

2010

Μελέτη Και Ανάπτυξη Πληροφοριακού Συστήματος ΤΕΙ Πάτρας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ



Θεοδώρα Μαρία
Κυλάφης Κωνσταντίνος
Φλώρος Πάυλος





T.E.I.
ΠΑΤΡΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**«ΜΕΛΕΤΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΠΛΗΡΟΦΟΡΙΑΚΟΥ
ΣΥΣΤΗΜΑΤΟΣ ΤΕΙ ΠΑΤΡΑΣ»**

ΦΟΙΤΗΤΕΣ: ΘΕΟΔΩΡΟΥ ΜΑΡΙΑ

ΚΥΛΑΦΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΦΛΩΡΟΣ ΠΑΥΛΟΣ

ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ: ΣΤΑΜΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΠΑΤΡΑ 2010

ΠΡΟΛΟΓΟΣ

Στην Πτυχιακή Εργασία μας, καλούμαστε να αναπτύξουμε τόσο σε θεωρητικό όσο και πρακτικό επίπεδο την έννοια του Πληροφοριακού Συστήματος και όλα εκείνα τα στοιχεία που θεωρούνται απαραίτητα για την κατανόηση των βασικών εννοιών που διέπουν τα Πληροφοριακά Συστήματα, τις προδιαγραφές που θα πρέπει να καλύπτει ένα τέτοιο Σύστημα και τη μελέτη των τρόπων υλοποίησής του.

Το θέμα της Πτυχιακής είναι ιδιαίτερα εξειδικευμένο και αποτελεί πρόκληση η μελέτη και ανάπτυξη ενός Πληροφοριακού Συστήματος τόσο σύνθετου όσο του ΤΕΙ της Πάτρας και αυτός αποτέλεσε και λόγος επιλογής του συγκεκριμένου θέματος Πτυχιακής.

Κύριο μέλημα μας είναι να κατανοήσουμε πρώτα σε θεωρητικό επίπεδο τις βασικές έννοιες που διέπουν τέτοια Συστήματα και κατόπιν να δημιουργήσουμε ένα λειτουργικό και ευέλικτο Πληροφοριακό Σύστημα ειδικά για ένα τόσο μεγάλο Εκπαιδευτικό Ίδρυμα, το οποίο καλείται να υπηρετήσει τις ανάγκες δεκάδων εργαζομένων του Ιδρύματος και να είναι ικανό να εξυπηρετήσει τους χιλιάδες Σπουδαστές σε όλα τα Τμήματα που εδρεύουν τόσο στην Πάτρα όσο και τα Τμήματα που εδρεύουν σε άλλες πόλεις και εντάσσονται στην δικαιοδοσία του ΤΕΙ Πάτρας.

Μέσα από την ανάλυση όλων των παραμέτρων που διέπουν τα Πληροφοριακά Συστήματα καθώς και από την διαδικασία υλοποίησης του Συστήματος μας, κατανοήσαμε την μοναδική χρησιμότητα ενός Πληροφοριακού Συστήματος τόσο για έναν οργανισμό όσο και για μία μεγάλη επιχείρηση.

Τέλος εκτιμήσαμε πολύπλευρα όλες τις γνώσεις που αποκομίσαμε από μία τέτοια Εργασία, την οποία ολοκληρώθηκε με χρήση εξειδικευμένης βιβλιογραφίας και χαιρόμαστε πολύ την φέραμε εις πέρας.

ΠΕΡΙΛΗΨΗ

Στο νέο τεχνολογικό περιβάλλον η πληροφορία αποτελεί πολυτιμότεο μέρος του ενεργητικού ενός οργανισμού, ανοίγοντας νέες προοπτικές στα πλαίσια της λειτουργίας αυτού διότι παρέχει στα συστήματα μια νέα δυναμική που συμβάλει στην επιβίωση ενός οργανισμού, την ομαλή λειτουργία του, την ανάπτυξη του και την ανταγωνιστικότητά του. Από επιχειρηματική άποψη, η Θεωρία των Πληροφοριακών Συστημάτων αποτελεί μια οργανωτική και διοικητική λύση στις απαιτήσεις του περιβάλλοντος ενός οργανισμού. Ουσιαστικά συντονίζει και κατευθύνει τη διοίκηση, την οργάνωση και την υπάρχουσα Πληροφοριακή Τεχνολογία προς τον εντοπισμό των επιχειρηματικών λύσεων τις οποίες απαιτούν οι προκλήσεις του ανταγωνισμού. Όσον αφορά τη σχέση αποφάσεων και σχεδιασμού ενός Πληροφοριακού Συστήματος είναι ευρέως γνωστό ότι οι απαιτήσεις των διοικητικών στελεχών από τα πληροφοριακά συστήματα είναι συνάρτηση των αυξανόμενων αναγκών τους σε πληροφόρηση μέσα σε ένα σύνθετο και ραγδαία μεταβαλλόμενο περιβάλλον.

Ένα πληροφοριακό σύστημα αποτελεί ένα ιδεατό κατασκεύασμα το οποίο δημιουργείται για να αντιπροσωπεύσει μία φυσική οντότητα, η οποία υπάρχει μέσα σε έναν οργανισμό. Οι συνιστώσες που καλύπτουν τη λειτουργία ενός πληροφοριακού συστήματος είναι το Υλικό, το Λογισμικό, το Προσωπικό και οι Επικοινωνίες και ανάλογα με τα χαρακτηριστικά που επιθυμούμε να χαρακτηρίζουν το Πληροφοριακό μας Σύστημα διακρίνονται διάφοροι τύποι όπως τα Συστήματα Επεξεργασίας Δοσοληψιών, τα Γνωστικά Συστήματα Εργασίας, τα Συστήματα Αυτοματισμού Γραφείου, τα Πληροφοριακά Συστήματα Διοίκησης, τα Συστήματα Υποστήριξης Αποφάσεων και τα Συστήματα Υποστήριξης Της Εκτελεστικής Εξουσίας. Η πορεία ενός Πληροφοριακού Συστήματος, από τη στιγμή του καθορισμού του προβλήματος ή των προβλημάτων που καλείται να επιλύσει, μέχρι τη λειτουργία του, τη συντήρησή του και, τέλος, την απόσυρσή του, είναι γνωστή στη βιβλιογραφία ως Κύκλος Ζωής του Πληροφοριακού Συστήματος και ακολουθεί σχεδόν πάντα μία συγκεκριμένη πορεία: δημιουργείται, αναπτύσσεται, εξελίσσεται, και τελικά, αποσύρεται.

Σκοπός της σχεδίασης ενός Πληροφοριακού Συστήματος είναι οι διαμόρφωση των παραγόντων που επιδρούν πάνω σε μία εύκολη χρήση του συστήματος από τους χρήστες τους και στην άνεση που μπορεί αυτό να τους προσφέρει. Τέτοιοι παράγοντες μπορεί είναι Κοινωνικοί, Φυσικοεργονομικοί, και Ψυχολογικοεργονομικοί, η οποίοι σχετίζονται με το σχεδιασμό του λογισμικού και επιδρούν πάνω στη γνωστική – αντιληπτική άνεση του χρήστη, όπως επίσης και η διαθεσιμότητα και η ανταπόκριση του Συστήματος.

Έχοντας ως στόχος την ικανοποίηση των παραπάνω δεδομένων, το Πληροφοριακό Σύστημα που καλούμαστε να σχεδιάσουμε και να αναπτύξουμε, μπορεί να δημιουργηθεί είτε με Διαγραμματικές Τεχνικές (οι οποίες είναι ευρύτερα διαδεδομένες στην ανάπτυξη του μοντέλου ενός Πληροφοριακού Συστήματος) είτε με την Προμήθεια και Παραγωγή του Λογισμικού. Ένας συνδυασμός των καλύτερων πρακτικών, οι οποίες ήδη έχουν αποδείξει πόσο επιτυχημένες ήταν στη μοντελοποίηση μεγάλων και σύνθετων συστημάτων είναι η UML, μία γλώσσα που χρησιμοποιείται για αναπαράσταση με οπτικό τρόπο, δημιουργία και τεκμηρίωση των τμημάτων των συστημάτων λογισμικού, καθώς και για μοντελοποίηση εταιρικών και άλλων συστημάτων που δεν αφορούν λογισμικό. Πολλές εταιρίες την έχουν ενσωματώσει ως πρότυπο στη διαδικασία ανάπτυξής τους και στα προϊόντα τους.

Αυτό συμβαίνει διότι όσο αυξάνει για πολλές εταιρίες η στρατηγική αξία του λογισμικού, η βιομηχανία αναζητά νέες τεχνικές για αυτοματοποίηση της παραγωγής αυτού ώστε να βελτιωθεί η ποιότητα και να μειωθεί το κόστος.

Το πρόβλημα της υποστήριξης και της ανάπτυξης λογισμικού οδήγησε και στη «γέννηση» ενός ιδιαίτερου Κλάδου της Τεχνολογίας Λογισμικού, ο οποίος ονομάστηκε Τεχνολογία Λογισμικού Υποβοηθούμενη από Υπολογιστή – CASE. Η υποστήριξη που παρέχουν τα εργαλεία CASE στους μηχανικούς λογισμικού στοχεύει στη βελτίωση της παραγωγικότητας, αλλά και της ποιότητας του λογισμικού. Έτσι πολλά από τα εργαλεία CASE που χρησιμοποιούνται σήμερα αποτελούν αυτόνομες λύσεις σε μεμονωμένα προβλήματα. Από την άλλη, η λήψη λανθασμένων αποφάσεων και γενικά οι μη ακριβείς εκτιμήσεις σχετικά με ένα εργαλείο CASE, είναι πιθανό να έχει τα αντίθετα από τα αναμενόμενα αποτελέσματα. Δηλαδή, μπορεί να μειώνει την παραγωγικότητα και την ποιότητα και να αυξάνει το κόστος παραγωγής του λογισμικού.

Εξετάζοντας κάθε θεωρητικό κομμάτι που διέπει τα Πληροφοριακά Συστήματα και προκειμένου να διαπιστώσουμε τη πολύπλευρη χρησιμότητά τους σε έναν οργανισμό τόσο πολύπλοκο όσο ένα Εκπαιδευτικό Σύστημα, δημιουργήσαμε μία βάση δεδομένων (επεξεργαστείτε την στο cd- rom που εσωκλείεται), η οποία αφορά το ΤΕΙ Πάτρας, και όλους τους παράγοντες που μπορούν να υποστηριχθούν από ένα τέτοιο Σύστημα, όπως το σύνολο των Φοιτητών ανά Τμήμα, τους Καθηγητές με τα μαθήματα που διδάσκουν ανά Τμήμα, τις Γραμματείες του κάθε Τμήματος κ.ο.κ.

Δημιουργώντας τη Βάση Δεδομένων του ΤΕΙ Πάτρας, διαπιστώνουμε πόσο συμβάλει στην ομαλή οργάνωση και διεξαγωγή των βασικών εργασιών σε κάθε τμήμα του ΤΕΙ Πάτρας επιβεβαιώνοντας τον ορισμό Avital (2003) σύμφωνα με τον οποίο ένα «Πληροφοριακό Σύστημα συμβάλει τα μέγιστα στη συλλογή, αποθήκευση, παραγωγή και διανομή των πληροφοριών που εξυπηρετούν τις πληροφοριακές ανάγκες ενός οργανισμού και υποστηρίζουν τις δραστηριότητές του, τόσο σε επίπεδο διαχείρισης και λειτουργίας, όσο και σε επίπεδο σχεδιασμού και λήψης αποφάσεων για τον οργανισμό».

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	
ΠΕΡΙΛΗΨΗ.....	
ΕΙΣΑΓΩΓΗ.....	
ΚΕΦΑΛΑΙΟ 1: ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ.....	1
Ενότητα 1.1: Πληροφοριακή Τεχνολογία – Η Έννοια Του Συστήματος.....	2
Ενότητα 1.2: Συνιστώσες Του Πληροφοριακού Συστήματος.....	3
Ενότητα 1.3: Χαρακτηριστικά Πληροφοριακών Συστημάτων.....	13
Ενότητα 1.4: Κατηγορίες Πληροφοριακών Συστημάτων.....	13
Ενότητα 1.5: Τύποι Πληροφοριακών Συστημάτων.....	15
Ενότητα 1.6: Υποσυστήματα Πληροφοριακού Συστήματος.....	21
Ενότητα 1.7: Στάδια Εξέλιξης Πληροφοριακού Συστήματος Ενός Οργανισμού.....	22
Ενότητα 1.8: Η Έννοια της Ολοκλήρωσης – Ολοκληρωμένα Συστήματα.....	23
Ενότητα 1.9: Κύκλος Ζωής Πληροφοριακού Συστήματος.....	24
Ενότητα 1.10: Διαδικασία Λήψης Αποφάσεων Και Πληροφοριακά Συστήματα.....	30
ΚΕΦΑΛΑΙΟ 2: ΜΕΘΟΔΟΙ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	36
Ενότητα 2.1: Επικοινωνία Ανθρώπου - Μηχανής.....	37
Ενότητα 2.2: Σχεδιασμός Συστημάτων – Διαγραμματικές τεχνικές.....	38
Ενότητα 2.3: Σχεδιασμός Συστημάτων – Προμήθεια Και Παραγωγή Λογισμικού.....	43
2.3.1 Αγορά Λογισμικού.....	43
2.3.2 Προγραμματισμός.....	45
Ενότητα 2.4: Κύκλος Ζωής Λογισμικού – Γλώσσες Προγραμματισμού.....	45
2.4.1 Κύκλος Ζωής Λογισμικού.....	45
2.4.2 Γλώσσες Προγραμματισμού.....	48
2.4.3 Τεχνικές Σχεδίασης Προγραμμάτων.....	51
Ενότητα 2.5: Μοντελοποιώντας Με Χρήση Της UML.....	52
2.5.1 Κίνητρα Για τον Ορισμό Της UML – Το πεδίο Εφαρμογής.....	52
2.5.2 Κύρια Στοιχεία Και Χαρακτηριστικά Της UML.....	54
2.5.3 Διαγράμματα Στατικής Δομής Της UML.....	55
2.5.4 Διάγραμμα Περιπτώσεων Χρήσης – Use Case Diagram.....	56
2.5.5 Διαγράμματα Ακολουθίας – Sequence Diagrams.....	60
2.5.6 Διαγράμματα Συνεργασίας – Collaboration Diagrams.....	62
2.5.7 Διαγράμματα Κατάστασης - Statechart Diagrams.....	64
2.5.8 Διαγράμματα Δραστηριότητας – Activity Diagrams.....	66
2.5.9 Διαγράμματα Υλοποίησης – Implementation Diagrams.....	68
Ενότητα 2.6: Εργαλεία CASE.....	71
2.6.1 Χαρακτηριστικά Και Πλαίσιο Λειτουργίας Των Εργαλείων Case..	71
2.6.2 Ταξινόμηση Των Εργαλείων Case.....	74
2.6.3 Αξιολόγηση Και Επιλογή Εργαλείων Case.....	78

ΚΕΦΑΛΑΙΟ 3: ΔΗΜΙΟΥΡΓΙΑ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ Τ.Ε.Ι

<u>ΠΑΤΡΑΣ</u>	80
Ενότητα 3.1: Γενικές Παρατηρήσεις Για την Βάση Δεδομένων Που θα Δημιουργήσουμε.....	81
Ενότητα 3.2: Δημιουργία Πινάκων ΤΕΙ Πάτρας.....	83
Ενότητα 3.3 Δημιουργία Σχέσεων Μεταξύ Πινάκων ΤΕΙ Πάτρας.....	86
Ενότητα 3.4: Δημιουργία Φόρμας Βάσεων Δεδομένων ΤΕΙ Πάτρας.....	89
Ενότητα 3.5: Δημιουργία Ερωτημάτων Βάσεων Δεδομένων ΤΕΙ Πάτρας.....	91
Ενότητα 3.6: Δημιουργία Εκθέσεων Βάσεων Δεδομένων ΤΕΙ Πάτρας.....	102
ΣΥΜΠΕΡΑΣΜΑΤΑ	108
ΒΙΒΛΙΟΓΡΑΦΙΑ	109

ΕΙΣΑΓΩΓΗ

Το πρώτο Κεφάλαιο της Πτυχιακής μας εργασίας αναφέρεται σε εκείνα τα θεωρητικά στοιχεία που σχετίζονται με τις Βασικές Έννοιες των Πληροφοριακών Συστημάτων και που είναι πολύ χρήσιμα να γνωρίζει ο αναγνώστης, ώστε να μπορέσει να κατανοήσει τις περαιτέρω πληροφορίες που αναφέρονται στα παρακάτω Κεφάλαια που απαρτίζουν την Εργασία μας. Έτσι στο Κεφάλαιο αυτό αναφέρονται οι ορισμοί των Π.Σ., τα χαρακτηριστικά, οι συνιστώσες, οι τύποι, τα στάδια εξέλιξης του Π.Σ. καθώς επίσης και η άμεση επίδραση του Πληροφοριακού Συστήματος στη λήψη αποφάσεων.

Στο δεύτερο Κεφάλαιο δίδεται όσο το δυνατόν πληρέστερη και εκτενέστερη εικόνα των λειτουργιών, των διεργασιών και των παραγόντων που συνίσταται στην επιτυχή σχεδίαση λογισμικού για την δημιουργία ενός Πληροφοριακού Συστήματος. Πιο συγκεκριμένα αναφέρονται οι Διαγραμματικές Τεχνικές ανάπτυξης λογισμικού, οι Προγραμματιστικές Έννοιες και οι Γλώσσες Προγραμματισμού, η Γλώσσα Μοντελοποίησης UML και τα Διαγράμματα αυτής καθώς επίσης γίνεται αναφορά στα Εργαλεία CASE και την χρησιμότητά τους.

Το τρίτο και τελευταίο Κεφάλαιο της Εργασίας μας περιγράφει την Βάση Δεδομένων που δημιουργήσαμε μέσω τις ACCESS 2007. Αναφέρει λεπτομερώς όλες τις ενέργειες που πραγματοποιήσαμε για να δημιουργήσουμε την Βάση Δεδομένων μας καθώς επίσης παρουσιάζονται με οπτικό τρόπο κάποια Ερωτήματα, Φόρμες και Εκθέσεις από αυτές που δημιουργήσαμε στη Βάση μας.

ΚΕΦΑΛΑΙΟ 1

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Στο παρόν κεφάλαιο γίνεται επιγραμματική αναφορά σε βασικά θέματα που σχετίζονται με την έννοια των πληροφοριακών συστημάτων, και τα οποία θεωρούνται απαραίτητα για την κατανόηση των γραφομένων που έπονται στα κεφάλαια της πτυχιακής μας εργασίας με θέμα: «Μελέτη και ανάπτυξη πληροφοριακού συστήματος ΤΕΙ Πάτρας».



Information Systems

1.1 Πληροφοριακή Τεχνολογία - Η έννοια του συστήματος

Η ανάπτυξη της **Πληροφοριακής Τεχνολογίας και των Τηλεπικοινωνιών** δίνει στα συστήματα μια νέα δυναμική και τους ανοίγει νέες προοπτικές στα πλαίσια της λειτουργίας του οργανισμού. Είναι πλέον αδιανόητο να θεωρούνται τα συστήματα ερήμην αυτών των τεχνολογιών. Το Πληροφοριακό Σύστημα βασίζεται πλέον στον Η/Υ και ονομάζεται Πληροφοριακό Σύστημα Βασισμένο στον Ηλεκτρονικό Υπολογιστή (Computer Based information Systems- CBIS).

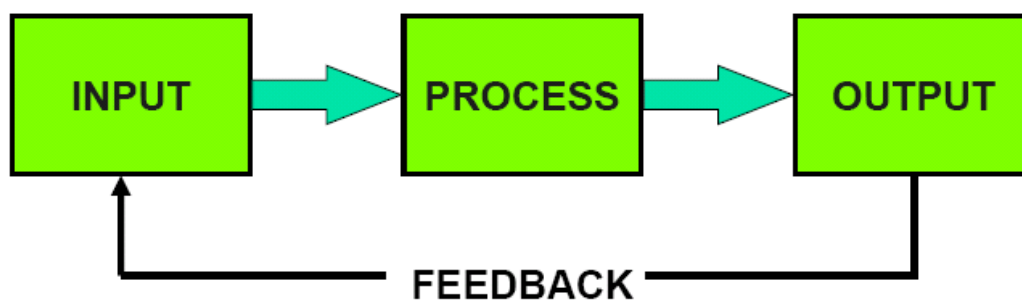
Στο νέο τεχνολογικό περιβάλλον η πληροφορία αποτελεί πολυτιμότεο μέρος του ενεργητικού του οργανισμού. Αποτελεί βασικότατο στοιχείο για την επιβίωση του οργανισμού, την ομαλή λειτουργία του, την ανάπτυξη του και την ανταγωνιστικότητά του. Αυτή η νέα σημασία της πληροφορίας δίνει νέα διάσταση στη σχέση της οργάνωσης και των συστημάτων. Διότι:

- ∅ Πολλές χειρωνακτικές εργασίες αυτοματοποιούνται. Οι επεξεργασίες των δεδομένων με τη βοήθεια του Η/Υ θα καλούνται εφεξής «διεργασίες».
- ∅ Οι διοικητικές δραστηριότητες αφενός και το πληροφοριακό σύστημα αφετέρου αλληλεπιδρούν και αλληλοεξαρτώνται.
- ∅ Ο οργανισμός αποκτά πληροφοριακή αρχιτεκτονική. Γίνεται διαχωρισμός των βασικών διεργασιών του Η/Υ από την παροχή πληροφόρησης στα διάφορα επίπεδα της οργανωτικής πυραμίδας.

Στη γενικότερη έννοια ένα σύστημα (system) είναι ένα σύνολο συνιστωσών (components) οι οποίες αλληλεπιδρούν μεταξύ τους για την επίτευξη κάποιου κοινού σκοπού. Οι συνιστώσες αυτές μπορεί να είναι όντα, υλικά, ιδέες, αξίες κ.τ.λ. Τα διάφορα μέρη του συστήματος είναι με τη σειρά τους συστήματα σε μικρότερη κλίμακα αλλά αποτελούν υποσυστήματα του αρχικού συστήματος. Σε κάθε περίπτωση ένα σύστημα είναι υπερσύστημα κάποιων συστημάτων αλλά αποτελεί ταυτόχρονα υποσύστημα κάποιου άλλου συστήματος.

Κάθε σύστημα δέχεται Είσοδο (Input), που με τη βοήθεια Επεξεργασιών (Processings), τη μετασχηματίζει σε Έξοδο (Output). Η αλληλεπίδραση μεταξύ των διαφόρων συστημάτων επιτυγχάνεται όταν ένα σύστημα χρησιμοποιεί σαν είσοδο την έξοδο κάποιων άλλων συστημάτων.

Η **Θεωρία των Πληροφοριακών Συστημάτων** άρχισε να αναπτύσσεται κατά τη διάρκεια της δεκαετίας του 1950 στο χώρο των Θετικών Επιστημών και καθορίζεται από συγκεκριμένα χαρακτηριστικά. Από επιχειρηματική άποψη το **Πληροφοριακό Σύστημα Επιχειρήσεων** (ΠΣΕ) αποτελεί μια οργανωτική και διοικητική λύση στις απαιτήσεις του περιβάλλοντος της επιχείρησης. Ουσιαστικά συντονίζει και κατευθύνει με ενιαίο τρόπο τη διοίκηση, την οργάνωση και την υπάρχουσα Πληροφοριακή τεχνολογία προς τον εντοπισμό των επιχειρηματικών λύσεων τις οποίες απαιτούν οι προκλήσεις του ανταγωνισμού. Ταυτόχρονα οι απαιτήσεις του ανταγωνισμού μορφοποιούν ανάλογα το πληροφοριακό σύστημα.



Ορισμός 1

<<Πληροφοριακό σύστημα είναι εκείνος ο μηχανισμός, ο οποίος παρέχει τα μέσα για τη συλλογή, την αποθήκευση, την παραγωγή και τη διανομή των πληροφοριών που εξυπηρετούν τις πληροφοριακές ανάγκες ενός οργανισμού και υποστηρίζουν τις δραστηριότητές του, τόσο σε επίπεδο διαχείρισης και λειτουργίας, όσο και σε επίπεδο σχεδιασμού και λήψης αποφάσεων για τον οργανισμό(Avital, 2003).>>

Ορισμός 2

«Πληροφοριακό σύστημα είναι ένα σύστημα το οποίο δέχεται πληροφορίες, τις αποθηκεύει, τις ανακτά, τις μετασχηματίζει, τις επεξεργάζεται και τις διανέμει στους διάφορους χρήστες του οργανισμού, χρησιμοποιώντας υπολογιστές ή άλλα μέσα». [Aktas (1987), Ahitur και Neumann (1990)]

1.2 Συνιστώσες Του Πληροφοριακού Συστήματος

Κάθε πληροφοριακό σύστημα αποτελείται από τέσσερις βασικές Συνιστώσες (Components). Οι συνιστώσες αυτές καλύπτουν τη λειτουργία του πληροφοριακού συστήματος και είναι οι εξής:

A) Το Υλικό (Hardware): Περιλαμβάνει το ηλεκτρονικό - μηχανικό μέρος των χρησιμοποιούμενων Η/Υ. Κεντρικό σημείο αναφοράς είναι ο ηλεκτρονικός υπολογιστής. Καθένας από αυτούς συντίθεται από επιμέρους Μονάδες (Units), οι οποίες ποικίλουν ως προς την ποσότητα και τη ποιότητα και οι οποίες καθαρίζουν τη Σύνθεση (Configuration) του Η/Υ. Αυτές οι μονάδες κατηγοριοποιούνται ανάλογα με τη συνεισφορά τους στις διεργασίες ως εξής:

Ø Κεντρική Μονάδα Επεξεργασίας: Εδώ εκτελούνται όλες οι διεργασίες των δεδομένων. Μια διεργασία αποτελείται από εντολές. Εκτέλεση της διεργασίας από τον ηλεκτρονικό υπολογιστή σημαίνει την εκτέλεση των εντολών, οι οποίες την αποτελούν, από τον Επεξεργαστή (Processor) του Η/Υ. Οι προς εκτέλεση εντολές καταχωρούνται στη Κεντρική Μνήμη (Main Memory) του Η/Υ, διαβιβάζονται στον επεξεργαστή για εκτέλεση και τα εξαγόμενα της εκτέλεσης επιστρέφουν στη κεντρική μνήμη. Η απόδοση του υπολογιστικού συστήματος του Η/Υ εξαρτάται από την

απόδοση του επεξεργαστή, την απόδοση της μνήμης αλλά και από την ποιότητα της συνεργασίας επεξεργαστή και μνήμης.

Ø Μονάδες Εισόδου: Ο ρόλος των Μονάδων Εισόδου (Data Input Units) είναι να επιτρέπουν την είσοδο των δεδομένων στον Η/Υ. Τέτοιες είναι οι:

1. **Τερματικές Οθόνες**. Αποτελούνται, είτε από μία οθόνη και ένα πληκτρολόγιο (ή και από «ποντίκι»). Η εισαγωγή των δεδομένων γίνεται από το χρήστη με πληκτρολόγηση ή χειρισμό του ποντικιού. Επιτρέπουν το διάλογο χρήστη και Η/Υ, τη λήψη αποφάσεων, τις διορθώσεις κατά την εισαγωγή των δεδομένων και είναι ευρύτατα διαδεδομένες λόγω της ευκολίας κ.λπ.
2. **Ταμειακές Μηχανές**. Η διαφορά τους από τις τερματικές οθόνες είναι ότι περιέχουν οθόνη περιορισμένων δυνατοτήτων και ότι είναι δυσκολότερες οι διορθώσεις των εισερχομένων δεδομένων.
3. **Μονάδες Οπτικής Και Μαγνητικής Αναγνώρισης**. Ο χρήστης εισάγει τα δεδομένα χειριζόμενος το σύστημα αναγνώρισης. Η ταχύτητα εισαγωγής είναι μεγάλη και η χρήση τους πολύ εύκολη. Ακόμη δεν αποκτούν συχνή συντήρηση και τιμή τους είναι προσιτή.
4. **Αισθητήρες Και Χρονόμετρα**. Οι αισθητήρες επιτρέπουν την άμεση είσοδο δεδομένων στον Η/Υ, τα οποία προέρχονται από μηχανικές, ηλεκτρικές και ηλεκτρονικές εγκαταστάσεις. Τα χρονόμετρα επιτρέπουν την είσοδο χρονικών δεδομένων τα οποία καθορίζουν τη λειτουργία των διενεργειών.
5. **Κάμερες**. Επιτρέπουν την είσοδο εικόνων αφού τις μετατρέψουν σε ψηφιακά σήματα.
6. **Μικρόφωνα**. Επιτρέπουν την είσοδο ήχου μετατρέποντάς τον σε ψηφιακά σήματα.

Ø Μονάδες Εξόδου: Οι Μονάδες Εξόδου (Data Output Units) επιτρέπουν την έξοδο της πληροφορίας από τον Η/Υ, και τη διάθεσή της στους εξουσιοδοτημένους χρήστες προς χρήση. Το περιεχόμενο και η μορφή της εξερχόμενης πληροφορίας καθορίζουν σε μεγάλο βαθμό την ποιότητα λήψης των αποφάσεων οι οποίες στηρίζονται σε αυτά τα δεδομένα. Τέτοιες Μονάδες Εξόδου είναι:

1. **Εκτυπωτές**: Η έξοδος της πληροφόρησης γίνεται με απλή εκτύπωση σε χαρτί, διαφάνειες κ.λπ.
2. **Σχεδιογράφοι**: Δημιουργούν σχέδια χρησιμοποιώντας κατάλληλες γραφίδες. Οι συχνότερα χρησιμοποιούμενοι είναι οι σχεδιογράφοι τυποποιημένων διαστάσεων χαρτιού και οι σχεδιογράφοι συνεχούς χαρτιού.
3. **Οθόνες**: Σε σχέση με τις προηγούμενες μονάδες εξόδου, παρουσιάζουν κάποια ιδιαίτερα χαρακτηριστικά όπως δεν απαιτούν αναλώσιμα, επιτρέπουν την εμφάνιση κίνησης σχημάτων και εικόνων και τέλος επιτρέπουν και μεταβολές χρωματισμού, σχημάτων, προοπτικής, γραμματοσειράς κλπ.
4. **Ηχεία**: Επιτρέπουν την έξοδο των δεδομένων με ηχητική μορφή.

5. **Ολογράμματα:** Είναι ραγδαία εξελισσόμενες μονάδες εξόδου. Γίνεται προβολή της εξερχόμενης πληροφορήσης στο χώρο με στερεοσκοπική μορφή (τρισδιάστατη εικόνα), χρησιμοποιώντας laser.

Ø Μονάδες Αποθήκευσης Δεδομένων: Οι μονάδες αποθήκευσης των δεδομένων (data store units) επιτρέπουν την αποθήκευση των εισαχθέντων δεδομένων στο Η/Υ για να χρησιμοποιηθεί όταν αυτό απαιτηθεί. Ονομάζονται ακόμη και περιφερειακά αποθήκευσης ή βοηθητικές μνήμες. Ορισμένες φορές αυτές οι μονάδες χρησιμοποιούνται και για την είσοδο δεδομένων σε άλλον Η/Υ. Οι συχνότερα χρησιμοποιούμενες μονάδες αποθήκευσης δεδομένων είναι:

1. Οι Μαγνητικές Ταινίες (Tapes).
2. Οι Κασέτες.
3. Οι Σκληροί Μαγνητικοί Δίσκοι (Hard Disks).
4. Οι Δισκέτες (Floppy Disks).
5. Οι Οπτικοί Δίσκοι (Optical Disks).

Οι μονάδες, εκτός της κεντρικής μονάδας επεξεργασίας, ονομάζονται και Περιφερειακά (Peripheral) του Η/Υ. Έτσι η ελάχιστη σύνθεση ενός Η/Υ περιλαμβάνει:

1. Μια Κεντρική Μονάδα Επεξεργασίας.
2. Μια Μονάδα Εισόδου.
3. Μια Μονάδα Εξόδου.
4. Μια Μονάδα Βοηθητικής Μνήμης.

B) Λογισμικό: Αποτελείται από διάφορα προγράμματα τα οποία αντιστοιχούν στις διεργασίες στις οποίες υποβάλλονται τα δεδομένα. Σε κάθε Η/Υ μπορεί να υπάρχουν αρκετές κατηγορίες προγραμμάτων.

Ø Λειτουργικό Σύστημα: Το Λειτουργικό Σύστημα (Operating System) αποτελείται από ένα σύνολο προγραμμάτων τα οποία έχουν δημιουργηθεί από ειδικούς και συνοδεύουν το Υλικό κατά την αγορά του. Πρόκειται για κατηγορία Λογισμικού απαραίτητη για την λειτουργία του Η/Υ. Μερικοί από τους στόχους του λειτουργικού συστήματος είναι:

1. Η εύκολη και αποδοτική χρήση του Η/Υ από τον χρήστη.
2. Η ασφάλεια του Η/Υ.
3. Η ομαλή διεκπεραίωση των διεργασιών του Η/Υ.

Το λειτουργικό σύστημα είναι σε μεγάλο βαθμό υπεύθυνο για την ασφαλή και αποδοτική λειτουργία του Η/Υ. Προσβολή του σημαίνει προσβολή ολόκληρου του συστήματος Η/Υ με απρόβλεπτες συνέπειες για την ακεραιότητα των δεδομένων. Έτσι ασφάλεια του λειτουργικού συστήματος σημαίνει:

1. Το λειτουργικό σύστημα προστατεύεται από τους χρήστες.
2. Το λειτουργικό σύστημα προστατεύεται από το περιβάλλον.
3. Το λειτουργικό σύστημα αυτοπροστατεύεται.
4. Οι χρήστες προστατεύονται από άλλους χρήστες.

5. Οι χρήστες αυτοπροστατεύονται.

Αν και υπάρχει μεγάλη ποικιλία λειτουργικών συστημάτων, όλα ταξινομούνται σε δύο κατηγορίες. Αυτές είναι:

1. **Ενός χρήστη (One User):** Πρόκειται για λειτουργικά συστήματα τα οποία χρησιμοποιούνται από προσωπικούς Η/Υ. Είναι απλά και αμετάβλητα, περιορισμένων δυνατοτήτων, ιδιαίτερα φιλικά προς τους χρήστες. Γνωστότερος εκπρόσωπος αυτής της κατηγορίας είναι το MS-DOS.
2. **Πολλών χρηστών (Multi-User):** Πρόκειται για λειτουργικά συστήματα τα οποία χρησιμοποιούνται σε Η/Υ οι οποίοι εξυπηρετούν πολλούς χρήστες παράλληλα. Είναι σύνθετα, μεγάλων και μεταβλητών δυνατοτήτων. Γνωστότερος εκπρόσωπος αυτής της κατηγορίας είναι η UNIX.

Ø Μεταγλωττιστές Προγραμμάτων: Οι Μεταγλωττιστές (Compilers) αποτελούνται από εξειδικευμένα προγράμματα τα οποία μετατρέπουν προγράμματα, γραμμένα σε κάποια προγραμματισμού, σε προγράμματα γραμμένα σε γλώσσα μηχανής, που είναι η μόνη που μπορεί να αντιληφθεί ο Η/Υ. Ανεξάρτητα από τη χρησιμοποιούμενη γλώσσα προγραμματισμού, κάθε πρόγραμμα εφαρμογής θα πρέπει να εκφραστεί σε γλώσσα μηχανής. Για κάθε γλώσσα προγραμματισμού αντιστοιχεί και ένας μεταγλωττιστής. Δεν υπάρχουν επί του παρόντος μεταγλωττιστές οι οποίοι εξυπηρετούν περισσότερες από μία γλώσσα προγραμματισμού. Το λογισμικό αυτό θα πρέπει να συνοδεύει την αγορά του Η/Υ.

Ø Εργαλεία: Πρόκειται για αυτοτελή προγράμματα ή σύνολα προγραμμάτων γενικής χρήσης, τα οποία διευκολύνουν, είτε τον προγραμματισμό του Η/Υ, είτε τη λειτουργία του Η/Υ. Τα Utilities είναι μεμονωμένα προγράμματα τα οποία παρέχουν λύσεις σε γενικά προβλήματα τα οποία πιθανόν να απασχολούν πολλούς χρήστες. Όταν απαιτηθεί η λύση ενός τέτοιου προβλήματος το πρόγραμμα εφαρμογής καλεί το αντίστοιχο utility. Τα Tools είναι σύνολα προγραμμάτων τα οποία παρέχουν διευρυμένες γενικές δυνατότητες (π.χ. διόρθωση αρχείων, ανάκτηση σβησμένων αρχείων κ.λπ.). Και αυτό το λογισμικό συνοδεύει την αγορά του Η/Υ.

Ø Σύστημα Διαχείρισης Βάσης Δεδομένων: Το σύστημα Διαχείρισης (Διοίκησης) Βάσης Δεδομένων – ΣΔΒΔ (Data Base Management System) αποτελείται από ένα σύνολο προγραμμάτων τα οποία χρησιμοποιούνται για τη διαχείριση των δεδομένων όταν αυτά είναι οργανωμένα σε Βάση Δεδομένων. Η κυριότερη λειτουργία του είναι ότι επιτρέπει στους χρήστες της Βάσης Δεδομένων να την χρησιμοποιούν χωρίς να ασχολούνται με τα τεχνικά προβλήματα της διασφάλισης των δεδομένων. Κάθε Βάση Δεδομένων συνοδεύεται και από το αντίστοιχο Σύστημα Διαχείρισης Βάσης Δεδομένων.

Ø Προγράμματα Εφαρμογής: Η λειτουργία κάθε οργανισμού διαχωρίζεται σε τομείς λειτουργίας και κάθε τομέας του εξυπηρετεί ένα σύνολο προγραμμάτων. Ας θεωρηθεί, για παράδειγμα, ότι οι βασικότεροι τομείς λειτουργίας ενός οργανισμού γενικότερα είναι:

1. Πωλήσεις και προμήθειες.
2. Μισθοδοσία και προσωπικό.
3. Πληρωμές και εισπράξεις.

4. Μετατροπές, απογραφές, διαχείριση αποθεμάτων.
5. Διαχείριση ταμείου και χαρτοφυλακίου.

Τότε, ανεξάρτητα από τον τομέα, σε καθένα από αυτούς αντιστοιχεί ένα σύνολο προγραμμάτων τα οποία ονομάζονται **Προγράμματα Εφαρμογής** (Application Programs) και το σύνολο αυτών των προγραμμάτων αποτελούν το **Σύστημα Εφαρμογής** (Application System) αυτού του τομέα.

Αυτά τα προγράμματα δημιουργούνται, χρησιμοποιώντας κάποια γλώσσα προγραμματισμού, είτε από εξειδικευμένο προσωπικό του οργανισμού (Κέντρο Πληροφοριακής Τεχνολογίας), είτε από εξειδικευμένες εταιρίες παραγωγής Λογισμικού (Software Houses).

Ø **Πακέτα Προγραμμάτων:** Τα Πακέτα (Packages) είναι σύνολα προγραμμάτων εφαρμογής τα οποία έχουν δημιουργηθεί από εταιρίες Λογισμικού και ανταποκρίνονται σε τυποποιημένες λειτουργίες ομοειδών οργανισμών. Τα Πακέτα παρουσιάζουν τα ακόλουθα χαρακτηριστικά:

1. Είναι φθηνότερα στην προμήθειά τους από την παραγωγή τους.
2. Δεν μπορεί να τα μεταβάλει ο χρήστης παρά μόνο ο δημιουργός τους.
3. Δεν συνδέονται συνήθως με άλλα προγράμματα εφαρμογής ή με άλλα πακέτα και για αυτό το λόγο προσανατολίζονται συνήθως στην εξυπηρέτηση ανεξαρτήτων IS.
4. Είναι ιδιαίτερα χρήσιμα όταν εξυπηρετούν πολύπλοκα ή ιδιόμορφα IS.

Ø **Προγράμματα σε JCL:** Πρόκειται για προγράμματα τα οποία αποτελούνται από εντολές του λειτουργικού συστήματος. Τα προγράμματα αυτά κατευθύνουν προγραμματισμένα τη ροή των εργασιών του Η/Υ χωρίς την παρέμβαση του ανθρώπινου παράγοντα. Έτσι, για παράδειγμα, μπορεί να καθοριστεί η αλληλοδιαδοχή των προς εκτέλεση προγραμμάτων μιας μέρας από την προηγούμενη μέρα, είτε καθορίζοντας σειριακά τη διαδοχή αυτής της εκτέλεσης, είτε χρονοπρογραμματίζοντάς την, είτε και τα δύο μαζί.

Γ) Προσωπικό: Σε κάθε πληροφοριακό σύστημα μπορούν να διακριθούν οι ακόλουθες κατηγορίες προσωπικού:

1. **Τα Διοικητικά Στελέχη:** Ασχολούνται με τη διοίκηση – διαχείριση του πληροφοριακού συστήματος.
2. **Το Παραδοσιακό Προσωπικό:** Πρόκειται για προσωπικό το οποίο δεν έχει άμεση σχέση με τον Η/Υ αλλά υποστηρίζει έμμεσα τη λειτουργία του. Στη κατηγορία αυτή ανήκουν, για παράδειγμα:

- Ø Όσοι ασχολούνται με την έρευνα αγοράς.
- Ø Όσοι ασχολούνται με την κοστολόγηση των παρεχομένων, από τον Η/Υ, υπηρεσιών.
- Ø Όσοι, με διάφορους τρόπους, προσκομίζουν πληροφορίες στο πληροφοριακό σύστημα.

3. **Οι τελικοί χρήστες (End Users):** Είναι οι τελικοί αποδέκτες των υπηρεσιών του. Τα απαιτούμενα εφόδιά τους είναι μόνο η γνώση χειρισμού του πληροφοριακού συστήματος. Πρόκειται για την κατηγορία του προσωπικού

προς την οποία στρέφεται η παροχή υπηρεσιών του πληροφοριακού συστήματος. Οι ανάγκες και οι απαιτήσεις τους καθορίζουν και την πορεία εξέλιξής του πληροφοριακού συστήματος.

4. **Το Προσωπικό της Πληροφοριακής Τεχνολογίας (Computerpeople):** Στη κατηγορία αυτή του προσωπικού, η οποία υπάρχει μόνο στη περίπτωση ύπαρξης στον οργανισμό Κέντρου Πληροφοριακής Τεχνολογίας, περιλαμβάνονται οι Ειδικοί του Συστήματος Η/Υ.

∅ Προσωπικό Της Πληροφοριακής Τεχνολογίας: Συνήθως διακρίνονται οι παρακάτω ειδικότητες:

1. **Χειριστής Η/Υ (Computer Operator):** ασχολείται με το χειρισμό της κεντρικής μονάδας και των περιφερειακών μονάδων καθώς και με την επίβλεψη της καλής λειτουργίας τους.
2. **Χειριστής Δικτύου (Network Operator):** Ασχολείται με τη διαχείριση του δικτύου επικοινωνιών.
3. **Προγραμματιστής Εφαρμογής (Application Programmer):** Δημιουργεί προγράμματα στη βάση των προδιαγραφών τις οποίες του παρέχει κάποιος Αναλυτής – Προγραμματιστής ή Αναλυτής συστήματος.
4. **Προγραμματιστής Συστήματος (System Programmer):** Σχεδιάζει και κωδικοποιεί μέρη Λειτουργικών Συστημάτων, βοηθητικές ρουτίνες, τμήματα προγραμμάτων επικοινωνίας, και προσαρμόζει το Λογισμικό σε εγκαταστάσεις χρηστών.
5. **Προγραμματιστής – Αναλυτής Εφαρμογής (Programmer – Analyst):** Αναλύει και σχεδιάζει μέρος μιας εφαρμογής ή ολόκληρη μικρή εφαρμογή δημιουργώντας και τα αντίστοιχα προγράμματα.
6. **Αναλυτής Συστήματος (System Analyst):** Σχεδιάζει και αναπτύσσει συστήματα Λογισμικού και επικοινωνίας δεδομένων, συνεργάζεται στενά με τους χρήστες συστημάτων εφαρμογής τα οποία αναπτύσσει.
7. **Μηχανικός Συστήματος (System Engineer):** Σχεδιάζει, εποπτεύει τη συντήρηση συστημάτων Η/Υ ή τμημάτων τους.
8. **Διαχειριστής Δεδομένων (Data Administrator):** Σχεδιάζει, εποπτεύει και συντηρεί τα δεδομένα του πληροφοριακού συστήματος.
9. **Διαχειριστής βάσης Δεδομένων (Database Administrator):** Σχεδιάζει, εποπτεύει, συντηρεί το σύστημα Βάσεων Δεδομένων.

Δ) Επικοινωνίες: Οι επικοινωνίες αφορούν τη γεωγραφική μετάβαση των δεδομένων:

- ∅ Μεταξύ των περιφερειακών και της κεντρικής μονάδας του Η/Υ.
- ∅ Μεταξύ διαφορετικών Η/Υ οι οποίοι συνδέονται μεταξύ τους.

Οι επικοινωνίες απαιτούν την ύπαρξη ενός μέσου μετάδοσης το οποίο καλείται **Γραμμή ή Δίαυλος (Channel)** επικοινωνίας.

Η Έννοια του Δικτύου: Τα μέσα τα οποία χρησιμοποιούνται για την μετάδοση των δεδομένων αποτελούν το **Δίκτυο Επικοινωνίας Δεδομένων (Data Communication Network)**. Στην περίπτωση κατά την οποία συνδέονται και επικοινωνούν μεταξύ τους

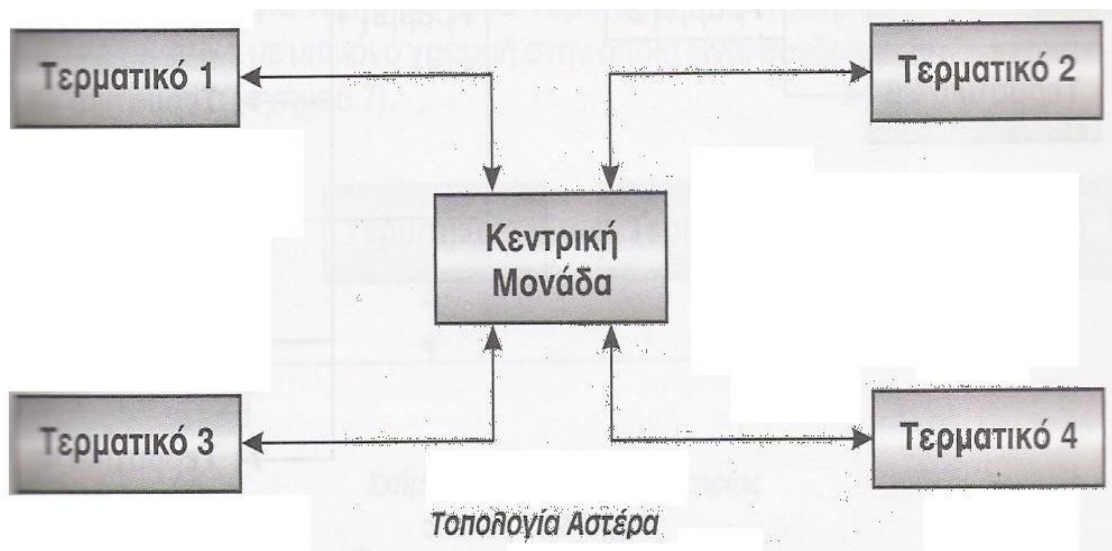
διαφορετικοί Η/Υ το προκύπτουν δίκτυο ονομάζεται Δίκτυο Υπολογιστών. Έτσι τα δίκτυα ανάλογα με τα βασικά τους χαρακτηριστικά ταξινομούνται σε διάφορες κατηγορίες.

Ως προς την Τοπολογία τους: Με τον όρο Τοπολογία Δικτύου εννοείται η διάταξη του Υλικού του Δικτύου (Network Topology), εννοείται η διάταξη του Υλικού του δικτύου σε σχέση με τις γραμμές επικοινωνίας. Κάθε διάταξη περιέχει εγγενώς τα ποιοτικά μειονεκτήματα/πλεονεκτήματα της ανεξάρτητα από την ποιότητα του Υλικού. Η ποιότητα του Υλικού επηρεάζει τα μειονεκτήματα/πλεονεκτήματα του δικτύου μόνο ποσοτικά.

Στην **Τοπολογία Αστέρα (Star Topology)** η κεντρική μονάδα του κεντρικού Η/Υ συνδέεται απευθείας και με ιδιαίτερη γραμμή με τα τερματικά.

Πλεονεκτήματα: Μπορούν να χρησιμοποιούνται απλά τερματικά, υπάρχει καλή απόδοση (αποκλειστική γραμμή για κάθε τερματικό) και μεγάλη αξιοπιστία (η βλάβη μιας γραμμής δεν διακόπτει την επικοινωνία με άλλα τερματικά) .

Μειονεκτήματα: Υψηλό κόστος δικτύου εξαιτίας των πολλών γραμμών επικοινωνίας.

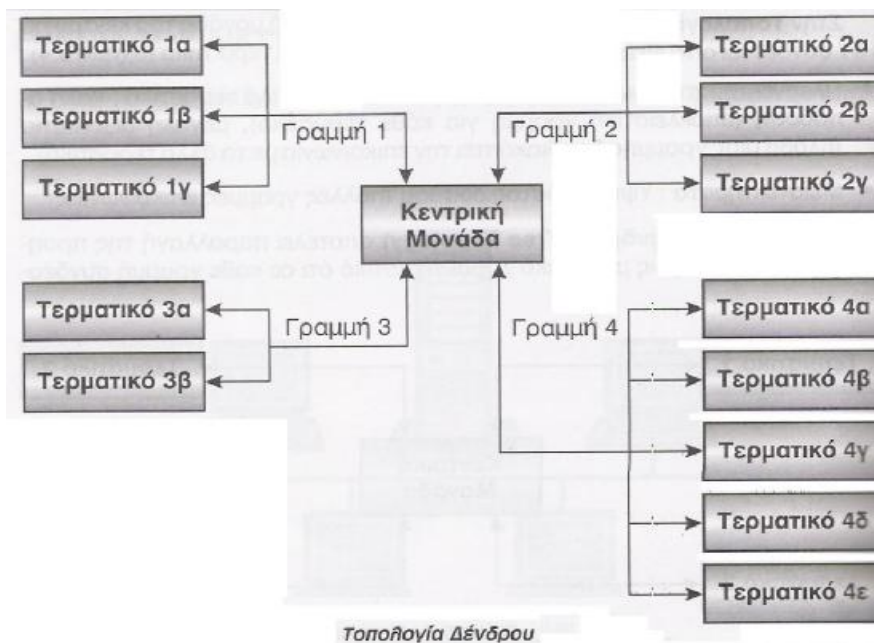


Η **Τοπολογία Δένδρου (Tree Topology)** αποτελεί παραλλαγή της προηγούμενης τοπολογίας με βασικό χαρακτηριστικό ότι σε κάθε γραμμή συνδέονται περισσότερα του ενός τερματικά. Ο καθορισμός του τερματικού το οποίο δέχεται ή αποστέλλει μηνύματα, γίνεται από την κεντρική μονάδα ή τον μετωπικό επεξεργαστή.

Πλεονεκτήματα: Μικρότερο κόστος γραμμών για τον ίδιο λόγο που εξηγήσαμε και στην Τοπολογία Αστέρα.

Μειονεκτήματα: Συνθετότερη τοπολογία και μικρότερη απόδοση.

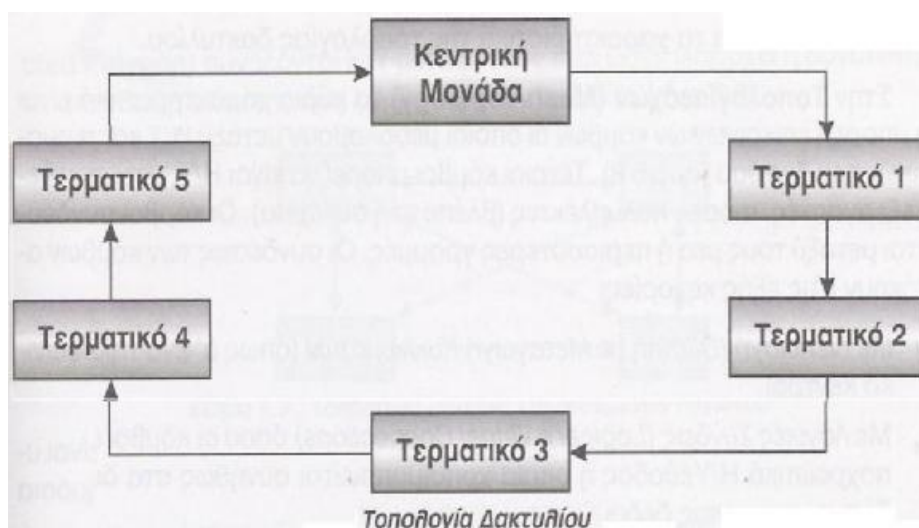
Η χρήση πολυπλεκτών αυξάνει την απόδοση αυτής της τοπολογίας του δικτύου.



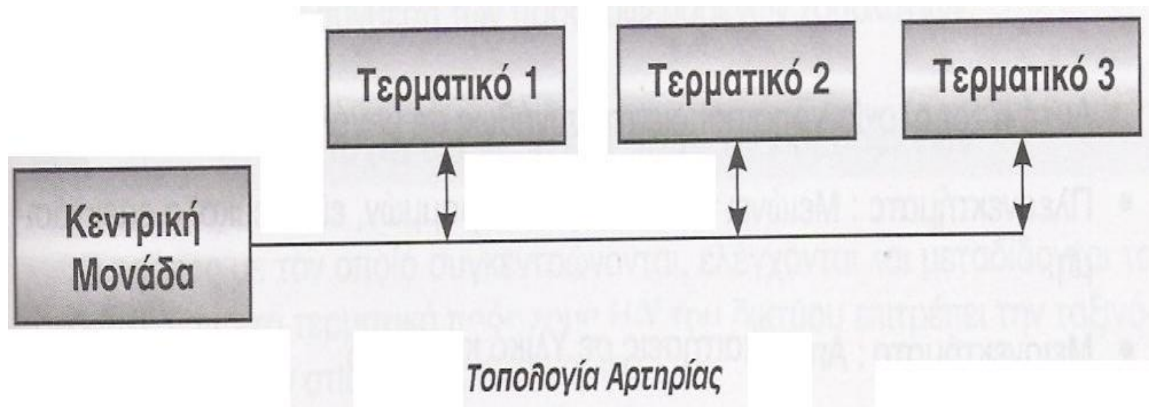
Στην **Τοπολογία Δακτυλίου (Ring Topology)** η κεντρική μονάδα και τα τερματικά συνδέονται με γραμμές σχηματίζοντας δακτύλιο. Τα μηνύματα μεταδίδονται κατά μήκος του δακτυλίου με την ίδια πάντα φορά και περιέχουν τη διεύθυνση του τερματικού ή της κεντρικής μονάδας τα οποία αποτελούν τον αποδέκτη του μηνύματος. Κάθε τερματικό συλλαμβάνει όλα τα μηνύματα τα οποία διαρρέουν το δακτύλιο, αλλά επεξεργάζεται μόνο εκείνα τα οποία περιέχουν τη διεύθυνση του συγκεκριμένου τερματικού. Το τερματικό-αποστολέας στέλνει με το μήνυμα και τη διεύθυνση προορισμού.

Πλεονεκτήματα: Μικρό κόστος δικτύου.

Μειονεκτήματα: Μία βλάβη σε οποιοδήποτε σημείο του δικτύου διακόπτει τη λειτουργία ολόκληρου του δικτύου, μικρή απόδοση γραμμών σε περίπτωση μεγάλης διακίνησης μηνυμάτων.



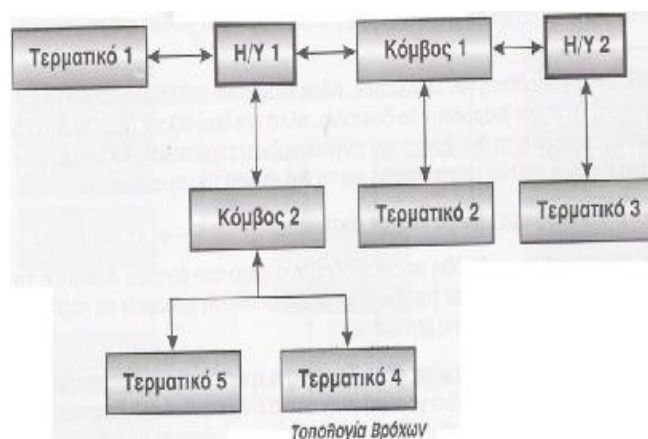
Στην **Τοπολογία Αρτηρίας (Bus Topology)** η κεντρική μονάδα συνδέεται με τα τερματικά με μία μόνο γραμμή στην οποία είναι συνδεδεμένα τα τερματικά παράλληλα και παρουσιάζει όλα τα χαρακτηριστικά της τοπολογίας δακτυλίου.



Στην **Τοπολογία Βρόχων (Mesh Topology)** το κύριο χαρακτηριστικό είναι η ύπαρξη επικοινωνιακών κόμβων οι οποίοι μεσολαβούν μεταξύ Η/Υ και τερματικών του δικτύου. Τέτοιοι κόμβοι μπορεί να είναι Η/Υ, μετωπικοί επεξεργαστές, modems ή πολυπλέκτες. Οι κόμβοι συνδέονται μεταξύ τους με μία ή περισσότερες γραμμές. Οι συνδέσεις των κόμβων ανήκουν στις εξής κατηγορίες:

- ∅ Με **Διεπιλογή (Dial-up)** ή με **Μεταγωγή Κυκλωμάτων**, όπως σε ένα τηλεφωνικό κέντρο.
- ∅ Με **Λογικές Συνδέσεις (logical ή Virtual Connections)** όπου οι κόμβοι είναι υποχρεωτικά Η/Υ, μέθοδος η οποία χρησιμοποιείται συνήθως στα δημόσια δίκτυα επικοινωνίας δεδομένων.

Η τοπολογία αυτή χρησιμοποιείται συνήθως σε μεγάλα δίκτυα και έχει σαν πλεονεκτήματα το μειωμένο κόστος γραμμών και ότι είναι εύκολα προσπελάσιμη. Από την άλλη μεριά υπάρχουν αυξημένες απαιτήσεις τόσο σε Υλικό, όσο και σε Λογισμικό.



Ως προς την ιεραρχία ελέγχου και ροής των δεδομένων: Ο τρόπος με τον οποίο συγκεντρώνονται, ελέγχονται και μεταδίδονται τα δεδομένα από τα τερματικά προς τους Η/Υ του δικτύου επιτρέπει την ταξινόμηση των δικτύων στις ακόλουθες κατηγορίες:

Στα **Μη Ιεραρχικά Δίκτυα (Non-Hierarchical Networks)** τα τερματικά συνδέονται κατευθείαν με τον Η/Υ και ελέγχονται από αυτόν. Το δίκτυο δεν περιλαμβάνει συγκεντρωτές. Οι τοπολογίες οι οποίες χρησιμοποιούνται είναι τύπου αστέρα, δένδρου, δακτυλίου ή αρτηρίας. Η άλλη διάκριση είναι σε **Ιεραρχικά Δίκτυα (Hierarchical Networks)**. Εδώ τα τερματικά συνδέονται με τον Η/Υ μέσω συγκεντρωτών ή σύνθετων πολυπλεκτών, οι οποίοι και τα ελέγχουν. Οι γραμμές «τερματικών – μονάδων ελέγχου τερματικών» ελέγχονται από τις μονάδες ελέγχου τερματικών. Οι γραμμές «μονάδων ελέγχου τερματικών – συγκεντρωτών» αποτελούν τα Δίκτυα Τοπικής Πρόσβασης (local Access Networks) και ελέγχονται από τους συγκεντρωτές. Οι γραμμές των «συγκεντρωτών – Η/Υ» αποτελούν τη Σπονδυλική Στήλη (Backbone) του δικτύου και ελέγχονται από τον Η/Υ. Κατά αυτόν τον τρόπο καθορίζεται μία ιεραρχία ελέγχου κατά τη ροή των δεδομένων.

Ως προς την ευρύτητα: Τα **Τοπικά Δίκτυα (Local Area Networks - LAN)** καλύπτουν περιορισμένες γεωγραφικές περιοχές. Χρησιμοποιούν ιδιωτικές καλωδιακές γραμμές με υψηλή απόδοση αλλά το μήκος τους δεν υπερβαίνει συνήθως τα 6 χιλιόμετρα. Τα **Δίκτυα Ευρείας Περιοχής (Wide Area Networks – WAN)** καλύπτουν μεγάλες γεωγραφικές περιοχές (πόλεις, χώρες, ηπείρους). Χρησιμοποιούν κυρίως δημόσιες τηλεφωνικές και δορυφορικές γραμμές. Τέλος τα **Μεικτά Δίκτυα** αποτελούν συνδυασμό των δύο προηγούμενων περιπτώσεων. Ένα ή περισσότερα τοπικά δίκτυα συνδέονται μεταξύ τους με τη βοήθεια τηλεφωνικής ή δορυφορικής γραμμής.

Ως προς την τεχνική μεταγωγής: Με τον όρο Μεταγωγή (Switching) εννοείται το δρομολόγιο το οποίο ακολουθεί η μετάδοση ενός μηνύματος μέσω των γραμμών. Συνήθως χρησιμοποιούνται οι ακόλουθες τεχνικές μεταγωγής:

Στην **Μεταγωγή Κυκλώματος (Circuit Switching)** η επικοινωνία μεταξύ δυο σταθμών επιτυγχάνεται με τη δημιουργία ενός φυσικού κυκλώματος μεταξύ τους (π.χ. τηλεφωνική επικοινωνία δυο συνδρομητών ΟΤΕ). Στη **Μεταγωγή Μηνύματος (Message Switching)** κάθε μήνυμα προωθείται προς τον αποδέκτη του σταδιακά, αποθηκευμένο και προωθούμενο από κόμβο σε κόμβο ανάλογα με τις δυνατότητες του δικτύου. Στη **Μεταγωγή Πακέτου (Packet Switching)** κάθε μήνυμα χωρίζεται σε τμήματα τα οποία προωθούνται καθένα ξεχωριστά με την τεχνική της αποθήκευσης – προώθησης από κόμβο σε κόμβο προς τον τελικό αποδέκτη τους.

Ως προς την κυριότητα και χρήση τους: Τα **Ιδιωτικά Δίκτυα** χρησιμοποιούνται από μία ή περισσότερες εταιρίες. Οι χρησιμοποιούμενες γραμμές ανήκουν, είτε στους ιδιοκτήτες των Η/Υ (τοπικά δίκτυα), είτε σε τρίτους (π.χ. ΟΤΕ). Ο ιδιοκτήτης σε κάθε περίπτωση είναι αποκλειστικός αρμόδιος για τον έλεγχο του δικτύου. Στα **Δημόσια Δίκτυα (Public Data Networks)** αποτελούνται από επικοινωνιακούς κόμβους και γραμμές οι οποίοι ανήκουν στην κυριότητα του δημοσίου και παρέχουν υπηρεσίες σε ιδιωτικά συστήματα και δίκτυα. Έτσι ένα τοπικό δίκτυο ή ένα σύστημα μιας εταιρίας μπορεί να συνδεθεί με δημόσιο δίκτυο και να έχει σαν χρήστες του τους χρήστες του δημοσίου δικτύου, φυσικά με την ανάλογη χρέωση.

1.3 Χαρακτηριστικά Πληροφοριακών Συστημάτων

Σύμφωνα με τους DeLone και McLean (1992) η **Θεωρία των Πληροφοριακών Συστημάτων** διέπεται από τα ακόλουθα χαρακτηριστικά:

- ∅ Διαίρεση των συστημάτων σε κατηγορίες, έτσι ώστε να διευρύνονται και να ελέγχονται τα ειδικά χαρακτηριστικά τους.
- ∅ Αντιμετώπιση ενός συστήματος ως σύνολο για την επίλυση των προβλημάτων του.
- ∅ Ανάπτυξη μοντέλων για την ανάλυση της λειτουργίας ενός συστήματος.
- ∅ Δυναμική παρέμβαση στο σύστημα (τί είναι και πώς λειτουργεί).
- ∅ Εισροές (υλικά, πληροφορίες, ανθρώπινο δυναμικό, πόροι κλπ.).
- ∅ Διαδικασίες μετατροπής (μηχανισμοί σχεδιασμού, οργάνωσης και ελέγχου).
- ∅ Δραστηριότητες παραγωγής (έρευνα και ανάπτυξη, κλπ.).
- ∅ Εκροές (υπηρεσίες, προϊόντα, ιδέες, κλπ.).

1.4 Κατηγορίες Πληροφοριακών Συστημάτων

Η οργανωτική πυραμίδα του οργανισμού περιλαμβάνει τέσσερα βασικά επίπεδα σε σχέση με το εργασιακό προφίλ του προσωπικού αυτά τα επίπεδα αντιστοιχούν ανάλογα πληροφοριακά συστήματα:

1. Στρατηγικό Επίπεδο (Strategic level)

Περιλαμβάνει τα Επιτελικά Διοικητικά Στελέχη (Senior Managers) τα οποία ασχολούνται με την χάραξη στρατηγικής της επιχείρησης. Στο επίπεδο αυτό αντιστοιχούν τα **Στρατηγικά Συστήματα** (Strategic Systems), τα οποία βοηθούν τα επιτελικά διοικητικά στελέχη να προσαρμόσουν μακροπρόθεσμα το εσωτερικό περιβάλλον της επιχείρησης στις μεταβολές του εξωτερικού περιβάλλοντος(π.χ. ποια θα είναι μακροπρόθεσμα η εξέλιξη των τιμών των πρώτων υλών).

2. Διοικητικό Επίπεδο (Management level)

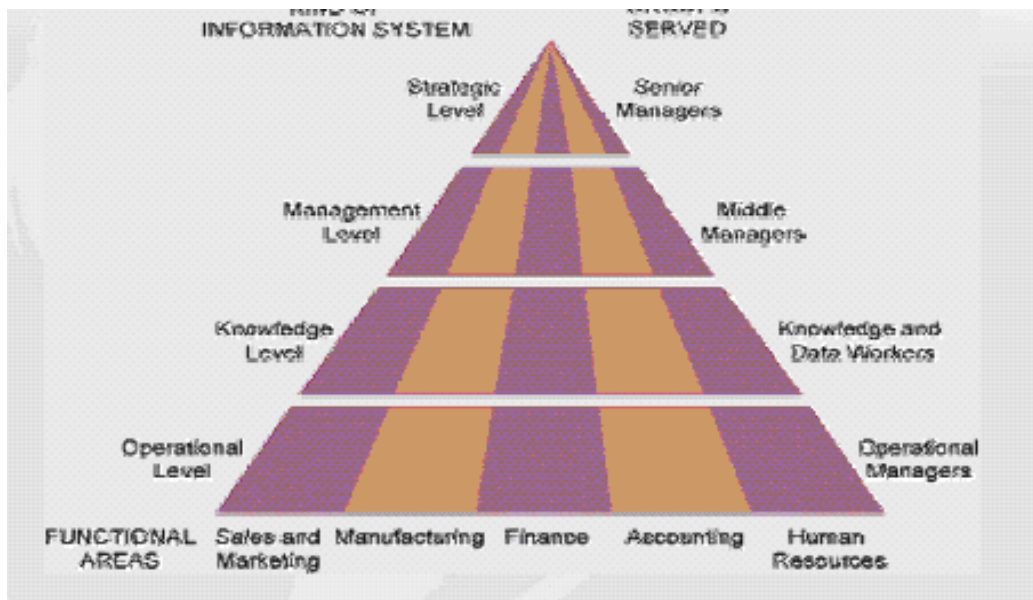
Σε αυτό το επίπεδο ανήκουν τα Μεσαία Διοικητικά Στελέχη (Middle Managers), τα οποία ασχολούνται με την διεκπεραίωση των προγραμμάτων και των σχεδίων των επιτελικών διοικητικών στελεχών. Τα **Διοικητικά Συστήματα** (Management Systems) εξυπηρετούν τους χρήστες αυτού του επιπέδου σε σχέση με την διαχείριση, τον έλεγχο, την λήψη αποφάσεων και τις διοικητικές δραστηριότητες. Προσφέρουν πληροφόρηση η οποία απαιτείται για να απαντηθεί το ερώτημα: <<Ο οργανισμός λειτουργεί σωστά>>.

3. Γνωστικό Επίπεδο (Knowledge level)

Αυτό το επίπεδο περιλαμβάνει εξειδικευμένα στελέχη (π.χ. μηχανικοί) και προσωπικό το οποίο ασχολείται με την διαχείριση δεδομένων (data workers) και τα **Γνωστικά Συστήματα** (Knowledge Systems) βοηθούν στην ενσωματωμένη και εφαρμογή των νέων γνώσεων σε ολόκληρο τον οργανισμό (π.χ. πως καταπολεμείται η γραφειοκρατία)

4. Λειτουργικό Επίπεδο (Operational level)

Περιλαμβάνει τα λειτουργικά Διεκπεραιωτικά Διοικητικά Στελέχη (Operational Managers), τα οποία ασχολούνται με την διεκπεραίωση των καθημερινών εργασιών της επιχείρησης. Τα **Λειτουργικά Συστήματα** (Operational Systems) στηρίζουν αυτά τα διοικητικά στελέχη αναφορικά με τις καθημερινές δοσοληψίες της επιχείρησης και παρέχουν πληροφόρηση ικανή να απαντηθούν ερωτήματα τα οποία προκύπτουν από αυτές τις δοσοληψίες (π.χ. ποιος ο σημερινός τζίρος: ποίον το χρεωστικό ή πιστωτικό υπόλοιπο των σημερινών συναλλαγών).



Σχήμα 1.1 Σχέση Επιπέδων Οργάνωσης και Κλάδων Λειτουργίας

Στο Σχήμα 1.1 τα τρίγωνα τα οποία ορίζονται από την κορυφή της πυραμίδας και τη βάση της καθορίζουν τα συστήματα των κλάδων λειτουργίας, τα τμήματα της πυραμίδας τα οποία ορίζονται από τις οριζόντιες γραμμές και τις πλευρές της καθορίζουν τα συστήματα των διαφόρων οργανωτικών επιπέδων και οι τομές των τριγώνων και των τμημάτων καθορίζουν τα τμήματα των συστημάτων των κλάδων λειτουργίας τα οποία ανήκουν στα αντίστοιχα οργανωτικά επίπεδα.

1.5 Τύποι Πληροφοριακών Συστημάτων

Όλα τα συστήματα που έχουν αναφερθεί παραπάνω ανήκουν σε έναν από τους ακόλουθους τύπους πληροφοριακών συστημάτων. Οι τύποι αυτοί είναι:

A) Συστήματα Επεξεργασίας Δοσοληψιών (Transaction Processing Systems – TPS)

Πρόκειται για συστήματα τα οποία εξυπηρετούν το λειτουργικό οργανωτικό επίπεδο της επιχείρησης. Υποστηρίζουν τις βασικές καθημερινές τυποποιημένες και προαποφασισμένες **λειτουργίες της επιχείρησης** και συλλέγουν – καταγράφουν τα δεδομένα τα οποία προέρχονται από αυτές (π.χ. παραγωγή, λογιστήριο, προσωπικό κ.λπ.). Η λήψη απόφασης περιορίζεται από στενά πλαίσια τα οποία έχουν προκαθοριστεί από υψηλότερο οργανωτικό επίπεδο. Ακόμη αποτελούν το μεγαλύτερο τμήμα του συστήματος και η λειτουργία τους είναι κρίσιμη για τον οργανισμό. Αυτή η κατηγορία συστημάτων εξυπηρετεί κυρίως :

- Ø Πωλήσεις,
- Ø Προμήθειες,
- Ø Μισθοδοσία Προσωπικού,
- Ø Πληρωμές, κ.λπ.

B) Γνωστικά Συστήματα Εργασίας (Knowledge Work Systems – KWS)

Τα συστήματα αυτά απευθύνονται στο γνωστικό οργανωτικό επίπεδο και εξυπηρετούν εκείνη την κατηγορία του εξειδικευμένου προσωπικού του οργανισμού (π.χ. μηχανικοί, γιατροί, δικηγόροι) η οποία είναι επιφορτισμένη με την παραγωγή νέων πληροφοριών και νέας γνώσης καθώς και την ενσωμάτωσή τους στον οργανισμό. Τα **γνωστικά συστήματα εργασίας** είναι πληροφοριακά συστήματα τα οποία χρησιμοποιούν την Πληροφοριακή Τεχνολογία προκειμένου να διευκολύνουν τη διανοητική εργασία κάποιων ομάδων ατόμων τα οποία ασχολούνται επί το πλείστον με την έρευνα. Σε επίπεδο οργανισμού ο ρόλος αυτών των ομάδων ατόμων συνίσταται:

- Ø Στη μετάφραση των μεταβολών του περιβάλλοντος του οργανισμού σε επιπτώσεις για τον ίδιο τον οργανισμό.
- Ø Στη διαρκή παροχή γνώσης και συμβουλών προς τη διοίκηση του οργανισμού.
- Ø Στη προετοιμασία οργανωτικών μεταβολών του οργανισμού στη βάση των μεταβολών του περιβάλλοντος του οργανισμού.

Αυτά τα πληροφοριακά_συστήματα παρουσιάζουν τα ακόλουθα χαρακτηριστικά:

- Ø Απαιτούν ειδικά δεδομένα, τα οποία πιθανά δεν αφορούν το υπόλοιπο προσωπικό του πληροφοριακού συστήματος και τα οποία συλλέγονται από το περιβάλλον του οργανισμού.
- Ø Απαιτούν ειδικό Λογισμικό το οποίο να διαθέτει ισχυρές δυνατότητες γραφικών, εικόνας, διαχείρισης εγγράφων, επικοινωνιών κ.λπ.
- Ø Απαιτούν ισχυρή υπολογιστική ικανότητα από τον Η/Υ.
- Ø Απαιτούν ιδιαίτερα φιλικά ενδιάμεσα (interfaces) για τον χρήστη.
- Ø Απαιτούν την ύπαρξη ισχυρών τερματικών σταθμών εργασίας.

Τα γνωστικά συστήματα εργασίας περιλαμβάνουν και τα εξής:

1. **Συστήματα Σχεδιασμού (Computer–Aided Design–CAD):** Πρόκειται για συστήματα τα οποία επιτρέπουν τη δημιουργία και αναθεώρηση σύνθετων και πολύπλοκων σχεδίων (τεχνικό, αρχιτεκτονικό, βιομηχανικό κ.λπ.). Παρέχουν τη δυνατότητα σχεδιασμού εγκαταστάσεων (π.χ. ηλεκτρονικές, ηλεκτρικές), κτιρίων, προϊόντων κ.λπ.
2. **Συστήματα Οικονομικών Συναλλαγών:** Επιτρέπουν την πρόβλεψη της εξέλιξης βασικών οικονομικών μεγεθών με βάση την ανάλυση της εξέλιξής τους κατά το παρελθόν (π.χ. χρηματιστηριακοί δείκτες).
3. **Εικονική Πραγματικότητα (Virtual Reality):** Στα συστήματα Εικονικής Πραγματικότητας, ήχος, εικόνα, κείμενα, κίνηση, video κ.λπ. δημιουργούν την αίσθηση ενός αληθινού περιβάλλοντος όπου ο χρήστης έχει την εντύπωση άμεσης συμμετοχής στα δρώμενα δημιουργώντας τις συνθήκες περιβάλλοντος τις οποίες επιθυμεί. Παρέχει δυνατότητες προσομοίωσης (simulation) οι οποίες επιτρέπουν την παραγωγή κινούμενων εικόνων με τη βοήθεια των οποίων ο χρήστης βλέπει το μέλλον ή το παρελθόν. Έτσι, για παράδειγμα, το μοντέλο ενός νέου προϊόντος μπορεί να αξιολογηθεί αισθητικά, τεχνικά κ.λπ. πριν την παραγωγή του, με μόνη βάση το σχέδιό του, το οποίο μπορεί να παρουσιαστεί κάτω από τις επιθυμητές συνθήκες για τον χρήστη <<πραγματικές>> συνθήκες (διάφορες οπτικές γωνίες, διαφορετικά χρώματα, πως αντιδρά σε διαφορετικές συνθήκες <<πίεσης>>, λειτουργικότητα κ.λπ.) ή ο χρήστης να <<βλέπει>> την εικόνα των ηπείρων της Γης όπως ήταν πριν μερικά εκατομμύρια χρόνια! Άλλο γνωστό παράδειγμα χρήσης της Εικονικής Πραγματικότητας αποτελεί ο Προσομοιωτής Πτήσης ο οποίος χρησιμοποιείται για την εκπαίδευση πιλότων, δημιουργώντας πραγματικές συνθήκες πτήσης σε ειδικό εργαστήριο.

Γ) Συστήματα Αυτοματισμού Γραφείου (Office Automation Systems – OAS)

Τα συστήματα αυτά, όπως και τα προηγούμενα, απευθύνονται στο γνωστικό οργανωτικό επίπεδο και εξυπηρετούν τους χρήστες των δεδομένων, οι οποίοι δεν διαθέτουν ιδιαίτερες επιστημονικές γνώσεις. Στην πράξη δεν παράγουν νέες πληροφορίες και νέα γνώση. Επιπλέον επικοινωνούν με πελάτες και προμηθευτές ή με άλλους οργανισμούς και χρησιμοποιούν εργαλεία ροής πληροφοριών (π.χ. κειμενογράφοι, συστήματα εκδόσεων εντύπων κ.λπ.). Τα συστήματα αυτά καλύπτουν τις απαιτήσεις ενός σύγχρονου γραφείου (ή ενός οργανισμού παροχής υπηρεσιών). Πρόκειται δηλαδή για κάθε εφαρμογή της **Πληροφοριακής Τεχνολογίας** με στόχο την αύξηση της παραγωγικότητας της εργασίας των χρηστών των πληροφοριών στο περιβάλλον γραφείου.

Ø Παραγωγή Εγγράφων

Με την λέξη έγγραφο εννοούμε κάθε κείμενο, σχέδιο ή εικόνα. Η παραγωγή εγγράφων βασίζεται στη χρήση:

1. ***Επεξεργαστή κειμένου:*** επιτρέπει την παραγωγή κειμένων με την επιθυμητή διαμόρφωση, όπως σελιδοποίηση, εισαγωγή εικόνων και σχημάτων, την επιλογή της γραμματοσειράς κ.λπ.

2. **Desktop Publisher:** παρέχει τη δυνατότητα για την δημιουργία μιας πλήρους εκδοτικής εργασίας, όπως δημιουργία σχεδίων, μακετών, εκδόσεις βιβλίων και περιοδικών κ.λπ.

Ø Ηλεκτρονική Διαχείριση Εγγράφων

Η ηλεκτρονική διαχείριση εγγράφων παρέχει τη δυνατότητα στους χρήστες να αποθηκεύουν, να διαχειρίζονται και να διακινούν έγγραφα και πληροφορίες πολλαπλών μορφών με καθαρά ψηφιακό τρόπο. Πρόκειται για συστήματα τα οποία χρησιμοποιούνται για την ευέλικτη διοίκηση και περιορισμό της γραφειοκρατίας. Ένα τέτοιο σύστημα αποτελείται από ένα εξειδικευμένο Λογισμικό και Υλικό με τα εξής βασικά στοιχεία:

1. Ένα Server για την εγκατάσταση και διαχείριση των βάσεων δεδομένων οι οποίες συντηρούν τα δεδομένα αρχειοθέτησης των εγγράφων.
2. Ένα Server ο οποίος διαθέτει μονάδες οπτικών δίσκων (single ή jukebox) και ο οποίος αναλαμβάνει τη διαχείριση των αποθηκευμένων στους οπτικούς δίσκους αρχείων.
3. Συστήματα οπτικής αναγνώρισης (scanners) για την εισαγωγή εγγράφων τα οποία βρίσκονται σε χαρτί, μικροαφίσες ή μικροφίλμ.
4. Σταθμούς εργασίας για την αρχειοθέτηση των εγγράφων ώστε να είναι δυνατή η ανάκλησή τους με βάση κάποια κριτήρια.
5. Σταθμούς εργασίας για τον εντοπισμό και την ανάκληση εγγράφων.
6. Σταθμούς διαχείρισης εισερχομένων και εξερχομένων fax.
7. Σταθμούς για την αναπαραγωγή των αποθηκευμένων εγγράφων.
8. Δίκτυο για την διασύνδεση των παραπάνω περιφερειακών.

Ø Συστήματα διαχείρισης ροής εργασιών (Workflow management systems)

Τα συστήματα αυτά παρέχουν την δυνατότητα στο χρήστη για προσδιορισμό, εκτέλεση, παρακολούθηση και συντονισμό της ροής των επιμέρους τμημάτων εργασίας και των αντίστοιχων πληροφοριών στα πλαίσια είτε μιας ομάδας εργασίας είτε ολόκληρου του οργανισμού.

Αυτά τα συστήματα αναλαμβάνουν τη μετατροπή των διαδικασιών ενός γραφείου σε **μετάδοση πληροφοριών** από έναν σταθμό εργασίας σε έναν άλλον, με παράλληλο συντονισμό και παρακολούθηση της διαδικασίας. Με τον τρόπο αυτό εξασφαλίζεται η βέβαιη μετάδοση εγγράφων και πληροφοριών στους αρμόδιους, ο αποδοτικός καταμερισμός των πόρων (χρόνου εργασίας, χώρου αποθήκευσης, εκτυπωτών, μονάδων, fax κ.λπ.) ενώ εξασφαλίζεται η παρακολούθηση του **βαθμού ολοκλήρωσης** κάθε διαδικασίας ώστε να ρυθμίζεται η έγκαιρη ολοκλήρωσή της. Με αυτά τα συστήματα μπορούν να διακινηθούν:

1. Χειρόγραφα ή εκτυπωμένα έντυπα.
2. Έγγραφα τα οποία δημιουργήθηκαν σε επεξεργαστές κειμένων.
3. Φύλλα εργασίας.
4. Φωνή – Ήχος.
5. Video.

Ø Υπερκείμενα, Υπερμέσα, Πολυμέσα (Hypertext, Hypermedia, Multimedia)

Στα **Υπερκείμενα (Hypertext)** οι πληροφορίες είναι σε μορφή κειμένου και βρίσκονται αποθηκευμένες σε ένα δίκτυο από κόμβους (nodes), οι δε κόμβοι επικοινωνούν μεταξύ τους με συνδέσμους (links). Στην ουσία κόμβος είναι ένα σύνολο δεδομένων, οργανωμένα γύρω από ένα θέμα σε μορφή κειμένου. Τα κύρια χαρακτηριστικά των Υπερκειμένων:

1. Αποτελούνται από ένα σύνολο πληροφοριών αποκλειστικά σε μορφή κειμένου, η πρόσβαση δε στην πληροφορία μπορεί να γίνει μέσα από πολλούς διαδρόμους.
2. Στους τρόπους παρουσίασης, χρησιμοποιείται η μέθοδος των πολλαπλών παραθύρων, κατά την οποία σε διαφορετικά παράθυρα στην οθόνη, απεικονίζονται ταυτόχρονα πολλά διαφορετικά κείμενα. Επιτρέπουν ένα σημαντικό βαθμό ελευθερίας σε επίπεδο διαλογικότητας. Παρέχουν τη δυνατότητα στο χρήστη να κάνει τις επιλογές του σχετικά με την διαδρομή που θα ακολουθήσει για να φτάσει στην επιζητούμενη πληροφορία.

Στα **Υπερμέσα (Hypermedia)** τα στοιχεία της πληροφορίας, τα οποία ορίζονται σαν κόμβοι, δεν περιέχουν μόνο δεδομένα σε μορφή κειμένου, αλλά οποιασδήποτε μορφής δεδομένα, όπως ήχος, γραφικά, εικόνες, προσομοίωση κίνησης (animation), video κ.λπ.

Στα **Πολυμέσα ή Πολλαπλά Μέσα (Multimedia)** γίνεται συλλογή και χρησιμοποίηση ενός συνδυασμού δεδομένων κειμένου, γραφικών, ήχου, animation, video ή κάποιου από αυτά. Ο χρήστης της εφαρμογής Πολυμέσων δεν μπορεί να παρέμβει κατά την διάρκεια του << τρεξίματος>> της εφαρμογής, δεν έχει κανενός είδους έλεγχο πάνω στη ροή της εφαρμογής, δεν μπορεί να επιλέξει τί και πότε θα δει (ή θα ακούσει) και δεν μπορεί παρά να μείνει απλός θεατής των όσων εξελίσσονται μέσα στον Η/Υ.

Στα **Διαλογικά Πολυμέσα ή Διαλογικά Πολλαπλά Μέσα (Interactive Multimedia)** αυτή η έλλειψη ελέγχου παύει να ισχύει και ο χρήστης μπορεί να παρεμβαίνει και να κατευθύνει την πορεία διαλόγου. Την εφαρμογή των Πολυμέσων σε ευρεία κλίμακα ενισχύει η χρήση των οπτικών δίσκων (CD – ROM) οι οποίοι διαθέτουν μεγάλη χωρητικότητα αποθήκευσης δεδομένων με πολύ μικρό κόστος. Έτσι αποφεύγεται η απαίτηση ύπαρξης μεγάλων και κοστοβόρων βάσεων δεδομένων, με συνέπεια τα Πολυμέσα να μπορούν να χρησιμοποιούνται ακόμη και σε περιβάλλον γραφείου.

Ø Διαχείριση Δεδομένων

Πρόκειται για την διαχείριση δεδομένων τα οποία ανήκουν σε ανεξάρτητα πληροφοριακά συστήματα, περιορισμένου και προσωπικού χαρακτήρα (π.χ. προσωπικά δεδομένα ενός διοικητικού στελέχους τα οποία τον διευκολύνουν στο έργο του και δεν ανήκουν στο κεντρικό σύστημα βάσης δεδομένων του πληροφοριακού συστήματος).

Φύλλα Δεδομένων: Εμφανίζουν στο χρήστη ένα φύλλο εργασίας αποτελούμενο από κελιά (πινακοειδής μορφή) στα οποία μπορούν να εισαχθούν δεδομένα, να συνδεθούν μεταξύ τους με την βοήθεια αλγορίθμων, να διαταχθούν κ.λπ. χρησιμοποιώντας το πληκτρολόγιο ή το ποντίκι (π.χ. LOTUS, EXCEL).

Προσωπικές Βάσεις Δεδομένων (Desktop Databases): Πρόκειται για μικρά συστήματα βάσης δεδομένων τα οποία λειτουργούν σε περιβάλλον προσωπικού υπολογιστή.

Ανάκτηση Πληροφοριών: Ανάκτηση πληροφοριών είναι μία τεχνική διαδικασία που ακολουθείται για να εντοπιστούν και παραληφθούν οι πληροφορίες τις οποίες κάποιος χρειάζεται στην κατάλληλη μορφή, ποιότητα και ποσότητα, από τις ήδη υπάρχουσες στο πληροφοριακό σύστημα. Αυτή η μεθοδολογία εφαρμόζεται συνήθως όταν ανάμεσα στα διάφορα κείμενα απαιτείται η επιλογή εκείνων των τμημάτων τους, τα οποία αναφέρονται σε κάποια λέξη «κλειδί».

Ø Διαχείριση Έργων

Πρόκειται για πληροφοριακό σύστημα προσωπικής χρήσης το οποίο επιτρέπει το προγραμματισμό – σχεδιασμό και τον έλεγχο κάποιου έργου. Σε αυτή την κατηγορία ανήκουν τα ακόλουθα:

1. **Εργαλεία Χρονικού Προγραμματισμού:** Πρόκειται για το λογισμικό το οποίο επιτρέπει την εφαρμογή μεθόδων της επιχειρησιακής έρευνας (π.χ. PERT, COST κ.λπ.).
2. **Εργαλεία Διαχείρισης Έργου:** Πρόκειται για το λογισμικό το οποίο επιτρέπει τον προγραμματισμό – σχεδιασμό και παρακολούθηση της εξέλιξης κάποιου έργου (π.χ. MS PROJECT).

Δ) Πληροφοριακά Συστήματα Διοίκησης (Management Information Systems – MIS)

Εξυπηρετούν το **διοικητικό οργανωτικό επίπεδο** εφοδιάζοντας τα μεσαία διοικητικά στελέχη με κατηγοριοποιημένες πληροφορίες, υπό μορφή αναφορών, οι οποίες προέρχονται, είτε από τα προηγουμένως αναφερθέντα συστήματα, είτε από αρχεία περασμένων χρήσεων. Αυτές οι αναφορές αποτελούν απαντήσεις σε προκαθορισμένα ερωτήματα γενικού στατιστικού χαρακτήρα και εκδίδονται σε τακτά χρονικά διαστήματα.

Ε) Συστήματα Υποστήριξης Αποφάσεων (Decision – Support Systems – DSS)

Και αυτός ο τύπος των συστημάτων εξυπηρετεί το **διοικητικό οργανωτικό επίπεδο** του οργανισμού. Στόχος της ύπαρξης και λειτουργίας τους είναι η υποστήριξη της λήψης αποφάσεων από τα μεσαία διοικητικά στελέχη. Αναφέρονται σε ημιδομημένες, μοναδικές ή ταχέως μεταβαλλόμενες αποφάσεις (π.χ. η εκτίμηση του κόστους ενός έργου ανατρέχοντας στις αναλυτικές τιμές κόστους των συνιστωσών του και με πρόβλεψη της εξέλιξης τους στο άμεσο μέλλον). Τα συστήματα αυτά τροφοδοτούνται

κυρίως από τις εξόδους των TPS και MIS αλλά και από εξωτερικά δεδομένα του οργανισμού. Αποτελούν ένα συνδυασμό των δυνατοτήτων του ανθρώπου και του Η/Υ στα πλαίσια του οποίου, ο λαμβάνων κάποια απόφαση χρησιμοποιεί μορφοποιημένες ή αμορφοποίητες μεθόδους για να διερευνήσει κάποιο πρόβλημα χαμηλού βαθμού δόμησης και ενισχύοντας τη συλλογική του καταλήγει στη λήψη αποφάσεων. Κατά συνέπεια τα DSS είναι άρρηκτα δεμένα με τη διαδικασία λήψης απόφασης. Αυτά τα πληροφοριακά συστήματα χρησιμοποιούνται ευρέως στις μεγάλες επιχειρήσεις (π.χ. America Airline, αεροπορική εταιρία για το συσχετισμό δρομολογίου και τιμής).

Ø Συνιστώσες Των DSS:

Βάση Δεδομένων του DSS: Αποτελεί συλλογή από τρέχοντα ή ιστορικά δεδομένα τα οποία προέρχονται από άλλα πληροφοριακά συστήματα. Το DSS δεν δημιουργεί αυτά τα δεδομένα αλλά τα διαφυλάσσει κατά την επεξεργασία τους.

Βάση Των Μοντέλων: Αποτελεί συλλογή από μαθηματικά και αναλυτικά μοντέλα τα οποία περιγράφουν τη σχέση μεταξύ των παραγόντων οι οποίοι υπεισέρχονται στη μελέτη κάποιου προβλήματος και τα οποία μπορούν να χρησιμοποιηθούν από τον λαμβάνοντα την απόφαση.

Λογισμικό του DSS: Επιτρέπει την χρήση του DSS από τον χρήστη. Η λειτουργία στρέφεται προς:

1. Τη διαχείριση της βάσης δεδομένων.
2. Τη διαχείριση της βάσης των μοντέλων.
3. Τη διαχείριση του ενδιαμέσου του χρήστη.

ΣΤ) Συστήματα Υποστήριξης Της Εκτελεστικής Εξουσίας (Executive Support Systems – ESS)

Αυτά τα συστήματα εξυπηρετούν το **στρατηγικό οργανωτικό επίπεδο** και επιτρέπουν στα επιτελικά διοικητικά στελέχη να λαμβάνουν αποφάσεις. Σαν είσοδο χρησιμοποιούν δεδομένα από το εσωτερικό και εξωτερικό περιβάλλον του οργανισμού καθώς και τις εξόδους των MIS και DSS. Ακόμη αφορούν αδόμητες αποφάσεις γενικού χαρακτήρα. Στοχεύουν όπως και τα DSS στο να παρέχουν βοήθεια στη λήψη αποφάσεων. Παρουσιάζουν ομοιότητες με τα DSS τόσο στη δομή όσο και στις συνιστώσες τους. Οι βασικές του διαφορές από τα DSS είναι:

- Ø Χρησιμοποιούνται από υψηλόβαθμα στελέχη (senior managers) για την λήψη αποφάσεων οι οποίες αφορούν γενικά προβλήματα στρατηγικής και τακτικής.
- Ø Στη βάση δεδομένων τους περιλαμβάνονται δεδομένα τα οποία προέρχονται και από το εσωτερικό (άλλα συστήματα) αλλά και από το εξωτερικό του οργανισμού.
- Ø Αυτά τα συστήματα παρέχουν την πληροφόρησή τους στον χρήστη συνήθως με τη βοήθεια γραφικών και έχουν τη δυνατότητα της ανάλυσης αυτής της πληροφόρησης σε αναλυτικότερο επίπεδο (drill down) με τη βοήθεια εργαλείων ανάλυσης. Η ανάπτυξη αυτών των συστημάτων παρουσιάζει τα ακόλουθα προβλήματα:

1. Δυσκολία στην **αναλυτική περιγραφή** των απαιτήσεων των χρηστών, δεδομένου ότι οι απαιτήσεις ποικίλουν από χρήστη σε χρήστη και από οργανισμό σε οργανισμό και ότι πρόκειται για μη τυποποιημένες απαιτήσεις.
2. Δυσκολία στη **συλλογή** όλων εκείνων των δεδομένων τα οποία τροφοδοτούν τη βάση τους. Πολλά δεδομένα προέρχονται από το εξωτερικό του οργανισμού αλλά και για τα προερχόμενα από το εσωτερικό πιθανά θα πρέπει να μεταβληθούν τα υπάρχοντα συστήματα τα οποία το παράγουν.
3. Δημιουργία αντιθέσεων στην **ανάπτυξη** τους, προερχόμενες από διοικητικά στελέχη χαμηλότερου βαθμού (απειλή για την εξέλιξη τους, έλεγχο από τους προϊσταμένους τους).
4. Δυσκολία στην **αξιολόγηση** της σχέσης κόστους – ωφέλειας, δεδομένου ότι υποστηρίζουν την αδόμητη εργασία των διοικητικών στελεχών.

1.6 Υποσυστήματα Πληροφοριακού Συστήματος

Το Πληροφοριακό Σύστημα δεν είναι μονολιθικό. Αποτελείται από διάφορα συστατικά μέρη τα οποία αποτελούν τα **υποσυστήματα** του. Τα υποσυστήματα αυτά θα καλούνται εφεξής Πληροφοριακά Συστήματα (Information Systems) και το καθένα από αυτά μπορούμε να το εξετάσουμε ως ένα ξεχωριστό και ανεξάρτητο σύστημα. Κάθε υποσύστημα, όπως ήδη αναφέραμε διαιρείται σε άλλα μικρότερα, και έτσι αναπτύσσεται μία ιεραρχική δομή, όπου σε κάθε επίπεδο ανήκουν υποσυστήματα, τα οποία είναι ανεξάρτητα και λειτουργικά και βρίσκονται σε στενή σχέση και συνεργασία με τα αντίστοιχα υποσυστήματα του παραπάνω και παρακάτω επιπέδου.

Η έκταση και η ποιότητα των **Πληροφοριακών Συστημάτων** εξαρτάται από τους στόχους και την ποιότητα της διοίκησης καθώς και από την χρησιμοποιούμενη Πληροφοριακή Τεχνολογία. Επιπλέον τα συστήματα αυτά χρησιμοποιούνται από όλα τα επίπεδα της διοικητικής πυραμίδας και εξυπηρετούν δυο βασικές επιδιώξεις:

- ∅ Τον καταμερισμό της ευθύνης.
- ∅ Την υλοποίηση του ελέγχου.

Τα πληροφοριακά συστήματα εκτός από τους υπολογιστές περιλαμβάνουν τους **ανθρώπους** που συλλέγουν και χρησιμοποιούν τις πληροφορίες, τις διαδικασίες που χρησιμοποιούνται για την καταγραφή, την οργάνωση και την χρήση των πληροφοριών, τα μέσα στα οποία καταχωρούνται οι πληροφορίες κ.λπ.

1.7 Στάδια Εξέλιξης Πληροφοριακού Συστήματος Ενός Οργανισμού

Η ποιότητα του πληροφοριακού συστήματος είναι συνάρτηση **απαιτήσεων** της διοίκησης από την υπάρχουσα οργάνωση του εκάστοτε οργανισμού και της συμβολής του στις δραστηριότητες του. Οι απαιτήσεις αυτές οφείλουν να λαμβάνουν υπόψη το εξελικτικό στάδιο στο οποίο βρίσκεται αυτή η οργάνωση.

Η εξέλιξη ενός πληροφοριακού συστήματος διέρχεται από τρία στάδια τα οποία είναι η **Μύηση**, η **Επέκταση** και η **Ωριμότητα**, η οποία αποτελείται από δύο φάσεις: **1)** τον Έλεγχο και **2)** την ολοκλήρωση.

Τα κριτήρια για την ένταξη της εξέλιξης του πληροφοριακού συστήματος σε ένα από αυτά τα στάδια είναι:

- Ø Η ποιότητα οργάνωσης του πληροφοριακού συστήματος.
- Ø Ο βαθμός συμμετοχής του προσωπικού του οργανισμού στην ανάπτυξη του πληροφοριακού συστήματος.
- Ø Η ποιότητα ελέγχου του πληροφοριακού συστήματος.
- Ø Η πορεία ανάπτυξης των εφαρμογών στον οργανισμό.

Ο εντοπισμός αυτού του **εξελικτικού σταδίου**, επιτρέπει στη διοίκηση να καθορίζει στόχους γνωρίζοντας τις αντιδράσεις, οι οποίοι ασκούνται στον ανθρώπινο παράγοντα και στις διοικητικές λειτουργίες και να καθορίζει αντίστοιχα τις απαιτήσεις του ελέγχου. Έτσι πιο αναλυτικά μπορούμε να επισημάνουμε τα εξής:

Μύηση: Σ' αυτό το στάδιο γίνονται τα πρώτα βήματα της χρήσης της σύγχρονης Πληροφοριακής τεχνολογίας στα πληροφοριακά συστήματα. Έμφαση δίνεται στην ικανοποίηση των αναγκών του λειτουργικού υποσυστήματος του οργανισμού, δηλαδή δημιουργούνται τα πρώτα TPS (πρώτες εφαρμογές Λογιστηρίου, Μισθοδοσία κ.λπ.)

Επέκταση: Σ' αυτό το στάδιο έχει γίνει η αποδοχή της χρήσης της σύγχρονης Πληροφοριακής Τεχνολογίας και γίνονται προσπάθειες για την επέκταση της χρήσης της και σε άλλους τομείς του λειτουργικού υποσυστήματος του οργανισμού (επέκταση των TPS).

Ωριμότητα: Σ' αυτό το στάδιο γίνεται προσπάθεια για την ανάπτυξη MIS, OAS, KWS και DSS δεδομένου ότι έχουν ικανοποιηθεί σε μεγάλο βαθμό οι ανάγκες του λειτουργικού υποσυστήματος του οργανισμού και προβάλλει επιτακτικά η ανάγκη υλοποίησης των αναγκών της διοίκησης. Στον οργανισμό, κάτω από το βάρος των νέων απαιτήσεων, συγκεντρώνεται εξειδικευμένο προσωπικό (Αναλυτές, Προγραμματιστές, Μηχανικοί κ.λπ.) και δημιουργείται ιδιαίτερο τμήμα υπεύθυνο για τη διοίκηση του πληροφοριακού συστήματος, το Κέντρο Πληροφορικής (EDP Center) ή Κέντρο Πληροφοριακής Τεχνολογίας (Information Technology – IT Center). Πρόκειται για τις σύγχρονες ονομασίες εκείνου το οποίο παλαιότερα ονομαζόταν Κέντρο Μηχανογράφησης. Ο **έλεγχος** αποτελεί τη πρώτη φάση του σταδίου ωριμότητας. Σ' αυτή τη φάση το ενδιαφέρον της διοίκησης του οργανισμού στρέφεται στη δημιουργία δομών ελέγχου οι οποίες αφορούν το πληροφοριακό σύστημα. Η δε **ολοκλήρωση** αποτελεί τη δεύτερη φάση του σταδίου ωριμότητας κατά το οποίο το

πληροφοριακό σύστημα καλύπτει ολόκληρο τον οργανισμό. Γίνονται προσπάθειες για την έκτακτη και την εντατική του ανάπτυξη. Εδώ εμφανίζονται και τα ESS.

1.8 Η Έννοια Της Ολοκλήρωσης – Ολοκληρωμένα Συστήματα

Η μετατροπή των ανεξάρτητων συστημάτων σε ολοκληρωμένα συστήματα ονομάζεται **Ολοκλήρωση** (Integration). Το αρχικό ερώτημα το οποίο τίθεται είναι αν η ολοκλήρωση αποτελεί αντικειμενική επιδίωξη κάθε οργανισμού και πόσο ολοκληρωμένο θα πρέπει να είναι το πληροφοριακό σύστημα.

Σε γενικές γραμμές η σχέση κόστους/ωφέλειας γέρνει προς την πλευρά των ολοκληρωμένων πληροφοριακών συστημάτων, τουλάχιστον για τους μεσαίους και μεγάλους οργανισμούς. Η χρήση **ανεξάρτητων πληροφοριακών συστημάτων** (κάθε σύστημα σχεδιάζεται για να ικανοποιήσει μια κατηγορία διοικητικών απαιτήσεων) ενδείκνυται είτε στη περίπτωση μικρών οργανισμών, είτε στη περίπτωση κατά την οποία κάποιες εργασίες του οργανισμού έχουν συμπληρωματικό ή βοηθητικό χαρακτήρα για τους κλάδους λειτουργίας, οπότε υποστηρίζουν έμμεσα τα ολοκληρωμένα συστήματα. Το δεύτερο ερώτημα αφορά τη διαδικασία της ολοκλήρωσης. Η ζωή του πληροφοριακού συστήματος ξεκινά συνήθως με τη «**συρραφή**» υπαρχόντων ανεξάρτητων πληροφοριακών συστημάτων. Η ολοκλήρωσή του μπορεί να γίνεται με τους εξής τρόπους:

- ∅ Τα ανεξάρτητα πληροφοριακά συστήματα ολοκληρώνονται στη βάση κάποιου συνολικού διοικητικού σχεδίου.
- ∅ Γίνεται ριζική αναδιάρθρωση του υπάρχοντος πληροφοριακού συστήματος και το νέο προκύπτει με διαδικασία top-down.

Η ολοκλήρωση του πληροφοριακού συστήματος, με όποιο τρόπο και αν προήλθε, γίνεται με την ποσοτική και την εντατική (ποιοτική) του βελτίωση. Αυτές οι βελτιώσεις επιφέρονται στις βασικές του λειτουργίες και είναι η ακόλουθες:

Ολοκλήρωση της εισόδου: Αναφέρεται στις βελτιώσεις συλλογής και εισαγωγής των δεδομένων στα συστήματα.

Ολοκλήρωση οργάνωσης: Αναφέρεται στις βελτιώσεις στον τομέα οργάνωσης και αποθήκευσης των δεδομένων στους φορείς τους. Βέλτιστη μορφή ολοκλήρωσης είναι η οργάνωση σε βάση δεδομένων.

Ολοκλήρωση διεργασιών: Αναφέρεται στις βελτιώσεις (ποιοτικές-ποσοτικές) των υφιστάμενων διεργασιών των δεδομένων.

Ολοκλήρωση της ροής: Αναφέρεται στις βελτιώσεις οι οποίες αφορούν τη συνεχή ροή των δεδομένων από σύστημα σε σύστημα.

Ολοκλήρωση εξόδου: Αναφέρεται στις βελτιώσεις οι οποίες αφορούν την παρουσίαση των επεξεργασμένων πληροφοριών στο χρήστη. Σε σχέση με την ολοκλήρωση των συστημάτων θα πρέπει να σημειωθούν τα ακόλουθα:

- ∅ Η ολοκλήρωση μπορεί να γίνεται παράλληλα σε πολλά επίπεδα.
- ∅ Η εντατική τους ολοκλήρωση αναφέρεται, είτε στην αύξηση της ποσότητας δεδομένων και διεργασιών, είτε στην κάλυψη νέων διοικητικών απαιτήσεων.
- ∅ Η εντατική τους ολοκλήρωση αναφέρεται στην ποιοτική βελτίωσή τους, είτε με πιο ορθολογική χρήση των υπαρχουσών διεργασιών, είτε με τη χρήση νεώτερης και πιο αποδοτικής Πληροφοριακής Τεχνολογίας, είτε με συνδυασμό και των δυο.

1.9 Κύκλος Ζωής Πληροφοριακού συστήματος

Ένα πληροφοριακό σύστημα (Π.Σ) αποτελεί κάτι μη απτό. Είναι ένα ιδεατό κατασκεύασμα το οποίο δημιουργείται για να αντιπροσωπεύσει μία φυσική οντότητα, η οποία υπάρχει μέσα σε μία επιχείρηση ή σε έναν οργανισμό.

Ένα πληροφοριακό σύστημα ακολουθεί σχεδόν πάντα μία συγκεκριμένη πορεία: δημιουργείται, αναπτύσσεται, εξελίσσεται, και τελικά, αποσύρεται. Η ύπαρξη του οριοθετείται τη χρονική στιγμή που η επιχείρηση ή ο οργανισμός, παίρνει την απόφαση για τη δημιουργία του. Στη συνέχεια ακολουθεί μία περίοδος κατά την οποία προσδιορίζονται οι βασικές απαιτήσεις των λειτουργιών του και σχεδιάζονται οι λειτουργίες που ικανοποιούν τις απαιτήσεις αυτές. Από εκεί και πέρα ξεκινάει μία μεγάλη χρονική περίοδος κατά την οποία πραγματοποιείται η ανάπτυξη του και η διαρκής εξέλιξη του ώστε να είναι σε θέση να ικανοποιεί διαρκώς τις ανάγκες της επιχείρησης ή του οργανισμού, στον οποίο ανήκει.

Φυσικά, κάποια στιγμή έρχεται η ώρα που πρέπει να αποσυρθεί όταν η επιχείρηση ή ο οργανισμός αποφασίσει ότι είναι πια ξεπερασμένο, αναποτελεσματικό και μη αποδοτικό.

Η αποδοτικότητα και αποτελεσματικότητά του, εξαρτάται από πολλούς ενδογενείς ή και εξωγενείς παράγοντες (όπως για παράδειγμα η τεχνολογία υλικού και λογισμικού που χρησιμοποιείται). Η πορεία ενός Π.Σ, από τη στιγμή του καθορισμού του προβλήματος ή των προβλημάτων που καλείται να επιλύσει, μέχρι τη λειτουργία του, τη συντήρησή του και, τέλος, την απόσυρσή του, είναι γνωστός στη βιβλιογραφία ως **Κύκλος Ζωής** του πληροφοριακού συστήματος.

Η έννοια Π.Σ. δεν είχε πάντοτε την ερμηνεία και την υπόσταση που της έχουν δώσει σήμερα οι ερευνητές. Από τη δεκαετία του 90 μέχρι και σήμερα έχει επικρατήσει η άποψη ότι ένα Π.Σ. αποτελείται από τις εξής τρεις αλληλοεξαρτώμενες παραμέτρους:

- ∅ τεχνική (υλικό + λογισμικό)
- ∅ οργανωτική (διαδικασίες)
- ∅ κοινωνική (ανθρώπινες σχέσεις και συναισθήματα).

Γνωρίζουμε επίσης ότι σε μία επιχείρηση ή οργανισμό ένα Π.Σ., αποτελεί το συστατικό που συνδέει το φυσικό σύστημα παραγωγής με το σύστημα λήψης

αποφάσεων. Εκτός από αυτές τις βασικές λειτουργίες ένα ολοκληρωμένο Π.Σ. πρέπει να προσφέρει και επιπλέον δυνατότητες για:

- Ø συνεχή εξέλιξη για την ικανοποίηση νέων ή αυξανόμενων αναγκών.
- Ø βοήθεια στις διαδικασίες ελέγχου και διοίκησης της επιχείρησης ή του οργανισμού.
- Ø βοήθεια στον προγραμματισμό και τη δημιουργία της στρατηγικής ανάπτυξης της επιχείρησης ή του οργανισμού.
- Ø συνεισφορά στη δημιουργία αλλαγών ώστε η επιχείρηση ή ο οργανισμός να είναι σε θέση να προσαρμόζεται συνεχώς στο περιβάλλον του ένα ολοκληρωμένο Π.Σ. πρέπει να είναι ευέλικτο και προσαρμόσιμο ώστε να ανταποκρίνεται στις αλλαγές και τις διαφορετικές απαιτήσεις διαφόρων ομάδων χρηστών.
- Ø εκπαίδευση και μάθηση.

Από όλα αυτά μπορεί να συμπεράνει κανείς ότι κάθε προσπάθεια ανάπτυξης ενός ολοκληρωμένου Π.Σ. είναι ένα συνολικό αποτέλεσμα αποτελεσματικού συνδυασμού των διαφόρων τεχνικών/οργανωτικών και ανθρώπινων στοιχείων. Οι περισσότεροι ερευνητές αποδέχονται ότι ο **παραδοσιακός κύκλος ζωής** ενός Π.Σ. αποτελείται από τις ακόλουθες επτά φάσεις:

1. Διερευνητική μελέτη (ή καθορισμός του προβλήματος).
2. Μελέτη σκοπιμότητας.
3. Ανάλυση απαιτήσεων.
4. Σχεδιασμός του συστήματος.
5. Υλοποίηση – κωδικοποίηση.
6. Εγκατάσταση.
7. Λειτουργία – συντήρηση.

Θεωρητικά, οι φάσεις αυτές πρέπει να βρίσκονται διατεταγμένες **σειριακά**, δηλαδή η ολοκλήρωση κάθε φάσης να οδηγεί πάντα στην αμέσως επόμενη της. Αυτό, φυσικά, προϋποθέτει ότι τα αποτελέσματα κάθε φάσης θα οδηγούν κατά τρόπο αναμφισβήτητο στην επόμενη, δηλαδή το προϊόν που παράγει κάθε φάση θα γίνεται δεκτό όπως είναι χωρίς να υπάρχει η περίπτωση αλλαγής του αργότερα. Μόνο σε αυτή την περίπτωση είναι δυνατό να παγιωθεί η σειρά εκτέλεσης των διαφόρων φάσεων. Συνήθως οι χρήστες για διάφορους λόγους **αλλάζουν συνέχεια τις απαιτήσεις τους**, με αποτέλεσμα οι αναλυτές του Π.Σ. να οδηγούνται σε μία συνεχή επανεξέταση των προηγούμενων φάσεων. Κάθε φάση του παραδοσιακού κύκλου ζωής παράγει κάποια καθορισμένα προϊόντα και έχει ως στόχο να απαντά σε κάποιες συγκεκριμένες ερωτήσεις καθοριστικής σημασίας για την ανάπτυξη του συστήματος.

Οι απαιτήσεις μετασχηματίζονται σε βασικές λειτουργικές απαιτήσεις, δηλαδή σε λειτουργίες του συστήματος, ανεξάρτητα από την τεχνολογία που ακολουθείται για την υλοποίησή του. Στη συνέχεια, με βάση την ανάλυση των απαιτήσεων προσδιορίζονται τα χαρακτηριστικά απόδοσης, από τα οποία προκύπτουν οι **τεχνικές προδιαγραφές** του υλικού/λογισμικού. Προκύπτει επομένως μια πορεία μετάβασης από το «**τι πρέπει να γίνει**» στο «**πώς πρέπει να γίνει**». Η πορεία αυτή θεωρείται σωστή, γιατί έχει ως αφετηρία το ίδιο το πρόβλημα και όχι την τεχνολογία που θα χρησιμοποιηθεί για την επίλυσή του.

Κατά την διάρκεια του Κύκλου Ζωής, γίνονται και κάποιοι έλεγχοι που χωρίζονται σε δυο μεγάλες ομάδες:

1. τους ελέγχους που αντιστοιχούν στην **επικύρωση** και
2. τους ελέγχους που αντιπροσωπεύουν την **επαλήθευση**.

Στην επικύρωση οι έλεγχοι στοχεύουν να αποδείξουν ότι το προϊόν ανταποκρίνεται στις ανάγκες του χρήστη (απάντηση στο ερώτημα: «φτιάχνω το σωστό προϊόν;»). Από την άλλη πλευρά στην επαλήθευση ελέγχουμε αν το προϊόν φτιάχνεται σύμφωνα με τις προδιαγραφές του χρήστη (απάντηση στο ερώτημα: «φτιάχνω το προϊόν σωστά;»)

Κάποιοι από όλους αυτούς παίζουν ιδιαίτερα σημαντικό ρόλο. Είναι οι κύριοι τεχνικοί έλεγχοι, όπως είναι ευρύτερα γνωστοί. Οι έλεγχοι αυτοί αντιστοιχούν στα βασικά παραδοτέα κάθε φάσης, τα οποία αποτελούν και την αρχή της επόμενης φάσης. Μ' αυτόν τον τρόπο ο παραδοσιακός κύκλος ζωής ενός Π.Σ. γίνεται πιο λειτουργικός και κυρίως πιο αποτελεσματικός, ώστε να μην μειονεκτεί σε σχέση με τις άλλες υπάρχουσες μεθοδολογίες.

Ας δούμε τις φάσεις, κάθε μία ξεχωριστά:

1η Φάση: Διερευνητική Μελέτη

Η φάση αυτή ξεκινάει τη στιγμή που ο χρήστης θέτει το πρόβλημα και ζητάει τη βοήθεια από κάποιο ειδικό αναλυτή/σχεδιαστή Π.Σ., επειδή ο χρήστης συνήθως δεν έχει καλή γνώση ούτε του περιβάλλοντος, ούτε της τεχνολογίας που θα χρησιμοποιηθεί για την υλοποίησή του. Ο βασικός και αντικειμενικός στόχος της φάσης αυτής είναι ο **προσδιορισμός του προβλήματος**. Αρκετοί ισχυρίζονται ότι η φάση αυτή θα μπορούσε σε πολλές περιπτώσεις να παραληφθεί, όμως υπάρχουν περιπτώσεις που η παρουσία της κρίνεται απαραίτητη για την ομαλή και αποδοτική κατασκευή του Π.Σ.

Η διερευνητική μελέτη είναι ιδιαίτερα απαραίτητη για:

1. Μεσαία και μεγάλα έργα με υψηλό προϋπολογισμό και μεγάλα χρονικά περιθώρια ανάπτυξης.
2. Έργα που δεν είναι εγγυημένα η αναγκαιότητά τους.
3. Έργα που δεν είναι καλώς και με σαφήνεια ορισμένα.
4. Έργα ευαίσθητα από τεχνολογικής πλευράς.

Τέλος, η φάση αυτή μπορεί να περιλαμβάνει και τη διαδικασία **αποτίμησης** του Π.Σ., η οποία αποτελεί τη διαδικασία καθορισμού της αξίας ενός προτεινόμενου ή ενός ήδη υλοποιημένου συστήματος. Η αναγκαιότητα της αποτίμησης ενός συστήματος είναι πολύ μεγάλη και βοηθάει τον αναλυτή/σχεδιαστή στην ολοκλήρωση ενός Π.Σ. από πολλές πλευρές:

1. Περιορίζει τον επιχειρηματικό κίνδυνο.
2. Βοηθάει και στηρίζει τις χρηματοοικονομικές επενδύσεις.
3. Υποστηρίζει τις οργανωτικές αλλαγές που απαιτεί η επιτυχή ενσωμάτωση ενός Π.Σ.
4. Προσφέρει τη δυνατότητα γνώσης της κατάστασης του συστήματος και τον ορθό έλεγχό του.

5. Αξιοποιεί την εμπειρία που υπάρχει στην αντιμετώπιση των διαφόρων προβλημάτων.

2η Φάση: Μελέτη Σκοπιμότητας

Η δεύτερη φάση ξεκινάει μετά την έγκριση της διερευνητικής μελέτης και έχει ως σκοπό να αναλύσει παραπέρα τη ή/και τις λύση/εις που έχουν επιλεγεί κατά τη φάση της διερευνητικής μελέτης. Η ανάλυση γίνεται κυρίως ως προς τις εξής παραμέτρους:

1. Χρόνος (Πόσο χρόνο χρειάζεται η υλοποίηση κάθε λύσης;).
2. Κόστος (Ποια είναι η εκτίμηση για το συνολικό κόστος κάθε λύσης;).
3. Όφελος (Ποια είναι τα οφέλη που θα προσφέρει η κάθε λύση; Ποια είναι τα οφέλη από την εισαγωγή των υπολογιστών στην επιχείρηση;).
4. Τεχνολογία (Είναι η λύση εφικτή τεχνολογικά; Υπάρχουν δυσκολίες στην υλοποίησή της;).

3η Φάση: Ανάλυση Απαιτήσεων

Ο προσδιορισμός και στη συνέχεια η ανάλυση των απαιτήσεων, πρέπει να είναι ταυτόχρονα πλήρεις και ακριβείς, κάτι το οποίο είναι ένα εξαιρετικά δύσκολο έργο. Συνήθως οι αναλυτές ζητάνε μόνο από το δυνητικό χρήστη να προσδιορίσει τις απαιτήσεις του. Αυτό όμως όπως έχει δείξει η πρακτική δεν είναι αρκετό. Η ανάγκη προσδιορισμού των απαιτήσεων, μέσα στα πλαίσια του σχεδιασμού και της ανάπτυξης ενός ολοκληρωμένου Π.Σ. αναφέρεται σε δύο επίπεδα:

1. Σε ολόκληρο το κυρίως σύστημα, οπότε πρέπει να εξετασθεί η συνολική δομή του οργανισμού/επιχείρησης.
2. Στον αναλυτικό προσδιορισμό των απαιτήσεων των συγκεκριμένων εφαρμογών, που υποστηρίζει το Π.Σ., οπότε πρέπει να εξετασθούν όλες οι λειτουργίες που υλοποιούνται και όλα τα παραδοτέα που παράγονται από τα επιμέρους κομμάτια του συστήματος.

Η διαδικασία που ακολουθείται για τον προσδιορισμό των απαιτήσεων είναι η ίδια και στις δύο περιπτώσεις. Εκείνο που διαφέρει είναι το **εύρος** και το **βάθος** της ανάλυσης. Πιο ευρύτερη και γενική είναι η εξέταση που γίνεται για ολόκληρο το σύστημα από ότι η ανάλυση που γίνεται για τις επί μέρους εφαρμογές. Προφανώς αυτό το στοιχείο επηρεάζει σημαντικά την επιλογή της μεθοδολογίας και των τεχνικών που πρόκειται να χρησιμοποιηθούν σε κάθε περίπτωση.

Αξίζει να τονισθεί ότι οι προδιαγραφές που τίθενται κατά τη διάρκεια της ανάλυσης θα πρέπει να αντιστοιχούν στις πραγματικές απαιτήσεις του υπό ανάπτυξη συστήματος.

4η Φάση: Σχεδιασμός Συστήματος

Ο σχεδιασμός ενός Π.Σ. πραγματοποιείται σε δύο στάδια:

1. στο **λογικό σχεδιασμό**, που ξεκινάει επικυρώνοντας το ορθολογικό μοντέλο του νέου συστήματος, που έχει δημιουργηθεί στο τέλος της φάσης της ανάλυσης. Ο λογικός σχεδιασμός ασχολείται κυρίως με τα εξής σημεία:

- ∅ Τους στόχους/περιορισμούς του συστήματος.
- ∅ Τους εναλλακτικούς τρόπους φυσικού σχεδιασμού.
- ∅ Τα διάφορα θέματα σχεδιασμού: πλατφόρμες υλοποίησης, αρχιτεκτονικές για Βάσεις Δεδομένων, αρχεία, επικοινωνίες, δίκτυα, κ.λπ.
- ∅ Τα χρονοδιαγράμματα.
- ∅ Τις εκτιμήσεις επικοινωνιακού/λειτουργικού φόρτου.
- ∅ Την τυποποίηση και τους ελέγχους που πιθανόν να υιοθετηθούν.

2. στο **φυσικό σχεδιασμό**, που αφορά το σχεδιασμό του Π.Σ. για θέματα σχετικά με τον τύπο και τη μορφή των δεδομένων, εκθέσεων, αναφορών, φυσικού σχεδιασμού του λογισμικού, κλπ.

Στον Πίνακα 1.1 παρουσιάζονται οι **κύριες φάσεις του κύκλου ζωής** ενός Π.Σ. και ο **ρόλος των πέντε συστατικών** στοιχείων του (άνθρωποι, διαδικασίες, δεδομένα, υλικό, λογισμικό) σε κάθε φάση.

	ΑΝΘΡΩΠΟΙ	ΔΙΑΔΙΚΑΣΙΕΣ	ΔΕΔΟΜΕΝΑ	ΥΛΙΚΟ	ΛΟΓΙΣΜΙΚΟ
Διερευνητική Μελέτη	Καθορισμός προβλήματος. Καθορισμός των ορίων του συστήματος. Καθορισμός έργου.	Καθορισμός βασικών λειτουργιών συστήματος.	Καθορισμός δεδομένων που εισάγει, επεξεργάζεται και εξάγει το σύστημα.	Προσδιορισμός της ύπαρξης και της γενικής μορφής του.	Προσδιορισμός της ύπαρξης και της γενικής μορφής του.
Μελέτη Σκοπιμότητας	Καθορισμός και επιλογή των χρηστών.	Εκτίμηση του κόστους υλοποίησης κάθε εναλλακτικής λύσης.	Εκτίμηση κόστους των εναλλακτικών τρόπων εισόδου, εξόδου και αποθήκευσης δεδομένων.	Εξέταση εναλλακτικών τρόπων παροχής του.	Εξέταση εναλλακτικών τρόπων παροχής του.
Ανάλυση Απαιτήσεων	Αναλυτική περιγραφή των νέων ρόλων των χρηστών.	Αναλυτική περιγραφή του τι κάνει κάθε επεξεργασία που πραγματοποιείται στο σύστημα.	Αναλυτική παρουσίαση της δομής των δεδομένων.	Γενικές προδιαγραφές απαιτήσεων υλικού.	Αναλυτική περιγραφή των απαιτήσεων του τι πρέπει να κάνει το λογισμικό των εφαρμογών.
Σχεδιασμός	Οργάνωση και καθορισμός των νέων ρόλων των χρηστών.	Σχεδιασμός των διαδικασιών.	Σχεδιασμός αρχείων, Βάσης Δεδομένων.	Αναλυτικές προδιαγραφές απαιτήσεων υλικού.	Λεπτομερειακή περιγραφή του σχεδιασμού του λογισμικού των εφαρμογών.

Υλοποίηση	Εκπαίδευση των χρηστών.	Βελτίωση και αναμόρφωση των διαδικασιών.	Υλοποίηση αρχείων, Βάσης Δεδομένων.	Αγορά υλικού.	Κωδικοποίηση λογισμικού εφαρμογών. Αγορά λογισμικού συστήματος.
Εγκατάσταση	Έλεγχος.	Έλεγχος και τεκμηρίωση.	Έλεγχος και τεκμηρίωση	Έλεγχος και τεκμηρίωση.	Έλεγχος και τεκμηρίωση

Πίνακας 1.1 Ο ρόλος των πέντε συστατικών στοιχείων ενός Π.Σ. στις κύριες φάσεις του κύκλου ζωής του.

Με στόχο την περιγραφή των διαφόρων σταδίων του κύκλου ζωής του λογισμικού, έχουν αναπτυχθεί διάφορα μοντέλα ανάπτυξής του (Yourdon, 1989, Norris and Rigby, 1992, Γιακουμάκης, 1994), όπως:

1. **Μοντέλο του καταρράκτη** - waterfall model (Royce, 1970; Boehm, 1984; IEEE, 1983; Rook, 1986)
2. **Σπειροειδές μοντέλο** - spiral model (Boehm, 1988) **Μοντέλο σταδιακής βελτίωσης και επαναληπτικού εμπλουτισμού** – stepwise refinement and iterative enhancement model (Wirth, 1971)
3. **Μοντέλο πρωτοτυποποίησης** - prototyping model (Agresti, 1986)
4. **Λειτουργικό μοντέλο** - operational model (Zave, 1984)
5. **Μοντέλο αυτόματου προγραμματισμού** - automatic programming model (Agresti, 1986; Stahl, 1986; Parnas, 1985)
6. **Το μοντέλο της επαναχρησιμοποίησης λογισμικού** (Davis, 1988).

Τα μοντέλα αυτά περιγράφουν κατά κανόνα γενικές κατηγορίες και περιπτώσεις ανάπτυξης λογισμικού παρακάμπτοντας λεπτομερείς αναφορές σε συγκεκριμένες περιπτώσεις. Στην πραγματικότητα τα μοντέλα αυτά καλύπτουν τις γενικές γραμμές ανάπτυξης λογισμικού, ενώ η ανάπτυξη και συντήρηση ενός προϊόντος λογισμικού είναι προσαρμοσμένα στις συνθήκες ανάπτυξής του (επιχείρηση, χρήστες, ομάδα ανάπτυξης, ύπαρξη στελεχών, εμπειρία αναλυτών, υφιστάμενος εξοπλισμός). Σε κάθε φάση ανάπτυξης εφαρμόζονται οι τεχνικές που καλύπτονται από την επιστήμη της τεχνολογίας λογισμικού.

1.10 Διαδικασία Λήψης Αποφάσεων Και Πληροφοριακά Συστήματα

Η λήψη αποφάσεων ακολουθεί βασικές λογικές φάσεις. Αποτελεί ένα σύστημα κλειστού βρόχου εισροής, παραγωγής, αξιολόγησης και εκροής πληροφοριών.

Ø Καθορισμός Προβλήματος: Η έναρξη της διαδικασίας λήψης απόφασης γίνεται με την εμφάνιση κάποιου προβλήματος. Ο καθορισμός του προβλήματος απαιτεί τη διατύπωση των **πραγματικών ερωτημάτων** τα οποία ζητούν απάντηση. Η ύπαρξη του προβλήματος συνδέεται άμεσα με την ύπαρξη απόκλισης μεταξύ τεθέντος στόχου και του αποτελέσματος από την απόκλιση και επιπλέον με την απόκλιση μεταξύ νέου στόχου και της τρέχουσας πραγματικότητας. Κατά συνέπεια ο καθορισμός του προβλήματος ακολουθεί τα ακόλουθα βήματα:

A) Εντοπισμός της απόκλισης: Αυτός ο εντοπισμός θα πρέπει να είναι έγκαιρος ώστε να υπάρχει δυνατότητα περιορισμού της απόκλισης. Μπορεί να γίνεται με πρόβλεψη (της απόκλισης η οποία πρόκειται να συμβεί) ή με διαπίστωση (η απόκλιση έχει συμβεί). Αν ο εντοπισμός γίνεται όταν το πρόβλημα είναι μη αναστρέψιμο, τότε ο στόχος είναι ο περιορισμός των δυσάρεστων συνεπειών από την εμφάνιση της απόκλισης.

B) Μελέτη της απόκλισης: Αναζητούνται τα πραγματικά αίτια τα οποία προκάλεσαν την απόκλιση. Τα βασικά σημεία τα οποία θα πρέπει να προσεχθούν ιδιαίτερα είναι τα ακόλουθα:

1. Ο διαχωρισμός των συμπτωμάτων, με τα οποία εμφανίζεται το πρόβλημα, από τα πραγματικά αίτια τα οποία προκάλεσαν το πρόβλημα.
2. Τα ίδια ακριβώς συμπτώματα μπορεί να παράγονται από εντελώς διαφορετικά αίτια.
3. Το εμφανισθέν πρόβλημα μπορεί να αποτελεί την ιδιαίτερη έκφραση ενός γενικότερου προβλήματος του οργανισμού.

Γ) Καθορισμός των επιδιώξεων: Ο εντοπισμός των πραγματικών ερωτημάτων τα οποία ζητούν απάντηση θα πρέπει να συνοδεύεται από την ύπαρξη προτύπων των στόχων ή των επιθυμητών αποτελεσμάτων προς τα οποία θα κατευθύνονται οι απαντήσεις και με την βοήθεια των οποίων κρίνεται η σοβαρότητα κάθε προβλήματος. Στο βήμα αυτό όπως καταλαβαίνουμε κυρίαρχο ρόλο παίζει η αντίληψη ότι κάτι συμβαίνει κάπου στον οργανισμό, για συγκεκριμένους λόγους και με κάποιες συνέπειες στη λειτουργία του οργανισμού. Αυτή η αντίληψη αποτελεί προϊόν του διοικητικού και λειτουργικού ελέγχου του οργανισμού. Κύρια πηγή πληροφοριών μπορεί να αποτελέσει ένα MIS.

Ø Εναλλακτικές Λύσεις: Ο καθορισμός των ερωτημάτων τα οποία ζητούν απάντηση ακολουθείται από την παράθεση των διάφορων απαντήσεων οι οποίες μπορεί να ικανοποιούν αυτά τα ερωτήματα. Οι απαντήσεις αυτές αποτελούν και τις εναλλακτικές λύσεις του προβλήματος. Σε αυτή τη φάση παρατίθενται όλες οι

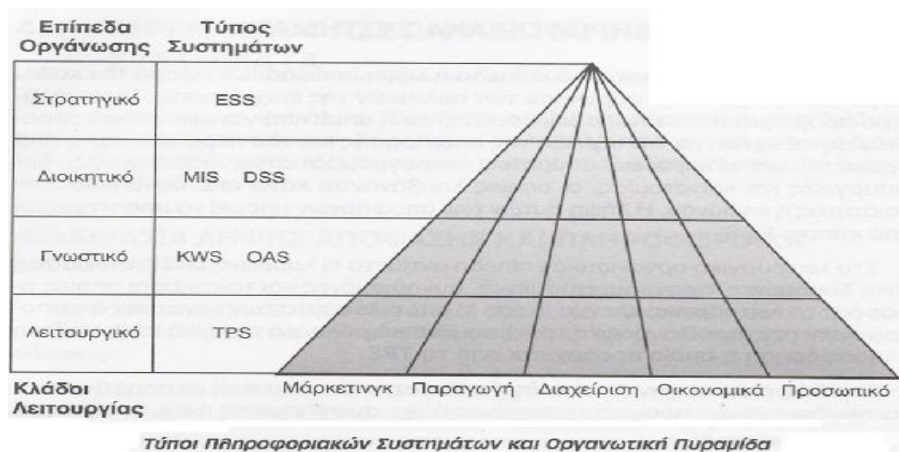
πιθανές λύσεις του προβλήματος, καταγράφοντας τις προϋποθέσεις εφαρμογής τους, ανεξάρτητα από τη δυνατότητα υλοποίησής τους. Στο βήμα της καταγραφής των εναλλακτικών λύσεων απαιτείται μια πιο προσεκτική θεώρηση της λειτουργίας του οργανισμού. Σε αυτή τη περίπτωση μπορεί να αξιοποιηθεί ένα μικρό DSS.

Ø Αξιολόγηση Λύσεων : Με την ολοκλήρωση της προηγούμενης φάσης, ο λαμβάνων την απόφαση κατέχει μια ομάδα εναλλακτικών λύσεων οι οποίες θα πρέπει να **αξιολογηθούν** και να **ιεραρχηθούν** με τη βοήθεια κάποιου κριτηρίου. Τα βήματα της αξιολόγησης είναι:

- 1) Διαβάθμιση των χαρακτηριστικών
- 2) Αποτίμηση του κόστους και της ωφέλειας
- 3) Πρόβλεψη των συνθηκών εφαρμογής.

Ø Επιλογή Της Λύσης : Από τις διάφορες αξιολογήσιμες λύσεις μια μόνο θα επιλεγεί. Το βασικότερο κριτήριο είναι η **σύγκριση** του πηλίκου κόστους/ωφέλεια συνεπικουρούμενου και από δευτερεύοντα κριτήρια. Μια λύση πάντως είναι άμεσα απορριπτέα εφόσον παρουσιάζει κόστος μεγαλύτερο της ωφέλειας.

Στα βήματα της αξιολόγησης των εναλλακτικών λύσεων και της επιλογής της τελικής λύσης απαιτούνται τεχνικές οι οποίες επιτρέπουν την αξιολόγηση των συνεπειών στον οργανισμό από την επιλογή κάθε εναλλακτικής λύσης. Ένα μεγάλο DSS μπορεί να αποτελέσει χρήσιμο εργαλείο για τον λαμβάνοντα την απόφαση. Τέλος στην εφαρμογή της επιλεγμένης λύσης βασικό ζητούμενο είναι η αξιολόγηση του αποτελέσματος της εφαρμοσμένης λύσης. Δεδομένου ότι αυτό αποτελεί προϊόν ελέγχου, τα MIS μπορεί να αποτελέσει και πάλι χρήσιμο εργαλείο.



Έτσι αφού παρουσιάσαμε την διαδικασία λήψης αποφάσεων μπορούμε να αναφέρουμε επιγραμματικά τόσο τους τύπους αποφάσεων όσο και τα μοντέλα αποφάσεων.

Τύποι Αποφάσεων: Ο τρόπος με τον οποίο λαμβάνεται μια απόφαση δεν είναι καθορισμένος και εξαρτάται από πολλούς παράγοντες όπως ο τύπος της απόφασης για παράδειγμα.

Ø Κατά τον **H. A Simon** οι αποφάσεις διακρίνονται σε:

1. **Προγραμματισμένες** (programmed): Αφορούν επαναλαμβανόμενα προβλήματα, έχουν συγκεκριμένη δομή και έχουν σχεδόν τυποποιημένο τρόπο αντιμετώπισης.
2. **Απρογραμμάτιστες** (non programmed): Αναφέρονται σε πρωτοεμφανιζόμενα προβλήματα, αδόμητα και δεν έχουν συγκεκριμένο τρόπο αντιμετώπισης.

Ø Κατά τον **P. Drucker**, οι αποφάσεις διακρίνονται σε:

1. **Τακτικές** (tactical): Στρέφονται στη φάση της υλοποίησης του προγραμματισμού, με κύριους αντικειμενικούς στόχους τη μεγιστοποίηση του κέρδους ή την ελαχιστοποίηση του κόστους.
2. **Στρατηγικές** (strategic): Αναφέρονται στον καθορισμό αντικειμενικών στόχων, τον προσδιορισμό των διαθέσιμων πόρων κ.λπ.

Ø Κατά τον **A. L. Debbek** οι αποφάσεις διακρίνονται:

1. **Συνηθισμένες** (routine) : Παρουσιάζουν μεγάλη συχνότητα εμφάνισης.
2. **Δημιουργικές** (creative) : Αφορούν πρωτοεμφανιζόμενα προβλήματα.
3. **Διαπραγμάτευσης** (negotiated) : Αποτελούν αντικείμενο διαπραγμάτευσης.

Ø Κατά τον **W. J. Gore** οι αποφάσεις διαιρούνται σε:

1. **Συνηθισμένες** (routine): οι οποίες με τη σειρά τους διαχωρίζονται σε Προσαρμογής (adaptive) και σε Καινοτομίας.

Κατά άλλους συγγραφείς όπως ο **Kenneth** και ο **Jane Laudon**, οι αποφάσεις διαιρούνται σε:

1. **Δομημένες**: Είναι τυποποιημένες και επαναλαμβανόμενες.
2. **Αδόμητες**: Είναι μη τυποποιημένες και σπάνια επαναλαμβάνονται.

Κατά την **E. Καρασαββίδου** όλες οι παραπάνω κατηγορίες μπορούν να συνδυαστούν σε δύο βασικές κατηγορίες:

1. Αποφάσεις **προγραμματισμένες, συνηθισμένες, τακτικές** και **διαπραγμάτευσης**, οι οποίες λαμβάνονται συνήθως κάτω από συνθήκες βεβαιότητας, αφορούν μεσαία και κατώτερα διοικητικά στελέχη.
2. Αποφάσεις **απρογραμμάτιστες, στρατηγικές, δημιουργικές** και **καινοτομίας**, οι οποίες λαμβάνονται συνήθως κάτω από συνθήκες αβεβαιότητας ή κινδύνου, αφορούν τα ανώτερα διοικητικά στελέχη και σχετίζονται με τον καθορισμό γενικών στόχων του οργανισμού, την προσαρμογή της πολιτικής του στις συνθήκες του περιβάλλοντος κ.λπ.

Μοντέλα Αποφάσεων: Με βάση τα προαναφερόμενα είναι προφανές ότι η λήψη απόφασης για να οδηγήσει σε επιλογή και εφαρμογή μιας λύσης θα πρέπει να πάρει συγκεκριμένη μορφή, η οποία να λαμβάνει υπόψη όλες τις προαναφερόμενες

παραμέτρους. Η συγκεκριμενοποίηση της διαδικασίας λήψης της απόφασης οδηγεί στο **Μοντέλο Απόφασης** (Decision Model), το οποίο ορισμένες φορές μπορεί να τυποποιηθεί. Ο καθορισμός του μοντέλου βασίζεται συχνότατα στα μαθηματικά, την έρευνα, τη στατιστική και τη θεωρία των πιθανοτήτων. Η χρήση των μοντέλων αποφάσεων παρουσιάζει τα ακόλουθα πλεονεκτήματα:

- ∅ Ορισμένες φορές τυποποιούνται και χρησιμοποιούνται γενικευμένα από πλήθος διοικητικών στελεχών.
- ∅ Επιταχύνουν τη λήψη απόφασης.
- ∅ Περιορίζουν, σε λογικά πλαίσια, την υποκειμενικότητα του λαμβάνοντος την απόφαση.
- ∅ Λαμβάνονται παρόμοιες αποφάσεις κάτω από παρόμοιες συνθήκες.

Τα μοντέλα αποφάσεων μπορεί να κατηγοριοποιηθούν χρησιμοποιώντας διαφορετικά κριτήρια.

Ως προς τη **δυνατότητα γενίκευσης** την οποία παρέχουν διακρίνονται σε:

- ∅ **Εικονικά** (Iconic): Τα χαρακτηριστικά του συστήματος παριστάνονται υπό κλίμακα (μακέτες, σχέδια, φωτογραφίες κ.λπ.)
- ∅ **Αναλογικά** (Analogue): Τα χαρακτηριστικά του συστήματος παρίστανται από διαγράμματα ροής πληροφοριών, σχέδια ηλεκτρονικών εγκαταστάσεων κ.λπ.)
- ∅ **Συμβολικά** (Symbolic): Χρησιμοποιούν σύμβολα για την αναπαράσταση του συστήματος όπως αλγόριθμους, μαθηματικές σχέσεις κ.λπ.

Ως προς τον **έλεγχο των παραμέτρων** οι οποίες υπεισέρχονται στα μοντέλα, διακρίνονται σε:

- ∅ **Περιγραφικά** (Descriptive): Δεν περιέχουν ελεγχόμενες παραμέτρους και η λειτουργία του συστήματος περιγράφεται λεκτικά.
- ∅ **Επεξηγηματικά** (Explanatory): Όλες οι περιεχόμενες παράμετροι είναι ελεγχόμενες και η περιγραφή του συστήματος γίνεται με την παράθεση των σχέσεων μεταξύ των παραμέτρων (αλγόριθμοι με τη βοήθεια της στατιστικής, της αριθμητικής ανάλυσης κ.λπ.).

Ως προς την **εξέλιξή τους στο χρόνο** τα μοντέλα διακρίνονται σε:

- ∅ **Στατικά** (Static): Παρέχουν λύσεις μόνο κάποια χρονική στιγμή.
- ∅ **Δυναμικά** (Dynamic): Παρέχουν λύσεις κάθε χρονική στιγμή.

Ως προς τον **τύπο και το περιεχόμενό τους**, τα μοντέλα διακρίνονται σε:

- ∅ **Αναμονής** (Queuing): Αναφέρονται στο σειριακό ρυθμό εξυπηρέτησης στοιχείων τα οποία φθάνουν σε σταθμό εξυπηρέτησης (ουρές αναμονής, επεξεργασία προγραμμάτων από Η/Υ κ.λπ.).
- ∅ **Αποθεμάτων** (Inventory): Παρέχουν τρόπους διαχείρισης των αποθεμάτων.
- ∅ **Κατανομής** (Allocation): Παρέχουν τρόπους κατανομής πόρων σε αντίστοιχα έργα.
- ∅ **Αντικατάστασης και συντήρησης** (Replacement And Maintenance): Επιτρέπουν την παρακολούθηση της φθοράς του εξοπλισμού και προβλέπουν

την απαιτούμενη προληπτική αντικατάσταση εξαρτημάτων καθώς και την προληπτική συντήρηση του εξοπλισμού.

- Ø **Δρομολογίων** (Routing And Scheduling): Επιτρέπουν τη δρομολόγηση των διαδικασιών ενός έργου (μέθοδοι PERT, COST κλπ) π.χ. χρονικός προγραμματισμός έργου.
- Ø **Αναζήτησης** (Search): Επιτρέπουν την αναζήτηση των προϋποθέσεων ώστε να μεγιστοποιείται η πιθανότητα επίτευξης ενός στόχου.
- Ø **Ανταγωνισμού** (Competition): Αποτελούν μοντελοποίηση των «παιγνίων».

Ως προς τον **τρόπο επεξεργασίας τους από τον Η/Υ**, τα μοντέλα διακρίνονται σε:

- Ø **Αλγοριθμικά** (Algorithmic): Η λύση παρέχεται μέσω μαθηματικής σχέσης των παραμέτρων του μοντέλου.
- Ø **Ευριστικά** (Heuristic): Η απόφαση λαμβάνεται με τη βοήθεια εμπειρικών ή λογικών κανόνων (π.χ. προσομοίωση).

Ως προς τη **συμβολή τους στη διοίκηση**, τα μοντέλα διακρίνονται σε :

- Ø **Βελτιστοποίησης** (Optimization): Επιτρέπουν τον εντοπισμό της βέλτιστης μεταξύ των εναλλακτικών λύσεων.
- Ø **Προσομοίωσης** (Simulation): Αναπαράγουν εικονικά μία πραγματική διαδικασία.

Όσον αφορά τη **σχέση αποφάσεων και σχεδιασμού ενός πληροφοριακού συστήματος** είναι ευρέως γνωστό ότι οι απαιτήσεις των διοικητικών στελεχών από τα πληροφοριακά συστήματα είναι συνάρτηση των αυξανόμενων αναγκών τους σε πληροφόρηση μέσα σε ένα σύνθετο και ραγδαία μεταβαλλόμενο περιβάλλον. Αυτό συμβαίνει διότι τα διοικητικά στελέχη απαιτούν τόσο τυποποιημένα συστήματα για τον σχεδιασμό, την οργάνωση και τον συντονισμό, όσο και εύκολα και πολυδύναμα συστήματα τα οποία να επιτρέπουν την κάλυψη ευρέος φάσματος απαιτήσεων και την προσαρμογή τους στις μεταβολές που υφίσταται το περιβάλλον.

Όμως αν και τα τυποποιημένα συστήματα έχουν μεγάλες δυνατότητες στο λειτουργικό τομέα του κάθε οργανισμού, διαθέτουν περιορισμένες δυνατότητες στο διοικητικό τομέα δεδομένου ότι η διοίκηση περιέχει μεγάλο ποσοστό ατυποποίητης ανθρώπινης αυτενέργειας (διαίσθηση, ψυχολογική συγκρότηση κ.λπ.). Σε αυτό θα πρέπει να προστεθεί ότι μία απόφαση διαφέρει από άτομο σε άτομο και από κατάσταση σε κατάσταση ως προς τη σημασία των στόχων, την ποιότητα του λαμβάνοντος την απόφαση, το οργανωτικό μοντέλο κ.λπ. Κατά συνέπεια τα συστήματα λειτουργούν σαν εργαλεία υποστήριξης της λήψης των αποφάσεων, σεβόμενα την πρωτοκαθεδρία του υποκειμενικού παράγοντα. Δηλαδή υποστηρίζουν τη λήψη αποφάσεων χωρίς να παράγουν αποφάσεις.

Τα προαναφερόμενα έχουν άμεσες **επιπτώσεις στον σχεδιασμό νέων συστημάτων**. Αυτό γίνεται διότι η λήψη αποφάσεων σε έναν οργανισμό λαμβάνεται από ομάδες ατόμων με τη χρήση οργανωτικών διαδικασιών και τα συστήματα καλούνται να υποστηρίξουν και τις ομάδες αυτές αλλά και τις οργανωτικές διαδικασίες.

Λαμβάνοντας υπόψη τα προηγούμενως γραφόμενα, τα βασικότερα **χαρακτηριστικά των συστημάτων**, τα οποία θα πρέπει να σεβαστεί ο σχεδιαστής νέων συστημάτων, θα πρέπει να είναι:

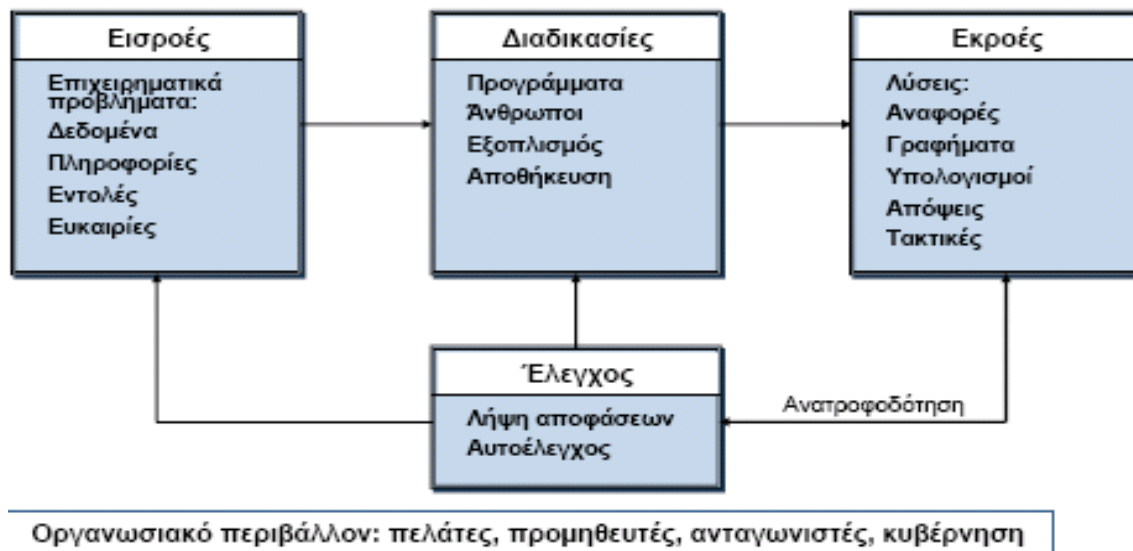
∅ Η **ευλυγισία** τους, προσφέροντας στους χρήστες τις περισσότερες δυνατές εναλλακτικές δυνατότητες στη συλλογή και αξιολόγηση των δεδομένων και στις μεταβολές των περιβαλλοντικών συνθηκών.

∅ Η δυνατότητα **υποστήριξης** ατόμων με διαφορετικές απόψεις διοίκησης, με διαφορετικές ικανότητες, με διαφορετικές γνώσεις, με τον υποκειμενισμό τους, τα οποία λειτουργούν σε διαφορετικά οργανωτικά περιβάλλοντα.

∅ Η δυνατότητα παροχής μοντέλων με ισχυρές αναλυτικές και συνθετικές **ικανότητες** τα οποία επιτρέπουν την παρουσίαση εναλλακτικών αποτελεσμάτων με βάση τα εισερχόμενα δεδομένα.

∅ Να αντνακλούν τις οργανωτικές και πολιτικές **αναζητήσεις** του οργανισμού.

∅ Να ακολουθούν τις **μεταβολές** της οργάνωσης, της πολιτικής και των διαδικασιών και να είναι γνωστές οι δυνατότητές τους.



Γραφική Αναπαράσταση ΠΣ

ΚΕΦΑΛΑΙΟ 2

ΜΕΘΟΔΟΙ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

Σκοπός του συγκεκριμένου κεφαλαίου είναι να μας προσφέρει μια όσο το δυνατόν πληρέστερη και εκτενέστερη εικόνα των λειτουργιών και των διεργασιών, της υποδομής και των παραγόντων που συνίσταται στην επιτυχή σχεδίαση ενός πληροφοριακού συστήματος. Μέσα από τη Θεωρία των Συστημάτων θα εξετάσουμε πώς διενεργείται η ανάλυση και ο σχεδιασμός ενός πληροφοριακού συστήματος γνωρίζοντας τις ποιοτικές παραμέτρους αξιολόγησης και επιτυχίας του, ώστε στην συνέχεια να είμαστε σε θέση να αναπτύξουμε το «Πληροφοριακό Σύστημα του ΤΕΙ Πάτρας».



Η σωστή ανάπτυξη και η εύρυθμη λειτουργία του πληροφοριακού συστήματος εξαρτώνται σε μεγάλο βαθμό από ορισμένα συγκεκριμένα στάδια, τα οποία ακολουθούνται, προκειμένου να πραγματοποιηθεί η ανάλυση και ο σχεδιασμός του.

2.1 Επικοινωνία Ανθρώπου – Μηχανής

Δύο είναι τα βασικά κριτήρια με τα οποία μπορεί να αξιολογηθεί ένα Π.Σ: η ακρίβεια και η ευκολία χρήσης. Ο χρήστης, όπως είναι φυσικό, έχει την απαίτηση το σύστημα να παράγει όχι μόνο τα σωστά αποτελέσματα στη σωστή χρονική στιγμή, αλλά και να είναι εύκολο στη χρήση του. Αυτό που τον ενδιαφέρει είναι να μην απαιτείται να αλλάξει ριζικά τις συνήθειές του και να μην απαιτούνται επιπλέον γνώσεις ή δεξιότητες για να το χρησιμοποιήσει. Ο χρήστης βλέπει το σύστημα μέσω του συστήματος διεπαφής ανθρώπου μηχανής (HCI - Human Computer Interaction) και το κρίνει μέσω των υπηρεσιών που του παρέχει το HCI. Η ακρίβεια αναλύεται και υλοποιείται σχετικά εύκολα, αφού από τη φύση της σχετίζεται με μετρήσιμες οντότητες, ποσοτικές ή ποιοτικές. Από την άλλη πλευρά, ένα σύστημα είναι εύκολο στη χρήση του, όταν παρέχει άνεση στις συναλλαγές που κάνουν οι χρήστες μαζί του. Έτσι οι παράγοντες που επιδρούν πάνω στην εύκολη χρήση, είναι εκείνοι που επιδρούν πάνω στην άνεση που παρέχει. Οι παράγοντες αυτοί μπορούν να ομαδοποιηθούν σε τρεις μεγάλες κατηγορίες:

- Ø **Κοινωνικούς παράγοντες**, που σχετίζονται με το εργασιακό περιβάλλον και επιδρούν στη συγκινησιακή άνεση του χρήστη.
- Ø **Φυσικοεργονομικούς παράγοντες**, που σχετίζονται με το υλικό και επιδρούν στη σωματική άνεση του χρήστη.
- Ø **Ψυχολογικοεργονομικούς παράγοντες**, που σχετίζονται με το σχεδιασμό του λογισμικού και επιδρούν πάνω στη γνωστική – αντιληπτική άνεση του χρήστη.

Η γνωστική – αντιληπτική άνεση είναι στενά συνδεδεμένη με το λογισμικό και τη φιλικότητά του. Δύο άλλα στοιχεία που επηρεάζουν τη γνωστική – αντιληπτική άνεση είναι η διαθεσιμότητα και η ανταπόκριση, που χαρακτηρίζουν ένα σύστημα. Η επικοινωνία του χρήστη με τον υπολογιστή απαιτεί προφανώς τη δημιουργία μιας **γλώσσας επικοινωνίας**. Η γλώσσα αυτή χωρίζεται σε δύο στενά συνδεδεμένα υποσύνολα. Το πρώτο περιγράφει το πώς γίνεται η επικοινωνία του χρήστη με τον υπολογιστή ενώ το δεύτερο τον τρόπο που ο υπολογιστής επικοινωνεί με το χρήστη.

Ένα ολοκληρωμένο σύστημα διεπαφής ανθρώπου μηχανής πρέπει να έχει μεταξύ άλλων να έχει και τα παρακάτω βασικά χαρακτηριστικά:

1. Απλότητα, δηλαδή να μην είναι σύνθετο ή/και πολύπλοκο.
2. Να ικανοποιεί όλες τις ανάγκες του χρήστη.
3. Άμεση απόκριση στις εντολές του χρήστη.
4. Όλες οι ενέργειές του να ξεκινούν και να ελέγχονται από το χρήστη.
5. Αξιοπιστία.
6. Συνέπεια.
7. Σταθερότητα.
8. Ασφάλεια και αντιμετώπιση λαθών, (π.χ. ανεπιθύμητη διαγραφή αρχείων ή παραβιάσεις από τρίτους).
9. Ομοιομορφία στα επιμέρους τμήματά του.

Το γνωστικό αντικείμενο στην επικοινωνία ανθρώπου μηχανής είναι πολυσύνθετο με αποτέλεσμα να σχετίζονται με αυτό, εκτός από την ανάλυση συστημάτων, και η τεχνολογία λογισμικού, ο προγραμματισμός υπολογιστών, ο σχεδιασμός του υλικού, η τεχνητή νοημοσύνη, η ψυχολογία, η κοινωνιολογία, η θεωρία συστημάτων, η θεωρία οργάνωσης επιχειρήσεων, κα.

Στις παρακάτω ενότητες περιγράφονται μερικοί τρόποι με τους οποίους μπορούμε να αναπτύξουμε το Πληροφοριακό μας Σύστημα, ανάλογα με τις απαιτήσεις που θα πρέπει αυτό, να ικανοποιεί.

2.2 Σχεδιασμός Συστημάτων – Διαγραμματικές Τεχνικές

Για την **κατασκευή του μοντέλου** ενός ολοκληρωμένου πληροφοριακού συστήματος απαιτείται να πραγματοποιηθούν οι εξής διεργασίες:

- ∅ Ανάλυση και μοντελοποίηση των επεξεργασιών που γίνονται στο σύστημα (κατασκευή του μοντέλου επεξεργασιών του συστήματος).
- ∅ Ανάλυση και μοντελοποίηση των δεδομένων που διαχειρίζεται το σύστημα (κατασκευή του μοντέλου δεδομένων του συστήματος).
- ∅ Δημιουργία του μοντέλου επεξεργασίας δεδομένων (συγχώνευση των δύο προηγούμενων μοντέλων).

Η κατασκευή των αντίστοιχων μοντέλων και στις τρεις περιπτώσεις γίνεται κυρίως με τη χρήση **διαγραμματικών τεχνικών**. Οι τεχνικές αυτές που είναι ευρύτερα διαδεδομένες και χρησιμοποιούνται στην ανάπτυξη του μοντέλου ενός Π.Σ, βοηθούν ιδιαίτερα στην κατανόηση, την ανάλυση και την παρουσίαση των προβλημάτων και των απαιτήσεων του συστήματος αφού έχουν την δυνατότητα να περιγράφουν τις λειτουργίες του με χρήση εικόνων (διαγραμμάτων, δένδρων, πινάκων και χωρίζονται κυρίως σε δύο κατηγορίες. Σε αυτές που χρησιμοποιούνται για τη **γενική περιγραφή** μιας λειτουργίας ή ενός προβλήματος και σε αυτές που χρησιμοποιούνται για την **λεπτομερή περιγραφή** τους.

Οι πιο γνωστές και χρησιμοποιούμενες τεχνικές που ανήκουν στην κατηγορία των **τεχνικών γενικής περιγραφής** είναι οι εξής:

- ∅ διαγράμματα ροής δεδομένων
- ∅ διαγράμματα δομής (structure diagrams)
- ∅ διαγράμματα HIPO (Hierarchical Input – Process – Output)
- ∅ διαγράμματα HOS (Higher Order Software)
- ∅ διαγράμματα Warnier –Orr
- ∅ διαγράμματα ενεργειών
- ∅ διαγράμματα Jackson

Οι πιο γνωστές και χρησιμοποιούμενες τεχνικές που ανήκουν στην κατηγορία των **τεχνικών λεπτομερούς περιγραφής** είναι οι εξής:

- Ø δομημένη περιγραφή και ψευδοκώδικας
- Ø δέντρα αποφάσεων
- Ø πίνακες αποφάσεων
- Ø διαγράμματα Nassi-Shneiderman
- Ø διαγράμματα HIPO (Hierarchical Input – Process – Output)
- Ø διαγράμματα HOS (Higher Order Software)
- Ø διαγράμματα Warnier Orr
- Ø διαγράμματα ενεργειών

Οι περισσότερες από τις παραπάνω διαγραμματικές τεχνικές χρησιμοποιούνται καθ' όλη τη διάρκεια της ανάλυσης, του σχεδιασμού, της υλοποίησης και της λειτουργίας ενός Π.Σ.

Από την κατηγορία των τεχνικών λεπτομερούς ανάπτυξης αξίζει να σταθούμε πιο λεπτομερειακά στην Δομημένη Περιγραφή, στο Ψευδοκώδικα και στα Δένδρα Αποφάσεων, έτσι:

Δομημένη περιγραφή και ψευδοκώδικας: Οι περισσότερες από τις τεχνικές μοντελοποίησης που παρουσιάστηκαν παραπάνω χρησιμοποιούν ιεραρχικά διαγράμματα δενδρικής μορφής για την περιγραφή του υπό ανάλυση συστήματος ή προγράμματος. Εδώ, η τεχνικές που παρουσιάζεται (δομημένη περιγραφή και ψευδοκώδικας) δε χρησιμοποιεί διαγράμματα τέτοιας μορφής. Αποτελεί μια διασκευασμένη διάλεκτο της ομιλούμενης γλώσσας που χρησιμοποιείται στη περιγραφή της δομής προγραμμάτων.

Οι διαφορές του ψευδοκώδικα και της δομημένης περιγραφής είναι ελάχιστες. Ο ψευδοκώδικας χρησιμοποιεί περισσότερο τυπικό συμβολισμό και απευθύνεται κυρίως σε ειδικούς ανάπτυξης λογισμικού. Η μορφή του μοιάζει περισσότερο με τη μορφή που θα έχει ο κώδικας που θα υλοποιήσει τη διεργασία που περιγράφεται. Πολλές φορές περιλαμβάνει εντολές που αντιστοιχούν σε αντίστοιχες εντολές της γλώσσας προγραμματισμού που θα χρησιμοποιηθεί για την τελική υλοποίηση.

Ο **ψευδοκώδικας** παρουσιάζει τα εξής βασικά χαρακτηριστικά :

1. έχει ιεραρχική δομή.
2. έχει δομή παρόμοια με τη δομή του κώδικα προγράμματος που θα χρησιμοποιηθεί για την υλοποίησή του.
3. είναι εύκολη η κατανόησή του.
4. υποστηρίζει τη χρήση επεξηγηματικών σχολίων.

Η **δομημένη περιγραφή** υποστηρίζει τους εξής βασικούς τύπους δομών:

1. ακολουθία, όπου οι ενέργειες εκτελούνται με τη σειρά που εμφανίζονται (ακολουθιακά) από πάνω προς τα κάτω.
2. επιλογή (if then else), όπου αν μια συνθήκη αληθεύει ,τότε μια συγκεκριμένη ενέργεια εκτελείται. Σε αντίθετη περίπτωση, εκτελείται μια διαφορετική ενέργεια.

3. δομή case, όπου υπάρχουν πολλές αμοιβαία αποκλειόμενες συνθήκες. Όταν μια από τις συνθήκες αυτές αληθεύει, εκτελείται μια συγκεκριμένη ενέργεια που αντιστοιχεί στη συγκεκριμένη συνθήκη.
4. επανάληψη (repeat while, repeat until), όπου η εκτέλεση ενός καθορισμένου συνόλου ενεργειών επαναλαμβάνεται όσο ισχύει μία συγκεκριμένη συνθήκη.

Ένα παράδειγμα ψευδοκώδικα ενός απλού προγράμματος παρουσιάζεται για ένα πρόγραμμα που ζητάει από το χρήστη δύο αριθμούς (πρώτα ένα μεγάλο και στη συνέχεια ένα μικρότερο) :

```

Program Enter a large and smaller number

Print "Please enter a big number"

Get the number

Print "Please enter a smaller number"

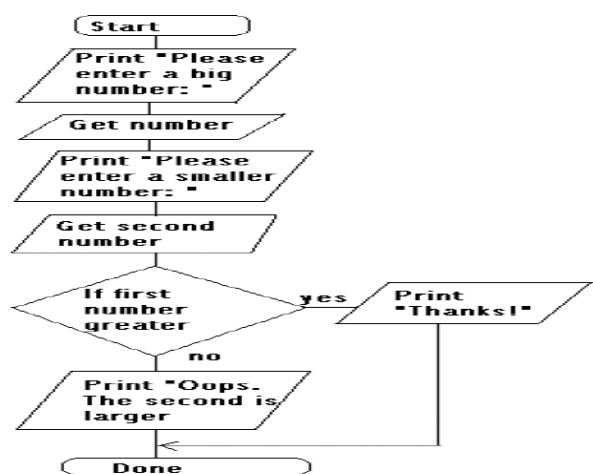
Get the second number

If the first number is larger
Then
    Print "Thanks!"
Else
    Print "Oops! The second is bigger."
End if

End Program Enter a large and smaller number.

```

Στο Σχήμα 2.1 γίνεται διαγραμματική αναπαράσταση του ψευδοκώδικα του σχήματος με χρήση Διαγράμματος Ροής.

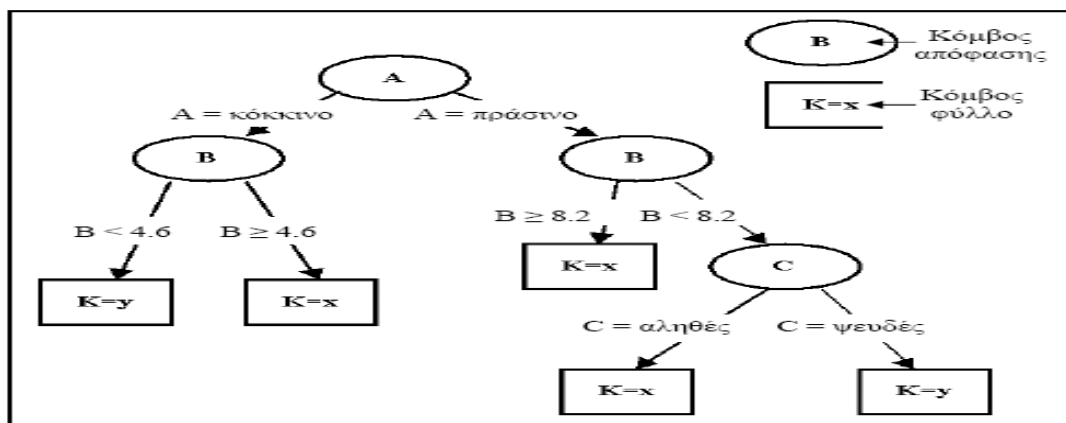


Σχήμα 2.1

Δένδρα αποφάσεων: Τα δένδρα αποφάσεων ανήκουν στις διαγραμματικές τεχνικές λεπτομερούς περιγραφής. Εκτός από το χώρο της ανάπτυξης πληροφοριακών συστημάτων, χρησιμοποιούνται και σε διάφορους άλλους επιστημονικούς/τεχνολογικούς χώρους:

1. ανάλυση αλγορίθμων.
2. αναγνώριση προτύπων.
3. σχεδίαση λογικών κυκλωμάτων και έλεγχος αξιοπιστίας τους.

Ένα δέντρο απόφασης στην πραγματικότητα αποτελεί μια διακριτή συνάρτηση η οποία ανάλογα με την τιμή μιας μεταβλητής καθορίζει την εκτέλεση ή μη μιας συγκεκριμένης ενέργειας. Η ενέργεια αυτή μπορεί να είναι είτε η εμφάνιση της τιμής της συνάρτησης είτε η ανάθεσή της τιμής αυτής σε μια άλλη μεταβλητή η οποία με τη σειρά της μπορεί να οδηγήσει στην εκτέλεση μιας άλλης ενέργειας. Επομένως η εκτέλεση μιας ενέργειας εξαρτάται τόσο από την τρέχουσα τιμή μιας μεταβλητής όσο και από όλες τις υπόλοιπες ενέργειες που έχουν ολοκληρωθεί πριν από αυτή. Τα δένδρα αποφάσεων κατασκευάζονται συνήθως με βάση την περιγραφή ενός προβλήματος και προσδιορίζουν κυρίως τη διαδικασία λήψης αποφάσεων που απαιτείται για την επίλυση του προβλήματος ενώ παράλληλα καθορίζουν τις μεταβλητές ελέγχου, τις ενέργειες που εκτελούνται και τη σειρά με την οποία πρέπει να εκτελεστούν. Κάθε φορά που εκτελείται ένα δέντρο απόφασης, ακολουθείται ένα μονοπάτι πάνω στο δέντρο, το οποίο εξαρτάται από τις τιμές που παίρνουν οι μεταβλητές ελέγχου. Το μονοπάτι αυτό ξεκινάει πάντα από τη ρίζα και καταλήγει σε ένα φύλλο του δέντρου.



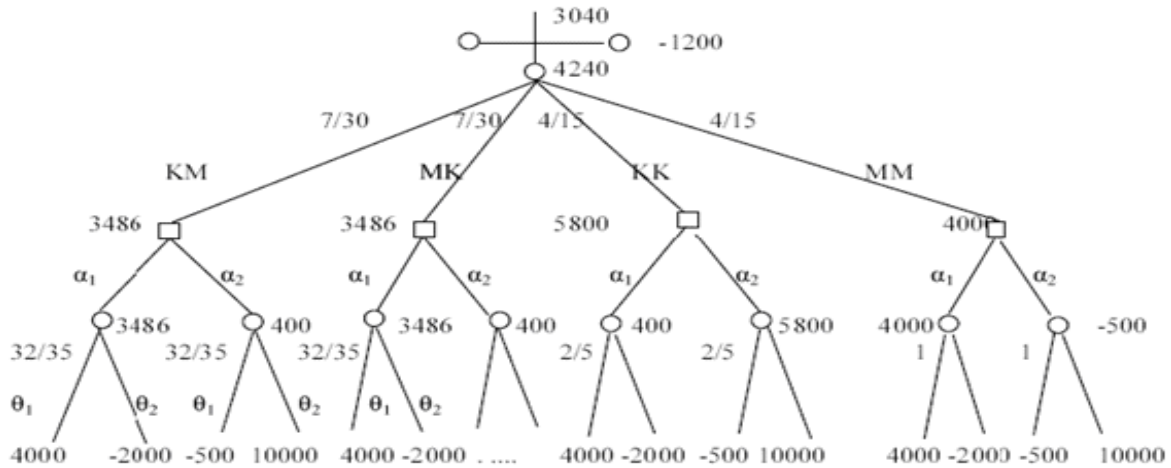
Ένα απλό παράδειγμα δέντρου απόφασης.

Στο δένδρο απόφασης του παραπάνω σχήματος υπάρχουν δύο είδη κόμβων:

1. **κόμβοι απόφασης:** αντιπροσωπεύουν σημεία στα οποία η επιχείρηση πρέπει να πάρει κάποια απόφαση επιλέγοντας μέσα από ένα σύνολο δυνατών ενεργειών. Η απόφαση συνήθως εξαρτάται είτε από την τιμή μιας μεταβλητής είτε από μια συγκεκριμένη κατάσταση.

2. τερματικοί κόμβοι (κόμβοι φύλλα): Οι τερματικοί κόμβοι περιέχουν το τελικό αποτέλεσμα που θα έχει η πορεία μέσα στο δένδρο των αντίστοιχων αποφάσεων που οδηγούν σε αυτούς.

Στο Σχήμα 2.2 όμως παρουσιάζεται ένα δένδρο απόφασης που περιέχει ένα επιπλέον είδος κόμβων, του κόμβους τυχαιότητας (συμβολίζονται με κύκλο). Οι κόμβοι τυχαιότητας αντιπροσωπεύουν σημεία στα οποία η τύχη ή η πιθανότητα παίζει καθοριστικό ρόλο στη λήψη κάποιας απόφασης (ή στην εκτέλεση κάποιας ενέργειας).



Σχήμα 2.2

Πίνακες αποφάσεων: Αποτελούν ένα εναλλακτικό τρόπο παρουσίασης των πληροφοριών που παρέχουν τα δέντρα αποφάσεων. Χρησιμοποιούνται κυρίως για τον προσδιορισμό των ενεργειών που πρέπει να εκτελεστούν όταν ισχύουν ορισμένοι συνδυασμοί συνθηκών. Αποτελούνται από τέσσερα βασικά τμήματα όπως φαίνεται στο Σχήμα 2.3.

Τα τμήματα Α (Συνθήκες) και Β (Κανόνες) αποτελούν το μέρος των συνθηκών και των κανόνων, ενώ τα τμήματα Γ (Υλοποίηση ενεργειών) και Δ (Ενέργειες) αποτελούν τα τμήματα των ενεργειών. Για να είναι πλήρης ένας πίνακας, οι κανόνες του πρέπει συνολικά να εξαντλούν όλους τους δυνατούς συνδυασμούς των συνθηκών, διαφορετικά ο πίνακας είναι ατελής.



Η γενική μορφή ενός πίνακα αποφάσεων

Σχήμα 2.3

Υπάρχουν **τρεις βασικοί τύποι** πινάκων αποφάσεων:

1. Οι πίνακες αποφάσεων περιορισμένων εγγραφών, όπου όλες οι συνθήκες προσδιορίζονται με ακρίβεια στο τμήμα των συνθηκών.
2. Οι πίνακες αποφάσεων εκτεταμένων εγγράφων, όπου το τμήμα των συνθηκών απλώς περιγράφει τις συνθήκες του κάθε κανόνα και όχι τις τιμές των συνθηκών.
3. Οι πίνακες αποφάσεων μικτών εγγραφών, όπου οι συνθήκες μπορεί να είναι είτε περιορισμένες είτε εκτεταμένες.

Οι πίνακες αποφάσεων χρησιμοποιούνται με κύριο σκοπό τη λογική απεικόνιση σύνθετων περιπτώσεων επεξεργασίας δεδομένων. Για το λόγο αυτό η χρήση τους βρίσκει μεγάλη απήχηση σε εμπορικές ή/και διοικητικές εφαρμογές. Έτσι χρησιμοποιούνται από κατασκευαστικές εταιρίες, ασφαλιστικές εταιρίες, χρηματοδοτικούς οργανισμούς, κ.λπ. Επίσης χρησιμοποιούνται για την υποστήριξη διαφόρων φάσεων έργων πληροφορικής όπως η ανάλυση και ο σχεδιασμός Π.Σ., ο προγραμματισμός, ο εντοπισμός σφαλμάτων, η τεκμηρίωση των συστημάτων/προγραμμάτων, κ.λπ.

2.3 Σχεδιασμός Συστημάτων – Προμήθεια & Παραγωγή Λογισμικού

Σε αντίθεση με το Υλικό το οποίο πάντοτε αγοράζεται, το Λογισμικό μπορεί είτε να αγοραστεί είτε να παραχθεί από την επιχείρηση ή τον οργανισμό, εφόσον βέβαια διαθέτει τα απαιτούμενα μέσα, είτε μέρος του να παραχθεί και μέρος του να αγοραστεί.

Ø 2.3.1 Αγορά Λογισμικού

Οι απαιτήσεις που πρέπει να καλύπτονται είναι:

1. Δυνατότητες σε on line, batch, spooling, multiuser, τηλεπικοινωνίες κ.λπ.
2. Utilities και Tools
3. Πακέτα
4. Λειτουργικό Σύστημα
5. Μεταγλωττιστές
6. Γεννήτορες προγραμμάτων, εκτυπωμένων καταστάσεων κ.λπ.
7. DBMS

Ζητούμενες Πληροφορίες Από Προμηθευτή :

A)Χαρακτηριστικά του Λογισμικού :

Ø Λειτουργικά Συστήματα:

1. Γενικά χαρακτηριστικά της λειτουργίας τους
2. Ποιες γλώσσες προγραμματισμού υποστηρίζουν
3. Ποιο ποσοστό της μνήμης απασχολούν σε μόνιμη βάση
4. Τρόπος διαχείρισης πληροφοριών ο οποίος υποστηρίζεται

5. Τρόπος λειτουργίας του τηλεπικοινωνιακού συστήματος
6. Ποιες οι δυνατότητες για πολυπρογραμματισμό
7. Ποιες οι δυνατότητες αυτοπροστασίας

Ø Γλώσσες Προγραμματισμού:

1. Ποιοι μεταγλωττιστές διατίθενται
2. Αν Διατίθενται γεννήτορες προγραμμάτων, καταστάσεων κ.λπ. και ποιοι
3. Utilities, tools
4. Απαιτήσεις των παραπάνω σε περιφερειακά και μνήμη

B) Τεχνική Υποστήριξη:

- Ø Υποστήριξη σε ανάλυση και προγραμματισμό νέων εφαρμογών και τροποποίηση παλιών
- Ø Ποιος ο αριθμός, οι γνώσεις και η πείρα του διατιθέμενου προσωπικού και το χρονοδιάγραμμα διάθεσης του
- Ø Συντήρηση του διατιθέμενου Λογισμικού

Γ) Εκπαίδευση Προσωπικού:

- Ø Ποιο πρόγραμμα σεμιναρίων επιμόρφωσης για τι προσωπικό προτείνεται και ποιο το χρονοδιάγραμμά του
- Ø Ποια η μορφή ενημέρωσης για τροποποιήσεις – βελτιώσεις του υπάρχοντος Λογισμικού

Δ) Εγχειρίδια λειτουργίας Λογισμικού:

- Ø Ποια προτείνονται και τι περιέχουν
- Ø Ποια η δυνατότητα παροχής επιπλέον πληροφοριών εκτός από τα προτεινόμενα εγχειρίδια

Ε) Κόστος του Λογισμικού:

- Ø Τιμές διατιθέμενων έτοιμων προγραμμάτων
- Ø Κόστος αγοράς έτοιμου Λογισμικού
- Ø Κόστος υποστήριξης
 1. Κόστος εκπαίδευσης προσωπικού
 2. Πόσο κοστίζει η ανάλυση – σχεδιασμός και ο προγραμματισμός αν τα αναλάμβανε ο προμηθευτής.

ΣΤ) Αναφορικά με γενικά ζητήματα:

- Ø Χρόνος παράδοσης (από υπογραφή σύμβασης μέχρι τη λειτουργία του Λογισμικού)
- Ø Δυνατότητες επίδειξης του Λογισμικού το οποίο προτείνεται και κατάλογος πελατών του προμηθευτή από τους οποίους μπορούν να παρασχεθούν πληροφορίες

Η αξιολόγηση των προσφορών των προμηθευτών ακολουθεί ανάλογη διαδικασία με εκείνη της αξιολόγησης των προσφορών του υλικού.

Ø 2.3.2 Προγραμματισμός

Με τον όρο **Προγραμματισμός** (Programming) εννοείται η παραγωγή Λογισμικού και αποτελεί την υλοποίηση του μεγαλύτερου τμήματος του σχεδιασμού. Είναι , κατά συνέπεια, υπεύθυνος σε μεγάλο βαθμό για την επιτυχία ή όχι του συστήματος.

Ποιοτικά χαρακτηριστικά Λογισμικού: Σε κάθε σύστημα Η/Υ μπορεί να υπάρχουν αρκετές κατηγορίες προγραμμάτων. Κάθε πρόγραμμα μπορεί να γίνει με πολλούς τρόπους. Το θέμα είναι όμως ποια είναι εκείνα τα κριτήρια τα οποία επιτρέπουν την κρίση για το ποιο πρόγραμμα εξυπηρετεί καλύτερα τους στόχους του συστήματος. Τα βασικότερα ποιοτικά χαρακτηριστικά ενός προγράμματος είναι τα ακόλουθα:

1. Η **αποτελεσματικότητα (Effectiveness)**.
2. Η **ικανότητα (Efficiency)**.
3. Η **ευκαμψία (Flexibility)**.
4. Η **απλότητα (Simplicity)**.
5. Η **συμβατότητα και φορητότητα (Compatibility and Portability)**.
6. Η **τεκμηρίωση (Documentation)**.

2.4 Κύκλος Ζωής Λογισμικού Και Γλώσσες Προγραμματισμού

2.4.1 Κύκλος Ζωής Λογισμικού

Η ζωή κάθε προγράμματος αρχίζει από τη στιγμή κατά την οποία ο προγραμματιστής έχει στα χέρια του τις σχεδιαστικές προδιαγραφές του προγράμματος και ολοκληρώνεται με τη θέση αυτού του προγράμματος σε λειτουργία και τη συντήρησή του. Η πορεία αυτή αποτελεί τον Κύκλο Ζωής του Προγράμματος (Program Life Cycle).

Ø **Προγραμματισμός εργασιών:** Αποτελεί για κάθε πρόγραμμα εκείνη τη φάση κατά την οποία λαμβάνονται γενικές αποφάσεις για την υλοποίησή του. Μερικές από αυτές τις αποφάσεις είναι:

1. Ο καθορισμός των απαιτούμενων, για το πρόγραμμα, πόρων.
2. Η επιλογή μεθοδολογίας σχεδιασμού του.

3. Ο τρόπος εγκατάστασης του.
4. Η διαδικασία δοκιμής και ενσωμάτωσης του στο ήδη υπάρχον σύστημα.
5. Ο καθορισμός των μέτρων για την εξασφάλιση της επιζητούμενης ασφάλειας του προγράμματος.
6. Ο καθορισμός της διαδικασίας έλεγχου για πιθανές μεταβολές του προγράμματος.
7. Η μορφή οργάνωσης της ομάδας εργασίας η οποία θα αναλάβει την υλοποίηση του προγράμματος.

∅ **Ανάλυση και σχεδιασμός προγράμματος (Program analysis and design):** Το βασικό πρόβλημα του προγραμματιστή σε αυτή τη φάση είναι η δημιουργία της λογικής δομής του προγράμματος. Αυτό απαιτεί την απόλυτη κατανόηση της διεργασίας η οποία θα αποτυπωθεί στο πρόγραμμα και την επιλογή μιας λογικής δομής η οποία αναδεικνύει κατά τον καλύτερο τρόπο τα ποιοτικά χαρακτηριστικά του προγράμματος τα οποία προαναφέρθηκαν.

∅ **Κωδικοποίηση προγράμματος (Coding):** Αποτελεί τη φάση εκείνη κατά την οποία ο σχεδιασμός του προγράμματος μετατρέπεται σε ένα ιεραρχημένο σύνολο εντολών κάποιας γλώσσας προγραμματισμού. Τα σημεία τα οποία είναι καθοριστικά για την **καλή κωδικοποίηση** του προγράμματος είναι:

1. Η δόμηση του προγράμματος: Ο τρόπος με τον οποίο ιεραρχούνται οι εντολές της επιλεγμένης γλώσσας προγραμματισμού, αποτελεί σημαντικό ποιοτικό χαρακτηριστικό της κωδικοποίησης. Κατά το Δομημένο Προγραμματισμό (Structured Programming), κάθε στοιχειώδης ενέργεια του προγράμματος αποτελεί ένα σχετικά ανεξάρτητο τμήμα (module) του προγράμματος. Με τη μέθοδο αυτή δεν έχει σημασία η σειρά κωδικοποίησης των τμημάτων αλλά η σειρά κλήσης τους. Αυτό έχει σαν συνέπεια την εύκολη κωδικοποίηση των λειτουργικών τμημάτων τα οποία προέκυψαν από τον σχεδιασμό, την περιγραφή αυτών των λειτουργικών τμημάτων σε ορισμένες θέσεις του προγράμματος με προφανείς ευεργετικές επιδράσεις στην πιθανή μεταβολή του προγράμματος.
2. Επιλογή της γλώσσας προγραμματισμού: Η επιλογή αυτή συνδέεται με κριτήρια τα οποία συχνά είναι αντιφατικά, όπως οι δυνατότητες της κάθε γλώσσας, οι απαιτήσεις του προγράμματος, οι δυνατότητες του Υλικού, η ευκολία προγραμματισμού, η τεχνογνωσία των προγραμματιστών, η συμβατότητα μιας γλώσσας με τις γλώσσες προγραμματισμού άλλων υπάρχοντων προγραμμάτων του Λογισμικού.
3. Ορθολογιστική χρήση της γλώσσας προγραμματισμού: Η ποιότητα της κωδικοποίησης δεν εξαρτάται τόσο από τη γνώση της γλώσσας προγραμματισμού αλλά κυρίως από την ορθολογιστική της χρήση. Αυτό αποτελεί άλλωστε και το χάρισμα ενός ικανού προγραμματιστή. Είναι γνωστό ότι σε μία γλώσσα προγραμματισμού η ίδια απαίτηση μπορεί να ικανοποιηθεί με διαφορετικούς τρόπους, χρησιμοποιώντας διαφορετικές εντολές. Ακόμη είναι γνωστό ότι εκτός από τις κλασσικές εντολές προσφέρονται από τους

κατασκευαστές Η/Υ και επεκτάσεις (extensions) οι οποίες συχνά βελτιώνουν την ταχύτητα προγραμματισμού. Το πρώτο το οποίο θα πρέπει να σημειωθεί είναι ότι γενικά θα πρέπει να αποφεύγονται προσφερόμενες «εξεζητημένες» εντολές οι οποίες καθιστούν δέσμιο το πρόγραμμα στο συγκεκριμένο τύπο Η/Υ. Το δεύτερο σημείο είναι ότι θα πρέπει να επιλέγονται οι εντολές οι οποίες εκτελούνται ταχύτερα, σεβόμενες τη δόμηση του προγράμματος. Το τρίτο σημείο είναι η έξυπνη χρήση εντολών.

4. Φιλικότητα των μηνυμάτων προς το χρήστη: Τα μηνύματα, τα οποία εμφανίζονται και καθοδηγούν το χρήστη κατά τη λειτουργία του προγράμματος, θα πρέπει να είναι γραμμένα στη γλώσσα του χρήστη, σαφή μη επιδεχόμενα διαφορετική ερμηνεία, απολύτως κατανοητά.

∅ **Δοκιμή Προγράμματος (Program test)**: Ο έλεγχος της σωστής λειτουργίας του προγράμματος, αν και φαινομενικά φαίνεται σαν μία εύκολη υπόθεση, στην πράξη αποτελεί ιδιαίτερα σύνθετη διαδικασία και αυτό διότι θα πρέπει να προβλεφθούν:

1. Οι πιθανές ενέργειες των χρηστών.
2. Η εμφάνιση ακραίων περιπτώσεων δεδομένων.
3. Ο φόρτος του συστήματος Η/Υ.
4. Η εξέλιξη του όγκου των δεδομένων στο μέλλον.
5. Η συμπεριφορά του προγράμματος σε ανώμαλες καταστάσεις (βλάβες Υλικού, διακοπές ηλεκτρικού ρεύματος κ.λπ.).

Η δοκιμή καλής λειτουργίας του προγράμματος μπορεί να γίνει με τους ακόλουθους τρόπους:

1. **Με χρήση δείγματος δεδομένων**: Επιλέγεται ένα ειδικό δείγμα (αρχεία, έντυπα κ.λπ.) το οποίο περιλαμβάνει κοινές και ακραίες περιπτώσεις που μπορούν να εμφανιστούν.
2. **Με παράλληλη λειτουργία προγράμματος και συμβατικής εργασίας**: Στη περίπτωση αυτή το πρόγραμμα λειτουργεί παράλληλα με τη συμβατική εργασία, την οποία πρόκειται να αντικαταστήσει, και συγκρίνονται τα τελικά αποτελέσματα. Με τη μέθοδο αυτή ο έλεγχος είναι πιο κοντά στη πραγματικότητα, όμως το μειονέκτημά της είναι ότι ο έλεγχος γίνεται για ορισμένες μόνο περιπτώσεις.
3. **Με συλλογικό έλεγχο από την ομάδα προγραμματισμού**: Ο δημιουργός του προγράμματος παρουσιάζει στην ομάδα προγραμματισμού το φάκελο προγραμματισμού και κάθε μέλος της ομάδας αφού τον μελετήσει προσπαθεί να εντοπίσει τις αδυναμίες του, μετατρέπόμενος σε «συνήγορο του διαβόλου». Επίσης αρμοδιότητα του δημιουργού είναι να συλλέγει τις επισημάνσεις οι οποίες προέρχονται από την ομάδα και διορθώνει το πρόγραμμα.
4. **Με δοκιμαστική χρήση από χρήστες**: Το πρόγραμμα παραδίδεται για δοκιμαστική λειτουργία σε χρήστες με διαφορετικό επίπεδο γνώσης του συστήματος. Συλλέγονται οι παρατηρήσεις για σφάλματα, παραλείψεις κ.λπ. και το πρόγραμμα διορθώνεται.

- Ø **Λειτουργία και συντήρηση προγράμματος:** Κατά τη λειτουργία του προγράμματος εμφανίζονται συχνά ανάγκες μετατροπής του, που οφείλονται σε:
1. Λογικά σφάλματα τα οποία εμφανίζονται κατά τη δοκιμή του προγράμματος.
 2. Απαιτήσεις προσαρμογής του προγράμματος σε μεταβολές του περιβάλλοντος του συστήματος.
 3. Βελτιωτικές προσαρμογές.

2.4.2 Γλώσσες Προγραμματισμού

Οι γλώσσες προγραμματισμού είναι τεχνητές γλώσσες με αυστηρά καθορισμένο συντακτικό που χρησιμοποιούνται για την ανάπτυξη λογισμικού. Μια γλώσσα προγραμματισμού επιτρέπει σε ένα προγραμματιστή να προδιαγράψει έναν υπολογισμό (computation) σε ένα επίπεδο αφαίρεσης, και να αφήσει σε ένα πρόγραμμα όπως είναι ο μεταγλωττιστής, ο διερμηνευτή κλπ, να υλοποιήσει τις συγκεκριμένες προδιαγραφές στην λεπτομερή μορφή που απαιτείται για την εκτέλεση από ένα υπολογιστή (Ben-Ari). Μια γλώσσα προγραμματισμού έχει σαν στόχο:

1. να επιτρέπει την εύκολη περιγραφή των υπολογισμών.
2. να εκμεταλλεύεται αποδοτικά την υπολογιστική ισχύ των μηχανών.

Με βάση το εύρος των προβλημάτων τα οποία μπορούν να αντιμετωπίσουν, οι γλώσσες προγραμματισμού χαρακτηρίζονται ως **γενικές ή εξειδικευμένες**, ενώ με βάση την ιστορική τους εξέλιξη ταξινομούνται σε "**γενιές**".

Έτσι έχει σαν στόχο:

1. να επιτρέπει την εύκολη περιγραφή των υπολογισμών.
2. να εκμεταλλεύεται αποδοτικά την υπολογιστική ισχύ των μηχανών.

Ø **Γλώσσες 1ης γενιάς ή γλώσσες μηχανής (machine languages):** Ένα πρόγραμμα σε γλώσσα μηχανής είναι μία ακολουθία δυαδικών ψηφίων που αποτελούν εντολές προς τον επεξεργαστή για στοιχειώδεις λειτουργίες. Οι εντολές αυτές είναι κατανοητές από τον υπολογιστή αλλά ακατανόητες από τον άνθρωπο καθώς απαιτούν βαθιά γνώση του Υλικού και της Αρχιτεκτονικής του υπολογιστή.

Ø **Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου (assembly languages):** Μια συμβολική γλώσσα ενώ έχει έννοια για τον άνθρωπο μετατρέπεται εσωτερικά από τον υπολογιστή στις αντίστοιχες ακολουθίες από «0» και «1». Το έργο της μετάφρασης αναλαμβάνει ένα ειδικό πρόγραμμα ο συμβολομεταφραστής. Οι εντολές σε συμβολική γλώσσα αποτελούνται από συμβολικά ονόματα που αντιστοιχούν σε εντολές σε γλώσσα μηχανής.

Τα **μειονεκτήματα** των συμβολικών γλωσσών είναι τα εξής:

1. Παραμένουν συνδεδεμένες στενά με την αρχιτεκτονική του υπολογιστή.

2. Δεν διαθέτουν εντολές πιο σύνθετων λειτουργιών οδηγώντας έτσι σε μακροσκελή προγράμματα που είναι δύσκολο να γραφούν και κυρίως να συντηρηθούν.
3. Δεν μπορούν να μεταφερθούν σε άλλον διαφορετικό υπολογιστή ακόμα και του ίδιου κατασκευαστή.

Ø **Γλώσσες υψηλού επιπέδου 3^{ης} γενιάς:** Οι γλώσσες υψηλού επιπέδου χρησιμοποιούν ως εντολές απλές λέξεις της αγγλικής γλώσσας ακολουθώντας αυστηρούς κανόνες σύνταξης. Οι εντολές αυτές μεταφράζονται από τον ίδιο υπολογιστή σε εντολές σε γλώσσας μηχανής. Οι κυριότερες γλώσσες υψηλού επιπέδου είναι οι εξής:

1. **FORTRAN:** Αναπτύχθηκε το 1957 ως γλώσσα κατάλληλη για την επίλυση μαθηματικών και επιστημονικών προβλημάτων και μετά από αλλαγές, προσθήκες και βελτιώσεις χρησιμοποιείται ακόμη και σήμερα για επιστημονικές εφαρμογές. Η FORTRAN υστερεί στη διαχείριση αρχείων δεδομένων και γενικότερα αλφαριθμητικών πληροφοριών.
2. **COBOL:** Το 1960 αναπτύχθηκε ως γλώσσα κατάλληλη για ανάπτυξη εμπορικών συναλλαγών. Η COBOL καθιερώθηκε ως πρότυπο και χρησιμοποιήθηκε από πολλές επιχειρήσεις και από όλη τη δημόσια διοίκηση. Πολλές εφαρμογές βρίσκονται σε χρήση ακόμη και σήμερα.
3. **ALGOL:** Επηρέασε ιδιαίτερα τον προγραμματισμό και τις επόμενες γλώσσες. Αναπτύχθηκε με σκοπό τη δημιουργία προγραμμάτων γενικής φύσεως που να μη συνδέονται με συγκεκριμένες εφαρμογές.
4. **PL/1:** Προσπάθησε ανεπιτυχώς να αντικαταστήσει την FORTRAN και την COBOL.
5. **LISP** και **PROLOG:** Αυτές οι δυο διαφορετικές γλώσσες αναπτύχθηκαν στο χώρο της τεχνητής νοημοσύνης.
6. **BASIC:** Αναπτύχθηκε ως γλώσσα για την εκπαίδευση αρχαρίων στον προγραμματισμό. Η ανάπτυξη των μικροϋπολογιστών και οι συνεχείς εκδόσεις της την κατέστησαν τη δημοφιλέστερη, ίσως, γλώσσα στους προσωπικούς υπολογιστές.
7. **PASCAL:** Παρουσιάστηκε το 1970 και στηρίχθηκε στην ALGOL. Είναι γλώσσα γενικής χρήσης, κατάλληλη για εκπαίδευση αρχαρίων αλλά και για την δημιουργία ισχυρών προγραμμάτων κάθε τύπου. Είναι κατάλληλη για δημιουργία δομημένων προγραμμάτων. Γνωρίζει τεράστια ανάπτυξη στο χώρο των μικροϋπολογιστών.

8. **C:** Γνώρισε μεγάλη διάδοση και χρησιμοποιήθηκε για την ανάπτυξη του λειτουργικού συστήματος UNIX. Είναι γλώσσα κατάλληλη για την ανάπτυξη δομημένων εφαρμογών αλλά και με πολλές δυνατότητες γλώσσας χαμηλού επιπέδου. Έχει εξελιχθεί στη γλώσσα **C++** που είναι αντικειμενοστραφής.
9. **JAVA:** Τα τελευταία χρόνια χρησιμοποιείται ειδικά για προγραμματισμό στο Διαδίκτυο. Η JAVA είναι μια αντικειμενοστραφής γλώσσα που αναπτύχθηκε με σκοπό την ανάπτυξη εφαρμογών που θα εκτελούνται σε κατακευματισμένα περιβάλλοντα, δηλαδή σε διαφορετικούς υπολογιστές που είναι συνδεδεμένοι στο Διαδίκτυο. Τα προγράμματα αυτά μπορούν να εκτελούνται χωρίς αλλαγές από διαφορετικούς υπολογιστές, προσωπικούς ή μεγάλα συστήματα με διαφορετικά λειτουργικά συστήματα.

Η εμφάνιση των γραφικών περιβαλλόντων εργασίας οδήγησε στην εμφάνιση νέων γλωσσών ή νέων εκδόσεων παλαιότερων γλωσσών. Οι πιο διαδεδομένες γλώσσες προγραμματισμού σε γραφικό περιβάλλον για προσωπικούς υπολογιστές είναι η Visual Basic, η Visual C++ και η JAVA.

Τα **πλεονεκτήματα** των γλωσσών υψηλού επιπέδου σε σχέση με τις συμβολικές είναι τα εξής:

1. Ο φυσικός και πιο «ανθρώπινος» τρόπος έκφρασης των προβλημάτων.
2. Η ανεξαρτησία από τον τύπο του υπολογιστή και η δυνατότητα μεταφερσιμότητας που αυτή συνεπάγεται.
3. Η ευκολία εκμάθησης και εκπαίδευσης.
4. Η ευκολία διόρθωσης λαθών και συντήρησης προγραμμάτων.

Συνολικά οι γλώσσες υψηλού επιπέδου ελάττωσαν σημαντικά το κόστος και το χρόνο παραγωγής νέων προγραμμάτων αφού λιγότεροι προγραμματιστές μπορούν σε λιγότερο χρόνο να αναπτύξουν προγράμματα που χρησιμοποιούνται στους περισσότερους ηλεκτρονικούς υπολογιστές.

∅ Γλώσσες Υψηλού Επιπέδου 4^{ης} γενιάς: Οι γλώσσες 4^{ης} γενιάς σε αντίθεση με τις γλώσσες 3^{ης} γενιάς, είναι γλώσσες εφοδιασμένες με εργαλεία προγραμματισμού που αποκρύπτουν πολλές λεπτομέρειες από τις τεχνικές υλοποίησης και με αυτά ο χρήστης μπορεί να επιλύει μόνος του μικρά προβλήματα εφαρμογών. Ο χρήστης ενός υπολογιστή μπορεί σχετικά εύκολα να υποβάλλει ερωτήσεις στο σύστημα ή να αναπτύσσει εφαρμογές που αποκτούν πληροφορίες από βάσεις δεδομένων και να καθορίζει τον ακριβή τρόπο εμφάνισης αυτών των πληροφοριών.

Αυτή τη στιγμή χρησιμοποιούνται μερικές εκατοντάδες γλώσσες. Η επιλογή της γλώσσας για την ανάπτυξη μιας εφαρμογής εξαρτάται:

1. Από το είδος της εφαρμογής.
2. Το υπολογιστικό περιβάλλον στο οποίο θα εκτελεστεί.
3. Τα προγραμματιστικά περιβάλλοντα, τα οποία διαθέτουμε.

4. Τις γνώσεις του προγραμματιστή.

∅ **Φυσικές και Τεχνητές Γλώσσες:** Οι φυσικές γλώσσες χρησιμοποιούνται για την επικοινωνία των ανθρώπων ενώ οι τεχνητές χρησιμοποιούνται για την επικοινωνία μεταξύ του ανθρώπου και της μηχανής. Μία βασική διαφορά μεταξύ των φυσικών και των τεχνητών γλωσσών είναι η δυνατότητα εξέλιξης τους. Οι φυσικές γλώσσες εξελίσσονται διαρκώς ενώ οι τεχνητές χαρακτηρίζονται από στασιμότητα. Μία γλώσσα μπορεί να προσδιοριστεί από: **1)** Το αλφάβητό της. **2)** Το λεξιλόγιό της. **3)** Τη γραμματική της. **4)** Τη σημασιολογία της.

2.4.3 Τεχνικές Σχεδιάσεις Προγραμμάτων

∅ **Ιεραρχική Σχεδίαση Προγραμμάτων:** Η ιεραρχική σχεδίαση προγραμμάτων ή διαδικασία σχεδιασμού «από πάνω προς τα κάτω» χρησιμοποιεί τη στρατηγική της συνεχούς διαίρεσης του προβλήματος σε υποπροβλήματα. Για την υποβοήθεια της ιεραρχικής σχεδίασης χρησιμοποιούνται διάφορες διαγραμματικές τεχνικές.

∅ **Τμηματικός Προγραμματισμός:** Η ιεραρχική σχεδίαση προγράμματος υλοποιείται με το τμηματικό προγραμματισμό. Μετά την ανάλυση του προβλήματος σε υποπροβλήματα, κάθε υποπρόβλημα αποτελεί ανεξάρτητη «οντότητα» που γράφεται ξεχωριστά από τα υπόλοιπα τμήματα προγράμματος. Ο τμηματικός προγραμματισμός μειώνει τα λάθη και επιτρέπει την ευκολότερη παρακολούθηση, κατανόηση και συντήρηση του προγράμματος από τρίτους.

∅ **Δομημένος Προγραμματισμός:** Ο δομημένος προγραμματισμός αναπτύχθηκε από την ανάγκη να υπάρχει μία κοινή μεθοδολογία στην ανάπτυξη προγραμμάτων και στην ανάγκη να μειωθεί η χρήση εντολών GOTO, δηλαδή εντολών που αλλάζουν τη ροή του προγράμματος. Είναι μία μεθοδολογία σύνταξης προγραμμάτων που έχει σκοπό να βοηθήσει τον προγραμματιστή στην ανάπτυξη σύνθετων προγραμμάτων, να μειώσει τα λάθη, να εξασφαλίσει την εύκολη κατανόηση των προγραμμάτων και να διευκολύνει τις διορθώσεις και τις αλλαγές σε αυτά. Επίσης στηρίζεται στη χρήση τριών και μόνο στοιχειωδών λογικών δομών, τη δομή ακολουθίας, τη δομή της επιλογής και τη δομή της επανάληψης. Ο όρος δομημένος προγραμματισμός περιέχει τόσο την ιεραρχική σχεδίαση όσο και τον τμηματικό προγραμματισμό. Τα πλεονεκτήματα του δομημένου προγραμματισμού είναι τα εξής:

- 1) Δημιουργία απλούστερων προγραμμάτων.
- 2) Άμεση μεταφορά αλγορίθμων σε προγράμματα.
- 3) Διευκόλυνση ανάλυσης του προγράμματος.
- 4) Περιορισμός των λαθών κατά την ανάπτυξη του προγράμματος.
- 5) Διευκόλυνση στην ανάγνωση και κατανόηση του προγράμματος από τρίτους.
- 6) Ευκολότερη διόρθωση και συντήρηση

∅ **Αντικειμενοστραφής Προγραμματισμός:** Είναι μία μέθοδος για την δόμηση ενός προγράμματος σε ιεραρχικά οργανωμένες τάξεις που περιγράφουν τα

δεδομένα και τις λειτουργίες αντικειμένων, τα οποία μπορούν να αλληλεπιδρούν με άλλα αντικείμενα. Η αντικειμενοστραφής σχεδίαση έχει ως πρωτεύοντα δομικά στοιχεία ενός προγράμματος τα δεδομένα, από τα οποία με κατάλληλες μορφοποιήσεις δημιουργούνται τα αντικείμενα. Αυτή η σχεδίαση επιφέρει καλύτερα αποτελέσματα, αφού τα προγράμματα που δημιουργούνται είναι περισσότερο ευέλικτα και επαναχρησιμοποιούμενα.

∅ **Παράλληλος Προγραμματισμός:** Μία μορφή προγραμματισμού που αναπτύσσεται τελευταία είναι ο παράλληλος προγραμματισμός. Πρόσφατα εμφανίστηκαν υπολογιστές που διαθέτουν περισσότερους από έναν επεξεργαστές. Οι επεξεργαστές αυτοί μοιράζονται την ίδια μνήμη και λειτουργούν παράλληλα εκτελώντας διαφορετικές εντολές του ίδιου προγράμματος. Εμφανίζονται θεωρητικά να επιτυγχάνουν ασύλληπτες ταχύτητες επεξεργασίας. Για να εκμεταλλευτούν όμως τις δυνατότητες που προσφέρει η αρχιτεκτονική τους, πρέπει το πρόβλημα να διαιρεθεί σε τμήματα που εκτελούνται παράλληλα και στη συνέχεια να προγραμματιστεί σε ένα προγραμματιστικό περιβάλλον που να επιτρέπει τον παράλληλο προγραμματισμό. Μία γλώσσα προγραμματισμού που υποστηρίζει παράλληλο προγραμματισμό είναι η OCCAM.

2.5 Μοντελοποιώντας με χρήση της UML

Η Unified Modeling Language (UML) είναι μία γλώσσα που χρησιμοποιείται για προδιαγραφές, αναπαράσταση με οπτικό τρόπο (visualizing), δημιουργία και τεκμηρίωση των τμημάτων των συστημάτων λογισμικού, καθώς και για μοντελοποίηση εταιρικών και άλλων συστημάτων που δεν αφορούν λογισμικό. Αποτελεί ένα συνδυασμό των καλύτερων πρακτικών, οι οποίες ήδη έχουν αποδείξει πόσο επιτυχημένες ήταν στη μοντελοποίηση μεγάλων και σύνθετων συστημάτων. Η UML έχει αναπτυχθεί από τη Rational Software και τους εταίρους της. Επίσης πολλές εταιρίες την έχουν ενσωματώσει ως πρότυπο στη διαδικασία ανάπτυξής τους και στα προϊόντα τους, τα οποία καλύπτουν περιοχές όπως η business modeling, η διαχείριση απαιτήσεων, η ανάλυση και ο σχεδιασμός, ο προγραμματισμός και ο έλεγχος.

2.5.1 Κίνητρα για τον ορισμό της UML – Το πεδίο εφαρμογής

Η ανάπτυξη ενός μοντέλου για ένα σύστημα λογισμικού που θα μπορέσει να χρησιμοποιηθεί στη βιομηχανία, προτού δομηθεί ή επανασχεδιαστεί το σύστημα, είναι τόσο απαραίτητο όσο το να υπάρχουν τα αρχιτεκτονικά σχέδια για ένα μεγάλο κτίσμα. Τα καλά μοντέλα είναι απαραίτητα για την επικοινωνία ανάμεσα στις ομάδες ενός έργου και επίσης για να υπάρχει μία ορθή αρχιτεκτονική του συστήματος.

Όσο αυξάνει η πολυπλοκότητα των συστημάτων, αυξάνει και η σημασία της ύπαρξης καλών τεχνικών μοντελοποίησης. Στην περίπτωση των όλο και πιο πολύπλοκων συστημάτων η οπτική αναπαράσταση και η μοντελοποίηση γίνονται απαραίτητες. Η UML είναι μία καλά ορισμένη και γενικά αποδεκτή απάντηση για την ανάγκη αυτή.

Όσο αυξάνει για πολλές εταιρίες η στρατηγική αξία του λογισμικού, η βιομηχανία αναζητά νέες τεχνικές για αυτοματοποίηση της παραγωγής του λογισμικού. Αναζητούμε τεχνικές για να βελτιωθεί η ποιότητα και να μειωθεί το κόστος. Οι τεχνικές αυτές περιλαμβάνουν την τεχνολογία των συνιστωσών, τον οπτικό προγραμματισμό (visual programming), τα patterns και τα πλαίσια (frameworks). Επίσης αναζητούμε τεχνικές για να διαχειριζόμαστε την πολυπλοκότητα των συστημάτων, όσο αυξάνουν σε κλίμακα. Συγκεκριμένα, αναγνωρίζουμε την ανάγκη για επίλυση επαναλαμβανόμενων προβλημάτων αρχιτεκτονικής, όπως η φυσική κατανομή, η συνδρομικότητα (concurrency), η ύπαρξη αντιγράφων (replication), η ασφάλεια, η κατανομή φορτίου και η ανοχή σε σφάλματα.

Η πολυπλοκότητα μπορεί να διαφέρει σε διάφορα πεδία εφαρμογής και φάσεις επεξεργασίας. Ένα από τα βασικά κίνητρα κατά την ανάπτυξη της UML ήταν η δημιουργία ενός συνόλου σημασιολογίας και συμβολισμού που να αντιμετωπίζει επαρκώς όλα τα επίπεδα αρχιτεκτονικής πολυπλοκότητας, σε όλα τα πεδία.

Πριν τη UML δεν υπήρχε μία γλώσσα μοντελοποίησης που να έχει ένα σαφές προβάδισμα. Οι χρήστες επέλεγαν μεταξύ πολλών παρόμοιων γλωσσών, οι οποίες είχαν ελάχιστες διαφορές στην ολική εκφραστικότητά τους. Οι περισσότερες γλώσσες μοντελοποίησης είχαν ένα σύνολο κοινά αποδεκτών εννοιών, οι οποίες εκφράζονταν λίγο διαφορετικά από γλώσσα σε γλώσσα. Αυτή η έλλειψη συμφωνίας αποθάρρυνε τους νέους χρήστες από το να χρησιμοποιήσουν την αντικειμενοστραφή τεχνολογία. Οι χρήστες ήθελαν να υιοθετήσει η βιομηχανία μία – ή ελάχιστες – κοινά υποστηριζόμενες και αποδεκτές γλώσσες μοντελοποίησης, οι οποίες θα ήταν κατάλληλες για γενική χρήση.

Το αυξανόμενο κόστος χρήσης και υποστήριξης πολλών γλωσσών μοντελοποίησης έδωσε το κίνητρο σε πολλές από τις εταιρίες που προωθούν το αντικειμενοστραφές μοντέλο να υποστηρίξουν και να προωθήσουν τη UML. Ενώ η UML δεν εγγυάται την επιτυχία ενός έργου, βελτιώνει πολλά πράγματα. Για παράδειγμα, μειώνει σημαντικά το κόστος εκμάθησης και χρησιμοποίησης νέων εργαλείων όταν γίνονται μεταφορές από έργο σε έργο ή από εταιρία σε εταιρία. Επίσης παρέχει τη δυνατότητα ολοκλήρωσης ανάμεσα σε νέα εργαλεία, νέες διαδικασίες και νέες περιοχές.

Πρώτον, πρέπει να τονιστεί ότι η UML συνδυάζει τις έννοιες των τεχνικών Booch, OMT και OOSE. Το αποτέλεσμα είναι μία, κοινή και ευρέως χρησιμοποιήσιμη γλώσσα μοντελοποίησης για τους χρήστες αυτών και άλλων μεθόδων.

Δεύτερον, η UML προχωράει πιο πέρα το ζήτημα του τι μπορεί να κάνει κανείς με τις μεθόδους αυτές. Για παράδειγμα, οι συγγραφείς της UML έδωσαν σημασία στη μοντελοποίηση ταυτόχρονων και κατανεμημένων συστημάτων, ώστε να επιβεβαιώσουν ότι η UML θα μπορεί να αντιμετωπίσει με ευκολία αυτά τα πεδία εφαρμογών.

Τρίτον, η UML εστιάζει σε μία πρότυπη γλώσσα μοντελοποίησης, όχι σε μία πρότυπη διαδικασία μοντελοποίησης. Για πολλούς λόγους, οι προσπάθειες έχουν επικεντρωθεί πρώτα στη δημιουργία ενός κοινού μεταμοντέλου (που θα ενοποιεί τη σημασιολογία) και στη συνέχεια στη δημιουργία ενός κοινού συμβολισμού (που θα παρέχει την άποψη των ατόμων για τη σημασιολογία). Οι συγγραφείς της UML προωθούν μία διαδικασία ανάπτυξης που προσανατολίζεται σε 1. περιπτώσεις χρήσης, 2. εξαρτάται από την αρχιτεκτονική και 3. είναι επαναληπτική και αυξητική.

Η UML προδιαγράφει μία γλώσσα μοντελοποίησης που περιλαμβάνει τη συμφωνία της αντικειμενοστραφούς κοινότητας σχετικά με βασικές έννοιες μοντελοποίησης, επιτρέπει να εκφράζονται οι αποκλίσεις με χρήση των μηχανισμών επέκτασης και παρέχει τα ακόλουθα:

1. Σημασιολογία και συμβολισμό για να αντιμετωπίζεται ένα ευρύ φάσμα σύγχρονων θεμάτων μοντελοποίησης με άμεσο και οικονομικό τρόπο.
2. Σημασιολογία για να αντιμετωπιστούν ορισμένα από τα αναμενόμενα μελλοντικά ζητήματα μοντελοποίησης, τα οποία σχετίζονται με την τεχνολογία συνιστωσών, τους καταναμημένους υπολογισμούς, τα πλαίσια και την εκτελεσιμότητα.
3. Μηχανισμούς επέκτασης ώστε συγκεκριμένα έργα να μπορούν να επεκτείνουν το μεταμοντέλο για την εφαρμογή τους με χαμηλό κόστος. Δε θέλουμε να αλλάζουν οι χρήστες απ ευθείας το μεταμοντέλο της UML.
4. Μηχανισμούς επέκτασης, ώστε να είναι δυνατό να ενσωματωθούν οι μελλοντικές προσεγγίσεις μοντελοποίησης μέσα στη UML.
5. Σημασιολογία για να διευκολύνεται η ανταλλαγή μοντέλων σε διαφορετικά εργαλεία.
6. Σημασιολογία που να καθορίζει τη διασύνδεση με αποθηκευτικούς χώρους, για διαμοιρασμό και αποθήκευση των στοιχείων μοντελοποίησης.

2.5.2 Κύρια στοιχεία και χαρακτηριστικά της UML

Καθώς οι συγγραφείς της UML σχεδιάζανε, είχαν σαν στόχο της προσπάθειας να διατηρηθεί απλή, να εκλείψουν τα στοιχεία που υπήρχαν στις τρεις μεθοδολογίες και δε λειτουργούσαν στην πράξη, να προστεθούν στοιχεία από άλλες μεθοδολογίες, τα οποία ήταν πιο αποδοτικά και να ανακαλυφθούν νέα στοιχεία μόνο στις περιπτώσεις που δεν υπήρχε ήδη λύση. Έχοντας αυτό κατά νου ήταν ιδιαίτερα προσεκτικοί όταν έπρεπε να προσθέσουν ένα νέο στοιχείο, καθώς δε θέλανε να κάνουν τη UML πολύπλοκη άνευ λόγου.

Η επιλογή σχετικά με το ποια μοντέλα και ποια διαγράμματα θα δημιουργηθούν επηρεάζει σημαντικά τον τρόπο προσέγγισης και επίλυσης του προβλήματος. Το βασικό στοιχείο στην προσπάθεια επικοινωνίας και μάθησης είναι η έννοια της αφάιρεσης, να εστιάζουμε δηλαδή σε σχετικές λεπτομέρειες, ενώ αγνοούμε τις υπόλοιπες. Ως συνέπειες αυτού είναι:

1. Κάθε σύνθετο σύστημα προσεγγίζεται καλύτερα χρησιμοποιώντας ένα μικρό σύνολο ανεξαρτήτων όψεων του συστήματος. Δεν αρκεί μία όψη.
2. Κάθε μοντέλο μπορεί να εκφραστεί σε διαφορετικά επίπεδα πιστότητας.
3. Τα καλύτερα μοντέλα συνδέονται με την πραγματικότητα.

Οι βασικοί στόχοι σχεδιασμού της UML είναι οι ακόλουθοι:

- ∅ Να παρέχει στους χρήστες μία έτοιμη προς χρήση, οπτική γλώσσα μοντελοποίησης, ώστε αυτοί να μπορούν να αναπτύξουν και να ανταλλάξουν κατανοητά μοντέλα.
- ∅ Να παρέχει μηχανισμούς επέκτασης και εξειδίκευσης, ώστε να είναι δυνατή η επέκταση των βασικών εννοιών.

- Ø Να υποστηρίζει προδιαγραφές οι οποίες είναι ανεξάρτητες από συγκεκριμένες γλώσσες προγραμματισμού ή διαδικασίες ανάπτυξης.
- Ø Να παρέχει μία τυπική βάση για την κατανόηση της γλώσσας μοντελοποίησης.
- Ø Να ενθαρρύνει την ανάπτυξη της αγοράς αντικειμενοστραφών εργαλείων.
- Ø Να υποστηρίζει υψηλότερου επιπέδου έννοιες ανάπτυξης, όπως συνιστώσες, συνεργασίες, πλαίσια και patterns.
- Ø Να ενοποιεί τις καλύτερες προσεγγίσεις.

2.5.3 Διαγράμματα στατικής δομής της UML

Τα διαγράμματα κλάσης (class diagrams) παρουσιάζουν τη στατική δομή του μοντέλου και συγκεκριμένα τα υπάρχοντα στοιχεία (όπως κλάσεις και τύποι), την εσωτερική τους δομή και τη σχέση τους με άλλα στοιχεία και δεν παρουσιάζουν χρονική πληροφορία, αν και μπορούν να περιέχουν εμφανίσεις στοιχείων που έχουν ή που περιγράφουν χρονική συμπεριφορά. Από την άλλη μεριά, ένα διάγραμμα αντικειμένων (object diagram) παρουσιάζει στιγμιότυπα συμβατά με ένα συγκεκριμένο διάγραμμα κλάσης.

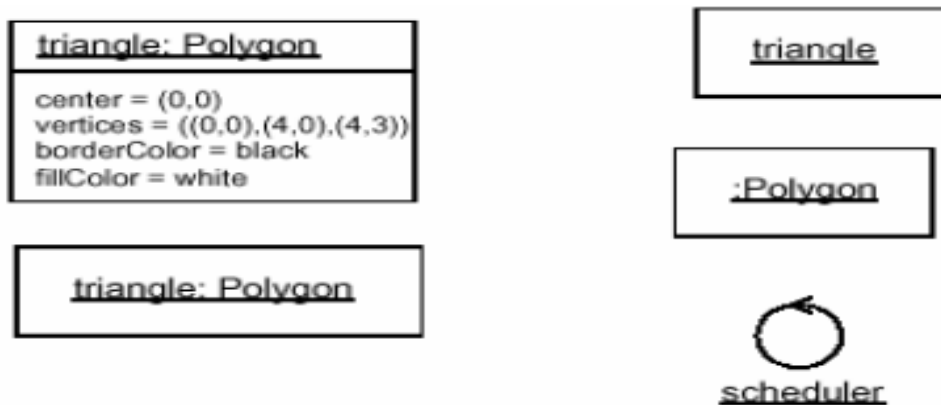
- Ø **Διάγραμμα κλάσης – Class diagram:** Ένα διάγραμμα κλάσης είναι ένας γράφος από στοιχεία classifier που συνδέονται με διάφορες στατικές σχέσεις. Σημειώστε ότι ένα διάγραμμα κλάσης μπορεί να περιέχει διασυνδέσεις, πακέτα, σχέσεις ακόμα και στιγμιότυπα όπως αντικείμενα και συνδέσμους. Ίσως ένα καλύτερο όνομα να ήταν το «διάγραμμα στατικής δομής» αλλά το «διάγραμμα κλάσης» είναι μικρότερο και έχει ήδη γίνει αποδεκτό. Ένα διάγραμμα κλάσης είναι η γραφική όψη του μοντέλου στατικής δομής. Τα διαγράμματα αυτά αναπαριστούν μία συλλογή (στατικών) δηλωτικών στοιχείων μοντελοποίησης, όπως κλάσεις, διασυνδέσεις και οι σχέσεις τους που συνδέονται ως ένας γράφος και μπορούν να οργανωθούν σε πακέτα είτε με τα υποκείμενα μοντέλα είτε ως χωριστά πακέτα που δημιουργούνται πάνω από τα πακέτα του υποκείμενου μοντέλου.
- Ø **Διάγραμμα αντικειμένων – Object diagram:** Ένα διάγραμμα αντικειμένων είναι ένας γράφος στιγμιότυπων, περιλαμβανομένων αντικειμένων και τιμών δεδομένων. Μπορούμε να πούμε πως πρόκειται για ένα στιγμιότυπο ενός διαγράμματος κλάσης, που παρουσιάζει ένα στιγμιότυπο της λεπτομερούς κατάστασης ενός συστήματος σε μία συγκεκριμένη χρονική στιγμή. Η χρήση των διαγραμμάτων αντικειμένων είναι σχετικά περιορισμένη. Τα διαγράμματα κλάσεων μπορούν να περιέχουν αντικείμενα, συνεπώς ένα διάγραμμα κλάσεων χωρίς κλάσεις είναι ένα διάγραμμα αντικειμένων.

Ιδιαίτερη σημασία έχουν για την UML και τα διαγράμματά της, οι παρακάτω έννοιες:

- Ø **Κλάση – Class:** Μία κλάση είναι ένας περιγραφέας για ένα σύνολο αντικειμένων με κοινή δομή, συμπεριφορά και σχέσεις. Το μοντέλο ενδιαφέρεται με την περιγραφή των κανόνων που περιγράφουν την κλάση. Η UML παρέχει συμβολισμό για τον ορισμό της κλάσης και τον καθορισμό των ιδιοτήτων της, καθώς και για τη χρησιμοποίηση των κλάσεων με διάφορους τρόπους. Οι κλάσεις ορίζονται στα διαγράμματα κλάσεων και χρησιμοποιούνται στα περισσότερα άλλα διαγράμματα. Είναι γνωστό πως μία κλάση αναπαριστά μία έννοια του μοντελοποιούμενου συστήματος και έχει δομή δεδομένων και συμπεριφορά,

καθώς και σχέσεις προς άλλα στοιχεία. Επίσης σχεδιάζεται ως ένα ορθογώνιο με τρία τμήματα τα οποία διαχωρίζονται από οριζόντιες γραμμές. Το πάνω τμήμα, το τμήμα ονόματος, περιέχει το όνομα της κλάσης και άλλες γενικές ιδιότητές της (συμπεριλαμβανομένων του στερεοτύπου). Το μεσαίο τμήμα περιέχει μία λίστα ιδιοχαρακτηριστικών, ενώ το τελευταίο τμήμα περιέχει μία λίστα πράξεων. Εξ ορισμού, μία κλάση που εμφανίζεται σε ένα πακέτο θεωρείται ότι έχει οριστεί σε αυτό.

Ø **Αντικείμενο – Object:** Ένα αντικείμενο αναπαριστά ένα συγκεκριμένο στιγμιότυπο μίας κλάσης. Έχει δική του ταυτότητα και τιμές ιδιοχαρακτηριστικών. Ο συμβολισμός του αντικειμένου προέρχεται από το συμβολισμό της κλάσης υπογραμμίζοντας τα στοιχεία επιπέδου στιγμιότυπου. Ένα αντικείμενο παρουσιάζεται με ένα ορθογώνιο με δύο τμήματα.



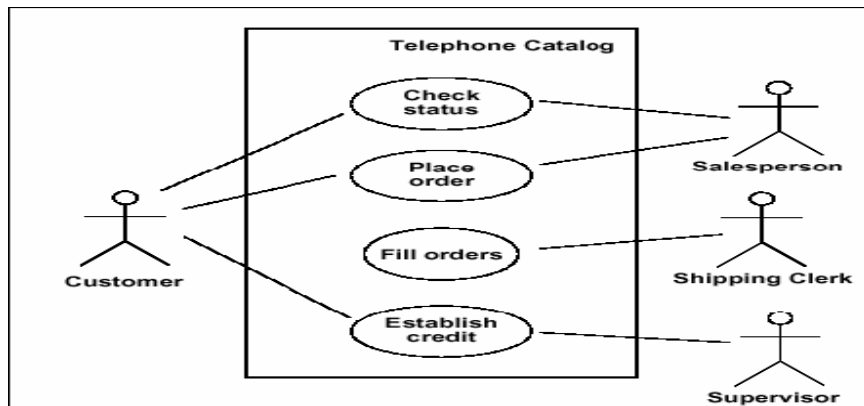
Σχήμα 2.4 Παράδειγμα Αντικειμένων

2.5.4 Διάγραμμα περιπτώσεων χρήσης – Use Case Diagram

Ένα διάγραμμα περιπτώσεων χρήσης παρουσιάζει τη σχέση ανάμεσα στους actors και τις περιπτώσεις χρήσης ενός συστήματος. Τα διαγράμματα αυτά παρουσιάζουν τους actors και τις περιπτώσεις χρήσης ενός συστήματος μαζί με τις σχέσεις μεταξύ τους. Οι περιπτώσεις χρήσης αναπαριστούν λειτουργικότητα ενός συστήματος ή ενός classifier, όπως ένα υποσύστημα ή μία κλάση, όπως παρουσιάζεται σε όσους αλληλεπιδρούν με το σύστημα ή τον classifier, ενώ βρίσκονται εκτός του.

Ένα διάγραμμα περιπτώσεων χρήσης είναι ένας γράφος από actors, ένα σύνολο περιπτώσεων χρήσης, πιθανώς ορισμένες διασυνδέσεις και τις σχέσεις ανάμεσα σε αυτά τα στοιχεία. Οι σχέσεις είναι συσχετίσεις ανάμεσα στους actors και τις περιπτώσεις χρήσης, γενικεύσεις ανάμεσα σε actors και γενικεύσεις, επεκτάσεις (extends) και περιλήψεις (includes) ανάμεσα σε περιπτώσεις χρήσης. Οι περιπτώσεις χρήσης προαιρετικά μπορούν να περικλείονται σε ένα ορθογώνιο, το οποίο αναπαριστά τα όρια του περιέχοντος συστήματος ή του classifier.

Παράδειγμα:



Σχήμα 2.5 Διάγραμμα περιπτώσεων χρήσης.

Μία περίπτωση χρήσης (use case) είναι ένα είδος classifier που αναπαριστά μία συνεπή μονάδα λειτουργικότητας που παρέχεται από το σύστημα, ένα υποσύστημα, ή μία κλάση, όπως παρουσιάζεται από ακολουθίες μηνυμάτων που ανταλλάσσονται ανάμεσα στο σύστημα και έναν ή περισσότερους εξωτερικούς χρήστες (οι οποίοι ονομάζονται actors) μαζί με τις πράξεις που θα πραγματοποιηθούν από το σύστημα. Μία περίπτωση χρήσης παρουσιάζεται με μία έλλειψη η οποία περιέχει το όνομα της περίπτωσης χρήσης. Πάνω από το όνομα μπορεί να τοποθετηθεί μία προαιρετική λέξη κλειδί στερεοτύπου, ενώ κάτω από το όνομα συμπεριλαμβάνεται μία λίστα ιδιοτήτων. Ως classifier, η περίπτωση χρήσης μπορεί να έχει επίσης τμήματα που παρουσιάζουν ιδιοχαρακτηριστικά και πράξεις.

Ένα σημείο επέκτασης (extension point) είναι μία αναφορά σε μία περιοχή μέσα στην περίπτωση χρήσης στην οποία μπορούν να εισαχθούν ακολουθίες πράξεων από άλλες περιπτώσεις χρήσης. Κάθε σημείο επέκτασης έχει ένα μοναδικό όνομα μέσα στην περίπτωση χρήσης και μία περιγραφή της περιοχής μέσα στη συμπεριφορά της περίπτωσης χρήσης. Τα σημεία επέκτασης μπορούν να παρουσιάζονται σε ένα τμήμα της περίπτωσης χρήσης με την επικεφαλίδα extension points. Η περιγραφή της περιοχής του σημείου επέκτασης παρουσιάζεται στην κατάλληλη μορφή, συνήθως ως απλό κείμενο, αλλά μπορεί επίσης να παρουσιαστεί με άλλες μορφές, όπως το όνομα μίας κατάστασης σε μία μηχανή κατάστασης ή με μία pre-condition ή post-condition.

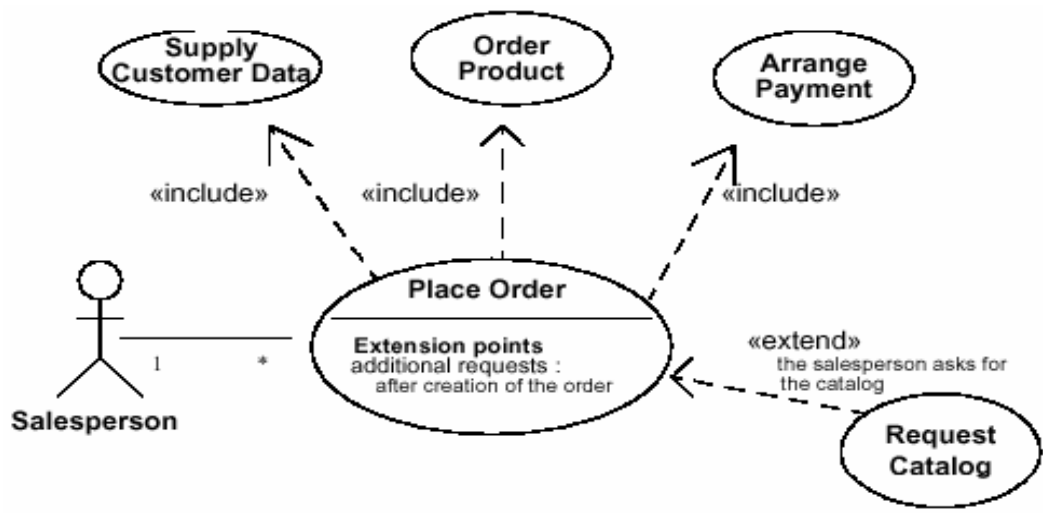
Βασικές έννοιες των Διαγραμμάτων Περιπτώσεων Χρήσης είναι:

∅ **Actor:** Ένας actor ορίζει ένα συνεπές σύνολο ρόλων τους οποίους μπορούν να «παιξουν» οι χρήστες μίας οντότητας όταν αυτοί αλληλεπιδρούν με την οντότητα. Ένας actor μπορεί να θεωρηθεί ότι παίζει ένα ξεχωριστό ρόλο σε σχέση με κάθε περίπτωση χρήσης με την οποία επικοινωνεί. Το κλασικό εικονίδιο στερεοτύπου για έναν actor είναι ένα σχήμα που δείχνει ένα «ανθρωπάκι» με το όνομα του actor κάτω από το σχήμα. Ο actor μπορεί ακόμα να αναπαρασταθεί ως ένα ορθογώνιο κλάσης, με τη λέξη κλειδί «actor» και το συνήθη συμβολισμό για τα τμήματα.

Ø **Σχέσεις περιπτώσεων χρήσης – Use Case Relationships:** Υπάρχουν αρκετές πρότυπες σχέσεις ανάμεσα στις περιπτώσεις χρήσης ή ανάμεσα σε actors και σε περιπτώσεις χρήσης:

1. Συσχέτιση (association) – Η συμμετοχή ενός actor σε μία περίπτωση χρήσης, δηλαδή τα στιγμιότυπα του actor και τα στιγμιότυπα της περίπτωσης χρήσης επικοινωνούν μεταξύ τους. Είναι η μόνη σχέση ανάμεσα σε actors και σε περιπτώσεις χρήσης. Μία συσχέτιση ανάμεσα σε έναν actor και μία περίπτωση χρήσης παρουσιάζεται ως μία γραμμή ανάμεσα σε έναν actor και μία περίπτωση χρήσης.
2. Επέκταση (extend) – Μία σχέση επέκτασης από την περίπτωση χρήσης A στην περίπτωση χρήσης B δηλώνει ότι ένα στιγμιότυπο της περίπτωσης χρήσης B μπορεί να αυξηθεί (υποκείμενη στις συγκεκριμένες συνθήκες που καθορίζονται από την επέκταση) από τη συμπεριφορά που καθορίζεται από την A. Η συμπεριφορά εισάγεται στη θέση που καθορίζεται από το σημείο επέκτασης στη B, στο οποίο αναφέρεται από τη σχέση extend. Μία σχέση επέκτασης ανάμεσα σε περιπτώσεις χρήσης παρουσιάζεται με μία διακεκομμένη γραμμή με βέλος με ανοικτή κεφαλή, από την περίπτωση χρήσης που παρέχει την επέκταση στη βασική περίπτωση χρήσης. Το βέλος έχει ετικέτα με τη λέξη κλειδί «extend».
3. Γενίκευση (generalization) - Μία γενίκευση από την περίπτωση χρήσης A στην περίπτωση χρήσης B δηλώνει ότι η A είναι μία εξειδίκευση της B. Μία γενίκευση ανάμεσα σε δύο περιπτώσεις χρήσης παρουσιάζεται με ένα βέλος γενίκευσης, δηλαδή μία γραμμή με ένα κλειστό, άδειο βέλος που δείχνει στη γονική περίπτωση χρήσης.
4. Συμπερίληψη (include) – Μία σχέση συμπερίληψης από την περίπτωση χρήσης A στην περίπτωση χρήσης B δηλώνει ότι ένα στιγμιότυπο της περίπτωσης χρήσης A, θα περιέχει επίσης τη συμπεριφορά που καθορίζει η B. Η συμπεριφορά περιλαμβάνεται στη θέση που καθορίζει η A. Μία σχέση συμπερίληψης ανάμεσα σε δύο περιπτώσεις χρήσης παρουσιάζεται με μία γραμμή με διακεκομμένο βέλος με ανοικτή κεφαλή από τη βασική περίπτωση χρήσης στη συμπεριλαμβανόμενη. Το βέλος έχει ετικέτα με τη λέξη κλειδί «include».

Παράδειγμα:

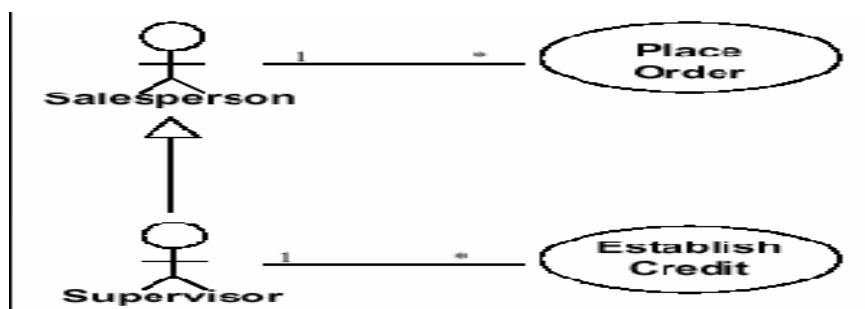


Σχήμα 2.6

∅ Σχέσεις μεταξύ actor – Actor Relationships: Υπάρχει μία σχέση ανάμεσα σε actors και μία σχέση ανάμεσα actors και περιπτώσεις χρήσης:

1. Συσχέτιση (association) – Η συμμετοχή ενός actor σε μία περίπτωση χρήσης, δηλαδή στιγμιότυπα του actor και της περίπτωσης χρήσης επικοινωνούν μεταξύ τους. Αυτή είναι η μόνη σχέση ανάμεσα actors και περιπτώσεις χρήσης. Μία συσχέτιση ανάμεσα σε έναν actor και μία περίπτωση χρήσης παρουσιάζεται ως μία γραμμή από τον actor προς την περίπτωση χρήσης.
2. Γενίκευση (generalization) – Η γενίκευση από έναν actor A σε έναν actor B δηλώνει ότι ένα στιγμιότυπο του A μπορεί να επικοινωνήσει με τα ίδια στιγμιότυπα περιπτώσεων χρήσης όπως ένα στιγμιότυπο του B. Μία γενίκευση ανάμεσα σε δύο actors παρουσιάζεται με ένα βέλος γενίκευσης, δηλαδή μία γραμμή με ένα κλειστό, άδειο βέλος. Η κεφαλή του βέλους δείχνει στον πιο γενικό actor.

Παράδειγμα:



Σχήμα 2.7 Σχέσεις μεταξύ actors.

2.5.5 Διαγράμματα ακολουθίας – Sequence Diagrams

Είδη διαγραμμάτων αλληλεπίδρασης: Ένα σχέδιο αλληλεπίδρασης ανάμεσα σε στιγμιότυπα παρουσιάζεται με χρήση ενός διαγράμματος αλληλεπίδρασης. Τα διαγράμματα αλληλεπίδρασης έχουν δύο μορφές για την ίδια βασική πληροφορία η οποία καθορίζεται από την αλληλεπίδραση, αλλά κάθε μορφή δίνει έμφαση σε μία συγκεκριμένη όψη της αλληλεπίδρασης. Οι δύο μορφές είναι: τα διαγράμματα ακολουθίας (sequence diagrams) και τα διαγράμματα συνεργασίας (collaboration diagrams).

Τα διαγράμματα ακολουθίας είναι καλύτερα για προδιαγραφές πραγματικού χρόνου και σύνθετα σενάρια. Επίσης παρουσιάζουν τις σχέσεις ανάμεσα στα στιγμιότυπα και είναι καλύτερα για την κατανόηση όλων των επιπτώσεων σε ένα συγκεκριμένο στιγμιότυπο και για διαδικαστικό σχεδιασμό. Τα διαγράμματα συνεργασίας θα παρουσιαστούν στη συνέχεια.

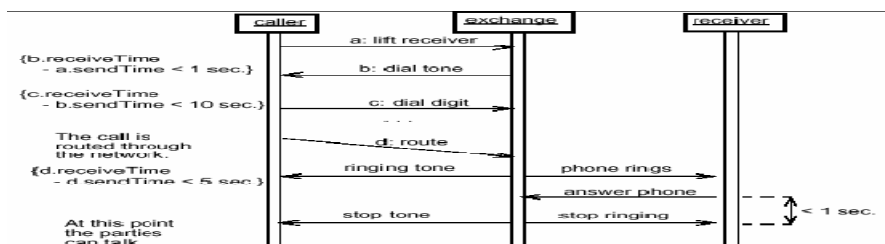
Ένα διάγραμμα ακολουθίας παρουσιάζει την αλληλεπίδραση όπως αυτή παρουσιάζεται σε χρονικές ακολουθίες. Συγκεκριμένα, παρουσιάζει τα στιγμιότυπα που συμμετέχουν στην αλληλεπίδραση με τις «γραμμές ζωής» τους και τα ερεθίσματα που ανταλλάσσουν τοποθετημένα στη χρονική ακολουθία. Δεν παρουσιάζει τις συσχετίσεις ανάμεσα στα αντικείμενα.

Ακόμη τα διαγράμματα αυτά έχουν λίγο διαφορετικές μορφές ανάλογα με τους διαφορετικούς σκοπούς, όπως να δίνεται έμφαση στον έλεγχο εκτέλεσης, στη συνδρομικότητα κλπ.

Ένα διάγραμμα ακολουθίας αναπαριστά την αλληλεπίδραση, δηλαδή μία σειρά από μηνύματα ανάμεσα σε classifiers μέσα σε μία συνεργασία ώστε να πραγματοποιηθεί μία επιθυμητή πράξη ή αποτέλεσμα και έχει δύο διαστάσεις: 1) την κάθετη διάσταση που αναπαριστά το χρόνο και 2) την οριζόντια διάσταση που αναπαριστά τα διάφορα αντικείμενα. Κανονικά ο χρόνος προχωρά προς τα κάτω. Αν είναι επιθυμητό οι διαστάσεις μπορούν να αλλάξουν θέση.

Οι χρονικοί περιορισμοί μπορούν να εκφραστούν με χρονικές εκφράσεις σε ονόματα μηνυμάτων. Οι συναρτήσεις `sendTime` (ο χρόνος κατά τον οποίο απεστάλη το μήνυμα από κάποιο αντικείμενο) και `receiveTime` (ο χρόνος κατά τον οποίο λήφθηκε το μήνυμα από το αντικείμενο) μπορούν να εφαρμοστούν σε ονόματα μηνυμάτων για να ληφθεί ένας χρόνος. Το σύνολο των συναρτήσεων των χρόνων είναι ανοικτό, συνεπώς οι χρήστες μπορούν να ανακαλύψουν νέες όπως τις χρειάζονται για ειδικές περιπτώσεις ή για διαχωρισμούς υλοποίησης (όπως `elapsedTime`, `executionStartTime`, `queuedTime`, `handledTime`).

Παράδειγμα:



Σχήμα 2.8 Απλά διαγράμματα ακολουθίας με ταυτόχρονα αντικείμενα.

Βασικές έννοιες των διαγραμμάτων αυτών είναι:

Ø **Γραμμή ζωής αντικειμένου – Object Lifeline:** Σε ένα διάγραμμα ακολουθίας η γραμμή ζωής αντικειμένου δηλώνει ένα αντικείμενο που παίζει ένα συγκεκριμένο ρόλο. Τα βέλη ανάμεσα στις γραμμές ζωής δηλώνουν επικοινωνία ανάμεσα στα αντικείμενα που παίζουν αυτούς τους ρόλους. Μέσα σε ένα διάγραμμα ακολουθίας παρουσιάζονται η ύπαρξη και η διάρκεια ενός αντικειμένου σε ένα ρόλο, αλλά όχι οι σχέσεις ανάμεσα στα αντικείμενα. Ένα αντικείμενο αναπαριστάται με μία διακεκομμένη κάθετη γραμμή που ονομάζεται γραμμή ζωής (lifeline).

Η γραμμή ζωής παριστάνει την ύπαρξη του αντικειμένου σε ένα συγκεκριμένο χρόνο. Αν το αντικείμενο δημιουργείται ή καταστρέφεται στη χρονική περίοδο που αναπαριστά το διάγραμμα, τότε η γραμμή ζωής του ξεκινά ή σταματά στο κατάλληλο σημείο, διαφορετικά πηγαίνει από την κορυφή προς το τέλος του διαγράμματος δημιουργεί το αντικείμενο, σχεδιάζεται με την κεφαλή του πάνω στο σύμβολο του αντικειμένου.


Η γραμμή ζωής μπορεί να διαχωριστεί σε δύο ή περισσότερες συντρέχουσες γραμμές ζωής για να δείξουν ύπαρξη συνθήκης. Κάθε ξεχωριστή γραμμή αντιστοιχεί σε ένα κλάδο συνθήκης στην επικοινωνία. Σε επόμενο σημείο οι γραμμές ζωής μπορούν να ξαναενώνονται.


Ø **Μηνύματα και ερεθίσματα – Message and Stimulus:** Ένα ερέθισμα είναι μία επικοινωνία ανάμεσα σε δύο αντικείμενα η οποία μεταφέρει πληροφορία αναμένοντας ότι θα συμβεί κάποια δραστηριότητα. Ένα ερέθισμα θα προκαλέσει την κλήση μίας πράξης, θα προκαλέσει ένα σήμα ή θα δημιουργήσει ή θα καταστρέψει ένα αντικείμενο.


Ένα μήνυμα είναι μία προδιαγραφή ερεθίσματος, δηλαδή καθορίζει τους ρόλους στους οποίους πρέπει να υπάγονται ο αποστολέας και ο δέκτης, καθώς και τις δράσεις που θα εκτελεστούν.


Στα διαγράμματα ακολουθίας το ερέθισμα παρουσιάζεται ως ένα οριζόντιο γεμάτο βέλος από τη γραμμή ζωής ενός αντικειμένου στη γραμμή ζωής ενός άλλου αντικειμένου.

Οι ακόλουθες επιλογές για την κεφαλή του βέλους επιλέγονται για να δείξουν διαφορετικό είδος επικοινωνίας.

Γεμάτη κεφαλή βέλους  : Πρόκειται για κλήση διαδικασίας ή άλλη ένθετη ροή ελέγχου. Πρέπει να ολοκληρωθεί ολόκληρη η ένθετη ακολουθία πριν συνεχίσει το εξωτερικό επίπεδο ακολουθίας. Μπορεί να χρησιμοποιηθεί για συνήθεις κλήσεις διαδικασιών. Μπορεί επίσης να χρησιμοποιηθεί με ταυτόχρονα ενεργά αντικείμενα όταν ένα από αυτά στέλνει ένα σήμα και περιμένει να ολοκληρωθεί μία ένθετη ακολουθία συμπεριφορών.

Απλή κεφαλή βέλους  : Απλή ροή ελέγχου. Κάθε βέλος δείχνει την πρόοδο στο επόμενο βήμα της ακολουθίας. Κανονικά όλα τα μηνύματα είναι ασύγχρονα.

Μισή κεφαλή βέλους  : Ένα ασύγχρονο ερέθισμα. Χρησιμοποιείται αντί της απλής κεφαλής για να δηλώσει ασύγχρονη επικοινωνία ανάμεσα σε δύο αντικείμενα σε διαδικαστική ακολουθία.

Διακεκομμένη γραμμή με απλή κεφαλή βέλους  : Επιστροφή μίας κλήσης διαδικασίας.

∅ **Χρόνοι μετάβασης – Transition times:** Ένα μήνυμα μπορεί να καθορίσει αρκετούς διαφορετικούς χρόνους, π.χ. χρόνος αποστολής και χρόνος λήψης. Αυτά είναι τυπικά ονόματα και μπορούν να χρησιμοποιηθούν σε εκφράσεις περιορισμών. Το σύνολο των διαφόρων ειδών χρόνων είναι ανοικτό και οι χρήστες μπορούν να προσθέσουν νέα, όπως τα χρειάζονται για ειδικές περιπτώσεις όπως *elapsedTime* και *startExecutionTime*. Οι εκφράσεις αυτές μπορούν να χρησιμοποιηθούν σε περιορισμούς για να καθορίσουν ειδικούς χρονικούς περιορισμούς που είναι έγκυροι για το μήνυμα.

Ο χρονικός περιορισμός μπορεί να παρουσιάζεται στο αριστερό περιθώριο, ευθυγραμμισμένος με το βέλος (σε ένα διάγραμμα ακολουθίας) ή κοντά στην ουρά του βέλους (σε ένα διάγραμμα συνεργασίας). Οι περιορισμοί μπορούν να καθορίζονται με χρήση λογικών εκφράσεων, οι οποίες πιθανά περιλαμβάνουν χρονικές εκφράσεις, σε άγκιστρα στα διαγράμματα ακολουθίας.

2.5.6 Διαγράμματα συνεργασίας – Collaboration Diagrams

Ένα διάγραμμα συνεργασίας μπορεί να εμφανιστεί σε δύο διαφορετικές μορφές: είτε στο **επίπεδο προδιαγραφών** (το διάγραμμα παρουσιάζει ρόλους classifier, ρόλους συσχετίσεων και μηνύματα) είτε στο **επίπεδο στιγμιότυπων** (το διάγραμμα παρουσιάζει αντικείμενα, συνδέσμους και ερεθίσματα). Το πρώτο παρουσιάζει τους ρόλους και τη δομή τους όπως αυτοί ορίζονται στην υποκείμενη συνεργασία, ενώ το δεύτερο εστιάζει στα στιγμιότυπα που υπακούουν στους ρόλους της συνεργασίας. Στη συνέχεια με τον όρο **αντικείμενο** εννοείται κάθε είδος στιγμιότυπου και παρουσιάζονται οι βασικές έννοιες των συγκεκριμένων διαγραμμάτων.

Βασικές έννοιες:

∅ **Συνεργασία – Collaboration:** Η συμπεριφορά υλοποιείται με ένα σύνολο αντικειμένων που ανταλλάσσουν ερεθίσματα σε μία πλήρη αλληλεπίδραση για να επιτύχουν ένα σκοπό. Για να γίνουν κατανοητοί οι μηχανισμοί που χρησιμοποιούνται στο σχεδιασμό είναι σημαντικό να παρουσιαστούν μόνο εκείνα τα αντικείμενα και η αλληλεπίδρασή τους που έχουν να κάνουν με την ικανοποίηση του στόχου ή ενός σχετιζόμενου συνόλου στόχων, όπως έχουν προβληθεί από το μεγαλύτερο σύστημα στο οποίο ανήκουν για άλλους στόχους. Μία τέτοια στατική δομή ονομάζεται **συνεργασία** (collaboration).

Μία συνεργασία ορίζει ένα σύνολο συμμετεχόντων και σχέσεων που έχουν νόημα για το συγκεκριμένο σύνολο στόχων. Ο προσδιορισμός των συμμετεχόντων και των σχέσεων τους δεν έχει καθολική έννοια. Οι συμμετέχοντες ορίζουν τους ρόλους που παίζουν τα αντικείμενα όταν αλληλεπιδρούν μεταξύ τους. Συνεπώς, μία συνεργασία καθορίζει ένα σύνολο ρόλων classifier και ρόλων συσχετίσεων. Τα αντικείμενα που ικανοποιούν (ή συνδέονται) με τους ρόλους classifier παίζουν

τους ρόλους που αυτοί καθορίζουν, ενώ οι σύνδεσμοι ανάμεσα στα αντικείμενα θα πρέπει να ικανοποιούν τους ρόλους συσχέτισης της συνεργασίας. Ένα ρόλος classifier (ρόλος συσχέτισης) ορίζει μία χρήση του αντικειμένου (συνδέσμου), ενώ μία κλάση (συσχέτιση) καθορίζει όλες τις ιδιότητες του αντικειμένου (συνδέσμου).

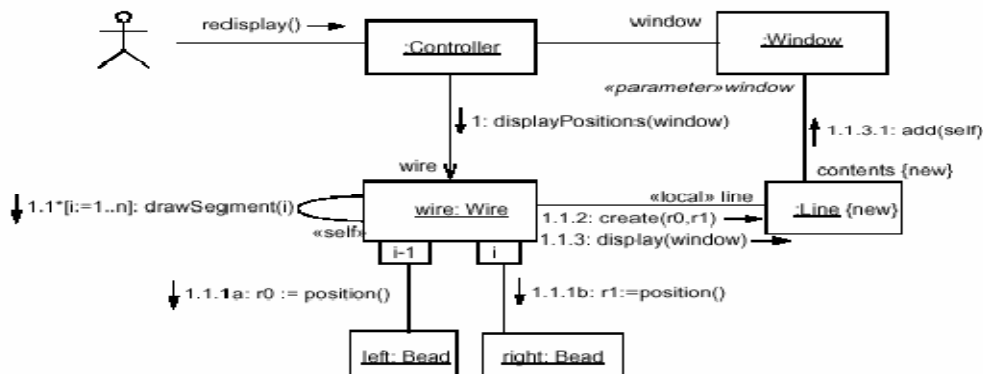
Στο πλαίσιο μίας συνεργασίας ορίζεται μία αλληλεπίδραση. Καθορίζει τα patterns επικοινωνίας ανάμεσα στους ρόλους. Πιο συγκεκριμένα, περιέχει ένα σύνολο μερικά ταξινομημένων μηνυμάτων, κάθε ένα από τα οποία καθορίζει μία επικοινωνία, π.χ. ποιο σήμα θα σταλεί ή ποια πράξη θα ενεργοποιηθεί, καθώς και τους ρόλους που θα πρέπει να παίξουν ο αποστολέας και ο παραλήπτης, αντίστοιχα.

Η περιγραφή της συμπεριφοράς περιλαμβάνει δύο όψεις: 1) τη δομική περιγραφή των συμμετεχόντων και 2) την περιγραφή των patterns επικοινωνίας. Οι δύο απόψεις περιγράφονται συχνά μαζί σε ένα διάγραμμα, αλλά ορισμένες φορές είναι χρήσιμο να περιγραφούν η δομική και η αλληλεπιδραστική άποψη ξεχωριστά. Η δομή των αντικειμένων που παίζουν κάποιους ρόλους σε μία συμπεριφορά και οι σχέσεις τους ονομάζονται συνεργασία. Ένα διάγραμμα συνεργασίας παρουσιάζει το πλαίσιο μέσα στο οποίο λαμβάνει χώρα η συνεργασία

Ένα διάγραμμα συνεργασίας παρουσιάζει μία συνεργασία, η οποία περιέχει ένα σύνολο ρόλων που θα παίξουν τα αντικείμενα, καθώς και τις απαιτούμενες σχέσεις για ένα δεδομένο πλαίσιο. Το διάγραμμα μπορεί επίσης να παρουσιάζει μία αλληλεπίδραση που ορίζει ένα σύνολο μηνυμάτων που καθορίζουν την αλληλεπίδραση ανάμεσα στα αντικείμενα που παίζουν τους ρόλους μέσα στη συνεργασία, ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα.

Επειδή τα διαγράμματα συνεργασίας χρησιμοποιούνται συχνά για να σχεδιαστούν διαδικασίες, συνήθως παρουσιάζουν τη δυνατότητα πλοήγησης με χρήση βελών στις γραμμές που αναπαριστούν τους συνδέσμους ή τους ρόλους συσχέτισεων. (Ένα βέλος σε μία γραμμή που ενώνει δύο πλαίσια δηλώνει ένα σύνδεσμο ή ένα ρόλο συσχέτισης με μονόδρομη δυνατότητα πλοήγησης. Ένα βέλος δίπλα σε μία γραμμή δηλώνει ερέθισμα προς τη δεδομένη κατεύθυνση. Προφανώς ένα τέτοιο βέλος δεν μπορεί να δείχνει αντίθετα σε μία μονόδρομη γραμμή.)

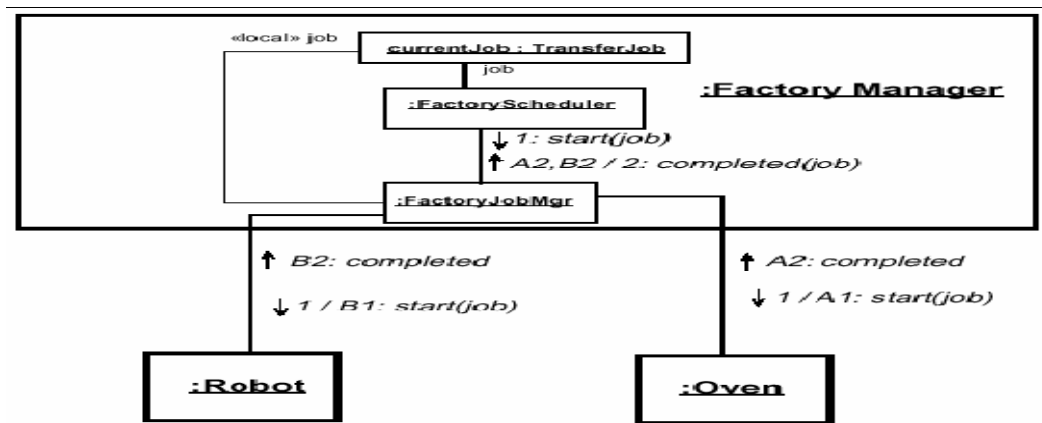
Παράδειγμα:



Σχήμα 2.14

∅ **Ενεργό αντικείμενο – Active Object:** Ένα ενεργό αντικείμενο έχει ένα νήμα ελέγχου και μπορεί να ξεκινήσει μία δραστηριότητα ελέγχου. Ένα παθητικό αντικείμενο διατηρεί δεδομένα, αλλά δεν ξεκινά μία δραστηριότητα ελέγχου. Όμως ένα παθητικό αντικείμενο μπορεί να στείλει ερεθίσματα κατά τη διαδικασία επεξεργασίας μίας αίτησης που έχει λάβει. Σε ένα διάγραμμα συνεργασίας ένας ρόλος classifier που είναι ενεργή κλάση αναπαριστά τα ενεργά αντικείμενα που θα εμφανιστούν κατά τη διάρκεια της εκτέλεσης. Ένα ενεργό αντικείμενο είναι ένα αντικείμενο στο οποίο ανήκει το νήμα ελέγχου. Οι διεργασίες και τα έργα είναι παραδοσιακά είδη ενεργών αντικειμένων. Ο ρόλος του ενεργού αντικειμένου παρουσιάζεται ως ορθογώνιο με χοντρό πλαίσιο. Συχνά οι ρόλοι των ενεργών αντικειμένων αναπαριστώνται ως σύνθετα με ενσωματωμένα μέρη. Επίσης μπορεί να χρησιμοποιηθεί η λέξη κλειδί {active} για να δηλώσει ένα ενεργό αντικείμενο.

Παράδειγμα:



Σχήμα 2.15 Ένα σύνθετο ενεργό αντικείμενο.

2.5.7 Διαγράμματα κατάστασης – Statechart Diagrams

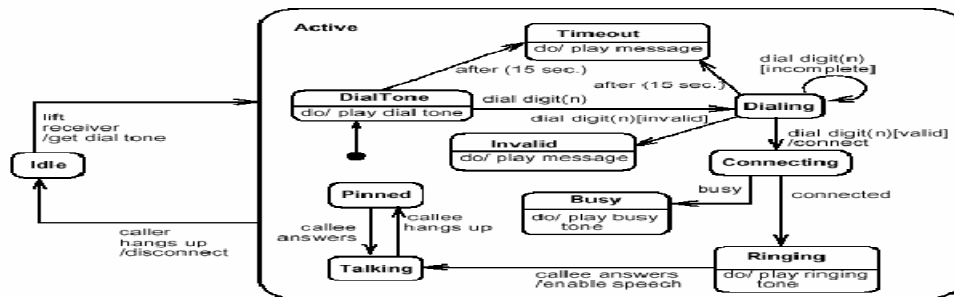
Ένα διάγραμμα κατάστασης μπορεί να χρησιμοποιηθεί για να περιγράψει τη συμπεριφορά ενός στοιχείου μοντελοποίησης όπως ένα αντικείμενο ή μία αλληλεπίδραση. Συγκεκριμένα, περιγράφει πιθανές ακολουθίες καταστάσεων και δράσεων μέσω των οποίων το στοιχείο μπορεί να προχωρήσει κατά τη διάρκεια της ζωής του, ως αποτέλεσμα της αντίδρασης σε διακριτά συμβάντα (π.χ. σήματα, κλήσεις πράξεων).

Η σημασιολογία και ο συμβολισμός που περιγράφεται εδώ διαφέρει σημαντικά από αυτά των διαγραμμάτων κατάστασης του David Harel, με τροποποιήσεις για να γίνουν αντικειμενοστραφή.

Τα διαγράμματα κατάστασης παρουσιάζουν τη συμπεριφορά των οντοτήτων που έχουν τη δυνατότητα δυναμικής συμπεριφοράς, καθορίζοντας την αντίδρασή τους κατά τη λήψη στιγμιότυπων συμβάντων. Τυπικά χρησιμοποιείται για να περιγράψει τη συμπεριφορά των κλάσεων, αλλά τα διαγράμματα κατάστασης μπορούν επίσης να περιγράψουν τη συμπεριφορά άλλων οντοτήτων του μοντέλου, όπως περιπτώσεις χρήσης, actors, υποσυστήματα, πράξεις και μέθοδοι.

Ένα απλό παράδειγμα διαγράμματος κατάστασης είναι αυτό που ακολουθεί και περιγράφει ένα τηλεφωνικό σύστημα.

Παράδειγμα:



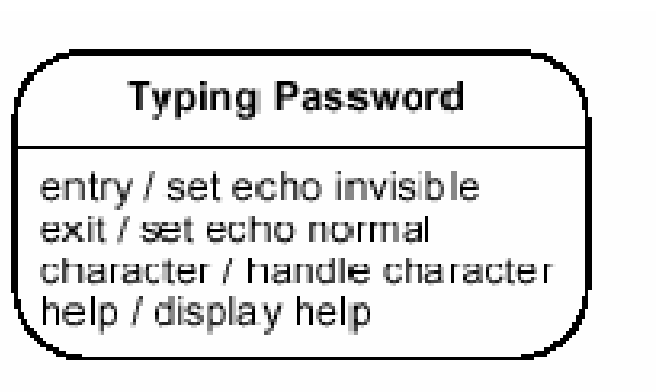
Σχήμα 2.16 Διάγραμμα κατάστασης τηλεφωνικού συστήματος

Βασικές έννοιες των Διαγραμμάτων Κατάστασης αναλύονται παρακάτω:

∅ **Κατάσταση – State:** Μία κατάσταση είναι μία συνθήκη κατά τη διάρκεια της ζωής ενός αντικειμένου ή μία αλληλεπίδραση κατά την οποία ικανοποιείται κάποια συνθήκη, πραγματοποιείται κάποια δράση ή αναμένει για κάποιο συμβάν. Μία *σύνθετη* κατάσταση είναι μία κατάσταση η οποία, σε αντίθεση με μία *απλή* κατάσταση, μπορεί να διασπαστεί γραφικά. Ενωσιολογικά, ένα αντικείμενο παραμένει σε μία κατάσταση για μία χρονική περίοδο. Όμως η σημασιολογία επιτρέπει να μοντελοποιούνται καταστάσεις μέσα από τις οποίες περνάει η ροή σχεδόν στιγμιαία, καθώς και μη στιγμιαίες μεταβάσεις.

Μία κατάσταση μπορεί να χρησιμοποιηθεί για να μοντελοποιήσει μία δραστηριότητα που λαμβάνει χώρα. Μία τέτοια δραστηριότητα καθορίζεται είτε με μία ένθετη μηχανή καταστάσεων ή με μία υπολογιστική έκφραση. Μία κατάσταση παρουσιάζεται ως ένα ορθογώνιο με στρογγυλεμένες γωνίες.

Παράδειγμα:



Σχήμα 2.17 Κατάσταση

∅ **Σύνθετη κατάσταση – Composite State:** Μία σύνθετη κατάσταση αποσυντίθεται σε δύο ή περισσότερες συντρέχουσες υποκαταστάσεις (οι οποίες ονομάζονται *περιοχές*) ή σε αμοιβαία αποκλειόμενες ξένες μεταξύ τους υποκαταστάσεις. Μία δεδομένη κατάσταση μπορεί να εκλεπτυνθεί με έναν ακριβώς από τους δύο

αυτούς τρόπους. Φυσικά, οποιαδήποτε υποκατάσταση μίας σύνθετης κατάστασης μπορεί να είναι επίσης μία σύνθετη κατάσταση οποιουδήποτε τύπου.

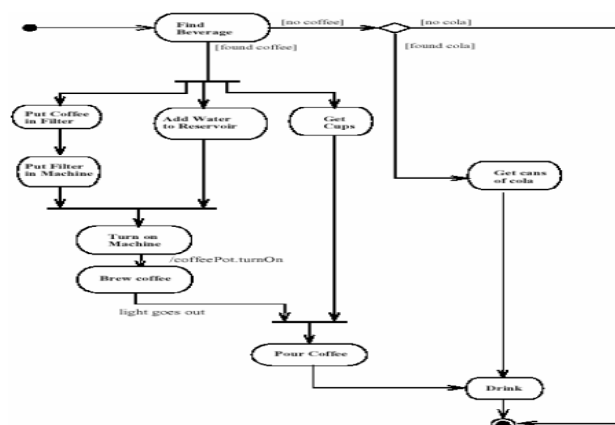
∅ **Συμβάντα – Events:** Ένα συμβάν είναι κάτι που συνέβη και έχει σημασία. Για πρακτικούς σκοπούς στα διαγράμματα κατάστασης, είναι κάτι που συμβαίνει και προκαλεί μία μετάβαση κατάστασης. Ο ορισμός ενός συμβάντος έχει πεδίο δράσης του το πακέτο στο οποίο εμφανίζεται και μπορεί να χρησιμοποιηθεί στα διαγράμματα καταστάσεων των κλάσεων που έχουν ορατότητα μέσα στο πακέτο.

∅ **Απλές μεταβάσεις – Simple Transitions:** Μία απλή μετάβαση είναι μία σχέση ανάμεσα σε δύο καταστάσεις, η οποία δηλώνει ότι ένα αντικείμενο που βρίσκεται στην πρώτη κατάσταση θα μεταβεί στη δεύτερη κατάσταση και θα κάνει κάποιες συγκεκριμένες δράσεις όταν λάβει χώρα ένα συγκεκριμένο συμβάν και με την προϋπόθεση ότι ικανοποιούνται ορισμένες συνθήκες. Σε μία τέτοια αλλαγή κατάστασης η μετάβαση λέγεται ότι «πυροδοτείται. Μία μετάβαση παρουσιάζεται ως μία γραμμή που ξεκινά από την πηγαία κατάσταση και τερματίζεται με ένα βέλος στην κατάσταση στόχο.

2.5.8 Διαγράμματα δραστηριότητας – Activity diagrams

Ένα διάγραμμα δραστηριότητας είναι μία ειδική κατάσταση ενός διαγράμματος κατάστασης στο οποίο όλες (ή τουλάχιστον οι περισσότερες) από τις μεταβάσεις προκαλούνται από την ολοκλήρωση των δράσεων ή των υποδραστηριοτήτων των πηγαίων καταστάσεων. Το διάγραμμα δραστηριότητας προσκολλάται (μέσω του μοντέλου) σε μία κλάση, όπως μία περίπτωση χρήσης, ή σε ένα πακέτο ή στην υλοποίηση μίας πράξης. Σκοπός του διαγράμματος αυτού είναι να εστιάσει σε ροές που προκαλούνται από την εσωτερική επεξεργασία (σε αντιδιαστολή με τα εξωτερικά συμβάντα). Χρησιμοποιούμε διαγράμματα δραστηριότητας σε περιπτώσεις όπου όλα ή τα περισσότερα από τα συμβάντα αναπαριστούν την ολοκλήρωση εσωτερικά παραγόμενων δράσεων (δηλαδή, τη διαδικαστική ροή ελέγχου). Χρησιμοποιούμε απλά διαγράμματα καταστάσεων σε περιπτώσεις που συμβαίνουν ασύγχρονα συμβάντα.

Παράδειγμα:



Σχήμα 2.18 Διάγραμμα δραστηριότητας.

Έννοιες που χρησιμοποιούνται για τα διαγράμματα αυτής της μορφής είναι:

∅ **Κατάσταση δράσης – Action state:** Μία κατάσταση δράσης είναι μία σύντηξη για μία κατάσταση με μία δράση εισόδου και τουλάχιστον μία εξερχόμενη μετάβαση η οποία περιλαμβάνει το εννοούμενο συμβάν της ολοκλήρωσης της δράσης εισόδου (μπορούν να υπάρχουν πολλές τέτοιες μεταβάσεις αν έχουν συνθήκες φύλακες). Η συνήθης χρήση μίας κατάστασης δράσης είναι για να μοντελοποιήσει ένα βήμα στην εκτέλεση ενός αλγορίθμου (μίας διαδικασίας) ή μία διαδικασία ροής εργασίας.

Μία κατάσταση δράσης παρουσιάζεται ως ένα σχήμα με το πάνω και το κάτω μέρος του να είναι ευθεία, ενώ οι δύο πλευρές του είναι κυρτά τόξα. Μέσα στο σύμβολο υπάρχει μία έκφραση-δράσης. Η έκφραση δράσης δε χρειάζεται να είναι μοναδική μέσα στο διάγραμμα.

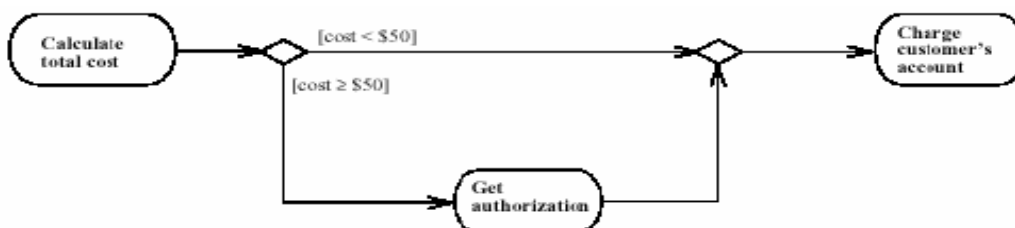
Παράδειγμα



Σχήμα 2.19 Καταστάσεις δράσης.

∅ **Αποφάσεις – Decisions:** Ένα διάγραμμα κατάστασης (και κατ' επέκταση ένα διάγραμμα δραστηριότητας) εκφράζει μία απόφαση όταν χρησιμοποιούνται συνθήκες φύλακες για να δηλώσουν διαφορετικές πιθανές μεταβάσεις που εξαρτώνται από λογικές συνθήκες του αντικειμένου ιδιοκτήτη. Η UML παρέχει μία σύντηξη για να δείχνει τις αποφάσεις και να τις ενώνει τις ξεχωριστές διαδρομές μαζί. Μία απόφαση μπορεί να παρουσιαστεί τοποθετώντας ετικέτες σε πολλές εξερχόμενες μεταβάσεις μίας δράσης με διαφορετικές συνθήκες φύλακες. Το εικονίδιο που παρέχεται για την απόφαση είναι ο παραδοσιακός ρόμβος, με ένα εισερχόμενο βέλος και δύο ή περισσότερα εξερχόμενα, κάθε ένα από τα οποία έχει ως ετικέτα μία συνθήκη φύλακα χωρίς trigger ενός συμβάντος. Θα πρέπει όλα τα πιθανά αποτελέσματα να εμφανίζονται σε μία από τις εξερχόμενες μεταβάσεις. Μπορεί να οριστεί ένας προκαθορισμένος φύλακας [else] σε πολύ μία από τις εξερχόμενες μεταβάσεις. Η μετάβαση αυτή ενεργοποιείται αν όλοι οι φύλακες των άλλων μεταβάσεων είναι ψευδείς.

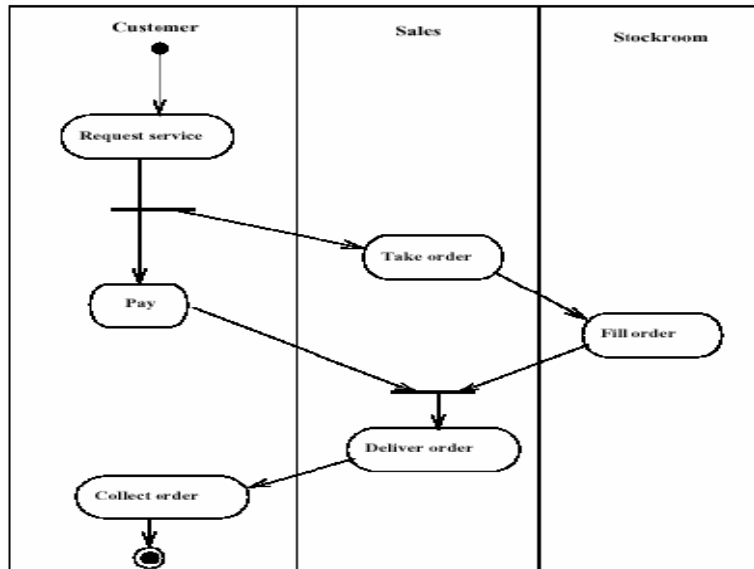
Παράδειγμα:



Σχήμα 2.20 Απόφαση και ένωση.

Ø **Swimlanes:** Οι πράξεις και οι υποδραστηριότητες μπορούν να οργανωθούν σε *swimlanes*. Οι *swimlanes* χρησιμοποιούνται για να οργανώσουν την υπευθυνότητα για δράσεις και υποδραστηριότητες ανάλογα με την κλάση. Συχνά αντιστοιχούν σε οργανωτικές μονάδες στο εταιρικό μοντέλο. Κάθε *swimlane* αναπαριστά αρμοδιότητα για μέρος της ολικής δραστηριότητας και μπορεί τελικά να υλοποιηθεί με ένα ή περισσότερα αντικείμενα. Η σχετική σειρά των *swimlanes* δεν έχει σημασιολογική σημασία, αλλά μπορεί να δηλώνει κάποια συνάφεια.

Παράδειγμα:



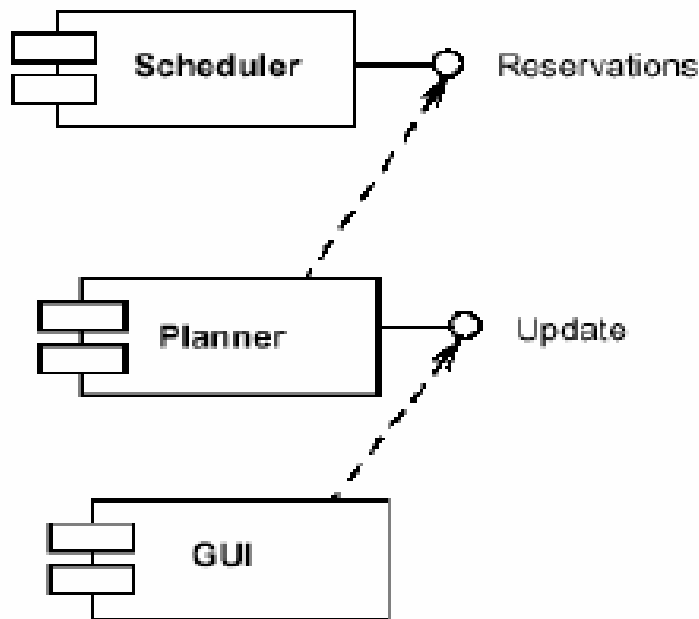
2.21 Swimlanes στο διάγραμμα δραστηριότητας.

2.5.9 Διαγράμματα υλοποίησης – Implementation diagrams

Τα διαγράμματα υλοποίησης παρουσιάζουν όψεις της υλοποίησης, συμπεριλαμβανομένης της δομής του κώδικα και της δομής της υλοποίησης. Υπάρχουν δύο μορφές τους: (1) τα διαγράμματα συνιστωσών παρουσιάζουν τη δομή του ίδιου του κώδικα και (2) τα διαγράμματα deployment παρουσιάζουν τη δομή του συστήματος που θα εκτελεστεί. Επίσης μπορούν να εφαρμοστούν με πιο ευρεία έννοια στη μοντελοποίηση μίας εταιρίας όπου «κώδικας» είναι οι διαδικασίες της εταιρίας και τα κείμενά της και η «δομή του συστήματος που θα εκτελεστεί» είναι οι οργανωτικές δομές και οι πόροι (ανθρώπινοι και άλλοι) της εταιρίας.

Ένα διάγραμμα συνιστώσας παρουσιάζει τις εξαρτήσεις ανάμεσα στις συνιστώσες λογισμικού, συμπεριλαμβανομένων των συνιστωσών πηγαίου κώδικα, συνιστωσών δυαδικού κώδικα και εκτελέσιμων συνιστωσών. Ένα τμήμα λογισμικού μπορεί να αναπαρασταθεί με ένα στερεότυπο συνιστώσας. Ορισμένες συνιστώσες υπάρχουν κατά το χρόνο μεταγλώττισης, ορισμένες κατά το χρόνο σύνδεσης, ορισμένες κατά την εκτέλεση, ενώ άλλες μπορούν να υπάρχουν σε περισσότερους από έναν χρόνους. Μία συνιστώσα μόνο μεταγλώττισης είναι μία συνιστώσα που έχει νόημα μόνο κατά το χρόνο της μεταγλώττισης. Μία συνιστώσα εκτέλεσης στην περίπτωση αυτή θα μπορούσε να είναι ένα εκτελέσιμο πρόγραμμα.

Παράδειγμα:



Σχήμα 2.22 Διάγραμμα συνιστώσας.

Τα διαγράμματα deployment παρουσιάζουν τη σύνθεση των επεξεργαστικών στοιχείων κατά το χρόνο εκτέλεσης και τις συνιστώσες λογισμικού, τις διεργασίες και τα αντικείμενα που ζουν πάνω από αυτά. Τα στιγμιότυπα των συνιστωσών λογισμικού αναπαριστούν εμφανίσεις των μονάδων του κώδικα κατά το χρόνο εκτέλεσης. Οι συνιστώσες που δεν υφίστανται ως οντότητες του χρόνου εκτέλεσης (γιατί έφυγαν λόγω της μεταγλώττισης) δεν εμφανίζονται σε αυτά τα διαγράμματα και θα πρέπει να εμφανίζονται στα διαγράμματα συνιστωσών.

Ένα διάγραμμα deployment είναι ένας γράφος κόμβων οι οποίοι είναι συνδεδεμένοι με συσχετίσεις επικοινωνίας. Οι κόμβοι μπορούν να περιέχουν στιγμιότυπα συνιστωσών. Αυτό δηλώνει ότι οι συνιστώσες ζουν ή εκτελούνται στον κόμβο.

Το διάγραμμα deployment μπορεί επίσης να χρησιμοποιηθεί για να δείξει ποιες συνιστώσες μπορούν να βρισκονται σε ποιους κόμβους, χρησιμοποιώντας διακεκομμένα βέλη με το στερεότυπο «support» από το σύμβολο της συνιστώσας στο σύμβολο του κόμβου ή γραφικά εμφανίζοντας το σύμβολο της συνιστώσας ένθετο στο σύμβολο του κόμβου.

Πρέπει να δώσουμε ιδιαίτερη σημασία στις παρακάτω έννοιες:

∅ **Κόμβος – Node:** Ένας κόμβος είναι ένα φυσικό αντικείμενο που γενικά αναπαριστά έναν πόρο επεξεργασίας, ο οποίος έχει τουλάχιστον μνήμη και συχνά και δυνατότητα επεξεργασίας. Οι κόμβοι περιλαμβάνουν τις συσκευές υπολογισμού αλλά και ανθρώπινους πόρους ή πόρους μηχανικής επεξεργασίας. Οι κόμβοι μπορούν να αναπαρασταθούν ως τύποι και ως στιγμιότυπα. Οι κόμβοι μπορεί να είναι συνδεδεμένοι με συσχετίσεις με άλλους κόμβους. Μία συσχέτιση ανάμεσα σε κόμβους δηλώνει μία διαδρομή επικοινωνίας ανάμεσα

στους κόμβους. Η συσχέτιση μπορεί να έχει ένα στερεότυπο που να δηλώνει το είδος της διαδρομής επικοινωνίας (για παράδειγμα, το είδος του καναλιού ή του δικτύου).

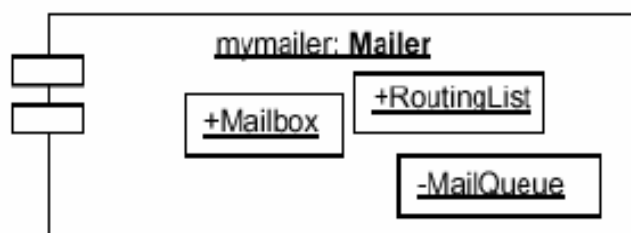
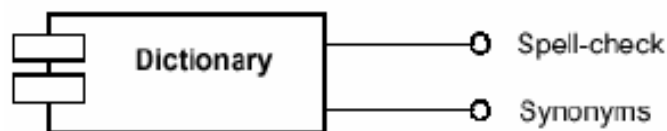
Ø **Συνιστώσα – Component:** Ένας τύπος συνιστώσας αναπαριστά ένα τμήμα της υλοποίησης του συστήματος το οποίο έχει τη δυνατότητα να κατανεμηθεί, συμπεριλαμβανομένου του κώδικα λογισμικού (πηγαίου, δυαδικού ή εκτελέσιμου), αλλά και εταιρικών κειμένων κλπ. σε ένα ανθρώπινο σύστημα. Οι συνιστώσες μπορούν να χρησιμοποιηθούν για να παρουσιάσουν εξαρτήσεις, όπως εξαρτήσεις μεταγλώττισης ή χρόνου εκτέλεσης ή πληροφοριακές εξαρτήσεις σε έναν οργανισμό ανθρώπων. Ένα στιγμιότυπο συνιστώσας αναπαριστά μία μονάδα υλοποίησης σε χρόνο εκτέλεσης και μπορεί να χρησιμοποιηθεί για να δείξει τις μονάδες υλοποίησης που έχουν ταυτότητα κατά το χρόνο εκτέλεσης, συμπεριλαμβανομένης της θέσης τους στους κόμβους.

Μία συνιστώσα αναπαριστάται με ένα ορθογώνιο με δύο μικρά ορθογώνια να προεξέχουν στη μία του πλευρά. Ένας τύπος συνιστώσας έχει όνομα τύπου:

τύπος-συνιστώσας

Το παράδειγμα του Σχήματος 2.23 που ακολουθεί, δείχνει μία συνιστώσα με διασυνδέσεις και επίσης μία συνιστώσα που περιέχει αντικείμενα κατά το χρόνο εκτέλεσης.

Παράδειγμα:



Σχήμα 2.23 Συνιστώσες.

2.6 Εργαλεία CASE

Μολονότι η φράση «αυτοματοποίηση μεγάλων και σύνθετων εργασιών» δημιουργεί συνειρμούς προς διάφορα αντικείμενα εφαρμογών (οικονομικά, επιστημονικά, επικοινωνίας, διοίκησης, εκπαίδευσης κ.ά.), ένα τέτοιο αντικείμενο είναι και η ίδια η ανάπτυξη του λογισμικού. Τίθεται, δηλαδή, υπό εξέταση, το πρόβλημα της υποστήριξης και της ανάπτυξης λογισμικού. Έτσι γεννήθηκε ένας ιδιαίτερος κλάδος της Τεχνολογίας Λογισμικού, ο οποίος ονομάστηκε **Τεχνολογία Λογισμικού Υποβοηθούμενη από Υπολογιστή – CASE**. Στην πράξη ο όρος χρησιμοποιείται για να αναφερθεί περισσότερο σε εργαλεία και λιγότερο σε ολοκληρωμένες προσεγγίσεις ανάπτυξης λογισμικού.

Τα εργαλεία CASE, αν και υποσχέθηκαν πολλά, στην αρχή τουλάχιστον δεν είχαν τα αναμενόμενα αποτελέσματα. Ωστόσο καθώς η κοινότητα του λογισμικού ωριμάζει, τα πράγματα γίνονται ολοένα και καλύτερα. Η σωστή εφαρμογή των εργαλείων CASE μπορεί να οδηγήσει σε μείωση του κόστους και του χρόνου ανάπτυξης του λογισμικού, καθώς και στη βελτίωση της ποιότητάς του. Επίσης, μπορεί να επιφέρει τον επαναπροσδιορισμό κάποιων εργασιών ανάπτυξης λογισμικού, έτσι ώστε αυτές να είναι περισσότερο αποτελεσματικές. Μια πιο λεπτομερή αναφορά στα εργαλεία CASE επιχειρείται στις παρακάτω ενότητες.

2.6.1 Χαρακτηριστικά και πλαίσιο λειτουργίας των Εργαλείων CASE

Η υποστήριξη που παρέχουν τα εργαλεία CASE στους μηχανικούς λογισμικού στοχεύει στη βελτίωση της παραγωγικότητας, αλλά και της ποιότητας του λογισμικού. Από τις αρχές της δεκαετίας του '80, έχει παρουσιαστεί ένα ευρύ φάσμα τέτοιων εργαλείων. Ο όρος που επικράτησε για να περιγράφονται τα εργαλεία αυτά είναι όπως έχουμε αναφέρει «**Τεχνολογία Λογισμικού Υποβοηθούμενη από Υπολογιστή**» (CASE, Computer Aided ή Assisted Software Engineering).

Σήμερα, μπορούμε να καταγράψουμε **τρία γενικά επίπεδα της τεχνολογίας** των εργαλείων CASE:

1. **Επίπεδο υποστήριξης εργασιών – παραγωγής:** Εδώ κατατάσσεται η υποστήριξη σε επίπεδο εργασιών που εκτελούνται κατά την ανάπτυξη λογισμικού, όπως ο ορισμός και η διαχείριση των απαιτήσεων από το λογισμικό, η ανάλυση, η γέννηση κώδικα και ο έλεγχος. Τα εργαλεία αυτά είναι από τα πρώτα που εμφανίστηκαν και συνεπώς είναι, σήμερα, από τα πιο ώριμα εργαλεία CASE.
2. **Επίπεδο διαχείρισης – διοίκησης εργασιών:** Στο επίπεδο αυτό κατατάσσεται η μοντελοποίηση και διαχείριση των διαφόρων φάσεων της ανάπτυξης λογισμικού. Αποτελεί, δηλαδή, υπερσύνολο της προηγούμενης κατηγορίας και χρησιμοποιεί υπηρεσίες που παρέχονται απ' αυτή για την υποστήριξη συγκεκριμένων εργασιών μέσα σε κάποια φάση της ανάπτυξης.

- 3. Επίπεδο Meta-CASE:** Στο επίπεδο αυτό ανήκουν εργαλεία τα οποία χρησιμοποιούνται για τη γέννηση (δημιουργία) περιβαλλόντων CASE των δυο παραπάνω επιπέδων. Ένας μικρός αριθμός τέτοιων εργαλείων είναι διαθέσιμος σήμερα στην αγορά. Η ρευστότητα των πραγμάτων στις προσεγγίσεις αλλά και στα περιβάλλοντα ανάπτυξης λογισμικού καθιστά τέτοιες ιδέες δύσκολα αποδεκτές από τους κατασκευαστές λογισμικού.

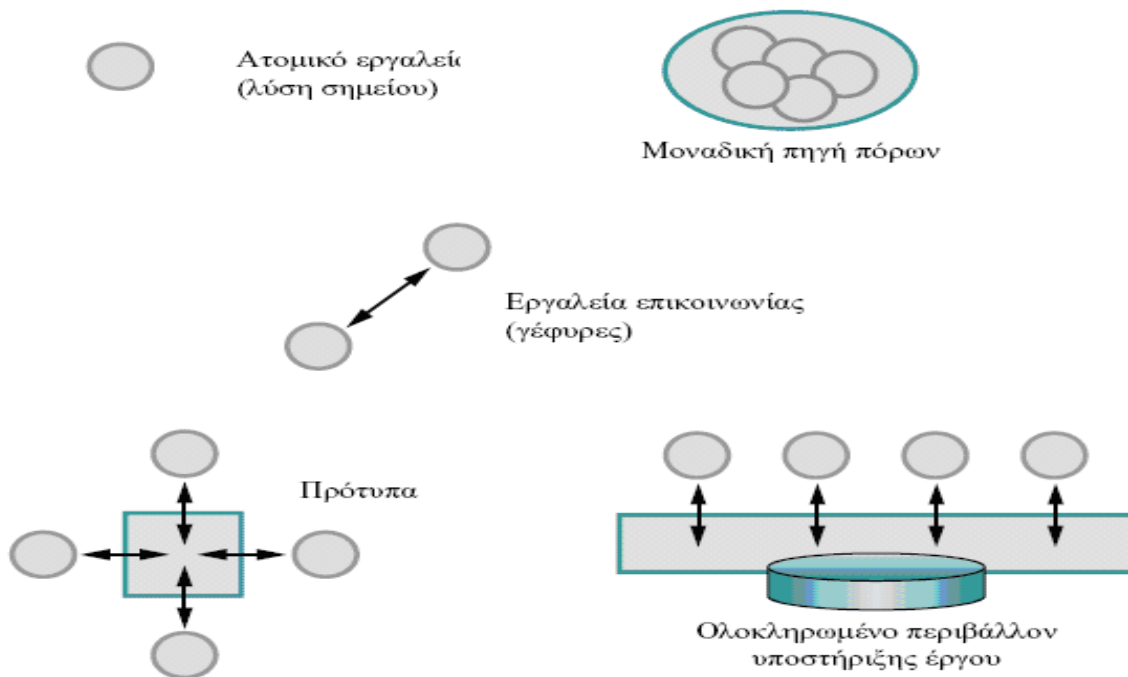
Ένα εργαλείο CASE μπορεί να είναι τόσο απλό, όσο χρειάζεται για την υποστήριξη μιας και μόνο συγκεκριμένης εργασίας κατά την ανάπτυξη του λογισμικού, ή τόσο σύνθετο, όσο ένα ολοκληρωμένο περιβάλλον υποστήριξης ολόκληρου ή μεγάλου μέρους του κύκλου ζωής, το οποίο μπορεί να περιλαμβάνει άλλα εργαλεία, μια βάση δεδομένων σχετικά με το λογισμικό και την ανάπτυξή του, υλικό και δίκτυο υπολογιστών, πρότυπα εγγράφων τεκμηρίωσης, κ.ά. Στο Σχήμα 2.24 που ακολουθεί, φαίνεται μια ιεραρχία εννοιών αναφοράς για την τεχνολογία εργαλείων CASE. Καθεμία από αυτές αποτελεί το υπόβαθρο για την επόμενη, με τα εργαλεία να τοποθετούνται στην κορυφή του σωρού.



Σχήμα 2.24 Ιεραρχία εννοιών αναφοράς της τεχνολογίας CASE

Η αρχιτεκτονική περιβάλλοντος, στηρίζει την πλατφόρμα υλικού και το υποστηριζόμενο λειτουργικό περιβάλλον. Σ' αυτό περιλαμβάνεται το λειτουργικό σύστημα, ένα σύστημα διαχείρισης βάσεων δεδομένων καθώς και λογισμικό δικτυακών υπηρεσιών. Ακολουθώντας την ιεραρχία προς τα πάνω, περνάμε από ένα επίπεδο όπου παρέχονται υπηρεσίες μεταφερσιμότητας. Οι υπηρεσίες αυτές παρέχουν μια γέφυρα επικοινωνίας των εργαλείων CASE με το περιβάλλον χρήσης τους, το οποίο μπορεί να είναι ανομοιογενές και να περιλαμβάνει διαφορετικές πλατφόρμες υλικού και λειτουργικά συστήματα. Το ενοποιημένο πλαίσιο εργασίας μπορεί να ιδωθεί ως ένα κέλυφος που ενοποιεί εργαλεία CASE και τους επιτρέπει να επικοινωνούν μεταξύ τους, να αναφέρονται σε μια κοινή βάση πληροφοριών σχετικά με το λογισμικό, καθώς και να παρουσιάζουν την ίδια εμφάνιση και αίσθηση στον τελικό χρήστη.

Πολλά από τα εργαλεία CASE που χρησιμοποιούνται σήμερα αποτελούν αυτόνομες λύσεις σε μεμονωμένα προβλήματα και όχι τμήματα ενός ενοποιημένου περιβάλλοντος CASE. Έχοντας επίγνωση αυτής της αναμφίβολα μη ιδανικής κατάστασης, σημειώνουμε ότι ένα εργαλείο CASE μπορεί να χρησιμοποιηθεί ιδιαίτερα αποτελεσματικά, ακόμη και ως μεμονωμένη λύση. Το επόμενο λογικό βήμα είναι η ενοποίηση των εργαλείων CASE που όπως είναι αντιληπτό, μπορεί να γίνει με πολλούς τρόπους και σε πολλά επίπεδα. Τα επίπεδα ενοποίησης εργαλείων CASE φαίνονται στο Σχήμα 2.25 .



Σχήμα 2.25 Επίπεδα ενοποίησης και επικοινωνίας εργαλείων CASE

Στο χαμηλότερο από τα επίπεδα αυτά τοποθετούνται τα ανεξάρτητα εργαλεία. Όταν διατίθενται υπηρεσίες ανταλλαγής ή μοιράσματος δεδομένων μεταξύ ανεξάρτητων εργαλείων, βελτιώνεται και το επίπεδο ενοποίησης. Στην περίπτωση αυτή, καθένα από τα εργαλεία παράγει εξόδους ή δέχεται εισόδους σε μορφές αναγνωρίσιμες από τα άλλα εργαλεία, ώστε να επιτυγχάνεται η επιθυμητή συνεργασία. Το χαρακτηριστικό αυτό, έλλειπε για μεγάλο χρονικό διάστημα από τα περισσότερα εργαλεία CASE. Η εποχή αυτή ευτυχώς φαίνεται να έχει παρέλθει και σήμερα, σε αρκετές περιπτώσεις, οι κατασκευαστές εργαλείων CASE συνεργάζονται για τον σχηματισμό γεφυρών επικοινωνίας μεταξύ αυτών. Με την προσέγγιση αυτή βελτιώνεται η παραγωγικότητα και αυξάνεται η χρησιμότητα των εργαλείων CASE για τον κατασκευαστή λογισμικού. Στο υψηλότερο σημείο της κλίμακας, βρίσκεται το ολοκληρωμένο περιβάλλον υποστήριξης έργου (Integrated Project Support Environment – IPSE), το οποίο μπορεί να είναι και ανομοιογενές. Στην προσέγγιση αυτή δημιουργούνται πρότυπα επικοινωνίας για κάθε ένα από τα επίπεδα.

2.6.2 Ταξινόμηση των εργαλείων CASE

Όπως ισχύει γενικότερα στην Τεχνολογία Λογισμικού, έτσι και στην τεχνολογία των εργαλείων CASE, παρατηρείται πλουραλισμός ορολογίας για το χαρακτηρισμό των εργαλείων CASE, χωρίς οι λόγοι γι' αυτό να είναι πάντα τεχνικοί αλλά εμπορικοί. Όροι όπως εργαλείο, πάγκος εργαλείων και περιβάλλον, είναι ευρέως αποδεκτοί και χρησιμοποιούνται σε συνεχή βάση. Μια ταξινόμηση των εργαλείων CASE θα ήταν χρήσιμη, καθιστώντας εφικτή τη σύγκριση διαφορετικών τύπων εργαλείων και παρέχοντας παράλληλα ένα σημείο έναρξης για την κατανόηση και την προσέγγισή τους από τους κατασκευαστές λογισμικού. Ωστόσο, δε μπορούμε να πούμε ότι σήμερα υπάρχει μια πρότυπη τέτοια ταξινόμηση. Τα εργαλεία CASE μπορούν να ταξινομηθούν σύμφωνα με τη λειτουργία που επιτελούν, τη χρήση τους στις διάφορες φάσεις της ανάπτυξης του λογισμικού, την αρχιτεκτονική περιβάλλοντος (υλικού ή λογισμικού) που τα υποστηρίζει, ή ακόμη και σύμφωνα με την «καταγωγή» ή το κόστος τους.

Ταξινόμηση σύμφωνα με τη λειτουργία: Η ταξινόμηση αυτή δεν είναι απόλυτη, καθώς ένα εργαλείο μπορεί να ανήκει σε παραπάνω από μια κατηγορία. Επίσης, δε μπορεί να αποδειχθεί θεωρητικά η πληρότητά της, καθώς οι εξελίξεις τρέχουν με μεγάλους ρυθμούς. Παρά ταύτα, δίνει μια χρήσιμη εικόνα των εργαλείων που έχουν αναπτυχθεί έως και σήμερα και έτσι μπορεί να αποτελέσει σημείο αναφοράς. Έτσι η ταξινόμηση τους, έχει ως ακολούθως:

∅ **Εργαλεία διαχείρισης, ελέγχου και συντήρησης του έργου.** Στην κατηγορία αυτή μπορούμε να διακρίνουμε τις εξής υποκατηγορίες:

1. **Εργαλεία σχεδιασμού του έργου (Project Planning tools).** Τα εργαλεία αυτά ασχολούνται με δυο σημαντικά ζητήματα: την εκτίμηση της προσπάθειας και του κόστους για την ολοκλήρωση του έργου, καθώς και το σχεδιασμό ανάπτυξής του.
2. **Εργαλεία μοντελοποίησης και διαχείρισης των επιχειρηματικών διεργασιών (Process Modeling and Management tools).** Τα εργαλεία της κατηγορίας αυτής χρησιμοποιούνται για την παράσταση των χαρακτηριστικών μιας επιχειρηματικής διεργασίας, καθώς και για τη λεπτομερή περιγραφή της, έτσι ώστε να γίνονται χρήσιμα στην κατανόηση της και στην επιτυχημένη ένταξή της μέσα σ' ένα ευρύτερο πλαίσιο (integration).
3. **Εργαλεία ανάλυσης κινδύνου (Risk Analysis tools).** Με τη βοήθειά τους αναγνωρίζονται οι πιθανοί κίνδυνοι κατά την ανάπτυξη του λογισμικού και αναπτύσσεται ένα σχέδιο παρακολούθησης και αντιμετώπισής τους.
4. **Εργαλεία διοίκησης έργου (Project Management tools).** Το σχέδιο εκτέλεσης ενός έργου ανάπτυξης λογισμικού οφείλει να καταγράφεται και να παρακολουθείται σε συνεχή βάση. Επίσης, ο υπεύθυνος της ανάπτυξης θα πρέπει να συλλέγει μετρήσεις οι οποίες θα του παρέχουν ενδείξεις για την πρόοδο και ενδεχομένως για την ποιότητα του αναπτυσσόμενου προϊόντος λογισμικού. Τα εργαλεία της κατηγορίας αυτής υποστηρίζουν τέτοιες εργασίες και συνήθως είναι επεκτάσεις εργαλείων σχεδιασμού του έργου.

5. **Εργαλεία μετρικών λογισμικού (Metrics tools).** Χρησιμοποιούνται ως μέσα για την ποσοτική αποτίμηση χαρακτηριστικών του λογισμικού και την τεκμηρίωση της ποιότητας αυτού. Τα εργαλεία της κατηγορίας αυτής υποστηρίζουν την εφαρμογή μετρικών τόσο στην ανάπτυξη, όσο και στα ίδια τα συστατικά λογισμικού.
6. **Εργαλεία διασφάλισης ποιότητας (Quality Assurance tools).** Τα περισσότερα εργαλεία, τα οποία χρησιμοποιούνται για να υποστηρίξουν την εργασία διασφάλισης ποιότητας, είναι ουσιαστικά εργαλεία μετρήσεων, τα οποία αναλύουν πηγαίο κώδικα για τον προσδιορισμό του ποσοστού συμμόρφωσής του με τα επιθυμητά από τον κατασκευαστή πρότυπα για τη γλώσσα προγραμματισμού στην οποία είναι γραμμένος.
7. **Εργαλεία διαχείρισης βάσεων δεδομένων λογισμικού (Software Database Management tools).** Αυτά εξυπηρετούν τη δημιουργία και διατήρηση μιας βάσης δεδομένων σχετικά με το ίδιο το λογισμικό.
8. **Εργαλεία διοίκησης σχηματισμών λογισμικού (Software Configuration Management tools).** Τα εργαλεία της κατηγορίας αυτής εντοπίζονται στον πυρήνα κάθε περιβάλλοντος CASE. Με τη βοήθειά τους μπορούν να πραγματοποιηθούν οι πέντε ακόλουθες σημαντικές εργασίες διοίκησης σχηματισμών λογισμικού: αναγνώριση, έλεγχος της έκδοσης, αλλαγή του ελέγχου, σύγκριση και καταγραφή της τρέχουσας κατάστασης. Η βάση δεδομένων του έργου παρέχει ένα μηχανισμό για την αναγνώριση κάθε αντικειμένου της διάταξης και τη συσχέτισή του με άλλα αντικείμενα.
9. **Εργαλεία ενοποίησης και ελέγχου (Integration and Testing tools).** Στην κατηγορία αυτή διακρίνουμε τα παρακάτω εργαλεία:
 - Ø **Ανάκτησης δεδομένων:** Τα εργαλεία αυτά ανακτούν δεδομένα τα οποία χρησιμοποιούνται κατά τη διαδικασία του ελέγχου.
 - Ø **Στατικών μετρήσεων:** Τα εργαλεία αυτά αναλύουν τον πηγαίο κώδικα χωρίς να εκτελούν τις περιπτώσεις ελέγχου.
 - Ø **Δυναμικών μετρήσεων:** Τα εργαλεία αυτά αναλύουν τον πηγαίο κώδικα κατά τη διαδικασία του ελέγχου.
 - Ø **Εξομοίωσης:** Τα εργαλεία αυτά εξομοιώνουν λειτουργίες του υλικού ή άλλων εξωτερικών μονάδων.
 - Ø **Διαχείρισης του ελέγχου:** Τα εργαλεία αυτά βοηθούν στο σχεδιασμό, την ανάπτυξη και τη διαχείριση του ελέγχου.
10. **Εργαλεία διαχείρισης ελέγχου (Test Management tools).** Χρησιμοποιούνται για το συντονισμό ολόκληρης της διεργασίας ελέγχου του υπό ανάπτυξη λογισμικού. Επιπροσθέτως, πολλά εργαλεία του είδους χρησιμοποιούνται ως γενικοί οδηγοί ελέγχου. Ένας τέτοιος οδηγός, διαβάζει μια ή περισσότερες περιπτώσεις ελέγχου από ένα αρχείο και διαμορφώνει τα δεδομένα ελέγχου έτσι ώστε να συμφωνούν με τις ανάγκες της μονάδας λογισμικού που βρίσκεται υπό έλεγχο και στη συνέχεια την εκτελεί.

- 11.Εργαλεία ελέγχου συστημάτων πελάτη – εξυπηρετητή (Client Server Testing tools).** Το περιβάλλον πελάτη – εξυπηρετητή απαιτεί ειδικευμένα εργαλεία ελέγχου τα οποία εξετάζουν τη λειτουργία του γραφικού περιβάλλοντος του χρήστη και τη δικτυακή επικοινωνία μεταξύ πελατών και εξυπηρετητών.
- 12.Εργαλεία αντίστροφης ανάπτυξης σε προδιαγραφές (reverse engineering to specification tools):** Τα εργαλεία αυτά παίρνουν σαν είσοδο πηγαίο κώδικα και δίνουν ως έξοδο γραφικά μοντέλα δομημένης ανάλυσης, λίστες που περιγράφουν τι χρησιμοποιείται και πού, καθώς και άλλες πληροφορίες σχεδιασμού που μπορούν να εξαχθούν μόνο από πηγαίο κώδικα.
- 13.Εργαλεία ανακατασκευής και ανάλυσης του κώδικα (code restructuring and analysis tools):** Τα εργαλεία αυτά αναλύουν τη σύνταξη του πηγαίου κώδικα και «γεννούν» ένα διάγραμμα ροής, καθώς και μια δομημένη εκδοχή του πηγαίου κώδικα.
- 14.Εργαλεία ανακατασκευής σε σύνδεση με το σύστημα (on–line system reengineering tools):** Τα εργαλεία αυτά χρησιμοποιούνται για την on–line τροποποίηση συστημάτων βάσεων δεδομένων.

Τα περισσότερα από τα παραπάνω εργαλεία περιορίζονται στη χρήση συγκεκριμένων γλωσσών προγραμματισμού, απαιτούν πολύ καλή γνώση από το μηχανικό λογισμικού και, ασφαλώς, έχουν αποτελέσματα μόνο υπό συνθήκες.

∅ **Εργαλεία προδιαγραφής απαιτήσεων, ανάλυσης και σχεδίασης λογισμικού.**
Στην κατηγορία μπορούμε να διακρίνουμε τις εξής υποκατηγορίες:

- 1. Εργαλεία διαχείρισης προδιαγραφών (Requirements Management tools).** Ο σκοπός τους είναι να παρέχουν μια συστηματική προσέγγιση στον προσδιορισμό και την καταγραφή των προδιαγραφών, ξεκινώντας από τις απαιτήσεις του πελάτη. Ένα τυπικό εργαλείο του είδους, συνδυάζει είσοδο από το χρήστη με κάποια βάση δεδομένων όπου αποθηκεύει και κατηγοριοποιεί κάθε προδιαγραφή του συστήματος.
- 2. Εργαλεία ανάλυσης και σχεδίασης (Analysis and Design tools).** Βοηθούν το μηχανικό λογισμικού στη δημιουργία μοντέλων για το υπό ανάπτυξη σύστημα. Τα μοντέλα αυτά περιλαμβάνουν αναπαράσταση των δεδομένων, της λειτουργίας και της συμπεριφοράς τους, καθώς και σχέδια διαπροσωπειών, δεδομένων, λειτουργικών μονάδων ή της αρχιτεκτονικής του συστήματος. Κατά την εκτέλεση ελέγχων σε ένα τέτοιο μοντέλο, τα εργαλεία ανάλυσης και σχεδίασης παρέχουν στο μηχανικό λογισμικού μια εικόνα του τελικού προϊόντος, βοηθώντας στη διόρθωση των σφαλμάτων πριν αυτά παγιωθούν κατά την φάση της κωδικοποίησης και, ασφαλώς, υποστηρίζοντας την ικανοποίηση των απαιτήσεων.

∅ **Εργαλεία κατασκευής και γέννησης προγραμμάτων.** Στην κατηγορία μπορούμε να διακρίνουμε τις εξής υποκατηγορίες:

1. **Εργαλεία προγραμματισμού (Programming tools).** Στην κατηγορία αυτή ανήκουν εργαλεία, όπως οι μεταγλωττιστές (compilers), οι εκδότες προγραμμάτων (editors) και οι διορθωτές. Τα εργαλεία αυτά συνήθως υποστηρίζονται από το περιβάλλον μιας γλώσσας προγραμματισμού. Επίσης, στην κατηγορία αυτή κατατάσσονται εργαλεία όπως αντικειμενοστραφή περιβάλλοντα προγραμματισμού, γλώσσες προγραμματισμού τέταρτης γενιάς, γραφικά περιβάλλοντα προγραμματισμού, γεννήτριες εφαρμογών και γλώσσες αναζήτησης σε βάσεις δεδομένων.
2. **Εργαλεία σχεδίασης και ανάπτυξης διαπροσωπειών (Interface Design and Development tools).** Είναι ουσιαστικά συλλογές προγραμματιστικών πόρων όπως μενού, κουμπιά, παράθυρα, εικόνες, θυρίδες ολίσθησης, οδηγοί μονάδων κ.ά., οι οποίες χρησιμοποιούνται από την περιγραφή έως και τη μερική γέννηση των συστατικών της διαπροσωπείας του λογισμικού με το χρήστη.

∅ **Εργαλεία ανάλυσης και ελέγχου του πηγαίου κώδικα.** Στην κατηγορία αυτή μπορούμε να διακρίνουμε τις εξής υποκατηγορίες:

1. **Εργαλεία στατικής ανάλυσης (Static Analysis tools).** Τα εργαλεία στατικού ελέγχου βοηθούν το μηχανικό λογισμικού στην εύρεση περιπτώσεων ελέγχου. Σήμερα, εμφανίζονται τρεις διαφορετικοί τύποι της κατηγορίας αυτής: τα εργαλεία ελέγχου που βασίζονται στον πηγαίο κώδικα, οι εξειδικευμένες γλώσσες ελέγχου και τα εργαλεία ελέγχου που βασίζονται στις προδιαγραφές.
2. **Εργαλεία δυναμικής ανάλυσης (Dynamic Analysis tools).** Τα εργαλεία αυτά αλληλεπιδρούν με ένα πρόγραμμα το οποίο εκτελείται, ελέγχουν το μονοπάτι εκτέλεσής του, την εισαγωγή διαφόρων τιμών συγκεκριμένων μεταβλητών και καθοδηγούν γενικότερα την εκτέλεση του προγράμματος που βρίσκεται υπό έλεγχο.

∅ **Εργαλεία προσομοίωσης και μοντελοποίησης** Στη κατηγορία μπορούμε να διακρίνουμε τις εξής υποκατηγορίες:

1. **Εργαλεία προσομοίωσης (Simulation tools).** Τα εργαλεία αυτά βοηθούν το μηχανικό λογισμικού να αντιμετωπίσει την πραγματική συμπεριφορά ενός συστήματος πριν ακόμη αυτό υλοποιηθεί, προσομοιώνοντάς την, όσο αυτό είναι δυνατό. Συχνά χρησιμοποιούνται σαν μέσο επικοινωνίας μεταξύ του πελάτη και του κατασκευαστή, ώστε να αποκτηθεί κοινή αντίληψη για το υπό ανάπτυξη λογισμικό.

2. Εργαλεία προτυποποίησης (Prototyping tools). Τα εργαλεία της κατηγορίας αυτής συνήθως παρέχονται ως επεκτάσεις σε εργαλεία CASE άλλων κατηγοριών.

∅ **Εργαλεία τεκμηρίωσης (Documentation tools).** Οι περισσότερες εταιρείες κατασκευής λογισμικού «ξοδεύουν» ένα σημαντικό ποσοστό του χρόνου ανάπτυξης του προϊόντος στη συγγραφή εγγράφων τεκμηρίωσης του λογισμικού. Η εργασία αυτή είναι δύσκολη, χρονοβόρα και σπάνια εκτελείται σωστά, ώστε η τεκμηρίωση να ανταποκρίνεται στην τρέχουσα έκδοση του λογισμικού. Τα εργαλεία τεκμηρίωσης υποστηρίζουν την εργασία αυτή και παρέχουν τη δυνατότητα βελτίωσης της παραγωγικότητάς της.

Ταξινόμηση σύμφωνα με την εμβέλεια

Αν θέσουμε ως κριτήριο την εμβέλεια υποστήριξης της διαδικασίας ανάπτυξης του λογισμικού, καταλήγουμε σε τρεις κύριες κατηγορίες:

∅ Τα **εργαλεία (tools)** τα οποία υποστηρίζουν μεμονωμένες εργασίες, όπως ο έλεγχος της ορθότητας ενός σχεδίου, η μετάφραση ενός προγράμματος, η σύγκριση των αποτελεσμάτων ενός ελέγχου κ.ά. Τα tools μπορεί να είναι γενικού – σκοπού, ανεξάρτητα εργαλεία (όπως ένας επεξεργαστής κειμένου) ή μπορεί να είναι ομαδοποιημένα.

∅ Οι **πάγκοι εργασίας (workbenches)**, οι οποίοι υποστηρίζουν συγκεκριμένες φάσεις της διαδικασίας ανάπτυξης του λογισμικού, όπως ο ορισμός των προδιαγραφών, η σχεδίαση κ.ά. Συνήθως αποτελούνται από ένα σύνολο εργαλείων με κάποιο βαθμό ενοποίησης.

∅ Τα **ολοκληρωμένα περιβάλλοντα (integrated environments)**, τα οποία υποστηρίζουν ολόκληρη ή τουλάχιστον ένα σημαντικό τμήμα της διαδικασίας ανάπτυξης του λογισμικού. Συνήθως περιέχουν έναν αριθμό διαφόρων πάγκων εργασίας, οι οποίοι με κάποιο τρόπο ενοποιούνται και συνεργάζονται.

2.6.3 Αξιολόγηση και Επιλογή Εργαλείων CASE.

Η εισαγωγή ενός κατασκευαστή λογισμικού στην τεχνολογία CASE και η χρήση των αντίστοιχων εργαλείων, απαιτεί πολύ προσεκτικό σχεδιασμό. Τα εργαλεία CASE είναι ιδιαίτερα ακριβά και πρέπει να εξασφαλίζεται τουλάχιστον η απόδοση της επένδυσης που απαιτούν. Η απόδοση αυτή δεν είναι πάντα δεδομένη.

Μια όχι πολύ καλή εφαρμογή λογισμικού, μπορεί να επιστρέψει στο μέσο χρήστη την επένδυσή της περισσότερο εύκολα από ότι μπορεί να το κάνει για έναν κατασκευαστή λογισμικού ένα μη κατάλληλο εργαλείο CASE.

Η λήψη λανθασμένων αποφάσεων και γενικά οι μη ακριβείς εκτιμήσεις σχετικά με ένα εργαλείο CASE, είναι πιθανό να έχει τα αντίθετα από τα αναμενόμενα αποτελέσματα.

Δηλαδή, μπορεί να μειώνει την παραγωγικότητα και την ποιότητα και να αυξάνει το κόστος παραγωγής του λογισμικού.

Δεν υπάρχει καλύτερος σύμβουλος στην επιλογή εργαλείων CASE από την γνώση, τη σύνεση και το ρεαλισμό. Για παράδειγμα, ένα εργαλείο που «κάνει τα πάντα» μπορεί να μην τρέχει στην υπάρχουσα πλατφόρμα ανάπτυξης του κατασκευαστή ή/και να έχει μεγάλο κόστος. Το συμπέρασμα είναι το αναμενόμενο, ότι δηλαδή η επιλογή του καλύτερου είναι σχετική υπόθεση.

ΚΕΦΑΛΑΙΟ 3

ΔΗΜΙΟΥΡΓΙΑ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ Τ.Ε.Ι. ΠΑΤΡΑΣ

Μετά από εκτενή αναφορά σε όλα εκείνα τα θεωρητικά στοιχεία που ήταν απαραίτητα για την κατανόηση και τον σχεδιασμό του Συστήματός μας, προχωρούμε στην ανάπτυξη της εφαρμογής μας. Στο τελευταίο αυτό κεφάλαιο θα περιγραφεί βήμα προς βήμα όλη η διαδικασία σχεδιασμού του πληροφοριακού μας συστήματος, δημιουργώντας μία βάση δεδομένων με την βοήθεια του προγράμματος MS ACCESS 2007. Βασισμένοι στην Εφαρμογή, θα δημιουργηθούν Ερωτήματα, Εκθέσεις και Φόρμες για την καλύτερη δυνατή χρήση της Βάσης Δεδομένων.



3.1 Γενικές Παρατηρήσεις Για Την Βάση Δεδομένων Που Θα Δημιουργήσουμε

Η φάση του σχεδιασμού είναι **βασικό τμήμα του κύκλου ζωής** ενός συστήματος λογισμικού, ειδικότερα δε μίας βάσης δεδομένων. Βασικός στόχος της φάσης αυτής είναι η κατασκευή ενός άτυπου μοντέλου, κατάλληλου για την περιγραφή των πληροφοριών τις οποίες διαχειρίζεται η βάση μας και ανεξάρτητο από λεπτομέρειες υλοποίησης (όπως ο τρόπος αποθήκευσης αρχείων ή πλατφόρμα όπου εκτελούνται οι βασικές εφαρμογές του ΣΔΒΔ).

Θα προσπαθούμε, με τα δεδομένα που εμφανίζονται συνολικά, να οδηγηθούμε σε οντότητες και σε συσχετίσεις. Αυτό εξυπηρετεί τη φάση της σχεδίασης, αφού δείχνει με τρόπο σαφή τις αναλογίες που πρέπει να έχει η βάση μας με το σύστημα του πραγματικού κόσμου που προσπαθούμε να περιγράψουμε.

Το κυρίαρχο εννοιολογικό μοντέλο που χρησιμοποιείται είναι το μοντέλο **Οντοτήτων – Συσχετίσεων (ΟΣ)**. Οι βασικές έννοιες που περιλαμβάνει το μοντέλο ΟΣ είναι οι οντότητες, οι ιδιότητες και οι συσχετίσεις. Για την αναπαράσταση του χρησιμοποιούνται τα διαγράμματα ΟΣ. Την κατασκευή του διαγράμματος ΟΣ ακολουθεί η υλοποίηση της λογικής δομής της βάσης. Συμφωνά με το σχεσιακό μοντέλο όλες οι πληροφορίες καταγράφονται στη βάση μας ως πίνακες ή αλλιώς ως σχέσεις. Οι οντότητες που προκύπτουν από το μοντέλο ΟΣ μεταφράζονται συνήθως σε μεγάλες σχέσεις.

Οι βασικές έννοιες που χρησιμοποιούνται ευρέως είναι:

Ø **Οντότητα** είναι κάθε αντικείμενο, πρόσωπο, γεγονός, κατάσταση ή και αφηρημένη έννοια, που προσδιορίζεται από την ανεξάρτητη ύπαρξη του. Είναι, δηλαδή, κάτι που υπάρχει, μπορούμε να το διακρίνουμε και ενδιαφερόμαστε να συλλέξουμε πληροφορίες για αυτό. Οι οντότητες απεικονίζονται διαγραμματικά στο μοντέλο ΟΣ σαν ορθογώνια παραλληλόγραμμα τα οποία φέρουν ετικέτα με το όνομα τους.

Ø **Ιδιότητες ή Χαρακτηριστικά** είναι τα συστατικά στοιχεία που περιγράφουν μια οντότητα ή μια συσχέτιση. Οι ιδιότητες αντιστοιχούν στην έννοια του πεδίου που αναφέρεται στα αρχεία αλλά και στην έννοια της μεταβλητής σε μία γλώσσα προγραμματισμού. Στο διάγραμμα ΟΣ οι ιδιότητες απεικονίζονται με ελλειπτικά (οβάλ) σχήματα, που συνδέονται με τις αντίστοιχες οντότητες με μία γραμμή.

Ø **Κλειδιά** . Αν και η λέξη «κλειδιά» είναι ευρύτερη έννοια, η οποία έχει ήδη χρησιμοποιηθεί σε πολλές εφαρμογές που έχουν αναπτυχθεί με γλώσσες 3ης γενιάς, θα τις αναλύσουμε με βάση το σχεσιακό μοντέλο.

Τα κλειδιά διαχωρίζονται σε Πρωτεύοντα και Ξένα :

Ø **Πρωτεύον Κλειδί** είναι το πεδίο ή ένα σύνολο πεδίων που η τιμή του ή ο συνδυασμός των τιμών τους κάνουν κάθε εγγραφή του πίνακα μοναδική. Δηλαδή,

το πρωτεύον κλειδί έχει διαφορετική τιμή για κάθε γραμμή ενός πίνακα και δεν μπορεί ποτέ να πάρει μια μηδενική τιμή ή κενό (null).

Ø **Ξένο Κλειδί** ονομάζεται το πεδίο ενός πίνακα που είναι ίδιο με το Πρωτεύον Κλειδί κάποιου άλλου πίνακα. Το ξένο κλειδί εμφανίζεται, όταν υλοποιούμε συσχετίσεις σε μία βάση.

Η χρησιμοποίηση Βάσεων Δεδομένων(Databases) εξασφαλίζει ένα γενικό τρόπο αποθήκευσης των δεδομένων. Τα δεδομένα είναι ολοκληρωμένα (integrated) και καταμερισμένα (shared) έτσι ώστε, αφενός μεν τα πλεονάζοντα (redundant) δεδομένα να αποθηκεύονται όσο το δυνατόν λιγότερες φορές, αφετέρου δε να είναι προσπελάσιμα, από διάφορους χρήστες, για ποικίλες εφαρμογές. Βασικά πλεονεκτήματα από την οργάνωση των δεδομένων της Βάσης με τον τρόπο αυτό είναι ότι μειώνεται ο χώρος αποθήκευσης στο δίσκο καθώς ο χρόνος για ενημέρωση των δεδομένων.

Για την υλοποίηση του πληροφοριακού συστήματος που θα δημιουργήσουμε , χρησιμοποιούμε όπως έχουμε ήδη αναφέρει, το πρόγραμμα **Microsoft Office Access 2007** της εταιρείας Microsoft.



3.2 Δημιουργία Πινάκων ΤΕΙ Πάτρας

Με την είσοδό μας στο περιβάλλον της Access, προχωρούμε στην δημιουργία των βασικών πινάκων, που θα πρέπει να περιλαμβάνει η βάση του Πληροφοριακού μας Συστήματος. Οι Πίνακες αυτοί, δημιουργούνται σε **Προβολή Σχεδίασης** της Access. Κάθε Πίνακας αποτελείται από ορισμένα πεδία ένα ή περισσότερα εκ των οποίων χαρακτηρίζονται ως «Πρωτεύοντα Κλειδιά».

Επισημαίνοντας τις παραπάνω έννοιες, ξεκινάμε τον προσδιορισμό των Οντοτήτων και των Ιδιοτήτων τους, που αποτελούν τα απαραίτητα στοιχεία για την δημιουργία μιας Βάσης Δεδομένων.

Έτσι κρίνουμε απαραίτητο να δημιουργηθούν οι παρακάτω βασικοί πίνακες:

«**ΚΑΘΗΓΗΤΕΣ**»

«**ΦΟΙΤΗΤΕΣ**»

«**ΜΑΘΗΜΑΤΑ ΤΜΗΜΑΤΟΣ**»

«**ΚΛΑΔΟΣ ΣΧΟΛΗΣ**»

«**ΤΜΗΜΑΤΑ ΤΕΙ**»

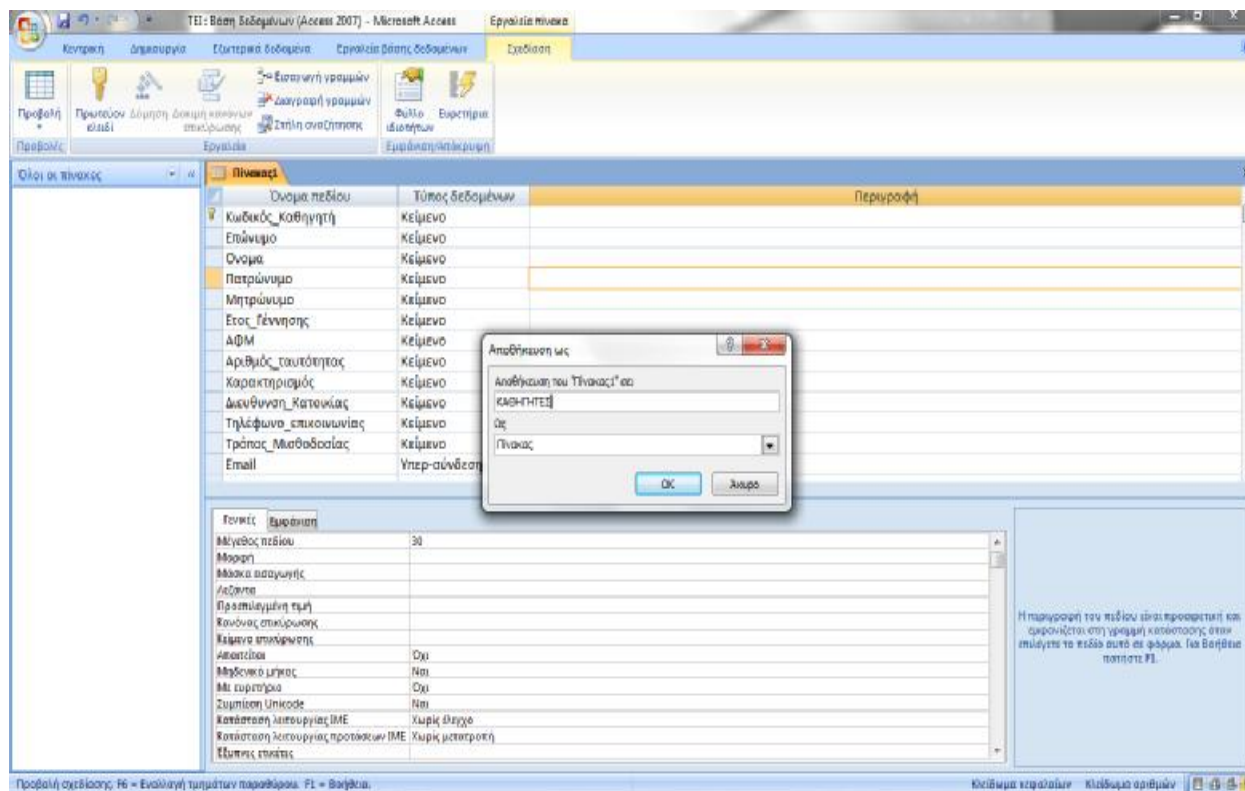
«**ΓΡΑΜΜΑΤΕΙΑ ΤΜΗΜΑΤΟΣ**»

«**ΥΠΑΛΛΗΛΟΙ ΓΡΑΜΜΑΤΕΙΑΣ**»

Θέλοντας αρχικά να δημιουργήσουμε τον Πίνακα «**ΚΑΘΗΓΗΤΕΣ**» επιλέγουμε στο περιβάλλον της Access επιλέγουμε την Επιλογή «Δημιουργία», κατόπιν το αντικείμενο «Πίνακας» και το «Δημιουργία Πεδίου» όπως απεικονίζεται στα Σχήματα 3.1 και 3.2.



Σχήμα 3.1 Δημιουργία πίνακα.



Σχήμα 3.2 Δημιουργία πίνακα «ΚΑΘΗΓΗΤΕΣ» σε Προβολή Σχεδίασης.

Στην Προβολή Σχεδίασης, εισάγουμε τα πεδία του Πίνακα, τον τύπο δεδομένων καθώς επίσης και την περιγραφή, όπου αυτή απαιτείται, όπως φαίνεται στο παραπάνω Σχήμα 3.2. Κατά τον ίδιο τρόπο δημιουργούμε και τους υπόλοιπους Βασικούς Πίνακες που θα χρησιμοποιήσουμε στην Βάση μας, και οι όποιοι περιέχουν με την σειρά τους τα εξής πεδία:

«ΦΟΙΤΗΤΕΣ»: Αριθμός Μητρώου Φοιτητή (αποτελεί το Πρωτεύον Κλειδί), Επώνυμο, Όνομα, Πατρώνυμο, Μητρώνυμο, Τόπος Γέννησης, Αριθμός Ταυτότητας, Τηλέφωνο Επικοινωνίας, Email, Σπουδαστικό Έτος Εγγραφής, Τρέχον Εξάμηνο, Κωδικός Τμήματος (το οποίο αργότερα χρησιμοποιήθηκε ως ξένο κλειδί).

«ΚΛΑΔΟΣ ΣΧΟΛΗΣ»: Κωδικός Σχολής (αποτελεί το Πρωτεύον Κλειδί), Τίτλος Σχολής.

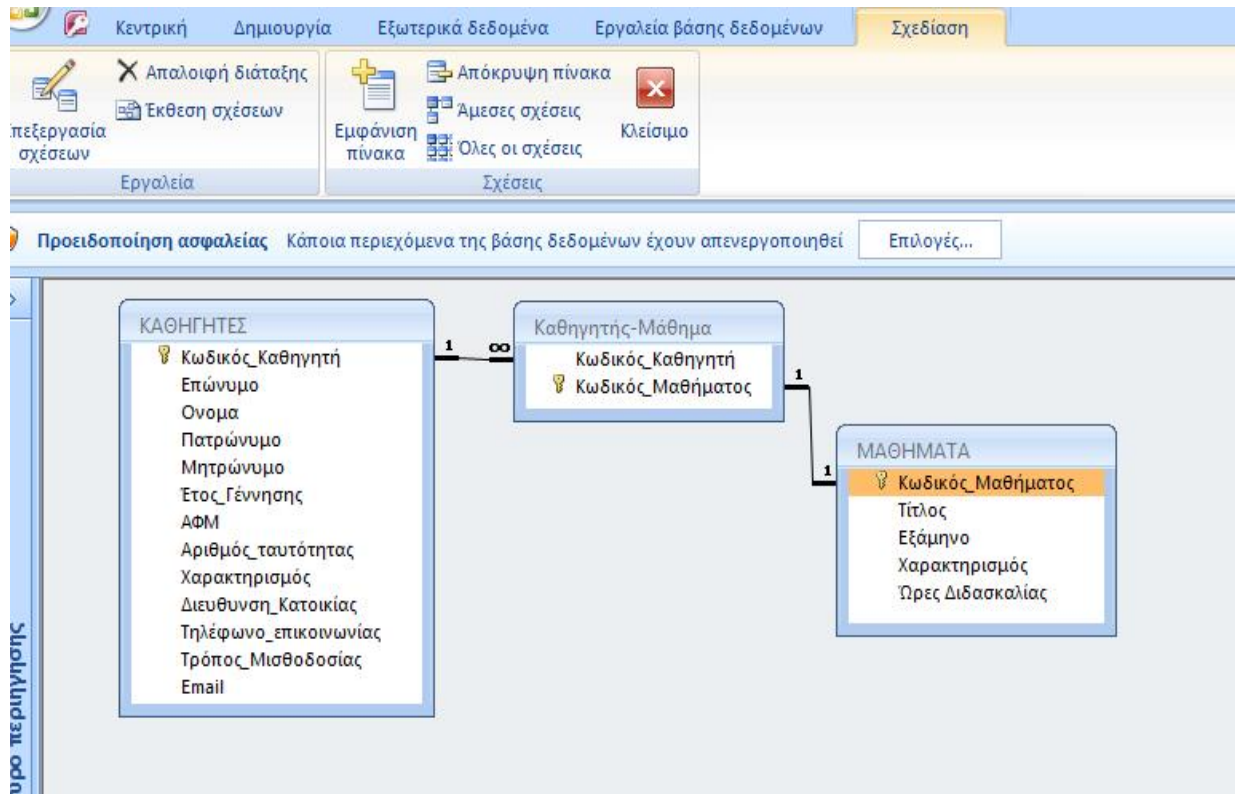
«ΤΜΗΜΑΤΑ ΤΕΙ»: Κωδικός Τμήματος (αποτελεί το Πρωτεύον Κλειδί), Τίτλος Τμήματος, Κωδικός Σχολής(το οποίο αργότερα χρησιμοποιήθηκε ως ξένο κλειδί), Παράρτημα.

«ΥΠΑΛΛΗΛΟΙ ΓΡΑΜΜΑΤΕΙΑΣ»: Κωδικός Υπάλληλου (αποτελεί το Πρωτεύον Κλειδί), Επώνυμο, Όνομα, Πατρώνυμο, Ειδικότητα, Τηλέφωνο Επικοινωνίας, Τρόπος Μισθοδοσίας.

«ΓΡΑΜΜΑΤΕΙΑ ΤΜΗΜΑΤΟΣ»: Κωδικός Γραμματείας (αποτελεί το Πρωτεύον Κλειδί), Τίτλος Τμήματος, Τηλέφωνο Επικοινωνίας, Fax, Email, Δικτυακός Τόπος.

«ΜΑΘΗΜΑΤΑ»: Κωδικός Μαθήματος (αποτελεί το Πρωτεύον Κλειδί), Τίτλος Μαθήματος, Εξάμηνο, Χαρακτηρισμός, Ώρες Διδασκαλίας.

Κατόπιν θα έπρεπε να προχωρούμε στην Σύνδεση των Βασικών Πινάκων. Ωστόσο προκειμένου να γίνει αυτό, απαιτείται η δημιουργία και η χρήση Βοηθητικών Πινάκων (π.χ. Καθηγητές- Μάθημα), έτσι ώστε να υλοποιηθούν οι Σχέσεις μεταξύ αυτών, όπως φαίνεται ενδεικτικά στο Σχήμα 3.3.



Σχήμα 3.3 Σύνδεση Βασικών Πινάκων με χρήση Βοηθητικού Πίνακα.

Μελετώντας την Βάση μας, παρατηρούμε ότι πρέπει να δημιουργηθούν οι εξής Βοηθητικοί Πίνακες :

«Καθηγητής- Μάθημα»: Κωδικός Καθηγητή, Κωδικός Μαθήματος (Αποτελεί Πρωτεύον Κλειδί).

«Σχολές- Τμήματα»: Κωδικός Σχολής, Κωδικός Τμήματος (Αποτελούν και τα δυο αυτά πεδία συνδυασμό Πρωτευόντων Κλειδιών).

«Γραμματεία - Υπάλληλοι»: Κωδικός Γραμματείας, Κωδικός Υπαλλήλου (Αποτελούν και τα δυο αυτά πεδία συνδυασμό Πρωτευόντων Κλειδιών).

«Τμήμα ΤΕΙ - Γραμματεία Τμήματος»: Κωδικός Γραμματείας, Κωδικός Τμήματος (Αποτελούν και τα δυο αυτά πεδία συνδυασμό Πρωτευόντων Κλειδιών).

«Τμήμα ΤΕΙ - Καθηγητής»: Κωδικός Τμήματος, Κωδικός Καθηγητή (Αποτελούν και τα δυο αυτά πεδία συνδυασμό Πρωτευόντων Κλειδιών).

«Φοιτητής – Μάθημα»: Αριθμός Μητρώου Φοιτητή, Κωδικός Μαθήματος (Αποτελούν και τα δυο αυτά πεδία συνδυασμό Πρωτευόντων Κλειδιών), Βαθμολογία Μαθήματος.

3.3 Δημιουργία Σχέσεων μεταξύ των Πινάκων ΤΕΙ Πάτρας

Το επόμενο βήμα για την υλοποίηση της Βάσης μας είναι η **δημιουργία Σχέσεων** μεταξύ των Πινάκων μας. Για να μπορέσουμε να δημιουργήσουμε τις Σχέσεις πρέπει από την Γραμμή των Εργαλείων να επιλέξουμε το εικονίδιο «**Εργαλεία Βάσης Δεδομένων**» και κατόπιν το εικονίδιο «**Σχέσεις**».

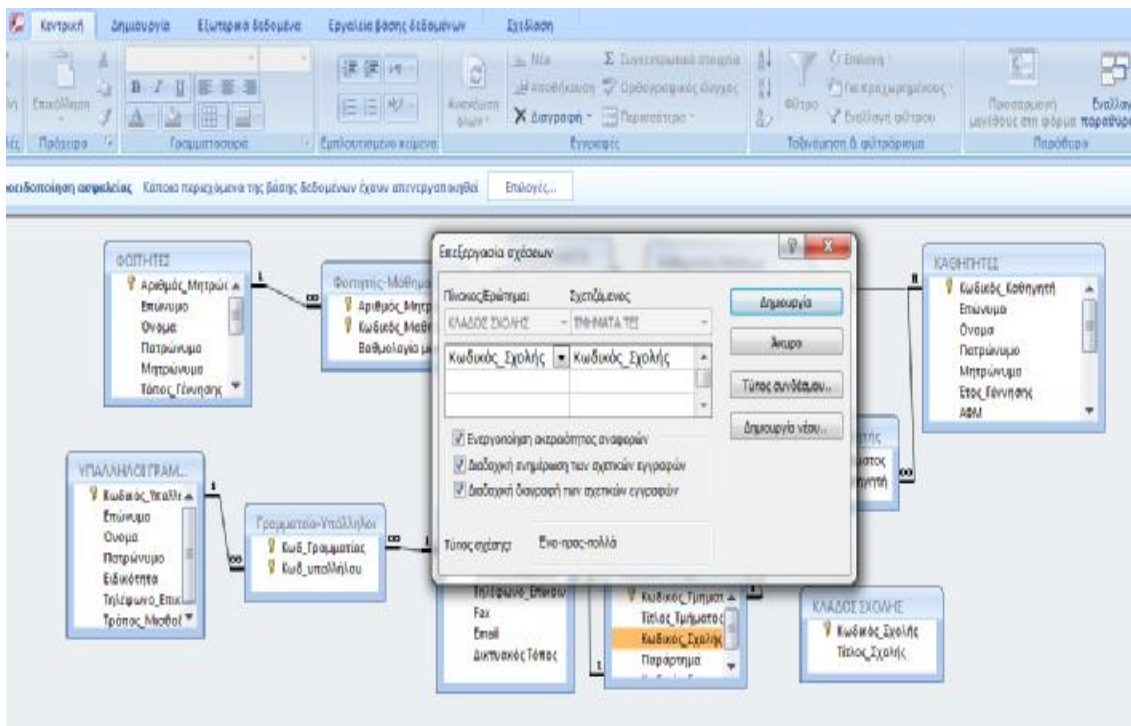
Αφού επιλέξουμε το εικονίδιο «Σχέσεις», κάνοντας δεξί κλικ, εμφανίζουμε όλους τους πίνακες της Βάσης δεδομένων και δημιουργούμε τις μεταξύ τους Σχέσεις.

Για να δημιουργηθούν αυτές οι Σχέσεις επιλεγούμε το Πρωτεύον Κλειδί ενός Πίνακα και κρατώντας το πατημένο, το σύρουμε στο αντίστοιχο πεδίο του Πίνακα που θέλουμε να δημιουργήσουμε την Σχέση.

Έτσι εμφανίζεται Παράθυρο στο οποίο τσεκάρουμε τα:

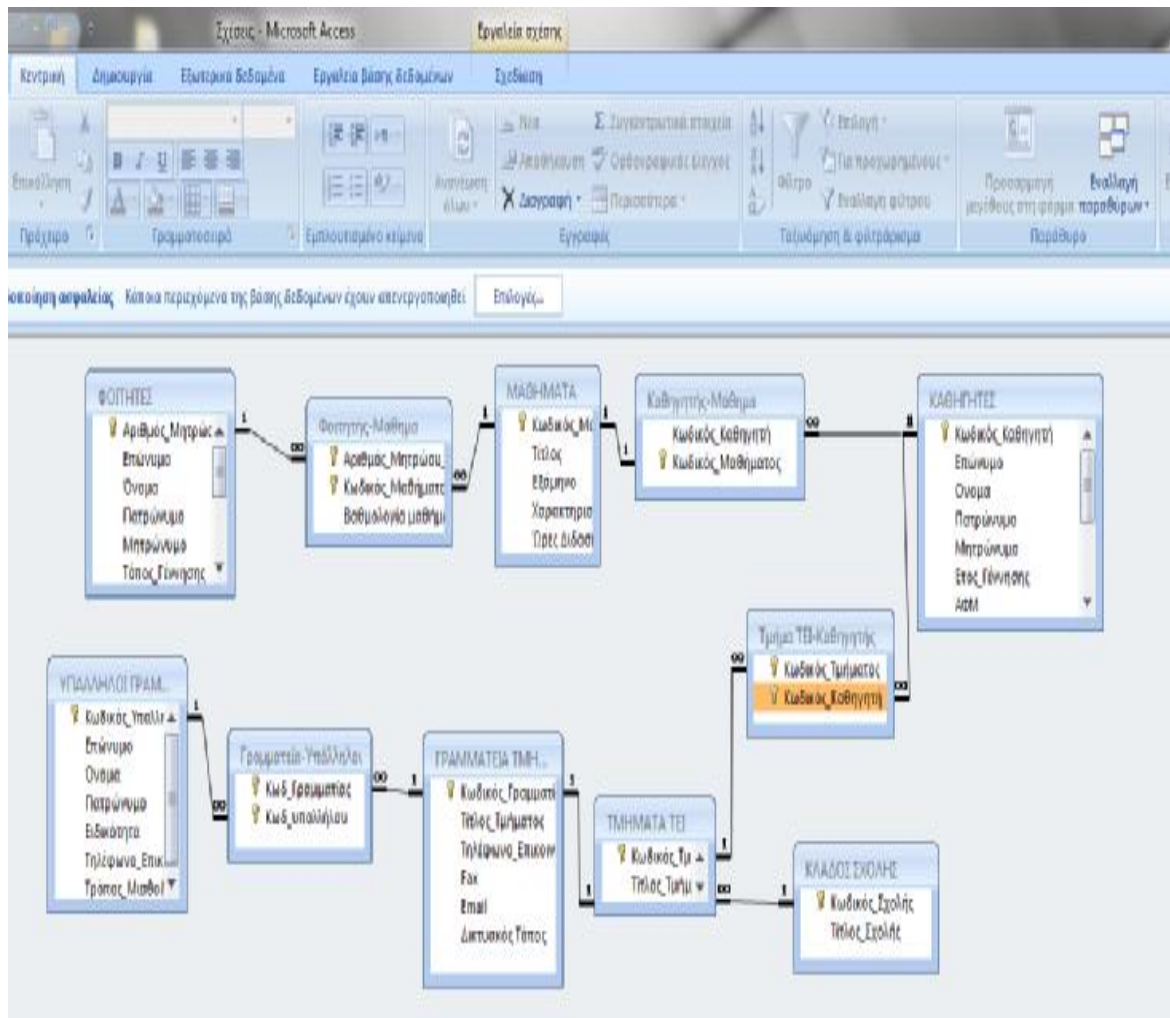
- ü Ενεργοποίηση Ακεραιότητας Αναφοράς
- ü Διαδοχική Ενημέρωση των Σχετικών Εγγραφών
- ü Διαδοχική Διαγραφή των Σχετικών Εγγραφών

Για να επιλέξουμε τον Τύπο Σχέσης που καλύπτει τις απαιτήσεις που έχουμε από το Σύστημα που θέλουμε να δημιουργήσουμε, επιλέγουμε το «**Τύπο Συνδέσμου**». Εκεί βρίσκουμε τις Σχέσεις «**Ένα προς Ένα**», «**Ένα προς Πολλά**», «**Πολλά προς Πολλά**». Τέλος επιλέγουμε το «**Δημιουργία**» και η σχέση έχει ήδη δημιουργηθεί (Σχήμα 3.4).



Σχήμα 3.4 Δημιουργία Σχέσεων Μεταξύ Των Πινάκων Της Βάσης.

Αφού πραγματοποιήσουμε παραπάνω τα βήματα για κάθε Πίνακα, φτάνουμε στο αποτέλεσμα του Σχήματος 3.5 στο οποίο φαίνονται ολοκληρωμένες οι Σχέσεις που έχουν δημιουργηθεί μεταξύ των Πινάκων.



Σχήμα 3.5 Σχέσεις Μεταξύ Των Πινάκων.

Σε αυτό το σημείο έχουμε δημιουργήσει και ολοκληρώσει την Βάση μας και μπορούμε να αρχίσουμε την εισαγωγή δεδομένων στους Πίνακές μας. Πλέον στη Βάση, μπορούμε να αποθηκεύσουμε, να επεξεργαστούμε και να προσθέσουμε όσα δεδομένα χρειαζόμαστε με άνεση και σε ένα εύχρηστο και φιλικό περιβάλλον για κάθε χρήστη.

Κωδικός_Κα -	Επώνυμο -	Όνομα -	Πατρώνυμο -	Μητρώνυμο -	Έτος_Γέννησ -	ΑΦΜ -	Αριθμός_ταυτότη -	Χαρακτηρισμό -	Διευθυνση_Κατου -	Τηλέφωνο_επικοινων -	Τρόπος_Μισθο
0001	Ραφαηλίδης	Απόστολος	Νικόλαος	Μαρία	1964	123852963	T285296	ΜΟΝΙΜΟΣ	Αγ. Νικολάου 45	2610859612	Τραπ. Λογ/σμο
0002	Κουνετάς	Κωνσταντίνος	Παναγιώτης	Ευαγγελία	1975	239648523	AE52263	ΜΟΝΙΜΟΣ	Καραϊσκάκη 152	2610485296	Τραπ. Λογ/σμο
0003	Μητρόπουλος	Παναγιώτης	Βασίλειος	Αγαθή	1966	520036983	Π429632	ΜΟΝΙΜΟΣ	Κορίνθου 103	2610529478	Τραπ. Λογ/σμο
0004	Χαλκιάδης	Αλέξανδρος	Κωνσταντίνος	Παναγιώτα	1962	258596322	AB41526	ΣΥΜΒΑΣΙΟΥΧΟΣ	Κορίνθου 205	2610252383	Τραπ. Λογ/σμο
0005	Γιωτόπουλος	Κωνσταντίνος	Αντώνιος	Παρασκευή	1978	788996633	Π753218	ΣΥΜΒΑΣΙΟΥΧΟΣ	Νόρμαν 28	2610438373	Τραπ. Λογ/σμο
0006	Γκουμπόπουλος	Χρήστος	Κωνσταντίνος	Μαρία	1966	255263212	AE78961	ΜΟΝΙΜΟΣ	Γερμανού 25	2610859374	Τραπ. Λογ/σμο
0007	Χαλκιάς	Κωνσταντίνος	Αθανάσιος	Αλεξάνδρα	1982	686822337	AE85369	ΣΥΜΒΑΣΙΟΥΧΟΣ	Αγ. Ανδρέου 185	2610753951	Τραπ. Λογ/σμο
0008	Γκόμας	Απόστολος	Δημήτριος	Μαρία	1966	276315898	0531185	ΣΥΜΒΑΣΙΟΥΧΟΣ	Κορίνθου 160	2610963665	Τραπ. Λογ/σμο
0009	Μπακμοπούλη	Αθανασία	Σπυρίδων	Νικολέττα	1970	777556321	K425232	ΣΥΜΒΑΣΙΟΥΧΟΣ	Παπαφλέσσα 75	2610747452	Τραπ. Λογ/σμο
0010	Μπαληγιάννης	Γρηγόριος	Κωνσταντίνος	Ελένη	1976	526341897	A289632	ΜΟΝΙΜΟΣ	Κορίνθου 302	2610453963	Τραπ. Λογ/σμο
0011	Μπακούλης	Αρταπόδης	Βασίλειος	Κωνσταντίνα	1962	793185263	AB39715	ΜΟΝΙΜΟΣ	Κοραή 12	2610789615	Τραπ. Λογ/σμο
0012	Δαρανός	Βασίλειος	Παναγιώτης	Εσφία	1965	525285462	A296748	ΜΟΝΙΜΟΣ	Υψηλάντων 48	2610323263	Τραπ. Λογ/σμο
0013	Κατζίνης	Αθανάσιος	Παύλος	Σωτηρία	1960	686896541	Π796315	ΣΥΜΒΑΣΙΟΥΧΟΣ	Μαζιώνας 215	2610859612	Τραπ. Λογ/σμο
0014	Κουτσονίκος	Γρηγόριος	Δημήτριος	Ασπασία	1866	524522213	AE75963	ΜΟΝΙΜΟΣ	Γούναρη 48	2610315272	Τραπ. Λογ/σμο
0015	Σωτηροπούλοι	Βασιλική	Αθανάσιος	Αγαθή	1968	436312874	K731562	ΜΟΝΙΜΟΣ	Ρίο	2610479638	Τραπ. Λογ/σμο
0016	Κακαρελίδης	Παύλος	Αλέξανδρος	Μαρία	1962	797900037	Z123987	ΜΟΝΙΜΟΣ	Αθ. Διάκου 45	2610732117	Τραπ. Λογ/σμο
0017	Στάμος	Παναγιώτης	Γεώργιος	Αμαλία	1963	781637953	T634158	ΣΥΜΒΑΣΙΟΥΧΟΣ	Καραϊσκάκη 78	2610328954	Τραπ. Λογ/σμο
0018	Καλαπόδη	Ελένη	Γεώργιος	Μαρία	1975	784663125	A789645	ΣΥΜΒΑΣΙΟΥΧΟΣ	Αγ. Νικολάου 45	2610732545	Τραπ. Λογ/σμο
0019	Βασιλείου	Κωνσταντίνος	Παύλος	Άρτεμις	1970	456123852	I256245	ΣΥΜΒΑΣΙΟΥΧΟΣ	Αποστόλου Παύλου 1	2610214225	Τραπ. Λογ/σμο
0020	Καιμάκης	Κωνσταντίνος	Βασίλειος	Αγλαΐα	1980	727310093	A221002	ΣΥΜΒΑΣΙΟΥΧΟΣ	Ρίο	2610321041	Τραπ. Λογ/σμο
0021	Γεωργίου	Δημήτριος	Παντελής	Ευδοξία	1975	100085123	H856374	ΣΥΜΒΑΣΙΟΥΧΟΣ	Μισοιλή 42	2610236387	Τραπ. Λογ/σμο
0022	Αδαμάδης	Γεώργιος	Κωνσταντίνος	Ειρήνη	1982	793000963	I248432	ΣΥΜΒΑΣΙΟΥΧΟΣ	Καραϊσκάκη 76	2610889412	Τραπ. Λογ/σμο
0023	Μαραγκουδάς	Σπυρίδων	Γεώργιος	Γεωργία	1979	200001974	AH14746	ΜΟΝΙΜΟΣ	Κορίνθου 290	2610546587	Τραπ. Λογ/σμο
0024	Τσαγκανός	Γεώργιος	Αλέξανδρος	Παρασκευή	1965	237491477	A211582	ΣΥΜΒΑΣΙΟΥΧΟΣ	Μαζιώνας 107	2610217439	Τραπ. Λογ/σμο

Σχήμα 3.6. Εισαγωγή δεδομένων στον Πίνακα «ΚΑΘΗΓΗΤΕΣ»

Για την υλοποίηση της Βάσης Δεδομένων του ΤΕΙ Πάτρας χρησιμοποιήθηκαν οι εξής εγγραφές:

ΒΑΣΙΚΟΙ ΠΙΝΑΚΕΣ

- Για τον Πίνακα « **ΚΑΘΗΓΗΤΕΣ** »: 384 εγγραφές
- Για τον Πίνακα « **ΦΟΙΤΗΤΕΣ** »: 1006 εγγραφές
- Για τον Πίνακα « **ΜΑΘΗΜΑΤΑ** »: 888 εγγραφές
- Για τον Πίνακα « **ΥΠΑΛΛΗΛΟΙ ΓΡΑΜΜΑΤΕΙΑΣ** »: 57 εγγραφές
- Για τον Πίνακα « **ΤΜΗΜΑΤΑ ΤΕΙ** »: 16 εγγραφές
- Για τον Πίνακα « **ΓΡΑΜΜΑΤΕΙΑ ΤΜΗΜΑΤΟΣ** »: 16 εγγραφές
- Για τον Πίνακα « **ΚΛΑΔΟΣ ΣΧΟΛΗΣ** »: 3 εγγραφές

ΒΟΗΘΗΤΙΚΟΙ ΠΙΝΑΚΕΣ

- Για τον Πίνακα «Φοιτητής – Μάθημα»: 60.089 εγγραφές
- Για τον Πίνακα « Καθηγητής – Μάθημα »: 888 εγγραφές
- Για τον Πίνακα «Τμήμα ΤΕΙ – Καθηγητής»: 384 εγγραφές
- Για τον Πίνακα « Γραμματεία – Υπάλληλοι »: 57 εγγραφές
- Για τον Πίνακα «Τμήμα ΤΕΙ – Γραμματεία Τμήματος»: 17 εγγραφές
- Για τον Πίνακα «Σχολές – Τμήματα»: 16 εγγραφές

3.4 Δημιουργία Φόρμας Βάσεως Δεδομένων ΤΕΙ Πάτρας

Δημιουργώντας φόρμες εξασφαλίζουμε ένα πιο ευχάριστο και εύχρηστο περιβάλλον εργασίας από τους πίνακες. Μέσω μιας φόρμας μπορούμε να εισάγουμε δεδομένα σε περισσότερους από έναν πίνακες ταυτόχρονα. Ακόμα επειδή μια φόρμα αναφέρεται σε μια εγγραφή κάθε φορά, είμαστε βέβαιοι ότι δουλεύουμε με τα πεδία της ίδιας εγγραφής του πίνακα. Οι φόρμες μπορεί να είναι απλές, αλλά και περίτεχνες, με γραφικά και με πλήκτρα αυτοματοποίησης των εντολών.

Ο σχεδιασμός μιας φόρμας πραγματοποιείται με έναν πολύ εύχρηστο Οδηγό Φορμών, ο οποίος σε καθοδηγεί και σου επιτρέπει να προσθέσεις λεπτομέρειες σχετικά με την εμφάνιση της φόρμας σύμφωνα με το δικό σου γούστο Σχήμα 3.7.

Για το λόγο αυτό και στη δική μας Βάση Δεδομένων δημιουργήσαμε κάποιες φόρμες ώστε ο εμπλουτισμός της να πραγματοποιηθεί με μεγαλύτερη ευκολία, όπως φαίνεται στα παρακάτω Σχήματα 3.7 και 3.8 και 3.9.

Οδηγός φορμών

Ποια πεδία θέλετε στη φόρμα;
Έχετε τη δυνατότητα να επιλέξετε από περισσότερους από έναν πίνακες ή ερωτήματα.

Πίνακες/Ερωτήματα
Πίνακας: ΚΑΘΗΓΗΤΕΣ

Διαθέσιμα πεδία:

Επιλεγμένα πεδία:
Ετος_Γέννησης
ΑΦΜ
Αριθμός_ταυτότητας
Χαρακτηρισμός
Διευθυνση_Κατοικίας
Τηλέφωνο_επικοινωνίας
Τρόπος_Μισθοδοσίας
Email

Ακυρο < Προηγούμενο Επόμενο > Τέλος

Σχήμα 3.7 Οδηγός Φορμών

ΕΙΣΑΓΩΓΗ ΚΑΘΗΓΗΤΩΝ

Κωδικός Καθηγητή	<input type="text"/>
Επώνυμο	<input type="text"/>
Όνομα	<input type="text"/>
Πατρώνυμο	<input type="text"/>
Μητρώνυμο	<input type="text"/>
Έτος Γέννησης	<input type="text"/>
ΑΦΜ	<input type="text"/>
Αριθμός Ταυτότητας	<input type="text"/>
Χαρακτηρισμός	<input type="text"/>
Διευθυνση Κατοικίας	<input type="text"/>
Τηλέφωνο Επικοινωνίας	<input type="text"/>
Τρόπος Μισθοδοσίας	<input type="text"/>
Email	<input type="text"/>

γραφή: 385 από 385 | Χωρίς φίλτρο | Αναζήτηση

Σχήμα 3.8 Φόρμα Εισαγωγής Καθηγητών

ΕΙΣΑΓΩΓΗ ΦΟΙΤΗΤΩΝ

Α.Μ.	<input type="text" value="1000"/>
Επώνυμο	<input type="text" value="ΓΚΑΝΑ"/>
Όνομα	<input type="text" value="ΕΥΘΥΜΙΑ"/>
Πατρώνυμο	<input type="text" value="ΠΑΝΑΓΙΩΤΗΣ"/>
Μητρώνυμο	<input type="text" value="ΜΑΡΙΑ"/>
Τόπος Γέννησης	<input type="text" value="ΑΓ. ΝΙΚΟΛΑΟΣ ΚΡΗΤΗΣ"/>
Αριθμος Ταυτότητας	<input type="text" value="ΑΖ971769"/>
Τηλέφωνο Επικοινωνίας	<input type="text" value="6971357852"/>
Email	<input type="text" value="gana@teipat.gr"/>
Σπουδαστικό Έτος Εγγραφής	<input type="text" value="2005"/>
Τρέχον Εξάμηνο	<input type="text" value="ΠΤΥΧΙΟ Α'"/>
Κωδικός Τμήματος	<input type="text" value="6"/>

Σχήμα 3.9 Φόρμα Εισαγωγής Φοιτητών

3.5 Δημιουργία Ερωτημάτων Βάσεως Δεδομένων ΤΕΙ Πάτρας

Τα Ερωτήματα είναι «**εργαλεία**» με τα οποία από έναν ή περισσότερους συσχετισμένους πίνακες δημιουργούμε νέες δυναμικές δομές δεδομένων δηλαδή νέους δυναμικούς πίνακες. Οι πίνακες που δημιουργούνται με μια ερώτηση είναι δυναμικοί πίνακες, δηλαδή αποθηκεύεται μόνο η δομή του πίνακα και όχι το περιεχόμενό του.

Τα Ερωτήματα είναι πιο ισχυρά από τα Φίλτρα! Εντοπίζουν πληροφορίες από πολλούς συνδεδεμένους πίνακες ή και από άλλα ερωτήματα, με δυνατότητα υπολογισμού συνολικών τιμών σε αντίθεση με τα φίλτρα τα οποία επιλέγουν δεδομένα μόνο από τον ίδιο πίνακα.

Ένα ερώτημα μας **επιτρέπει** να καθορίσουμε:

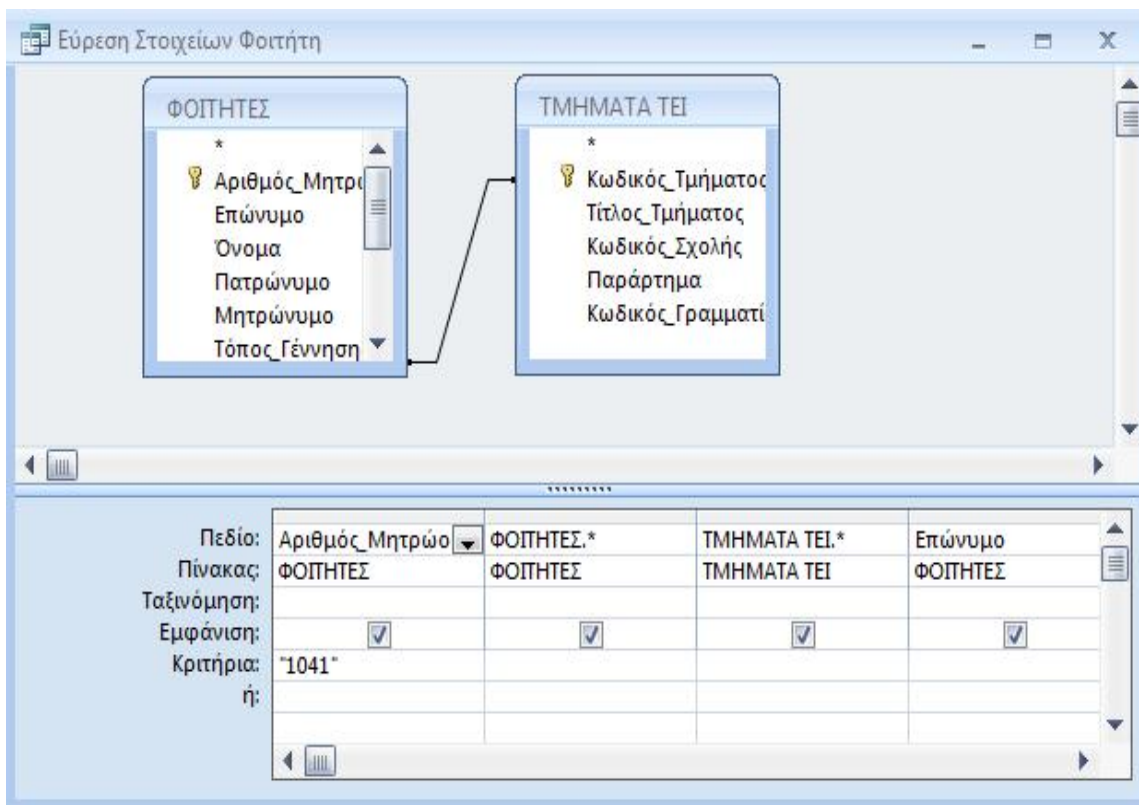
- Ø Ποια πεδία θέλουμε να βλέπουμε.
- Ø Τη σειρά με την οποία θα εμφανίζονται τα πεδία.
- Ø Κριτήρια φιλτραρίσματος για κάθε πεδίο.
- Ø Τη σειρά με την οποία θα ταξινομείται κάθε πεδίο.

Συνοπτικά τα Ερωτήματα:

- Ø Μπορούν να συνδυάζουν δεδομένα από διάφορους πίνακες.
- Ø Παρουσιάζουν τις λεγόμενες "όψεις" (views) των δεδομένων.
- Ø Τα αποτελέσματα αυτά των ερωτημάτων, έχουν τη μορφή πίνακα.
- Ø Τα αποτελέσματα των ερωτημάτων είναι πάντοτε «επίκαιρα». Υπάρχει η δυνατότητα όμως τα αποτελέσματα κάποιας ερώτησης να αποθηκευτούν σε ένα πίνακα, για να γνωρίζουμε τη κατάσταση των δεδομένων μιας συγκεκριμένης στιγμής.

Στην Βάση μας δημιουργήσαμε αρκετά ερωτήματα τα οποία χρησιμοποιούνται ευρέως από τους χρήστες μίας τέτοιας Βάσης Δεδομένων με σκοπό την άμεση αξιοποίηση και εύρεση των καταχωρημένων πληροφοριών. Κάποια από αυτά παρουσιάζονται παρακάτω. (Για την επεξεργασία όλων των Ερωτημάτων της Βάσης Δεδομένων μπορείτε να επεξεργαστείτε το cd – rom που εσωκλείεται στην Πτυχιακή Εργασία).

Ερώτημα 1 : «Εύρεση στοιχείων Φοιτητή με Αριθμό Μητρώου 1041»



Σχήμα 3.10. Δημιουργία Ερωτήματος 1.

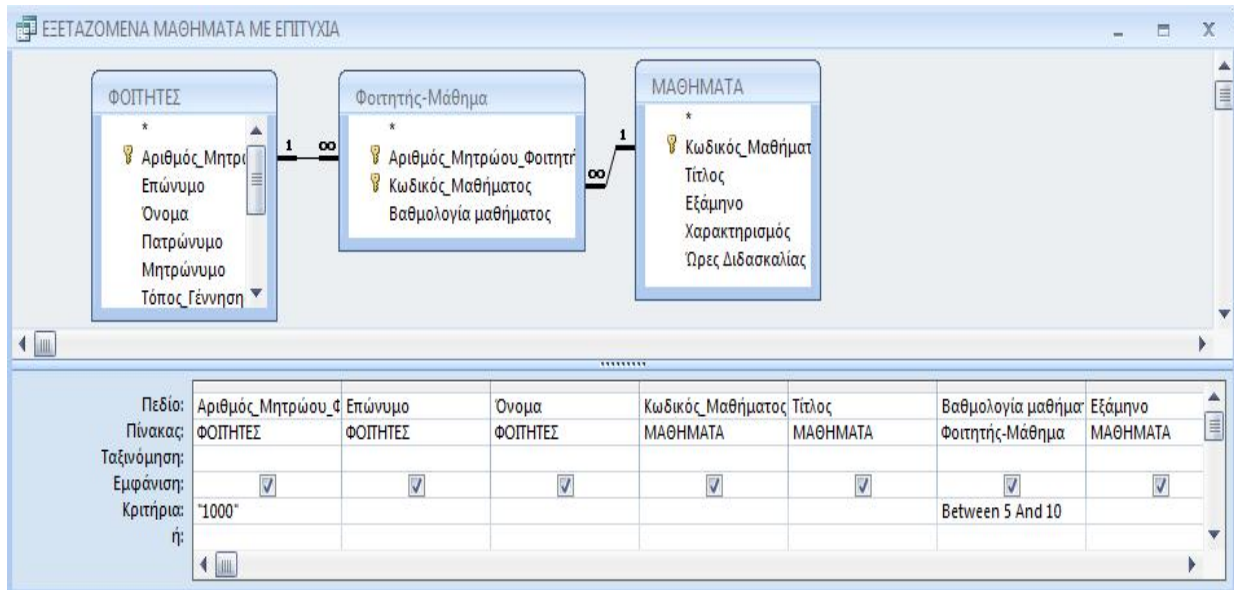
Αποτέλεσμα Ερωτήματος 1:

ΦΟΙΤΗΤΕΣ.Α	Πεδίο	ΦΟΙΤΗΤΕΣ.Επώνυ	Όνομα	Πατρώνυμο	Μητρώνυμο	Τόπος_Γέννι	Αριθμος_Ταυτότη	Τηλέφωνο_Επικοινων
1041	1041	ΘΕΟΔΩΡΟΥ	ΜΑΡΙΑ	ΧΡΗΣΤΟΣ	ΒΑΣΙΛΙΚΗ	ΑΓΡΙΝΙΟ	A426161	2641053703
*								

Εγγραφή: 1 από 1 Χωρίς φίλτρο Αναζήτηση

Σχήμα 3.11 Εύρεση Απάντησης Στο Ερώτημα 1.

Ερώτημα 2: «Ο Φοιτητής με Αριθμό Μητρώου 1000 σε ποια μαθήματα έχει εξεταστεί με επιτυχία»



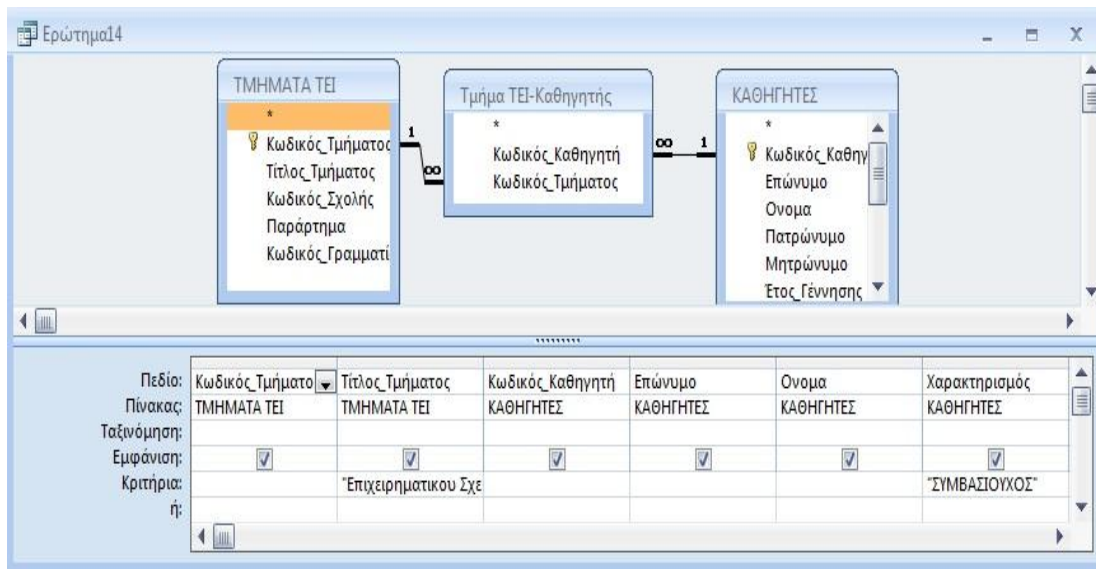
Σχήμα 3.12 Δημιουργία Ερωτήματος 2.

Αποτέλεσμα Ερωτήματος 2:

Αριθμός_Μι	Επώνυμο	Όνομα	Κωδικός_Μαθήμα	Τίτλος	Βαθμολογία_μ	Εξάμηνο
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ149	ΜΙΚΡΟΟΙΚΟΝΟΜΙΚΗ	7	A
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ150	ΛΟΓΙΣΤΙΚΗ Ι	7	A
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ151	ΑΣΤΙΚΟ ΔΙΚΑΙΟ	5	A
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ152	ΓΕΝΙΚΑ ΜΑΘΗΜΑΤΙΚΑ	6	A
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ153	ΚΒΣ-ΟΡΓΑΝΩΣΗ- ΛΟΓΙΣΤΗΡΙΟΥ	5	A
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ154	ΑΡΧΕΣ ΔΙΟΙΚΗΣΗΣ	8	A
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ155	ΛΟΓΙΣΤΙΚΗ ΙΙ	9	B
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ156	ΟΙΚΟΝΟΜΙΚΑ ΜΑΘΗΜΑΤΙΚΑ	5	B
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ157	ΣΤΑΤΙΣΤΙΚΗ	7	B
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ158	ΕΡΓΑΤΙΚΟ ΔΙΚΑΙΟ	7	B
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ159	ΜΑΚΡΟΟΙΚΟΝΟΜΙΚΗ	5	B
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ160	ΟΙΚΟΝΟΜΙΚΗ ΕΠΙΧΕΙΡΗΣΕΩΝ	6	B
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ161	ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	5	Γ
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ162	Ε.Λ.Σ.	8	Γ
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ163	ΑΥΤΟΜΑΤΙΣΜΟΣ ΓΡΑΦΕΙΟΥ	7	Γ
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ164	ΛΟΓΙΣΤΙΚΗ ΕΤΑΙΡΙΩΝ	5	Γ
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ165	ΕΜΠΟΡΙΚΟ ΔΙΚΑΙΟ	6	Γ
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ166	ΦΟΡΟΛ.ΕΜ. ΦΟΡΩΝ	5	Δ
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ167	ΣΤΑΤΙΣΤΙΚΗ ΙΙ	8	Δ
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ168	ΝΟΜ.ΘΕΩΡ. & ΠΟΛΙΤΙΚΗ	9	Δ
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ169	ΑΡΧΕΣ ΜΑΡΚΕΤΙΝΚ	5	Δ
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ170	Λ.Π.Σ.	7	Δ
1000	ΓΚΑΝΑ	ΕΥΘΥΜΙΑ	Θ171	ΟΙΚ/ΚΕΣ ΜΕΛΕΤΕΣ	7	Δ

Σχήμα 3.13 Εύρεση Απάντησης Στο Ερώτημα 2.

Ερώτημα 3: «Συμβασιούχοι ΚΑΘΗΓΗΤΕΣ του Τμήματος Επιχειρηματικού Σχεδιασμού Και Πληροφοριακών Συστημάτων»



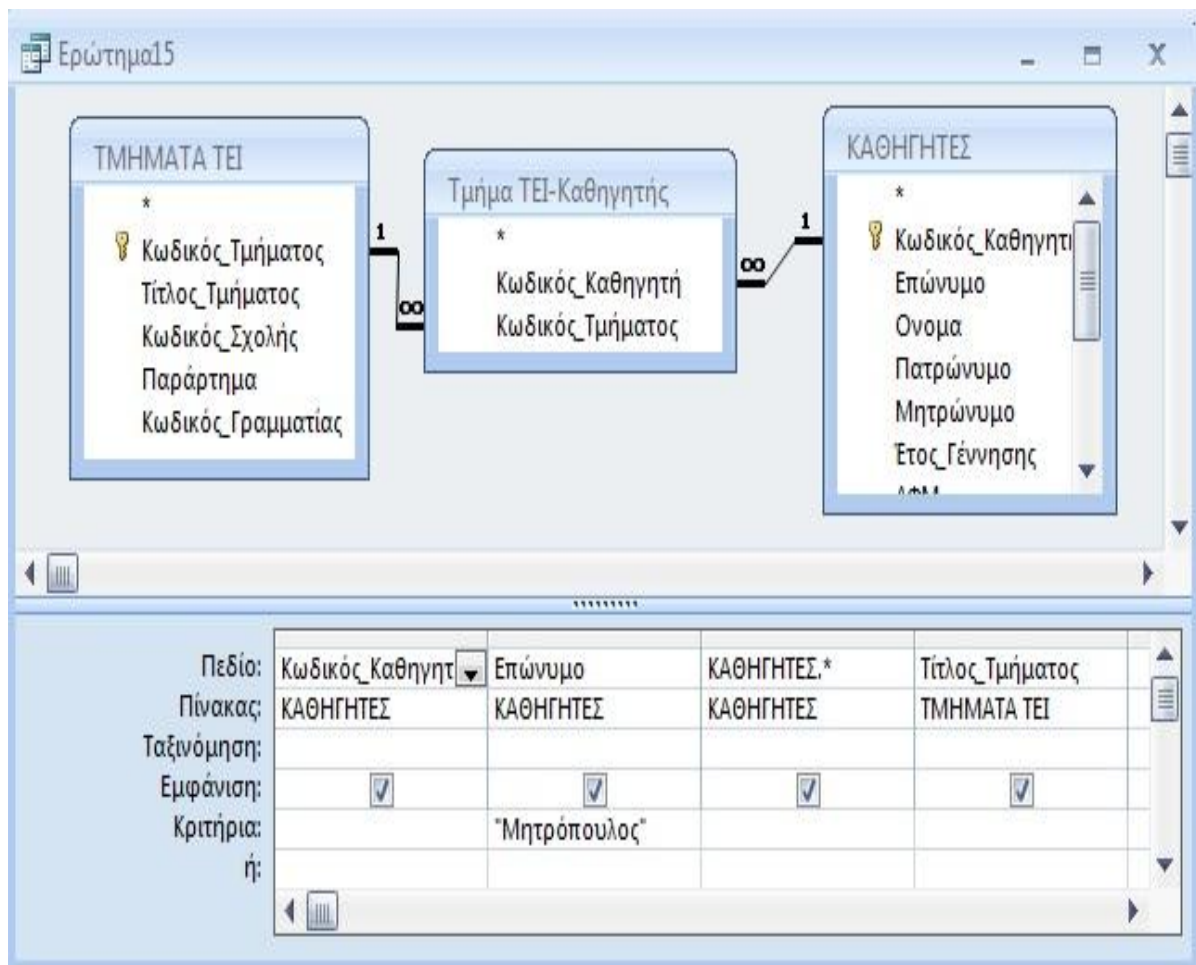
Σχήμα 3.14 Δημιουργία Ερωτήματος 3.

Αποτέλεσμα Ερωτήματος 3:

Κωδικός_Τμήμα	Τίτλος_Τμήματος	Κωδικός_Κα	Επώνυμο	Ονομα	Χαρακτηρισμός
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0004	Χαλκίοπουλος	Αλέξανδρος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0005	Γιωτόπουλος	Κωνσταντίνος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0007	Χαλκιάς	Κωνσταντίνος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0008	Γκάμας	Απόστολος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0009	Μπουμπούλη	Αθανασία	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0013	Χατζίνας	Αθανάσιος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0017	Στάμος	Παντελής	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0018	Καλαπόδη	Ελένη	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0019	Βασιλείου	Κωνσταντίνος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0020	Καμάκης	Κωνσταντίνος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0021	Γεωργίου	Δημήτριος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0022	Αδαμόπουλος	Γεώργιος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0024	Τσαγκανός	Γεώργιος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0026	Μαστρογιάννη	Αλέξανδρος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0027	Μάνδαλος	Ιάκωβος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0028	Μιχοπούλου	Μαρία	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0030	Αναστασίου	Σπυρίδων	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0031	Παπαιωάννου	Χρήστος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0033	Ρεπούσης	Σπυρίδων	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0034	Θεοδωράτος	Νικόλαος	ΣΥΜΒΑΣΙΟΥΧΟΣ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	0384	Σπυροπούλου	Ελένη	ΣΥΜΒΑΣΙΟΥΧΟΣ

Σχήμα 3.15 Εύρεση Απάντησης Στο Ερώτημα 3.

Ερώτημα 4: «Εύρεση στοιχείων ΚΑΘΗΓΗΤΗ με όνομα ΜΗΤΡΟΠΟΥΛΟΣ»



Σχήμα 3.16 Δημιουργία Ερωτήματος 4.

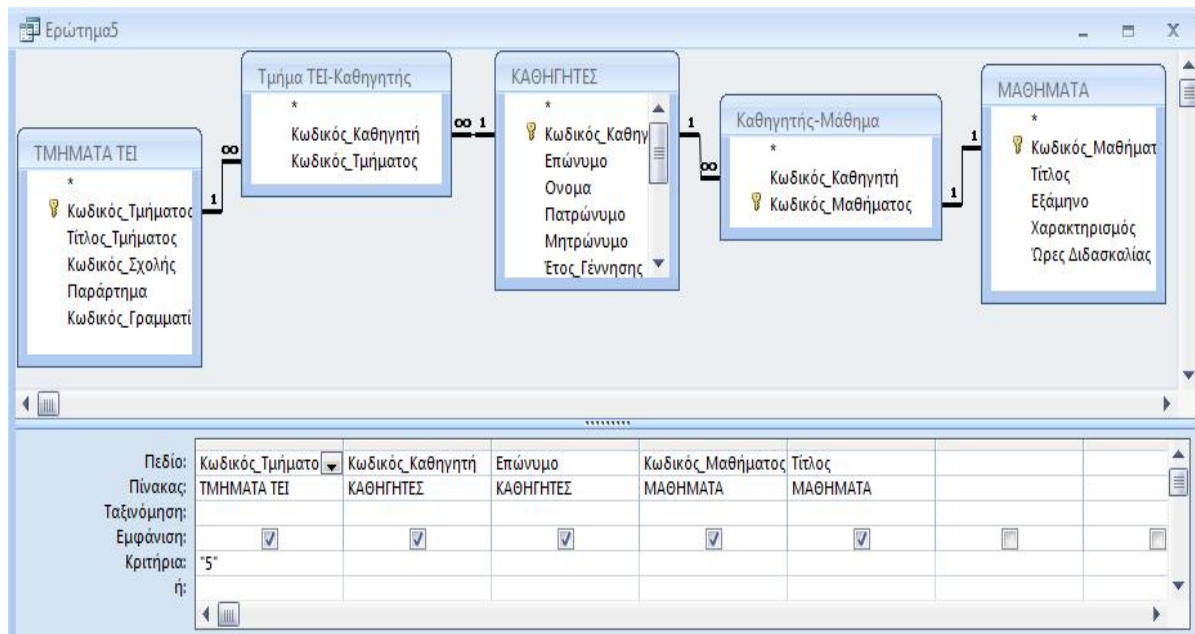
Αποτέλεσμα Ερωτήματος 4:

The screenshot shows the results of the query 'Ερώτημα15' in a table view. The table has the following columns and data:

ΚΑΘΗΓΗΤΕΣ	ΚΑΘΗΓΗΤΕΣ	Πεδίο0	Πεδίο1	Όνομα	Πατρώνυμο	Μητρώνυμο	Έτος_Γέννης	ΑΦΜ	Αριθμός_ταυτότη
0003	Μητρόπουλος	0003	Μητρόπουλος	Παναγιώτης	Βασίλειος	Αγαθη	1966	520036983	Π429632
*									

Σχήμα 3.17 Εύρεση Απάντησης Στο Ερώτημα 4.

Ερώτημα 5: « Ποια μαθήματα διδάσκουν οι ΚΑΘΗΓΗΤΕΣ του Τμήματος Διοίκησης Επιχειρήσεων»



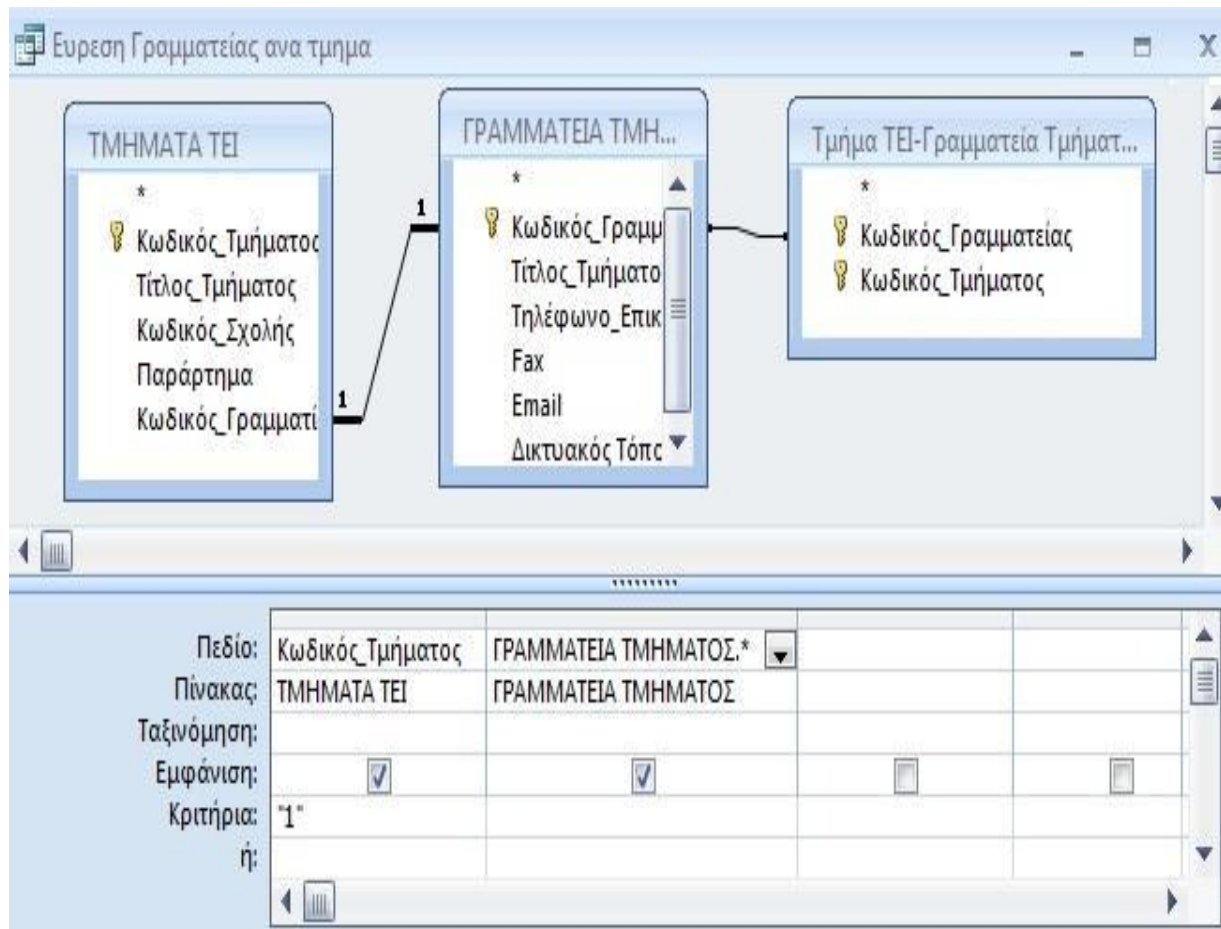
Σχήμα 3.18 Δημιουργία Ερωτήματος 5.

Αποτέλεσμα Ερωτήματος 5:

Κωδικός_Τμήματος	Κωδικός_Κα	Επώνυμο	Κωδικός_Μαθήματος	Τίτλος
5	0074	Γαλανού	0109	Μικροοικονομία
5	0075	Παναγόπουλος	0110	Αστικό Δίκαιο
5	0075	Παναγόπουλος	0115	Εργατικό Δίκαιο
5	0075	Παναγόπουλος	0127	Διεθνείς Οικονομικές Σχέσεις
5	0075	Παναγόπουλος	0133	Εμπορικό Δίκαιο
5	0076	Φαζάκης	0111	Διοίκηση Επιχειρήσεων
5	0076	Φαζάκης	0116	Μακροοικονομία
5	0076	Φαζάκης	E068	Μακροοικονομία
5	0076	Φαζάκης	E071	Διοίκηση Ολικής Ποιότητας
5	0076	Φαζάκης	0117	Διοίκηση και Οργανωτική Συμπεριφορά
5	0076	Φαζάκης	0128	Επιχ. Επικοινωνίες
5	0076	Φαζάκης	0134	Μάρκετινγκ Υπηρεσιών
5	0077	Κασμάτης	0112	Μαθηματικά Διοίκησης
5	0077	Κασμάτης	E065	Μαθηματικά Διοίκησης
5	0077	Κασμάτης	0118	Στατιστική Επιχειρήσεων
5	0077	Κασμάτης	0119	Οικονομικά Μαθηματικά
5	0077	Κασμάτης	E077	Δημόσια Οικονομική
5	0078	Παπανικολάου	0113	Γενική Λογιστική
5	0078	Παπανικολάου	E066	Γενική Λογιστική
5	0078	Παπανικολάου	E078	Δημόσια Οικονομική

Σχήμα 3.19 Εύρεση Απάντησης Στο Ερώτημα 5.

Ερώτημα 6: « Εύρεση ΓΡΑΜΜΑΤΕΙΑΣ στο Ηλεκτρολογίας»



Σχήμα 3.20 Δημιουργία Ερωτήματος 6.

Αποτέλεσμα Ερωτήματος 6:

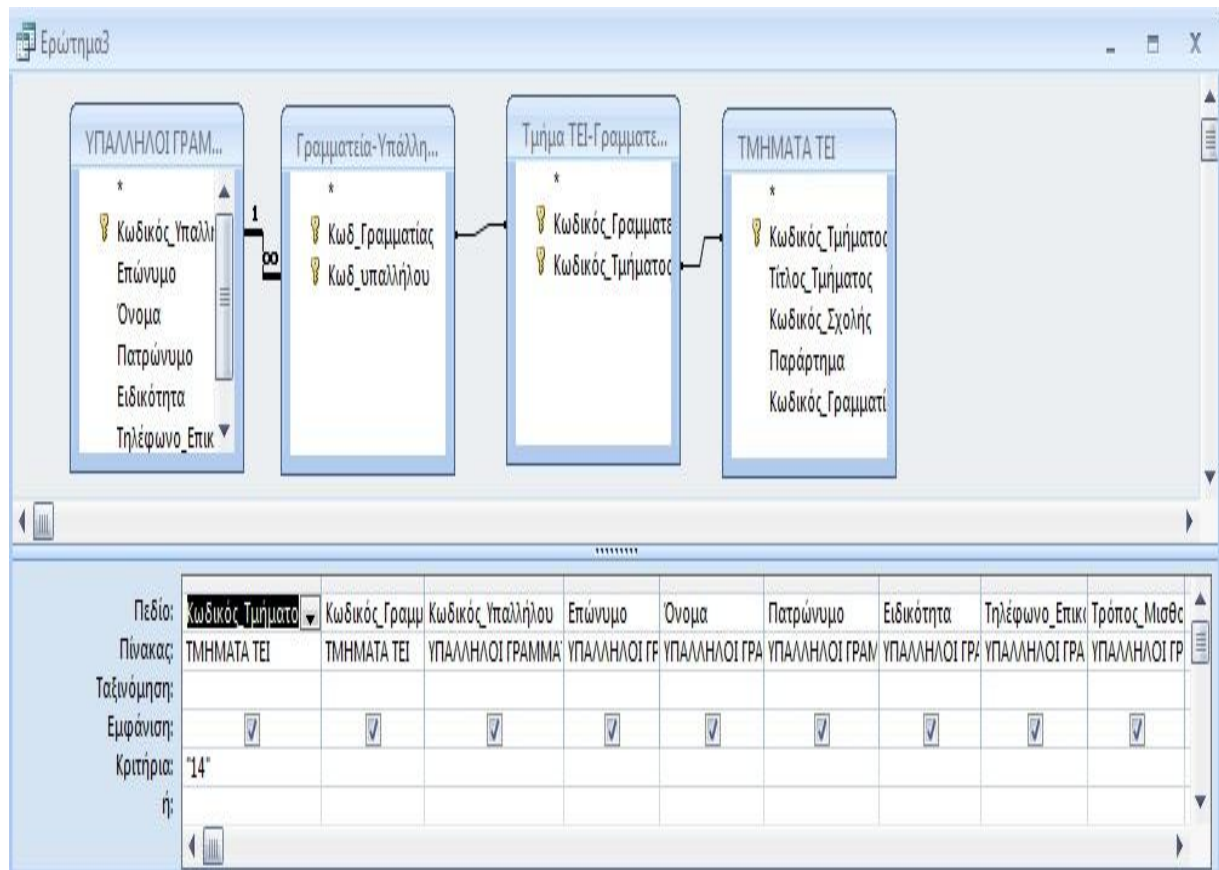
The screenshot shows the result of the query in a table:

Κωδικός_Τμ	Κωδικός_Γρα	Τίτλος_Τμήματ	Τηλέφωνο_Ε	Fax	Email	Δικτυακός_Τόπος
1	8	ΗΛΕΚΤΡΟΛΟΓΙΑ	2610369236	2610369193	ilektrologia@teipat.gr	http://www.teipat.gr/pages/ele/
*						

At the bottom of the screenshot, there is a navigation bar with the text: Εγγραφή: 2 από 2, Χωρίς φίλτρο, Αναζήτηση.

Σχήμα 3.21 Εύρεση Απάντησης Στο Ερώτημα 6.

Ερώτημα 7: « ΥΠΑΛΛΗΛΟΙ ΓΡΑΜΜΑΤΕΙΑΣ στο Τμήμα Λογοθεραπείας»



Σχήμα 3.22 Δημιουργία Ερωτήματος 7.

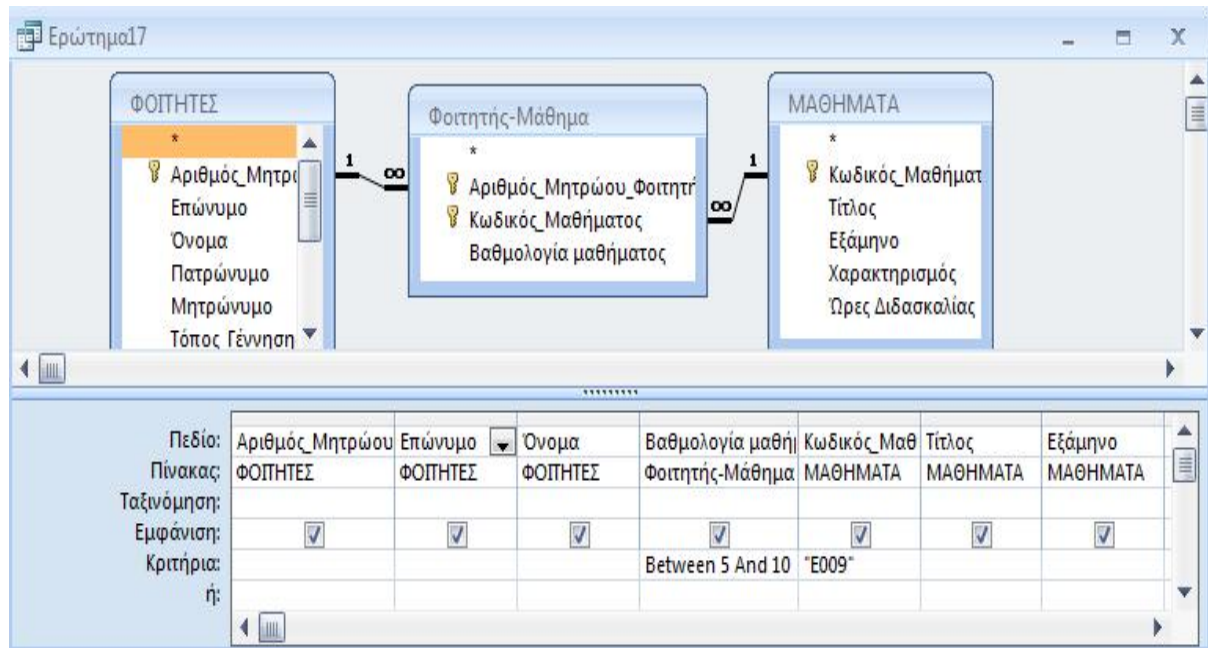
Αποτέλεσμα Ερωτήματος 7:

Κωδικός Τμήμ	Κωδικός Γραμμ	Κωδικός Υπαλλ	Επώνυμο	Όνομα	Πατρώνυμο	Ειδικότητα	Τηλέφωνο_Επικ	Τρόπος_Μισθοδ
14	14	48	ΚΑΡΔΟΥΣΗ	ΚΥΡΙΑΚΗ	ΣΑΒΒΑΣ	ΠΡΟΙΣΤΑΜΕΝΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6984253612	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣ
14	14	49	ΞΑΝΘΟΠΟΥΛΟΣ	ΓΕΩΡΓΙΟΣ	ΠΑΥΛΟΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	2610821560	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣ
14	14	50	ΔΗΜΗΤΡΙΑΔΗΣ	ΟΡΕΣΤΗΣ	ΚΟΣΜΑΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	2610899260	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣ
14	14	51	ΣΑΒΒΑΣ	ΞΕΝΟΦΩΝ	ΓΕΩΡΓΙΟΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	2610744581	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣ

Εγγραφή: 1 από 4 Χωρίς φίλτρο Αναζήτηση

Σχήμα 3.23 Εύρεση Απάντησης Στο Ερώτημα 7.

Ερώτημα 8: «ΦΟΙΤΗΤΕΣ με επιτυχή εξέταση στο ΜΑΘΗΜΑ Εργαστηρίου “Συστήματα Διαχείρισης Βάσεως Δεδομένων και Κειμένων Στο Επιχειρηματικό Περιβάλλον” και οι βαθμολογίες αυτών»



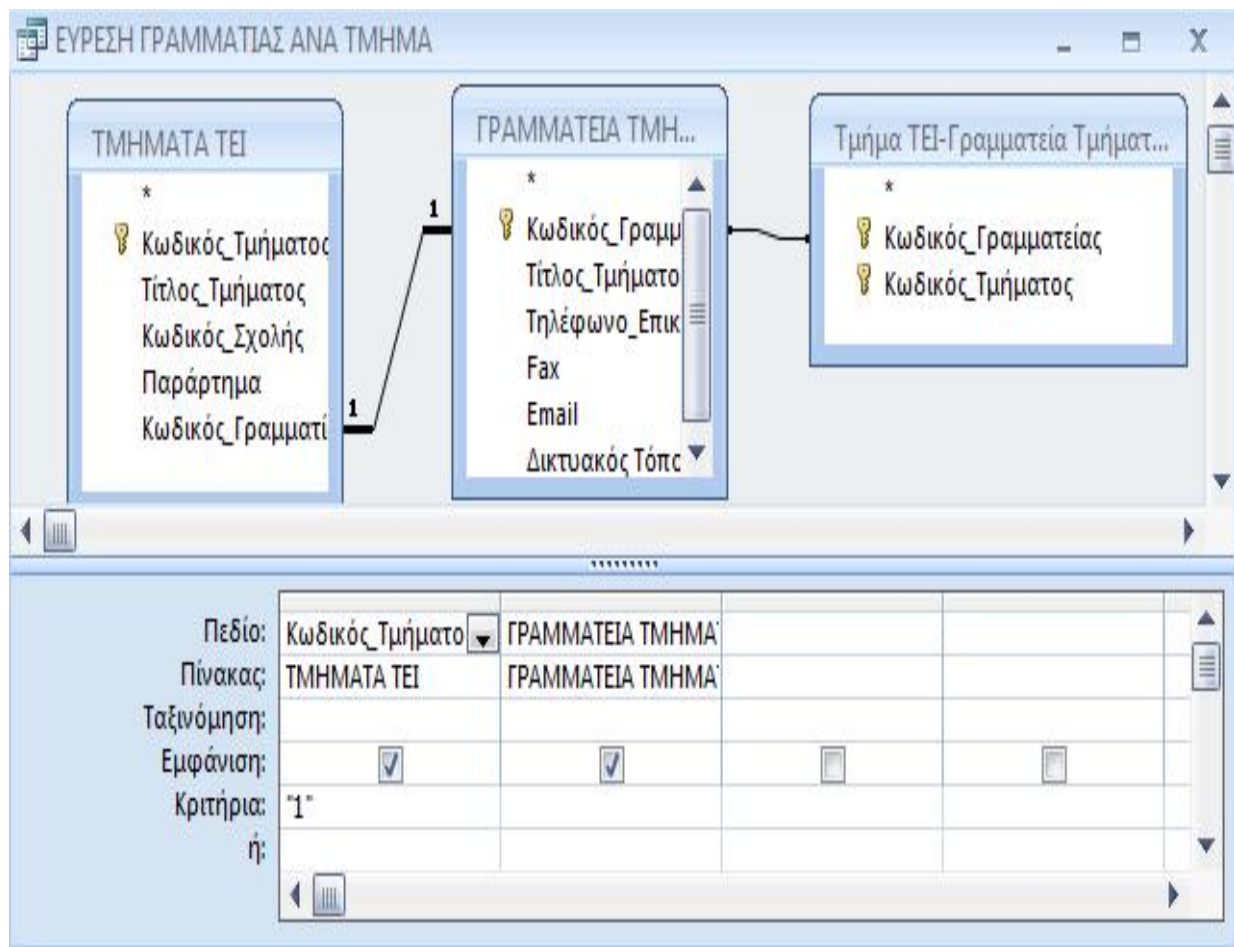
Σχήμα 3.24 Δημιουργία Ερωτήματος 8.

Αποτέλεσμα Ερωτήματος 8:

Αριθμός_Μι	Επώνυμο	Όνομα	Βαθμολογία_μ	Κωδικός_Μαθ	Τίτλος	Εξάμηνο
954	ΚΥΛΑΦΗΣ	ΚΩΝΣΤΑΝΤΙΝΟΣ	5	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
1041	ΘΕΟΔΩΡΟΥ	ΜΑΡΙΑ	8	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
1163	ΑΓΓΙΔΗΣ	ΑΘΑΝΑΣΙΟΣ	7	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
1283	ΑΓΙΑΣΩΤΕΛΗ	ΣΟΦΙΑ	6	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
1421	ΑΠΟΣΤΟΛΙΔΟΥ	ΧΡΥΣΑΝΘΗ	5	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
226	ΑΠΟΣΤΟΛΟΠΟΥΛΟΥ	ΜΕΛΠΟΜΕΝΗ	6	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
312	ΑΡΓΥΡΙΟΥ	ΝΙΚΗΦΟΡΟΣ	5	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
380	ΒΛΑΧΟΚΥΡΙΑΚΟΥ	ΗΛΙΑΝΑ	8	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
336	ΓΚΟΥΜΑΣ	ΗΛΙΑΣ	8	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
1620	ΔΗΜΗΤΡΙΟΥ	ΕΛΕΝΗ	7	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
2732	ΚΑΡΝΑΡΟΥ	ΔΗΜΗΤΡΑ	5	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
174	ΕΥΑΓΓΕΛΟΥ	ΜΑΡΙΑΝΑ	9	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
1355	ΚΑΡΑΧΡΗΣΤΟΥ	ΕΛΕΝΗ	8	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
3881	ΚΟΤΟΠΟΥΛΟΥ	ΘΕΟΔΩΡΟΣ	8	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
1643	ΛΟΝΤΟΣ	ΠΡΟΚΟΠΙΟΣ	6	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
1860	ΣΙΔΕΡΗΣ	ΑΝΔΡΕΑΣ	8	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
2719	ΤΖΕΛΕΠΗΣ	ΑΓΓΕΛΟΣ	8	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
1324	ΤΖΕΛΙΟΥ	ΣΤΑΜΑΤΙΑ	9	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
2508	ΑΛΜΠΑΝΙΔΗ	ΠΑΝΑΓΙΩΤΑ	5	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
2802	ΑΝΑΓΝΩΣΤΟΠΟΥΛΟ	ΔΕΣΠΟΙΝΑ	6	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
135	ΓΡΑΜΜΑΤΙΚΟΣ	ΔΗΜΗΤΡΙΟΣ	7	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	
2658	ΕΥΑΓΓΕΛΟΥ	ΕΥΣΤΑΘΙΑ	5	E009	Συστήματα Διαχ. ΒΔ και κειμε Δ	

Σχήμα 3.25 Εύρεση Απάντησης Στο Ερώτημα 8.

Ερώτημα 9: « Εύρεση ΓΡΑΜΜΑΤΕΙΑΣ ανά Τμήμα»



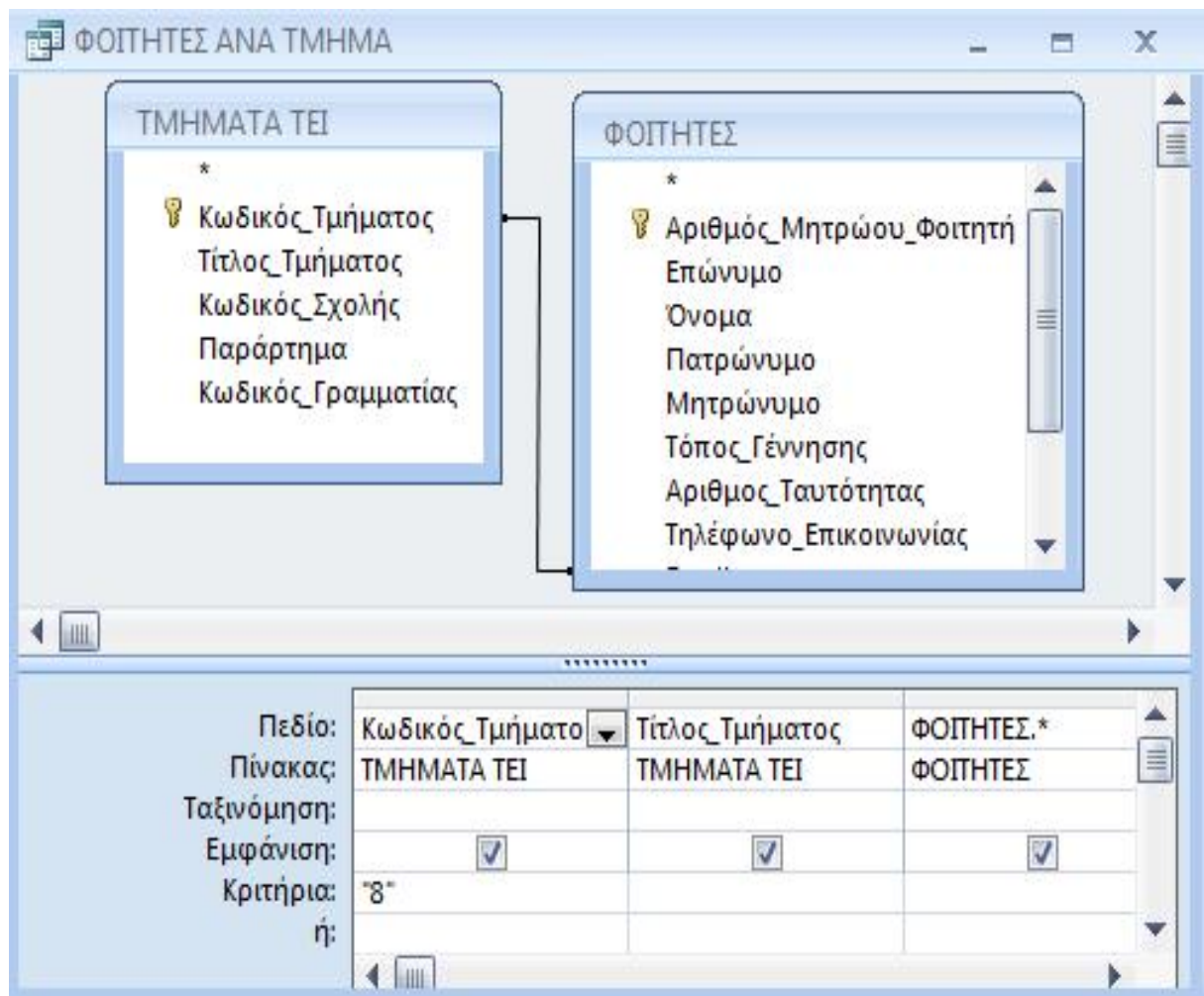
Σχήμα 3.26 Δημιουργία Ερωτήματος 9.

Αποτέλεσμα Ερωτήματος 9:

Κωδικός_Τμή...	Κωδικός_Γρα...	Τίτλος_Τμήμα...	Τηλέφωνο_Ε...	Fax	Email
1	8	ΗΛΕΚΤΡΟΛΟΓΙΑ	2610369236	2610369193	ilektrologia@teipat.gr

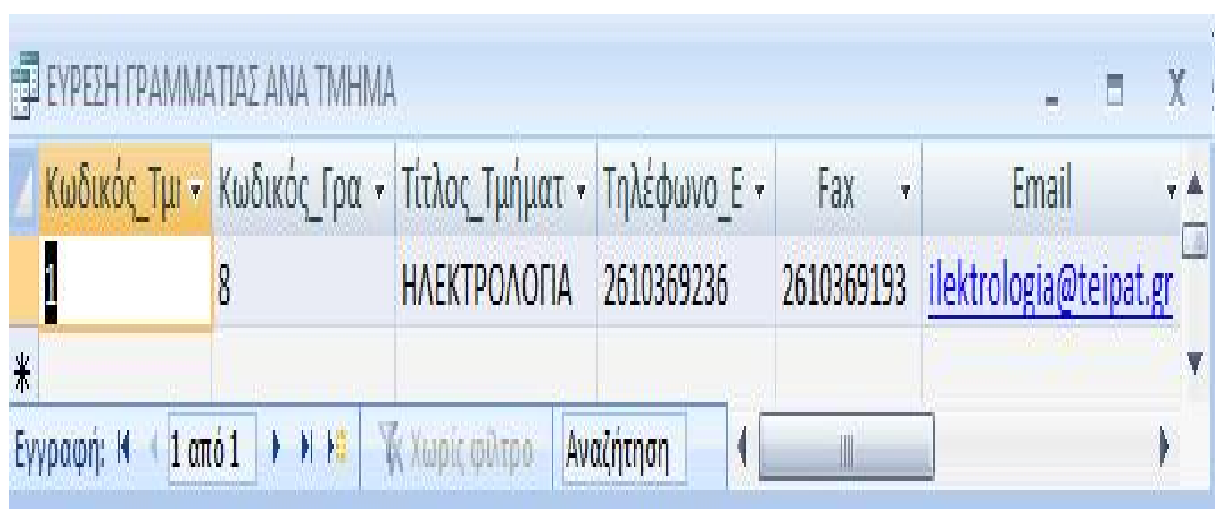
Σχήμα 3.27 Εύρεση Απάντησης Στο Ερώτημα 9.

Ερώτημα 10: « Εύρεση ΦΟΙΤΗΤΩΝ ανά Τμήμα»



Σχήμα 3.28 Δημιουργία Ερωτήματος 10.

Αποτέλεσμα Ερωτήματος 10:



Σχήμα 3.29 Εύρεση Απάντησης Στο Ερώτημα 10.

3.6 Δημιουργία Εκθέσεων Βάσεως Δεδομένων ΤΕΙ Πάτρας

Οι εκθέσεις αποτελούν έναν αποτελεσματικό τρόπο για την παρουσίαση και την εκτύπωση επιλεγμένων πληροφοριών μιας Βάσης Δεδομένων. Μας επιτρέπουν να ομαδοποιούμε τα δεδομένα σε διάφορα επίπεδα, να καθορίζουμε τη διάταξή τους και να τα μορφοποιούμε κατάλληλα. Δημιουργούμε μια έκθεση αποκλειστικά με σκοπό την εκτύπωση.

Όπως οι Φόρμες έτσι και οι Εκθέσεις δημιουργούνται με μεγάλη ευκολία μέσω ενός οδηγού. Οι φόρμες που δημιουργήσαμε παρουσιάζονται στα παρακάτω Σχήματα.

Οδηγός εκθέσεων

Ποια πεδία θέλετε στην έκθεση;

Εχετε τη δυνατότητα να επιλέξετε από περισσότερους από έναν πίνακες ή ερωτήματα.

Πίνακες/Ερωτήματα

Ερώτημα: Ερώτημα17

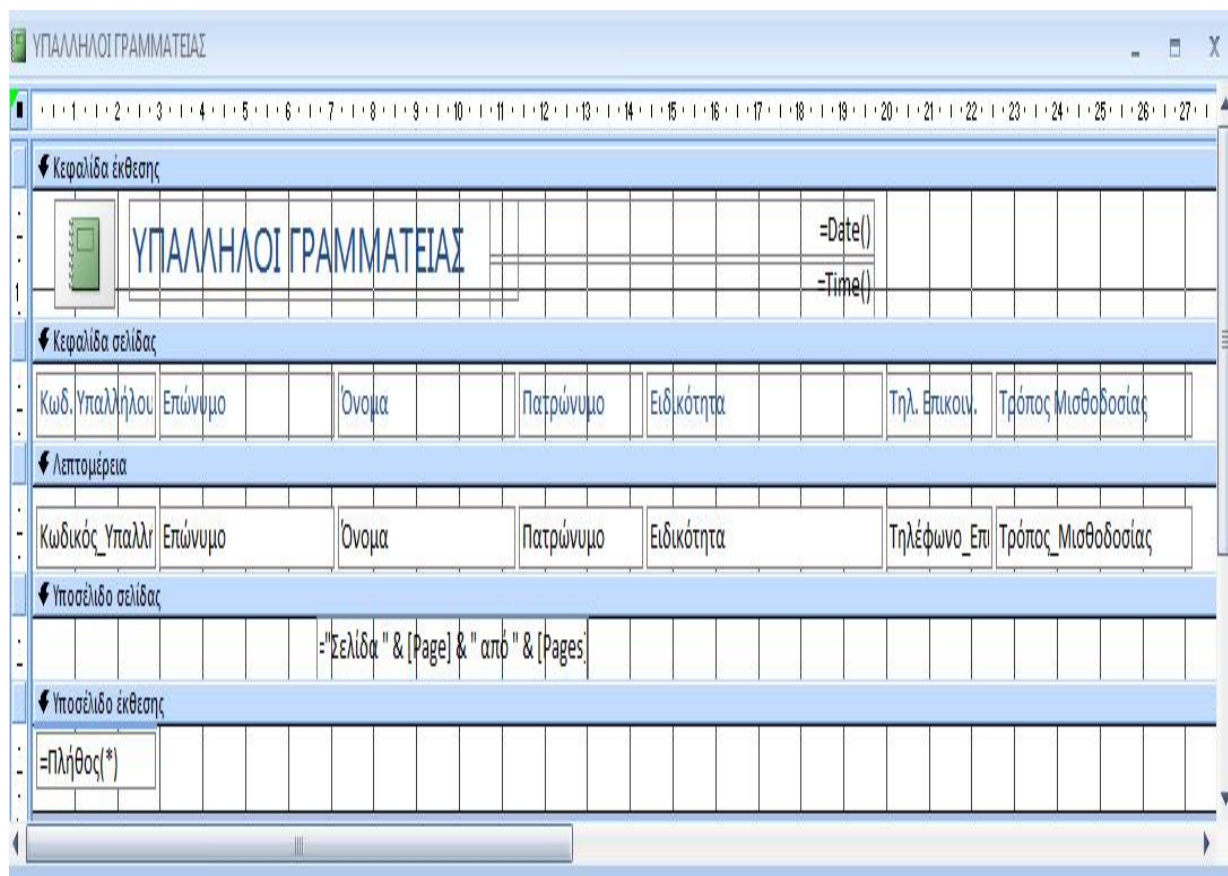
Διαθέσιμα πεδία:

Επιλεγμένα πεδία:

Αριθμός_Μητρώου_Φοιτητή
Επώνυμο
Όνομα
Βαθμολογία μαθήματος
Κωδικός_Μαθήματος
Τίτλος
Εξάμηνο

Άκυρο < Προηγούμενο Επόμενο > Τέλος

Σχήμα 3.30 Οδηγός Δημιουργίας Μιας Έκθεσης.



Σχήμα 3.31 Έκθεση Σε Προβολή Σχεδίασης Υπαλλήλων Γραμματείας.

The screenshot shows the data for the 'ΥΠΑΛΛΗΛΟΙ ΓΡΑΜΜΑΤΕΙΑΣ' spreadsheet. The data is as follows:

Κωδ. Υπαλλήλου	Επώνυμο	Όνομα	Πατρώνυμο	Ειδικότητα	Τηλ. Επικου.	Τρόπος Μισθοδοσίας
1	ΑΝΑΣΤΑΣΙΑΔΗΣ	ΠΑΝΑΠΩΤΗΣ	ΑΓΓΕΛΟΣ	ΠΡΟΙΣΤΑΜΕΝΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6958626398	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ
2	ΒΛΑΜΗ	ΑΡΙΣΤΕΑ	ΙΑΣΩΝΑΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6974238410	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ
3	ΙΑΚΩΒΙΔΟΥ	ΕΙΡΗΝΗ	ΧΡΗΣΤΟΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6974858592	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ
4	ΠΑΡΑΣΚΕΥΑΣ	ΑΝΤΩΝΙΟΣ	ΑΝΑΣΤΑΣΙΟΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6932801452	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ
5	ΖΑΧΑΡΟΠΟΥΛΟΥ	ΟΛΓΑ	ΑΘΑΝΑΣΙΟΣ	ΠΡΟΙΣΤΑΜΕΝΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6975621452	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ
6	ΑΣΗΜΑΚΗΣ	ΠΑΝΑΠΩΤΗΣ	ΣΩΤΗΡΙΟΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6954268658	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ
7	ΡΑΖΟΥΛΗ	ΕΛΕΝΗ	ΟΡΕΣΤΗΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6974889623	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ
8	ΔΗΜΑΚΗ	ΕΥΔΟΞΙΑ	ΠΑΝΑΓΙΩΤΗΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6947847596	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ
9	ΓΑΪΤΑΝΟΥ	ΑΡΓΥΡΟ	ΗΛΙΑΣ	ΠΡΟΙΣΤΑΜΕΝΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6935625230	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ
10	ΦΑΚΗ	ΓΕΩΡΓΙΑ	ΝΙΚΟΛΑΟΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6977421962	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ
11	ΠΑΧΗΣ	ΚΩΝ/ΝΟΣ	ΓΕΩΡΓΙΟΣ	ΥΠΑΛΛΗΛΟΣ ΓΡΑΜΜΑΤΕΙΑΣ	6951525395	ΤΡΑΠΕΖΙΚΟΣ ΛΟΓΑΡΙΣΜΟΣ

Σχήμα 3.32 Έκθεση Υπαλλήλων Γραμματείας.

ΦΟΙΤΗΤΕΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ

Κεφαλίδα έκθεσης

ΦΟΙΤΗΤΕΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ

Κεφαλίδα σελίδας

Κοδ. Τμήματος	Τίτλος Τμήματος	A.M.	Επώνυμο	Όνομα	Πατρώνυμο	Μητρώνυμο	A.T.	Τηλ. Επικον.	Email
---------------	-----------------	------	---------	-------	-----------	-----------	------	--------------	-------

Λεπτομέρεια

[ΤΜΗΜΑΤΑ ΤΕΠ]Κωδικός_Τμήματος	Τίτλος_Τμήματος	Αριθμ_ός_Μητρώου_Φοι	Επώνυμο	Όνομα	Πατρώνυμο	Μητρώνυμο	Αριθμός_Τα_υτότητας	Τηλέφωνο_Επ_ικοινωνίας	Email
-------------------------------	-----------------	----------------------	---------	-------	-----------	-----------	---------------------	------------------------	-------

Υποσέλιδο σελίδας

=Now()

Υποσέλιδο έκθεσης

Σχήμα 3.33 Έκθεση Σε Προβολή Σχεδίασης Φοιτητών Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων.

ΦΟΙΤΗΤΕΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ

ΦΟΙΤΗΤΕΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ

Κοδ. Τμήματος	Τίτλος Τμήματος	A.M.	Επώνυμο	Όνομα	Πατρώνυμο	Μητρώνυμο	A.T.	Τηλ. Επικον.	Email	Σπουδαστικό έτος Εγγραφής	Τρέχον Εξάμηνο
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	2082	ΤΖΟΚΑ	ΛΑΜΠΡΙΝΗ	ΕΥΣΤΑΘΙΟΣ	ΕΛΕΥΘΕΡΙΑ	X379764	2271065100	tzoka@teipat.gr	2009	Γ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	1856	ΤΖΟΥΜΑΚΑ	ΣΟΦΙΑ	ΠΕΤΡΟΣ	ΓΕΩΡΓΙΑ	P799779	6977599635	tzoumaka@teipat.gr	2007	Ε'
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	1950	ΤΟΥΜΑΝΙΔΟΥ	ΣΟΦΙΑ	ΙΑΚΩΒΟΣ	ΝΑΤΑΛΙΑ	AB392484	2610654123	toumanidou@teipat.gr	2008	Δ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	1999	ΤΡΙΑΝΤΑΦΥΛΑ ΛΟΠΟΥΛΟΥ	ΜΑΡΙΑ	ΓΙΑΝΝΗΣ	ΕΥΑΓΓΕΛΙΑ	M904124	2610987610	triantaf@teipat.gr	2009	Β'
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	1585	ΣΙΔΕΡΗ	ΘΕΑΝΩ	ΕΛΙΣΣΑΙΟΣ	ΑΓΓΕΛΙΚΗ	AB724847	6955212544	sideri@teipat.gr	2009	Γ
8	Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων	2000	ΤΡΙΑΝΤΑΦΥΛΑ ΛΟΥ	ΠΑΝΑΓΙΩΤΗΣ	ΧΡΗΣΤΟΣ	ΠΑΡΑΣΚΕΥΗ	X958836	6955455210	triantafillou@teipat.gr	2010	Α'

Σχήμα 3.34 Έκθεση Φοιτητών Τμήματος Επιχειρηματικού Σχεδιασμού Και Πληροφοριακών Συστημάτων.

ΚΑΘΗΓΗΤΕΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΚΑΙ ΔΙΔΑΧΘΕΝΤΑ ΜΑΘΗΜΑΤΑ

Κεφαλίδα έκθεσης

ΚΑΘΗΓΗΤΕΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΚΑΙ ΔΙΔΑΧΘΕΝΤΑ ΜΑΘΗΜΑΤΑ

Κεφαλίδα σελίδας

Κωδικός Τμήματος	Κωδικός Καθηγητή	Επώνυμο	Κωδικός Μαθήματος	Τίτλος
------------------	------------------	---------	-------------------	--------

Κεφαλίδα Κωδικός Τμήματος

Λεπτομέρεια

Κωδικός Τμήματος	Κωδικός Καθηγητή	Επώνυμο	Κωδικός Μαθήματος	Τίτλος
------------------	------------------	---------	-------------------	--------

Υποσέλιδο σελίδας

=Now()

Υποσέλιδο έκθεσης

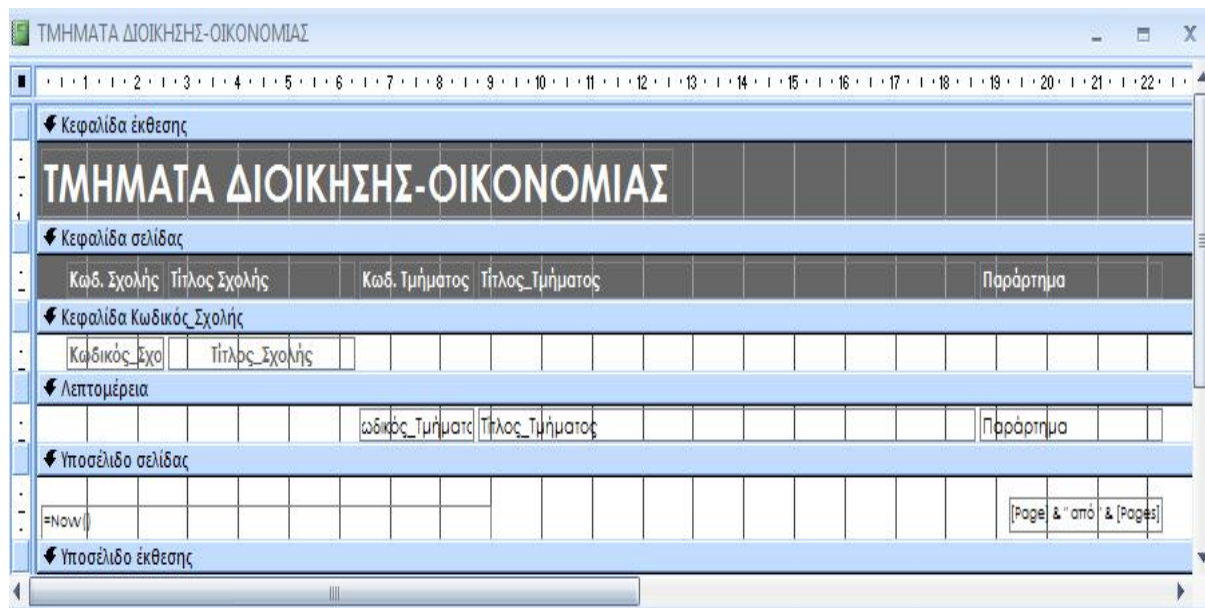
Σχήμα 3.35 Έκθεση Σε Προβολή Σχεδίασης Καθηγητών και Διδαχθέντων Μαθημάτων Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων

ΚΑΘΗΓΗΤΕΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΚΑΙ ΔΙΔΑΧΘΕΝΤΑ ΜΑΘΗΜΑΤΑ

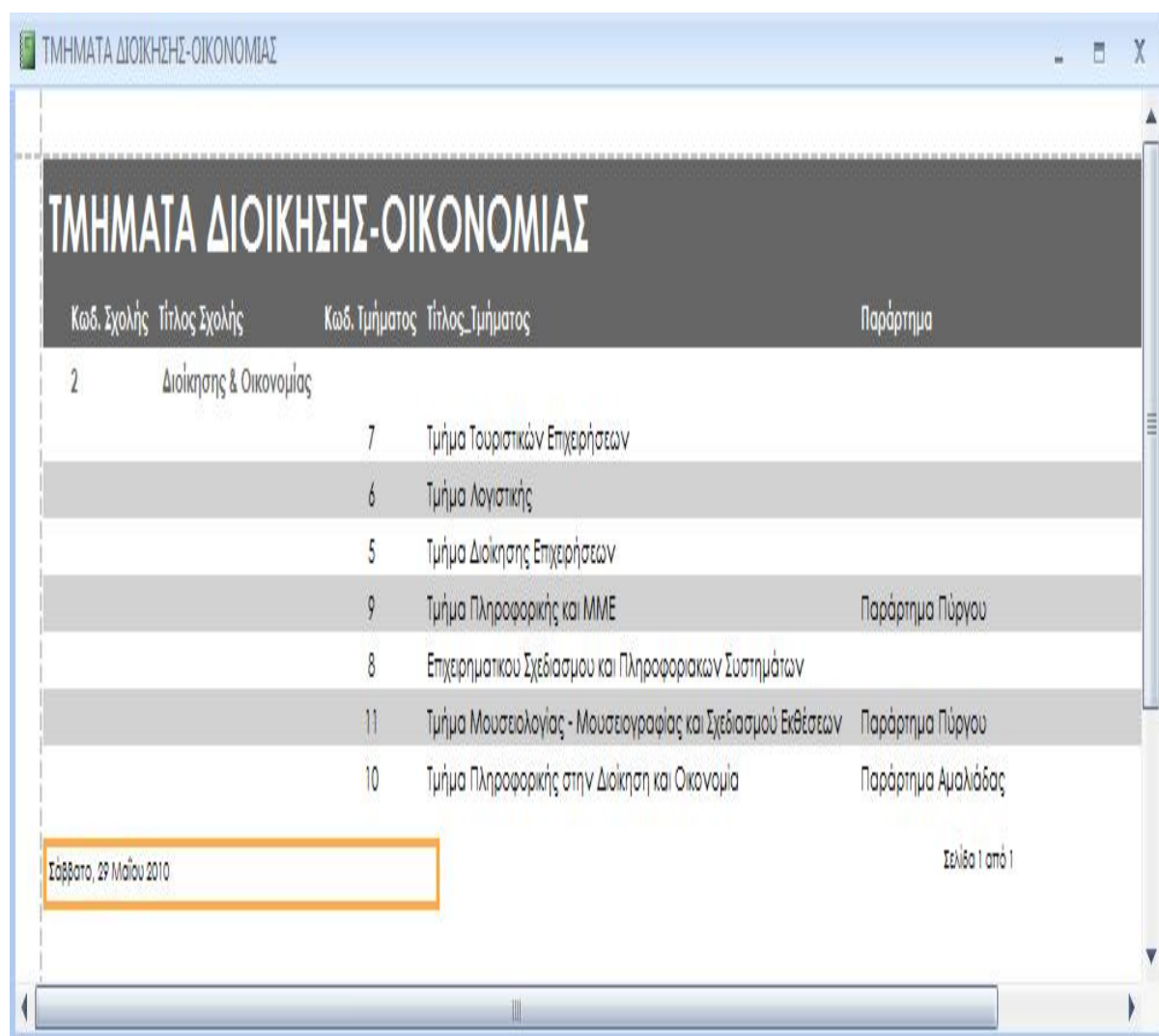
ΚΑΘΗΓΗΤΕΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΚΑΙ ΔΙΔΑΧΘΕΝΤΑ ΜΑΘΗΜΑΤΑ

Κωδικός Τμήματος	Κωδικός Καθηγητή	Επώνυμο	Κωδικός Μαθήματος	Τίτλος
5	0077	Κασιμάτης	Θ118	Στατιστική Επιχειρήσεων
	0074	Γαλανού	Θ109	Μικροοικονομία
	0079	Αντωνοπούλου	Ε072	Γραμμικός Προγραμματισμός
	0079	Αντωνοπούλου	Θ121	Γραμμικός Προγραμματισμός
	0079	Αντωνοπούλου	Θ120	Εφαρμογές Η/Υ στην Διοίκηση Ι
	0079	Αντωνοπούλου	Ε067	Η/Υ Εισαγωγή στην Πληροφορική
	0079	Αντωνοπούλου	Θ114	Η/Υ Εισαγωγή στην Πληροφορική
	0078	Παπανικολάου	Ε078	Δημόσιο Λογιστικό
	0078	Παπανικολάου	Θ138	Δημόσιο Λογιστικό
	0078	Παπανικολάου	Ε076	Διοικητική Λογιστική
	0078	Παπανικολάου	Θ135	Διοικητική Λογιστική
	0078	Παπανικολάου	Ε066	Γενική Λογιστική
	0078	Παπανικολάου	Θ113	Γενική Λογιστική
	0079	Αντωνοπούλου	Ε073	Εφαρμογές Η/Υ στην Διοίκηση ΙΙ

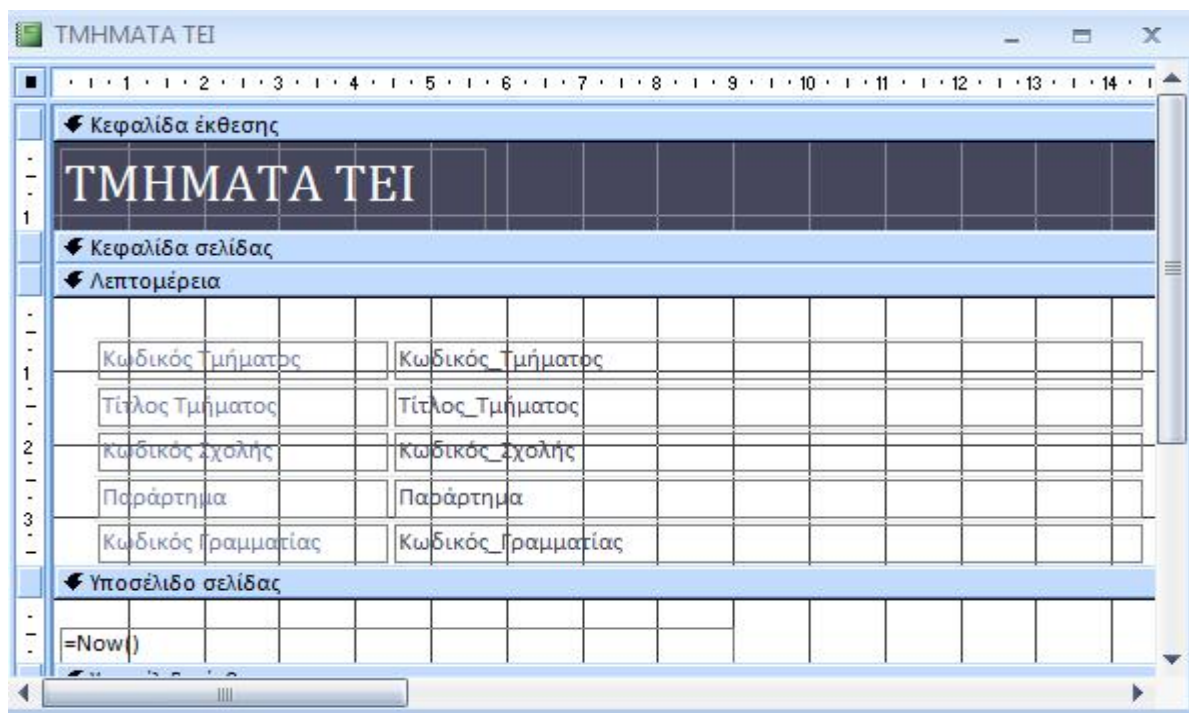
Σχήμα 3.36 Έκθεση Καθηγητών Τμήματος Επιχειρηματικού Σχεδιασμού Και Πληροφοριακών Συστημάτων Και Διδαχθέντα Μαθήματα.



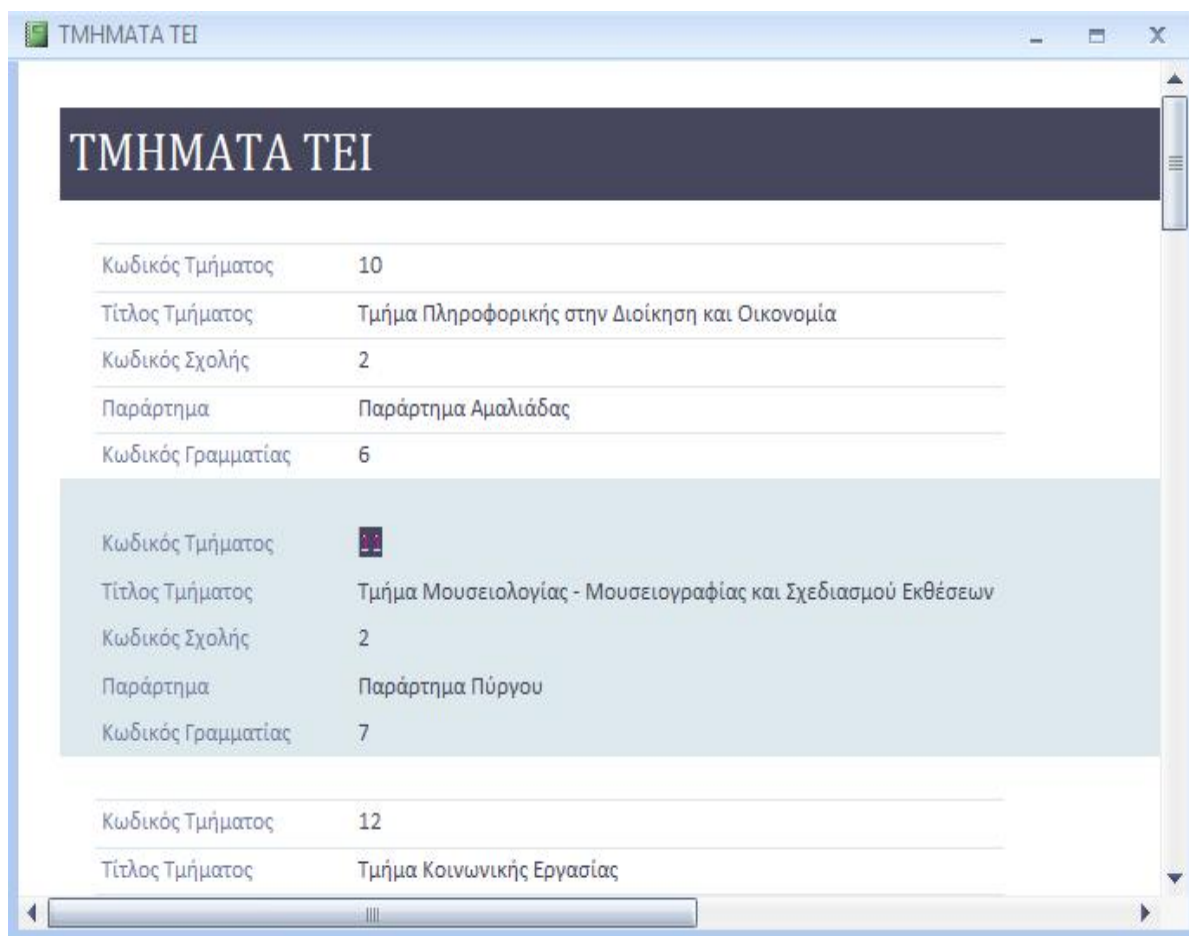
Σχήμα 3.37 Έκθεση Σε Προβολή Σχεδίασης Τμημάτων ΤΕΙ Κλάδου Διοίκησης και Οικονομίας



Σχήμα 3.38 Έκθεση Τμημάτων ΤΕΙ Κλάδου Διοίκησης και Οικονομίας



Σχήμα 3.39 Έκθεση Σε Προβολή Σχεδίασης Τμημάτων ΤΕΙ Πάτρας.



Σχήμα 3.40 Έκθεση Τμημάτων ΤΕΙ Πάτρας.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Με το τέλος της Πτυχιακής Εργασίας κατανοήσαμε πόσο πολύπλοκο αλλά παράλληλα πόσο εύχρηστο είναι ένα Πληροφοριακό Σύστημα μέσα σε έναν οργανισμό και δει σε ένα κλάδο που έχει ως αντικείμενο την Εκπαίδευση.

Το ΤΕΙ της Πάτρας αποτελεί ένα από τα πιο σύγχρονα Εκπαιδευτικά Ιδρύματα της Χώρας μας. Η ποικιλία των Κλάδων των διαφόρων Τμημάτων από τα οποία απαρτίζεται το συγκεκριμένο Εκπαιδευτικό Ίδρυμα, επιτάσσει πραγματικά την υλοποίηση ενός σύγχρονου και εύχρηστου Πληροφοριακού Συστήματος τόσο για την διευκόλυνση των Υπαλλήλων που απαρτίζουν κάθε Τμήμα σε ότι έχει σχέση με τις εργασίες που έχουν αναλάβει, όσο και για τους ίδιους τους Σπουδαστές που επιθυμούν να εξυπηρετούνται με άνεση και ταχύτητα.

Το θεωρητικό κομμάτι της Εργασίας μας διευκόλυνε στο να κατανοήσουμε ότι οποίο Σύστημα καλούμαστε να δημιουργήσουμε πρέπει να γίνεται με συνέπεια, εξετάζοντας όλες εκείνες τις παραμέτρους που θα το καταστήσουν εύχρηστο, ταχύτατο και αποδοτικό με όσο το δυνατόν χαμηλότερο κόστος παραγωγής.

Το πρακτικό κομμάτι αποτέλεσε και το πιο δύσκολο τμήμα της Εργασίας μας. Αυτό συνέβη διότι τα δεδομένα που έπρεπε να συνδυαστούν ήταν μεγάλα σε αριθμό και απαιτούσαν πολύ καλή γνώση Προγραμματισμού και οικειότητα με το πρόγραμμα δημιουργίας μιας Βάσης Δεδομένων. Προσπαθήσαμε λοιπόν με τις γνώσεις που διαθέταμε και με όσες αποκτήσαμε στην πορεία μέσω της ανάγνωσης αρκετών βιβλίων και άρθρων σχετιζόμενων με το θέμα μας, να δημιουργήσουμε μια Βάση Δεδομένων όχι πλασματική ή ενδεικτική αλλά να πλησιάζει όσο το δυνατόν περισσότερο σε μία πραγματική Βάση Δεδομένων ενός τέτοιου Εκπαιδευτικού Ιδρύματος. Έτσι αφού δημιουργήθηκαν οι βασικοί και βοηθητικοί Πίνακες, θεωρήσαμε σωστό να δημιουργήσουμε έναν αρκετά μεγάλο αριθμό εγγραφών ώστε να υπάρχει ποικιλία στα δεδομένα μας αλλά και να πλησιάζει στις προδιαγραφές και στις απαιτήσεις ενός πραγματικού Πληροφοριακού Συστήματος που θα μπορούσε να διαθέτει ένα τόσο μεγάλο Εκπαιδευτικό Ίδρυμα όπως το ΤΕΙ της Πάτρας.

Θεωρούμε ότι το Σύστημα που δημιουργήσαμε μέσω του προγράμματος Microsoft Access 2007 καλύπτει σε ικανοποιητικό βαθμό τις προσδοκίες και τις απαιτήσεις τόσο του Χρήστη – Υπαλλήλου όσο και τις ανάγκες του Τελικού Αποδέκτη, ο οποίος μπορεί να εκμεταλλευτεί και ικανοποιηθεί από τις «εκροές-αποτελέσματα» αυτού του Συστήματος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Δημητριάδης Α. (2001), «**Διοίκηση – Διαχείριση Πληροφοριακών Συστημάτων (Management Information Systems)**», Εκδόσεις Νέων Τεχνολογιών.
- Προβατάς Δ. (2001), «**Ανάπτυξη Εφαρμογών Σε Προγραμματιστικό Περιβάλλον**», Εκδόσεις Σύγχρονη Πέννα (Σχολική Έκδοση: Γ΄ Λυκείου)
- Γιαννακόπουλος, Δημόπουλος, Πολλαλής (2004), «**Πληροφοριακά Συστήματα Επιχειρήσεων**», Εκδόσεις Σταμούλης.
- Κιουντούζης (2000), «**Μεθολογίας Ανάλυσης Και Σχεδιασμού Πληροφοριακών συστημάτων**», Εκδόσεις Μπένου (2^η Έκδοση).
- Χατζηγεωργίου Α. (2006), «**Αντικειμενοστραφής Σχεδίαση**», Εκδόσεις Κλειδάριθμος.
- Δημητριάδης Α. (1998), «**Διοίκηση – Διαχείριση Πληροφοριακών Συστημάτων**», Εκδόσεις Νέων Τεχνολογιών.
- Χαϊκάς Κ. (2005), «**Βασικά Θέματα Για Τα Πληροφοριακά Συστήματα Διαχείρισης Επιχειρησιακών Πόρων (E.R.P)**», Εκδόσεις Γκιούρδας.
- Γερογιάννης Β., Κακαρόντζας Γ., Καμέας Α., Σταμέλος Γ., Φιτσιλής Π. (2006), «**Αντικειμενοστραφής Ανάπτυξης Λογισμικού Με Την UML**», Εκδόσεις Κλειδάριθμος.
- Λαοπόδης Βασίλιος (1996), «**Ανάπτυξη Πληροφοριακών Συστημάτων: Ανάλυση και Σχεδιασμός Συστημάτων**», Εκδόσεις Νέων Τεχνολογιών.
- Παπαιοκονόμου Ε. – Συνεργαζόμενο Ίδρυμα : ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ (2006), «**Αντικειμενοστραφής Ανάλυση Και Σχεδιασμός Συστήματος Ελέγχου Διοίκησης (management control system) με την χρήση της Unified Modeling Language (UML)**», Αρχείο Ψηφιακής Βιβλιοθήκης Πειραιώς.
- Δενδρινός Μ. (2003), «**Βάσεις Δεδομένων**», Εκδόσεις Libris-Tech A.E.
- Ζομπανάκης, Παρσόπουλος, Ανδρέου, Συλλογικό Έργο (2003), «**Intelligent Information Systems Applied To Complicated Defence Problems**», Εκδόσεις Παπαζήσης.
- Λόντον Τζέιν – Λόντον Κάνεθ (2009), «**Πληροφοριακά Συστήματα Διοίκησης**», Εκδόσεις Κλειδάριθμος.
- Pflaegeer (2004), «**Τεχνολογία Λογισμικού: Θεωρία Και Πράξη**» (Τόμος 1).
- Avison David – Fitzgerald Guy (2006), «**Ανάπτυξη Προηγμένων Πληροφοριακών Συστημάτων**», Εκδόσεις Νέων Τεχνολογιών.
- Presman, R.S. (2005), «**Software Engineering: A Practitioner’s Approach**». 6th ed. International Edition, McGraw – Hill.
- Sommerville (2004), «**Software Engineering**», 7th ed. Pearson.
- Avgerou, C And Conford T. (1998), «**Developing Information Systems: Concepts, Issues And Practice**», Macmillan 2nd ed.
- Flynn D. (1998), «**Information Systems Requirement: Determination and Analysis**», 2nd ed. McGraw – Hill.
- Elmarsi R. – S.B. Navathe (2001), «**Θεμελιώδης Αρχές Συστημάτων Βάσεων Δεδομένων**», Εκδόσεις Δίαυλος (Τόμος Α΄, 3^η Αναθεωρημένη Έκδοση).
- Ian Sommerville, Addison-Wesley (1998), «**Software Engineering**», 5th ed.
- Stephen R. Schach (1999), «**Classical and Object-Oriented Software Engineering with UML and Java**», McGraw-Hill, 4th Ed.
- Dunham Margaret (2004), «**Data Mining**», Εκδόσεις Νέων Τεχνολογιών.

- Grady Booch and Ivar Jacobson and James Rumbaugh (1998), «**The Unified Modeling Language User Guide**», Addison-Wesley, Object Technology Series.
- Chris Marshall (1999), «**Enterprise Modeling with UML: Designing Successful Software through Business Analysis**», Addison-Wesley, Object Technology Series.
- Roger A. Pressman, Roger A. Pressman (1997), «**Software Engineering: A Practitioner's Approach**», McGraw-Hill, 4th Ed.
- Laudon, K.C. (2004), «**Management Information Systems: Managing the digital firm**», Prentice Hall, 8th Ed.
- Laudon & Laudon (2002), «**Συστήματα Πληροφοριών Διοίκησης, Οργάνωση και τεχνολογία στη δικτυωμένη επιχείρηση**» Εκδόσεις Κλειδάριθμος.
- Davis W., «**Management Information Systems: A Managerial Perspective**» West Publishing Co.
- Gupta U, «**Management Information Systems: A Managerial Perspective**» West Publishing Co.
- Hicks, J.O.Jr, «**Management Information Systems: A User Perspective**» West Publishing Co.
- Γουλτίδης Χρήστος (2004), «**Βάσεις Δεδομένων Access 2002**», Εκδόσεις Κλειδάριθμος.
- Σελλής Τιμολέων (1997), «**Αντικειμενοστραφή Συστήματα Διαχείρισης Β.Δ.**», Εκδόσεις Παπασωτηρίου.
- Πάγκαλος Γεώργιος (1999), «**Βάσεις Δεδομένων**», Εκδόσεις Κυριακίδη ΑΦΟΙ.