναγνωρίζεται ως _____ από το MATLAB.
`% It counts the income`
a) επικεφαλίδα.
b) σχόλιο.
c) εντολή τερματισμού συνάρτησης.
18. Η παρακάτω σειρά αναγνωρίζεται ως _____ από το MATLAB.
`function [bmi] = BMI(weight,height)`
d) επικεφαλίδα.
e) σχόλιο.
f) εντολή τερματισμού συνάρτησης.
19. Η παρακάτω επικεφαλίδα έχει 2 ορίσματα εξόδου.
`function [bmi] = BMI(weight,height)`
a) Σωστό.
b) Λάθος.
20. Η παρακάτω επικεφαλίδα είναι εσφαλμένη διότι:
`FUNCTION [kerdı] = Eisodima(esoda,eksoda)`
a) Η εντολή function έχει πληκτρολογηθεί με κεφαλαία γράμματα.
b) Δεν έπρεπε να υπάρχουν οι αγκύλες στη μεταβλητή εξόδου.
c) Όλα τα παραπάνω.
21. Η επικεφαλίδα `function [kerdı] = Eisodima(esoda,eksoda)` έχει:
a) Ένα όρισμα εξόδου, τη μεταβλητή kerdı και δύο ορίσματα εισόδου, τις μεταβλητές esoda και eksoda.
b) Ένα όρισμα εισόδου, τη μεταβλητή kerdı και δύο ορίσματα εξόδου, τις μεταβλητές esoda και eksoda.
c) Κανένα από τα παραπάνω.
22. Έστω η επικεφαλίδα μίας συνάρτησης `function bmi = BMI(weight,height)` η οποία υπολογίζει το Δείκτη Μάζας Σώματος ενός ατόμου. Ποιός από τους παρακάτω τρόπους κλήσης της συνάρτησης, από το Παράθυρο Εντολών του MATLAB (Command Window), είναι ο σωστός για ένα άτομο 60 κιλών και 1.65 ύψος;
a) BMI(60,1.65)
b) BMI(1.65,60)
c) BMI [1.65,60]

23. Συμπληρώστε τα κατάλληλα πεδία έτσι ώστε να δημιουργηθεί μία συνάρτηση που θα δέχεται ως όρισμα έναν οποιοδήποτε αριθμό και θα τον εμφανίζει πάλι στο χρήστη. Το όνομα της συνάρτησης είναι DisplayNumber και η μεταβλητή εξόδου η number.

```

_____ number = DisplayNumber(_____)
number=num2str(num);
disp(['Ο αριθμός που εισάγετε είναι ο : ']);
end

```

24. Συμπληρώστε τα κατάλληλα πεδία έτσι ώστε να δημιουργηθεί **μία συνάρτηση** που θα μετατρέπει μία τιμή θερμοκρασίας από βαθμούς Κελσίου(C) στην αντίστοιχη θερμοκρασία σε βαθμούς Φαρενάιτ (F). Η σχέση μεταξύ θερμοκρασίας Κελσίου (C) και Φαρενάιτ (F) είναι: $F = \frac{9}{5}C + 32$. Η συνάρτηση θα περιέχει **μία μεταβλητή εξόδου, την F** (θερμοκρασία σε βαθμούς Φαρενάιτ) και **μία μεταβλητή εισόδου τη C** (θερμοκρασία σε βαθμούς Κελσίου). Το όνομα του αρχείου που είναι αποθηκευμένη η συνάρτηση είναι **CelciusToFahrenheit.m**

FUNCTION ____ = _____(C)

F = (____)*C + 32;

end

25. Οι δομές ελέγχου ροής του προγράμματος χρησιμοποιούνται κυρίως στις περιπτώσεις όπου ένα κομμάτι του κώδικα εκτελείται συχνά ή όπου επαναλαμβάνονται τα ίδια βήματα πολλές φορές.

- Σωστό.
- Λάθος.

26. Με την εντολή **if x<5**:

- Ελέγχεται αν η τιμή της μεταβλητής x είναι μικρότερη από το πέντε.
- Καθορίζεται η τιμή της μεταβλητής x να είναι πάντα μικρότερη από το πέντε.
- Τίποτα από τα παραπάνω.

27. Τί θα τυπωθεί στο Παράθυρο Εντολών του MATLAB ύστερα από την εκτέλεση των παρακάτω εντολών:

- Adult
- Underage
- Τίποτα από τα παραπάνω.

```

x=5;
if x>=18
    disp('Adult');
else
    disp('Underage');
end

```

28. Τί θα τυπωθεί στο Παράθυρο Εντολών του MATLAB ύστερα από την εκτέλεση των παρακάτω εντολών, όταν ο χρήστης εισάγει τον αριθμό πέντε:

- Wrong choice!
- You selected the first choice.
- Τίποτα από τα παραπάνω.

```

x = input('Select a number (1-3): ');
if x==1
    disp('You selected the first choice. ');
elseif x==2
    disp('You selected the second choice. ');
elseif x==3
    disp('You selected the third choice. ');
else
    disp('Wrong choice!');
end

```

29. Η εντολή **for** χρησιμοποιείται για την επανάληψη της εκτέλεσης ενός συνόλου εντολών συγκεκριμένες φορές.
 a) Σωστό.
 b) Λάθος.
30. Τί θα τυπωθεί στο Παράθυρο Εντολών του MATLAB ύστερα από την εκτέλεση των παρακάτω εντολών:
 a) Θα εκτυπωθεί μία φορά το μήνυμα Hello!
 b) Θα εκτυπωθεί πέντε φορές το μήνυμα Hello!
 c) Τίποτα από τα παραπάνω.
- ```
for i=1:1:5
 disp('Hello!');
end
```
31. Στην εντολή **for** το βήμα-μεταβολής του δείκτη μπορεί να παραλειφθεί όταν αυτό ισούται με τη μονάδα.  
 a) Σωστό.  
 b) Λάθος.
32. Ποιά είναι η διαφορά των δύο παρακάτω εντολών:  
 a) Έχουν διαφορετικές αρχικές τιμές δείκτη.  
 b) Έχουν διαφορετικές τελικές τιμές δείκτη.  
 c) Τίποτα από τα παραπάνω.
- ```
for i=0:2:5
    x=i^2;
end
disp(x);
```
- ```
for i=1:2:5
 x=i^2;
end
disp(x);
```
33. Τί θα τυπωθεί στο Παράθυρο Εντολών του MATLAB ύστερα από την εκτέλεση των παρακάτω εντολών:  
 a) Θα εμανιστεί το τετράγωνο του αριθμού 4, δηλαδή ο αριθμός 16.  
 b) Θα εμανιστεί το τετράγωνο του αριθμού 5, δηλαδή ο αριθμός 25.  
 c) Θα εμανιστούν όλα τα τετράγωνα των αριθμύ του 0 και του 5.
- ```
for i=0:2:5
    x=i^2;
end
disp(x);
```
34. Στην περίπτωση όπου το βήμα-μεταβολής δείκτη είναι θετικό, η αρχική τιμή δείκτη πρέπει να είναι μικρότερη από την τελική τιμή δείκτη.
 a) Σωστό.
 b) Λάθος.
35. Στην περίπτωση όπου το βήμα-μεταβολής δείκτη είναι αρνητικό, η αρχική τιμή δείκτη πρέπει να είναι μικρότερη από την τελική τιμή δείκτη.
 a) Σωστό.
 b) Λάθος.
36. Συμπληρώστε τα κατάλληλα πεδία προκειμένου να υπολογίζεται το άθροισμα των πρώτων δέκα ζυγών αριθμών.

```
sum=0;
for i=2:__:10
    sum=sum+i;
end
disp(sum)
```

37. Στην περίπτωση των δισδιάστατων πινάκων είναι απαραίτητη η χρήση εμφωλευμένων βρόχων προκειμένου να εμφανίσουμε ή να μεταβάλλουμε τα περιεχόμενά του.
- Σωστό.
 - Λάθος.
38. Συμπληρώστε τα κατάλληλα πεδία προκειμένου να υπολογίζεται το γινόμενο των πρώτων είκοσι μονών αριθμών.

```
prod=1;
for i=__:2:20
    prod=prod*i;
end
disp(prod)
```

39. Έστω ο δισδιάστατος πίνακας $A = \begin{bmatrix} 2 & 5 & 6 \\ 3 & 1 & 9 \end{bmatrix}$. Συμπληρώστε τα κατάλληλα πεδία έτσι ώστε να εμφανίζεται κάθε στοιχείο του πίνακα.

```
for i=1:size(A,1)
    for j=1:size(A,2)
        A( __ , __ )
    end
end
```

40. Να συμπληρωθούν τα κενά ώστε να υπολογίζεται το άθροισμα των στοιχείων του πίνακα A:

```
for ___=1:size(A,___)
    for ___=1:size(A,___)
        s = s + A(r,c)
    end
end
```

41. Συμπληρώστε τα κατάλληλα πεδία, με τις λέξεις και τα σύμβολα που βρίσκονται στην παρένθεση, για την εύρεση της μέγιστης και της ελάχιστης τιμής του πίνακα $A = \begin{bmatrix} 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}$. ($>$, $<$, Min , Max)

```
Min = A(1,1);
Max = A(1,1);

for i=1:size(A,1)
    for j=1:size(A,2)
        if A(i,j)___ Min
            ___ = A(i,j);
        end

        if A(i,j) ___ Max
            ___ = A(i,j);
        end
    end
end
```

42. Η εντολή **while** χρησιμοποιείται στις περιπτώσεις όπου γνωρίζουμε εκ των προτέρων τον ακριβή αριθμός επαναλήψεων των εντολών.
 a) Σωστό.
 b) Λάθος.
43. Στη δομή επανάληψης **while**, αν η συνθήκη ελέγχου είναι εξαρχής ψευδής υπάρχει η πιθανότητα να μην εκτελεστεί ποτέ το σύνολο εντολών.
 a) Σωστό.
 b) Λάθος.

44. Να συμπληρωθούν τα κενά ώστε το παρακάτω πρόγραμμα να δέχεται την εισαγωγή από το πληκτρολόγιο του βαθμού ενός σπουδαστή. Σε περίπτωση που οι βαθμός που εισήγαγε δεν είναι έγκυρος, δηλαδή δεν ικανοποιείται η συνθήκη $0 \leq \text{βαθμός} \leq 10$, το πρόγραμμα πρέπει να ζητάει ξανά από το χρήστη να εισάγει τον βαθμό του μέχρις ότου αυτός να είναι έγκυρος.

```
bathmos = input('Παρακαλώ εισάγετε τον βαθμό σας: ');
while (bathmos____0 | bathmos____10 )
    bathmos = input('Μη έγκυρος βαθμός.Παρακαλώ εισάγετε πάλι τον βαθμό σας: ');
end
```

45. Να συμπληρωθούν τα κενά ώστε το παρακάτω πρόγραμμα να εμφανίζει το μήνυμα «Καλημέρα σας» 10 φορές, δηλαδή μέχρις ότου η βοηθητική μεταβλητή time ξεπεράσει το όριο των δέκα φορών.

```
_____ (time____ 10)
    disp('Καλημέρα σας.');
```

46. Να συμπληρωθούν τα κενά ώστε το παρακάτω πρόγραμμα να δέχεται την εισαγωγή ενός αριθμού από το πληκτρολόγιο, ο οποίος θα αντιστοιχεί στον αριθμό των ποδιών που έχει το αρθρόποδο που αναζητάει. Σε περίπτωση που ο αριθμός που εισήγαγε είναι το έξι θα του εμφανίζει το μήνυμα «Είναι έντομο.», ενώ στην περίπτωση που εισήγαγε τον αριθμό οκτώ θα του εμφανίζει το μήνυμα «Είναι αράχνη.». Σε κάθε άλλη περίπτωση θα του εμφανίζει το μήνυμα «Είναι πολύποδο.». Το πρόγραμμα πρέπει να εκτελείται μέχρις ότου ο χρήστης εισάγει μη έγκυρο αριθμό ποδιών, δηλαδή έναν αρνητικό αριθμό.

```
FEET = INPUT('ΠΟΣΑ ΠΟΔΙΑ ΕΧΕΙ ΤΟ ΑΡΘΡΟΠΟΔΟ; : ');
_____ FEET>0
    IF (FEET==6)
        DISP('ΕΙΝΑΙ ΕΝΤΟΜΟ.');
```

```
_____ (FEET==8)
        DISP('ΕΙΝΑΙ ΑΡΑΧΝΗ.');
```

```
ELSE
        DISP('ΕΙΝΑΙ ΠΟΛΥΠΟΔΟ.');
```

```
END
FEET = INPUT('ΠΟΣΑ ΠΟΔΙΑ ΕΧΕΙ ΤΟ ΑΡΘΡΟΠΟΔΟ; : ');
END
```

47. Να συμπληρωθούν τα κενά ώστε το παρακάτω πρόγραμμα να διαβάζει ένα ακέραιο αριθμό date από το 1 μέχρι το 7 και να εμφανίζει την αντίστοιχη ημέρα της εβδομάδας (1 για Δευτέρα, ..., 7 για Κυριακή).

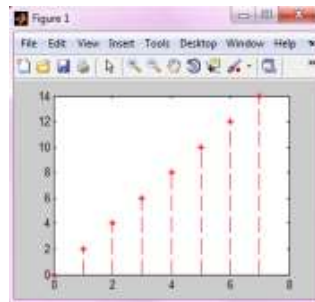
_____ = input('Παρακαλώ εισάγετε έναν ακέραιο αριθμό από το 1 έως το 7 για την ημέρα που επιθυμείτε: ');

```
switch date
    _____ 1
        disp('Η μέρα είναι ο : Δευτέρα');
    case 2
        disp('Η μέρα είναι ο : Τρίτη');
    case 3
        disp('Η μέρα είναι ο : Τετάρτη');
    case 4
        disp('Η μέρα είναι ο : Πέμπτη');
    case 5
        disp('Η μέρα είναι ο : Παρασκευή');
    case 6
        disp('Η μέρα είναι ο : Σάββατο');
    case 7
        disp('Η μέρα είναι ο : Κυριακή');
    _____
        disp('Λάθος μέρα! Ο αριθμός πρέπει να είναι ακέραιος από το 1 έως το 7! ');
end
```

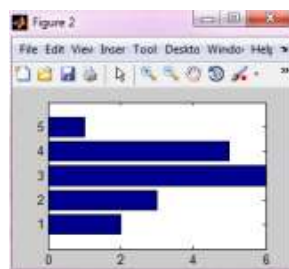
ΚΕΦΑΛΑΙΟ 12: ΓΡΑΦΙΚΑ ΣΤΟ MATLAB

- Με την εντολή **figure** δημιουργείται ένα νέο παράθυρο γραφικών στο οποίο μπορεί να απεικινιστεί μία γραφική παράσταση.
 - Σωστό.
 - Λάθος.
- Για το κλείσιμο του τρέχοντος figure χρησιμοποιείται η εντολή:
 - close all
 - close
 - clf
- Η συνάρτηση **plot** δημιουργεί γραφικές παραστάσεις καμπυλών μόνο για ισομήκη διανύσματα.
 - Σωστό.
 - Λάθος.
- Η προκαθορισμένη μορφή εμφάνισης της γραφικής παράστασης που δημιουργείται από την εντολή plot είναι πράσινα σημεία ενωμένα από μία πράσινη εννιαία γραμμή.
 - Σωστό.
 - Λάθος.
- Έστω τα διανύσματα $x=[0 -1 -2 -3 -4 -5 -6 -7]$ και $y=[10 20 40 60 80 100 120 140]$. Ποιά από τις παρακάτω μορφές σύνταξης της εντολής plot είναι σωστή;
 - plot{x,y}

- b) `plot(x,y)`
 c) `plot[x,y]`
6. Σε ποιά από τις παρακάτω μορφές σύνταξης της εντολής `plot` τα σημεία της γραφικής παράστασης έχουν οριστεί με κόκκινους αστερίσκους;
 a) `plot(x,y,'g-')`
 b) `plot(x,y,'r*')`
 c) `plot(x,y,'r')`
7. Η γραφική παράσταση που απεικονίζεται στην παρακάτω εικόνα είναι αποτέλεσμα της εντολής:



- a) `plot(x,y,'r--*')`
 b) `stem(x,y,'r--*')`
 c) `hist(x,y,'r--*')`
8. Η συνάρτηση `bar` χρησιμοποιείται για τη δημιουργία κάθετων δισδιάστατων ραβδογραμμάτων.
 a) Σωστό.
 b) Λάθος.
9. Για τη δημιουργία οριζόντιων δισδιάστατων ραβδογραμμάτων χρησιμοποιείται η συνάρτηση:
 a) `stem`
 b) `bar`
 c) `barh`
10. Η γραφική παράσταση που απεικονίζεται στην παρακάτω εικόνα είναι αποτέλεσμα της εντολής:



- d) `plot(x,y)`
 e) `barh(x,y)`
 f) `bar(x,y)`
11. Η εντολή `xlabel` χρησιμοποιείται για την εισαγωγή κειμένου, ως ετικέτας, στον άξονα x.
 a) Σωστό.

- b) Λάθος.
12. Ποιά από τις παρακάτω εντολές εισάγει ένα κείμενο στο πάνω μέρος του γραφήματος, ως τίτλο της γραφικής παράστασης;
- a) title('κείμενο')
 - b) ylabel('κείμενο')
 - c) legend('κείμενο')
13. Η εντολή **axis([XMIN XMAX YMIN YMAX])** αλλάζει τα όρια των αξόνων ορίζοντας μία αρχική και τελική τιμή για τον άξονα xx' και yy'
- a) Σωστό.
 - b) Λάθος.
14. Η εντολή **axis([0 100 0 100])**:
- a) ορίζει τους αριθμούς 0 και 100 ως αρχική και τελική τιμή αντίστοιχα στους άξονες X,Y.
 - b) ορίζει τους αριθμούς 100 και 0 ως αρχική και τελική τιμή αντίστοιχα στους άξονες X,Y.
 - c) Τίποτα από τα παραπάνω.
15. Συμπληρώστε το κενό με την κατάλληλη εντολή _____, η οποία εισάγει ένα πλαίσιο, ως υπόμνημα, σε ένα γράφημα.
16. Ποιά από τις παρακάτω εντολές δημιουργεί ένα πλέγμα στο γράφημα;
- a) grid
 - b) grid on
 - c) Όλα τα παραπάνω.
17. Με την συνάρτηση subplot μπορούμε να απεικονίσουμε μόνο δύο γραφήματα στο ίδιο παράθυρο γραφικών (figure).
- a) Σωστό.
 - b) Λάθος.
18. Ποιά από τις παρακάτω εντολές μας επιτρέπει την ταυτόχρονη απεικόνιση διαφορετικών γραφικών παραστάσεων στο ίδιο παράθυρο γραφικών (figure);
- a) grid
 - b) hold
 - c) subplot