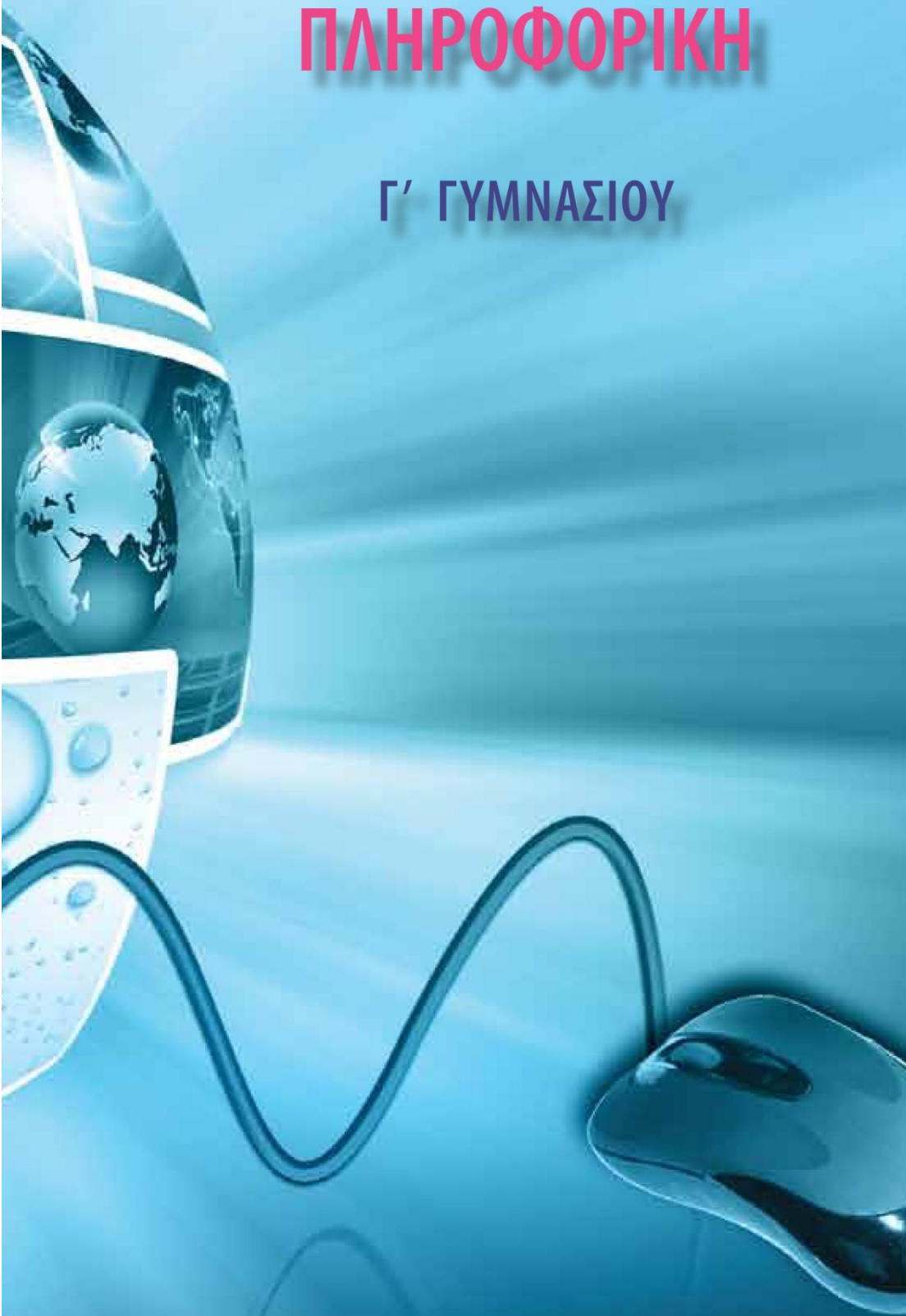


ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΠΟΛΙΤΙΣΜΟΥ

ΠΛΗΡΟΦΟΡΙΚΗ

Γ' ΓΥΜΝΑΣΙΟΥ

Σημειώσεις



Πληροφορική και Επιστήμη Ηλεκτρονικών Υπολογιστών

Γ' Γυμνασίου

Σημειώσεις

Συγγραφή:

Οι καθηγητές Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών που αναφέρονται στην αρχή της κάθε ενότητας με την καθοδήγηση και συμβολή των ακόλουθων Συμβούλων Καθηγητών Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών:

Σωκράτης Μυλωνάς (ΒΔ)
Νικόλαος Ζάγγουλος
Αλέξανδρος Παπαλυσάνδρου (ΒΔ)
Μιχάλης Διονυσίου (ΒΔ)
Πάυλος Παυλικκάς
Ξένιος Ξενοφώντος
Ιωάννης Ιωάννου
Μιλτιάδης Χαριλάου
Ηλίας Θεοδώρου

Ηλεκτρονική σελίδωση:

Σωκράτης Μυλωνάς
Βοηθός Διευθυντής Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών

Νικόλαος Ζάγγουλος
Καθηγητής Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών

Εποπτεία:

Μάριος Μιλτιάδου
Μιχάλης Τορτούρης
Επιθεωρητές Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών

Γλωσσική επιμέλεια:

Μαριάννα Χριστόφια Παλάτου
Λειτουργός Υπηρεσίας Ανάπτυξης Προγραμμάτων

Σχεδιασμός εξωφύλλου:

Μιχάλης Θεοχαρίδης
Λειτουργός Υπηρεσίας Ανάπτυξης Προγραμμάτων

Συντονισμός έκδοσης:

Χρίστος Παρπούνας
Συντονιστής Υπηρεσίας Ανάπτυξης Προγραμμάτων

Α' Έκδοση 2013

Ανατύπωση 2016 (Αναθεωρημένη)

Εκτύπωση: ΣΥΚΑΡΗΣ GRAPH

© ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΠΟΛΙΤΙΣΜΟΥ
ΠΑΙΔΑΓΩΓΙΚΟ ΙΝΣΤΙΤΟΥΤΟ ΚΥΠΡΟΥ
ΥΠΗΡΕΣΙΑ ΑΝΑΠΤΥΞΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ

ISBN: 978-9963-0-4685-0



Στο εξώφυλλο χρησιμοποιήθηκε ανακυκλωμένο χαρτί σε ποσοστό τουλάχιστον 50%, προερχόμενο από διαχείριση απορριμμάτων χαρτιού. Το υπόλοιπο ποσοστό προέρχεται από υπεύθυνη διαχείριση δασών.

Πρόλογος

Με ιδιαίτερη χαρά και ικανοποίηση προλογίζω το έντυπο υλικό για την υποστήριξη της διδασκαλίας του μαθήματος της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών στη Γ' Γυμνασίου.

Το υλικό αυτό είναι διαχωρισμένο σε δύο τόμους, τις *Σημειώσεις* και το *Τετράδιο*. Ο κάθε τόμος είναι οργανωμένος σε επτά ενότητες. Κάθε ενότητα καλύπτει το φάσμα των γνώσεων, των ικανοτήτων και των δεξιοτήτων που προκύπτουν από τον γενικό σκοπό του μαθήματος, να προετοιμάσει δηλαδή τους μαθητές και τις μαθήτριες για την ένταξή τους στην Κοινωνία της Πληροφορίας. Τους παρέχει ικανοποιητικές γνώσεις και τους καλλιεργεί τις απαραίτητες ικανότητες, δεξιότητες και στάσεις που θα τους επιτρέψουν την υπεύθυνη, την ενσυνείδητη, την ασφαλή, την αποδοτική και τη δημιουργική χρήση σύγχρονων τεχνολογιών της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών.

Οι *Σημειώσεις* δεν αποτελούν διδακτικό εγχειρίδιο αλλά υλικό αναφοράς, το οποίο περιέχει παραδείγματα, επεξηγήσεις και επιπρόσθετες πληροφορίες και χρησιμοποιείται ως πηγή άντλησης πληροφοριών. Στην αρχή του κάθε κεφαλαίου διατυπώνονται οι διδακτικοί στόχοι και στο τέλος υπάρχει σύνοψη των βασικών εννοιών για εύκολη αναφορά.

Το *Τετράδιο*, αποτελείται από ενδεικτικά φύλλα εργασίας, διαχωρισμένα σε δραστηριότητες εμπέδωσης για την ανάπτυξη νέων γνώσεων, ικανοτήτων και δεξιοτήτων και αξιολόγησης του βαθμού απόκτησης των γνώσεων αυτών, σε επιπρόσθετες δραστηριότητες για εμβάθυνση και διεύρυνση των γνώσεων, των ικανοτήτων και των δεξιοτήτων και σε δραστηριότητες για το σπίτι. Περιλαμβάνονται ποικίλες δραστηριότητες με διαφοροποιημένο δείκτη δυσκολίας, ώστε να είναι δυνατή η προσαρμογή του μαθήματος στις ικανότητες, στις απαιτήσεις και στις ανάγκες όλων των μαθητών και των μαθητριών με γνώμονα τους δείκτες επιτυχίας του αναλυτικού προγράμματος. Υιοθετούνται σύγχρονα διδακτικά μοντέλα, που στηρίζονται στην προώθηση εξελιγμένων μεθόδων οικοδόμησης της γνώσης. Βασικό στοιχείο της διδακτικής προσέγγισης είναι η σύνδεση του μαθήματος με την καθημερινότητα των μαθητών και των μαθητριών με απώτερο στόχο την ανάπτυξη της δημιουργικής και κριτικής σκέψης και την καλλιέργεια δεξιοτήτων επικοινωνίας, συνεργασίας και συστηματικών τρόπων προσέγγισης για την επίλυση προβλημάτων, ώστε να είναι έτοιμοι/ες για να λειτουργήσουν στο συνεχώς μεταβαλλόμενο περιβάλλον των τεχνολογιών της πληροφορίας και των επικοινωνιών.

Για τη δημιουργία του υλικού αυτού αλλά και του ηλεκτρονικού υλικού που το συνοδεύει, η ομάδα υποστήριξης της εφαρμογής του Νέου Αναλυτικού Προγράμματος για το μάθημα της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών έχει υιοθετήσει ένα καινοτόμο μοντέλο: Το υλικό δημιουργείται και διαμορφώνεται από τους υπό επιμόρφωση μάχιμους εκπαιδευτικούς, υπό την καθοδήγηση των υποστηρικτών και μέσα από τις εισηγήσεις τους αναθεωρείται συνεχώς.

Επιθυμώ να εκφράσω τις ευχαριστίες μου στην ομάδα επιμόρφωσης και υποστήριξης, στην Υπηρεσία Ανάπτυξης Προγραμμάτων αλλά και στους/στις μάχιμους/ες εκπαιδευτικούς για τη συμβολή τους στην ανάπτυξη του υλικού αυτού.

Δρ Ζήνα Πουλλή
Διευθύντρια Μέσης Εκπαίδευσης

ΠΕΡΙΕΧΟΜΕΝΑ

Εισαγωγή	11
ΕΝΟΤΗΤΑ Γ2 Το Υλικό/ Αρχιτεκτονική Ηλεκτρονικών Υπολογιστών	13
Γ2.1 Η Λειτουργική Σχέση του Υλικού και του Λογισμικού	15
1. Εισαγωγή	15
2. Δεδομένα, Πληροφορίες και μονάδες μέτρησής τους	15
3. Τα κύρια μέρη ενός υπολογιστή	16
4. Θύρες σύνδεσης	16
5. Η Κεντρική Μονάδα Επεξεργασίας/Αρχιτεκτονική ηλεκτρονικών υπολογιστών	17
6. Μονάδες εισόδου και μονάδες εξόδου	18
7. Μονάδες αποθήκευσης (Κύρια και Βοηθητική Μνήμη)	21
8. Το λογισμικό	24
Βασικές Έννοιες	25
Γ2.2 Το Υπολογιστικό Σύστημα ως Ενιαίο Σύνολο	27
1. Εισαγωγή	27
2. Το Λογισμικό Εφαρμογών	27
3. Το Λειτουργικό Σύστημα και άλλο Λογισμικό Συστήματος	28
4. Ο Χρήστης	29
5. Το Υπολογιστικό Σύστημα	29
Βασικές Έννοιες	30
Γ2.3 Προβλήματα του Υπολογιστικού Συστήματος και η Αντιμετώπισή τους.....	31
1. Προβλήματα που Σχετίζονται με το Υλικό	31
2. Προβλήματα που Σχετίζονται με το Λογισμικό.....	34
3. Προβλήματα που Σχετίζονται με τον Χρήστη	38
Βασικές Έννοιες.....	42
Γ2.4 Πώς Συγκρίνουμε και Πώς Επιλέγουμε Υπολογιστικά Συστήματα	45
1. Εισαγωγή	45
2. Ο χρήστης και οι υπολογιστικές ανάγκες του	45
3. Επιλογή λογισμικού εφαρμογών και λειτουργικού συστήματος	46
4. Επιλογή υλικού	47
5. Πώς επιλέγουμε ένα υπολογιστικό σύστημα για κάποιον χρήστη	47
Βασικές Έννοιες.....	54
ΕΝΟΤΗΤΑ Γ6 Βάσεις Δεδομένων και Ανάπτυξη Πληροφοριακών Συστημάτων.....	55
Γ6.1 Εισαγωγή στις Βάσεις Δεδομένων.....	57
1. Βάση Δεδομένων (Database)	57

2. Ηλεκτρονικές (μηχανογραφημένες) βάσεις δεδομένων	57
3. Πλεονεκτήματα των μηχανογραφημένων βάσεων δεδομένων	58
4. Μειονέκτημα των μηχανογραφημένων βάσεων δεδομένων	59
5. Βάσεις Δεδομένων από την καθημερινή μας ζωή	59
Βασικές Έννοιες	60
Γ6.2 Βασικά Δομικά Συστατικά Στοιχεία και Κύρια Αντικείμενα μιας Βάσης Δεδομένων.....	61
1. Βασικά Δομικά Συστατικά Στοιχεία μιας Βάσης Δεδομένων	61
2. Ειδικές Σχέσεις μεταξύ πινάκων (Relationships)	62
3. Κύρια Αντικείμενα Βάσης Δεδομένων (Objects)	64
4. Η Δομή μιας Βάσης Δεδομένων	64
5. Εφαρμογή Βάσεων Δεδομένων (Microsoft Access)	64
6. Ξεκίνημα της Εφαρμογής Βάσεων Δεδομένων	65
7. Δημιουργία νέας κενής βάσης δεδομένων (Blank database).....	65
8. Αποθήκευση βάσης δεδομένων (Save).....	65
9. Άνοιγμα βάσης δεδομένων (Open).....	66
10. Κλείσιμο βάσης δεδομένων (Close Database)	66
11. Κλείσιμο Εφαρμογής Βάσεων Δεδομένων (Exit)	66
Βασικές Έννοιες	67
Υπόμνημα	68
Γ6.3 Πίνακες.....	69
1. Πίνακας (Table).....	69
2. Δημιουργία Πίνακα με τη χρήση της Προβολής Σχεδίασης (Table Design)	69
3. Δημιουργία Πίνακα με τη χρήση της Προβολής Φύλλου Δεδομένων (Table)	70
4. Κανόνες Ονομασίας Πεδίων.....	71
5. Τύποι Δεδομένων (Data Type)	71
6. Ιδιότητες Πεδίων (Field Properties)	72
7. Πρωτεύον Κλειδί (Primary Key)	73
8. Πώς καθορίζουμε το Πρωτεύον Κλειδί.....	75
9. Διαγραφή Πίνακα (Delete Table)	75
10. Προβολές για Πίνακα (Views)	75
Υπόμνημα	77
Γ6.4 Εγγραφές.....	79
1. Εισαγωγή, Επεξεργασία, Διαγραφή και Μετακίνηση μεταξύ των εγγραφών σε ένα πίνακα	79
2. Μορφοποίηση Πίνακα	80
3. Ταξινόμηση Δεδομένων (Sort).....	83
4. Χρήση της Εντολής για την Αναζήτηση Πληροφοριών (Find)	84
5. Χρήση Φίλτρων για την Αναζήτηση Πληροφοριών (Filter)	85
Υπόμνημα	88

Γ6.5	Σχέσεις Πινάκων	89
1.	Εισαγωγή	89
2.	Πλεονεκτήματα που προκύπτουν από τις Σχέσεις μεταξύ των Πινάκων	90
3.	Ο ρόλος του Πρωτεύοντος Κλειδιού στον συσχετισμό των πινάκων.....	90
4.	Είδη Ειδικών Σχέσεων (Relationships)	91
	Βασικές Έννοιες.....	94
Γ6.6	Δημιουργία και Διαγραφή Σχέσεων Μεταξύ δύο Πινάκων	95
1.	Εισαγωγή	95
2.	Δημιουργία σχέσης Ένα-Προς-Πολλά μεταξύ δύο πινάκων (One-To-Many, 1-∞)	95
3.	Διαγραφή Σχέσης μεταξύ δύο πινάκων (Delete Relationship)	97
4.	Καταχώριση Εγγραφών στον δεύτερο συσχετιζόμενο πίνακα	97
	Υπόμνημα	98
Γ6.7	Φόρμες Εισαγωγής Δεδομένων	99
1.	Φόρμα Εισαγωγής Δεδομένων (Form)	99
2.	Δημιουργία Φόρμας με το Εργαλείο Αυτόματης Φόρμας (Form)	99
3.	Δημιουργία Φόρμας με τη χρήση του Οδηγού Φορμών (Form Wizard)	99
4.	Εισαγωγή, Επεξεργασία και Μετακίνηση μεταξύ των εγγραφών σε μια φόρμα εισαγωγής δεδομένων	100
5.	Διαγραφή Εγγραφής σε μια φόρμα (Delete Record)	100
6.	Προβολές για Φόρμα (Views).....	100
7.	Μορφοποίηση Φόρμας	102
	Υπόμνημα	107
Γ6.8	Ερωτήματα	109
1.	Ερωτήματα (Queries).....	109
2.	Δημιουργία Ερωτήματος με τη χρήση του Οδηγού Ερωτημάτων (Query Wizard)	110
3.	Δημιουργία Ερωτήματος με τη χρήση της Προβολή Σχεδίασης (Query Design)	110
4.	Τροποποίηση Ερωτήματος.....	111
5.	Διαγραφή Ερωτήματος (Delete Query)	112
6.	Το Παράθυρο της Προβολής Σχεδίασης ενός Ερωτήματος.....	112
7.	Κριτήρια Ερωτημάτων (Criteria).....	113
	Υπόμνημα	115
Γ6.9	Ερωτήματα από Δύο Συσχετιζόμενους Πίνακες	117
1.	Ερωτήματα από δύο Συσχετιζόμενους Πίνακες (Queries in Related Tables).....	117
2.	Δημιουργία ερωτήματος με τη χρήση του Οδηγού Ερωτημάτων (Query Wizard)	117
3.	Δημιουργία ερωτήματος με τη χρήση της Προβολής Σχεδίασης (Query Design)	118
	Υπόμνημα	119
Γ6.10	Εκθέσεις	121
1.	Εκθέσεις (Reports).....	121
2.	Δημιουργία Έκθεσης με το Εργαλείο Αυτόματης Έκθεσης (Report).....	121
3.	Δημιουργία Έκθεσης με τη χρήση του Οδηγού Εκθέσεων (Report Wizard).....	122

ΠΕΡΙΕΧΟΜΕΝΑ

4. Προβολές για Έκθεση (Views)	122
5. Τροποποίηση Έκθεσης.....	124
6. Διαγραφή Έκθεσης (Delete Report)	124
7. Εκτύπωση Έκθεσης (Print Report)	124
Υπόμνημα	125
ΕΝΟΤΗΤΑ Γ7 Αλγοριθμική Σκέψη, Προγραμματισμός και Σύγχρονες Εφαρμογές Πληροφορικής.....	127
Γ7.1 Εισαγωγή στους Αλγορίθμους.....	129
1. Ο Κύκλος Ανάπτυξης Εφαρμογής – Ποια βήματα/φάσεις πρέπει να ακολουθήσουμε για να δημιουργήσουμε ένα πρόγραμμα	129
2. Τι είναι αλγόριθμος.....	132
3. Τα χαρακτηριστικά ενός αλγορίθμου	133
4. Τρόποι διατύπωσης αναπαράστασης αλγορίθμων	135
5. Αποδοτικός Αλγόριθμος	135
Βασικές Έννοιες	137
Γ7.2 Αναπαράσταση Αλγορίθμων με Λογικό Διάγραμμα	139
1. Λογικά Διαγράμματα	139
2. Δεδομένα, Επεξεργασία, Πληροφορίες (Ζητούμενα),	141
3. Σταθερές και Μεταβλητές.....	141
4. Αναπαράσταση αλγορίθμου με τη χρήση λογικών διαγραμμάτων	142
Βασικές Έννοιες	144
Γ7.3 Προβλήματα Ακολουθιακής Δομής	145
1. Ακολουθιακή Δομή	145
2. Προκαταρκτική Εκτέλεση	145
3. Μαθηματικές εκφράσεις	146
4. Προβλήματα με μία (1) είσοδο και έξοδο	146
5. Προβλήματα με μία (1) είσοδο και περισσότερες εξόδους.....	147
6. Προβλήματα με περισσότερες εισόδους και 1 έξοδο.....	149
7. Προβλήματα με περισσότερες εισόδους και εξόδους	150
Βασικές Έννοιες	152
Γ7.4 Προβλήματα Δομής Διακλάδωσης.....	153
1. Δομή διακλάδωσης ή Δομή επιλογής.....	153
2. Λογικοί τελεστές	161
3. Προβλήματα με περισσότερες από μια ανεξάρτητες συνθήκες	167
4. Φωλιασμένη Δομή Διακλάδωσης.....	169
Βασικές Έννοιες	172
Γ7.5 Γλώσσες Προγραμματισμού.....	173
1. Εισαγωγή.....	173
2. Γλώσσα Μηχανής	173
3. Γλώσσες Προγραμματισμού	174

4. Μεταγλωττιστές και Διερμηνείς	176
Βασικές Έννοιες	177
Γ7.6 Η Γλώσσα Προγραμματισμού Pascal.....	179
1. Εισαγωγή	179
2. Ενεργοποίηση του Pame	179
3. Δημιουργία νέου αρχείου	179
4. Το παράθυρο του περιβάλλοντος <i>Pame</i>	180
5. Εισαγωγή Κώδικα	180
6. Αποσφαλμάτωση (Debugging)	183
7. Μεταγλώττιση (Compile)	184
8. Εκτέλεση (Run)	184
9. Αποθήκευση (Save, Save as)	184
10. Άνοιγμα αρχείου (Open)	184
11. Η εντολή if...then...else	185
12. Μετατροπή Λογικού Διαγράμματος σε Πρόγραμμα	185
Βασικές Έννοιες	189
Υπόμνημα	190

Εισαγωγή

Όταν τον Μάρτιο του 2009 ξεκίνησε η προσπάθεια για εκπόνηση του Νέου Αναλυτικού Προγράμματος του μαθήματος της Πληροφορικής και Επιστήμης των Ηλεκτρονικών Υπολογιστών, τέθηκε ως προτεραιότητα η θεώρηση του μαθήματος μέσα από το πρίσμα των τριών πυλώνων της εκπαιδευτικής μεταρρύθμισης. Μέσα από το πρίσμα δηλαδή της απόκτησης ενός επαρκούς σώματος γνώσεων από τους μαθητές και τις μαθήτριες, την καλλιέργεια αξιών και την εκδήλωση συμπεριφορών που να συνάδουν με τη σύγχρονη έννοια της δημοκρατικής πολιτότητας και την καλλιέργεια κομβικών ικανοτήτων και δεξιοτήτων από τους μαθητές και τις μαθήτριες, ώστε να μπορούν να λειτουργούν στη διαμορφούμενη κοινωνία του 21^{ου} αιώνα.

Στην επιτροπή για την εκπόνηση του Αναλυτικού Προγράμματος συμμετείχαν ο ακαδημαϊκός Ανδρέας Ανδρέου (Αναπληρωτής Καθηγητής στο Τεχνολογικό Πανεπιστήμιο Κύπρου), οι επιθεωρητές Πληροφορικής Μάριος Μιλτιάδου και Μιχάλης Τορτούρης και οι καθηγητές Πληροφορικής Νικόλαος Ζάγγουλος, Ιωάννης Ιωάννου, Μηνάς Καραολής, Θεόδουλος Κωνσταντίνου, Μαρία Μαρδά, Σωκράτης Μυλωνάς και Παναγιώτης Παπέττας. Αποτέλεσμα της προσπάθειας που έγινε είναι το Αναλυτικό Πρόγραμμα που ήδη έχετε στα χέρια σας.

Το νέο Αναλυτικό Πρόγραμμα δεν περιέχει μεγάλο αριθμό νέων θεμάτων. Έγινε αναδιάρθρωση της θεματολογίας, ώστε να ενταχθεί κάτω από επτά ενότητες, οι οποίες αναπτύσσονται παράλληλα από τη μια τάξη στην επόμενη και αντικατοπτρίζουν βασικούς θεματικούς άξονες της Πληροφορικής και Επιστήμης των Ηλεκτρονικών Υπολογιστών:

- Βασικές Έννοιες
- Το Υλικό/Αρχιτεκτονική Υπολογιστών
- Λειτουργικά Συστήματα
- Λογισμικό Εφαρμογών
- Δίκτυα και Διαδίκτυο
- Βάσεις Δεδομένων και Ανάλυση Συστημάτων
- Αλγόριθμοι, Προγραμματισμός και Σύγχρονες Εφαρμογές Πληροφορικής.

Οι ενότητες αυτές αναπτύσσονται σταδιακά μέσα από διαβαθμισμένους στόχους/δείκτες επιτυχίας από τη μία τάξη στην επόμενη, παρέχοντας επιπρόσθετες ευκαιρίες στους μαθητές και στις μαθήτριες που δεν μπόρεσαν να τους κατακτήσουν σε μία τάξη να το πράξουν σε επόμενη, με παράλληλη εμβάθυνση. Θα πρέπει εδώ να τονιστεί ότι κάθε ενότητα αναπτύσσεται σε διαφορετικό βαθμό στην κάθε τάξη, με γνώμονα το γενικό επίπεδο ετοιμότητας των μαθητών/τριών στη συγκεκριμένη ηλικιακή ομάδα. Σημαντικό στοιχείο της νέας προσέγγισης είναι και η αλληλεξάρτηση των ενοτήτων, αλλά και η συγκεκριμενοποίηση και ο περιορισμός των εννοιών που θα πρέπει να γνωρίζουν οι μαθητές/τριες, με έμφαση στην εφαρμογή τους στην πράξη.

Τα ενδεικτικά φύλλα εργασίας που ακολουθούν βασίστηκαν σε διδακτικές προτάσεις που ετοιμάστηκαν από μάχιμους εκπαιδευτικούς και ορισμένα από αυτά δοκιμάστηκαν σε περιβάλλον πραγματικής τάξης. Παρέχονται ως παραδείγματα διάφορων μεθοδολογικών προσεγγίσεων, αλλά οι καθηγητές/τριες ενθαρρύνονται να τα προσαρμόσουν ή και να δημιουργήσουν άλλα, τα οποία να ανταποκρίνονται καλύτερα, τόσο στα ιδιαίτερα χαρακτηριστικά των μαθητών και μαθητριών τους, όσο και στη δική τους διδακτική προσέγγιση στο κάθε μάθημα.

ΕΝΟΤΗΤΑ Γ2 Το Υλικό/ Αρχιτεκτονική Ηλεκτρονικών Υπολογιστών

Για την προετοιμασία και τη συγγραφή του υλικού αυτής της ενότητας εργάστηκαν οι ακόλουθοι/θες Καθηγητές/τριες Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών που παρακολούθησαν το πρόγραμμα επιμόρφωσης μάχιμων εκπαιδευτικών για το Νέο Αναλυτικό Πρόγραμμα:

Βαλανός Κύπρος
Γεώργιου Γιώργος
Ζάγγουλος Νικόλαος (Σύμβουλος)
Καπιτανή Ρέα
Καρεκλάς Χρήστος
Κεττίρης Γιώργος
Κλεάνθους Κλεάνθης
Κοφτερός Ξένιος
Μιχαήλ Μιχάλης
Μυλωνάς Σωκράτης (Σύμβουλος)
Ξάνθου Ανδρέας
Ξενή Κούλλα
Παπαδούρης Γιώργος
Χαραλάμπους Χαράλαμπος
Χειμαρίδου Χριστίνα

Γ2.1 Η Λειτουργική Σχέση του Υλικού και του Λογισμικού

Τι θα μάθουμε σήμερα:

- ✦ Να περιγράψουμε τον κύκλο επεξεργασίας δεδομένων (Είσοδος / Επεξεργασία / Έξοδος / Αποθήκευση, πρόγραμμα)
- ✦ Να συσχετίζουμε τα μέρη του κύκλου επεξεργασίας δεδομένων, με τις διάφορες μονάδες / συσκευές υλικού.

1. Εισαγωγή

Όπως έχουμε ήδη μάθει (*Πληροφορική Α΄ Γυμνασίου, Σημειώσεις*), ο υπολογιστής είναι μια **μηχανή** η οποία **επεξεργάζεται, αποθηκεύει και μεταδίδει πληροφορίες**, σύμφωνα με μια σειρά από **εντολές** που ονομάζονται **πρόγραμμα**. Κύριο χαρακτηριστικό του είναι η **μεγάλη ταχύτητα** και **ακρίβεια** με την οποία εκτελεί τις εντολές, με αποτέλεσμα να μπορεί να επεξεργάζεται με συστηματικό τρόπο μεγάλες ποσότητες **δεδομένων**. Ένα άλλο σημαντικό χαρακτηριστικό είναι η δυνατότητα ο χρήστης να καθορίζει τι θα κάνει ο υπολογιστής κάθε φορά, απλά επιλέγοντας ποιο από τα προγράμματα που έχει διαθέσιμα επιθυμεί να εκτελέσει ο υπολογιστής.



Εικόνα 1 Βασικές λειτουργίες ενός Ηλεκτρονικού Υπολογιστή

Ένας υπολογιστής αποτελείται από διάφορα μέρη/εξαρτήματα τα οποία κατατάσσονται, ανάλογα με τον σκοπό για τον οποίο χρησιμοποιούνται σε μονάδες εισόδου, εξόδου, επεξεργασίας και αποθήκευσης. Όλοι οι υπολογιστές ακολουθούν αυτό τον βασικό σχεδιασμό. Φαίνονται όμως πολύ διαφορετικοί, διότι ο κατασκευαστής ανάλογα με τη χρήση του κάθε υπολογιστή επιλέγει διαφορετικές μονάδες εισόδου, εξόδου, επεξεργασίας και αποθήκευσης, καθώς και άλλα βοηθητικά εξαρτήματα (π.χ. μπαταρίες) για να φτιάξει έναν υπολογιστή. Τα μέρη/εξαρτήματα ενός υπολογιστή στο σύνολό τους αναφέρονται ως το **υλικό (hardware)**, σε αντίθεση με τα προγράμματα, τα οποία αναφέρονται ως **λογισμικό (software)**.

2. Δεδομένα, Πληροφορίες και μονάδες μέτρησής τους

Παρόλο που ο όρος πληροφορίες περιγράφει γενικά τι επεξεργάζεται ο υπολογιστής, με τον όρο αυτό συνήθως αναφερόμαστε στο προϊόν, το αποτέλεσμα της επεξεργασίας. Οι ακατέργαστες «πληροφορίες» τις οποίες παίρνει ο υπολογιστής στην είσοδο για να τις επεξεργαστεί αναφέρονται με τον όρο «δεδομένα».

Όλα τα δεδομένα/πληροφορίες από τη στιγμή που εισέρχονται στον υπολογιστή μέχρι και την έξοδό τους από αυτόν θα πρέπει να αναπαριστούνται με τρόπο που να ταιριάζει με τον τρόπο κατασκευής και λειτουργίας του. Ο υπολογιστής είναι κατασκευασμένος από ηλεκτρονικά κυκλώματα, τα οποία στην ουσία συμπεριφέρονται ως πολύπλοκοι διακόπτες. Επειδή ένας διακόπτης είτε επιτρέπει να περνά ηλεκτρικό ρεύμα είτε όχι, έτσι και οι πληροφορίες αναπαριστούνται με μόνο δύο τιμές το μηδέν, 0 (δεν περνά ρεύμα) και το ένα, 1 (περνά ρεύμα), από όπου και προκύπτει η βασική μονάδα μέτρησης, το δυαδικό ψηφίο (**binary digit**, ή **bit**). Για να αναπαραστήσει πιο σύνθετες μορφές πληροφοριών, ο

υπολογιστής χρησιμοποιεί πολλά bit μαζί σε ομάδες, και έτσι προκύπτει η επόμενη βασική μονάδα, το **byte** το οποίο αποτελείται από 8 bit στη σειρά, π.χ. 10010101. Ένα byte μπορεί να αναπαραστήσει 256 διαφορετικούς συνδυασμούς τιμών. Έτσι, μπορεί να αναπαραστήσει ακέραιους αριθμούς από το 0 μέχρι το 255 (για μεγαλύτερους αριθμούς χρησιμοποιούνται περισσότερα byte, συνήθως 4).

Τα γράμματα του αλφαβήτου και τα διάφορα σύμβολα (χαρακτήρες) που χρησιμοποιούμε για να γράψουμε ένα κείμενο, ο υπολογιστής χρειάζεται να τα αναπαραστήσει επίσης ως σειρές από bit. Για τον κάθε χαρακτήρα αντιστοιχεί ένας διαφορετικός συνδυασμός από bit, ο οποίος συμφωνήθηκε να ακολουθείται από όλους τους κατασκευαστές υπολογιστών, ώστε να μην προκύπτουν προβλήματα όταν μεταφέρονται πληροφορίες από έναν υπολογιστή σε άλλον. Η πιο διαδεδομένη αντιστοίχιση (κωδικοποίηση) χαρακτήρων βασίζεται στον Πίνακα Κωδικοποίησης **ASCII** (American Standard Code for Information Interchange).

Για να περιγράψουμε μια ποσότητα πληροφοριών ή τη χωρητικότητα κάποιας μονάδας αποθήκευσης χρησιμοποιούμε πολλαπλάσια του Byte:

(α) KB 1 **KiloByte** = 2^{10} = 1024 Byte $\approx$ 1.000 Byte

(β) MB 1 **MegaByte** = 2^{20} = 1024 KB $\approx$ 1000 KB = 1.000.000 Byte

(γ) GB 1 **GigaByte** = 2^{30} = 1024 MB $\approx$ 1000 MB = 1.000.000.000 Byte

(δ) TB 1 **TeraByte** = 2^{40} = 1024 GB $\approx$ 1000 GB = 1.000.000.000.000 Byte

(ε) PB 1 **PetaByte** = 2^{50} = 1024 TB $\approx$ 1000 TB = 1.000.000.000.000.000 Byte

3. Τα κύρια μέρη ενός υπολογιστή

Ένας οποιοσδήποτε υπολογιστής στο εσωτερικό του περιλαμβάνει τη **Μητρική Κάρτα** ή **Μητρική Πλακέτα (Motherboard)** η οποία είναι ένα μεγάλο τυπωμένο κύκλωμα, στο οποίο συνδέονται για να επικοινωνούν μεταξύ τους όλες οι μονάδες του Η/Υ. Πάνω στη μητρική πλακέτα συνδέονται σε ειδικές υποδοχές η Κεντρική Μονάδα Επεξεργασίας, η Μνήμη Τυχαιάς Προσπέλασης και οι μονάδες αποθήκευσης. Επίσης υπάρχουν εξειδικευμένα ολοκληρωμένα κυκλώματα τα οποία καταλήγουν στις **θύρες σύνδεσης (ports)** στο κουτί του υπολογιστή και επιτρέπουν τη σύνδεση και την επικοινωνία περιφερειακών συσκευών οι οποίες βρίσκονται στο εξωτερικό του υπολογιστή. Οι περιφερειακές συσκευές συνδέονται με κατάλληλα καλώδια στις θύρες σύνδεσης.

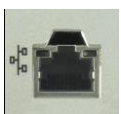
Στη μητρική πλακέτα υπάρχουν συνήθως κάποιες **υποδοχές επέκτασης** οι οποίες επιτρέπουν τη σύνδεση διάφορων καρτών επέκτασης οι οποίες δίνουν στον υπολογιστή νέες ή εξειδικευμένες δυνατότητες. Ενσωματωμένη στη μητρική κάρτα είναι επίσης η **Μνήμη Μόνο για Ανάγνωση (Read Only Memory, ROM)** στην οποία θα αναφερθούμε πιο κάτω.

Σε κάθε υπολογιστή συναντούμε επίσης το **Τροφοδοτικό (Power Supply)**, το οποίο συνδέει τον υπολογιστή με την παροχή ηλεκτρικού ρεύματος και προσαρμόζει (μετατρέπει) την ηλεκτρική τάση και το ρεύμα σε μορφή κατάλληλη για τα εξαρτήματα του Η/Υ. Σε φορητούς υπολογιστές υπάρχει μπαταρία, ενώ αντί το τροφοδοτικό να βρίσκεται εσωτερικά, υπάρχει εξωτερικός **φορτιστής** ο οποίος λειτουργεί με αντίστοιχο τρόπο.

4. Θύρες σύνδεσης

Οι Θύρες σύνδεσης (Ports) είναι υποδοχές στο κουτί του υπολογιστή οι οποίες χρησιμοποιούνται για τη σύνδεση του υπολογιστή με περιφερειακές συσκευές και με το

δίκτυο. Οι κυριότερες θύρες που συναντούμε σε έναν σύγχρονο υπολογιστή και ο ρόλος τους φαίνονται πιο κάτω:

Ονομασία Θύρας	Εικόνα	Περιγραφή/Λειτουργία
Θύρα USB (Universal Serial Bus)		Σύνδεση πολλών περιφερειακών συσκευών (εκτυπωτές, σαρωτές, ποντίκια, πληκτρολόγια, ψηφιακές φωτογραφικές μηχανές, μνήμη φλας, βιντεοκάμερες, κ.ά.).
Θύρα VGA		Σύνδεση οθόνης ή προβολέα.
Θύρα LAN/Ethernet		Σύνδεση υπολογιστή με το τοπικό δίκτυο.
Θύρα HDMI		Επιτρέπει τη μεταφορά ψηφιακού σήματος βίντεο Ψηλής Ευκρίνειας High Definition. Σύνδεση με τηλεόραση, οθόνη ή προβολέα.
Συνδέσεις κάρτας ήχου		- Πράσινο – Σύνδεση ηχείων (Audio Out) - Μπλε – Σύνδεση με άλλες συσκευές ήχου (line in) - Ροζ – Σύνδεση μικροφώνου (Mic)

Σε κάποιους όχι και τόσο καινούργιους υπολογιστές είναι πιθανό να συναντήσουμε και άλλα είδη θύρας σύνδεσης, όπως είναι οι θύρες **PS2** (για σύνδεση ποντικιού ή πληκτρολογίου), η **Σειριακή θύρα (Serial port)**, η **Παράλληλη θύρα (Parallel LPT port)** και η **Θύρα σύνδεσης παιχνιδιών (Game port)**. Αυτές έχουν αντικατασταθεί από θύρες USB, οι οποίες είναι απλούστερες και επιτρέπουν τη μεταφορά δεδομένων με μεγαλύτερη ταχύτητα.

5. Η Κεντρική Μονάδα Επεξεργασίας/Αρχιτεκτονική ηλεκτρονικών υπολογιστών

Η **Κεντρική Μονάδα Επεξεργασίας, ΚΜΕ (Central Processing Unit, CPU)** είναι ένα πολύπλοκο ολοκληρωμένο κύκλωμα το οποίο είναι υπεύθυνο για την επεξεργασία των δεδομένων. Λέγεται διαφορετικά και **μικροεπεξεργαστής (microprocessor)**.

Παρόλο που η κάθε ΚΜΕ εσωτερικά διαφέρει από κάθε άλλη, αφού οι κατασκευαστές συχνά εφευρίσκουν διάφορες τεχνικές για να βελτιώσουν είτε την ταχύτητα με την οποία η ΚΜΕ επεξεργάζεται τα δεδομένα, είτε το μέγεθος της ΚΜΕ, είτε κάποιο άλλο χαρακτηριστικό της, μπορούμε να πούμε ότι σε γενικές γραμμές κάθε ΚΜΕ περιέχει τα πιο κάτω βασικά μέρη (Εικόνα 2):

- Την **Αριθμητική και Λογική Μονάδα (Arithmetic and Logic Unit, ALU)**, η οποία είναι υπεύθυνη για την εκτέλεση αριθμητικών πράξεων και συγκρίσεων.



Εικόνα 2 Τα μέρη της Κεντρικής Μονάδας Επεξεργασίας

- Τη **Μονάδα Ελέγχου (Control Unit)**, η οποία είναι υπεύθυνη για τον έλεγχο και τον συντονισμό της ΚΜΕ, αλλά και του υπολογιστή γενικότερα. Εδώ έρχονται οι εντολές από ένα πρόγραμμα και εκτελούνται με τη σειρά. Η Μονάδα ελέγχου αναλύει την κάθε εντολή και ενεργοποιεί τις διάφορες μονάδες του υπολογιστή για να πάρει τα δεδομένα, δίνει οδηγίες στην Αριθμητική και Λογική Μονάδα ποια πράξη θα εκτελέσει και φροντίζει για την αποθήκευση και την έξοδο των πληροφοριών.
- Τους **Καταχωρητές (Registers)**, οι οποίοι είναι μικροί, αλλά γρήγοροι αποθηκευτικοί χώροι, οι οποίοι χρησιμοποιούνται για να κρατούν π.χ. τα ενδιάμεσα αποτελέσματα μιας πολύπλοκης επεξεργασίας.

Ο ρυθμός με τον οποίο η Κεντρική Μονάδα Επεξεργασίας εκτελεί τις εντολές καθορίζεται από τον **Χρονιστή (Clock)**, ο οποίος παράγει συγκεκριμένο αριθμό ηλεκτρικών παλμών ανά δευτερόλεπτο. Η συχνότητα των παλμών μετριέται σε MHz (εκατομμύρια ανά δευτερόλεπτο) ή σε GHz (δισεκατομμύρια ανά δευτερόλεπτο).

Επειδή η ΚΜΕ ελέγχει όλα τα μέρη του υπολογιστή καθορίζει σε μεγάλο βαθμό τον τρόπο με τον οποίο συνδέονται και επικοινωνούν τα διάφορα μέρη του υπολογιστή και κυρίως πώς κωδικοποιούνται οι εντολές σε ένα πρόγραμμα ώστε να αναγνωρίζονται ορθά από τη Μονάδα Ελέγχου. Έτσι, προκύπτουν «οικογένειες» ΚΜΕ που η καθεμιά ακολουθεί συγκεκριμένη **αρχιτεκτονική**.

Τα κύρια χαρακτηριστικά μιας ΚΜΕ, πέρα από την αρχιτεκτονική της (π.χ. x86, ARM, MIPS) είναι η μέγιστη ταχύτητα του χρονιστή (π.χ 2.2 GHz) και η κατανάλωση ενέργειας ή ισχύς σε **Βατ**, W (π.χ. 28 W).

6. Μονάδες εισόδου και μονάδες εξόδου

Μονάδες εισόδου είναι οι συσκευές με τις οποίες γίνεται η **εισαγωγή** των **δεδομένων** στον ηλεκτρονικό υπολογιστή για επεξεργασία. Στην κατηγορία αυτή ανήκουν το πληκτρολόγιο, το ποντίκι, η κάμερα, το μικρόφωνο, ο σαρωτής, κ.λπ.

Μονάδες εξόδου είναι οι συσκευές με τις οποίες γίνεται η **παρουσίαση** των **πληροφοριών**, οι οποίες παράγονται από την επεξεργασία που επιτελεί ο ηλεκτρονικός υπολογιστής. Στην κατηγορία αυτή ανήκουν η οθόνη, ο εκτυπωτής, ο προβολέας, τα ηχεία, τα ακουστικά, κ.λπ.

Το κάθε είδος μονάδας εισόδου ή εξόδου έχει τα δικά του χαρακτηριστικά. Για παράδειγμα, για έναν σαρωτή συνήθως ενδιαφέρει η ανάλυση (με πόση λεπτομέρεια σαρώνει μια εικόνα), το μέγιστο μέγεθος εικόνας που μπορεί να σαρώσει, σε τι είδους θύρα συνδέεται, το κόστος αγοράς του.

Πιο κάτω αναφέρονται κάποια βασικά χαρακτηριστικά των μονάδων εισόδου:

- **Τύπος σύνδεσης** με την οποία έχει τη δυνατότητα να συνδεθεί η συγκεκριμένη μονάδα με τον υπολογιστή (π.χ. USB, PS2, VGA, HDMI).
- **Ταχύτητα μεταφοράς δεδομένων** από τη μονάδα εισόδου προς τον υπολογιστή. Μετριέται σε bit ανά δευτερόλεπτο, **bits per second (bps)** και τα πολλαπλάσιά του, kbps, Mbps, κ.λπ. Εδώ χρειάζεται προσοχή στην ερμηνεία των πολλαπλασίων διότι 1kbps αντιστοιχεί σε 1000 bit ανά δευτερόλεπτο, ενώ όταν μετρούμε πολλαπλάσια του Byte, 1 kB ισοδυναμεί με 1024 Byte!
- **Ανάλυση** για συσκευές οι οποίες εισάγουν δεδομένα ή εξάγουν πληροφορίες σε μορφή εικόνας και η οποία αντιστοιχεί στη λεπτομέρεια της εικόνας. Για σαρωτές ή εκτυπωτές

μετριέται σε **κουκίδες ανά ίντσα (dots per inch, dpi)**, (π.χ. 600 dpi). Για οθόνες μετριέται ως ο αριθμός των εικονοστοιχείων σε κάθε σειρά (στήλες) επί τον αριθμό των σειρών (π.χ. 1280×1024). Για ψηφιακές κάμερες και βιντεοκάμερες μετριέται ως το σύνολο των εικονοστοιχείων που μπορούν να καταγράψουν (π.χ. 14 Megapixels).

- **Ταχύτητα** σάρωσης ή εκτύπωσης αντιστοιχεί με τον αριθμό **σελίδων ανά λεπτό (pages per minute, ppm)** που μπορεί αυτόματα να εκτυπώσει ο εκτυπωτής ή να σαρώσει ο σαρωτής. Για κάποια είδη εκτυπωτών που δέχονται συνεχές χαρτί αντί σε φύλλα, χρησιμοποιείται κάποτε ως μονάδα μέτρησης της ταχύτητας ο αριθμός **χαρακτήρων ανά δευτερόλεπτο (characters per second, cps)** που μπορεί να εκτυπώσει (π.χ. 70 cps).
- **Μέγεθος** της μονάδας εισόδου ή εξόδου. Για εκτυπωτές και σαρωτές αναφέρεται το μέγεθος του μεγαλύτερου εγγράφου το οποίο είναι αποδεκτό (π.χ. A4). Για τις οθόνες αναφέρεται ως το μήκος της διαγωνίου της εικόνας σε ίντσες (π.χ. 22”).
- **Λόγος διαστάσεων** οθόνης ή προβολέα, δηλαδή του πλάτους ως προς το ύψος (π.χ. 16:9).
- **Δυνατότητα χρώματος** με αναφορά στο βάθος χρώματος ή στα επίπεδα χρώματος (αποχρώσεις) που έχει τη δυνατότητα να αναγνωρίσει ή να εμφανίσει. Συνήθως μετριέται σε bit και είναι 8 ή 16 bit ανά χρώμα. Κάποτε αναφέρεται ο συνολικός αριθμός bit και για τα τρία βασικά χρώματα μαζί (δηλαδή 24 ή 48 bit). Σε κάποιες περιπτώσεις αναφέρεται ο συνολικός αριθμός των χρωμάτων/αποχρώσεων, π.χ. 16777216).
- **Ρυθμός ανανέωσης πλαισίων/καρέ (frame rate)** μιας οθόνης ή ενός βιντεοπροβολέα ή μιας βιντεοκάμερας, ο οποίος καθορίζει πόσες εικόνες ανά δευτερόλεπτο μπορεί να δημιουργήσει/παρουσιάσει και μετριέται σε Hz ή σε **frames per second, fps** (75 Hz, 16 fps).
- **Χρόνος ανταπόκρισης** για μια οθόνη ή έναν προβολέα είναι ο χρόνος που απαιτείται για να αλλάξει το χρώμα/η φωτεινότητα ενός εικονοστοιχείου. Μετριέται σε χιλιοστά του **δευτερολέπτου (milliseconds, ms)**, 5 ms.
- **Φωτεινότητα** ενός προβολέα (ή μιας οθόνης) είναι η ποσότητα του φωτός που εκπέμπει και μετριέται σε ANSI Lumens (π.χ. 1400).
- **Κόστος αγοράς** της συσκευής σε ευρώ.
- **Κόστος εκτύπωσης** για έναν εκτυπωτή είναι το κόστος των αναλωσίμων για εκτύπωση μιας σελίδας σε ευρώ (π.χ. €0.05).

Στον επόμενο πίνακα φαίνονται γνωστές μονάδες εισόδου και εξόδου, τα βασικά χαρακτηριστικά τους και οι μονάδες μέτρησής τους.

Μονάδα	Ονομασία	Χαρακτηριστικά και μονάδες μέτρησης
Εισόδου	Πληκτρολόγιο (Keyboard) 	Τύπος σύνδεσης (USB, PS2) Κόστος αγοράς (π.χ. €10)
	Ποντίκι (Mouse) 	Τύπος σύνδεσης (USB, PS2) Κόστος αγοράς (π.χ. €5)
	Κάμερα (Camera) 	Τύπος σύνδεσης (π.χ. USB) Ανάλυση (π.χ. 5 Megapixels) Δυνατότητα χρώματος (π.χ. 24 bit) Ρυθμός ανανέωσης πλαισίων/καρέ (π.χ. 16 fps) Κόστος αγοράς (π.χ. €60)
	Μικρόφωνο (Microphone) 	Κόστος αγοράς (π.χ. €20)
	Σαρωτής (Scanner) 	Το πιο διαδεδομένο είδος είναι ο επίπεδος σαρωτής (flatbed scanner) , αλλά υπάρχουν και σαρωτές εγγράφων (document scanners) με τροφοδότη (feeder). Τύπος σύνδεσης (π.χ. USB) Ανάλυση κουκίδες ανά ίντσα (π.χ. 1200 dpi) Ταχύτητα – με τροφοδότη (π.χ. 8 ppm) Μέγεθος (π.χ. A4) Δυνατότητα χρώματος (π.χ. 24 bit) Κόστος αγοράς (π.χ. €80)
Εξόδου	Οθόνη (Monitor) 	Τύπος σύνδεσης (VGA, DVI, HDMI) Ανάλυση (π.χ. 1280×1024) Μέγεθος (π.χ. 22") Λόγος διαστάσεων (π.χ. 16:9) Δυνατότητα χρώματος (π.χ. 24 bit) Ρυθμός ανανέωσης πλαισίων/καρέ (π.χ. 160 Hz) Χρόνος ανταπόκρισης (π.χ. 3 ms) Φωτεινότητα (π.χ. 1400 ANSI Lumens) Κόστος αγοράς (π.χ. €120)
	Βιντεο-προβολέας (Video Projector) 	Τύπος σύνδεσης (VGA, DVI, HDMI) Ανάλυση (π.χ. 1280×1024) Λόγος διαστάσεων (π.χ. 16:9) Δυνατότητα χρώματος (π.χ. 24 bit) Ρυθμός ανανέωσης πλαισίων/καρέ (π.χ. 160 Hz) Χρόνος ανταπόκρισης (π.χ. 5 ms) Φωτεινότητα (π.χ. 2800 ANSI Lumens) Κόστος αγοράς (π.χ. €300)

Μονάδα	Ονομασία	Χαρακτηριστικά και μονάδες μέτρησης
	Εκτυπωτής (Printer) 	Τα πιο γνωστά είδη εκτυπωτών είναι ο ακίδων (dot matrix)—τυπώνει χτυπώντας ακίδες σε μελανοταινία) ο ψεκασμού μελάνης (ink jet)—ψεκάζει κουκίδες μελανιού στο χαρτί) και ο λείζερ (laser)—ζωγραφίζει με φως το είδωλο της σελίδας σε κύλινδρο και χρησιμοποιεί λεπτή σκόνη, toner ως μέσο εκτύπωσης) Τύπος σύνδεσης (π.χ. USB) Ανάλυση (π.χ. 600 dpi) Ταχύτητα (π.χ. 24 ppm ή 70 cps) Μέγεθος (π.χ. A4) Δυνατότητα χρώματος (π.χ. 24 bit) Κόστος αγοράς (π.χ. €200) Κόστος εκτύπωσης (π.χ. €0.05/σελίδα)
Εξόδου	Σχεδιαστής (Plotter) 	Τύπος σύνδεσης (π.χ. USB) Μέγεθος (π.χ. A1) Κόστος αγοράς (π.χ. €500)
	Ηχεία (Speakers) 	Κόστος αγοράς (π.χ. €30)
	Ακουστικά (Headphones) 	Κόστος αγοράς (π.χ. €15)

Υπάρχουν πολλές εξειδικευμένες μονάδες εισόδου και εξόδου. Τα τελευταία χρόνια έχουν εμφανιστεί τα πολυμηχανήματα, τα οποία συνδυάζουν εκτυπωτή και σαρωτή μαζί, ενώ μπορούν να λειτουργούν ως τηλεομοιότυπα και φωτοτυπικές μηχανές. Επίσης, έχουν διαδοθεί οι οθόνες αφής, οι οποίες αποτελούνται από μια οθόνη καλυμμένη με μια ειδική διάφανη μεμβράνη ευαίσθητη στην αφή, η οποία σταδιακά αντικαθιστά το ποντίκι.

Σταδιακά θα εμφανίζονται νέες τεχνολογίες, όπως εύκαμπτες οθόνες και γυαλιά τρισδιάστατης προβολής, ενώ κάποιες παλιότερες τεχνολογίες σταδιακά εγκαταλείπονται (π.χ. οθόνες καθοδικού σωλήνα).

7. Μονάδες αποθήκευσης (Κύρια και Βοηθητική Μνήμη)

Οι μονάδες αποθήκευσης σε έναν υπολογιστή διαχωρίζονται σε δύο κατηγορίες, την **κύρια μνήμη**, η οποία αποτελείται από αποθηκευτικούς χώρους στους οποίους η ΚΜΕ έχει **άμεση πρόσβαση**, χωρίς τη μεσολάβηση πολύπλοκων κυκλωμάτων και στη **βοηθητική μνήμη** στην οποία η πρόσβαση περνά από πολύπλοκους μηχανισμούς.

Η κύρια μνήμη αποτελείται από ολοκληρωμένα κυκλώματα και είναι οργανωμένη με απλό τρόπο, ως μια σειρά από αριθμημένα πανομοιότυπα «κουτιά» («θυρίδες») με συγκεκριμένη χωρητικότητα δεδομένων το καθένα. Χρησιμοποιείται για (προσωρινή) αποθήκευση των εντολών οι οποίες αποτελούν τα προγράμματα τα οποία εκτελούνται από την ΚΜΕ, καθώς και τα δεδομένα και τις πληροφορίες που προκύπτουν από την επεξεργασία. Αποτελεί αποθηκευτικό χώρο με γρήγορη πρόσβαση για την ΚΜΕ. Σε αυτήν ανήκουν η **Μνήμη Τυχαίας Προσπέλασης (Random Access Memory, RAM)** η οποία έχει τη μορφή μικρών

καρτών με ολοκληρωμένα κυκλώματα οι οποίες τοποθετούνται σε ειδικές υποδοχές στη μητρική κάρτα και η **Μνήμη Μόνο για Ανάγνωση (Read Only Memory, ROM)**, η οποία αποτελείται από ολοκληρωμένο κύκλωμα ενσωματωμένο στη μητρική κάρτα.

Η μνήμη RAM έχει πολύ μεγαλύτερη χωρητικότητα από τη Μνήμη ROM και χρησιμοποιείται για προσωρινή αποθήκευση εντολών και δεδομένων για επεξεργασία από την ΚΜΕ. Εάν διακοπεί η παροχή ηλεκτρικού ρεύματος το περιεχόμενο της χάνεται. Κάθε κάρτα μνήμης RAM χαρακτηρίζεται από τον **τύπο** της (δηλαδή την αρχιτεκτονική της), τη **χωρητικότητά** της, το **χρόνο προσπέλασης** της, την **ταχύτητα μεταφοράς δεδομένων** και το **κόστος αγοράς** της. Ο πιο συνηθισμένος τύπος μνήμης για προσωπικούς υπολογιστές είναι ο Double Data Rate Synchronous Dynamic Random Access Memory (DDR SDRAM ή απλά DDR) και οι παραλλαγές του (π.χ. DDR3). Η χωρητικότητα μετριέται σε MB ή σε GB και ο χρόνος προσπέλασης σε τρισεκατομμυριοστά του δευτερολέπτου (ns). Αντί για τον χρόνο προσπέλασης, συνήθως δίνεται η **ταχύτητα μεταφοράς δεδομένων** σε MB/s (Megabyte ανά δευτερόλεπτο), π.χ. 6400 MB/s. Το κόστος είναι συνήθως ανάλογο με τα χαρακτηριστικά και συνήθως κυμαίνεται από € 10 μέχρι € 50.

Η μνήμη ROM κατασκευάζεται με διαφορετική τεχνολογία. Εδώ έχει αποθηκευτεί μόνιμα από τον κατασκευαστή το Βασικό Σύστημα Εισόδου/Εξόδου (Basic Input/Output System, BIOS), που είναι σειρά από εντολές που εκτελούνται κατά την εκκίνηση του υπολογιστή. Το περιεχόμενο της μνήμης ROM δεν επηρεάζεται από την απουσία ηλεκτρικού ρεύματος, ούτε και μπορεί να αλλάξει κατά τη χρήση του υπολογιστή.

Στην κατηγορία της βοηθητικής μνήμης ανήκουν όλες οι μονάδες αποθήκευσης, όπου αποθηκεύονται πληροφορίες πιο μακροπρόθεσμα, σε μορφή αρχείων. Η πρόσβαση της ΚΜΕ σε αυτές γίνεται μέσα από πολύπλοκα ηλεκτρονικά κυκλώματα και ο τρόπος και τα μέσα αποθήκευσης που χρησιμοποιούνται αποσκοπούν στην αποθήκευση όσο το δυνατόν μεγαλύτερης ποσότητας πληροφοριών, αντί στη γρήγορη πρόσβαση. Εδώ ανήκουν οι σκληροί δίσκοι (εσωτερικοί ή εξωτερικοί), η δισκέτα, η μνήμη φλας (USB, SD, κ.λπ.), οι οπτικοί δίσκοι (CD, CD-R/RW, DVD, DVD-R/RW), οι μαγνητικές ταινίες και όλες οι άλλες μονάδες «μόνιμης» αποθήκευσης πληροφοριών.

Με βάση τις τεχνολογίες που χρησιμοποιούνται σήμερα ως μέσα αποθήκευσης, οι μονάδες αυτές διαχωρίζονται σε μαγνητικά μέσα, σε οπτικά μέσα και σε μέσα στερεάς κατάστασης (δηλαδή ηλεκτρονικά μέσα). Τα μαγνητικά μέσα (σκληρός δίσκος, δισκέτα) καταγράφουν τις πληροφορίες σε επιφάνειες δίσκου καλυμμένες με μαγνητικό υλικό, διαχωρισμένες σε τροχιές (tracks) και σε τομείς (sectors) και είναι πλήρως επανεγγράψιμα. Τα οπτικά μέσα (CD-R, CD-RW, DVD-R, DVD-RW, κ.λπ.) καταγράφουν τις πληροφορίες σε μια τροχιά (track) με φωτεινή ακτίνα λέιζερ, η οποία αλλοιώνει κάποιο υλικό και το κάνει να αντανακλά ή να παρεμποδίζει το φως. Διατίθενται σε μορφή με δυνατότητα μίας μόνο εγγραφής ή που να είναι επανεγγράψιμη. Τα μέσα στερεάς κατάστασης (USB flash Drive, SD, Memory Stick) χρησιμοποιούν ηλεκτρονική τεχνολογία, κάτι μεταξύ της μνήμης ROM και της μνήμης RAM για την εγγραφή των πληροφοριών.

Τα βασικά χαρακτηριστικά των μέσων της βοηθητικής μνήμης είναι τα πιο κάτω:

- **Χωρητικότητα:** Πόσα byte συνολικά μπορούμε να αποθηκεύσουμε σε αυτό. (π.χ. 1.44 MB, 700 MB, 4.7 GB, 500 GB, 1 TB).
- **Ταχύτητα μεταφοράς δεδομένων:** Με ποια ταχύτητα μεταφέρονται τα δεδομένα από και προς τη μονάδα αποθήκευσης. Μετριέται σε bit ανά δευτερόλεπτο (bits per second, b/s) ή σε byte ανά δευτερόλεπτο (bytes per second, B/s) (π.χ. 50 kB/s, 1 Gb/s, 150

MB/s). Για σκληρούς δίσκους συνήθως δίνεται η ταχύτητα περιστροφής (πόσο γρήγορα περιστρέφεται) σε περιστροφές ανά λεπτό (revolutions per minute, rpm, π.χ. 7200 rpm), αλλά και ο χρόνος αναζήτησης (ο χρόνος που απαιτείται για να περιστραφεί ο σκληρός δίσκος και να μετακινηθούν οι κεφαλές στην κατάλληλη θέση για να διαβάσει ή να εγγράψει έναν τομέα, π.χ. 12 ms). Για οπτικούς δίσκους η ταχύτητα αναφέρεται ως πολλαπλάσιο μιας βασικής ταχύτητας, η οποία είναι 150 KB/s για CD και 1,32 MB/s για DVD (π.χ. CD στις 52x αντιστοιχεί σε 7,8 MB/s, ενώ DVD 16x αντιστοιχεί σε 21 MB/s).

- **Δυνατότητα εγγραφής/Επανεγγραφής:** κατά πόσο το αποθηκευτικό μέσο επιτρέπει εγγραφή ή/και επανεγγραφή (ισχύει ουσιαστικά για οπτικά μέσα αποθήκευσης).
- **Τύπος σύνδεσης:** Με ποιο τρόπο (πρότυπο, θύρα σύνδεσης, υποδοχή, καλώδιο) συνδέεται στον υπολογιστή (π.χ. USB, IDE, SATA, SCSI, SAS).
- **Κόστος:** Κόστος αγοράς (συχνά υπολογίζεται το κόστος ανά MB χωρητικότητας για σκοπούς σύγκρισης).

Στον πιο κάτω πίνακα φαίνονται διάφορες μονάδες αποθήκευσης και τα βασικά χαρακτηριστικά τους.

Κατηγορία	Ονομασία	Χωρητικότητα	Ταχύτητα Μεταφ. δεδομένων	Εγγραφή	Επανεγγραφή	Φορητός	Κόστος ανά MB
Μαγνητικά Μέσα	Δισκέττα (Floppy Disk) 	1,44 MB	50 KB/s	✓	✓	✓	€ 300
	Σκληρός Δίσκος (Hard Disk) Εσωτερικός 	160GB-4TB	150 MB/s	✓	✓		€ 0,08
	Σκληρός Δίσκος (Hard Disk) Εξωτερικός 	160GB-4TB	60 MB/s	✓	✓	✓	€ 0,10
Οπτικά Μέσα	CD-ROM	700 MB	7,8 MB/s			✓	-
	CD-R 			✓		✓	€ 0,70
	CD-RW			✓	✓	✓	€ 1,40
	DVD-ROM	4,7 GB	21 MB/s			✓	-
	DVD-R 			✓		✓	€ 0,14
	DVD-RW			✓	✓	✓	€ 0,28

Κατηγορία	Ονομασία	Χωρητικότητα	Ταχύτητα Μεταφ. δεδομένων	Εγγραφή	Επανεγγραφή	Φορητός	Κόστος ανά MB
	Blue Ray Disc 	25 GB 50 GB 100 GB 128 GB	72 MB/s	✓	✓	✓	€ 0,12
Μέσα Στερεάς Κατάστασης	USB Flash Drive 	2-128 GB	100 MB/s-500 MB/s	✓	✓	✓	€ 1,5
	SD 			✓	✓	✓	€ 1,5

8. Το λογισμικό

Ο όρος **λογισμικό** περιγράφει γενικά μια συλλογή από **προγράμματα**. Ο υπολογιστής ως συσκευή (δηλαδή το υλικό) έχει τρομερές δυνατότητες και μεγάλη ευελιξία για την επεξεργασία δεδομένων. Χωρίς όμως προγράμματα, δηλαδή σειρές από προσεκτικά τοποθετημένες **εντολές** (οδηγίες σε γλώσσα που αντιλαμβάνεται ο Η/Υ) οι οποίες να περιγράφουν στον υπολογιστή ποια ηλεκτρονικά κυκλώματα θα πρέπει να ενεργοποιηθούν και με ποια σειρά για να πραγματοποιηθεί (εκτελέσει) μια συγκεκριμένη λειτουργία (βλ. *Πληροφορική Α' & Β' Γυμνασίου, Σημειώσεις*) δεν θα είχε καμιά απολύτως χρήση. Είναι αυτός ο ευέλικτος συνδυασμός υλικού και λογισμικού ο οποίος διαφοροποιεί τον υπολογιστή από τις άλλες μηχανές και συσκευές που χρησιμοποιεί ο άνθρωπος, μετατρέποντάς τον από μια μηχανή απίστευτης πολυπλοκότητας σε ένα σχετικά εύχρηστο και ευέλικτο εργαλείο της καθημερινότητάς μας.

Βασικές Έννοιες

Ηλεκτρονικός Υπολογιστής:	Είναι μια <i>μηχανή</i> που έχει τη δυνατότητα να επεξεργάζεται , αποθηκεύει και μεταδίδει πληροφορίες με <i>μεγάλη ακρίβεια</i> και <i>ταχύτητα</i> , σύμφωνα με τις εντολές (οδηγίες) που της δίνει ο άνθρωπος μέσα από ένα πρόγραμμα .
Βασικές λειτουργίες του Υπολογιστή:	(1) <i>είσοδος δεδομένων</i> , (2) <i>επεξεργασία δεδομένων</i> , (3) <i>αποθήκευση πληροφοριών</i> (4) <i>έξοδος πληροφοριών</i> .
Πρόγραμμα (Programme):	Μια σειρά από εντολές (οδηγίες), που συντονίζουν τη λειτουργία του υπολογιστή για την εκτέλεση μιας συγκεκριμένης εργασίας.
Μονάδες μέτρησης πληροφοριών:	Η βασική μονάδα είναι το δυαδικό ψηφίο (bit), το Byte (B) το οποίο αντιστοιχεί σε 8 bit στη σειρά και τα πολλαπλάσιά του, KB, MB, GB, TB, PB (το καθένα είναι 1024 φορές μεγαλύτερο από το προηγούμενο).
Αρχιτεκτονική Η/Υ:	Ο τρόπος που σχεδιάστηκε ο Η/Υ, δηλαδή τα μέρη από τα οποία αποτελείται και πώς αυτά συνδέονται/επικοινωνούν.
Μητρική Πλακέτα (Motherboard):	Τυπωμένο κύκλωμα, στο οποίο συνδέονται και μέσω του οποίου επικοινωνούν όλες οι μονάδες του Η/Υ.
Κεντρική Μονάδα Επεξεργασίας, ΚΜΕ (Central Processing Unit, CPU):	Είναι το εξάρτημα που έχει ως βασική λειτουργία την επεξεργασία των δεδομένων, εκτελώντας τις εντολές ενός προγράμματος με τη σειρά. Τα κύρια μέρη της είναι η Αριθμητική και Λογική Μονάδα , η Μονάδα Ελέγχου και οι Καταχωρητές . Η ταχύτητα εκτέλεσης των εντολών καθορίζεται από τον χρονιστή (σε MHz GHz) και ο ρυθμός κατανάλωσης ενέργειάς της μετريέται σε Watt (W). Ονομάζεται και μικροεπεξεργαστής (microprocessor).
Μονάδες εισόδου και εξόδου:	Είναι συσκευές/εξαρτήματα τα οποία χρησιμεύουν για την είσοδο δεδομένων (π.χ. πληκτρολόγιο, ποντίκι, βιντεοκάμερα, μικρόφωνο, σαρωτής) και τη έξοδο πληροφοριών (π.χ. οθόνη, εκτυπωτής, ηχεία). Ανάλογα με το είδος τους έχουν συγκεκριμένα χαρακτηριστικά, όπως ο τύπος σύνδεσης (π.χ. USB, PS2, VGA, HDMI), η ταχύτητα μεταφοράς δεδομένων (π.χ. 1200 kbps, 12 Mbps, η ανάλυση (π.χ. 600 dpi, 1280×1024, 14 Megapixels), η ταχύτητα σάρωσης ή εκτύπωσης (π.χ. 12 ppm, 70 cps), το μέγεθος (π.χ. A4, 22"), ο λόγος διαστάσεων (π.χ. 16:9), η δυνατότητα χρώματος , ο ρυθμός ανανέωσης πλαισίων/καρέ (frame rate) (π.χ. 75 Hz, 16 fps), το κόστος αγοράς .

Κύρια Μνήμη:

Αποθηκευτικοί χώροι στους οποίους η ΚΜΕ έχει άμεσα πρόσβαση, δηλαδή η **Μνήμη Τυχαίας Προσπέλασης (Random Access Memory, RAM)**, η οποία αποτελείται από ολοκληρωμένα κυκλώματα στα οποία αποθηκεύονται προσωρινά εντολές και πληροφορίες, και η **Μνήμη Μόνο για Ανάγνωση (Read Only Memory, ROM)** η οποία είναι ολοκληρωμένο κύκλωμα ενσωματωμένο στη μητρική κάρτα με μόνιμα αποθηκευμένο περιεχόμενο που δεν χάνεται όταν διακοπεί το ηλεκτρικό ρεύμα και περιέχει το **Βασικό Σύστημα Εισόδου/Εξόδου (Basic Input/Output System, BIOS)**, που είναι σειρά από εντολές που εκτελούνται κατά την εκκίνηση του υπολογιστή.

Βοηθητική Μνήμη:

Είναι το σύνολο των περιφερειακών μονάδων αποθήκευσης, όπου οι πληροφορίες αποθηκεύονται οργανωμένα και μακροπρόθεσμα. Στη βοηθητική μνήμη ανήκουν οι **σκληροί δίσκοι** (εσωτερικοί ή εξωτερικοί), η **δισκέτα**, η **μνήμη φλας** (USB, SD, κ.λπ.), οι **οπτικοί δίσκοι** (CD, CD-R/RW, DVD, DVD-R/RW), οι μαγνητικές ταινίες και όλες οι άλλες μονάδες μόνιμης αποθήκευσης πληροφοριών. Κύρια χαρακτηριστικά τους είναι η **χωρητικότητα** (π.χ. 700 MB, 4.7 GB, 500 GB, 1 TB), η **ταχύτητα μεταφοράς δεδομένων** (π.χ. 50 kB/s, 1 Gb/s, 150 MB/s, CD στις 52x, DVD στις 16x), η **ταχύτητα περιστροφής** (π.χ. 7200 rpm), η **δυνατότητα εγγραφής/Επανεγγραφής**, η **φορητότητα**, ο **τύπος σύνδεσης** (π.χ. USB, IDE, SATA) και το **κόστος**.

Γ2.2 Το Υπολογιστικό Σύστημα ως Ενιαίο Σύνολο

Τι θα μάθουμε σήμερα:

- ❖ Να περιγράφουμε τον ρόλο του προγράμματος στον κύκλο επεξεργασίας
- ❖ Να συσχετίζουμε την επεξεργασία που πραγματοποιείται από ένα πρόγραμμα (λειτουργικό σύστημα/λογισμικό εφαρμογών) με τις φάσεις του κύκλου επεξεργασίας και με τις σχετιζόμενες μονάδες/συσκευές σε παραδείγματα
- ❖ Να περιγράφουμε τον ρόλο του χρήστη ως παρόχου/αποδέκτη δεδομένων/πληροφοριών στην πιο πάνω διαδικασία
- ❖ Να αναφέρουμε/αναγνωρίζουμε σε παραδείγματα την αλληλεξάρτηση Υλικού, Λογισμικού και Χρήστη.

1. Εισαγωγή

Για να πραγματοποιήσουμε μια οποιαδήποτε εργασία στον υπολογιστή μας, χρειάζεται να έχουμε στη διάθεσή μας κατάλληλο λογισμικό (εφαρμογή, πρόγραμμα), το οποίο ενεργοποιούμε και στη συνέχεια, αφού του δώσουμε κατάλληλα δεδομένα, το πρόγραμμα τα επεξεργάζεται ενώ παράλληλα ανακαλεί αποθηκευμένες, παράγει και αποθηκεύει τις αναμενόμενες πληροφορίες. Για παράδειγμα, έστω ότι θέλουμε να προσθέσουμε μια παράγραφο σε ένα κείμενο και να το εκτυπώσουμε. Ενεργοποιούμε κατάλληλο πρόγραμμα (επεξεργαστή κειμένου) και εισάγουμε (με το πληκτρολόγιο) το κείμενο που αντιστοιχεί στη νέα παράγραφο. Στη συνέχεια το αποθηκεύουμε και το εκτυπώνουμε (χρησιμοποιώντας το ποντίκι για να επιλέξουμε τις κατάλληλες λειτουργίες του επεξεργαστή κειμένου). Στον ίδιο υπολογιστή, εάν θέλουμε να ακούσουμε μουσική, ενεργοποιούμε κατάλληλο πρόγραμμα μέσα από το οποίο επιλέγουμε (με το ποντίκι) το αρχείο μουσικής που επιθυμούμε και το οποίο με κατάλληλη επεξεργασία των δεδομένων του μετατρέπεται σε ήχο, ο οποίος διοχετεύεται από την κάρτα ήχου στα ηχεία του υπολογιστή.

Στην πραγματικότητα, το **λογισμικό** έχει έναν ουσιαστικό ρόλο στην επεξεργασία, αφού είναι οι εντολές από το κάθε πρόγραμμα οι οποίες καθορίζουν και συντονίζουν τις λειτουργίες του **υλικού** για να έχουμε το επιθυμητό αποτέλεσμα. Στα πιο πάνω παραδείγματα εκτός από το λογισμικό και το υλικό, υπάρχει και ένας άλλος παράγοντας ο οποίος επηρεάζει την επεξεργασία: ο **χρήστης**. Είναι οι ανάγκες και οι ενέργειες του χρήστη που καθορίζουν με ποιο τρόπο θα χρησιμοποιηθεί το υλικό και το λογισμικό, ποια δεδομένα εισαγάγονται και ποιες πληροφορίες εξαγάγονται και τότε (Εικόνα 3).

2. Το Λογισμικό Εφαρμογών

Το **Λογισμικό Εφαρμογών (Application Software)** είναι τα προγράμματα που εκτελούν κάποια συγκεκριμένη εργασία η οποία βασίζεται σε κάποια ανάγκη του χρήστη (Εικόνα 3), π.χ. «να γράφει κείμενο», «να ακούει μουσική», «να αναζητά και να βλέπει πληροφορίες στον παγκόσμιο ιστό». Ο κάθε χρήστης έχει διαφορετικές ανάγκες και γι' αυτό στον υπολογιστή του έχει εγκατεστημένα διαφορετικά προγράμματα. Συχνά υπάρχουν πολλές επιλογές λογισμικού εφαρμογών κατάλληλου για την ίδια ανάγκη του χρήστη και έτσι ο χρήστης μπορεί να διαλέξει με βάση τη λειτουργικότητα, την ευκολία χρήσης και το κόστος αγοράς του κάθε προγράμματος.

Για παράδειγμα για την ανάγκη «να αναζητά και να βλέπει πληροφορίες στον παγκόσμιο ιστό», ο χρήστης έχει διάφορες επιλογές προγραμμάτων, όπως είναι ο *Internet Explorer* (από την εταιρεία Microsoft), ο *Chrome* (από την εταιρεία Google), ο *Firefox* (από τον οργανισμό

Mozilla Foundation), ο *Opera* (από την εταιρεία *Opera Software*), και ο *Safari* (από την εταιρεία *Apple*).

Ένα άλλο παράδειγμα, για την ανάγκη «να γράφει κείμενο», ο χρήστης έχει επιλογές όπως είναι το *Word* (από την εταιρεία *Microsoft*, το οποίο για να το χρησιμοποιήσει κάποιος θα πρέπει να το αγοράσει) και το *Writer* (από τον οργανισμό *The Document Foundation*, το οποίο είναι δωρεάν).

3. Το Λειτουργικό Σύστημα και άλλο Λογισμικό Συστήματος

Με τον όρο **Λογισμικό Συστήματος (System Software)** περιγράφουμε την κατηγορία των προγραμμάτων τα οποία δεν επιτελούν κάποια συγκεκριμένη εργασία που βασίζεται σε κάποια ανάγκη του χρήστη, αλλά σκοπό έχουν τον έλεγχο του υπολογιστή και την παροχή διάφορων γενικών *υπηρεσιών* στον χρήστη και σε άλλα προγράμματα (Εικόνα 3). Ο ρόλος του δηλαδή είναι περισσότερο ο έλεγχος και ο συντονισμός του υλικού και του Λογισμικού Εφαρμογών. Στο λογισμικό συστήματος περιλαμβάνονται κυρίως τα ακόλουθα:

- Το Λειτουργικό Σύστημα (Windows, Linux, Mac OS, κ.λπ.)
- Οδηγοί συσκευών (device drivers), δηλαδή λογισμικό που μας δίνουν συνήθως οι κατασκευαστές των συσκευών και το οποίο ενσωματώνεται στο λειτουργικό σύστημα, ώστε να επιτρέπει σε άλλα προγράμματα να χρησιμοποιούν τις συσκευές
- Διάφορα βοηθητικά προγράμματα (π.χ. διαγνωστικά εργαλεία που ελέγχουν και διορθώνουν προβλήματα στις Περιφερειακές Μονάδες Αποθήκευσης)
- Λογισμικό το οποίο υποστηρίζει την εκτέλεση άλλων προγραμμάτων.

Το **Λειτουργικό Σύστημα (Operating System)** είναι το κύριο μέρος του λογισμικού συστήματος (Εικόνα 3). Ο κύριος ρόλος του είναι να ελέγχει και να συντονίζει το υλικό, αλλά και να παρέχει στον χρήστη **περιβάλλοντα επικοινωνίας** με τον υπολογιστή, μηχανισμούς **διαχείρισης** (ενεργοποίηση, απενεργοποίηση) του λογισμικού εφαρμογών, και ένα

σύστημα αρχειοθέτησης για διαχείριση των πληροφοριών τις οποίες επιθυμεί να αποθηκεύσει στις μονάδες αποθήκευσης (δηλαδή στη βοηθητική μνήμη). Ο έλεγχος και ο συντονισμός του υλικού περιλαμβάνει τη **διαχείριση της ΚΜΕ** (καθορίζει ποια προγράμματα και με ποια σειρά θα εκτελούνται από την ΚΜΕ), τη **διαχείριση της μνήμης** (αποφασίζει σε ποιο χώρο της μνήμης θα αποθηκευτούν εντολές και πληροφορίες για κάθε πρόγραμμα που εκτελείται). Περιλαμβάνει επίσης τη **διαχείριση του συστήματος αρχειοθέτησης** (δηλαδή παρέχει μηχανισμούς για δημιουργία, διαγραφή, διάβασμα και εγγραφή αρχείων, ποιος χρήστης επιτρέπεται να έχει πρόσβαση στο αρχείο και πότε), τη



Εικόνα 3 Το υπολογιστικό σύστημα είναι ένα ενιαίο σύνολο και το κάθε μέρος συνεργάζεται με τα υπόλοιπα

διαχείριση της πρόσβασης των χρηστών (δηλαδή παρέχει στον κάθε χρήστη το δικό του προφίλ και κωδικούς για πρόσβαση, χώρο για αρχειοθέτηση και τη δυνατότητα να επιτρέπει ή να απαγορεύει σε άλλους χρήστες την πρόσβαση στα αρχεία του), τη **διαχείριση του δικτύου** (δηλαδή πότε και ποιος θα έχει πρόσβαση στο δίκτυο και ποιες πληροφορίες και με ποιο τρόπο θα στέλνει ή θα λαμβάνει) και τη **διαχείριση των περιφερειακών συσκευών** (δηλαδή ποιος θα έχει πρόσβαση και με ποια σειρά σε κάθε συσκευή, π.χ. έναν εκτυπωτή).

4. Ο Χρήστης

Χωρίς την παρέμβαση του **Χρήστη (User)**, το λογισμικό και το υλικό θα παρέμεναν σε αδράνεια (Εικόνα 3). Είναι ο χρήστης ο οποίος αποφασίζει ποιο πρόγραμμα θα ενεργοποιήσει κάθε στιγμή, ποια δεδομένα θα δώσει, ποιες πληροφορίες θα παραχθούν και πού θα αποσταλούν/αποθηκευτούν. Επίσης είναι οι ανάγκες του χρήστη οι οποίες καθορίζουν ποιο λογισμικό θα πρέπει να είναι εγκατεστημένο στον υπολογιστή, ακόμη ποιες περιφερειακές συσκευές θα πρέπει να είναι διαθέσιμες και ποια θα πρέπει να είναι τα χαρακτηριστικά του υλικού.

5. Το Υπολογιστικό Σύστημα

Ως συμπέρασμα, ερμηνεύοντας την Εικόνα 3, θα μπορούσαμε να πούμε ότι ο χρήστης καθορίζει τις προδιαγραφές του λογισμικού και του υλικού και αλληλεπιδρά τόσο με το λειτουργικό σύστημα, όσο και με το λογισμικό εφαρμογών, παρέχοντας δεδομένα και λαμβάνοντας πληροφορίες. Το λογισμικό εφαρμογών συνεργάζεται με το λειτουργικό σύστημα για να εξυπηρετήσει συγκεκριμένες ανάγκες του χρήστη. Το λειτουργικό σύστημα (και το υπόλοιπο λογισμικό συστήματος) παρέχει στο χρήστη βασικές δυνατότητες διαχείρισης ενώ ταυτόχρονα ελέγχει και συντονίζει το υλικό και υποστηρίζει το λογισμικό εφαρμογών.

Αυτή η αλληλένδετη τετράδα (χρήστης, λογισμικό εφαρμογών, λογισμικό συστήματος, υλικό) συχνά αναφέρεται ως το **Υπολογιστικό Σύστημα**, το οποίο για να υπάρχει είναι απαραίτητο να υπάρχουν όλα τα μέρη του και να συνεργάζονται αρμονικά.

Εδώ πρέπει να διευκρινιστεί τι εννοούμε όταν λέμε να «συνεργάζονται αρμονικά» τα διάφορα μέρη του υπολογιστικού συστήματος. Όπως έχουμε δει σε προηγούμενες τάξεις, αλλά και σε προηγούμενα μαθήματα, υπάρχουν στην αγορά πολλά διαφορετικά μοντέλα για κάθε είδος υλικού (π.χ. ΚΜΕ Intel Core 2 Quad, ARM 9, διάφορες οθόνες, εκτυπωτές, σαρωτές, κ.λπ.). Υπάρχουν, επίσης, διάφορα λειτουργικά συστήματα (π.χ. Windows 7, Windows 8, Android, iOS, Mac OS X, Linux, κ.λπ.) και πληθώρα από αντίστοιχες εφαρμογές (Λογισμικό εφαρμογών). Εάν πάρουμε έναν υπολογιστή τύπου ταμπλέτας της εταιρείας Apple (π.χ. Apple iPad 2), ο υπολογιστής αυτός συμπεριλαμβάνει το λειτουργικό σύστημα iOS. Εάν επιθυμούσαμε να αντικαταστήσουμε το λειτουργικό σύστημα με Windows 7, δεν θα ήταν δυνατόν, διότι το υλικό του υπολογιστή δεν υποστηρίζεται από τα Windows 7 (τα οποία είναι φτιαγμένα για προσωπικούς υπολογιστές με αρχιτεκτονική x86, δηλαδή ΚΜΕ όπως την Intel Core 2 Quad). Με αντίστοιχο τρόπο, δεν θα ήταν δυνατόν να εγκαταστήσουμε στον Apple iPad 2 το σχεδιαστικό πρόγραμμα AutoCAD (λογισμικό εφαρμογών), το οποίο είναι φτιαγμένο για το λειτουργικό σύστημα Windows, διότι ο τρόπος συνεργασίας είναι διαφορετικός στο iOS από ότι στο Windows. Με αντίστοιχη λογική, εάν ο χρήστης (π.χ. ένας συγγραφέας) επιθυμεί να έχει υπολογιστή για τη συγγραφή κειμένου, ο συγκεκριμένος υπολογιστής (Apple iPad 2) ίσως να μην είναι η πιο κατάλληλη επιλογή, διότι η απουσία πληκτρολογίου θα δυσχέραινε τη

συγγραφή (το πληκτρολόγιο στην οθόνη αφής δεν είναι τόσο εύχρηστο όσο ένα κανονικό πληκτρολόγιο).

Βασικές Έννοιες

Λογισμικό Συστήματος (System Software):

Κατηγορία λογισμικού (προγραμμάτων) τα οποία δεν επιτελούν κάποια συγκεκριμένη εργασία που βασίζεται σε κάποια ανάγκη του χρήστη, αλλά σκοπό έχουν τον έλεγχο του υπολογιστή και την παροχή διάφορων γενικών υπηρεσιών. Το κύριο μέρος του αποτελεί το Λειτουργικό Σύστημα. Συμπεριλαμβάνει, επίσης, τους οδηγούς συσκευών (device drivers), διάφορα βοηθητικά/διαγνωστικά προγράμματα για συντήρηση του υπολογιστή και λογισμικό για υποστήριξη άλλων προγραμμάτων.

Λειτουργικό Σύστημα (Operating System):

Το κύριο μέρος του Λογισμικού Συστήματος. Ο κύριος ρόλος του είναι να ελέγχει και να συντονίζει το υλικό (διαχείριση της ΚΜΕ, της μνήμης, του συστήματος αρχειοθέτησης, της πρόσβασης των χρηστών, του δικτύου και των περιφερειακών συσκευών. Επίσης, παρέχει στον χρήστη περιβάλλοντα επικοινωνίας με τον υπολογιστή, μηχανισμούς διαχείρισης του λογισμικού εφαρμογών και ένα σύστημα αρχειοθέτησης.

Λογισμικό Εφαρμογών (Application Software):

Κατηγορία λογισμικού (προγραμμάτων) τα οποία εκτελούν κάποια συγκεκριμένη εργασία η οποία βασίζεται σε κάποια ανάγκη του χρήστη.

Χρήστης (User):

Ο άνθρωπος, ο οποίος αποφασίζει ποιο πρόγραμμα θα ενεργοποιήσει κάθε στιγμή, ποια δεδομένα θα δώσει, ποιες πληροφορίες θα παραχθούν και πού θα αποσταλούν/αποθηκευτούν. Οι ανάγκες του καθορίζουν τις προδιαγραφές του λογισμικού και του υλικού.

Υπολογιστικό Σύστημα:

Η αλληλένδετη τετράδα **Χρήστης–Λογισμικό Εφαρμογών–Λογισμικό Συστήματος–Υλικό**, τα οποία είναι απαραίτητα να υπάρχουν και να συνεργάζονται αρμονικά. Ο χρήστης καθορίζει τις προδιαγραφές του λογισμικού και του υλικού και αλληλεπιδρά τόσο με το λειτουργικό σύστημα, όσο και με το λογισμικό εφαρμογών, παρέχοντας δεδομένα και λαμβάνοντας πληροφορίες. Το λογισμικό εφαρμογών συνεργάζεται με το λειτουργικό σύστημα για να εξυπηρετήσει συγκεκριμένες ανάγκες του χρήστη. Το Λογισμικό Συστήματος (κυρίως το λειτουργικό σύστημα) παρέχει στο χρήστη βασικές δυνατότητες διαχείρισης, ενώ ταυτόχρονα ελέγχει και συντονίζει το υλικό και υποστηρίζει το λογισμικό εφαρμογών.

Πηγές

1. Βουτηράς Γ., Κονιδάρη Ε., Κούτρας Μ., Σφώρος Ν., (2003) *Πληροφορική Γυμνασίου*, σελ. 61—69, ΟΕΔΒ.

Γ2.3

Προβλήματα του Υπολογιστικού Συστήματος και η Αντιμετώπισή τους

Τι θα μάθουμε σήμερα:

- ❖ Να αναφέρουμε συνήθη προβλήματα που σχετίζονται με τη δυσλειτουργία του υλικού και να τα αναγνωρίζουμε σε παραδείγματα
- ❖ Να αναφέρουμε τρόπους προστασίας/αντιμετώπισης των προβλημάτων που σχετίζονται με το υλικό και να προτείνουμε τους κατάλληλους σε παραδείγματα
- ❖ Να αναφέρουμε συνήθη προβλήματα που σχετίζονται με τη δυσλειτουργία του λογισμικού και τη φύλαξη δεδομένων και να τα αναγνωρίζουμε σε παραδείγματα
- ❖ Να αναφέρουμε τρόπους προστασίας/αντιμετώπισης των προβλημάτων που σχετίζονται με το λογισμικό και να προτείνουμε τους κατάλληλους σε παραδείγματα
- ❖ Να αναφέρουμε συνήθη προβλήματα που παρουσιάζονται στο χρήστη από τη χρήση του υπολογιστικού συστήματος και να τα αναγνωρίζουμε σε παραδείγματα
- ❖ Να αναφέρουμε τρόπους προστασίας/αντιμετώπισης των προβλημάτων που σχετίζονται με τον χρήστη και να προτείνουμε τους κατάλληλους σε παραδείγματα.

Κατά τη διάρκεια της ζωής ενός υπολογιστικού συστήματος, η λειτουργική σχέση μεταξύ υλικού λογισμικού και χρήστη μπορεί να κλονιστεί και να τεθεί σε αμφισβήτηση. Συνήθως, τα διάφορα προβλήματα που ενδέχεται να παρουσιαστούν, θα αφορούν είτε το υλικό είτε το λογισμικό του συστήματος ή ακόμη και τον χρήστη του. Παρακάτω, παρουσιάζονται κατηγοριοποιημένα τα συνήθη προβλήματα που σχετίζονται με τη δυσλειτουργία του υλικού, του λογισμικού ή με τον χρήστη, καθώς επίσης και οι προτεινόμενοι τρόποι αντιμετώπισής τους.

1. Προβλήματα που Σχετίζονται με το Υλικό

1.1 Περιβάλλον όπου βρίσκεται το Υπολογιστικό Σύστημα (θερμοκρασία, σκόνη και υγρασία)

Οι επιμέρους μονάδες του ηλεκτρονικού μας υπολογιστή όπως η ΚΜΕ και ο σκληρός δίσκος είναι κατασκευασμένες για να αποδίδουν καλύτερα σε συγκεκριμένες κλιματικές συνθήκες. Επειδή η θερμοκρασία και η υγρασία είναι δύο από τους παράγοντες που μπορεί να επηρεάσουν δυσμενώς τη λειτουργία του υπολογιστικού μας συστήματος, θα πρέπει να ανατρέχουμε και να πληροφορούμαστε από το εγχειρίδιο χρήσης του κατασκευαστή τις απαραίτητες πληροφορίες, αφού για κάθε διαφορετική συσκευή, αυτές διαφέρουν.

Κάποιες γενικές συμβουλές που πρέπει να ακολουθούμε για να αποφεύγουμε προβλήματα που μπορεί να προκύψουν στο υλικό του υπολογιστή και πηγάζουν από το περιβάλλον στο οποίο είναι τοποθετημένος είναι οι ακόλουθες:

- Να συμβουλευόμαστε πάντοτε τις πληροφορίες οι οποίες αφορούν στις ιδανικές κλιματικές συνθήκες χρήσης του υπολογιστικού σας συστήματος.
- Να μη φράζουμε τις οπές εξαερισμού του υπολογιστή μας και να μην τον τοποθετούμε επάνω σε πορώδεις επιφάνειες, όπως χαλιά, κουβέρτες, καναπέδες, κρεβάτια ή παρόμοια υλικά όπως κουρτίνες. Οι οπές εξαερισμού του υπολογιστικού μας συστήματος θα πρέπει να επιτρέπουν την απρόσκοπτη είσοδο κρύου αέρα και έξοδο του ζεστού, η οποία είναι απαραίτητη για τη μείωση της θερμότητας.
- Να αποφεύγουμε να μεταφέρουμε τον υπολογιστή μας απευθείας από μια κρύα τοποθεσία σε μια ζεστή διότι ενδέχεται να συσσωρευτεί υγρασία στο εσωτερικό του.
- Να αποφεύγουμε να τοποθετούμε τον υπολογιστή μας κοντά σε πηγές θερμότητας (σώματα θέρμανσης) ή πηγές μαγνητισμού (μεγάλα ηχεία στερεοφωνικών συστημάτων).

- Να μην τοποθετούμε κοντά στον υπολογιστή μας ποτά και φαγητά.
- Να μην τοποθετούμε τον ηλεκτρονικό μας υπολογιστή κοντά σε ανοικτά παράθυρα ή γενικότερα σε χώρους οι οποίοι παρουσιάζουν αυξημένα επίπεδα σκόνης.
- Για τον καθαρισμό του υπολογιστικού μας συστήματος να μη χρησιμοποιούμε νερό αλλά μόνο ειδικά καθαριστικά.

1.2 Διακοπή παροχής ηλεκτρικού ρεύματος

Η απότομη διακοπή του ρεύματος ή ακόμα και η διακύμανση στην ισχύ και στη συχνότητα του ρεύματος, φαινόμενα που δυστυχώς είναι ιδιαίτερα συχνά στη χώρα μας, πιθανό να προκαλέσουν σημαντικά προβλήματα στη λειτουργία του υπολογιστικού μας συστήματος. Ανάλογα με την περίπτωση, τα προβλήματα μπορεί να περιοριστούν σε βλάβη του τροφοδοτικού αλλά μπορεί να επεκταθούν και σε άλλες συσκευές του υπολογιστικού μας συστήματος όπως είναι ο σκληρός δίσκος, η μητρική κάρτα και η κεντρική μονάδα επεξεργασίας.

Για να προστατέψουμε το υπολογιστικό μας σύστημα από αυτό το πρόβλημα μπορούμε να χρησιμοποιήσουμε μια συσκευή **UPS** (Uninterruptible Power Supply / Αδιάλειπτη Παροχή Ενέργειας). Το UPS εξυπηρετεί δύο βασικούς σκοπούς: (1) την παροχή ηλεκτρικής ενέργειας για ένα χρονικό διάστημα, δίνοντας έτσι την ευκαιρία στον χρήστη να τερματίσει με ασφάλεια την λειτουργία του υπολογιστή και των εφαρμογών που χρησιμοποιεί τη δεδομένη στιγμή και (2) ασφαλίζει τις συσκευές που είναι συνδεδεμένες από υπερτάσεις ή χαμηλές τάσεις, φιλτράροντας το ρεύμα, έτσι ώστε να έχει τη σωστή συχνότητα (50 Hz - 60 Hz).



Εικόνα 4 Uninterruptable Power Supply (UPS)

Τα UPS διαθέτουν μπαταρίες οι οποίες μπορούν να τροφοδοτήσουν, για κάποιο χρονικό διάστημα, τον υπολογιστή μας με ρεύμα. Επιπλέον, διαθέτουν φίλτρα σταθεροποίησης και συνεπώς προστατεύουν και από διακυμάνσεις της τάσης εισόδου. Πολλά UPS παρέχουν προστασία και από τις υπερτάσεις των τηλεφωνικών γραμμών που μπορεί να προκληθούν από την πτώση ενός κεραυνού. Τα UPS διαθέτουν εξόδους τροφοδοσίας γνωστές ως IEC, οι οποίες χρησιμοποιούνται από όλους τους υπολογιστές. Τα περισσότερα UPS έχουν ενδείξεις οι οποίες ενημερώνουν τον χρήστη για την κατάστασή τους, το πόσο φορτισμένη είναι η μπαταρία τους και άλλα.



Εικόνα 5 Έξοδοι Τροφοδοσίας IEC

1.3 Φυσική φθορά

Παρόλο που προβλήματα λόγω φυσικής φθοράς μπορεί να παρουσιαστούν σε διάφορες συσκευές του υπολογιστικού μας συστήματος, ο σκληρός δίσκος θεωρείται ως μία από τις συσκευές η οποία μπορεί να παρουσιάσει πιο συχνά από άλλες τέτοια προβλήματα. Ο σκληρός δίσκος είναι μία ευαίσθητη συσκευή, η οποία αποτελείται από διάφορα μηχανικά και ηλεκτρονικά μέρη. Για όση ώρα ο υπολογιστής μας είναι ανοικτός, ο σκληρός δίσκος περιστρέφεται με πολύ υψηλές ταχύτητες, με αποτέλεσμα τη



Εικόνα 6 Αντίγραφα Ασφαλείας

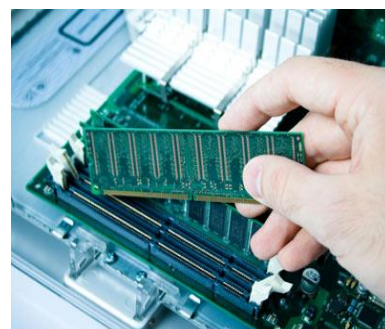
φθορά των υλικών του όσο περνάει ο χρόνος και από άλλες αιτίες (π.χ. μια απότομη πτώση ή ταρακούνημα μπορεί να προκαλέσει ανεπανόρθωτη βλάβη).

Γενικά, εάν μία συσκευή του υπολογιστικού μας συστήματος παρουσιάσει πρόβλημα λόγω φυσικής φθοράς μπορούμε να την αντικαταστήσουμε. Ειδικά για τις μονάδες αποθήκευσης όπως είναι ο σκληρός δίσκος, έχοντας πάντα υπόψη το ενδεχόμενο της απώλειας σημαντικών δεδομένων σε περίπτωση βλάβης από φυσική φθορά, είναι πολύ σημαντικό να έχουμε πάντα ένα ενημερωμένο αντίγραφο ασφαλείας των αρχείων (backup file) μας σε κάποια άλλη συσκευή αποθήκευσης, κατά προτίμηση εξωτερική, έτσι ώστε όταν επέλθει το μοιραίο να μη χάσουμε τα δεδομένα μας. Επιπρόσθετα, είναι καλό να χρησιμοποιήσουμε κατά διαστήματα το chkdsk (chkdsk.exe), ένα από τα βοηθητικά προγράμματα του λογισμικού συστήματος (εγκαθίσταται μαζί με το λειτουργικό μας σύστημα). Το εργαλείο chkdsk ελέγχει τον σκληρό δίσκο μας για τυχόν προβλήματα και στη συνέχεια προσπαθεί να επιδιορθώσει τα προβλήματα που έχει εντοπίζει.

1.4 Εμφάνιση νέων υπολογιστικών αναγκών

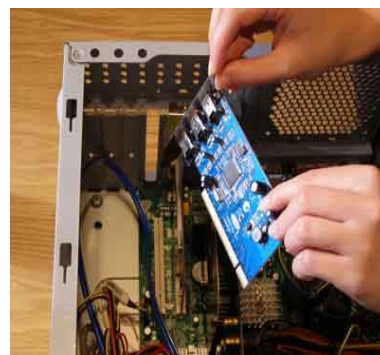
Κατά τη διάρκεια της ζωής του υπολογιστικού μας συστήματος, οι υπολογιστικές μας ανάγκες πιθανόν να μεταβληθούν, με αποτέλεσμα να είναι αναγκαία είτε η **αναβάθμιση** του συστήματός με την προσθήκη ή την αντικατάσταση κάποιων συσκευών από τις οποίες αποτελείται. Συνήθως, οι δυνατότητες αναβάθμισης ενός υπολογιστικού συστήματος εξαρτώνται από την αρχιτεκτονική του συγκεκριμένου συστήματος.

Για παράδειγμα, εάν αγοράσαμε έναν υπολογιστή ο οποίος διαθέτει 4 GB RAM, η δυνατότητα αναβάθμισης της κύριας μνήμης του εξαρτάται από (1) το μέγεθος της υποστηριζόμενης μνήμης, (2) το μέγεθος εγκατεστημένης μνήμης και (3) τον αριθμό υποδοχών μνήμης που διαθέτει η μητρική κάρτα του υπολογιστή μας. Κατά συνέπεια, για να αποφασίσουμε εάν είναι δυνατή η αναβάθμιση της μνήμης του υπολογιστή μας θα πρέπει να αφαιρέσουμε από το μέγεθος της υποστηριζόμενης μνήμης το μέγεθος της εγκατεστημένης μνήμης και στη συνέχεια να εξετάσουμε τις επιλογές μας, έχοντας υπόψη και τον αριθμό των υποδοχών μνήμης που διαθέτει η μητρική κάρτα του υπολογιστή μας.



Εικόνα 7 Αναβάθμιση Μνήμης RAM

Επιπρόσθετα, θα μπορούσαμε να εμπλουτίσουμε ή να αναβαθμίσουμε τις δυνατότητες του υπολογιστικού μας συστήματος με νέες, προσθέτοντας σε αυτό καινούργιες συσκευές ή αντικαθιστώντας τις παλιές συσκευές με νέες. Η αναβάθμιση του υπολογιστικού μας συστήματος ή η αντικατάσταση των συσκευών του μπορεί να αφορά είτε περιφερειακές συσκευές, όπως είναι οι σαρωτές και οι εκτυπωτές, είτε κάρτες επέκτασης, όπως είναι η κάρτα οθόνης (κάρτα γραφικών), η κάρτα ήχου, η κάρτα δικτύου, η κάρτα ραδιοφώνου, η κάρτα τηλεόρασης και η κάρτα διαμορφωτή/αποδιαμορφωτή (modem). Τόσο η ανάπτυξη των υπηρεσιών που προσφέρονται μέσω του διαδικτύου όσο και η τάση των κατασκευαστών να ενσωματώνουν τις εσωτερικές κάρτες επέκτασης στη μητρική πλακέτα, έχουν



Εικόνα 8 Υποδοχές Επέκτασης / Κάρτα Επέκτασης

περιορίσει αισθητά την ανάγκη εγκατάστασης νέων καρτών επέκτασης στο υπολογιστικό μας σύστημα.

1.5 Κλοπή

Η μείωση στο μέγεθος των υπολογιστικών συστημάτων καθώς και η δραματική αύξηση των χρηστών, τα έχουν μετατρέψει σε ελκυστική λεία για επίδοξους κλέφτες. Παρακάτω, δίνονται μία σειρά από εισηγήσεις οι οποίες ελαχιστοποιούν την πιθανότητα να μετατραπούμε σε θύματα κλοπής του υπολογιστικού μας συστήματος και των περιφερειακών του μονάδων.

- Να καταγράφουμε τον αριθμό σειράς (Serial Number) της συσκευής μας. Ταυτόχρονα, να καθορίσουμε/δημιουργήσουμε ένα δικό σας σημάδι κάπου πάνω στη συσκευή μας. Με αυτό τον τρόπο γίνεται πιο εύκολη η αναγνώρισή της σε περίπτωση κλοπής.
- Στο σπίτι, να μην αφήνουμε τον υπολογιστή μας κοντά σε ανοικτά παράθυρα ή πάνω σε τραπέζια στην αυλή, κοντά στο δρόμο.
- Σε χώρους όπου επικρατεί συνωστισμός, όπως είναι τα μέσα μαζικής μεταφοράς, να διατηρούμε τον φορητό μας υπολογιστή αθέατο τοποθετώντας τον στην τσάντα του.
- Όταν βρισκόμαστε σε χώρους διασκέδασης όπως είναι η καφετέρια, να μην αφήνουμε τον υπολογιστή μας χωρίς επίβλεψη.
- Να μην αφήνουμε τον υπολογιστή μας στο αυτοκίνητο. Σε περίπτωση που πρέπει να τον αφήσουμε μέσα στο αυτοκίνητο, να τον τοποθετούμε σε σημείο που δεν θα είναι ορατός στους περαστικούς.



Εικόνα 9 Κλοπή Υπολογιστή

2. Προβλήματα που Σχετίζονται με το Λογισμικό

2.1 Απώλεια δεδομένων

Όπως γνωρίζουμε, για τη μόνιμη αποθήκευση των δεδομένων μας χρησιμοποιούμε την περιφερειακή μνήμη. Συνήθως, στον σκληρό δίσκο του υπολογιστή μας, βρίσκεται αποθηκευμένο το λειτουργικό σύστημα το οποίο είναι απαραίτητο για τη λειτουργία του υπολογιστή μας. Επιπρόσθετα, τόσο στον σκληρό δίσκο όσο και σε άλλες μονάδες περιφερειακής μνήμης όπως είναι η μνήμη φλας, ο εξωτερικός δίσκος και οι διάφοροι τύποι οπτικών δίσκων αποθηκεύουμε όλα μας τα αρχεία και προγράμματα. Σε κάποιες περιπτώσεις, η απώλεια αυτών των αρχείων και των προγραμμάτων μπορεί να είναι καταστροφική.

Για να προστατεύσουμε τα δεδομένα μας, καταρχάς, πρέπει να χειριζόμαστε την κάθε συσκευή αποθήκευσης με προσοχή και να ακολουθούμε πιστά της οδηγίες χρήσης της. Για παράδειγμα, οι διάφοροι τύποι οπτικών δίσκων θα πρέπει να αποθηκεύονται στις θήκες αποθήκευσης για να προστατεύονται από τη σκόνη και τα γδαρσίματα της επιφάνειάς τους. Επιπρόσθετα, η αφαίρεση της μνήμης φλας από τη θύρα USB του υπολογιστικού μας συστήματος χωρίς να ακολουθείται η κανονική διαδικασία αποσύνδεσης είναι πιθανόν να προκαλέσει απώλεια δεδομένων. Τέλος, έχοντας υπόψη ότι οι μονάδες αποθήκευσης είναι συσκευές οι οποίες σε ανύποπτο χρόνο και για διάφορους λόγους είναι πιθανόν να σταματήσουν να λειτουργούν, θα πρέπει να δημιουργούμε



Εικόνα 10 Μνήμη φλάς σε μη λειτουργήσιμη κατάσταση

πάντοτε εφεδρικά αρχεία ασφαλείας, τουλάχιστον για τα δεδομένα που θεωρούμε σημαντικά. Τα αντίγραφα ασφαλείας μπορούμε να τα αποθηκεύσουμε παράλληλα είτε σε κάποιες άλλες μονάδες περιφερειακής μνήμης είτε σε διάφορους ιστότοπους στο διαδίκτυο όπως είναι το Dropbox και το Google Drive. Μπορεί η διαδικασία δημιουργίας αντιγράφων ασφαλείας να είναι χρονοβόρα αλλά πιθανόν να αποβεί σωτήρια σε περίπτωση που κάποια μονάδα αποθήκευσης καταστραφεί.

2.2 Κακόβουλα προγράμματα

Ο όρος κακόβουλο πρόγραμμα καλύπτει όλων των ειδών το κακόβουλο λογισμικό που έχει σχεδιαστεί με σκοπό να βλάψει κάποιον υπολογιστή ή κάποιο δίκτυο. Συγκεκριμένα, στα κακόβουλα προγράμματα συμπεριλαμβάνονται διάφορες κατηγορίες όπως είναι και οι εξής: Ιοί (Viruses), Δούρειοι Ίπποι (Trojan Horses) και Σκουλήκια (Worms). Κάποια από τα βασικότερα χαρακτηριστικά των κακόβουλων προγραμμάτων είναι η δυνατότητα τους να «κρύβονται», να αναπαράγουν τον εαυτό τους και να πολλαπλασιάζονται, να μετακινούνται από έναν υπολογιστή σε άλλο, να επιτρέπουν την πρόσβαση τρίτων στο υπολογιστικό μας σύστημα, να προκαλούν καταστροφές σε αρχεία και προγράμματα, να αντιγράφουν δεδομένα από έναν υπολογιστή σε άλλον και να απασχολούν όλο και περισσότερους πόρους του συστήματος με αποτέλεσμα να κάνουν όλο και πιο αργό τον υπολογιστή μας. Οι τρόποι με τους οποίους μπορούν να μεταδοθούν τα κακόβουλα προγράμματα είναι πολλοί. Πιο συχνά μεταδίδονται κατά τη μεταφορά αρχείων μέσω των μονάδων αποθήκευσης από έναν υπολογιστή σε άλλον, μέσω του ηλεκτρονικού ταχυδρομείου συνήθως με τη μορφή επισυναπτόμενων αρχείων και μέσω ενός δικτύου ή ακόμα και του Διαδικτύου (π.χ. από «μολυσμένα» αρχεία που κατεβάσαμε).



Εικόνα 11 Κατηγορίες Κακόβουλων Προγραμμάτων (Δούρειοι Ίπποι, Ιοί και Σκουλήκια)

Με στόχο την προστασία των δεδομένων μας από κακόβουλα προγράμματα θα πρέπει να ακολουθούμε τις παρακάτω οδηγίες:

- Να ενημερώνουμε το λειτουργικό μας σύστημα σε τακτά χρονικά διαστήματα, ώστε να αντιμετωπίζονται αδυναμίες ασφαλείας τις οποίες εντόπισε ο κατασκευαστής του.
- Να εγκαταστήσουμε στο υπολογιστικό μας σύστημα ένα πρόγραμμα αντιβίωσης (antivirus) με στόχο την αποτελεσματική αντιμετώπιση των κακόβουλων προγραμμάτων. Συνήθως, τα αντιβιοτικά προγράμματα, αφού εντοπίσουν ένα κακόβουλο πρόγραμμα, δίνουν στον χρήστη τις εξής επιλογές: (1) να καθαρίσει τα «μολυσμένα αρχεία», (2) να τα διαγράψει ή (3) να τα θέσει σε καραντίνα.
- Να ενημερώνουμε συχνά το αντιβιοτικό μας πρόγραμμα για να είναι σε θέση να αντιμετωπίζει τα νέα κακόβουλα προγράμματα.
- Να έχουμε ενεργοποιημένο και ρυθμισμένο ανάλογα το τείχος προστασίας (firewall) του ηλεκτρονικού μας υπολογιστή, ώστε να εντοπίζει και να ανακόπτει πιθανές προσπάθειες των κακόβουλων προγραμμάτων να μεταδοθούν ή να μεταφέρουν δεδομένα μέσω του δικτύου ή του Διαδικτύου.



Εικόνα 12 Προστατευμένος Υπολογιστής

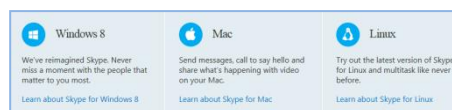
- Να είμαστε προσεκτικοί όταν ανοίγουμε αρχεία τα οποία επισυνάπτονται σε ηλεκτρονικά μηνύματα ή όταν κάνουμε λήψεις αρχείων από διάφορους ιστότοπους. Παράλληλα, να βεβαιωνόμαστε ότι το αντιβιοτικό μας πρόγραμμα είναι ρυθμισμένο με τρόπο που να ελέγχει αυτά τα αρχεία.
- Πριν αντιγράψουμε κάποιο αρχείο από μία μονάδα αποθήκευσης να βεβαιωνόμαστε ότι έχουμε ελέγξει τη μονάδα αποθήκευσης σαρώνοντάς την με το αντιβιοτικό μας πρόγραμμα.

2.3 Προβλήματα εγκατάστασης και συμβατότητας

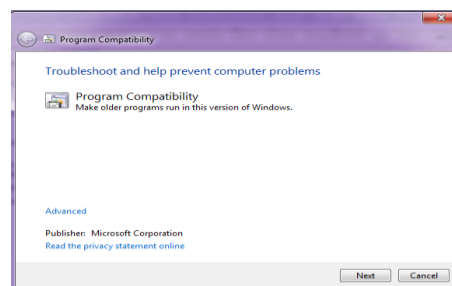
Όταν εγκαθιστούμε νέα προγράμματα στο υπολογιστικό μας σύστημα, είναι σημαντικό να βεβαιωνόμαστε ότι είναι συμβατά με το λειτουργικό σύστημα του υπολογιστή μας. Συνήθως, το κάθε πρόγραμμα είναι φτιαγμένο για να συνεργάζεται με ένα συγκεκριμένο λειτουργικό σύστημα. Κατά συνέπεια, μια εφαρμογή η οποία λειτουργεί σε έναν υπολογιστή ο οποίος διαθέτει ένα συγκεκριμένο λειτουργικό σύστημα, δεν μπορεί να λειτουργήσει σε έναν άλλο υπολογιστή ο οποίος διαθέτει κάποιο άλλο λειτουργικό σύστημα. Επειδή οι κατασκευαστές λογισμικού γνωρίζουν αυτόν τον περιορισμό, συνήθως ετοιμάζουν διαφορετικές εκδόσεις μιας εφαρμογής για διαφορετικά λειτουργικά συστήματα, ώστε εμείς να επιλέγουμε αυτήν που ταιριάζει με το λειτουργικό σύστημα που διαθέτει ο υπολογιστής μας. Επιπρόσθετα, εάν για οποιονδήποτε λόγο προχωρήσουμε στην αναβάθμιση του λειτουργικού μας συστήματος σε μια νέα έκδοση, είναι δυνατόν να αντιμετωπίσουμε κάποια προβλήματα με προγράμματα τα οποία δεν είναι συμβατά με τη νέα έκδοση του λειτουργικού συστήματος.

Κατά συνέπεια, πριν εγκαταστήσουμε μία νέα εφαρμογή στον υπολογιστή μας θα πρέπει να βεβαιωνόμαστε ότι η συγκεκριμένη έκδοση της εφαρμογής είναι συμβατή με το λειτουργικό σύστημα του υπολογιστή μας. Επιπρόσθετα, οι τελευταίες εκδόσεις των Windows μας δίνουν τη δυνατότητα να αντιμετωπίσουμε κάποια προβλήματα συμβατότητας τα οποία πιθανόν να προέκυψαν μετά από αναβάθμιση του λειτουργικού μας συστήματος. Συγκεκριμένα, για να ενεργοποιήσουμε τη συγκεκριμένη δυνατότητα θα πρέπει να ακολουθήσουμε τα εξής βήματα:

1. Κλικ στο **Start** από τη γραμμή εργασιών.
2. Κλικ στο **Control Panel**.
3. Από την κατηγορία **Programs and Features** κάνουμε κλικ στο **Run programs made for previous versions of Windows**.
4. Κλικ στο **Next** στο πλαίσιο διαλόγου με τίτλο **Program Compatibility**.
5. Από τη λίστα των προγραμμάτων που εμφανίζεται μπροστά μας, επιλέγουμε το πρόγραμμα με το οποίο αντιμετωπίζουμε το πρόβλημα και μέσα από μία σειρά βημάτων επιχειρείται η επίλυσή του από το ίδιο το λειτουργικό σύστημα.



Εικόνα 13 Οθόνη επιλογής οδηγού εγκατάστασης της εφαρμογής Skype, ανάλογα με το λειτουργικό σύστημα που διαθέτει η συσκευή



Εικόνα 14 Πλαίσιο διαλόγου του Program Compatibility

2.4 Εμφάνιση νέων υπολογιστικών αναγκών

Όπως και στην περίπτωση του υλικού έτσι και στην περίπτωση του λογισμικού, είναι πιθανόν κατά τη διάρκεια της ζωής του υπολογιστικού μας συστήματος, οι υπολογιστικές μας ανάγκες να μεταβληθούν. Κατά συνέπεια, πιθανόν να είναι αναγκαία είτε η αναβάθμιση των υφιστάμενων προγραμμάτων στις νέες εκδόσεις τους είτε η εγκατάσταση νέων προγραμμάτων, τα οποία ικανοποιούν τις νέες υπολογιστικές μας ανάγκες.

Κατά την αναβάθμιση ή την εγκατάσταση νέων προγραμμάτων θα πρέπει να έχουμε υπόψη και τα ακόλουθα:

- Να ελέγχουμε κατά πόσο ο υπολογιστής μας διαθέτει τις απαραίτητες τεχνικές προδιαγραφές οι οποίες απαιτούνται από το πρόγραμμα.
- Να βεβαιωνόμαστε ότι η συγκεκριμένη έκδοση της εφαρμογής είναι συμβατή με το λειτουργικό σύστημα του υπολογιστή μας.
- Εάν θα συνδέσουμε μια νέα περιφερειακή συσκευή στο υπολογιστικό μας σύστημα πιθανόν να μας ζητηθεί να εγκαταστήσουμε ταυτόχρονα και τους απαραίτητους οδηγούς (drivers) οι οποίοι κάνουν δυνατή την επικοινωνία της συσκευής με το λειτουργικό σύστημα. Σε αρκετές περιπτώσεις ο εντοπισμός και η εγκατάσταση αυτών των προγραμμάτων γίνεται αυτόματα από το λειτουργικό σύστημα.

2.5 Σφάλματα χρήστη

Συχνά, κάποια από τα προβλήματα που είναι δυνατόν να αντιμετωπίσουμε και αφορούν το λογισμικό του υπολογιστή μας, πηγάζουν από σφάλματα του ίδιου του χρήστη. Ένα από τα συχνότερα σφάλματα των χρηστών είναι η κατά παράληψη διαγραφή αρχείων ή η αποθήκευση διαφορετικού αρχείου με το όνομα ενός υφιστάμενου αρχείου στον ίδιο φάκελο, με αποτέλεσμα να χαθεί το αρχικό περιεχόμενο του αρχείου.

Σε κάποιες περιπτώσεις είναι δυνατή η επαναφορά ενός αρχείου χρησιμοποιώντας την εντολή Επαναφορά Στοιχείου (Restore this item) μέσω του Κάδου Ανακύκλωσης (Recycle Bin) του υπολογιστή μας. Στις περιπτώσεις που δεν μπορούμε να εντοπίσουμε ένα διαγραμμένο αρχείο στον κάδο ανακύκλωσης του υπολογιστή μας, για να το επαναφέρουμε μπορούμε να χρησιμοποιήσουμε εξειδικευμένα προγράμματα τα οποία κάνουν «Αποδιαγραφή» (Undelete) αυτών των αρχείων.

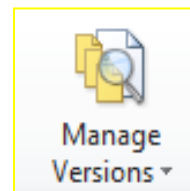


Εικόνα 15 Κάδος Ανακύκλωσης

Αυτού του τύπου το λογισμικό βασίζεται στο γεγονός ότι όταν διαγράφουμε ένα αρχείο, διαγράφεται η αναφορά στο αρχείο, αλλά τα δεδομένα εξακολουθούν να υπάρχουν στη μονάδα αποθήκευσης, μέχρι να χρειαστεί ο χώρος για την αποθήκευση κάποιου άλλου αρχείου. Το λογισμικό «Αποδιαγραφής» πλοηγείται στους δίσκους του συστήματός μας αναζητώντας αυτά τα χαμένα δεδομένα, με σκοπό να τα ανακτήσει γρήγορα και αποτελεσματικά. Αυτό μπορεί να το κάνει τόσο στους σκληρούς δίσκους του υπολογιστή μας όσο και σε μνήμες φλας, κάρτες μνήμης και εξωτερικούς δίσκους. Συνήθως, στην αρχή εντοπίζονται τα αρχεία που διαγράφηκαν και κατηγοριοποιούνται είτε κατά φάκελο είτε κατά τύπο αρχείου. Τις περισσότερες φορές, η λίστα περιλαμβάνει σημαντικά δεδομένα, όπως το όνομα του διαγραμμένου αρχείου, τη διαδρομή του, την κατάστασή του, το μέγεθος και την ημερομηνία τροποποίησης. Τέλος, επιλέγουμε το αρχείο ή τα αρχεία που επιθυμούμε να επαναφέρουμε και με ένα κλικ τα επαναφέρουμε.

Επιπρόσθετα, η ορθή οργάνωση των αρχείων μας σε φακέλους και υποφακέλους, καθώς και η χρήση αντιπροσωπευτικών ονομάτων για τα νέα αρχεία που δημιουργούμε, αναμένεται ότι

θα μας βοηθήσουν να αντιμετωπίσουμε αποτελεσματικά το πρόβλημα της απώλειας αρχείων λόγω άθελης επικάλυψης (overwrite) ενός παλιού αρχείου από ένα καινούργιο. Φυσικά, ακόμα και σε αυτή την περίπτωση, κάποια προγράμματα όπως είναι και οι τελευταίες εκδόσεις του επεξεργαστή κειμένου Microsoft Word, μας δίνουν τη δυνατότητα να επιστρέψουμε σε παλαιότερες εκδόσεις του αρχείου μας χρησιμοποιώντας τη διαδικασία της Διαχείρισης Εκδόσεων (Manage Versions). Τέλος, ακόμα ένας τρόπος προστασίας των προγραμμάτων του υπολογιστή μας από την κατά παράληψη διαγραφή τους είναι είτε η δημιουργία Σημείων Επαναφοράς (Restore Point) του συστήματος είτε η δημιουργία Ειδώλων Δίσκου (Disk Images) επαναφοράς του συστήματος.



Εικόνα 16 MS-Word, Διαχείριση Εκδόσεων

3. Προβλήματα που Σχετίζονται με τον Χρήστη

3.1 Φυσικές Παθήσεις

Στη σημερινή κοινωνία της πληροφορίας και της γνώσης, τόσο σε προσωπικό όσο και σε εργασιακό επίπεδο, οι ώρες χρήσης του υπολογιστή έχουν αυξηθεί κατακόρυφα. Η πολύωρη χρήση του ηλεκτρονικού υπολογιστή εγκυμονεί κινδύνους για την υγεία μας. Η κακή στάση του σώματος, η λανθασμένη τοποθέτηση και ρύθμιση του υπολογιστή και των περιφερειακών του μονάδων και οι επαναλαμβανόμενες κινήσεις μπορούν να προκαλέσουν διάφορα προβλήματα υγείας. Παρακάτω, περιγράφονται μερικές φυσικές παθήσεις οι οποίες σχετίζονται με τη λανθασμένη χρήση του υπολογιστικού μας συστήματος.

Σύνδρομο Καρπιαίου Σωλήνα (RSI - Repetitive Strain Injury): Είναι μια επώδυνη σταδιακά αυξανόμενη κατάσταση στο χέρι η οποία προκαλείται από συμπίεση του μέσου νεύρου στον καρπό. Αποκαλείται και «ασθένεια του υπολογιστή» καθώς πλήττει περισσότερο όσους χρησιμοποιούν για ώρες το πληκτρολόγιο καταπονώντας τον καρπό τους.

Αυχενικό σύνδρομο: Μια ομάδα ενοχλημάτων γύρω από τον αυχένα, που περιλαμβάνει αυχεναλγία, δυσκαμψία, αντανακλώμενο πόνο και νευραλγία. Ένα από τα βασικά αίτια του αυχενικού συνδρόμου είναι η κακή στάση του χρήστη μπροστά από τον υπολογιστή.

Πόνοι πλάτης: Σε αρκετές περιπτώσεις οφείλονται στη γενικότερα κακή και μονότονη σωματική μας στάση μπροστά από τον ηλεκτρονικό υπολογιστή και στην αδρανοποίηση των μυών. Η έλλειψη κίνησης και άσκησης εξασθενεί τους στηρικτικούς μυς και επαυξάνει την πιθανότητα πρόκλησης βλαβών σε μεσοσπονδύλιους δίσκους, συνδέσμους και άλλους ιστούς της πλάτης.

Κεφαλαλγίες/Πονοκέφαλος: Πόνος στη περιοχή του κεφαλιού. Σε αρκετές περιπτώσεις, η κακή στάση του σώματός μας μπροστά από τον υπολογιστή ή ακόμα η κακή ποιότητα της οθόνης (μέγεθος, φωτεινότητα, αντανάκλασεις και απόσταση) θεωρούνται από τα βασικότερα αίτια του πρωτοπαθή πονοκεφάλου.

Ενοχλήσεις ματιών: Τα προβλήματα από τη χρήση ηλεκτρονικών υπολογιστών αποτελούν τον δεύτερο πιο συχνό λόγο για νεαρά υγιή άτομα να επισκέπτονται τον οφθαλμίατρο. Αυτό που συμβαίνει σε άτομα που χρησιμοποιούν για πάρα πολλές ώρες την ημέρα ηλεκτρονικούς υπολογιστές είναι κατά βάση μία «ασθενωπεία», δηλαδή μία δυσκολία στην όραση. Παρόλο που οι επιπτώσεις που



Εικόνα 17 Κακή στάση μπροστά από τον υπολογιστή



Εικόνα 18 Λανθασμένη Στάση

μπορεί να έχουν οι υπολογιστές στην όραση και στα μάτια γενικότερα βρίσκονται ακόμα κάτω από διαρκή παρατήρηση και ακόμη δεν έχει διαπιστωθεί κάποια σημαντική μόνιμη βλάβη, φαίνεται ότι τα συμπτώματα αυτά, αρκετές φορές, επηρεάζουν αρνητικά τη γενικότερη υγεία του ατόμου.

Σύνδρομο του Ψημένου Δέρματος (Toasted Skin Syndrome): Δουλεύοντας με έναν φορητό υπολογιστή στα πόδια σας, μπορεί να αντιμετωπίσετε το «σύνδρομο του ψημένου δέρματος». Η θερμότητα που εκπέμπει ο φορητός μας υπολογιστής όταν τον ακουμπάμε στα

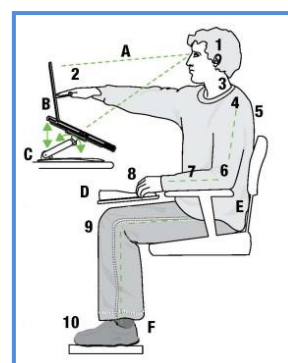


Εικόνα 19 Σύνδρομο του Ψημένου Δέρματος

πόδια μας σε πολλές περιπτώσεις αγγίζει τους 52 βαθμούς Κελσίου. Όταν λοιπόν για πολλές ημέρες κάνουμε όλες τις εργασίες μας με τον φορητό υπολογιστή στα πόδια τότε το δέρμα μας που δέχεται όλη αυτή τη θερμότητα, αρχίζει να αποχρωματίζεται, να παθαίνει εγκαύματα, ενώ οι επιστήμονες μιλάνε και για κίνδυνο εμφάνισης καρκίνου του δέρματος, σε πολύ μικρό ποσοστό βέβαια. Παλαιότερες έρευνες είχαν δείξει ότι η τοποθέτηση και χρήση των φορητών υπολογιστών στα πόδια, είναι πιθανόν να δημιουργήσει προβλήματα γονιμότητας στους άνδρες, διότι η υψηλή θερμοκρασία καταστρέφει το σπέρμα.

Με στόχο την πρόληψη και αντιμετώπιση των παραπάνω προβλημάτων είναι σημαντικό να τηρούμε τους παρακάτω κανόνες οι οποίοι απορρέουν από την επιστημονική έρευνα στον τομέα της εργονομίας. Η **εργονομία** έχει μεγάλη κοινωνική και πρακτική σημασία, αφού ασχολείται άμεσα με τις συνθήκες εργασίας, μεγιστοποιώντας την αποδοτικότητα και την παραγωγικότητα και παράλληλα δημιουργώντας τις προϋποθέσεις για τη διασφάλιση της υγείας όσων ασχολούνται συστηματικά με τη χρήση υπολογιστών.

1. Το κεφάλι μας θα πρέπει να στέκεται όρθιο και ευθύ, κάθετο στους ώμους (ή με μια πολύ μικρή κλίση προς τα κάτω). Θα πρέπει να τοποθετούμε την οθόνη του υπολογιστή μας απέναντί μας, ώστε να αποφεύγεται η κλίση και η στροφή του κεφαλιού σε σχέση με τον κορμό.
2. Με δεδομένη μια ελαφριά αποδεκτή κλίση του κεφαλιού προς τα κάτω θεωρούμε ότι το εύρος της άνετης όρασης μπορεί να φτάσει από την οριζόντια ευθεία των ματιών μας μέχρι μια γωνία 45 μοιρών χαμηλότερα. Αυτό σημαίνει ότι μέσα σε αυτό το πεδίο πρέπει να τοποθετήσουμε το κέντρο της οθόνης μας ανάλογα με το μέγεθός της. Η απόσταση της οθόνης μας πρέπει να είναι τουλάχιστον 50 εκατοστά από τα μάτια μας. Επιπρόσθετα, θα πρέπει να ρυθμίζουμε την κλίση της οθόνης έτσι ώστε να αποφεύγονται οι αντανakλάσεις λόγω φωτισμού. Τέλος, θα πρέπει να ρυθμίζουμε το μέγεθος της γραμματοσειράς μας ώστε να μας επιτρέπει να διαβάζουμε με άνεση.
3. Ο λαιμός (ο αυχένας μας) θα πρέπει να είναι χαλαρός και να αποφεύγουμε να τον τεντώνουμε μπροστά ή να τον πιέζουμε προς τα πίσω. Παράλληλα, όπως ισχύει και για το κεφάλι, δεν πρέπει η θέση εργασίας μας σε σχέση με τον υπολογιστή να είναι τέτοια που να επιβάλλει ο λαιμός να στρίβει σε σχέση με τον κορμό.



Εικόνα 20 Εργονομική Στάση

4. Οι ώμοι μας θα πρέπει να μην είναι σηκωμένοι ούτε να στρέφουν προς τα μέσα. Στις περιπτώσεις που χρησιμοποιούμε φορητό υπολογιστή μπορεί να είναι αναγκαίο να προσαρτήσουμε σε αυτόν επιπλέον πληκτρολόγιο για να επιτύχουμε εργονομικά αποδεκτή στάση.
5. Η πλάτη μας πρέπει να στέκεται όρθια, διατηρώντας τη φυσιολογική της καμπυλότητα. Η ύπαρξη πλάτης στο κάθισμα είναι απαραίτητη, ενώ σε κάθε περίπτωση πρέπει να είναι ρυθμιζόμενη ώστε να τοποθετείται ανάλογα με το μέγεθος του χρήστη στο κατάλληλο ύψος και να στηρίζει τη σπονδυλική μας στήλη.
6. Οι αγκώνες μας πρέπει να είναι σε γωνία 90-120 μοιρών. Κατά συνέπεια το πληκτρολόγιο θα πρέπει να είναι τοποθετημένο έτσι ώστε να μην κρέμονται οι καρποί καθώς πληκτρολογούμε αλλά ούτε και να πρέπει να τεντώνονται προς τα πάνω.
7. Θα πρέπει να υπάρχει υποστήριξη (μπράτσα καθίσματος) για τον πήχη.
8. Θα πρέπει να υπάρχει η δυνατότητα να χρησιμοποιηθούν βοηθητικά μαξιλαράκια για τους καρπούς έτσι ώστε να καλύπτεται η διαφορά του ύψους που μπορεί να έχει είτε το πληκτρολόγιο είτε το ποντίκι.
9. Τα γόνατα πρέπει να βρίσκονται ελαφρώς χαμηλότερα από τους γοφούς και παράλληλα θα πρέπει να υπάρχει η δυνατότητα κίνησής τους κάτω από το γραφείο.
10. Τα πόδια μας θα πρέπει να πατούν σταθερά στο έδαφος και τα πέλματα να μην αναγκάζονται να κρέμονται ή να λυγίζουν λόγω ψηλού ή κοντού καθίσματος αντίστοιχα. Για τη διόρθωση του ύψους μπορούμε να χρησιμοποιούμε είτε ρυθμιζόμενα καθίσματα είτε υποπόδια.

Εκτός από αυτά που έχουν αναφερθεί παραπάνω θα πρέπει, σε τακτά χρονικά διαστήματα, να διακόπτουμε την καθιστική μας εργασία ακολουθώντας κάποιες κινήσεις χαλάρωσης και «ξεπιασίματος». Θα μπορούσαμε να εναλλάσσουμε εργασίες που γίνονται με υπολογιστή, με εργασίες στις οποίες δεν χρειάζεται η χρήση υπολογιστή. Επίσης, ο χώρος στον οποίο είναι τοποθετημένος ο υπολογιστής μας θα πρέπει να φωτίζεται ομοιόμορφα και να μη βρίσκεται πηγή φωτός στο πλάι της οθόνης. Τέλος, θα πρέπει να ανοιγοκλείνουμε τα μάτια μας σε τακτά διαστήματα για λίγα λεπτά και κάθε 10 λεπτά να εστιάζουμε σε ένα σημείο εκτός οθόνης.

3.2 Παραγωγικότητα

Όπως προαναφέρθηκε, η υιοθέτηση των παραπάνω κανόνων εργονομίας, θα ελαχιστοποιήσει τις πιθανότητες προσβολής του χρήστη από τις φυσικές παθήσεις που περιγράφηκαν. Παράλληλα όμως, και σε συνδυασμό με ένα σωστά διαρρυθμισμένο περιβάλλον εργασίας μέσα στο οποίο χρησιμοποιείται εργονομικά σχεδιασμένο υλικό και λογισμικό, αναμένεται να αυξηθεί και η παραγωγικότητα στον χώρο εργασίας.

Κατά συνέπεια, ένας από τους σημαντικούς παράγοντες που επηρεάζουν την επιλογή του τρόπου διαρρύθμισης, της επίπλωσης, του υλικού που αναμένεται να χρησιμοποιηθεί αλλά και του λογισμικού είναι το κατά πόσο συμβάλλουν θετικά στο επίπεδο εργονομίας του εργασιακού μας περιβάλλοντος. Η απουσία αυτών των παραγόντων θα επιφέρει μείωση στην παραγωγικότητα και αυτό μεταφράζεται σε αύξηση των απαιτούμενων πόρων που πρέπει να δαπανηθούν για να ολοκληρωθεί μια συγκεκριμένη εργασία.

3.3 Εθισμός

Ο εθισμός στους υπολογιστές ή στο διαδίκτυο, αν και δεν έχει επισήμως αναγνωριστεί ως κλινική οντότητα παρά μόνο σε λίγες χώρες, αποτελεί μια κατάσταση που προκαλεί σημαντική έκπτωση στην κοινωνική και επαγγελματική ή ακαδημαϊκή λειτουργικότητα του ατόμου. Η διεθνής βιβλιογραφία στον τομέα της ψυχιατρικής αναφέρεται συχνά σε αυτού του τύπου τον εθισμό με επιστημονικά άρθρα τα οποία πραγματεύονται μεταξύ άλλων, τη Διαταραχή από την Χρήση Διαδικτυακών Παιχνιδιών (Internet Use Gaming Disorder), τη Διαταραχή από τον Εθισμό στο Διαδίκτυο (Internet Addiction Disorder - IAD), την Παθολογική Χρήση του Διαδικτύου (Pathological Internet Use) και την Προβληματική Χρήση του Διαδικτύου (Problematic Internet Use).

Σε αρκετές χώρες, η κατάχρηση της τεχνολογίας και η σχέση εξάρτησης που δημιουργείται από αυτήν, καθώς και ο εθισμός των νέων στο διαδίκτυο, έχουν πάρει ανησυχητικές διαστάσεις. Οι εθισμένοι χρήστες παρουσιάζουν διάφορα συμπτώματα όπως αίσθημα ευφορίας μόνο όταν βρίσκονται μπροστά από τον υπολογιστή, ανησυχία, νευρικότητα, έλλειψη ελέγχου του χρόνου ασχολίας τους με τους υπολογιστές, παραμέληση της προσωπικής τους ζωής και των υποχρεώσεών τους, δυσανάλογη σπατάλη χρημάτων για την αγορά λογισμικού και εξοπλισμού, διατροφικές διαταραχές, διαταραχές του ύπνου, μυοσκελετικές παθήσεις, μειωμένη αθλητική δραστηριότητα, ξηρά μάτια, ημικρανίες και παραμέληση της προσωπικής τους υγιεινής.

Με στόχο την αποτελεσματική πρόληψη και αντιμετώπιση της εξάρτησης αυτής θα πρέπει, αφού ευαισθητοποιηθούμε και ενημερωθούμε για το φαινόμενο του εθισμού, να σχεδιάσουμε και να οριοθετήσουμε το καθημερινό μας πρόγραμμα με τρόπο που η ώρα χρήσης του ηλεκτρονικού υπολογιστή και του διαδικτύου να είναι περιορισμένη και παράλληλα να το εμπλουτίσουμε με εναλλακτικές δραστηριότητες, οι οποίες μπορεί να περιλαμβάνουν την ενασχόλησή μας με ομαδικά αθλήματα, με χορωδίες και γενικότερα με άλλες ασχολίες που μας δίνουν την ευκαιρία να συναναστραφούμε με άλλα άτομα της ηλικίας μας.

3.4 Απώλεια προσωπικών δεδομένων

Αρκετά από τα δεδομένα που είναι αποθηκευμένα στον ηλεκτρονικό μας υπολογιστή και τις περιφερειακές του μονάδες θεωρούνται «ευαίσθητα» αφού η απώλειά τους ή η μη εξουσιοδοτημένη πρόσβαση σε αυτά μπορεί να μας προκαλέσει σημαντικά προβλήματα τόσο σε προσωπικό όσο και σε επαγγελματικό επίπεδο. Ένας τρόπος απώλειας ή διαρροής των δεδομένων αυτών μπορεί να σχετίζεται με την ανεπαρκή προστασία τους. Σε κάποιες άλλες περιπτώσεις, η υποκλοπή προσωπικών δεδομένων μπορεί να σχετίζεται με την παραπλάνηση ενός χρήστη μέσω ηλεκτρονικού μηνύματος ή μέσω μιας εφαρμογής διαδραστικής επικοινωνίας ή μέσω μιας ιστοσελίδας κοινωνικής δικτύωσης, έτσι ώστε να δώσει προσωπικές πληροφορίες όπως είναι ο αριθμός ταυτότητας, ο αριθμός τραπεζικού λογαριασμού και διάφοροι κωδικοί πρόσβασης. Συνήθως, σε αυτές τις περιπτώσεις, ο χρήστης παραπλέμπεται σε μια «πλαστή» ιστοσελίδα όπου του ζητείται η καταχώριση διάφορων προσωπικών στοιχείων.

Για να αντιμετωπίσουμε το συγκεκριμένο πρόβλημα μπορούμε να ακολουθούμε τις εξής συμβουλές:

- Οι κωδικοί πρόσβασης που χρησιμοποιούμε θα πρέπει να είναι δύσκολο να εντοπισθούν ή να τους μαντέψει κάποιος. Οι κωδικοί πρόσβασης θα πρέπει να είναι από 6 ως 8 χαρακτήρες τουλάχιστον, και σε αυτούς θα πρέπει να συμπεριλαμβάνονται τόσο

γράμματα όσο και αριθμοί. Δεν πρέπει να χρησιμοποιούμε ως κωδικούς πρόσβασης ονόματα συγγενών, το όνομα του/της φίλου/ης μας, την ημερομηνία γέννησής μας και άλλα.

- Οι κωδικοί πρόσβασης θα πρέπει να φυλάγονται σε ασφαλές μέρος και όχι να αναγράφονται είτε στο πίσω μέρος του πληκτρολογίου ή σε κάποιο άλλο εμφανές μέρος του χώρου εργασίας μας. Προτιμότερο είναι να τους αποστηθίζουμε και να μην τους γράφουμε πουθενά.
- Δεν πρέπει να αποκαλύπτουμε τον κωδικό πρόσβασης σε κανέναν άλλο.
- Θα πρέπει να αλλάζουμε τον κωδικό πρόσβασής μας σε τακτά χρονικά διαστήματα.
- Δεν δίνουμε ποτέ προσωπικές ή εμπιστευτικές πληροφορίες μέσω ηλεκτρονικού ταχυδρομείου.
- Σε περίπτωση που αντιληφθούμε ότι έχει γίνει υποκλοπή των προσωπικών μας δεδομένων μέσω μίας ιστοσελίδας κοινωνικής δικτύωσης, επικοινωνούμε άμεσα με τον διαχειριστή της και του ζητούμε είτε να καταργήσει άμεσα τον λογαριασμό μας είτε να αλλάξει τους κωδικούς πρόσβασής μας. Παράλληλα, καταγγέλλουμε το περιστατικό στην ιστοσελίδα www.cyberethics.info ή στο τηλέφωνο 22674747 (Γραμμή Καταγγελιών Hotline), ή/και στην αστυνομία.

Βασικές Έννοιες

Προβλήματα που σχετίζονται με το Υλικό:

- Περιβάλλον που Βρίσκεται το Υπολογιστικό Σύστημα (θερμοκρασία, σκόνη και υγρασία).
- Διακοπή Παροχής Ηλεκτρικού Ρεύματος
- Φυσική Φθορά
- Εμφάνιση Νέων Υπολογιστικών Αναγκών
- Κλοπή.

UPS (Uninterruptable Power Supply/ Αδιάλειπτη Παροχή Ενέργειας):

Συσκευή η οποία συνδέεται με το υπολογιστικό μας σύστημα και σε περίπτωση που για οποιονδήποτε λόγο διακοπεί η παροχή ηλεκτρικού ρεύματος, το τροφοδοτεί με ηλεκτρική ενέργεια για ένα χρονικό διάστημα, δίνοντας έτσι την ευκαιρία στον χρήστη να τερματίσει με ασφάλεια τη λειτουργία του υπολογιστή και των εφαρμογών που χρησιμοποιεί τη δεδομένη στιγμή. Ταυτόχρονα, φιλτράρει το ρεύμα, έτσι ώστε να έχει συνεχώς τη σωστή συχνότητα (50 Hz - 60 Hz).

Αναβάθμιση υλικού:

Είτε η αντικατάσταση παλαιότερων συσκευών του συστήματός μας είτε ο εμπλουτισμός του συστήματός μας με νέες συσκευές.

Προβλήματα που σχετίζονται με το Λογισμικό:

- Απώλεια Δεδομένων
- Κακόβουλα Προγράμματα
- Προβλήματα εγκατάστασης και συμβατότητας
- Εμφάνιση νέων υπολογιστικών αναγκών
- Σφάλματα χρήστη.

Αναβάθμιση Λογισμικού:

Είτε η αντικατάσταση παλαιότερων προγραμμάτων με νέα ή νέες εκδόσεις τους, είτε ο εμπλουτισμός του συστήματός μας με νέα προγράμματα.

Προβλήματα που σχετίζονται με τον Χρήστη:	• Φυσικές Παθήσεις • Παραγωγικότητα • Εθισμός • Απώλεια Προσωπικών Δεδομένων
Σύνδρομο Καρπιαίου Σωλήνα:	(RSI - Repetitive Strain Injury). Είναι μια επώδυνη κατάσταση στο χέρι, συγκεκριμένα στην περιοχή του καρπού, η οποία προκαλείται από συμπίεση του μέσου νεύρου στον καρπό.
Σύνδρομο του Ψημένου Δέρματος:	(Toasted Skin Syndrome). Αποχρωμάτιση του δέρματος στα πόδια ή σε άλλα μέρη του σώματός μας, η οποία προκαλείται από τη θερμότητα που εκπέμπει ο φορητός μας υπολογιστής όταν τον ακουμπάμε πάνω στα πόδια μας.
Εργονομία:	Επιστήμη η οποία ασχολείται με τις συνθήκες εργασίας, και έχει στόχο τόσο τη μεγιστοποίηση της αποδοτικότητας και της παραγωγικότητας όσο και για τη διασφάλιση της υγείας των εργαζομένων.
Παραγωγικότητα:	Η σχέση μεταξύ των απαιτούμενων πόρων (εισροών) που πρέπει να δαπανηθούν για να ολοκληρωθεί μια συγκεκριμένη εργασία και του αποτελέσματος (εκροών) της εργασίας.
Εθισμός:	Διαταραχή η οποία προκαλεί σημαντική έκπτωση στην κοινωνική και επαγγελματική ή ακαδημαϊκή λειτουργικότητα του ατόμου και σχετίζεται με την υπέρμετρη και προβληματική χρήση των υπολογιστών και του διαδικτύου.

Πηγές

1. Αράπογλου Α., Μαβόγλου Χ., Οικονομάκος Η., Φύτρος Κ., (2006) Πληροφορική Α', Β', Γ' Γυμνασίου, σελ. 109-112, ΟΕΔΒ.
2. Ασφάλεια στο Διαδίκτυο. (n.d.). Παιδαγωγικό Ινστιτούτο Κύπρου. Ανακτήθηκε Ιανουάριος 12, 2013 από, <http://www.pi.ac.cy/InternetSafety/parent.html>.
3. Βουτηράς Γ., Κονιδάρη Ε., Κούτρας Μ., Σφώρος Ν., (2003) Πληροφορική Γυμνασίου, σελ. 38-39, ΟΕΔΒ.
4. Γενικά προβλήματα ηλεκτρονικού υπολογιστή. *Vstore.gr*. Ανακτήθηκε Ιανουάριος 1, 2013 από, http://support.vstore.gr/index.php?option=com_content&view=article&id=14:2012-01-04-14-14-49&catid=1:2011-12-15-12-23-02.
5. Γιαλλουρίδης Κ., Γκιπέρτης Ε., Κόμης Β., Σιδερίδης Α., Σταθόπουλος Κ. (1999) Εφαρμογές Πληροφορικής Υπολογιστών Α', Β', Γ' Ενιαίου Λυκείου, σελ. 39-42, ΟΕΔΒ.
6. Γουλτίδης, Χ., (2008). *ECDL 4 Γρήγορα και Απλά*. σελ. 83-91, Κλειδάριθμος.
7. Ηλεκτρονικός Εθισμός ή Ηλεκτρονική Μορφίνη. (n.d.). *Olympia.gr*. Ανακτήθηκε Ιανουάριος 1, 2013 από, <http://olympia.gr/2010/10/11/>.
8. Ένας οδηγός για να παραμείνετε ασφαλείς στο διαδίκτυο. (n.d.). *Google*. Ανακτήθηκε Ιανουάριος 1, 2013 από, <http://www.google.com.cy/intl/el/goodtoknow/>.
9. Internet & Computer Addiction. (n.d.). *Helpguide.org*. Ανακτήθηκε Ιανουάριος 1, 2013 από, http://www.helpguide.org/mental/internet_cybersex_addiction.htm.

10. Περιγραφή της βασικής ορολογίας που χρησιμοποιείται για την περιγραφή ενημερωμένων εκδόσεων λογισμικού της Microsoft. (n.d.). *Microsoft*. Ανακτήθηκε Ιανουάριος 1, 2013 από, <http://support.microsoft.com/kb/824684/el>.
11. Προστασία Προσωπικών Δεδομένων. (n.d.). Αρχή Προστασίας Δεδομένων Προσωπικού Χαρακτήρα. Ανακτήθηκε Ιανουάριος 1, 2013 από, http://www.dpa.gr/portal/page?_pageid=33,15048&_dad=portal&_schema=PORTAL
12. Σωστή θέση εργασίας στον υπολογιστή; (n.d.). *PhysioAid*. Ανακτήθηκε Ιανουάριος 1, 2013 από, http://www.physio-aid.gr/index.php?option=com_content&view=article&id=87:2009-09-30-07-46-14&catid=4:2009-05-10-09-25-22&lang=en
13. Τροφοδοσία ηλεκτρικής ισχύος. *Κέντρο Πληροφορικής & Νέων Τεχνολογιών Ηλείας*. Ανακτήθηκε Ιανουάριος 1, 2013 από, <http://dide.ilei.sch.gr/keplinet/tech/ups.php>.
14. UPS. (n.d.). *In Wikipedia*. Ανακτήθηκε Ιανουάριος 1, 2013 από, <http://el.wikipedia.org/wiki/UPS>.
15. Χάλασε ο σκληρός δίσκος, η κάρτα μνήμης ή το USB; (n.d.). *Tictac Laboratories*. Ανακτήθηκε Ιανουάριος 1, 2013 από, <http://www.tictac.gr/>.

Γ2.4

Πώς Συγκρίνουμε και Πώς Επιλέγουμε Υπολογιστικά Συστήματα

Τι θα μάθουμε σήμερα:

- ❖ Να συγκρίνουμε υπολογιστικά συστήματα με βάση τα χαρακτηριστικά του υλικού (ταχύτητα ΚΜΕ, ποσότητα μνήμης RAM, χωρητικότητα δίσκου, χαρακτηριστικά οθόνης, εκτυπωτή και άλλων μονάδων/περιφερειακών συσκευών) και του λογισμικού που διαθέτουν
- ❖ Να επιλέγουμε από έναν κατάλογο κατάλληλο υπολογιστικό σύστημα (ή μέρη του) για εξυπηρέτηση του χρήστη σε ένα παράδειγμα και να στοιχειοθετούμε την απόφασή μας.

1. Εισαγωγή

Έχουμε ήδη ανακεφαλαιώσει τα βασικά χαρακτηριστικά του υλικού (εξαρτημάτων και περιφερειακών μονάδων) του υπολογιστή. Επίσης, θυμηθήκαμε τον ρόλο του λογισμικού εφαρμογών και του λογισμικού συστήματος (κυρίως του λειτουργικού συστήματος) και πώς αυτά σχετίζονται μεταξύ τους. Παράλληλα, αναφερθήκαμε στον ρόλο του χρήστη, που πέρα από την αλληλεπίδρασή του με τον υπολογιστή μέσα από το λογισμικό καθορίζει και ποιο λογισμικό και ποιο υλικό θα πρέπει να υπάρχει, ώστε να καλύπτονται οι υπολογιστικές ανάγκες του. Ακολούθησε ανάλυση των προβλημάτων που πιθανόν να παρουσιαστούν στο υλικό, το λογισμικό και τον χρήστη και παρουσιάστηκαν τρόποι πρόληψης ή αντιμετώπισής τους. Ο χρήστης, το λογισμικό εφαρμογών, το λογισμικό συστήματος και το υλικό μαζί αποτελούν αυτό που ονομάζουμε υπολογιστικό σύστημα. Για να λειτουργεί ικανοποιητικά το υπολογιστικό σύστημα, θα πρέπει τα μέρη του να συνεργάζονται ομαλά και αποτελεσματικά.

Σε αυτό το κεφάλαιο θα μελετήσουμε τι πρέπει να λαμβάνουμε υπόψη μας όταν επιλέγουμε υλικό και λογισμικό, ώστε να εξυπηρετούνται οι ανάγκες του χρήστη, αλλά να υπάρχει ταυτόχρονα και ομαλή συνεργασία μεταξύ τους, χωρίς να ξεχνάμε ότι το κόστος αγοράς και συντήρησης είναι ένας από τους καθοριστικούς παράγοντες.

2. Ο χρήστης και οι υπολογιστικές ανάγκες του

Ο κάθε άνθρωπος χρησιμοποιεί τον υπολογιστή σύμφωνα με τις δικές του ανάγκες. Ένας μαθητής/Μια μαθήτρια πιθανόν να χρησιμοποιεί τον υπολογιστή του για να αναζητά πληροφορίες στο Διαδίκτυο, να δημιουργεί παρουσιάσεις για τα μαθήματά του, να ακούει μουσική, να παίζει παιχνίδια και να επικοινωνεί με φίλους/φίλες του/της. Ένας υπάλληλος σε μια ασφαλιστική εταιρεία χρειάζεται έναν υπολογιστή που να κρατά ημερολόγιο των συναντήσεων με τους πελάτες του, να στέλνει και να λαμβάνει μηνύματα ηλεκτρονικού ταχυδρομείου, να επιτρέπει τον υπολογισμό του κόστους διάφορων ασφαλιστικών σχεδίων (π.χ. σε υπολογιστικό φύλλο) και να έχει διάρκεια μπαταρίας, ώστε να μη χρειάζεται να τον φορτίζει παρά μόνο στο τέλος κάθε εργάσιμης ημέρας. Ένας αρχιτέκτονας χρειάζεται έναν υπολογιστή με μεγάλη οθόνη, κατάλληλο σχεδιαστικό πρόγραμμα και ποντίκι ακριβείας ή γραφίδα και εκτυπωτή ή σχεδιαστή (plotter) για μεγάλο μέγεθος χαρτιού.

Στην κάθε περίπτωση θα πρέπει να εντοπίζονται τα χαρακτηριστικά και οι δυνατότητες που απαιτούνται (οι προδιαγραφές), για να έχουμε όσο το δυνατόν ένα σύστημα που από τη μία να εξυπηρετεί και να καλύπτει τις ανάγκες/απαιτήσεις του χρήστη και από την άλλη να μην περιλαμβάνει υλικό και λογισμικό που δεν είναι αναγκαίο, αυξάνοντας αχρείαστα το κόστος αγοράς. Συχνά παραλείπουμε να λάβουμε υπόψη το κόστος όταν αγοράζουμε ένα υπολογιστικό σύστημα για προσωπική μας χρήση (ιδιαίτερα εάν θα το πληρώσουν οι γονείς μας). Όταν όμως πρόκειται για επαγγελματική χρήση, μια επιχείρηση ή ένας οργανισμός θα

πρέπει να μελετήσει τις ανάγκες και το κόστος προσεκτικά, αφού οι οικονομικές συνέπειες μπορεί να είναι σοβαρές.

Για παράδειγμα, σε μια εταιρεία η αντικατάσταση 200 υπολογιστών με καινούργιους αξίας €1.000 ο καθένας θα στοιχίσει €200.000. Εάν αγοραστούν φθηνότεροι υπολογιστές αξίας €800 ο καθένας, νοουμένου ότι καλύπτονται οι ανάγκες των χρηστών, θα στοιχίσει €160.000, δηλαδή εξοικονομούνται €40.000.

Παράλληλα, μια λανθασμένη επιλογή με κριτήριο μόνο το κόστος, πιθανόν να αποδειχτεί ζημιογόνα, διότι εάν δεν εξυπηρετούνται οι ανάγκες των χρηστών θα χρειαστούν αντικατάσταση ή αναβάθμιση των υπολογιστικών συστημάτων, επιβαρύνοντας έτσι τον προϋπολογισμό.

3. Επιλογή λογισμικού εφαρμογών και λειτουργικού συστήματος

Το λογισμικό εφαρμογών που απαιτείται προκύπτει από τις ανάγκες του χρήστη, αφού με αυτό θα τις εξυπηρετεί. Οι κύριοι παράγοντες που επηρεάζουν την επιλογή είναι το **κόστος αγοράς** του λογισμικού και πιθανή προηγούμενη εμπειρία του χρήστη ή απαίτηση για συγκεκριμένο λογισμικό. Για παράδειγμα, ένας διευθυντής πωλήσεων, ο οποίος χρειάζεται να δημιουργεί ηλεκτρονικές παρουσιάσεις και είναι εξοικειωμένος με το πρόγραμμα *Microsoft PowerPoint* (το οποίο θα πρέπει να αγοράσει) είναι πιθανόν να μην επιθυμεί να χρησιμοποιήσει το αντίστοιχο πρόγραμμα παρουσιάσεων *LibreOffice Impress*, το οποίο προσφέρεται δωρεάν.

Σε αρκετές περιπτώσεις, ιδιαίτερα με εξειδικευμένο λογισμικό (π.χ. προγράμματα σχεδίασης για αρχιτέκτονες), το κόστος του λογισμικού μπορεί να ξεπεράσει το κόστος του υλικού του υπολογιστή όπου θα εγκατασταθεί.

Ο δημιουργός ενός προγράμματος (λογισμικού) καθορίζει τις **απαιτήσεις (requirements)** του υλικού οι οποίες απαιτούνται για τη λειτουργία του. Σε αρκετές περιπτώσεις δίνονται ξεχωριστά οι ελάχιστες απαιτήσεις (minimum requirements), δηλαδή τα απολύτως απαραίτητα για τη λειτουργία του (αλλά όχι υποχρεωτικά την καλή λειτουργία του). Για το λογισμικό εφαρμογών ως μέρος των απαιτήσεων καθορίζεται και το λειτουργικό σύστημα το οποίο είναι αναγκαίο για τη λειτουργία του.

Παραδείγματα:

- Για εγκατάσταση και χρήση του λειτουργικού συστήματος *Windows 7* οι απαιτήσεις είναι: ΚΜΕ αρχιτεκτονικής x86 με ταχύτητα επεξεργασίας τουλάχιστον 1 GHz, μνήμη RAM με χωρητικότητα τουλάχιστον 1 GB, σκληρό δίσκο με χωρητικότητα τουλάχιστον 16 GB.
- Για εγκατάσταση και χρήση του σχεδιαστικού προγράμματος *AutoCAD 2013* οι απαιτήσεις είναι: Λειτουργικό Σύστημα Microsoft Windows XP ή νεότερο, ΚΜΕ αρχιτεκτονικής x86 με ταχύτητα επεξεργασίας τουλάχιστον 1,6 GHz, μνήμη RAM με χωρητικότητα τουλάχιστον 2 GB (η εισήγηση είναι 4 GB, με 2 GB να είναι οι ελάχιστες προδιαγραφές), σκληρό δίσκο με ελεύθερο χώρο τουλάχιστον 6 GB, ανάλυση οθόνης τουλάχιστον 1024 x 768 (η εισήγηση είναι 1600 x 1050, με 1024 x 768 να είναι οι ελάχιστες προδιαγραφές).
- Για εγκατάσταση και χρήση του παιχνιδιού *SimCity* οι απαιτήσεις είναι: Λειτουργικό Σύστημα Microsoft Windows XP ή νεότερο, ΚΜΕ αρχιτεκτονικής x86 με ταχύτητα επεξεργασίας τουλάχιστον 2 GHz, μνήμη RAM με χωρητικότητα τουλάχιστον 2 GB,

σκληρό δίσκο με ελεύθερο χώρο τουλάχιστον 10 GB, κάρτα γραφικών με τουλάχιστον 512 MB Μνήμη RAM και σύνδεση με το Διαδίκτυο με ταχύτητα τουλάχιστον 256 kbps.

4. Επιλογή υλικού

Το υλικό του υπολογιστή θα πρέπει να ταιριάζει με τις απαιτήσεις του λογισμικού, όπως αναφέρθηκαν πιο πάνω. Επιπρόσθετα, θα πρέπει να επιλεγούν κατάλληλες περιφερειακές συσκευές ώστε να καλύπτουν τις ανάγκες του χρήστη. Παρά το γεγονός ότι οι περισσότερες περιφερειακές συσκευές σήμερα είναι φτιαγμένες (συμβατές) να λειτουργούν με διάφορα λειτουργικά συστήματα, επιβάλλεται να γίνει έλεγχος, διότι χωρίς κατάλληλο λογισμικό (οδηγό συσκευής) από τον κατασκευαστή για το συγκεκριμένο λειτουργικό σύστημα που επιλέξαμε, η περιφερειακή συσκευή πιθανόν να μην μπορεί να χρησιμοποιηθεί.

Επίσης, θα πρέπει να γίνει προσεκτικός συλλογισμός για τη μελλοντική χρήση του υπολογιστή, διότι υπάρχει πάντοτε η πιθανότητα να προστεθούν νέες ανάγκες του χρήστη μετά την αγορά του υπολογιστικού συστήματος. Ο χρόνος λειτουργίας ενός υπολογιστικού συστήματος πριν να εξεταστεί η αντικατάστασή του είναι συνήθως 3–4 χρόνια.

5. Πώς επιλέγουμε ένα υπολογιστικό σύστημα για κάποιον χρήστη

Παρακάτω θα εξετάσουμε μια συστηματική διαδικασία για επιλογή υπολογιστικού συστήματος (παρόμοια με αυτήν που υιοθετούν επαγγελματίες σύμβουλοι πληροφορικής).

5.1 Καταγραφή των αναγκών του χρήστη

Για να επιλέξουμε ένα υπολογιστικό σύστημα, αρχίζουμε με την καταγραφή των αναγκών του χρήστη. Με βάση τις ανάγκες αυτές, θα φανεί ποιο λογισμικό εφαρμογών απαιτείται, καθώς και κάποιες πληροφορίες, κυρίως για τις περιφερειακές συσκευές που χρειάζονται και για τον συνολικό αποθηκευτικό χώρο του σκληρού δίσκου που απαιτείται. Οι ανάγκες του χρήστη θα μας καθοδηγήσουν στο υλικό και το λογισμικό που απαιτείται. Παρακάτω δίνεται μια σειρά από ερωτήσεις, που απαντώντας τις μπορούμε να καταγράψουμε τις ανάγκες του χρήστη:

- **Χρειάζεται να μεταφέρεται ο υπολογιστής; Πόσο συχνά; Πόση αυτονομία (διάρκεια μπαταρίας) απαιτείται;** (Καθορίζεται εάν χρειάζεται φορητός υπολογιστής ή όχι και ποιοι τύπου).
- **Πόσο όγκο και τι είδους δεδομένα χρειάζεται να αποθηκεύονται μόνιμα;** (Καθορίζει το μέγεθος του σκληρού δίσκου ή άλλης βοηθητικής μνήμης).
- **Τι είδους προγράμματα (λογισμικό εφαρμογών) απαιτούνται για τις ανάγκες του χρήστη; Υπάρχουν περιορισμοί ή προτιμήσεις;**
- **Ποια λειτουργικά συστήματα μπορούν να χρησιμοποιηθούν; Υπάρχουν περιορισμοί ή προτιμήσεις;** (Κάποιοι οργανισμοί πιθανόν να ζητούν συγκεκριμένο λειτουργικό σύστημα, για το οποίο υπάρχει εμπειρία και υποστήριξη).
- **Χρειάζεται να γίνεται εκτύπωση/σάρωση έγχρωμη/μαυρόασπρη, ποιας ποιότητας, για ποιο σκοπό, ποιας ποσότητας;**
- **Χρειάζεται συγκεκριμένη οθόνη ή άλλες περιφερειακές συσκευές; Ποιες και με ποιες προδιαγραφές;**
- **Ποιες άλλες λειτουργίες χρειάζονται;** (π.χ. εγγραφή σε CD, επικοινωνία μέσω Διαδικτύου, κ.λπ.)
- **Η κατανάλωση ρεύματος είναι σημαντική για τον χρήστη;**
- **Τι άλλο θα πρέπει να προστεθεί για την προστασία του υλικού, του λογισμικού/των πληροφοριών και του χρήστη;** (Υλικό και λογισμικό για προστασία

του υπολογιστικού συστήματος, π.χ. συσκευές και λογισμικό για δημιουργία εφεδρικών αρχείων, πρόγραμμα καταπολέμησης ιών, κ.ά.)

5.2 Επιλογή λογισμικού εφαρμογών και λειτουργικού συστήματος

Με βάση τις ανάγκες του χρήστη που καταγράφηκαν προτείνεται συγκεκριμένο λογισμικό εφαρμογών. Σε περιπτώσεις όπου υπάρχουν εναλλακτικές επιλογές για κάποιο λογισμικό, μπορούν να εξεταστούν παράλληλα και να προταθούν στον χρήστη διαφορετικές εναλλακτικές επιλογές (ή να προταθεί η λύση με το χαμηλότερο κόστος).

Επιλέγοντας το λογισμικό εφαρμογών, σε πολλές περιπτώσεις καθορίζεται αυτόματα και το λειτουργικό σύστημα, από τις προδιαγραφές του λογισμικού εφαρμογών. Σε κάποιες περιπτώσεις όπου ο δημιουργός του λογισμικού εφαρμογών διαθέτει εκδοχές του για διαφορετικά λειτουργικά συστήματα (π.χ. *Mozilla Firefox*, *Skype*) ίσως χρειαστεί να εξεταστούν οι πιθανοί εναλλακτικοί συνδυασμοί.

Με βάση τις απαιτήσεις τόσο του λογισμικού εφαρμογών όσο και του λειτουργικού συστήματος καθορίζονται οι υπόλοιπες απαιτήσεις για το υλικό (ταχύτητα ΚΜΕ, χωρητικότητα μνήμης RAM και σκληρού δίσκου ή άλλων μέσων αποθήκευσης, κ.λπ.).

5.3 Επιλογή υλικού

Μέσα από τη διαδικασία που ακολουθούμε θα έχουν ήδη συγκεντρωθεί οι απαιτήσεις για το υλικό (ΚΜΕ Μνήμη, σκληρό δίσκο, κάρτα γραφικών, κάρτα δικτύου, περιφερειακές συσκευές, κ.λπ.).

Σε αυτό το στάδιο γίνεται αναζήτηση (από την αγορά) των τεχνικών προδιαγραφών και του κόστους εναλλακτικών επιλογών για αξιολόγηση. Αφού απορρίψουμε τυχόν επιλογές οι οποίες δεν ανταποκρίνονται στις απαιτήσεις, προχωρούμε στην αξιολόγηση και την επιλογή της καταλληλότερης για να προτείνουμε στον χρήστη.

Συχνά προκύπτει να έχουμε συσκευές των οποίων οι προδιαγραφές (τα χαρακτηριστικά τους) είναι αντίστοιχα με το κόστος αγοράς τους. Υπάρχουν όμως περιπτώσεις όπου δεν είναι ξεκάθαρο ποια είναι η καλύτερη ή η φθηνότερη λύση για τον χρήστη. Για παράδειγμα, έστω ότι έχουμε να διαλέξουμε από τους πιο κάτω δύο φορητούς υπολογιστές:

A: CPU 2.20 GHz, RAM 4 GB, HDD 500 GB, CD/DVD, οθόνη 17", τιμή €600

B: CPU 2.53 GHz, RAM 8 GB, HDD 250 GB, CD/DVD, οθόνη 15", τιμή €650

Ο **A** έχει χαμηλότερη ταχύτητα χρονιστή και λιγότερη μνήμη RAM αλλά διαθέτει σκληρό δίσκο μεγαλύτερης χωρητικότητας και μεγαλύτερη οθόνη από τον **B**. Σε τέτοιες περιπτώσεις, για να αποφασίσουμε ποιος είναι ο πιο κατάλληλος, θα πρέπει να αξιολογήσουμε «ζυγίζοντας» σε ποιο βαθμό οι διάφορες προδιαγραφές σχετίζονται με τις ανάγκες του χρήστη. Ένας τρόπος για να το κάνουμε είναι να καθορίσουμε πόση σημασία έχει η κάθε προδιαγραφή για τον χρήστη και να τις προσθέσουμε για να βγάλουμε τη συνολική «αξία» του κάθε υπολογιστή (ή περιφερειακής συσκευής) για τον συγκεκριμένο χρήστη, όπως έχουμε δει και στη Β' Τάξη με τον ακόλουθο τρόπο:

- Καταγράφουμε τα χαρακτηριστικά που είναι σημαντικά για τον χρήστη και την τιμή σε έναν πίνακα.
- Αποφασίζουμε και κατανέμουμε «μονάδες αξίας» σε κάθε χαρακτηριστικό ανάλογα με τις ανάγκες του χρήστη (σύνολο 100).

- Επιλέγουμε την τιμή η οποία χαρακτηρίζει την καλύτερη επιλογή για κάθε χαρακτηριστικό και υπολογίζουμε την επίδοση για το κάθε χαρακτηριστικό, από 0 (καθόλου) μέχρι 1 (τέλειο) διαιρώντας.
- Πολλαπλασιάζουμε την κάθε επίδοση με τις αντίστοιχες «μονάδες αξίας» και προσθέτουμε για να βρούμε τη συνολική αξία κάθε επιλογής.
- Μπορούμε, επίσης, να υπολογίσουμε την ποιότητα (αξία) ανά τιμή για κάθε επιλογή.

Αυτή η διαδικασία μπορεί να χρησιμοποιηθεί για κάθε συσκευή όπου χρειάζεται.

Στο τέλος προτείνουμε στον χρήστη με ολοκληρωμένο τρόπο το λογισμικό, το υλικό και το κόστος για το υπολογιστικό σύστημα.

5.4 Παράδειγμα 1

Η Δήμητρα μόλις έχει τελειώσει το Λύκειο και θα αρχίσει τις σπουδές της στον κλάδο των γραφικών τεχνών. Χρειάζεται ένα υπολογιστικό σύστημα, το οποίο να μπορεί να μεταφέρει και να χρησιμοποιεί τόσο στο πανεπιστήμιο όσο και στον χώρο διαμονής της κάθε μέρα. Η οθόνη θα πρέπει να είναι ψηλής ανάλυσης και μεγέθους τουλάχιστο 17" για να μπορεί να εμφανίζει σε αυτό τις διάφορες δημιουργίες της. Επίσης, θα πρέπει να διαθέτει αρκετά μεγάλο εσωτερικό αποθηκευτικό χώρο ώστε να μπορεί να αποθηκεύει τις δημιουργίες της για την τετραετή διάρκεια των σπουδών της (περίπου 30 τον χρόνο, η καθεμιά με μέγεθος 2 GB) παράλληλα χρειάζεται να γράφει εργασίες και να δημιουργεί παρουσιάσεις, αλλά και να επικοινωνεί με τους γονείς της μέσω Skype (ήχο και εικόνα). Δεν έχει συγκεκριμένες απαιτήσεις για το λογισμικό εφαρμογών, αλλά επιθυμεί να χρησιμοποιεί το λειτουργικό σύστημα Windows. Μπορεί να διαθέσει μόνο μέχρι 1000 ευρώ.

Ανάγκες του χρήστη:

- Χρειάζεται φορητός υπολογιστής. Δεν καθορίζεται συγκεκριμένη διάρκεια μπαταρίας.
- Όγκος δεδομένων για μόνιμη αποθήκευση: Δημιουργίες γραφικών 2 GB × 30 τον χρόνο × 4 χρόνια, σύνολο 240 GB (χρειάζεται επιπρόσθετος αποθηκευτικός χώρος για το λειτουργικό σύστημα και το λογισμικό εφαρμογών).
- Λογισμικό εφαρμογών που χρειάζεται: Επεξεργασία γραφικών, επεξεργασία κειμένου, πρόγραμμα παρουσιάσεων, Skype.
- Λειτουργικά συστήματα που μπορούν να χρησιμοποιηθούν: Windows.
- Οθόνη 17".
- Άλλες λειτουργίες: Σύνδεση με το Διαδίκτυο, μικρόφωνο, ηχεία, βιντεοκάμερα.
- Προστασία: Πρόγραμμα καταπολέμησης ιών.

Επιλογή λογισμικού εφαρμογών και λειτουργικού συστήματος:

Από τις ανάγκες προκύπτει ότι χρειάζεται:

- Λειτουργικό σύστημα: Windows (π.χ. Windows 7, το οποίο έχει απαιτήσεις ΚΜΕ αρχιτεκτονικής x86 με ταχύτητα επεξεργασίας τουλάχιστον 1 GHz, Μνήμη RAM με χωρητικότητα τουλάχιστον 1 GB, σκληρό δίσκο με χωρητικότητα τουλάχιστον 16 GB).
- Επεξεργαστής κειμένου/πρόγραμμα παρουσιάσεων: *Microsoft Office 2010 Home & Student Edition*, €119 ή εναλλακτικά *LibreOffice v.4*, €0 (και τα δύο χωρίς ιδιαίτερες προδιαγραφές για το υλικό πέρα από χωρητικότητα σκληρού δίσκου περίπου 1 GB).
- Πρόγραμμα γραφικών: *Adobe CC για Windows*, €1200 (ανάληψη οθόνης τουλάχιστον 1024x768, χώρος στον σκληρό δίσκο 2GB, σύνδεση με το Διαδίκτυο) ή εναλλακτικά *Inkscape και Gimp*, €0 (με τις ίδιες απαιτήσεις οθόνης και σκληρού δίσκου).

- *Skype* (χωρίς ιδιαίτερες προδιαγραφές για το υλικό, πέρα από ηχεία, μικρόφωνο και βιντεοκάμερα τα οποία ήδη σημειώθηκαν).
- Πρόγραμμα καταπολέμησης ιών: *Microsoft Security Essentials* (δωρεάν με τα *Windows*).

Από τις επιλογές που έχουν διερευνηθεί προτείνονται όσες είναι υπογραμμισμένες. Το *LibreOffice v.4* προτιμήθηκε διότι δεν έχει κόστος αγοράς, όπως και τα *Inkscape* και *Gimp*, διότι το εναλλακτικό *Adobe CC* από μόνο του ξεπερνά το διαθέσιμο ποσό.

Επιλογή υλικού:

Μελετώντας τα πιο πάνω προκύπτει ότι ο υπολογιστής που απαιτείται πρέπει να είναι φορητός, με ΚΜΕ αρχιτεκτονικής x86, με ταχύτητα επεξεργασίας τουλάχιστον 1 GHz, Μνήμη RAM με χωρητικότητα τουλάχιστον 2 GB, σκληρό δίσκο με χωρητικότητα τουλάχιστον 259 GB, οθόνη με ανάλυση 1024x768, ηχεία, μικρόφωνο και βιντεοκάμερα (παίρνουμε τις ψηλότερες απαιτήσεις από όλο το λογισμικό).

Έστω ότι αναζητήσαμε από προμηθευτές/καταστήματα πώλησης και πήραμε τις πιο κάτω εισηγήσεις (θεωρούμε ότι όλοι οι υπολογιστές είναι φορητοί και διαθέτουν CD/DVD, ηχεία, μικρόφωνο, βιντεοκάμερα, κάρτα δικτύου και εγκατεστημένο το Windows 7:

Y1: CPU 2,00 GHz, RAM 4 GB, HDD 500 GB, οθόνη 17" 1024x768, τιμή €480

Y2: CPU 2,00 GHz, RAM 4 GB, HDD 250 GB, οθόνη 15" 1200x800, τιμή €500

Y3: CPU 2,20 GHz, RAM 6 GB, HDD 500 GB, οθόνη 17" 1200x800, τιμή €650

Y4: CPU 2,53 GHz, RAM 8 GB, HDD 750 GB, οθόνη 17" 1800x1024, τιμή €800

Ο Y2 απορρίπτεται, διότι δεν έχει ικανοποιητική χωρητικότητα σκληρού δίσκου. Οι υπόλοιποι ικανοποιούν τις ανάγκες του χρήστη και με βάση την τιμή μπορούμε να επιλέξουμε τον Y1 ως την πιο οικονομική λύση. Εναλλακτικά μπορούμε να αξιολογήσουμε τους τρεις υπολογιστές με τη διαδικασία που αναφέρθηκε. Τα σημαντικά χαρακτηριστικά για τον χρήστη είναι η ταχύτητα χρονιστή της ΚΜΕ, η ποσότητα μνήμης RAM, η χωρητικότητα του σκληρού δίσκου και η ανάλυση οθόνης. Για το είδος της εργασίας, εκτός από τη χωρητικότητα του σκληρού δίσκου που είναι λιγότερο σημαντική τα υπόλοιπα χαρακτηριστικά είναι περίπου το ίδιο σημαντικά, έτσι τους δίνουμε 30, 30, 10 και 30 μονάδες αξίας αντίστοιχα (για ευκολία αξιολόγησης της ανάλυσης θα μετρήσουμε το σύνολο των εικονοστοιχείων) και καταγράφουμε τους υπολογισμούς στον ακόλουθο πίνακα:

Υπολογιστής	Χαρακτηριστικά				Κόστος	Επιδόσεις				Δείκτης Επίδοσης	Επίδοση/ Κόστος
	ΚΜΕ	RAM	Σκληρός Δίσκος	Οθόνη		ΚΜΕ	RAM	Σκληρός Δίσκος	Οθόνη		
						15	20	35	30		
Y1	2	4	500	786432	480	0,79	0,50	0,67	0,55	61,77	0,13
Y3	2,2	6	500	960000	650	0,87	0,75	0,67	0,67	75,25	0,12
Y4	2,53	8	750	1440000	800	1,00	1,00	1,00	1,00	100,00	0,13

2,53 8 750 1440000

Όπως βλέπουμε, ο φθηνότερος υπολογιστής έχει αξία 61, ενώ ο ακριβότερος 100. Επίσης, η επίδοση ανά κόστος είναι παρόμοια και στις τρεις περιπτώσεις. Αυτό δείχνει ότι ακόμη και αν προτείνουμε τον ακριβότερο υπολογιστή, ο χρήστης θα πάρει αξία αντίστοιχη με τα χρήματα

που θα διαθέσει (δηλαδή δεν θα πληρώσει πολλά χρήματα για μικρή βελτίωση στην επίδοση). Με άλλα λόγια και οι τρεις υπολογιστές είναι σχεδόν εξίσου καλές επιλογές και εφόσον θα αγοραστεί μόνο ένας και το διαθέσιμο ποσό το επιτρέπει, ο Υ4 είναι σαφώς η καλύτερη επιλογή.

Παράδειγμα 2:

Μια εταιρεία που ασχολείται με το λιανικό εμπόριο, εισάγει είδη τροφίμων, τα οποία διαθέτει σε αλυσίδα καταστημάτων. Οι παραγγελίες από τα καταστήματα αποστέλλονται μέσω διαδικτύου (με ηλεκτρονικό ταχυδρομείο) και στα κεντρικά γραφεία της εταιρείας τέσσερα μέλη του γραμματειακού προσωπικού θα καταχωρούν τις παραγγελίες και θα εκδίδουν τιμολόγια, τα οποία θα εκτυπώνονται και θα αποστέλλονται στην αποθήκη για εκτέλεση (η ποιότητα εκτύπωσης δεν χρειάζεται να είναι πολύ καλή, αλλά η ταχύτητα θα πρέπει να είναι τουλάχιστον 20 σελίδες ανά λεπτό). Η αποθήκευση και η εκτύπωση θα γίνεται σε εξειδικευμένο πρόγραμμα αποθήκης το οποίο ήδη αγοράστηκε (και υποστηρίζεται μόνο από λειτουργικό σύστημα Linux) και θα λειτουργεί σε ξεχωριστό κεντρικό υπολογιστή, όπου θα είναι συνδεδεμένος και ο εκτυπωτής για τουλάχιστον 3 χρόνια. Κάθε μέρα στον κεντρικό υπολογιστή θα καταχωρούνται περίπου 250 MB δεδομένων από 250 παραγγελίες. Οι υπολογιστές του γραμματειακού προσωπικού θα πρέπει να διαθέτουν οθόνες μεγέθους τουλάχιστο 20", ώστε να είναι δυνατόν ταυτόχρονα να βλέπουν με ευκολία τα μηνύματα ηλεκτρονικού ταχυδρομείου και τη φόρμα με τις παραγγελίες. Σε ορισμένες περιπτώσεις το γραμματειακό προσωπικό χρειάζεται να αναζητά πληροφορίες πελατών στον παγκόσμιο ιστό και να γράφει και να εκτυπώνει (ασπρόμαυρες) επιστολές με πρόγραμμα επεξεργασίας κειμένου (το προσωπικό εκπαιδεύτηκε στη χρήση του Microsoft Word και αυτό πρέπει να εγκατασταθεί). Η κατανάλωση ρεύματος δεν είναι σε ψηλή προτεραιότητα, αφού η εταιρεία εδράζεται σε βιομηχανική περιοχή και ωφελείται από χαμηλού κόστους ρεύμα. Χρειάζεται να προταθεί η λύση η οποία καλύπτει τις ανάγκες των χρηστών με το χαμηλότερο δυνατό κόστος.

Ανάγκες του χρήστη:

Εδώ υπάρχουν δύο κατηγορίες υπολογιστικών συστημάτων, του γραμματειακού προσωπικού και ο κεντρικός υπολογιστής, οι οποίες θα πρέπει να εξεταστούν ξεχωριστά.

Για το γραμματειακό προσωπικό:

- Χρειάζεται επιτραπέζιος υπολογιστής.
- Λογισμικό εφαρμογών που χρειάζεται: Επεξεργαστής κειμένου, φυλλομετρητής ιστού, πρόγραμμα ηλεκτρονικού ταχυδρομείου.
- Λειτουργικά συστήματα που μπορούν να χρησιμοποιηθούν: Windows.
- Οθόνη 20".
- Άλλες λειτουργίες: Σύνδεση με το Διαδίκτυο.
- Προστασία: Πρόγραμμα καταπολέμησης ιών.

Για τον κεντρικό υπολογιστή:

- Χρειάζεται επιτραπέζιος υπολογιστής.
- Όγκος δεδομένων για μόνιμη αποθήκευση: Δημιουργίες γραφικών 250 MB × 365 μέρες × 3 χρόνια, σύνολο 273750 MB, περίπου 268 GB (χρειάζεται επιπρόσθετος αποθηκευτικός χώρος για το λειτουργικό σύστημα και το λογισμικό εφαρμογών).
- Λογισμικό εφαρμογών που χρειάζεται: Εξειδικευμένο πρόγραμμα αποθήκης.
- Λειτουργικά συστήματα που μπορούν να χρησιμοποιηθούν: Linux.

- Εκτυπωτής με ταχύτητα τουλάχιστον 20 ppm.
- Άλλες λειτουργίες: Σύνδεση με δίκτυο.
- Προστασία: Προστασία από διακοπή του ηλεκτρικού ρεύματος (UPS), DVD Drive για εφεδρικά αρχεία ασφάλειας.

Επιλογή λογισμικού εφαρμογών και λειτουργικού συστήματος:

Από τις ανάγκες προκύπτει ότι χρειάζεται:

Για το γραμματειακό προσωπικό:

- Λειτουργικό σύστημα: Windows (π.χ. *Windows 7 Professional*, το οποίο έχει απαιτήσεις ΚΜΕ αρχιτεκτονικής x86 με ταχύτητα επεξεργασίας τουλάχιστον 1 GHz, μνήμη RAM με χωρητικότητα τουλάχιστον 1 GB, σκληρό δίσκο με χωρητικότητα τουλάχιστον 16 GB, €150).
- Επεξεργαστής κειμένου/πρόγραμμα ηλεκτρονικού ταχυδρομείου: Microsoft Office (Word, Outlook) 2010 Home & Student Edition, €119 (χωρίς ιδιαίτερες απαιτήσεις για το υλικό πέρα από χωρητικότητα σκληρού δίσκου περίπου 1 GB).
- Φυλλομετρητής ιστού: Internet Explorer ή Firefox ή Opera ή Chrome ή Safari (όλοι χωρίς κόστος ή ιδιαίτερες απαιτήσεις).
- Πρόγραμμα καταπολέμησης ιών: Kaspersky Pure 3.0 (€75).

Για τον κεντρικό υπολογιστή:

- Λειτουργικό σύστημα: Linux (π.χ. *Ubuntu Server 14LTS*, το οποίο έχει απαιτήσεις ΚΜΕ αρχιτεκτονικής x86 με ταχύτητα επεξεργασίας τουλάχιστον 1 GHz, μνήμη RAM με χωρητικότητα τουλάχιστον 512 MB, σκληρό δίσκο με χωρητικότητα τουλάχιστον 5 GB).
- Εξειδικευμένο πρόγραμμα αποθήκης (ήδη αγοράστηκε, δεν έχει συγκεκριμένες απαιτήσεις).

Από τις επιλογές που έχουν διερευνηθεί προτείνονται όσες είναι υπογραμμισμένες. Ως πρόγραμμα καταπολέμησης ιών θα μπορούσε να χρησιμοποιηθεί το δωρεάν Microsoft Security Essentials, αλλά προτείνεται το Kaspersky, το οποίο παρέχει μεγαλύτερη ασφάλεια.

Επιλογή υλικού:

Μελετώντας τα πιο πάνω προκύπτει ότι ο υπολογιστής που απαιτείται για το γραμματειακό προσωπικό πρέπει να είναι επιτραπέζιος, με ΚΜΕ αρχιτεκτονικής x86, με ταχύτητα επεξεργασίας τουλάχιστον 1 GHz, μνήμη RAM με χωρητικότητα τουλάχιστον 1 GB, σκληρό δίσκο με χωρητικότητα τουλάχιστον 16 GB και οθόνη με μέγεθος 20".

Ο κεντρικός υπολογιστής πρέπει να είναι επιτραπέζιος, με ΚΜΕ αρχιτεκτονικής x86, με ταχύτητα επεξεργασίας τουλάχιστον 1 GHz, μνήμη RAM με χωρητικότητα τουλάχιστον 512 GB, σκληρό δίσκο με χωρητικότητα τουλάχιστον 273 GB, εφοδιασμένο με DVD Drive, με εκτυπωτή ταχύτητας 20ppm και με UPS.

Έστω ότι αναζητήσαμε από προμηθευτές/καταστήματα πώλησης και πήραμε τις πιο κάτω εισηγήσεις (θεωρούμε ότι όλοι οι υπολογιστές είναι επιτραπέζιοι και διαθέτουν CD/DVD και κάρτα δικτύου):

HY1: CPU 1,60 GHz, RAM 4 GB, HDD 250 GB, οθόνη 17", τιμή €370.

HY2: CPU 1,60 GHz, RAM 4 GB, HDD 250 GB, οθόνη 20", τιμή €420.

HY3: CPU 2,20 GHz, RAM 4 GB, HDD 250 GB, οθόνη 20", τιμή €550.

HY4: CPU 2,20 GHz, RAM 4 GB, HDD 500 GB, οθόνη 17", τιμή €500.

HY5: CPU 2,50 GHz, RAM 8 GB, HDD 1 TB, οθόνη 20", τιμή €690

Για το γραμματειακό προσωπικό, ο ΗΥ1 και ο ΗΥ4 απορρίπτονται, διότι δεν έχουν αρκετά μεγάλη οθόνη. Επειδή στόχος είναι η ικανοποίηση των αναγκών με το χαμηλότερο κόστος, δεν χρειάζεται λεπτομερής αξιολόγηση και επιλέγουμε τον ΗΥ2.

Για τον κεντρικό υπολογιστή, οι ΗΥ1, ΗΥ2 και ΗΥ3 απορρίπτονται, διότι δεν έχουν αρκετή χωρητικότητα σκληρού δίσκου. Επομένως επιλέγουμε τον ΗΥ4, ο οποίος έχει χαμηλότερο κόστος. Εδώ εάν κάνουμε την ανάλυση δίνοντας διπλάσια βαρύτητα στην ταχύτητα της ΚΜΕ και στην ποσότητα της μνήμης RAM (για εξυπηρέτηση των χρηστών), παρά στον σκληρό δίσκο έχουμε τα αποτελέσματα που φαίνονται στον πιο κάτω πίνακα:

Υπολογιστής	Χαρακτηριστικά			Κόστος	Επιδόσεις			Δείκτης Επίδοσης	Επίδοση/Κόστος
	ΚΜΕ	RAM	Σκληρός Δίσκος		ΚΜΕ	RAM	Σκληρός Δίσκος		
					40	40	20	100	
ΗΥ4	2,2	4	500	500	0,88	0,50	0,50	65,20	0,13
ΗΥ5	2,5	8	1000	690	1,00	1,00	1,00	100,00	0,14
	2,5	8	1000						

Παρά το γεγονός ότι ο ΗΥ5 είναι σαφώς καλύτερος από τον ΗΥ4, η επίδοση ανά κόστος είναι παρόμοια, δείχνοντας ότι υπάρχει αντιστοιχία της επίδοσης με το κόστος.

Στην περίπτωση αυτή θα χρειαστούμε και έναν εκτυπωτή. Έστω ότι πήραμε από τους προμηθευτές τις πιο κάτω προδιαγραφές εκτυπωτών:

E1: Ψεκασμού μελάνης, έγχρωμος, 1200dpi, 20ppm, € 0,142 ανά σελίδα, τιμή €80.

E2: Λέιζερ, ασπρόμαυρος, 300dpi, 22ppm, € 0,022 ανά σελίδα, τιμή €200.

E3: Λέιζερ, ασπρόμαυρος, 600dpi, 24ppm, € 0,024 ανά σελίδα, τιμή €350.

Ο Ε1 φαίνεται φθηνότερος, αλλά εάν υπολογίσουμε το κόστος για εκτύπωση 250 σελίδων παραγγελιών για κάθε μέρα (περίπου 273750 σελίδες σε 3 χρόνια), τότε το κόστος του Ε1 είναι € 38953, πολύ ψηλότερο από τον Ε2 με € 6223 και τον Ε3 με € 6920. Επομένως επιλέγουμε τον Ε2 ως τη φθηνότερη επιλογή.

Παρόμοια διαδικασία κανονικά πρέπει να ακολουθηθεί και για το UPS (στοιχίζει περίπου € 60–100).

Βασικές Έννοιες**Υπολογιστικές ανάγκες του χρήστη:**

Λεπτομερής περιγραφή των αναγκών που έχει ο χρήστης (φορητότητα/αυτονομία, όγκος δεδομένων για μόνιμη αποθήκευση, λογισμικό εφαρμογών, λειτουργικό σύστημα, περιφερειακές συσκευές και προδιαγραφές τους, κατανάλωση ρεύματος, προστασία). Χρησιμοποιούνται για να καθοριστούν οι απαιτήσεις του λογισμικού και οι προδιαγραφές του υλικού.

Απαιτήσεις λογισμικού εφαρμογών:

Οι ελάχιστες προδιαγραφές του υλικού και του λειτουργικού συστήματος που απαιτούνται για την ομαλή λειτουργία του λογισμικού εφαρμογών.

Απαιτήσεις λειτουργικού συστήματος:

Οι ελάχιστες προδιαγραφές του υλικού που απαιτούνται για την ομαλή λειτουργία του λογισμικού εφαρμογών.

Προδιαγραφές υλικού:

Τα χαρακτηριστικά του υλικού που απαιτούνται ώστε να καλύπτονται οι απαιτήσεις του λογισμικού και κατά συνέπεια οι ανάγκες του χρήστη.

Επίδοση:

Μέτρηση που μας βοηθά να συγκρίνουμε εξαρτήματα, περιφερειακές συσκευές ή υπολογιστές με βάση κάποιο χαρακτηριστικό τους ή χαρακτηριστικά τους. Για ένα συγκεκριμένο χαρακτηριστικό η επίδοση υπολογίζεται διαιρώντας για κάθε επιλογή το συγκεκριμένο χαρακτηριστικό με την επιλογή που θεωρείται η καλύτερη.

Δείκτης Επίδοσης:

Συνολική μέτρηση της επίδοσης που υπολογίζεται αφού δώσουμε μονάδες στο κάθε χαρακτηριστικό ανάλογα με τη σημασία που έχει (σύνολο 100 μονάδες), στη συνέχεια πολλαπλασιάσουμε την επίδοση του κάθε χαρακτηριστικού με τις μονάδες που δόθηκαν και αθροίσουμε για να έχουμε το σύνολο για κάθε επιλογή.

Επίδοση ανά κόστος:

Δείκτης που υπολογίζεται διαιρώντας τον Δείκτη Επίδοσης με το κόστος της κάθε επιλογής. Μας βοηθά να συγκρίνουμε πόσο κοστίζει η επίδοση. Για συσκευές με αναλώσιμα, πρέπει να προστεθεί το κόστος λειτουργίας και η τιμή, για να υπολογιστεί το κόστος.

Πηγές

1. Προδιαγραφές Λειτουργικού συστήματος Windows 7. Ανακτήθηκε στις 04.07.2013.
<http://windows.microsoft.com/en-us/windows7/products/system-requirements>
2. Προδιαγραφές σχεδιαστικού προγράμματος AutoCAD 2013. Ανακτήθηκε στις 04.07.2013.
<http://docs.autodesk.com/ACD/2013/ENU/System%20Requirements/index.html>
3. Προδιαγραφές παιχνιδιού Η/Υ SimCity. Ανακτήθηκε στις 04.07.2013.
<https://help.ea.com/article/simcity-system-requirements>

ΕΝΟΤΗΤΑ Γ6 Βάσεις Δεδομένων και Ανάπτυξη Πληροφοριακών Συστημάτων

Για την προετοιμασία και τη συγγραφή του υλικού αυτής της ενότητας εργάστηκαν οι ακόλουθοι/θες Καθηγητές/τριες Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών που παρακολούθησαν το πρόγραμμα επιμόρφωσης μάχιμων εκπαιδευτικών για το Νέο Αναλυτικό Πρόγραμμα:

Αντωνίου Κωνσταντίνος
Αχιλλέως Χριστάκης
Ευθυμιάδου Έλενα
Ζεβεδαίος Λίνος
Θρασυβούλου Έλενα
Ιωάννου Ιωάννης (Σύμβουλος)
Καλλένος Μιχάλης
Καμιναρίδου Χριστίνα
Κλόκκος Πόλυς
Κόκκινος Κλεάνθης
Κουτούφαρη Αργυρώ
Κυπριανού Χρίστος
Κωνσταντίνου Μελίνα
Μάγκουλα Πόλα
Μιχαήλ Παναγιώτης
Μουμτζή Μαριάννα
Νικολάου Νεόφυτος
Ξενοφώντος Γιώργος

Ξενοφώντος Ξένιος (Σύμβουλος)
Παναγή Παύλος
Παναγίδου-Βέρελ Ιφιγένεια
Παπαχριστοδούλου Χρυστάλλα
Παυλίδου Αλεξία
Παύλου Γεωργία
Πρωτοπαπά Μαρία
Σιακαλλής Χαρίλαος
Τζιώρτας Χρήστος
Φυλακτού Δημήτρης
Φωτάκος Θωμάς
Χατζηγιάννη Αγγέλα
Χαριλάου Μιλτιάδης (Σύμβουλος)
Χατζηκωνσταντή Ανδριανή
Χατζηπαντελής Δημήτρης
Χατζηπολυκάρπου Ιωάννης
Χρυσοστόμου Χρύσης

Γ6.1 Εισαγωγή στις Βάσεις Δεδομένων

Τι θα μάθουμε σήμερα:

- Να ορίζουμε τι είναι μια βάση δεδομένων
- Να διακρίνουμε τα είδη των βάσεων δεδομένων (έντυπη/ηλεκτρονική)
- Να διακρίνουμε και να αναφέρουμε τα πλεονεκτήματα και τα μειονεκτήματα μιας βάσης δεδομένων
- Να αναφέρουμε βάσεις δεδομένων από την καθημερινή ζωή
- Να αναφέρουμε παραδείγματα προγραμμάτων διαχείρισης μηχανογραφημένων βάσεων δεδομένων.

1. Βάση Δεδομένων (Database)

Μια βάση δεδομένων είναι ένα σύνολο από πληροφορίες που σχετίζονται μεταξύ τους και οι οποίες έχουν οργανωθεί με τέτοιο τρόπο ώστε η χρήση τους να είναι γρήγορη και αποτελεσματική.

Καθημερινά, σε πολλές από τις δραστηριότητές μας χρειαζόμαστε πληροφορίες. Γι' αυτό τον λόγο οι βάσεις δεδομένων βρίσκονται παντού: Στο σπίτι, στην εργασία, στα σχολεία, στα καταστήματα, στις κυβερνητικές υπηρεσίες κ.λπ.

Μια **βάση δεδομένων** μπορεί να υπάρχει σε:

- **έντυπη μορφή** (π.χ. ο τηλεφωνικός κατάλογος) και σε
- **ηλεκτρονική μορφή** (π.χ. οι επαφές μας στο κινητό μας τηλέφωνο).

Στην ενότητα αυτή θα μελετήσουμε τις ηλεκτρονικές ή διαφορετικά τις μηχανογραφημένες βάσεις δεδομένων.

2. Ηλεκτρονικές (μηχανογραφημένες) βάσεις δεδομένων

Η δημιουργία και διαχείριση μηχανογραφημένων βάσεων δεδομένων ήταν ένας από τους κύριους λόγους της χρήσης υπολογιστών από μεγάλους οργανισμούς, από τη δεκαετία του 1960. Αρχικά είχαν προταθεί διάφοροι τρόποι οργάνωσης και διαχείρισης των δεδομένων, βασισμένοι σε διαφορετικά θεωρητικά «μοντέλα».

Από τη δεκαετία του 1970 και μετά επικράτησε η άποψη ότι ο πιο αποτελεσματικός τρόπος οργάνωσης και διαχείρισης των μηχανογραφημένων βάσεων δεδομένων είναι μέσα από ειδικό λογισμικό το οποίο ονομάζεται **Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ)** - **Database Management System (DBMS)**. Το λογισμικό αυτό είναι κατασκευασμένο έτσι ώστε να μπορεί να διαχειρίζεται διαφορετικές βάσεις δεδομένων οι οποίες έχουν σχεδιαστεί και οργανωθεί ακολουθώντας το θεωρητικό «μοντέλο» που προτείνει το ΣΔΒΔ.

Κάποια από τα πιο δημοφιλή συστήματα διαχείρισης μηχανογραφημένων βάσεων δεδομένων είναι και τα ακόλουθα: MySQL, PostgreSQL, SQLite, Microsoft SQL Server, **Microsoft Access**, Oracle, SAP, dBASE, FoxPro, IBM DB2 και FilemakerPro.

Τα επιθυμητά χαρακτηριστικά ενός **Συστήματος Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ)** είναι τα ακόλουθα:

- μείωση πλεοναζόντων δεδομένων
- ευκολία στην ανάπτυξη νέων εφαρμογών
- μηχανισμοί ασφαλείας.

Τα σύγχρονα (ΣΔΒΔ) έχουν **εργαλεία** για:

- δημιουργία πινάκων
- εισαγωγή και αλλαγή δεδομένων στους πίνακες μέσα από φόρμες
- άντληση πληροφοριών με τη δημιουργία ερωτημάτων
- εκτύπωση πληροφοριών με τη δημιουργία εκθέσεων

Την ίδια εποχή εξετάστηκαν διάφοροι τρόποι σχεδιασμού και οργάνωσης των δεδομένων και επικράτησε το μοντέλο της **Σχεσιακής Βάσης Δεδομένων (Relational Database)**, το οποίο προτείνει ότι μια βάση δεδομένων είναι οργανωμένη σε πίνακες οι οποίοι σχετίζονται μεταξύ τους με βάση κάποια από τα δεδομένα που περιέχουν.

Έτσι, μέχρι σήμερα έχουν επικρατήσει τα Συστήματα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (Relational Database Management System, RDBMS), όπως είναι η *Microsoft Access*.

3. Πλεονεκτήματα των μηχανογραφημένων βάσεων δεδομένων

(1) **Οι πληροφορίες καταχωρούνται και εντοπίζονται πολύ γρήγορα:** Σε μια μηχανογραφημένη βάση δεδομένων οι χρήστες μπορούν να καταχωρούν και να εντοπίζουν πολύ πιο γρήγορα και αποτελεσματικά τις πληροφορίες που θέλουν, ανεξάρτητα από τον τρόπο και τη σειρά καταχώρισης. Για παράδειγμα, μπορούμε να βρούμε πολύ πιο γρήγορα τον αριθμό τηλεφώνου ενός φίλου μας εάν ψάξουμε στις επαφές του κινητού μας παρά στον τηλεφωνικό κατάλογο.

(2) **Ο χειρισμός των πληροφοριών μπορεί να γίνει με μεγαλύτερη ευελιξία:** Σε μια μηχανογραφημένη βάση δεδομένων υπάρχει η δυνατότητα οι πληροφορίες να εξετάζονται με βάση συγκεκριμένα κριτήρια και να ετοιμάζονται λίστες με τις πληροφορίες που πληρούν τα κριτήρια αυτά.

Η γραμματεία ενός σχολείου για παράδειγμα θα μπορούσε σε μια μηχανογραφημένη βάση δεδομένων να ετοιμάσει με μεγάλη ευκολία έναν κατάλογο με τα στοιχεία των μαθητών (*π.χ. όνομα, επίθετο, διεύθυνση, αριθμός δικαιολογημένων απουσιών, αριθμός αδικαιολόγητων απουσιών, κ.λπ.*), που πληρούν κάποια κριτήρια (*π.χ. οι αδικαιολόγητες απουσίες τους υπερβαίνουν κάποιο αριθμό*). Για να ετοιμαστεί ο ίδιος κατάλογος σε μια έντυπη βάση δεδομένων θα ήταν πολύ πιο χρονοβόρο και επιπλέον θα υπήρχε αυξημένη πιθανότητα να γίνει λάθος.

(3) **Αποφεύγεται η επανάληψη δεδομένων:** Σε μια (σχεσιακή) βάση δεδομένων δεν καταχωρούνται τα ίδια δεδομένα περισσότερες από μια φορές. Για παράδειγμα, το ονοματεπώνυμο και η διεύθυνση ενός πελάτη μιας ασφαλιστικής εταιρείας καταχωρούνται ως μέρος των στοιχείων του πελάτη. Εάν ο πελάτης έχει κάνει ασφάλεια ζωής και ασφάλεια ιδιωτικού οχήματος, δεν θα καταχωρηθούν το όνομα και διεύθυνσή του ξεχωριστά μαζί με τα στοιχεία της κάθε ασφάλειας. Έτσι, εάν μελλοντικά δηλώσει ότι έχει αλλάξει διεύθυνση κατοικίας, αυτόματα θα ισχύει η ίδια διεύθυνση για όλες τις ασφάλειες του πελάτη και δεν θα υπάρχει η πιθανότητα να αποστέλλονται έγγραφα για μια ασφάλεια σε λανθασμένη διεύθυνση.

(4) **Υπάρχει έλεγχος στη καταχώριση των δεδομένων:** Κατά την καταχώριση των δεδομένων σε μια βάση δεδομένων μπορούν να εξασκούνται λογικοί έλεγχοι πάνω σε αυτά. Με αυτό τον τρόπο αποτρέπεται η εισαγωγή παράλογων και λανθασμένων δεδομένων, όπως για παράδειγμα μελλοντικές ημερομηνίες ως ημερομηνίες γεννήσεως, κείμενο αντί για αριθμό, αρνητικοί αριθμοί για ποσότητες προϊόντων, αριθμοί τηλεφώνων με πιο λίγα ή πιο πολλά ψηφία από ό,τι πρέπει, κ.λπ.

(5) **Υπάρχει δυνατότητα πρόσβασης (προσπέλασης) στα δεδομένα μιας βάσης δεδομένων από πολλούς χρήστες ταυτόχρονα:** Με τη χρήση των δικτύων μπορούν πολλοί χρήστες ταυτόχρονα από διάφορους χώρους να έχουν πρόσβαση στα δεδομένα μιας βάσης δεδομένων.

Π.χ. η βάση δεδομένων μιας τράπεζας, όπου οι υπάλληλοι από τα διαφορετικά καταστήματα της τράπεζας έχουν πρόσβαση στους ίδιους λογαριασμούς ή η βάση δεδομένων μιας αεροπορικής εταιρείας, όπου οποιοσδήποτε μπορεί μέσω του διαδικτύου να ψάξει για πτήσεις και να κάνει κρατήσεις.

4. Μειονέκτημα των μηχανογραφημένων βάσεων δεδομένων

Το κύριο μειονέκτημα των μηχανογραφημένων βάσεων δεδομένων είναι ο **χρόνος** και το **κόστος** που απαιτείται για να σχεδιαστεί, να αναπτυχθεί και να συντηρηθεί μια βάση δεδομένων. Όσο πιο πολλά είναι τα δεδομένα και όσο πιο πολύπλοκη ή συχνή η επεξεργασία τους, τόσο τα πλεονεκτήματα των μηχανογραφημένων βάσεων δεδομένων οδηγούν στην επιλογή τους αντί των έντυπων βάσεων δεδομένων, σε βαθμό που πολλές φορές μια έντυπη βάση δεδομένων να εξαγεται από μια μηχανογραφημένη (π.χ. ο τηλεφωνικός κατάλογος).

5. Βάσεις Δεδομένων από την καθημερινή μας ζωή

Μερικά **παραδείγματα** βάσεων δεδομένων από την καθημερινή μας ζωή είναι τα εξής:

- Βάσεις Δεδομένων Σχολείων
- Βάσεις Δεδομένων Τραπεζών
- Βάσεις Δεδομένων για διαχείριση της αποθήκης σε Υπεραγορές
- Ιατρικά Συστήματα Διαχείρισης Ασθενών σε Νοσοκομεία
- Συστήματα Κρατήσεων Αεροπορικών Εισιτηρίων
- Βάσεις Δεδομένων Τηλεφωνίας (*Μηχανογραφημένοι Τηλεφωνικοί Κατάλογοι*).

Βασικές Έννοιες***Βάση Δεδομένων
(Database):***

Είναι ένα σύνολο από πληροφορίες που σχετίζονται μεταξύ τους και έχουν οργανωθεί έτσι ώστε η χρήση τους να είναι γρήγορη και αποτελεσματική.

***Έντυπες
Βάσεις Δεδομένων:***

Είναι βάσεις δεδομένων που υπάρχουν σε έντυπη μορφή.

***Ηλεκτρονικές Βάσεις
Δεδομένων:***

Είναι οι μηχανογραφημένες βάσεις δεδομένων, δηλαδή αυτές που βρίσκονται εγκατεστημένες σε ηλεκτρονικούς υπολογιστές.

***Σύστημα Διαχείρισης
Βάσεων Δεδομένων
(ΣΔΒΔ) (Database
Management System,
DBMS):***

Είναι λογισμικό το οποίο είναι κατασκευασμένο έτσι ώστε να μπορεί να διαχειρίζεται διαφορετικές βάσεις δεδομένων οι οποίες έχουν σχεδιαστεί και οργανωθεί με συγκεκριμένο τρόπο.

***Σχεσιακή Βάση Δεδομένων
(Relational Database):***

Τρόπος σχεδιασμού και οργάνωσης των δεδομένων, το οποίο προτείνει ότι μια βάση δεδομένων πρέπει να είναι οργανωμένη σε πίνακες οι οποίοι σχετίζονται μεταξύ τους με βάση κάποια από τα δεδομένα που περιέχουν.

***Σύστημα Διαχείρισης
Σχεσιακών Βάσεων
Δεδομένων (Relational
Database Management
System, RDBMS):***

Είναι λογισμικό το οποίο είναι κατασκευασμένο έτσι ώστε να μπορεί να διαχειρίζεται βάσεις δεδομένων οι οποίες έχουν σχεδιαστεί και οργανωθεί ως Σχεσιακές Βάσεις Δεδομένων. Παραδείγματα ΣΔΒΔ είναι και τα ακόλουθα: MySQL, PostgreSQL, Oracle, SQLite, Microsoft SQL Server, Microsoft Access και SAP.

Γ6.2

Βασικά Δομικά Συστατικά Στοιχεία και Κύρια Αντικείμενα μιας Βάσης Δεδομένων

Τι θα μάθουμε σήμερα:

- ♦ Να αναγνωρίζουμε και να περιγράφουμε πώς είναι οργανωμένη μια βάση δεδομένων όσον αφορά τα βασικά δομικά συστατικά στοιχεία της (*πίνακες, εγγραφές και πεδία*)
- ♦ Να εντοπίζουμε και να περιγράφουμε τις σχέσεις που διέπουν τους πίνακες σε μια βάση δεδομένων και τα πλεονεκτήματα που προκύπτουν από αυτές
- ♦ Να δημιουργούμε και να αποθηκεύουμε μια βάση δεδομένων σε συγκεκριμένη θέση σε μονάδα δίσκου
- ♦ Να ανοίγουμε μια βάση δεδομένων από συγκεκριμένη θέση από μονάδα δίσκου
- ♦ Να αναγνωρίζουμε τα κύρια αντικείμενα μιας βάσης δεδομένων.

1. Βασικά Δομικά Συστατικά Στοιχεία μιας Βάσης Δεδομένων

1.1 Πίνακας (Table)

Το πιο βασικό δομικό συστατικό στοιχείο μιας βάσης δεδομένων είναι ο **πίνακας**. Η οργάνωση και η αποθήκευση των πληροφοριών (*δεδομένων*) μιας βάσης δεδομένων γίνεται με τη χρήση πινάκων.

Μια βάση δεδομένων μπορεί να αποτελείται από πολλούς πίνακες και σε κάθε πίνακα αποθηκεύονται πληροφορίες σχετικά με ένα συγκεκριμένο θέμα.

Για παράδειγμα, ένας πίνακας μπορεί να περιέχει όλα τα στοιχεία των μελών ενός Κέντρου Νεότητας, ένας άλλος πίνακας να περιέχει όλες τις δραστηριότητες που προσφέρει το Κέντρο Νεότητας κ.λπ.

Ο κάθε πίνακας έχει το δικό του **όνομα** και τα δεδομένα του είναι οργανωμένα σε γραμμές (*εγγραφές*) και στήλες (*πεδία*).

YouthMembers						
YouthMem	Firstname	Surname	Sex	Birthdate	Address	Town
1001	Αβράαμ	Κυριάκου	A	12/8/1999	Καντάρας 19	Λεμεσός
1002	Αχυλλέας	Αλέξη	A	13/8/1999	Καντάρας 15	Λεμεσός
1004	Αλέξανδρος	Νικόλαου	A	15/8/1999	Αμμοχώστου 37	Λεμεσός
1005	Αλέξανδρος	Αντώνιου	A	16/8/1999	Αμμοχώστου 13	Λεμεσός
1006	Αλεξάνδρα	Αθανάσιου	Θ	17/8/1999	Αμμοχώστου 11	Λεμεσός
1007	Αλέξης	Παντελή	A	18/8/1999	Μόρφου 26	Λεμεσός
1008	Αναστασία	Δέμη	Θ	19/8/1999	Μόρφου 16	Λεμεσός
1009	Άντρεα	Κυριάκου	Θ	20/8/1999	Κυρηνείας 7	Λεμεσός
1010	Άντρεα	Γεώργιου	Θ	21/8/1999	Ριζοκαρπάσου 912	Λεμεσός
1011	Άντρεα	Στυλιανού	Θ	22/8/1999	Ριζοκαρπάσου 7	Λεμεσός
1013	Ανδρέας	Κόκου	A	24/8/1999	Ριζοκαρπάσου 15	Λεμεσός
1014	Ανδρέας	Βάσου	A	25/8/1999	Αμμοχώστου 27	Λεμεσός
1015	Ανδρέας	Εφραίμ	A	26/8/1999	Καντάρας 27	Λεμεσός
1016	Ανδρέας	Σωκράτη	A	27/8/1999	Ριζοκαρπάσου 16	Λεμεσός

Ο Πίνακας:
YouthMembers

1.2 Εγγραφή (Record)

Εγγραφή είναι η κάθε **γραμμή** ενός πίνακα μιας βάσης δεδομένων όπου περιέχει όλα τα δεδομένα που περιγράφουν μια συγκεκριμένη καταχώριση του πίνακα και αποτελεί μια ενότητα στοιχείων.

YouthMem	Firstname	Surname	Sex	Birthdate	Address	Town
1001	Αβραάμ	Κυριάκου	A	12/8/1999	Καντάρας 19	Λεμεσός
1002	Αχλλέας	Αλέξη	A	13/8/1999	Καντάρας 15	Λεμεσός
1004	Αλέξανδρος	Νικόλαου	A	15/8/1999	Αμμοχώστου 37	Λεμεσός
1005	Αλέξανδρος	Αντώνιου	A	16/8/1999	Αμμοχώστου 13	Λεμεσός
1006	Αλεξάνδρα	Αθανάσιου	Θ	17/8/1999	Αμμοχώστου 11	Λεμεσός
1007	Αλέξης	Παντελή	A	18/8/1999	Μόρφου 26	Λεμεσός
1008	Αναστασία	Δέμη	Θ	19/8/1999	Μόρφου 16	Λεμεσός
1009	Άντρεα	Κυριάκου	Θ	20/8/1999	Κυρηνείας 7	Λεμεσός

εγγραφή

1.3 Πεδίο (Field)

Πεδίο είναι η κάθε **στήλη** ενός πίνακα μιας βάσης δεδομένων. Σε ένα πεδίο καταχωρούνται οι τιμές για ένα συγκεκριμένο χαρακτηριστικό των εγγραφών του πίνακα. Τα δεδομένα που βρίσκονται στο κάθε πεδίο είναι πάντα του ίδιου τύπου. Κάθε πεδίο έχει πάντα διαφορετικό όνομα.

Για παράδειγμα, σε έναν πίνακα πελατών, **ένα πεδίο** θα είναι τα ονόματα των πελατών, **άλλο πεδίο** τα επώνυμά τους, **άλλο πεδίο** η διεύθυνσή τους και **άλλο πεδίο** τα τηλέφωνα τους.

YouthMem	Firstname	Surname	Sex	Birthdate	Address	Town
1001	Αβραάμ	Κυριάκου	A	12/8/1999	Καντάρας 19	Λεμεσός
1002	Αχλλέας	Αλέξη	A	13/8/1999	Καντάρας 15	Λεμεσός
1004	Αλέξανδρος	Νικόλαου	A	15/8/1999	Αμμοχώστου 37	Λεμεσός
1005	Αλέξανδρος	Αντώνιου	A	16/8/1999	Αμμοχώστου 13	Λεμεσός
1006	Αλεξάνδρα	Αθανάσιου	Θ	17/8/1999	Αμμοχώστου 11	Λεμεσός
1007	Αλέξης	Παντελή	A	18/8/1999	Μόρφου 26	Λεμεσός
1008	Αναστασία	Δέμη	Θ	19/8/1999	Μόρφου 16	Λεμεσός
1009	Άντρεα	Κυριάκου	Θ	20/8/1999	Κυρηνείας 7	Λεμεσός

πεδίο

2. Ειδικές Σχέσεις μεταξύ πινάκων (Relationships)

Όπως έχουμε δει πιο πάνω, μια βάση δεδομένων μπορεί να αποτελείται από περισσότερους από έναν πίνακα και κάθε πίνακας αποθηκεύει δεδομένα για ένα συγκεκριμένο θέμα.

Οι διάφοροι πίνακες της βάσης δεδομένων μπορούν να σχετίζονται μεταξύ τους με μια ειδική σχέση. Αυτή η ειδική σχέση, η οποία ονομάζεται **Relationship**, είναι μια λογική σύνδεση μεταξύ δύο πινάκων που έχουν τουλάχιστον ένα κοινό πεδίο (δηλαδή ένα πεδίο του οποίου τα δεδομένα είναι κοινά για κάποιες εγγραφές στον ένα πίνακα και για κάποιες εγγραφές στον άλλο πίνακα).

Η δημιουργία αυτής της σχέσης ανάμεσα σε δύο πίνακες μιας βάσης δεδομένων, μας επιτρέπει να αντλήσουμε στοιχεία και πληροφορίες που ανήκουν και στους δύο πίνακες.

Αυτό έχει ως αποτέλεσμα να έχουμε τα πιο κάτω **πλεονεκτήματα**:

- Δεν χρειάζεται να καταχωρίσουμε τα ίδια δεδομένα πολλές φορές
- Γίνεται ευκολότερη η διαχείριση των δεδομένων (*ενημέρωση, διαγραφή κ.λπ.*)
- Μικρότερος όγκος δεδομένων, πιο γρήγορη αναζήτηση
- Μειώνεται το περιθώριο σφαλμάτων
- Περισσότερη σαφήνεια στην οργάνωση των δεδομένων.

Τα **είδη** των ειδικών σχέσεων που υπάρχουν μεταξύ των πινάκων είναι:

2.1 Ειδική Σχέση: Ένα-Προς-Ένα (One-To-One, 1-1)

Κάθε εγγραφή από τον ΠΡΩΤΟ πίνακα σχετίζεται με μόνο μία εγγραφή από τον ΔΕΥΤΕΡΟ πίνακα, αλλά και αντίστροφα.

2.2 Ειδική Σχέση: Ένα-Προς-Πολλά (One-To-Many, 1-∞)

Κάθε εγγραφή από τον ΠΡΩΤΟ πίνακα σχετίζεται με πολλές εγγραφές από τον ΔΕΥΤΕΡΟ πίνακα, αλλά κάθε εγγραφή από το ΔΕΥΤΕΡΟ πίνακα, σχετίζεται με μόνο μία εγγραφή από τον ΠΡΩΤΟ πίνακα.

Ένα **παράδειγμα** μιας ειδικής σχέσης Ένα-προς-Πολλά, είναι στη βάση δεδομένων του Κέντρου Νεότητας (**YouthCentre.accdb**). Εδώ υπάρχουν δύο πίνακες:

- Ο πρώτος πίνακας που έχει το όνομα **Activities** περιέχει την αθλητική δραστηριότητα με την οποία ασχολείται το κάθε μέλος.
- Ο δεύτερος πίνακας που έχει το όνομα **YouthMembers** περιέχει τα ονόματα των μελών του Κέντρου Νεότητας.
- Οι δύο πίνακες συνδέονται μεταξύ τους με την ειδική σχέση Ένα-Προς-Πολλά. Δηλαδή, κάθε μέλος από τον πίνακα **YouthMembers** λαμβάνει μέρος μόνο σε μία δραστηριότητα του πίνακα **Activities**, ενώ σε κάθε δραστηριότητα υπάρχουν πολλά μέλη.

YouthMembers		Activities						
ActivityCode		Description		Registration		Click to Add		
1 Ποδόσφαιρο		100,00 €						
YouthMem	Firstname	Surname	Sex	Birthdate	Address	Town	TelNumber	
1001	Αβράμ	Κυριάκου	A	12/8/1999	Καντάρας 19	Λεμεσός	99651298	
1002	Αχιλλέας	Αλέξη	A	13/8/1999	Καντάρας 15	Λεμεσός	99274730	
1004	Αλέξανδρος	Νικόλαου	A	15/8/1999	Αμμοχώστου 37	Λεμεσός	99434069	
1127	Λένα	Γκία	Θ	4/3/1998	Ριζοκαρπάσου 901	Λάρνακα	99827728	
1128	Λιούσι	Ρομπέρτ	Θ	5/3/1998	Καντάρας 10	Λάρνακα	99804020	
1215	Ζήνα	Ερρίκου	Θ	19/10/2002	Ριζοκαρπάσου 35	Λεμεσός	99974523	
2 Καλαθόσφαιρα		80,00 €						
YouthMem	Firstname	Surname	Sex	Birthdate	Address	Town	TelNumber	
1005	Αλέξανδρος	Αντώνιου	A	16/8/1999	Αμμοχώστου 13	Λεμεσός	99174661	
1027	Άννα	Γιώργου	Θ	10/2/2000	Ριζοκαρπάσου 13	Λεμεσός	99262876	
1045	Χαραλαμπία	Τάσου	Θ	28/2/2000	Ριζοκαρπάσου 911	Λευκωσία	99127935	
1060	Χρυστάλλα	Νικόλαου	Θ	14/3/2000	Καντάρας 35	Λευκωσία	99157570	
1185	Ραφαέλλα	Μιχαήλ	Θ	19/9/2002	Μόρφου 40	Λευκωσία	99210223	
3 Πετόσφαιρα		75,00 €						
YouthMem	Firstname	Surname	Sex	Birthdate	Address	Town	TelNumber	
1006	Αλεξάνδρα	Αθανάσιου	Θ	17/8/1999	Αμμοχώστου 11	Λεμεσός	99986377	
1010	Άντρεα	Γεωργίου	Θ	21/8/1999	Ριζοκαρπάσου 912	Λεμεσός	99222077	
1028	Άννα	Γεννάδιου	Θ	11/2/2000	Κυρηνείας 31	Λεμεσός	99645647	
1181	Πετρίνα	Κυριάκου	Θ	15/9/2002	Αμμοχώστου 21	Λευκωσία	99156880	
1182	Προκόπης	Σάββα	A	16/9/2002	Κυρηνείας 18	Λευκωσία	99927797	

3. Κύρια Αντικείμενα Βάσης Δεδομένων (Objects)

3.1 Πίνακας (Table)

Όπως έχουμε ήδη αναφέρει, ένας **πίνακας** είναι μια συλλογή δεδομένων που περιγράφουν ομοειδή αντικείμενα.

3.2 Φόρμα (Form)

Οι **φόρμες** είναι ένας εύκολος τρόπος για την καταχώριση και την επεξεργασία των δεδομένων στους πίνακες. Μια φόρμα είναι άμεσα συνδεδεμένη με έναν ή περισσότερους πίνακες και εμφανίζει συγκεντρωμένα τα στοιχεία που αφορούν την κάθε εγγραφή. Τα δεδομένα τα παίρνει από έναν πίνακα ή τα αποθηκεύει σε έναν πίνακα.

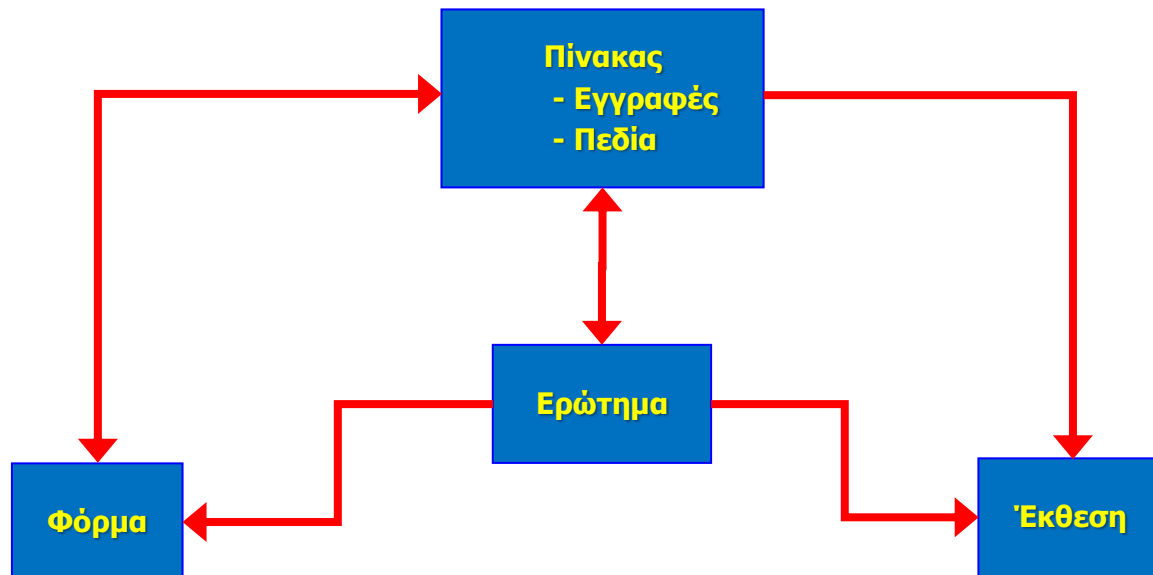
3.3 Ερωτήματα (Queries)

Με τη βοήθεια των **ερωτημάτων** και χρησιμοποιώντας τα κατάλληλα κριτήρια, μπορούμε να εντοπίζουμε και να παρουσιάζουμε τα δεδομένα που επιθυμούμε από μια βάση δεδομένων.

3.4 Εκθέσεις (Reports)

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.

4. Η Δομή μιας Βάσης Δεδομένων



5. Εφαρμογή Βάσεων Δεδομένων (Microsoft Access)

Όπως έχει ήδη αναφερθεί, για τη δημιουργία, διαχείριση και συντήρηση των ηλεκτρονικών βάσεων δεδομένων χρησιμοποιούμε ειδικά λογισμικά που ονομάζονται **Συστήματα Διαχείρισης Βάσης Δεδομένων (ΣΔΒΔ) - Database Management Systems (DBMS)**. Ένα από αυτά είναι η Microsoft Access.

6. Ξεκίνημα της Εφαρμογής Βάσεων Δεδομένων

(1) **Start→All Programs→Microsoft Office→**  **Microsoft Access**.

ή

Εάν υπάρχει ήδη το εικονίδιο της εφαρμογής στην επιφάνεια εργασίας (**Desktop**)

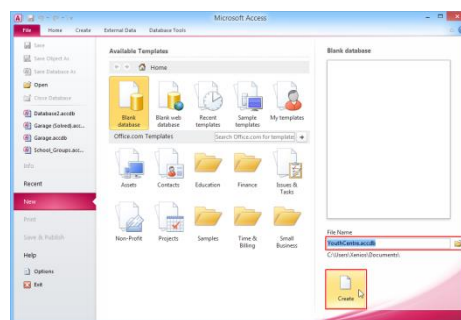


, μπορούμε απλώς να κάνουμε διπλό κλικ πάνω του.

7. Δημιουργία νέας κενής βάσης δεδομένων (Blank database)

Για τη δημιουργία μιας νέας βάσης δεδομένων, εφόσον έχει ήδη ξεκινήσει η εφαρμογή, κάνουμε τα ακόλουθα:

(1) Από την καρτέλα **File** επιλέγουμε την εντολή **New** (εάν δεν είναι ήδη προεπιλεγμένη) και από την περιοχή **Available Templates**, το εικονίδιο **Blank database** και μετά κλικ στο κουμπί **Create**.

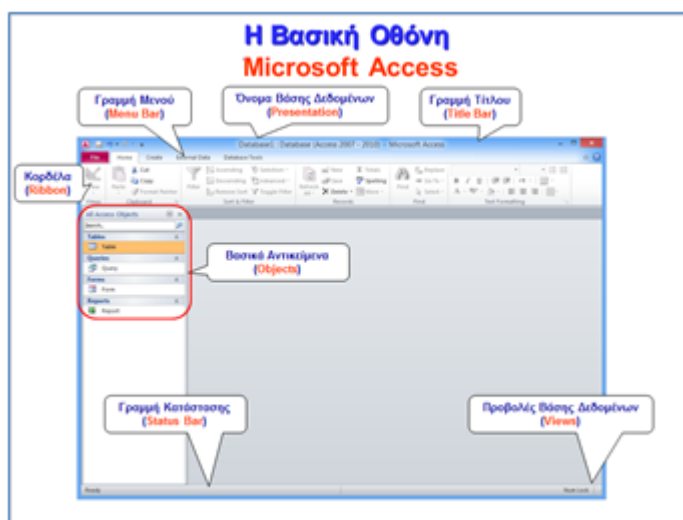


ή

Μπορούμε επίσης να χρησιμοποιήσουμε τον συνδυασμό πλήκτρων **CTRL + N**.

(2) Ακολουθως, κάτω δεξιά στο παράθυρο πληκτρολογούμε το όνομα που θέλουμε να δώσουμε στη βάση δεδομένων μας, επιλέγουμε το χώρο που θέλουμε να αποθηκευτεί και κάνουμε κλικ στο κουμπί **Create**.

(3) Στη συνέχεια, θα ανοίξει το παράθυρο της Microsoft Access, από το οποίο και θα γίνει η ανάπτυξη της βάσης δεδομένων.

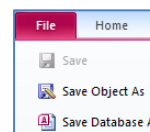


8. Αποθήκευση βάσης δεδομένων (Save)

Εάν θα αποθηκεύσουμε τη βάση δεδομένων μας σε Φορητή Μνήμη (Flash Memory), τοποθετούμε τη Φορητή Μνήμη μας σε μια θύρα USB.

Η προεπιλεγμένη μορφή αρχείου όταν αποθηκεύουμε είναι το **Microsoft Access Database**, το οποίο δίνει την προέκταση **.accdb** στο αρχείο.

- (1) Από την καρτέλα **File** επιλέγουμε την εντολή **Save Database As** για να αποθηκεύσουμε τη βάση δεδομένων μας, ή όταν θέλουμε να την αποθηκεύσουμε με άλλο όνομα ή σε άλλη θέση (*εναλλακτικά χρησιμοποιούμε τον συνδυασμό πλήκτρων* **ALT + F, A**).



- Χρησιμοποιούμε την εντολή **File→Save** για να αποθηκεύσουμε μία ήδη αποθηκευμένη βάση δεδομένων, με το ίδιο όνομα και στην ίδια θέση μετά από αλλαγές που κάναμε.
- Αντί της εντολής **File→Save** μπορούμε εναλλακτικά να κάνουμε κλικ στο κουμπί  ή πατάμε τον συνδυασμό πλήκτρων **CTRL + S**.

- (2) Καθορίζουμε το όνομα (**Filename**), τη θέση (**Save in**) και τον τύπο (**Save as type**).

- (3) Τέλος, κάνουμε κλικ στο κουμπί **Save** για να το αποθηκεύσουμε.

9. Άνοιγμα βάσης δεδομένων (Open)

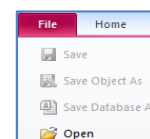
Για να ανοίξουμε μια υφιστάμενη βάση δεδομένων, εφόσον έχει ήδη ξεκινήσει η εφαρμογή, τότε:

- (1) Από την καρτέλα **File** επιλέγουμε την εντολή **Open**.

ή

Μπορούμε επίσης να χρησιμοποιήσουμε τον συνδυασμό πλήκτρων **CTRL + O**.

- (2) Από το παράθυρο που θα ανοίξει, εντοπίζουμε και ανοίγουμε τη βάση δεδομένων που επιθυμούμε.

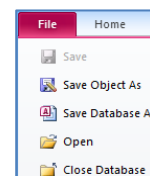


10. Κλείσιμο βάσης δεδομένων (Close Database)

- (1) Από την καρτέλα **File** επιλέγουμε την εντολή **Close Database**.

ή

Μπορούμε επίσης να χρησιμοποιήσουμε τον συνδυασμό πλήκτρων **ALT + F, C**.

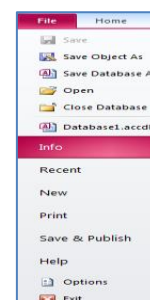


11. Κλείσιμο Εφαρμογής Βάσεων Δεδομένων (Exit)

- (1) Από την καρτέλα **File** επιλέγουμε την εντολή **Exit**.

ή

Μπορούμε επίσης να χρησιμοποιήσουμε τον συνδυασμό πλήκτρων **ALT + F, X**.



Βασικές Έννοιες**Πίνακας
(Table):**

Η οργάνωση των πληροφοριών μιας βάσης δεδομένων γίνεται με τη χρήση πινάκων. Ο κάθε πίνακας έχει το δικό του **όνομα** και αποτελείται από γραμμές και στήλες. Στον πίνακα τα δεδομένα είναι οργανωμένα σε στήλες (**πεδία**) και σε γραμμές (**εγγραφές**).

**Εγγραφή
(Record):**

Εγγραφή είναι η κάθε **γραμμή** ενός πίνακα μιας βάσης δεδομένων η οποία περιέχει όλα τα δεδομένα που περιγράφουν μια συγκεκριμένη καταχώριση του πίνακα και αποτελεί μια ενότητα στοιχείων.

**Πεδίο
(Field):**

Πεδία είναι οι **στήλες** ενός πίνακα μιας βάσης δεδομένων στις οποίες καταχωρούνται οι τιμές για ένα συγκεκριμένο χαρακτηριστικό των εγγραφών του πίνακα.

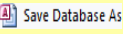
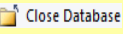
**Ειδικές Σχέσεις
πινάκων
(Relationships):**

Κάθε πίνακας αποτελεί ομάδα από στοιχεία που έχουν μια σχέση μεταξύ τους. Οι πίνακες όμως μπορεί να έχουν και αυτοί μεταξύ τους μια ειδική σχέση. Αυτή η ειδική σχέση ονομάζεται **Relationship**.

**Κύρια Αντικείμενα
Βάσης Δεδομένων
(Objects):**

Πίνακες (**Tables**)
Φόρμες (**Forms**)
Ερωτήματα (**Queries**)
Εκθέσεις (**Reports**)

Υπόμνημα

Περιγραφή (Description)	Καρτέλα (Tab)	Εικονίδιο (Icon)	Πλήκτρα (Keys)
Δημιουργία νέας βάσης δεδομένων (Blank database)			
Αποθήκευση βάσης δεδομένων (Save)			
Αποθήκευση βάσης δεδομένων ως (Save Database As)			
Άνοιγμα βάσης δεδομένων (Open)			
Κλείσιμο βάσης δεδομένων (Close Database)			
Κλείσιμο Εφαρμογής Βάσεων Δεδομένων (Exit)			

Γ6.3 Πίνακες

Τι θα μάθουμε σήμερα:

- ◆ Να δημιουργούμε και να αποθηκεύουμε έναν πίνακα σε μια βάση δεδομένων και να καθορίζουμε τα πεδία και τους τύπους δεδομένων των πεδίων
- ◆ Να καθορίζουμε τις ιδιότητες των πεδίων ενός πίνακα
- ◆ Να αναφέρουμε τα κριτήρια επιλογής προσδιορισμού ενός πρωτεύοντος κλειδιού και να καθορίζουμε το κατάλληλο πεδίο ως πρωτεύον κλειδί
- ◆ Να χρησιμοποιούμε την εναλλαγή προβολών πίνακα (Design View και Datasheet View)
- ◆ Να διαγράψουμε ένα πίνακα.

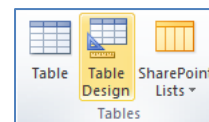
1. Πίνακας (Table)

Ο πίνακας είναι μια συλλογή δεδομένων που περιγράφουν ομοειδή (που έχουν σχέση μεταξύ τους) αντικείμενα. Όλα τα δεδομένα που καταχωρούμε σε μια βάση δεδομένων **αποθηκεύονται σε έναν ή περισσότερους πίνακες** και για αυτό θεωρούνται τα βασικότερα στοιχεία μιας βάσης δεδομένων.

Για **παράδειγμα**, ένας πίνακας μπορεί να περιέχει όλα τα στοιχεία των πελατών μας, ένας άλλος να περιέχει τα στοιχεία των προμηθευτών μας, κ.λπ.

2. Δημιουργία Πίνακα με τη χρήση της Προβολής Σχεδίασης (Table Design)

- (1) Από την καρτέλα **Create** και την ομάδα **Tables** επιλέγουμε το εικονίδιο **Table Design**.



- (2) Στο παράθυρο που εμφανίζεται, και από την **πρώτη γραμμή** του πλαισίου του **Field Name** πληκτρολογούμε το όνομα του πεδίου που θέλουμε να χρησιμοποιήσουμε.
- (3) Στο πλαίσιο του **Data Type** επιλέγουμε από τη λίστα τον τύπο δεδομένων του πεδίου που θέλουμε να χρησιμοποιήσουμε.
- (4) Στο πλαίσιο του **Description** πληκτρολογούμε την περιγραφή-σχόλια του πεδίου που θέλουμε να χρησιμοποιήσουμε.

Field Name	Data Type	Description (Optional)
Όνοματεπώνυμο	Short Text	Εδώ γράφουμε τα ονόματα των πελατών μας
Διεύθυνση	Short Text	Εδώ γράφουμε τις διευθύνσεις των πελατών μας

Field Properties

General | Lookup

Field Size: 255

Format:

Input Mask:

Caption:

Default Value:

Validation Rule:

Validation Text:

Required: No

Allow Zero Length: Yes

Indexed: No

Unicode Compression: No

IME Mode: No Control

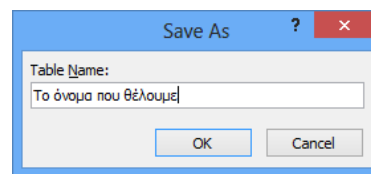
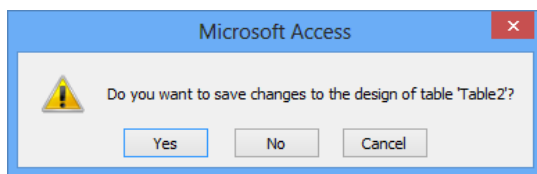
IME Sentence Mode: None

Text Align: General

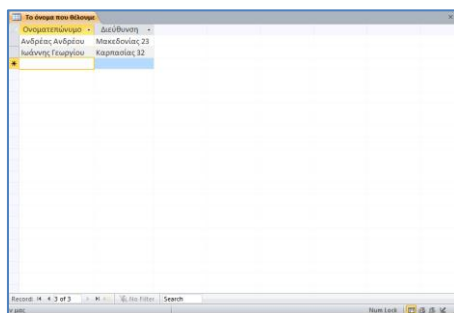
The maximum number of characters you can enter in the field. The largest maximum you can set is 255. Press F1 for help on field size.

- (5) Ακολουθώντας, στη **δεύτερη γραμμή**, πληκτρολογούμε ένα νέο πεδίο με τον νέο τύπο δεδομένων του και τη νέα περιγραφή του και συνεχίζουμε και στις υπόλοιπες γραμμές μέχρι να τελειώσουμε με τα πεδία που θέλουμε να χρησιμοποιήσουμε.

- (6) Στη συνέχεια, κλείνουμε τον πίνακα, του καθορίζουμε ένα **περιγραφικό όνομα** και αποθηκεύουμε τον πίνακά μας.

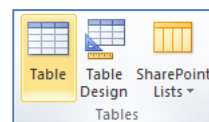


- (7) Τέλος, αρχίζουμε την πληκτρολόγηση των εγγραφών που θέλουμε να χρησιμοποιήσουμε στον πίνακά μας.

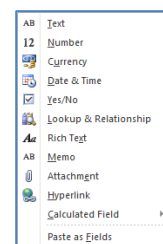


3. Δημιουργία Πίνακα με τη χρήση της Προβολής Φύλλου Δεδομένων (Table)

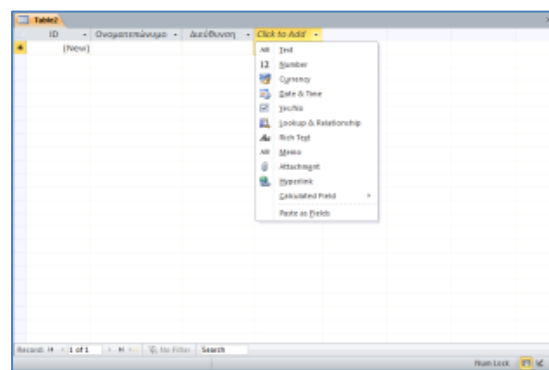
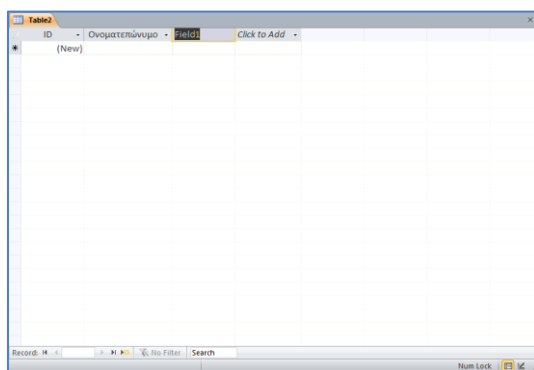
- (1) Από την καρτέλα Create και την ομάδα Tables επιλέγουμε το εικονίδιο Table.



- (2) Ακολούθως, κάνουμε κλικ στο **βελάκι**  δίπλα από το κουμπί **Click to Add** και επιλέγουμε πρώτα τον τύπο δεδομένων του πεδίου που θέλουμε χρησιμοποιήσουμε.

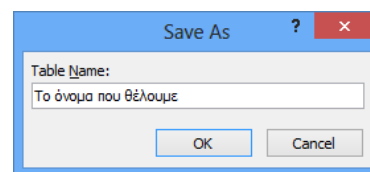
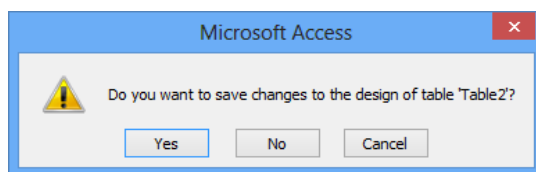


- (3) Στη συνέχεια, στο πλαίσιο του **Field1** πληκτρολογούμε το όνομα του πεδίου που θέλουμε να χρησιμοποιήσουμε.

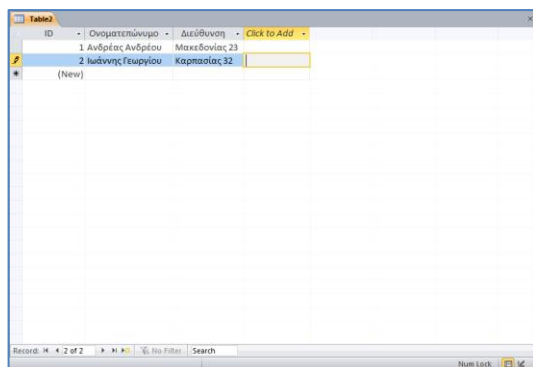


Ακολούθως, πατώντας το πλήκτρο **ENTER** μας πάει στο **επόμενο πεδίο** που θέλουμε να προσθέσουμε και συνεχίζουμε μέχρι να τελειώσουμε με τα πεδία που θέλουμε να χρησιμοποιήσουμε.

- (4) Στη συνέχεια, κλείνουμε τον πίνακα, του καθορίζουμε ένα **περιγραφικό όνομα** και αποθηκεύουμε τον πίνακά μας.



(5) Τέλος, αρχίζουμε την πληκτρολόγηση των εγγραφών που θέλουμε να χρησιμοποιήσουμε στον πίνακά μας.



4. Κανόνες Ονομασίας Πεδίων

Όταν δημιουργούμε έναν πίνακα καλό είναι να χρησιμοποιούμε **περιγραφικά ονόματα πεδίων**, ώστε να είναι κατανοητά και από άλλους χρήστες της βάσης δεδομένων:

- (1) Μπορούν να περιλαμβάνουν οποιονδήποτε συνδυασμό χαρακτήρων και αριθμών, **εκτός από** διάφορα σύμβολα π.χ. την τελεία (.), το θαυμαστικό (!) και τις αγκύλες ([]).
- (2) ΔΕΝ επιτρέπεται να αρχίζουν με ΚΕΝΟ διάστημα.

5. Τύποι Δεδομένων (Data Type)

Ο τύπος δεδομένων ορίζει το **είδος των δεδομένων** που θα μπορεί ο χρήστης να καταχωρίσει στο πεδίο. Χρησιμοποιώντας σωστά τους τύπους δεδομένων μπορούμε να περιορίσουμε τις λανθασμένες καταχωρίσεις στη βάση δεδομένων και να διασφαλίσουμε τη σωστή λειτουργία της.

Η Microsoft Access, όπως και τα περισσότερα ΣΔΒΔ υποστηρίζει τους εξής βασικούς **τύπους δεδομένων**:

- **Short Text** Κείμενο μικρής έκτασης
- **Number** Αριθμός
- **Date/Time** Ημερομηνία/Ωρα
- **Currency** Χρηματικό ποσό
- **AutoNumber** Αυτόματη αρίθμηση
- **Yes/No** Ναι/Όχι ή Σωστό/Λάθος ή Αληθές/Ψευδές

Η Microsoft Access, υποστηρίζει επίσης κάποιους επιπρόσθετους (εξειδικευμένους) τύπους δεδομένων:

- **Long Text** Κείμενο μεγάλης έκτασης

- **OLE Object** Αντικείμενο OLE - Αντικείμενα από άλλες εφαρμογές (*Εικόνα, Ήχο, Βίντεο*)
- **Hyperlink** Υπερσύνδεσμος (*Εισαγωγή συνδέσμου με ιστοσελίδα στο Διαδίκτυο*)
- **Attachment** Επισυναπτόμενο αρχείο
- **Calculated** Μαθηματικός υπολογισμός πεδίων
- **Lookup Wizard...** Καθορισμός τιμών στο πεδίο (*Λίστα Επιλογών*)

Παρακάτω ακολουθεί σύντομη περιγραφή των χαρακτηριστικών για τον κάθε τύπο πεδίου:

- (α) **Short Text** - Μπορεί να περιέχει **μέχρι και 255** χαρακτήρες κειμένου, σύμβολα ή αριθμούς.
- (β) **Number** - Μπορεί να περιέχει **αριθμούς** για να τους χρησιμοποιήσουμε για μαθηματικές πράξεις.
- (γ) **Date/Time** - Κατάλληλος για την καταχώριση δεδομένων που αναφέρονται σε **ημερομηνίες και ώρες**.
- (δ) **Currency** - Κατάλληλος τύπος δεδομένων για την καταχώριση δεδομένων που αναφέρονται σε **χρηματικά ποσά**.
- (ε) **AutoNumber** - Παίρνει μοναδικές αριθμητικές τιμές οι οποίες **καθορίζονται αυτόματα** από την εφαρμογή.
- (στ) **Yes/No** - Μας επιτρέπει να δημιουργούμε πεδία που περιέχουν **λογικές τιμές** (*Αληθές ή Ψευδές, Ναι ή Όχι*).
- (ζ) **Long Text** - Μπορεί να περιέχει **μέχρι και 65.535** χαρακτήρες κειμένου, σύμβολα ή αριθμούς.
- (η) **OLE Object** - Μας επιτρέπει να εισάγουμε αντικείμενα από άλλες εφαρμογές όπως **εικόνα, ήχο και βίντεο**.
- (θ) **Hyperlink** - Κατάλληλος για εισαγωγή διευθύνσεων **υπερσυνδέσμων**.
- (ι) **Attachment** - Κατάλληλος για εισαγωγή **επισυναπτόμενων αρχείων**.
- (ια) **Calculated** - Περιέχει **μαθηματικές πράξεις** για υπολογισμούς πεδίων.
- (ιβ) **Lookup Wizard...** - Περιέχει **προκαθορισμένες τιμές** όπως είναι μια λίστα επιλογών.

6. Ιδιότητες Πεδίων (Field Properties)

Οι ιδιότητες των πεδίων μάς επιτρέπουν να προσαρμόσουμε τον τρόπο με τον οποίο το σύστημα διαχείρισης βάσεων δεδομένων **αποθηκεύει, εμφανίζει και χειρίζεται** τα δεδομένα ενός πεδίου όπως:

- (1) Πόσους χαρακτήρες μπορούν να δεχτούν τα πεδία κειμένου.
- (2) Εάν ένα πεδίο πρέπει οπωσδήποτε να περιέχει κάποια τιμή.
- (3) Ποια μορφή πρέπει να έχουν τα δεδομένα ενός πεδίου.

Π.χ. οι βασικότερες **ιδιότητες των πεδίων** στη Microsoft Access είναι:

- **Field Size** Μέγεθος πεδίου
- **Format** Μορφή (τρόπος εμφάνισης)
- **Input Mask** Μάσκα εισαγωγής
- **Caption** Λεζάντα (Ταμπέλα)
- **Default Value** Προεπιλεγμένη (αρχική) τιμή
- **Validation Rule** Κανόνας επικύρωσης ή εγκυρότητας
- **Validation Text** Κείμενο επικύρωσης ή εγκυρότητας
- **Required** Απαιτείται (Εάν είναι υποχρεωτικό ή όχι)
- **Decimal Places** Δεκαδικές θέσεις ή Δεκαδικά ψηφία

Παρακάτω ακολουθεί σύντομη περιγραφή των χαρακτηριστικών για την κάθε ιδιότητα των πεδίων:

- (α) **Field Size** - Μας επιτρέπει να καθορίσουμε το **μέγιστο πλήθος** των χαρακτήρων του πεδίου.
- (β) **Format** - Καθορίζουμε τη **μορφή** με την οποία θα παρουσιάζονται τα δεδομένα στον χρήστη.
- (γ) **Input Mask** - Καθορίζουμε τη **μορφή** στην οποία θα μπορεί να καταχωρεί τα δεδομένα ο χρήστης.
- (δ) **Caption** - Καθορίζουμε το **όνομα** που θα εμφανίζεται στον πίνακα αντί για το αρχικό όνομα του πεδίου.
- (ε) **Default Value** - Καθορίζουμε την τιμή που θα καταχωρείται **αυτόματα** στο συγκεκριμένο πεδίο σε κάθε νέα εγγραφή.
- (στ) **Validation Rule** - Εισάγουμε **κριτήρια** στο πεδίο προκειμένου να περιορίσουμε τις τιμές που μπορεί να δέχεται.
- (ζ) **Validation Text** - Ορίζουμε το **μήνυμα σφάλματος** που θα εμφανίζεται όταν οι τιμές που καταχωρούνται δεν πληρούν τα κριτήρια του κανόνα επικύρωσης.
- (η) **Required** - Επιλέγουμε εάν θα είναι **υποχρεωτική** η καταχώριση δεδομένων στο πεδίο ή αν θα επιτρέπεται να το αφήνουμε κενό.
- (θ) **Decimal Places** - Καθορίζουμε το πλήθος των **δεκαδικών ψηφίων** που θα εμφανίζονται στα δεξιά της υποδιαστολής.

7. Πρωτεύον Κλειδί (Primary Key)

Η επιλογή του πρωτεύοντος κλειδιού σε έναν πίνακα είναι μια από τις βασικότερες επιλογές που έχουμε να κάνουμε. Το πρωτεύον κλειδί είναι ένα **πεδίο** ή **συνδυασμός πεδίων** του πίνακα, που πρέπει πάντα να περιέχει/ουν έγκυρα μοναδικά δεδομένα, με αποτέλεσμα να διασφαλίζεται η μοναδικότητα κάθε εγγραφής του πίνακα. Με άλλα λόγια, εάν έχουμε έναν πίνακα και γνωρίζουμε, την τιμή του πρωτεύοντος κλειδιού, θα πρέπει να μπορούμε να εντοπίσουμε το πολύ μία εγγραφή η οποία να έχει αυτή την τιμή. Αυτό σημαίνει ότι δεν θα

υπάρχουν δύο ή περισσότερες εγγραφές στον πίνακα που να έχουν την ίδια τιμή για το πρωτεύον κλειδί.

Γενικά, ως πρωτεύον κλειδί ενός πίνακα επιλέγουμε ένα πεδίο που περιέχει κάποιο κωδικό ή αριθμό που πρέπει να είναι **μοναδικός για κάθε εγγραφή** του πίνακα, π.χ.: Αριθμός Ταυτότητας, Αριθμός Μητρώου Μαθητή κ.λπ. Θα πρέπει να είμαστε προσεκτικοί στην επιλογή μας, διότι πεδία όπου η καταχώριση είναι προαιρετική δεν είναι κατάλληλα.

Σε κάποιες περιπτώσεις όπου δεν υπάρχει ένα πεδίο για να λειτουργήσει ως πρωτεύον κλειδί, πιθανόν να υπάρχει ένας συνδυασμός πεδίων. Για παράδειγμα, στη βάση δεδομένων ενός σχολείου, στον πίνακα των απουσιών των μαθητών καταχωρούνται ο αριθμός μητρώου, η ημερομηνία απουσίας, η περίοδος απουσίας, το μάθημα και κατά πόσον είναι δικαιολογημένη ή όχι. Κανένα από αυτά τα πεδία από μόνο του δεν είναι κατάλληλο ως πρωτεύον κλειδί, γι αυτό και χρησιμοποιούμε τον συνδυασμό (αριθμός μητρώου, ημερομηνία, περίοδος) ως πρωτεύον κλειδί στον πίνακα αυτό.

Εάν στον πίνακά μας δεν έχουμε κάποιο πεδίο ή συνδυασμό πεδίων που να είναι μοναδικά σε κάθε εγγραφή, τότε η επιλογή του πεδίου αυτόματης αρίθμησης (**AutoNumber**) είναι ιδανική για τον ορισμό πρωτεύοντος κλειδιού, αφού οι τιμές του είναι εξ' ορισμού μοναδικές.

Παράδειγμα:

Για τον πίνακα των μελών ενός κέντρου νεότητας που φαίνεται στην επόμενη σελίδα, το τηλέφωνο (πεδίο «TelNumber») μπορεί να φαίνεται κατάλληλο ως πρωτεύον κλειδί, αλλά επειδή πιθανό να υπάρχουν μέλη που να μην έχουν το δικό τους τηλέφωνο ή να μοιράζονται (με τα αδέρφια τους) το ίδιο τηλέφωνο δεν μπορεί να το χρησιμοποιήσουμε. Για τον ίδιο λόγο το πεδίο Email είναι ακατάλληλο. Επίσης ο συνδυασμός των πεδίων Όνομα και Επώνυμο («Firstname» και «Surname») δεν είναι κατάλληλος, διότι υπάρχει η πιθανότητα να έχουμε μέλη με το ίδιο όνομα και επώνυμο (εξάλλου δεν συνηθίζεται να χρησιμοποιούμε πεδία αυτού του τύπου και έκτασης ως πρωτεύον κλειδί). Η καλύτερη επιλογή γι' αυτό τον πίνακα είναι η δημιουργία ενός πεδίου («YouthMemberID») με τύπο δεδομένων AutoNumber με σκοπό να λειτουργήσει ειδικά ως πρωτεύον κλειδί.

YouthMembers									
YouthMemberID	Firstname	Surname	Sex	Birthdate	Address	Town	TelNumber	Email	ActivityCode
1013	Ανδρέας	Κόκου	A	24/8/1999	Ριζοκαρπάσου 15	Λεμεσός	99715805	Andre4@gmail.com	8
1014	Ανδρέας	Βάσου	A	25/8/1999	Αμμοχώστου 27	Λεμεσός	99492649	Andrew@otenet.com.cy	9
1015	Ανδρέας	Εφραίμ	A	26/8/1999	Καντάρας 27	Λεμεσός	99404434	Andrew3@gmail.com	10
1016	Ανδρέας	Σωκράτης	A	27/8/1999	Ριζοκαρπάσου 16	Λεμεσός	99545302	Andrew2@gmail.com	10
1017	Ανδρέας	Σταύρου	A	28/8/1999	Καντάρας 21	Λεμεσός	99839582	Andrew@cytanet.com.cy	10
1018	Ανδρέας	Γεωργίου	A	1/2/2000	Μόρφου 19	Λεμεσός	99764158	Andrew1@gmail.com	8
1019	Ανδρέας	Γεωργίου	A	2/2/2000	Κυρηνείας 29	Λεμεσός	99857363	Andrew@gmail.com	9
1020	Ανδρέας	Κωνσταντίνου	A	3/2/2000	Καντάρας 18	Λεμεσός	99557156	Andrew@yahoo.com	9
1021	Ανδρέας	Πέτρου	A	4/2/2000	Ριζοκαρπάσου 913	Λεμεσός	99651574	Andrew@yahoo.gr	7
1022	Άντρια	Ανδρέα	Θ	5/2/2000	Μόρφου 35	Λεμεσός	99510430	Andria@cytanet.com.cy	6
1023	Ανδριανή	Νεοκλή	Θ	6/2/2000	Αμμοχώστου 3	Λεμεσός	99233241	Andriani@gmail.com	7
1024	Άγγελος	Λεωνίδα	A	7/2/2000	Ριζοκαρπάσου 919	Λεμεσός	99881071	Angel@cytanet.com.cy	5

Παρόλο που η Microsoft Access μας δίνει τη δυνατότητα να δημιουργήσουμε πίνακες χωρίς πρωτεύον κλειδί, αυτό γίνεται σε πολύ σπάνιες περιπτώσεις. Η Microsoft Access, όπως και άλλα παρόμοια ΣΔΒΔ, χρησιμοποιεί τα πεδία πρωτεύοντος κλειδιού για να συνδέσει τα δεδομένα από έναν πίνακα με αντίστοιχα σε άλλον. Κατά συνέπεια, επειδή τα δεδομένα ενός πίνακα δεν είναι απομονωμένα από τα δεδομένα των άλλων πινάκων σε μια βάση δεδομένων,

αλλά συνδέονται μεταξύ τους, σε κάθε πίνακα στη βάση δεδομένων πρέπει καθορίζεται το πρωτεύον κλειδί.

8. Πώς καθορίζουμε το Πρωτεύον Κλειδί

(1) Επιλέγουμε τον πίνακα και κάνουμε **διπλό κλικ** πάνω του για να τον ανοίξουμε πρώτα σε προβολή φύλλου δεδομένων (**Datasheet View**).

(2) Από την καρτέλα **Home** και την ομάδα **Views** επιλέγουμε το εικονίδιο **Design View** για να ανοίξουμε τον πίνακά μας σε προβολή σχεδίασης.

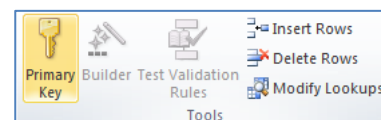


ή

Επιλέγουμε τον πίνακα και κάνουμε δεξί κλικ πάνω του και επιλέγουμε την εντολή **Design View** για να τον ανοίξει σε προβολή σχεδίασης.

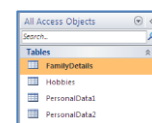
(3) Στη συνέχεια, επιλέγουμε το **κατάλληλο πεδίο ή πεδία** που θέλουμε να ορίσουμε ως πρωτεύον κλειδί του πίνακά μας.

(4) Από την καρτέλα **Table Tools**→**Design** και την ομάδα **Tools** επιλέγουμε το εικονίδιο **Primary Key** για να ορίσουμε με αυτό τον τρόπο το πρωτεύον κλειδί στον πίνακά μας που θέλουμε να χρησιμοποιήσουμε.



9. Διαγραφή Πίνακα (Delete Table)

(1) Από την περιοχή όλων των αντικειμένων της εφαρμογής (**All Access Objects**) κάτω από τον τίτλο **Tables**, όπου εμφανίζονται όλοι οι πίνακες που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε τον πίνακα που θέλουμε να διαγράψουμε.



(2) Ακολουθώντας, κάνουμε δεξί κλικ και επιλέγουμε την εντολή **Delete** για να τον διαγράψουμε από τη βάση δεδομένων μας.

ή

Από την καρτέλα **Home** και την ομάδα **Records** επιλέγουμε το εικονίδιο **Delete**.



ή

Μπορούμε επίσης να χρησιμοποιήσουμε το πλήκτρο **Delete**.

10. Προβολές για Πίνακα (Views)

10.1 Προβολή Σχεδίασης (Design View)

(1) Από την περιοχή όλων των αντικειμένων της εφαρμογής (**All Access Objects**) κάτω από τον τίτλο **Tables**, όπου εμφανίζονται όλοι οι πίνακες που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε τον πίνακα που θέλουμε και κάνουμε **διπλό κλικ** πάνω του για να τον ανοίξουμε πρώτα σε προβολή φύλλου δεδομένων (**Datasheet View**).

- (2) Ακολουθώντας, από την **καρτέλα Home** και την **ομάδα Views** επιλέγουμε το **εικονίδιο Design View** για να ανοίξουμε τον πίνακά μας σε **προβολή σχεδίασης**.

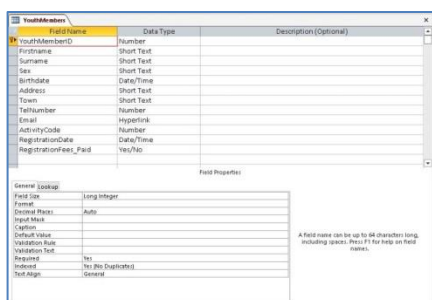


ή

Κάνουμε κλικ στο κουμπί  από τη γραμμή κατάστασης στην κάτω δεξιά γωνιά της οθόνης μας.

ή

Από την περιοχή **όλων των αντικειμένων της εφαρμογής (All Access Objects)** κάτω από τον τίτλο **Tables**, όπου εμφανίζονται όλοι οι πίνακες που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε τον πίνακα που θέλουμε και κάνουμε **δεξί κλικ** πάνω του και επιλέγουμε την εντολή **Design View** για να τον ανοίξει σε **προβολή σχεδίασης**.

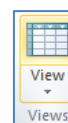


10.2 Προβολή Φύλλου Δεδομένων (Datasheet View)

- (1) Από την περιοχή **όλων των αντικειμένων της εφαρμογής (All Access Objects)** κάτω από τον τίτλο **Tables**, όπου εμφανίζονται όλοι οι πίνακες που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε τον πίνακα που θέλουμε και κάνουμε **διπλό κλικ** πάνω του για να τον ανοίξουμε σε **προβολή φύλλου δεδομένων (Datasheet View)**.

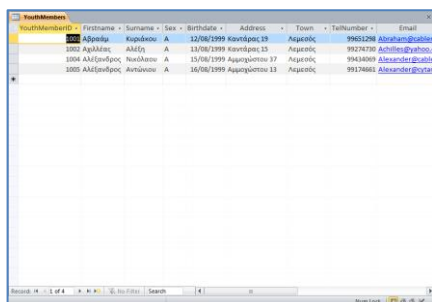
ή

Από την **καρτέλα Home** και την **ομάδα Views** επιλέγουμε το **εικονίδιο Datasheet View** για να ανοίξουμε τον πίνακά μας σε **προβολή φύλλου δεδομένων**.

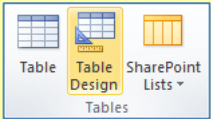
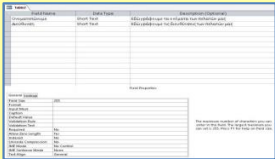
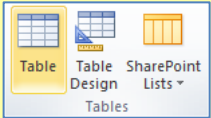
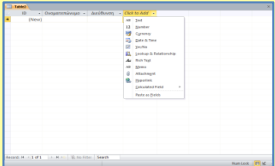
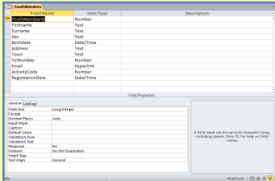
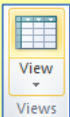
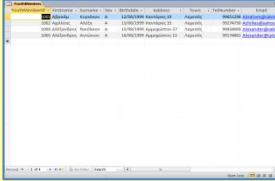
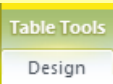
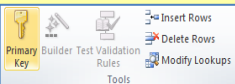
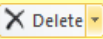


ή

Κάνουμε κλικ στο κουμπί  από τη γραμμή κατάστασης στην κάτω δεξιά γωνιά της οθόνης μας.



Υπόμνημα

Περιγραφή (Description)	Καρτέλα (Tab)	Εικονίδιο (Icon)	Ομάδα (Group)	Παράθυρο Διαλόγου (Dialog Box)	Πλήκτρα (Keys)
Δημιουργία Πίνακα με τη χρήση της Προβολής Σχεδίασης (Table Design)	Create				
Δημιουργία Πίνακα με τη χρήση της Προβολής Φύλλου Δεδομένων (Table)	Create				
Προβολή Σχεδίασης (Design View)	Home				
Προβολή Φύλλου Δεδομένων (Datasheet View)	Home				
Δημιουργία Πρωτεύοντος Κλειδιού (Primary Key)					
Διαγραφή Πίνακα (Delete Table)	Home				

Γ6.4 Εγγραφές

Τι θα μάθουμε σήμερα:

- Να καταχωρούμε, να επεξεργαζόμαστε και να διαγράφουμε εγγραφές σε πίνακα
- Να μετακινούμαστε μεταξύ των εγγραφών σε πίνακα.

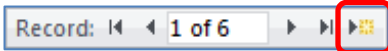
1. Εισαγωγή, Επεξεργασία, Διαγραφή και Μετακίνηση μεταξύ των εγγραφών σε ένα πίνακα

Μετά τη δημιουργία ενός πίνακα, μπορούμε να εισαγάγουμε, να επεξεργαστούμε και να διαγράψουμε εγγραφές σε έναν πίνακα απευθείας ή να μετακινηθούμε μεταξύ των εγγραφών του πίνακα, κάνοντας **διπλό κλικ** πάνω του και ανοίγοντάς τον σε προβολή φύλλου δεδομένων (**Datasheet View**).

1.1 Εισαγωγή Εγγραφής σε έναν πίνακα (New Record)

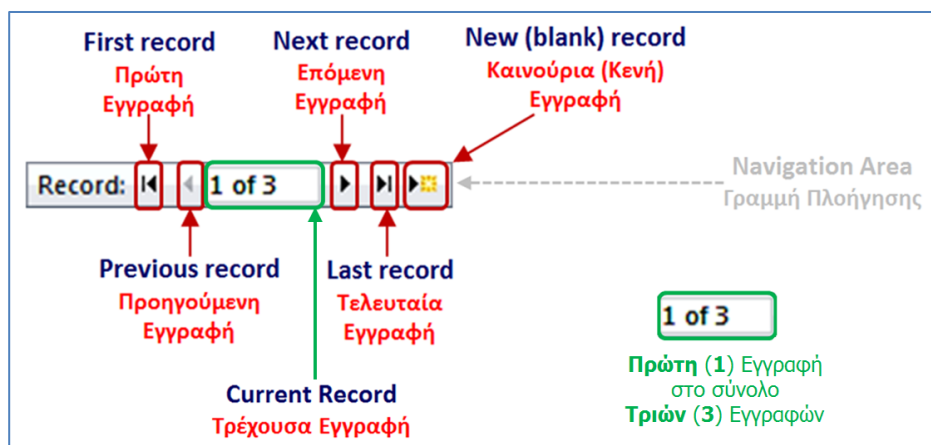
- (1) Κάνουμε δεξί κλικ σε μια οποιαδήποτε γραμμή (εγγραφή) του πίνακά μας.
- (2) Ακολουθώντας, κάνουμε κλικ στο κουμπί  και η νέα μας εγγραφή **εισάγεται τελευταία** στον πίνακά μας.

ή

Από τη γραμμή πλοήγησης (Navigation Area)  κάνουμε κλικ στο εικονίδιο New (blank) record και καταχωρούμε τη νέα μας εγγραφή ως τελευταία στον πίνακά μας.

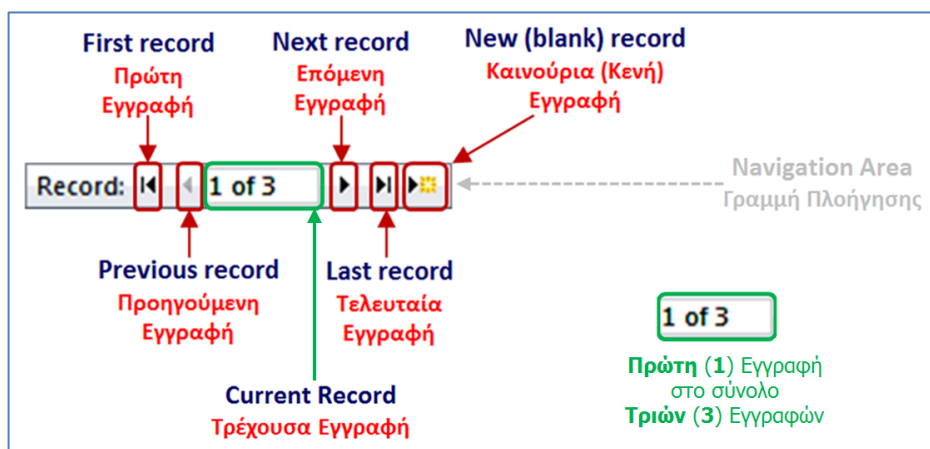
1.2 Μετακίνηση μεταξύ των εγγραφών σε έναν πίνακα

- (1) Μπορούμε να μετακινηθούμε μεταξύ των εγγραφών ενός πίνακα, κάνοντας κλικ σε ένα από τα εικονίδια της γραμμής πλοήγησης πιο κάτω:



1.3 Επεξεργασία Εγγραφής σε έναν πίνακα

- (1) Μπορούμε να πάμε σε οποιαδήποτε εγγραφή του πίνακά μας και να την επεξεργαστούμε (*αλλαγή, διόρθωση κ.λπ.*), κάνοντας κλικ σε ένα από τα εικονίδια της γραμμής πλοήγησης πιο κάτω:



1.4 Διαγραφή Εγγραφής σε έναν πίνακα (Delete Record)

- (1) Επιλέγουμε την **εγγραφή** που θέλουμε να διαγράψουμε από τον πίνακά μας.
- (2) Από την καρτέλα **Home** και την ομάδα **Records** επιλέγουμε το εικονίδιο **Delete**.
- (3) Ακολουθώντας, κάνουμε κλικ στο κουμπί και η εγγραφή μας **διαγράφεται** από τον πίνακά μας.

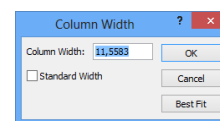


2. Μορφοποίηση Πίνακα

Μετά τη δημιουργία ενός πίνακα, μπορούμε να τον μορφοποιήσουμε, κάνοντας **διπλό κλικ** πάνω του και ανοίγοντάς τον σε προβολή φύλλου δεδομένων (**Datasheet View**).

2.1 Αλλαγή Πλάτους Στήλης-Πεδίου (Field Width)

- (1) Κάνουμε δεξί κλικ στη στήλη (**πεδίο**) που θέλουμε να αυξομειώσουμε το πλάτος της.
- (2) Ακολουθώντας, κάνουμε κλικ στο κουμπί και στο παράθυρο που εμφανίζεται πληκτρολογούμε το **επιθυμητό πλάτος** που θέλουμε να χρησιμοποιήσουμε.



ή

Κάνουμε κλικ στο **δεξιό όριο** μιας στήλης και όταν ο δείκτης του ποντικιού πάρει τη μορφή γραμμής με τα δύο αντίθετα βέλη, **σύρουμε ανάλογα** προς τα αριστερά ή προς τα δεξιά για να αλλάξουμε το πλάτος του.

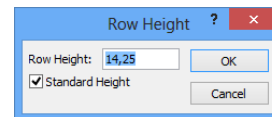


ή

Κάνουμε **διπλό κλικ** στο **δεξιό όριο** μιας στήλης και τα περιεχόμενα της στήλης προσαρμόζονται αυτόματα, ανάλογα με το περιεχόμενό της.

2.2 Αλλαγή Ύψους Γραμμής-Εγγραφής (Row Height)

- (1) Κάνουμε δεξί κλικ στη γραμμή (**εγγραφή**) της οποίας το ύψος θέλουμε να αυξομειώσουμε.
- (2) Ακολούθως, κάνουμε κλικ στο κουμπί  Row Height... και στο παράθυρο που εμφανίζεται πληκτρολογούμε το **επιθυμητό ύψος** που θέλουμε να χρησιμοποιήσουμε.



ή

Κάνουμε κλικ στο **κάτω** όριο μιας γραμμής και όταν ο δείκτης του ποντικιού πάρει τη μορφή γραμμής με τα δύο αντίθετα βέλη, **σύρουμε**  **ανάλογα** προς τα αριστερά ή προς τα δεξιά για να αλλάξουμε το ύψος όλων των γραμμών του πίνακα.

2.3 Απόκρυψη/Εμφάνιση Πεδίων (Hide/Unhide Fields)

- (1) Κάνουμε δεξί κλικ στη στήλη (**πεδίο**) που θέλουμε να αποκρύψουμε ή να εμφανίσουμε από τον πίνακά μας.
- (2) Ακολούθως, κάνουμε κλικ στο κουμπί  Hide Fields ή στο κουμπί  Unhide Fields και το πεδίο μας **εξαφανίζεται** ή **εμφανίζεται** ανάλογα στον πίνακά μας.

2.4 Εισαγωγή Πεδίου (Insert Field)

- (1) Κάνουμε δεξί κλικ σε μια στήλη (**πεδίο**) που πριν από αυτή θέλουμε να εισαγάγουμε ένα νέο πεδίο στον πίνακά μας.
- (2) Ακολούθως, κάνουμε κλικ στο κουμπί  Insert Field και το νέο μας πεδίο **εισάγεται** στον πίνακά μας.
- (3) Στη συνέχεια, δίνουμε μια κατάλληλη περιγραφή (ονομασία) στο νέο πεδίο, διαγράφοντας αυτή που έδωσε το σύστημα.

2.5 Διαγραφή Πεδίου (Delete Field)

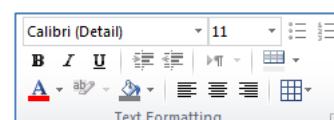
- (1) Κάνουμε δεξί κλικ στη στήλη (**πεδίο**) που θέλουμε να διαγράψουμε από τον πίνακά μας.
- (2) Ακολούθως, κάνουμε κλικ στο κουμπί  Delete Field και το πεδίο μας **διαγράφεται** από τον πίνακά μας.

2.6 Μετονομασία Πεδίου (Rename Field)

- (1) Κάνουμε δεξί κλικ στη στήλη (**πεδίο**) του οποίου το όνομα θέλουμε να αλλάξουμε στον πίνακά μας.
- (2) Ακολούθως, κάνουμε κλικ στο κουμπί  Rename Field και το πεδίο μας **μετονομάζεται** στον πίνακά μας.

2.7 Μορφοποίηση Κειμένου (Text Formatting)

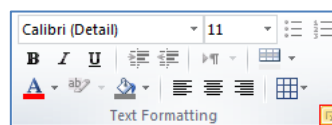
- (1) Από την καρτέλα **Home** και την ομάδα **Text Formatting** κάνουμε κλικ σε μια από τις



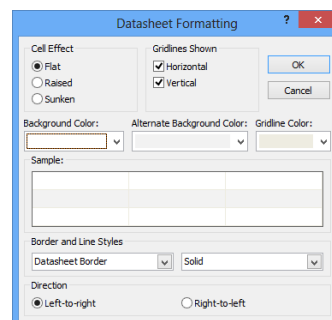
διαθέσιμες επιλογές μορφοποίησης που υπάρχουν και επιλέγουμε τη μορφοποίηση που θέλουμε να χρησιμοποιήσουμε.

2.8 Χρώμα Φόντου (Background Color)

(1) Από την καρτέλα **Home** και την ομάδα **Text Formatting** κάνουμε κλικ στο διαγώνιο βελάκι  στη δεξιά κάτω γωνία του.

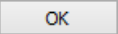


(2) Στο παράθυρο που εμφανίζεται, κάνουμε κλικ στη λίστα του **Background Color** και επιλέγουμε το βασικό χρώμα φόντου που θέλουμε να χρησιμοποιήσουμε.



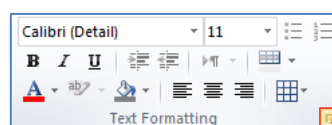
ή

Στο παράθυρο που εμφανίζεται, κάνουμε κλικ στη λίστα του **Alternate Background Color** και επιλέγουμε το εναλλακτικό χρώμα φόντου που θέλουμε να χρησιμοποιήσουμε.

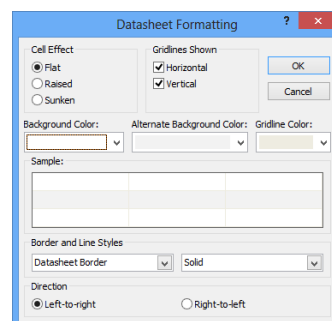
(3) Ακολουθώντας, κάνουμε κλικ στο κουμπί  για να το εφαρμόσει.

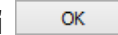
2.9 Εφέ Κελιών (Cell Effect)

(1) Από την καρτέλα **Home** και την ομάδα **Text Formatting** κάνουμε κλικ στο διαγώνιο βελάκι  στη δεξιά κάτω γωνία του.



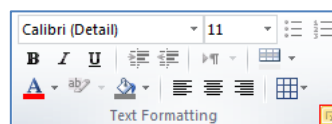
(2) Στο παράθυρο που εμφανίζεται, κάνουμε κλικ σε μια από τις επιλογές της περιοχής του **Cell Effect** και επιλέγουμε το υπερυψωμένο εφέ (Raised) ή το βυθισμένο εφέ (Sunken) που θέλουμε να χρησιμοποιήσουμε.



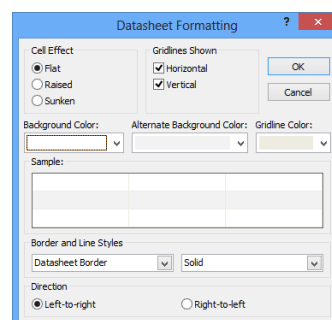
(3) Ακολουθώντας, κάνουμε κλικ στο κουμπί  για να το εφαρμόσει.

2.10 Γραμμές Πλέγματος και Χρώμα Πλέγματος (Gridlines Shown and Gridline Color)

(1) Από την καρτέλα **Home** και την ομάδα **Text Formatting** κάνουμε κλικ στο διαγώνιο βελάκι  στη δεξιά κάτω γωνία του.

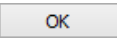


(2) Στο παράθυρο που εμφανίζεται, κάνουμε κλικ σε μια από τις επιλογές της περιοχής του **Gridlines Shown** και επιλέγουμε εάν θα εμφανίζονται οι οριζόντιες (Horizontal) ή οι κάθετες γραμμές πλέγματος (Vertical) που θέλουμε να χρησιμοποιήσουμε.



(3) Στο παράθυρο που εμφανίζεται, επίσης, κάνουμε

κλικ στη λίστα του **Gridline Color** και επιλέγουμε το χρώμα πλέγματος που θέλουμε να χρησιμοποιήσουμε.

(4) Ακολουθώντας, κάνουμε κλικ στο κουμπί  για να το εφαρμόσει.

3. Ταξινόμηση Δεδομένων (Sort)

Σε μια ηλεκτρονική βάση δεδομένων υπάρχει η δυνατότητα χρήσης της λειτουργίας της ταξινόμησης δεδομένων για να ρυθμίσουμε τη σειρά με την οποία θα εμφανίζονται οι διάφορες εγγραφές.

Με την κατάλληλη **ταξινόμηση δεδομένων** σε έναν πίνακα, μια φόρμα, ένα ερώτημα ή μια έκθεση, διευκολύνεται η κατανόηση των δεδομένων μας και επιτυγχάνεται η γρήγορη και πιο ασφαλής εξαγωγή συμπερασμάτων από μια βάση με μεγάλο όγκο δεδομένων. Επίσης, επιτυγχάνεται ο γρήγορος εντοπισμός συγκεκριμένων δεδομένων που θέλουμε.

Για **παράδειγμα**, ένας κατάλογος προϊόντων που περιλαμβάνει τα προϊόντα με αλφαβητική σειρά ή με βάση την τιμή μονάδας είναι πολύ ευκολότερος στη χρήση του από ότι ένας κατάλογος που περιλαμβάνει τα στοιχεία με τυχαία σειρά.

Γενικά, μπορούμε να ταξινομήσουμε τα δεδομένα με:

- **Αλφαβητική σειρά** ταξινομώντας τα, αρχίζοντας από το
 - **A** έως το **Ω** (*Αύξουσα/Ascending*) ή
 - **Ω** έως το **A** (*Φθίνουσα/Descending*).
- **Αριθμητική σειρά** ταξινομώντας τα, αρχίζοντας από το
 - **μικρότερο** στον **μεγαλύτερο αριθμό** (*Αύξουσα/Ascending*) ή
 - **μεγαλύτερο** στον **μικρότερο αριθμό** (*Φθίνουσα/Descending*).

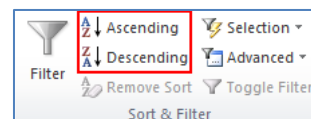
3.1 Ταξινόμηση (Sort) σε Πίνακα, Φόρμα, Ερώτημα και Έκθεση

(1) Ανάλογα:

- Εάν θέλουμε να ταξινομήσουμε πίνακα, τότε επιλέγουμε τον πίνακα και κάνουμε **διπλό κλικ** πάνω του για να τον ανοίξουμε πρώτα σε προβολή φύλλου δεδομένων (**Datasheet View**).
- Εάν θέλουμε να ταξινομήσουμε φόρμα, τότε επιλέγουμε τη φόρμα και κάνουμε **διπλό κλικ** πάνω της για να την ανοίξουμε πρώτα σε προβολή φόρμας (**Form View**).
- Εάν θέλουμε να ταξινομήσουμε ερώτημα, τότε επιλέγουμε το ερώτημα και κάνουμε **διπλό κλικ** πάνω του για να το ανοίξουμε πρώτα σε προβολή φύλλου δεδομένων (**Datasheet View**).
- Εάν θέλουμε να ταξινομήσουμε έκθεση, τότε επιλέγουμε την έκθεση και κάνουμε **δεξί κλικ** πάνω της και επιλέγουμε την εντολή **Layout View** για να την ανοίξουμε σε προβολή διάταξης έκθεσης (**Layout View**).

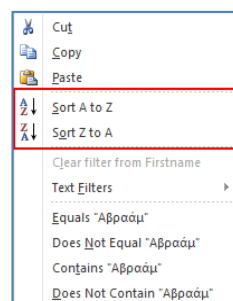
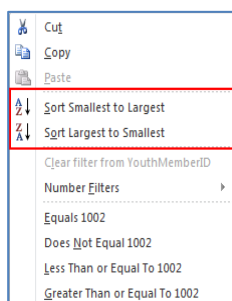
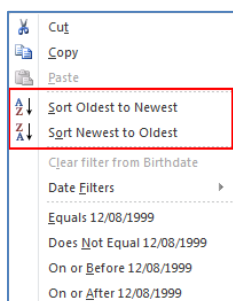
(2) Στη συνέχεια, κάνουμε κλικ στο **πεδίο** το οποίο θέλουμε να ταξινομήσουμε.

- (3) Ακολουθώντας, από την καρτέλα **Home** και την ομάδα **Sort & Filter** επιλέγουμε το εικονίδιο **Ascending** ή το εικονίδιο **Descending** για να **ταξινομήσουμε** το πεδίο μας είτε σε αύξουσα σειρά είτε σε φθίνουσα σειρά.



ή

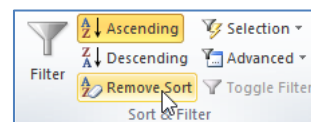
Κάνουμε δεξί κλικ στο **πεδίο** το οποίο θέλουμε να ταξινομήσουμε.



Ακολουθώντας, κάνουμε κλικ σε ένα από τα πιο πάνω κουμπιά για να **ταξινομήσουμε** το πεδίο μας είτε σε αύξουσα σειρά είτε σε φθίνουσα σειρά.

3.2 Κατάργηση Ταξινόμησης σε Πίνακα, Φόρμα, Ερώτημα και Έκθεση (Remove Sort)

- (1) Από την καρτέλα **Home** και την ομάδα **Sort & Filter** επιλέγουμε το εικονίδιο **Remove Sort** για να καταργήσουμε την ταξινόμηση που χρησιμοποιήσαμε προηγουμένως στο πεδίο μας.



4. Χρήση της Εντολής για την Αναζήτηση Πληροφοριών (Find)

Η εντολή αναζήτησης πληροφοριών (**Find**) χρησιμοποιείται για τη **γρήγορη εύρεση πληροφοριών** από έναν πίνακα ή από μια φόρμα. Αυτό μπορούμε να το πετύχουμε είτε όταν βρισκόμαστε σε προβολή φύλλου δεδομένων (**Datasheet View**) του πίνακα, είτε σε προβολή φόρμας (**Form View**) της φόρμας. Να σημειώσουμε ότι η εντολή αναζήτησης πληροφοριών είναι ένας προσωρινός τρόπος εύρεσης πληροφοριών και για αυτό τον λόγο **ΔΕΝ μπορεί να αποθηκευτεί** για μελλοντική χρήση.

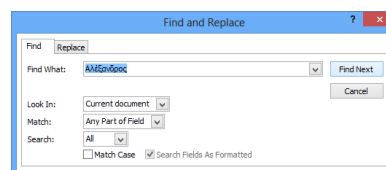
- (1) Επιλέγουμε τον πίνακα ή τη φόρμα και κάνουμε **διπλό κλικ** πάνω του/της για να τον/την ανοίξουμε πρώτα σε προβολή φύλλου δεδομένων (**Datasheet View**) / προβολή φόρμας (**Form View**).
- (2) Ακολουθώντας, από την καρτέλα **Home** και την ομάδα **Find** επιλέγουμε το εικονίδιο **Find**.

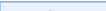


ή

Μπορούμε επίσης να χρησιμοποιήσουμε τον συνδυασμό πλήκτρων **CTRL + F**.

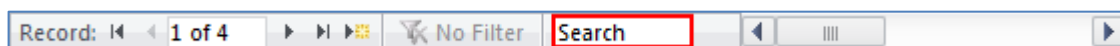
- (3) Από το παράθυρο που εμφανίζεται, στο πλαίσιο του **Find What:** πληκτρολογούμε τη λέξη ή τον αριθμό ή την ημερομηνία που αναζητούμε να βρούμε στις εγγραφές μας.



- (4) Από τη λίστα του **Look In:** επιλέγουμε το Current document και από τη λίστα του **Match:** επιλέγουμε το Any Part of Field.
- (5) Στη συνέχεια, κάνουμε κλικ στο κουμπί  για να μας πάει με αυτό τον τρόπο στην **πρώτη εγγραφή** που θα περιέχει αυτά τα δεδομένα από τις εγγραφές μας.
- (6) Τέλος, κάνοντας ξανά κλικ στο κουμπί  μας πάει στην **επόμενη εγγραφή** που θα περιέχει αυτά τα δεδομένα από τις εγγραφές μας.

ἡ

Από τη γραμμή πλοήγησης (**Navigation Area**) στο πλαίσιο της αναζήτησης (**Search**) πληκτρολογούμε τη λέξη ή τον αριθμό ή την ημερομηνία που αναζητούμε να βρούμε στις εγγραφές μας και μας πάει με αυτό τον τρόπο στην **πρώτη εγγραφή** που θα περιέχει αυτά τα δεδομένα από τις εγγραφές μας.

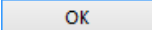


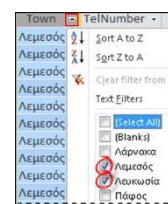
Ακολουθώντας, πατώντας το πλήκτρο  μας πάει στην **επόμενη εγγραφή** που θα περιέχει αυτά τα δεδομένα από τις εγγραφές μας μέχρι να φτάσει και στην τελευταία εγγραφή.

5. Χρήση Φίλτρων για την Αναζήτηση Πληροφοριών (Filter)

Ένας άλλος τρόπος **γρήγορης εύρεσης πληροφοριών** από έναν πίνακα ή από μια φόρμα είναι τα **φίλτρα (Filter)**. Αυτό μπορούμε να το πετύχουμε είτε όταν βρισκόμαστε σε **προβολή φύλλου δεδομένων (Datasheet View)** του πίνακα, είτε σε **προβολή φόρμας (Form View)** της φόρμας. Να σημειώσουμε ότι τα φίλτρα είναι και αυτός ένας **προσωρινός τρόπος εύρεσης πληροφοριών** και για αυτό τον λόγο **ΔΕΝ μπορούν να αποθηκευτούν** για μελλοντική χρήση.

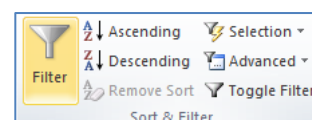
5.1 Δημιουργία Φίλτρων

- (1) Επιλέγουμε τον πίνακα ή τη φόρμα και κάνουμε **διπλό κλικ** πάνω του/της για να τον/τη ανοίξουμε πρώτα σε προβολή φύλλου δεδομένων (**Datasheet View**) / προβολή φόρμας (**Form View**).
- (2) Ακολούθως, κάνουμε κλικ στο **βελάκι**  δίπλα από το πεδίο το οποίο θέλουμε να φιλτράρουμε, (π.χ. *εάν θέλουμε να βρούμε από το πεδίο **Town** τις εγγραφές τις οποίες έχουν για πόλη τη Λεμεσό και τη Λευκωσία*) και επιλέγουμε τις δυο πόλεις.
- (3) Τέλος, κάνουμε κλικ στο κουμπί  για να εφαρμόσουμε το φίλτρο.

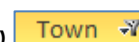


ή

Από την καρτέλα **Home** και την ομάδα **Sort & Filter** επιλέγουμε το εικονίδιο **Filter** για να μπορέσουμε να φιλτράρουμε με τον ίδιο τρόπο όπως και πιο πάνω.



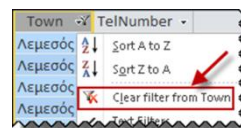
(Σημείωση: Θα εμφανιστεί το εικονίδιο του φίλτρου δίπλα από το πεδίο.)



5.2 Αφαίρεση Φίλτρων (Clear Filter)

(1) Επιλέγουμε τον πίνακα ή τη φόρμα και κάνουμε **διπλό κλικ** πάνω του/της για να τον/τη ανοίξουμε πρώτα σε προβολή φύλλου δεδομένων (Datasheet View) / προβολή φόρμας (Form View).

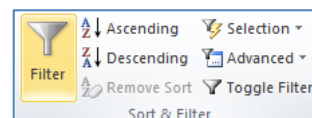
(2) Ακολούθως, κάνουμε κλικ στο **βελάκι**  δίπλα από το πεδίο του οποίου θέλουμε να αφαιρέσουμε το φιλτράρισμα.



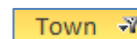
(3) Τέλος, επιλέγουμε το  **Clear filter from Town** για να αφαιρέσουμε το φίλτρο.

ή

Από την καρτέλα Home και την ομάδα Sort & Filter επιλέγουμε το εικονίδιο Filter για να μπορέσουμε να αφαιρέσουμε το φιλτράρισμα με τον ίδιο τρόπο όπως και πιο πάνω.



Σημείωση: Θα εμφανιστεί το εικονίδιο του φίλτρου δίπλα από το πεδίο.

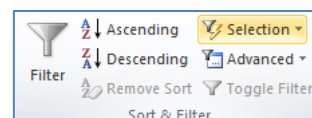


5.3 Φίλτρα κατ' Επιλογή (Selection Filter)

(1) Επιλέγουμε τον πίνακα ή τη φόρμα και κάνουμε **διπλό κλικ** πάνω του/της για να τον/την ανοίξουμε πρώτα σε προβολή φύλλου δεδομένων (Datasheet View) / προβολή φόρμας (Form View).

(2) Ακολούθως, κάνουμε κλικ σε ένα **πεδίο** ή κάνουμε διπλό κλικ σε **μέρος του πεδίου** που θέλουμε να φιλτράρουμε.

(3) Ακολούθως, από την καρτέλα Home και την ομάδα Sort & Filter επιλέγουμε το εικονίδιο Selection και επιλέγουμε στη συνέχεια, ένα από τα πιο κάτω ανάλογα με την περίπτωση:



Για κάθε είδος πεδίου που επιλέγουμε για φιλτράρισμα υπάρχουν και τα ανάλογα φίλτρα όπως το **TEXT Filters**, **NUMBER Filters** και **DATE Filters**.

Τα φίλτρα κατ' επιλογή εφαρμόζονται σε ένα πεδίο και έχουν τη μορφή:

Κριτήριο τιμή_σύγκρισης

Για παράδειγμα, έστω ότι θέλουμε να επιλέξουμε (να φιλτράρουμε) μόνο τις εγγραφές από τον πίνακα των μελών του κέντρου νεότητας (YouthMembers) για τις οποίες το επίθετο (Surname) τελειώνει σε «ου». Για να το πετύχουμε, στο πεδίο Surname προσθέτουμε φίλτρο (Home ⇒ Sort & Filter) επιλέγοντας το κριτήριο **Ends With** και την τιμή σύγκρισης **«ου»**.

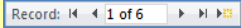
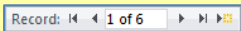
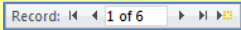
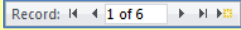
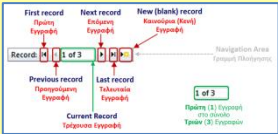
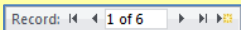
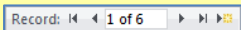
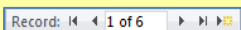
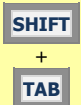
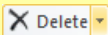
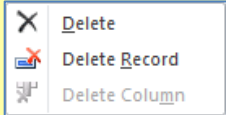
Ο πίνακας στην επόμενη σελίδα παρουσιάζει τα κριτήρια που μπορούν να χρησιμοποιηθούν για κάθε τύπο δεδομένων πεδίου.

Κριτήριο	Επιλέγει μόνο τις εγγραφές οι οποίες στο συγκεκριμένο πεδίο έχουν δεδομένα που ...	Εφαρμόζεται σε πεδία τύπου
Equals	<u>είναι ίσα</u> με την τιμή σύγκρισης	Text, Number, Date
Does Not Equal	<u>δεν είναι ίσα</u> με με την τιμή σύγκρισης	Text, Number, Date
Begins With	<u>αρχίζουν</u> με την τιμή σύγκρισης	Text
Does Not Begin With	<u>δεν αρχίζουν</u> με την τιμή σύγκρισης	Text
Contains	<u>έχουν μέσα</u> τους την τιμή σύγκρισης	Text
Does Not Contain	<u>δεν έχουν μέσα</u> τους την τιμή σύγκρισης	Text
Ends With	<u>τελειώνουν</u> με την τιμή σύγκρισης	Text
Does Not End With	<u>δεν τελειώνουν</u> με την τιμή σύγκρισης	Text
Less Than or Equal To	<u>είναι μικρότερα από ή ίσα</u> με την τιμή σύγκρισης	Number
Greater Than or Equal To	<u>είναι μεγαλύτερα από ή ίσα</u> με την τιμή σύγκρισης	Number
Between	<u>είναι ανάμεσα στις</u> τιμές σύγκρισης	Date, Number

Πιο κάτω δίνονται παραδείγματα εφαρμογής των φίλτρων με βάση τον πίνακα **YouthMembers**.

YouthMembers							
YouthMemberID	Firstname	Surname	Sex	Birthdate	Address	Town	TelNumber
1002	Αβραάμ	Κυριάκου	A	8/12/1999	Καντάρας 19	Λεμεσός	99651298
1015	Αχιλλέας	Αλέξη	A	8/13/1999	Καντάρας 15	Λεμεσός	99274730
1018	Ελένη	Άγγελου	Θ	2/5/1998	Καντάρας 11	Λευκωσία	99898162
1024	Ελεάνα	Πολύκαρπου	Θ	2/7/1998	Κυρηνείας 37	Λευκωσία	99610499
1027	Αλέξανδρος	Νικόλαου	A	8/15/1999	Αμμοχώστου 37	Λεμεσός	99434069
1052	Άννα	Νίκου	Θ	2/9/2000	Καντάρας 25	Λεμεσός	99216150
1087	Χρήστος	Γκέλα	A	2/29/2000	Καντάρας 7	Λευκωσία	99521594
1089	Χρίστος	Δήμου	A	3/13/2000	Αμμοχώστου 17	Λευκωσία	99551229
1091	Ελένη	Γιώργιου	Θ	2/6/1998	Μόρφου 6	Λεμεσός	99780312
1099	Άντρεα	Κυριάκου	Θ	8/20/1999	Κυρηνείας 7	Λεμεσός	99786239

Πεδίο	Κριτήριο	Τιμή	Εμφανίζει μόνο τις εγγραφές με YouthMemberID
Firstname	Equals To	«Ελένη»	1018 και 1091
Birthdate	Does Not Equal To	«8/12/1999»	1015, 1018, 1024, 1027, 1052, 1087, 1089, 1091 και 1099
Surname	Begins With	«Γ»	1087, και 1091
Address	Does Not Begin With	«Κ»	1027, 1089 και 1091
Town	Contains	«ΕΥΚΩ»	1018, 1024, 1087 και 1089
Town	Does Not Contain	«Εμ»	1018, 1024, 1087 και 1089
Surname	Ends With	«ΟΥ»	1002, 1018, 1024, 1027, 1052, 1089 και 1091
Surname	Does Not End With	«ΟΥ»	1015 και 1087
YouthMemberID	Less Than or Equal To	«1024»	1002, 1015, 1018 και 1024
Birthdate	Greater Than or Equal To	«9/2/2000»	1052, 1087 και 1089
YouthMemberID	Between	«1002» και «1027»	1002, 1015, 1018, 1024 και 1027

Υπόμνημα					
Περιγραφή (Description)	Καρτέλα (Tab)	Εικονίδιο (Icon)	Ομάδα (Group)	Παράθυρο Διαλόγου (Dialog Box)	Πλήκτρα (Keys)
Εισαγωγή Εγγραφής (New Record)	Home				
Γραμμή Πλοήγησης (Navigation Area)					
Πρώτη Εγγραφή (First record)					
Προηγούμενη Εγγραφή (Previous record)					
Τρέχουσα Εγγραφή (Current Record)	1 of 6				
Επόμενη Εγγραφή (Next record)					
Τελευταία Εγγραφή (Last record)					
Νέα Κενή Εγγραφή (New (blank) record)					
Μετακίνηση του δρομέα στην ίδια την εγγραφή (Επόμενο και Προηγούμενο)					
Μετακίνηση του δρομέα στην ίδια την εγγραφή (Επόμενο)					
Μετακίνηση του δρομέα στην ίδια την εγγραφή (Προηγούμενο)					
Διαγραφή Εγγραφής (Delete Record)	Home			 	

Γ6.5 Σχέσεις Πινάκων

Τι θα μάθουμε σήμερα:

- Να περιγράφουμε την ανάγκη χρήσης και το σκοπό των σχέσεων μεταξύ πινάκων σε μια βάση δεδομένων (*πλεονεκτήματα*)
- Να διακρίνουμε τα είδη των σχέσεων πινάκων «ένα προς ένα» και «ένα προς πολλά».

1. Εισαγωγή

Όπως ήδη γνωρίζουμε μια δεδομένων είναι μια **οργανωμένη συλλογή** από συσχετιζόμενα δεδομένα. Πιο συγκεκριμένα, σε μια Σχεσιακή Βάση Δεδομένων τα δεδομένα είναι οργανωμένα σε πίνακες οργανωμένους σε πεδία. Κάθε πίνακας αποθηκεύει δεδομένα για ένα συγκεκριμένο θέμα (αντικείμενο, πρόσωπο ή έννοια), οργανωμένα σε εγγραφές. Συνήθως μια βάση δεδομένων περιλαμβάνει περισσότερους από έναν πίνακες.

Παράδειγμα:

Η βάση δεδομένων για ένα σχολείο περιλαμβάνει τον πίνακα με τα δεδομένα των μαθητών, τον πίνακα με τα δεδομένα των καθηγητών, τον πίνακα με τα δεδομένα των μαθημάτων, τον πίνακα με τα δεδομένα των τμημάτων τον πίνακα με τα δεδομένα των απουσιών και πολλούς άλλους.

Σε μια (Σχεσιακή) βάση δεδομένων, οι διάφοροι πίνακες μπορούν να σχετίζονται και αυτοί μεταξύ τους με ειδικές σχέσεις οι οποίες ονομάζονται **Relationships**. Αυτές οι σχέσεις είναι που συνδέουν τους πίνακες μεταξύ τους και δημιουργούν την ανάγκη να ανήκουν στην ίδια βάση δεδομένων. Ο κάθε πίνακας σε μια βάση δεδομένων πιθανό να σχετίζεται με έναν ή περισσότερους άλλους πίνακες. Η κάθε σχέση εμφανίζεται συνήθως ως μια λογική σύνδεση μεταξύ δύο πινάκων και υποδηλώνεται με την παρουσία και στους δύο πίνακες πεδίων με κοινά δεδομένα.

Παράδειγμα:

Στη βάση δεδομένων του σχολείου, ο πίνακας των μαθητών σχετίζεται με τον πίνακα των τμημάτων (στον πίνακα των μαθητών, μαζί με τα προσωπικά τους στοιχεία είναι καταχωρημένο και το τμήμα στο οποίο ανήκουν). Επίσης, ο πίνακας των καθηγητών σχετίζεται με τα τμήματα στα οποία είναι υπεύθυνοι (στον πίνακα των τμημάτων συμπεριλαμβάνεται ο αριθμός αναφοράς του/της κάθε καθηγητή/τριας). Αντίστοιχα ο πίνακας των απουσιών περιλαμβάνει τον αριθμό μητρώου του/της μαθητή/τριας που έκανε την απουσία, μαζί με την ημερομηνία και την περίοδο.

Αυτή η ύπαρξη αυτών των σχέσεων μεταξύ των πινάκων, αλλά και ο απλός τρόπος αναπαράστασής τους είναι ένα από τους κύριους λόγους που οι Σχεσιακές Βάσεις Δεδομένων έχουν επικρατήσει. Στην πραγματικότητα υπάρχει ένα στέρεο θεωρητικό υπόβαθρο για τις Σχεσιακές Βάσεις Δεδομένων, το οποίο οδήγησε στη διατύπωση συστηματικών μεθόδων στην υλοποίησή τους. Αυτό μας δίνει «συνταγές» για τον καλύτερο τρόπο διαχωρισμού των δεδομένων σε πίνακες και τον καλύτερο τρόπο διατύπωσης των σχέσεων μεταξύ τους, τα οποία οδηγούν σε σημαντικά πλεονεκτήματα.

2. Πλεονεκτήματα που προκύπτουν από τις Σχέσεις μεταξύ των Πινάκων

Η δημιουργία αυτών των σχέσεων ανάμεσα σε πίνακες μιας βάσης δεδομένων, μάς επιτρέπει να αντλήσουμε τα σχετιζόμενα στοιχεία που ανήκουν στους διάφορους πίνακες με διάφορους τρόπους ανάλογα με τις ανάγκες που προκύπτουν (συνήθως με ένα κατάλληλο ερώτημα, query).

Η ύπαρξη σχετιζόμενων πινάκων έχει ως αποτέλεσμα να έχουμε τα πιο κάτω πλεονεκτήματα:

- Δεν χρειάζεται να καταχωρούμε τα ίδια δεδομένα πολλές φορές. Καταχωρούνται από μία φορά στον κατάλληλο πίνακα και τα ανακτούμε όταν χρειάζεται μέσω των σχέσεων των πινάκων.
- Γίνεται ευκολότερη η διαχείριση των δεδομένων (*ενημέρωση, διαγραφή κ.λπ.*), αφού το κάθε δεδομένο καταχωρείται σε έναν μόνο πίνακα.
- Ο συνολικός όγκος των δεδομένων στη βάση δεδομένων είναι όσο μικρότερος γίνεται, αφού δεν υπάρχει αχρείαστη επανάληψη δεδομένων.
- Μειώνεται το περιθώριο σφαλμάτων αφού σε κάθε αλλαγή ή διαγραφή κάποιας εγγραφής επηρεάζεται συνήθως μόνο ένας πίνακας.
- Υπάρχει περισσότερη σαφήνεια στην οργάνωση των δεδομένων, αφού τόσο η έννοια του πίνακα, όσο και ο τρόπος συσχέτισης είναι κατανοητά, ακόμη και σε μη επαγγελματίες της Πληροφορικής και της Επιστήμης των Ηλεκτρονικών Υπολογιστών.
- Εύκολα μπορούν να προστεθούν νέοι πίνακες ή να δημιουργηθούν νέες σχέσεις μεταξύ των πινάκων σε μια βάση δεδομένων, καθώς και να συνδυαστούν οι πίνακες και οι σχέσεις τους με νέους τρόπους για την εξαγωγή νέων πληροφοριών.

3. Ο ρόλος του Πρωτεύοντος Κλειδιού στον συσχετισμό των πινάκων

Πρωτεύον Κλειδί (**Primary Key**) έχει οριστεί να είναι το **πεδίο** (ή τα πεδία) που προσδιορίζει μοναδικά κάθε εγγραφή στον πίνακα (δηλαδή έχει ξεχωριστή, διαφορετική τιμή από οποιαδήποτε άλλη εγγραφή του πίνακα).

Το Πρωτεύον Κλειδί χρησιμοποιείται συνήθως για τη δημιουργία της σχέσης μεταξύ **κύριου πίνακα** (*Πρωτεύων Πίνακας - Primary Table*) και **συσχετιζόμενου πίνακα** (*Δευτερεύων Πίνακας - Secondary or Related Table*), ώστε να ξέρουμε σίγουρα σε ποια εγγραφή αναφερόμαστε.

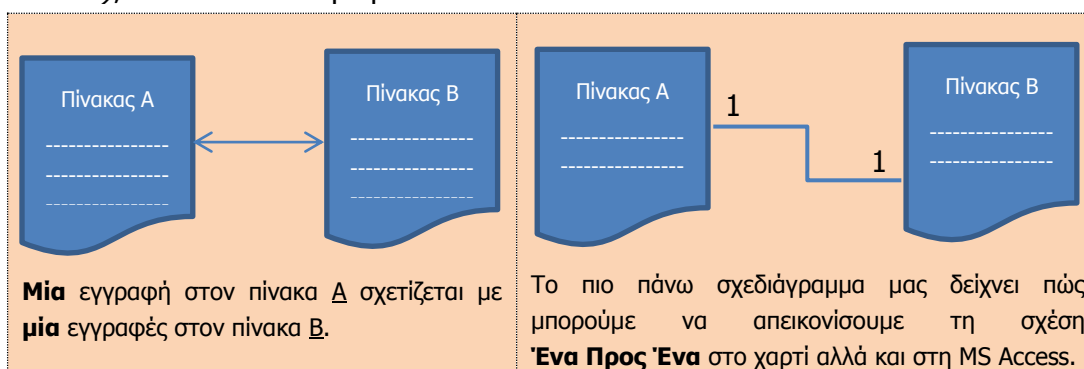
4. Είδη Ειδικών Σχέσεων (Relationships)

Τα **είδη** των ειδικών σχέσεων που υπάρχουν μεταξύ των πινάκων είναι:

- Ένα-Προς-Ένα (**One-To-One**)
- Ένα-Προς-Πολλά (**One-To-Many**)
- Πολλά-Προς-Πολλά (**Many-To-Many**) (Θα παρουσιαστεί αργότερα)

4.1 Ένα-Προς-Ένα (One-To-One, 1-1)

Κάθε εγγραφή από τον ΠΡΩΤΟ πίνακα (**Πρωτεύων Πίνακας - Primary Table**) σχετίζεται με μόνο μία εγγραφή από τον ΔΕΥΤΕΡΟ πίνακα (**Δευτερεύων Πίνακας - Secondary Table**), αλλά και αντίστροφα.



Παράδειγμα σε μια βάση δεδομένων **σχολείου**:

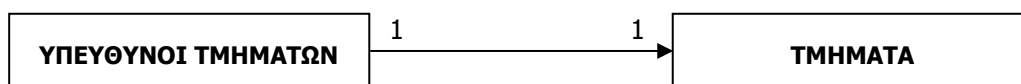
- Πίνακες:
 - **ΥΠΕΥΘΥΝΟΙ ΤΜΗΜΑΤΩΝ**
 - **ΤΜΗΜΑΤΑ**

Για κάθε **τμήμα** του σχολείου υπάρχει **ένας υπεύθυνος καθηγητής** και θέλουμε να γνωρίζουμε τα στοιχεία του. Άρα υπάρχει η ανάγκη του συσχετισμού των δύο πινάκων.

- (1) Αρχίζοντας πάντοτε από το ΕΝΑ αντικείμενο του ΠΡΩΤΟΥ πίνακα, ρωτάτε με πόσα αντικείμενα του ΔΕΥΤΕΡΟΥ πίνακα μπορεί να έχει σχέση, δηλαδή:

1 ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ με πόσα ΤΜΗΜΑΤΑ μπορεί να έχει σχέση;

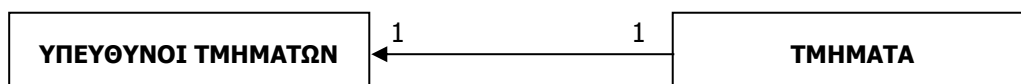
= 1 ΤΜΗΜΑ



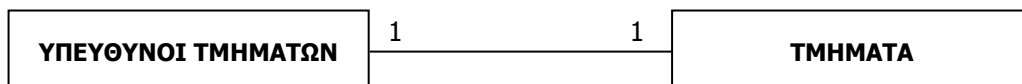
- (2) Στη συνέχεια, αρχίζοντας πάλιν από το ΕΝΑ αντικείμενο του ΔΕΥΤΕΡΟΥ πίνακα, ρωτάτε με πόσα αντικείμενα του ΠΡΩΤΟΥ πίνακα μπορεί να έχει σχέση:

1 ΤΜΗΜΑ με πόσους ΥΠΕΥΘΥΝΟΥΣ ΚΑΘΗΓΗΤΕΣ μπορεί να έχει σχέση;

= 1 ΥΠΕΥΘΥΝΟ ΚΑΘΗΓΗΤΗ



(3) Άρα το τελικό αποτέλεσμα της πιο πάνω διεργασίας είναι:



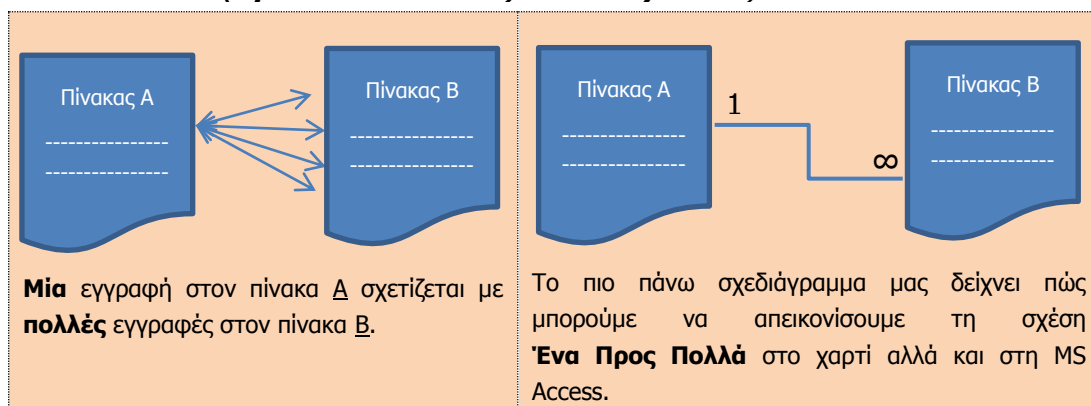
Άλλα **παραδείγματα** σχέσεων 'Ένα-Προς-Ένα' από την καθημερινότητά μας είναι:

- η σχέση που υπάρχει μεταξύ μιας **Πόλης** και ενός **Δημάρχου**.
- η σχέση που υπάρχει μεταξύ μιας **Χώρας** και του **Προέδρου** της.

Για να μπορεί να δημιουργηθεί αυτό το είδος σχέσης μεταξύ των πινάκων θα πρέπει να υπάρχει στον έναν από τους δυο ένα πεδίο που να αντιστοιχεί στο πρωτεύον κλειδί του άλλου (π.χ. στον πίνακα των τμημάτων να υπάρχει ένα πεδίο όπου θα καταχωρούνται τιμές οι οποίες αντιστοιχούν σε αυτές που υπάρχουν στο πρωτεύον κλειδί του υπεύθυνου τμήματος, ή αντίστοιχα, στον πίνακα των υπεύθυνων τμήματος να υπάρχει ένα πεδίο όπου θα καταχωρείται τιμές οι οποίες αντιστοιχούν σε αυτές που υπάρχουν στο πρωτεύον κλειδί του τμήματος, για να συσχετίζονται οι αντίστοιχες εγγραφές).

4.2 Ένα-Προς-Πολλά (One-To-Many, 1-∞)

Κάθε εγγραφή από τον ΠΡΩΤΟ πίνακα (**Πρωτεύων Πίνακας - Primary Table**) σχετίζεται με πολλές εγγραφές από το ΔΕΥΤΕΡΟ πίνακα (**Δευτερεύων Πίνακας - Secondary Table**), αλλά κάθε εγγραφή από το ΔΕΥΤΕΡΟ πίνακα, σχετίζεται με μόνο μία εγγραφή από τον ΠΡΩΤΟ πίνακα (**Πρωτεύων Πίνακας - Primary Table**).



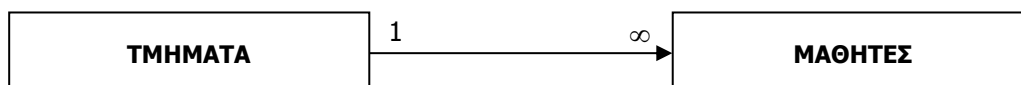
Παράδειγμα σε μια βάση δεδομένων **σχολείου**:

- Πίνακες:
 - **ΤΜΗΜΑΤΑ**
 - **ΜΑΘΗΤΕΣ**

Για κάθε **τμήμα** θέλουμε να γνωρίζουμε τα στοιχεία των **μαθητών** που ανήκουν σε αυτό, αλλά και για κάθε **μαθητή** τα στοιχεία του **τμήματος** που ανήκει, για αυτό υπάρχει η ανάγκη του συσχετισμού των δύο πινάκων.

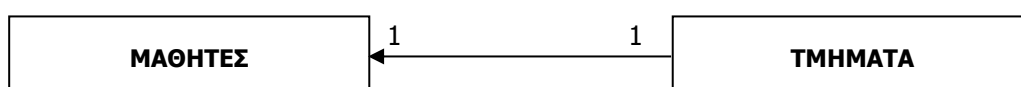
(1) Αρχίζοντας πάντοτε από το ΕΝΑ αντικείμενο του ΠΡΩΤΟΥ πίνακα, ρωτάτε με πόσα αντικείμενα του ΔΕΥΤΕΡΟΥ πίνακα μπορεί να έχει σχέση, δηλαδή:

1 ΤΜΗΜΑ με πόσους ΜΑΘΗΤΕΣ μπορεί να έχει σχέση;
= ΠΟΛΛΟΥΣ ΜΑΘΗΤΕΣ

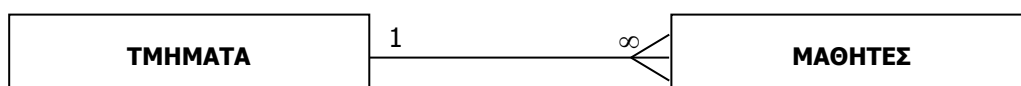


Στη συνέχεια, αρχίζοντας πάλι από το ΕΝΑ αντικείμενο του ΔΕΥΤΕΡΟΥ πίνακα, ρωτάτε με πόσα αντικείμενα του ΠΡΩΤΟΥ πίνακα μπορεί να έχει σχέση:

1 ΜΑΘΗΤΗΣ μπορεί να έχει σχέση με πόσα ΤΜΗΜΑΤΑ;
= 1 ΤΜΗΜΑ



(2) Άρα το τελικό αποτέλεσμα της πιο πάνω διεργασίας είναι:



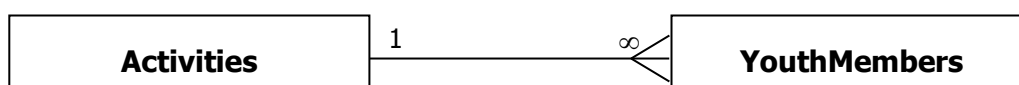
Άλλα **παράδειγματα** σχέσεων Ένα-Προς-Πολλά από την καθημερινότητά μας είναι:

- η σχέση της **Μητέρας** με τα **Παιδιά** της. Μία μητέρα μπορεί να έχει πολλά παιδιά, ένα παιδί όμως, έχει μόνο μία βιολογική μητέρα.
- η σχέση που υπάρχει μεταξύ μιας **Πόλης** και των **Κατοίκων** της. Η Πόλη έχει πολλούς Κατοίκους, ο κάθε Κάτοικος όμως ζει σε μία μόνο Πόλη.

Για να μπορεί να δημιουργηθεί αυτό το είδος σχέσης μεταξύ των πινάκων θα πρέπει να υπάρχει στον πίνακα από τη μεριά του «Πολλά» ένα πεδίο που να αντιστοιχεί στο πρωτεύον κλειδί του πίνακα από τη μεριά του «Ένα» (π.χ. στον πίνακα των μαθητών να υπάρχει ένα πεδίο όπου θα καταχωρούνται τιμές οι οποίες αντιστοιχούν σε αυτές που υπάρχουν στο πρωτεύον κλειδί του τμήματος για να συσχετίζονται οι αντίστοιχες εγγραφές).

Παράδειγμα:

Για να γίνει καλύτερα κατανοητός ο μηχανισμός των σχέσεων μεταξύ των πινάκων, θα χρησιμοποιήσουμε το παράδειγμα της βάσης δεδομένων του Κέντρου Νεότητας. Υπάρχει ο πίνακας των Μελών («YouthMembers») και ο πίνακας των Δραστηριοτήτων (Activities). Κάθε μέλος μπορεί να επιλέξει μία δραστηριότητα, ενώ κάθε δραστηριότητα περιλαμβάνει πολλά μέλη:



Πιο κάτω παρουσιάζονται οι πίνακες αυτοί, όπου σημειώνονται με διαφορετικό χρώμα οι εγγραφές από τους δύο πίνακες οι οποίες σχετίζονται μεταξύ τους. Φαίνεται ξεκάθαρα η σχέση «Ένα-Προς-Πολλά», όπως διατυπώθηκε πιο πάνω:

YouthMembers						
YouthMemberID	Firstname	Surname	Sex	Birthdate	Town	ActivityCode
1036	Αντώνης	Ηλία	A	19/2/2000	Λεμεσός	10
1037	Αργύρης	Ανδρέα	A	20/2/2000	Λεμεσός	10
1038	Αρτεμισία	Αντωνίου	Θ	21/2/2000	Λεμεσός	9
1039	Βασίλης	Πέτρου	A	22/2/2000	Λεμεσός	8
1040	Βασίλειος	Κυριάκου	A	23/2/2000	Λεμεσός	7
1041	Μπόρις	Σαμπών	A	24/2/2000	Λευκωσία	6
1042	Καλλιόπη	Μάριου	Θ	25/2/2000	Λευκωσία	5
1043	Καλλιόπη	Ιωάννη	Θ	26/2/2000	Λευκωσία	4
1044	Καρολίνα	Άμερ	Θ	27/2/2000	Λευκωσία	3
1045	Χαραλαμπία	Τάσου	Θ	28/2/2000	Λευκωσία	2
1046	Χρήστος	Γκέλα	A	29/2/2000	Λευκωσία	1
1047	Χρήστος	Ηρακλή	A	1/3/2000	Λευκωσία	3

Activities		
ActivityCode	Description	RegistrationFees
1	Ποδόσφαιρο	100,00 €
2	Καλαθόσφαιρα	80,00 €
3	Πετόσφαιρα	75,00 €
4	Επιτραπέζια Αντισφαίριση	60,00 €
5	Σκάκι	75,00 €
6	Σκι	100,00 €
7	Χειροσφαίριση	60,00 €
8	Πασσία	180,00 €
9	Τένις	120,00 €
10	Μπάντμιντον	90,00 €

Βασικές Έννοιες

Πρωτεύον κλειδί (Primary Key):

Πεδίο ή συνδυασμός πεδίων ενός πίνακα του οποίου το περιεχόμενο καθορίζει με μοναδικό τρόπο την κάθε εγγραφή του πίνακα. Χρησιμοποιείται συνήθως για τη δημιουργία μιας σχέσης μεταξύ των εγγραφών δύο πινάκων, ώστε να γνωρίζουμε ακριβώς ποιες εγγραφές σχετίζονται μεταξύ τους και με ποιο τρόπο.

Ειδικές Σχέσεις πινάκων (Relationships):

Κάθε πίνακας αποτελεί ομάδα από στοιχεία που αναφέρονται σε κάποιο πρόσωπο, αντικείμενο ή έννοια. Οι εγγραφές σε έναν πίνακα σε μια βάση δεδομένων μπορεί σχετίζονται με εγγραφές σε έναν άλλο πίνακα. Αυτή η ειδική σχέση ονομάζεται **Relationship**.

Είδη Ειδικών Σχέσεων:

Ένα Προς Ένα (**One-To-One, 1-1**)
Ένα Προς Πολλά (**One-To-Many, 1-∞**)

Ένα Προς Ένα (One-To-One) (1-1):

Κάθε εγγραφή από τον ΠΡΩΤΟ πίνακα σχετίζεται με μόνο μία εγγραφή από τον ΔΕΥΤΕΡΟ πίνακα, αλλά και αντίστροφα.

Ένα Προς Πολλά (One-To-Many) (1-∞):

Κάθε εγγραφή από τον ΠΡΩΤΟ πίνακα σχετίζεται με πολλές εγγραφές από το ΔΕΥΤΕΡΟ πίνακα, αλλά κάθε εγγραφή από τον ΔΕΥΤΕΡΟ πίνακα, σχετίζεται με μόνο μία εγγραφή από τον ΠΡΩΤΟ πίνακα.

Γ6.6 Δημιουργία και Διαγραφή Σχέσεων Μεταξύ δύο Πινάκων

Τι θα μάθουμε σήμερα:

- Να δημιουργούμε και να διαγράφουμε σχέσεις μεταξύ δύο πινάκων
- Να καταχωρούμε εγγραφές στο δεύτερο συσχετιζόμενο πίνακα.

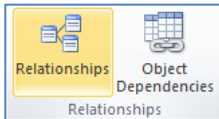
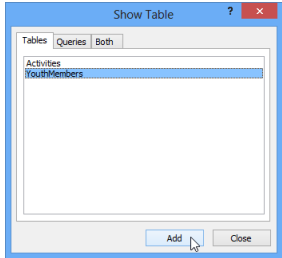
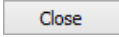
1. Εισαγωγή

Έχουμε ήδη μάθει πώς να δημιουργούμε σχέσεις μεταξύ των πινάκων πάνω στο χαρτί. Σε αυτό το μάθημα θα εξηγηθεί ο τρόπος που θα δημιουργηθούν αυτές οι σχέσεις μεταξύ των πινάκων στη Microsoft Access.

Για να δημιουργήσουμε αυτές τις ειδικές σχέσεις (**Relationships**) πρέπει να έχουμε ήδη δημιουργημένη τη βάση δεδομένων με τουλάχιστον δύο πίνακες που σχετίζονται μεταξύ τους. Οι σχέσεις σε μια βάση δεδομένων συνήθως, ορίζονται **πριν** την καταχώριση δεδομένων στους πίνακες. Εάν καταχωρίσουμε δεδομένα σε έναν πίνακα και στη συνέχεια, επιχειρήσουμε να συμπεριλάβουμε τον πίνακα σε κάποια σχέση, τότε μπορεί να αντιμετωπίσουμε προβλήματα στη δημιουργία της σχέσης, όταν στα πεδία τα οποία επιχειρούμε να συσχετίσουμε υπάρχουν σε κάποιες εγγραφές του ενός πίνακα τιμές οι οποίες δεν αντιστοιχούν με τις τιμές στο δεύτερο πίνακα.

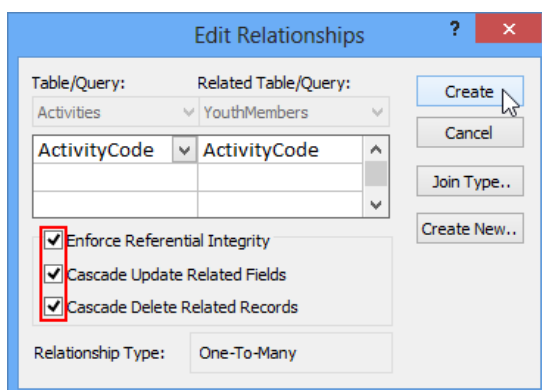
2. Δημιουργία σχέσης Ένα-Προς-Πολλά μεταξύ δύο πινάκων (One-To-Many, 1-∞)

Η διαδικασία που ακολουθούμε για να δημιουργήσουμε μια σχέση μεταξύ δύο πινάκων είναι η εξής:

- (1) Αποφασίζουμε πρώτα ποιοι πίνακες θα συμπεριληφθούν στη σχέση που θέλουμε να χρησιμοποιήσουμε και βεβαιωνόμαστε ότι αυτοί οι πίνακες είναι **κλειστοί**.
- (2) Από την καρτέλα Database Tools και την ομάδα Relationships επιλέγουμε το εικονίδιο Relationships για να ανοίξουμε το παράθυρο των σχέσεων. Μαζί του ανοίγει και το πλαίσιο διαλόγου εμφάνισης πινάκων, και έτσι μπορούμε να δημιουργήσουμε τη σχέση που θέλουμε να χρησιμοποιήσουμε.
- (3) Στη συνέχεια, από το πλαίσιο διαλόγου εμφάνισης πινάκων επιλέγουμε τον/τους **πίνακα/ες** ή το/τα **ερώτημα/τα** που θέλουμε να χρησιμοποιήσουμε και κάνουμε κλικ στο κουμπί  για να το/τα **προσθέσουμε** στο παράθυρο των σχέσεών μας.
- (4) Ακολούθως, κάνουμε κλικ στο κουμπί  για να **κλείσουμε** το πλαίσιο διαλόγου εμφάνισης πινάκων.
- (5) Μετά, επιλέγουμε το πεδίο που είναι το πρωτεύον κλειδί (**ActivityCode**) από τον πρωτεύοντα πίνακα (**Activities**) της σχέσης, το **σύρουμε** πάνω στο συσχετιζόμενο

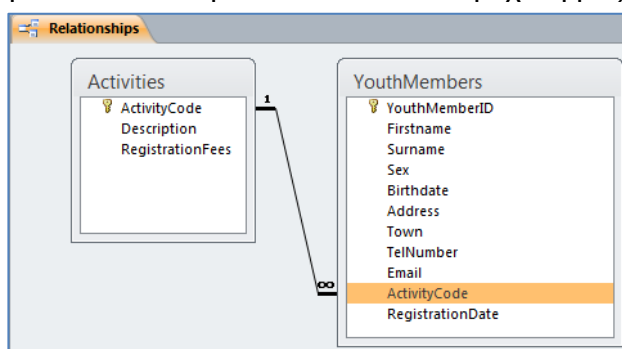
πεδίο (**ActivityCode**) του δευτερεύοντος πίνακα (**YouthMembers**) και το αφήνουμε.

- (6) Στο παράθυρο της επεξεργασίας των σχέσεων που εμφανίζεται, επιλέγουμε τα πιο κάτω:

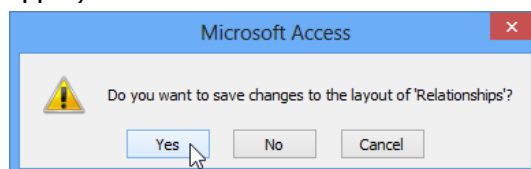
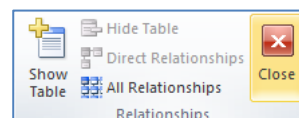


- **Enforce Referential Integrity** (Ενεργοποίηση Ακεραιότητας Αναφορών) - Το επιλέγουμε για να βεβαιωθούμε ότι τα συνδεδεμένα πεδία δεν περιέχουν τιμές που δεν ταιριάζουν.
- **Cascade Update Related Fields** (Διαδοχική Ενημέρωση Συσχετιζόμενων Πεδίων) - Το επιλέγουμε για να ενημερωθούν όλες οι συσχετιζόμενες εγγραφές σε περίπτωση που αλλάξουμε την τιμή του πρωτεύοντος κλειδιού μιας εγγραφής.
- **Cascade Delete Related Records** (Διαδοχική Διαγραφή Συσχετιζόμενων Εγγραφών) - Το επιλέγουμε όταν στην περίπτωση που διαγράψουμε την εγγραφή στον πρώτο πίνακα, επιθυμούμε να διαγράφονται και όλες οι συσχετιζόμενες εγγραφές στον άλλο πίνακα.

- (7) Ακολούθως, κάνουμε κλικ στο κουμπί **Create** και η σχέση μας θα δημιουργηθεί.



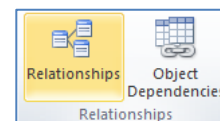
- (8) Τέλος, από την καρτέλα **Relationship Tools**→**Design** και την ομάδα **Relationships** επιλέγουμε το εικονίδιο **Close** για να κλείσουμε το παράθυρο των σχέσεων και αποθηκεύουμε στη συνέχεια, τη σχέση μας.



3. Διαγραφή Σχέσης μεταξύ δύο πινάκων (Delete Relationship)

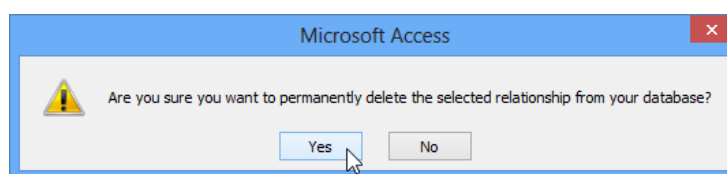
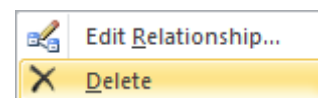
Η διαδικασία που ακολουθούμε για να διαγράψουμε μια σχέση μεταξύ δύο πινάκων είναι η εξής:

- (1) Από την καρτέλα Database Tools και την ομάδα Relationships επιλέγουμε το εικονίδιο Relationships για να ανοίξουμε το παράθυρο των σχέσεων.



- (2) Στη συνέχεια, κάνουμε **δεξί κλικ** πάνω στη γραμμή της σχέσης που συνδέει τους δύο πίνακες που θέλουμε να διαγράψουμε.

- (3) Ακολούθως, κάνουμε κλικ στο κουμπί **Delete**, επιβεβαιώνουμε την πρόθεσή μας για διαγραφή της σχέσης, κάνοντας κλικ στο κουμπί **Yes** και η σχέση μας έτσι **διαγράφεται**.

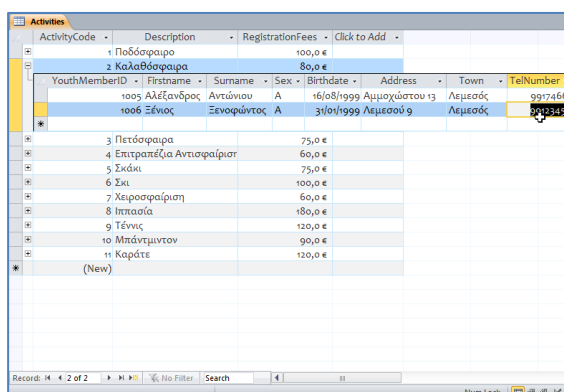


4. Καταχώριση Εγγραφών στον δεύτερο συσχετιζόμενο πίνακα

Για να καταχωρίσουμε εγγραφές στον δεύτερο συσχετιζόμενο πίνακα μπορούμε να ακολουθήσουμε την ίδια διαδικασία όπως και πριν, όταν οι πίνακες δεν σχετίζονταν. Μας παρέχεται όμως επιπρόσθετα, η δυνατότητα από τον πρώτο πίνακα να βλέπουμε και να καταχωρούμε στοιχεία στον δεύτερο συσχετιζόμενο πίνακα όπως πιο κάτω:

Ανοίγουμε και βλέπουμε τις εγγραφές του δεύτερου πίνακα πατώντας στο συν (+) που εμφανίζεται μπροστά από κάθε εγγραφή του πρώτου πίνακα. Έτσι, στο παράδειγμά μας, ανοίγοντας τον πίνακα δραστηριοτήτων **Activities** μπορούμε να βλέπουμε και τα μέλη που ανήκουν στην κάθε δραστηριότητα. Επίσης, μπορούμε να εισάγουμε νέα μέλη στον πίνακα **YouthMembers** μέσω της προβολής του πίνακα δραστηριοτήτων **Activities**.

- (1) Επιλέγουμε τον πρωτεύοντα πίνακα (Activities) της σχέσης και κάνουμε **διπλό κλικ** πάνω του για να τον ανοίξουμε σε προβολή φύλλου δεδομένων (Datasheet View).
- (2) Ακολούθως, κάνουμε κλικ στο κουμπί που βρίσκεται αριστερά από την εγγραφή που θέλουμε να καταχωρίσουμε τις σχετικές εγγραφές και **γράφουμε** την εγγραφή που θέλουμε να χρησιμοποιήσουμε.

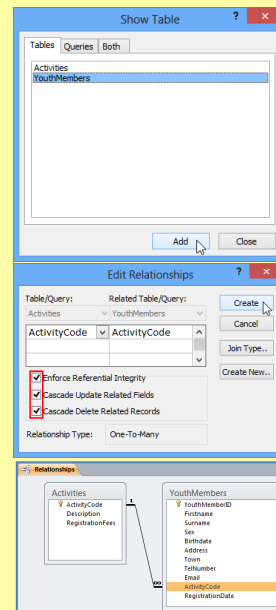
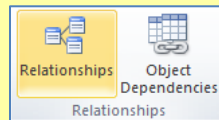
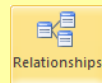


Υπόμνημα

Περιγραφή
(Description)**Καρτέλα**
(Tab)**Εικονίδιο**
(Icon)**Ομάδα**
(Group)**Παράθυρο Διαλόγου**
(Dialog Box)**Πλήκτρα**
(Keys)

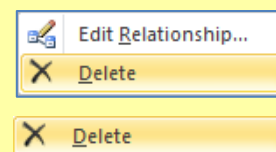
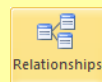
Δημιουργία Σχέσης
Ένα Προς Πολλά
(**One-To-Many**,
1-∞)

Database Tools

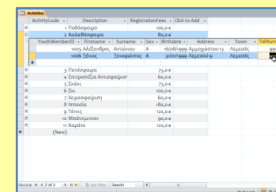


Διαγραφή Σχέσης
μεταξύ δύο πινάκων
(**Delete**
Relationship)

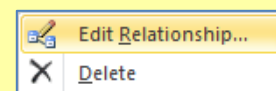
Database Tools



Καταχώριση
Εγγραφών
στο δεύτερο
συσχετιζόμενο πίνακα

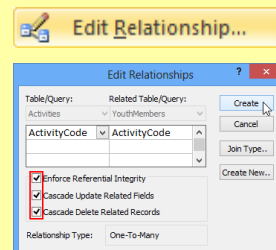
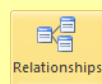


Δεξί κλικ



Επεξεργασία Σχέσεων
(**Edit Relationship**)

Database Tools



Γ6.7 Φόρμες Εισαγωγής Δεδομένων

Τι θα μάθουμε σήμερα:

- ◆ Να περιγράψουμε την ανάγκη χρήσης της φόρμας εισαγωγής δεδομένων (*πλεονεκτήματα*)
- ◆ Να δημιουργούμε μια απλή φόρμα εισαγωγής δεδομένων
- ◆ Να καταχωρούμε, να επεξεργαζόμαστε και να διαγράφουμε εγγραφές σε έναν πίνακα χρησιμοποιώντας τη φόρμα εισαγωγής δεδομένων
- ◆ Να μετακινούμαστε μεταξύ των εγγραφών σε φόρμα (*Επόμενη/Προηγούμενη, Πρώτη/Τελευταία*).

1. Φόρμα Εισαγωγής Δεδομένων (Form)

Ένας πιο **φιλικός** και πιο **εύκολος** τρόπος για την καταχώριση, διαγραφή και επεξεργασία των δεδομένων σε μια βάση δεδομένων, είναι η χρήση των φορμών εισαγωγής δεδομένων (**Forms**).

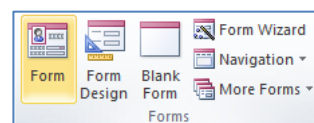
Κάθε φόρμα είναι συνδεδεμένη με έναν ή περισσότερους πίνακες, μέσα στους οποίους καταχωρεί τα δεδομένα (εγγραφές). Επίσης, εμφανίζει από τους πίνακες τα ήδη καταχωρημένα δεδομένα για σκοπούς πληροφόρησης ή και τροποποίησης.

Η δυνατότητα σχεδίασης και μορφοποίησης μιας φόρμας με βάση τις ανάγκες του χρήστη της βάσης δεδομένων, την κάνει πολύ πιο εύχρηστη και πιο φιλική προς τον χρήστη.

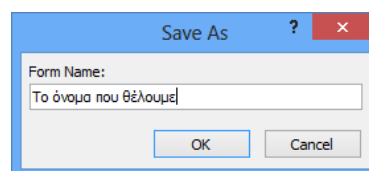
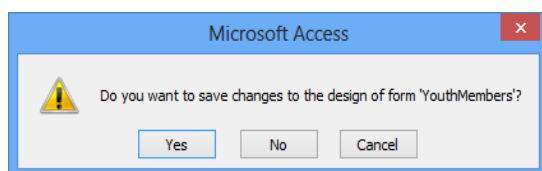
2. Δημιουργία Φόρμας με το Εργαλείο Αυτόματης Φόρμας (Form)

(1) Επιλέγουμε τον πίνακα.

(2) Από την καρτέλα **Create** και την ομάδα **Forms** επιλέγουμε το εικονίδιο **Form** και η φόρμα μας δημιουργείται με έναν **αυτόματο** σχετικά τρόπο.



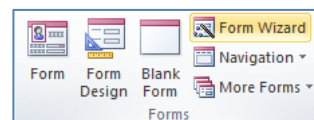
(3) Τέλος, κλείνουμε τη φόρμα, της καθορίζουμε ένα **περιγραφικό όνομα** και αποθηκεύουμε τη φόρμα μας.



3. Δημιουργία Φόρμας με τη χρήση του Οδηγού Φορμών (Form Wizard)

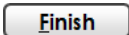
(1) Επιλέγουμε τον πίνακα.

(2) Από την καρτέλα **Create** και την ομάδα **Forms** επιλέγουμε το εικονίδιο **Form Wizard**.



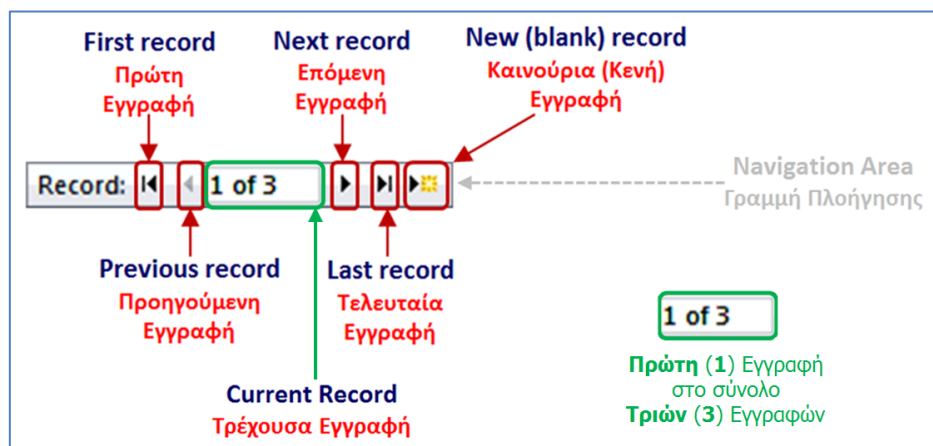
(3) Ακολουθώντας, κάνουμε κλικ στο κουμπί **>>** για να μεταφέρουμε ΟΛΑ τα πεδία στο πλαίσιο **Selected Fields:** ή κάνουμε κλικ στο κουμπί **>** για να μεταφέρουμε ΕΝΑ προς ΕΝΑ τα πεδία στο πλαίσιο **Selected Fields:** και μετά κάνουμε κλικ στο κουμπί **Next >** για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.

(4) Στη συνέχεια, επιλέγουμε την **επιθυμητή διάταξη** για τη φόρμα μας και κάνουμε ξανά κλικ στο κουμπί **Next >** για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.

- (5) Τέλος, καθορίζουμε ένα **περιγραφικό όνομα** και κάνουμε κλικ στο κουμπί  **αποθηκεύσουμε** τη φόρμα μας.

4. Εισαγωγή, Επεξεργασία και Μετακίνηση μεταξύ των εγγραφών σε μια φόρμα εισαγωγής δεδομένων

- (1) Επιλέγουμε τη φόρμα και κάνουμε **διπλό κλικ** πάνω της για να την ανοίξουμε πρώτα σε προβολή φόρμας (**Form View**).
- (2) Ακολούθως, κάνουμε κλικ σε ένα από τα εικονίδια της γραμμής πλοήγησης πιο κάτω:



5. Διαγραφή Εγγραφής σε μια φόρμα (Delete Record)

- (1) Επιλέγουμε τη φόρμα και κάνουμε **διπλό κλικ** πάνω της για να την ανοίξουμε πρώτα σε προβολή φόρμας (**Form View**).
- (2) Στη συνέχεια, επιλέγουμε την **εγγραφή** που θέλουμε να διαγράψουμε από τη φόρμα μας.
- (3) Από την καρτέλα **Home** και την ομάδα **Records** επιλέγουμε το εικονίδιο **Delete**.
- (4) Ακολούθως, κάνουμε κλικ στο κουμπί  για να **διαγράψουμε** την εγγραφή που θέλουμε από τη φόρμα μας.



6. Προβολές για Φόρμα (Views)

6.1 Προβολή Σχεδίασης (Design View)

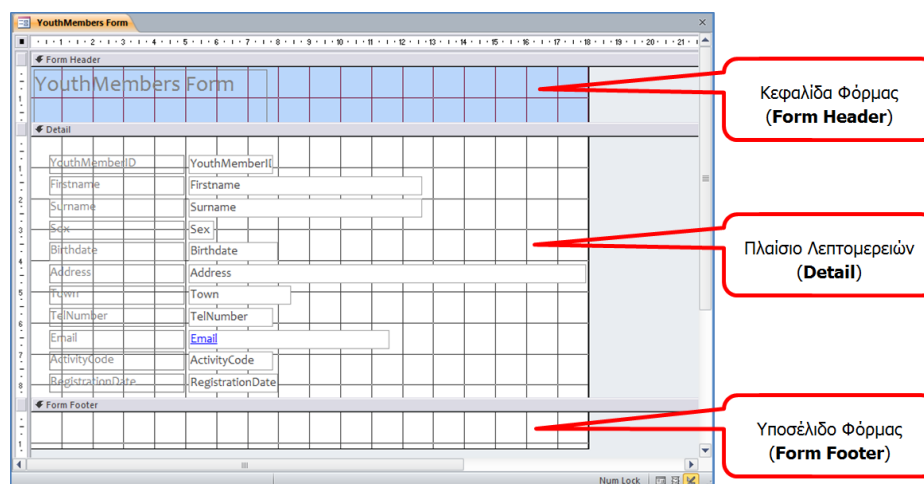
- (1) Από την περιοχή όλων των αντικειμένων της εφαρμογής (**All Access Objects**) κάτω από τον τίτλο **Forms**, όπου εμφανίζονται όλες οι φόρμες που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε τη φόρμα που θέλουμε και κάνουμε **διπλό κλικ** πάνω της για να την ανοίξουμε πρώτα σε προβολή φόρμας (**Form View**).
- (2) Ακολούθως, από την καρτέλα **Home** και την ομάδα **Views** επιλέγουμε το εικονίδιο **Design View** για να ανοίξουμε τη φόρμα μας σε προβολή σχεδίασης.



Κάνουμε κλικ στο κουμπί  από τη γραμμή κατάστασης στην κάτω δεξιά γωνιά της οθόνης μας.

ή

Από την περιοχή όλων των αντικειμένων της εφαρμογής (All Access Objects) κάτω από τον τίτλο **Forms**, όπου εμφανίζονται όλες οι φόρμες που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε τη φόρμα που θέλουμε και κάνουμε **δεξί κλικ** πάνω της και επιλέγουμε την εντολή **Design View** για να την ανοίξει σε προβολή σχεδίασης.



6.2 Προβολή Φόρμας (Form View)

- (1) Από την περιοχή όλων των αντικειμένων της εφαρμογής (All Access Objects) κάτω από τον τίτλο **Forms**, όπου εμφανίζονται όλες οι φόρμες που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε τη φόρμα που θέλουμε και κάνουμε **διπλό κλικ** πάνω της για να την ανοίξουμε σε προβολή φόρμας (Form View).

ή

Από την καρτέλα Home και την ομάδα Views επιλέγουμε το εικονίδιο Form View για να ανοίξουμε τη φόρμα μας σε προβολή φόρμας.



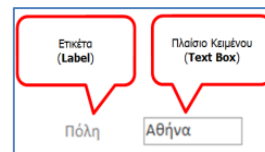
ή

Κάνουμε κλικ στο κουμπί  από τη γραμμή κατάστασης στην κάτω δεξιά γωνιά της οθόνης μας.

7. Μορφοποίηση Φόρμας

Μια φόρμα εισαγωγής δεδομένων μετά που θα τη δημιουργήσουμε, έχουμε τη δυνατότητα να τη μορφοποιήσουμε και να την τροποποιήσουμε έτσι ώστε να την κάνουμε πιο φιλική και ύχρηστη στον χρήστη της βάσης δεδομένων μας.

Για κάθε πεδίο της φόρμας έχουμε μια ετικέτα (Label) στην οποία αναγράφεται η περιγραφή του πεδίου και ένα πλαίσιο κειμένου (Text Box) στο οποίο ο χρήστης της φόρμας μπορεί να καταχωρεί ή να εμφανίζει την τιμή του αντίστοιχου πεδίου για κάθε εγγραφή.



Μπορούμε να μορφοποιήσουμε μια φόρμα, κάνοντας **διπλό κλικ** πάνω της και ανοίγοντάς την πρώτα σε προβολή φόρμας (Form View).

7.1 Αλλαγή Διαστάσεων της ενεργού περιοχής της φόρμας

7.1.1 Αλλαγή Πλάτους της ενεργού περιοχής της φόρμας

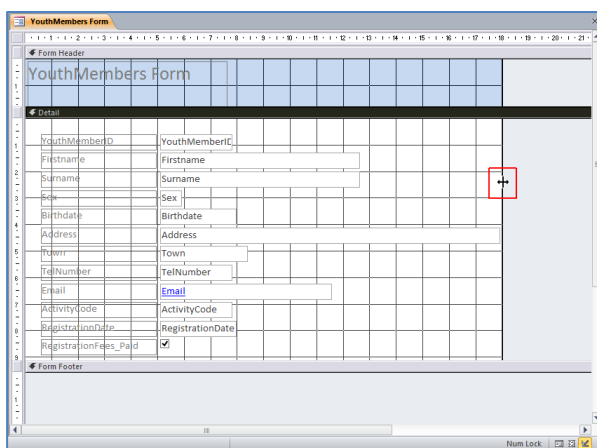
- (1) Από την καρτέλα Home και την ομάδα Views επιλέγουμε το εικονίδιο Design View για να ανοίξουμε τη φόρμα μας σε προβολή σχεδίασης.



ή

Επιλέγουμε τη φόρμα και κάνουμε δεξί κλικ πάνω της και επιλέγουμε την εντολή **Design View** για να την ανοίξει σε προβολή σχεδίασης.

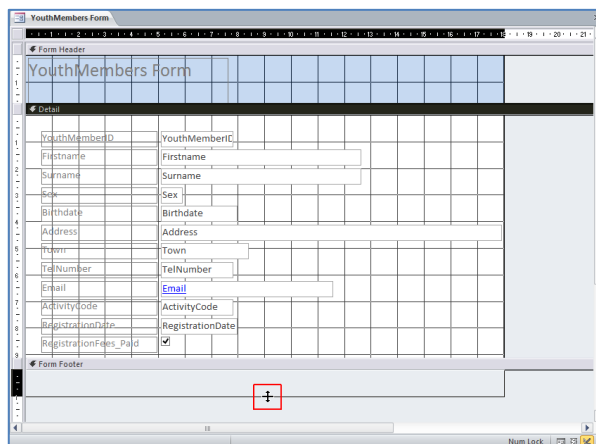
- (2) Στη συνέχεια, κάνουμε κλικ στη δεξιά πλευρά του περιγράμματος της ενεργού περιοχής της φόρμας.
- (3) Όταν ο δείκτης του ποντικιού πάρει τη μορφή γραμμής με τα δύο αντίθετα βέλη, , **σύρουμε ανάλογα** προς τα αριστερά ή προς τα δεξιά για να αλλάξουμε το πλάτος της ενεργού περιοχής της φόρμας.



7.1.2 Αλλαγή Ύψους της ενεργού περιοχής της φόρμας

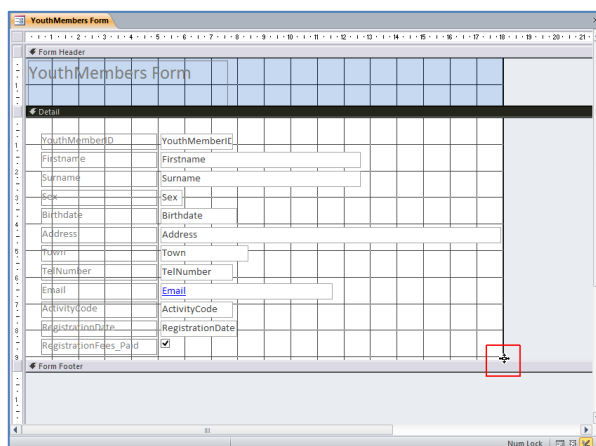
- (1) Κάνουμε κλικ στην κάτω πλευρά του περιγράμματος της ενεργού περιοχής της φόρμας.

- (2) Όταν ο δείκτης του ποντικιού πάρει τη μορφή γραμμής με τα δύο αντίθετα βέλη, , **σύρουμε ανάλογα** προς τα πάνω ή προς τα κάτω για να αλλάξουμε το ύψος της ενεργού περιοχής της φόρμας.



7.1.3 Αλλαγή Πλάτους και Ύψους της ενεργού περιοχής της φόρμας ταυτόχρονα

- (1) Κάνουμε κλικ στην **κάτω δεξιά πλευρά του περιγράμματος** της ενεργού περιοχής της φόρμας.
- (2) Όταν ο δείκτης του ποντικιού πάρει τη μορφή γραμμής με τα δύο αντίθετα βέλη, , **σύρουμε ανάλογα διαγώνια** προς τα πάνω ή προς τα κάτω για να αλλάξουμε τις διαστάσεις της ενεργού περιοχής της φόρμας.

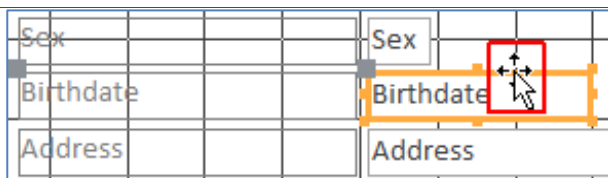


7.2 Μετακίνηση Στοιχείων Ελέγχου

- (1) Κάνουμε κλικ σε ένα στοιχείο ελέγχου (είτε στο πλαίσιο κειμένου είτε στην ετικέτα) πάνω όμως στο περίγραμμα του (όχι σε κάποια λαβή του).

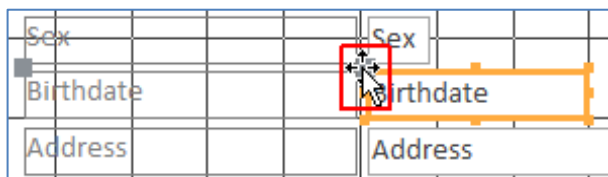
Σημείωση: Γύρω από το περίγραμμά του θα εμφανιστούν μικρά πορτοκαλί τετραγωνάκια τα οποία ονομάζονται λαβές (**handles**).

Όταν ο δείκτης του ποντικιού πάρει τη μορφή με τα τέσσερα αντίθετα βέλη, , **σύρουμε ανάλογα** προς την κατεύθυνση που θέλουμε να το μετακινήσουμε.



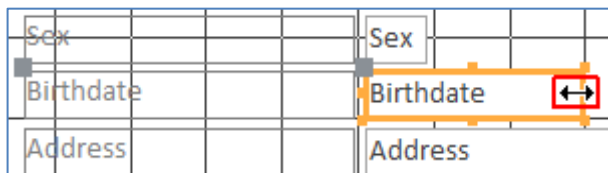
7.3 Μετακίνηση του Πλαισίου Κειμένου ή της Ετικέτας ξεχωριστά

- (1) Κάνουμε κλικ σε ένα στοιχείο ελέγχου (είτε στο πλαίσιο κειμένου είτε στην ετικέτα) πάνω όμως στο γκρίζο τετραγωνάκι του.
- (2) Όταν ο δείκτης του ποντικιού πάρει τη μορφή με τα τέσσερα αντίθετα βέλη, , **σύρουμε ανάλογα** προς την κατεύθυνση που θέλουμε να το μετακινήσουμε.



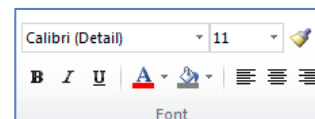
7.4 Αλλαγή Διαστάσεων του Πλαισίου Κειμένου ή της Ετικέτας ξεχωριστά

- (1) Κάνουμε κλικ σε ένα στοιχείο ελέγχου (είτε στο πλαίσιο κειμένου είτε στην ετικέτα) γύρω όμως από το περίγραμμά τους πάνω σε κάποια από τις λαβές του.
- (2) Όταν ο δείκτης του ποντικιού πάρει τη μορφή διπλού βέλους , **σύρουμε ανάλογα** προς τα αριστερά ή δεξιά ή πάνω ή κάτω ή διαγώνια για να του αλλάξουμε τις διαστάσεις του.

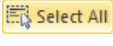


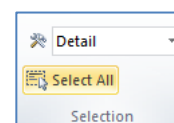
7.5 Μορφοποίηση Στοιχείων Ελέγχου (Formatting)

- (1) Κάνουμε κλικ σε ένα στοιχείο ελέγχου (είτε στο πλαίσιο κειμένου είτε στην ετικέτα) πάνω όμως στο γκρίζο τετραγωνάκι του.
- (2) Ακολουθώντας, από την καρτέλα Form Design Tools→**Format** και την ομάδα Font κάνουμε κλικ σε μια από τις διαθέσιμες επιλογές μορφοποίησης που υπάρχουν και επιλέγουμε τη μορφοποίηση που θέλουμε να χρησιμοποιήσουμε.



7.6 Επιλογή Όλων των Αντικειμένων (Select All)

- (1) Από την καρτέλα Form Design Tools→**Format** και την ομάδα Selection επιλέγουμε το εικονίδιο  για να επιλέξουμε όλα τα αντικείμενα που υπάρχουν στη φόρμα.



ή

Μπορούμε επίσης να χρησιμοποιήσουμε τον συνδυασμό πλήκτρων



7.7 Διαγραφή Πεδίου από τη φόρμα (Delete Field)

- (1) Κάνουμε κλικ σε ένα από τα πλαίσια κειμένου πάνω όμως στο γκρίζο τετραγωνάκι του.

(*Σημείωση: Το πρόγραμμα θα αφαιρέσει και το πλαίσιο κειμένου και την ετικέτα του μαζί από τη φόρμα σας.*)

- (2) Ακολουθώντας, από την καρτέλα **Home** και την ομάδα **Records** επιλέγουμε το εικονίδιο **Delete** για να το διαγράψουμε από τη φόρμα.



ή

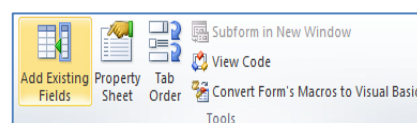
Κάνουμε δεξί κλικ και επιλέγουμε την εντολή **Delete** για να το διαγράψουμε από τη φόρμα.

ή

Μπορούμε επίσης να χρησιμοποιήσουμε το πλήκτρο **Delete**.

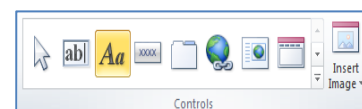
7.8 Προσθήκη Πεδίου στη φόρμα (Add Existing Fields)

- (1) Από την καρτέλα **Form Design Tools**→**Design** και την ομάδα **Tools** επιλέγουμε το εικονίδιο .
- (2) Ακολουθώντας, από το παράθυρο που εμφανίζεται επιλέγουμε το πεδίο που θέλουμε να χρησιμοποιήσουμε και το **σύρουμε** πάνω στη φόρμα.

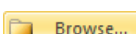


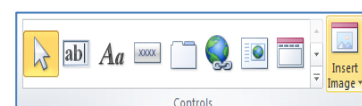
7.9 Προσθήκη Ετικέτας στη φόρμα (Label)

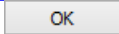
- (1) Από την καρτέλα **Form Design Tools**→**Design** και την ομάδα **Controls** επιλέγουμε το εικονίδιο .
- (2) Ακολουθώντας, κάνουμε κλικ σε ένα οποιοδήποτε σημείο πάνω στη φόρμα όπου ο δείκτης του ποντικιού θα μετατραπεί σε  και γράφουμε μέσα στον χώρο που θα δημιουργήσει αυτό που θέλουμε.



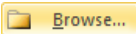
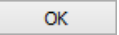
7.10 Προσθήκη Εικόνας στη φόρμα (Insert Image)

- (1) Από την καρτέλα **Form Design Tools**→**Design** και την ομάδα **Controls** επιλέγουμε το εικονίδιο .
- (2) Ακολουθώντας, επιλέγουμε το εικονίδιο  και στο πλαίσιο διαλόγου του **Insert Picture** που εμφανίζεται εντοπίζουμε τον φάκελο που περιέχει το αρχείο της εικόνας, και επιλέγουμε στη συνέχεια την εικόνα που θέλουμε να χρησιμοποιήσουμε.

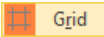


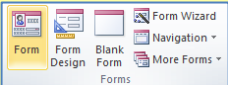
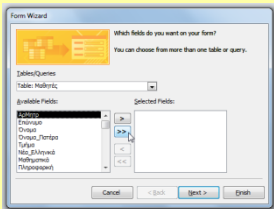
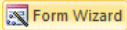
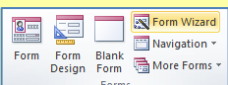
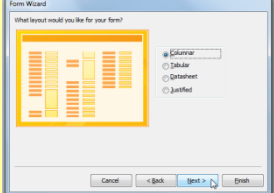
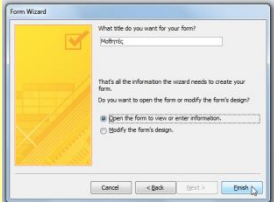
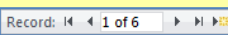
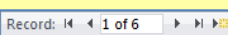
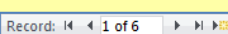
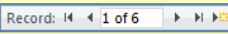
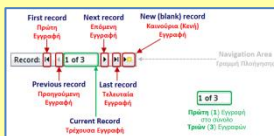
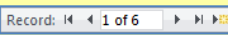
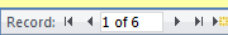
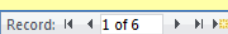
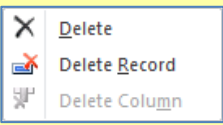
- (3) Μετά, κάνουμε κλικ στο κουμπί  και στη συνέχεια, ξανακάνουμε κλικ σε ένα οποιοδήποτε σημείο πάνω στη φόρμα όπου ο δείκτης του ποντικιού θα μετατραπεί σε  και η **εικόνα** εισάγεται στη φόρμα.

7.11 Προσθήκη Εικόνας Φόντου στη φόρμα (Background Image)

- (1) Από την καρτέλα **Form Design Tools**→**Format** και την ομάδα **Background** επιλέγουμε το εικονίδιο .
- (2) Ακολούθως, επιλέγουμε το εικονίδιο  και στο πλαίσιο διαλόγου του **Insert Picture** που εμφανίζεται εντοπίζουμε τον φάκελο που περιέχει το αρχείο της εικόνας, και επιλέγουμε στη συνέχεια την εικόνα φόντου που θέλουμε να χρησιμοποιήσουμε.
- (3) Μετά, κάνουμε κλικ στο κουμπί  και η **εικόνα φόντου** εισάγεται στη φόρμα.

7.12 Απόκρυψη/Εμφάνιση του Πλέγματος Σχεδίασης (Grid)

- (1) Κάνουμε δεξί κλικ οπουδήποτε πάνω στη φόρμα.
- (2) Ακολούθως, κάνουμε κλικ στο κουμπί  και **ενεργοποιούμε** ή **απενεργοποιούμε** ανάλογα το πλέγμα σχεδίασης στη φόρμα μας.

Υπόμνημα					
Περιγραφή (Description)	Καρτέλα (Tab)	Εικονίδιο (Icon)	Ομάδα (Group)	Παράθυρο Διαλόγου (Dialog Box)	Πλήκτρα (Keys)
Δημιουργία Φόρμας με το Εργαλείο Αυτόματης Φόρμας (Form)	Create				
Δημιουργία Φόρμας με τη χρήση του Οδηγού Φορμών (Form Wizard)	Create			 	
Γραμμή Πλοήγησης (Navigation Area)					
Πρώτη Εγγραφή (First record)					
Προηγούμενη Εγγραφή (Previous record)					
Τρέχουσα Εγγραφή (Current Record)	1 of 6				
Επόμενη Εγγραφή (Next record)					
Τελευταία Εγγραφή (Last record)					
Νέα Κενή Εγγραφή (New (blank) record)					
Διαγραφή Εγγραφής (Delete Record)	Home			 	

Μετακίνηση του
δρομέα στην ίδια την
εγγραφή
(**Επόμενο** και
Προηγούμενο)



Μετακίνηση του
δρομέα στην ίδια την
εγγραφή
(**Επόμενο**)

TAB

Μετακίνηση του
δρομέα στην ίδια την
εγγραφή
(**Προηγούμενο**)

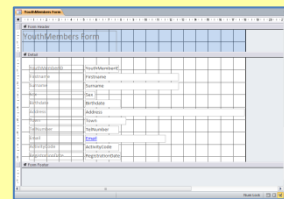
SHIFT

+

TAB

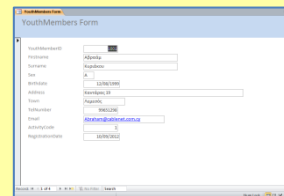
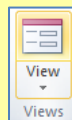
Προβολή
Σχεδίασης
(**Design View**)

Home



Προβολή
Φόρμας
(**Form View**)

Home



Γ6.8 Ερωτήματα

Τι θα μάθουμε σήμερα:

- Να περιγράφουμε την ανάγκη χρήσης των ερωτημάτων σε μια βάση δεδομένων (*πλεονεκτήματα*)
- Να δημιουργούμε, να τροποποιούμε, να αποθηκεύουμε και να διαγράφουμε απλά ερωτήματα
- Να εφαρμόζουμε, να προσθέτουμε και να διαγράφουμε κριτήρια σε ένα ερώτημα (*χρήση σταθερών τιμών και συγκριτικών τελεστών: =, <, >, <>, <=, >=, between ...and ...*)
- Να εφαρμόζουμε, να προσθέτουμε και να διαγράφουμε κριτήρια σε ένα ερώτημα (*χρήση λογικών τελεστών και χαρακτήρων μπαλαντέρ: AND, OR, NOT και *, ?*)
- Να ταξινομούμε αποτελέσματα ερωτημάτων βάσει των πεδίων σε αύξουσα/φθίνουσα, αλφαβητική/αριθμητική σειρά (*Ascending/Descending*).

1. Ερωτήματα (Queries)

Μια βάση δεδομένων μπορεί να περιέχει πολλούς πίνακες συνδεδεμένους μεταξύ τους, με χιλιάδες εγγραφές στον καθένα. Είναι λοιπόν πολύ δύσκολο να βρει ο χρήστης συγκεκριμένα δεδομένα που ψάχνει.

Η χρήση των φίλτρων ή της εντολής αναζήτησης που έχουν ήδη αναφερθεί, βοηθάνε σε κάποιο βαθμό αλλά έχουν και πολλά μειονεκτήματα, όπως το ότι δεν διατηρούνται και το ότι δεν μπορούν να πάρουν δεδομένα από δύο ή περισσότερους πίνακες. Λύση στα πιο πάνω προβλήματα δίνουν τα **Ερωτήματα (Queries)**.

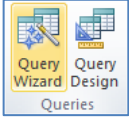
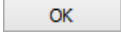
Πιο κάτω παραθέτουμε μερικές από τις δυνατότητες (**πλεονεκτήματα**) των **ερωτημάτων**:

- Μπορούμε να επιλέξουμε μόνο συγκεκριμένα πεδία ενός πίνακα και να τα παρουσιάσουμε με τη σειρά που θέλουμε.
- Μπορούμε να θέσουμε κριτήρια και να εμφανιστούν μόνο οι εγγραφές που ικανοποιούν αυτά τα κριτήρια.
- Μπορούμε να τα αποθηκεύσουμε στη βάση δεδομένων και να τα εκτελέσουμε όποτε τα χρειαστούμε.
- Μπορούμε να αντλούμε δεδομένα από έναν ή περισσότερους πίνακες.
- Μπορούν, επίσης, να κάνουν και υπολογισμούς.

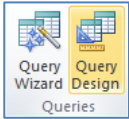
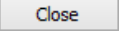
Παίρνουμε ως **παράδειγμα** τη βάση δεδομένων ενός σχολείου, η οποία υποθέτουμε ότι έχει τρεις πίνακες. Έναν πίνακα που περιέχει τα **προσωπικά στοιχεία** των μαθητών, άλλο που περιέχει τους **βαθμούς** και έναν τρίτο που περιέχει τις **απουσίες**:

- (1) Με τη χρήση ενός ερωτήματος από τον πίνακα με τα προσωπικά στοιχεία μπορούμε να πάρουμε κάποια από τα προσωπικά δεδομένα των μαθητών, όπως το ονοματεπώνυμο και το τηλέφωνο των γονέων τους, ενώ από τον πίνακα των απουσιών μπορούμε να εντοπίσουμε ποιοι μαθητές έχουν πρόβλημα απουσιών.
- (2) Με ένα άλλο ερώτημα από τον πίνακα με τους βαθμούς μπορούμε να βρούμε τους άριστους μαθητές κ.λπ.

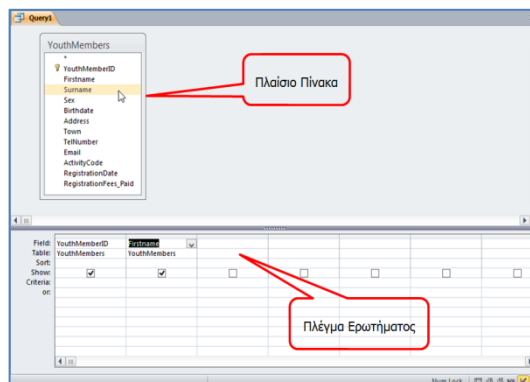
2. Δημιουργία Ερωτήματος με τη χρήση του Οδηγού Ερωτημάτων (Query Wizard)

- (1) Επιλέγουμε τον πίνακα.
- (2) Από την καρτέλα **Create** και την ομάδα **Queries** επιλέγουμε το εικονίδιο **Query Wizard**. 
- (3) Στη συνέχεια, επιλέγουμε **Simple Query Wizard** και κάνουμε κλικ στο κουμπί  για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- (4) Ακολούθως, κάνουμε κλικ στο κουμπί  για να μεταφέρουμε ΟΛΑ τα πεδία στο πλαίσιο **Selected Fields:** ή κάνουμε κλικ στο κουμπί  για να μεταφέρουμε ΕΝΑ προς ΕΝΑ τα πεδία στο πλαίσιο **Selected Fields:** και μετά κάνουμε κλικ στο κουμπί  για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- (5) Στη συνέχεια, επιλέγουμε την **επιθυμητή διάταξη** (Αναλυτικά ή Περιληπτικά) για το ερώτημά μας και κάνουμε ξανά κλικ στο κουμπί  για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- (6) Τέλος, καθορίζουμε ένα **περιγραφικό όνομα** και κάνουμε κλικ στο κουμπί  για να **αποθηκεύσουμε** το ερώτημά μας.

3. Δημιουργία Ερωτήματος με τη χρήση της Προβολή Σχεδίασης (Query Design)

- (1) Επιλέγουμε τον πίνακα.
- (2) Από την καρτέλα **Create** και την ομάδα **Queries** επιλέγουμε το εικονίδιο **Query Design**. 
- (3) Στη συνέχεια, από το πλαίσιο διαλόγου εμφάνισης πινάκων (**Show Table**) επιλέγουμε τον/τους **πίνακα/ες** που θέλουμε να χρησιμοποιήσουμε και κάνουμε κλικ στο κουμπί  για να τον/τους **προσθέσουμε** στην προβολή σχεδίασης (**Design View**) του ερωτήματός μας.
- (4) Ακολούθως, κάνουμε κλικ στο κουμπί  για να **κλείσουμε** το πλαίσιο διαλόγου εμφάνισης πινάκων.
- (5) Στο **πλαίσιο πίνακα**, κάνουμε **διπλό κλικ** στο πρώτο πεδίο για να το **προσθέσουμε** στην πρώτη στήλη του πλέγματος ερωτημάτων μας.

(*Σημείωση: Στο πάνω μισό τμήμα του παραθύρου μπορούμε να δούμε το **πλαίσιο πίνακα** με τα πεδία του. (Το * στην αρχή του καταλόγου αντιπροσωπεύει όλα τα πεδία.) Στο κάτω μισό τμήμα του παραθύρου βρίσκεται το πλέγμα ενός πίνακα – ονομάζεται **πλέγμα ερωτήματος**, στο οποίο μπορούμε να δημιουργήσουμε και να τροποποιήσουμε το ερώτημά μας με οπτικό τρόπο.*)

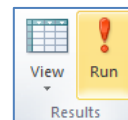


- (6) Στη συνέχεια, κάνουμε **διπλό κλικ** στο δεύτερο πεδίο για να το **προσθέσουμε** στη δεύτερη στήλη του πλέγματος ερωτημάτων μας με τον ίδιο τρόπο.

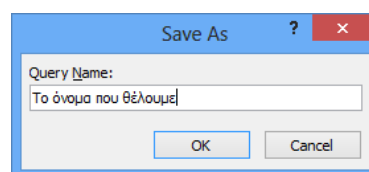
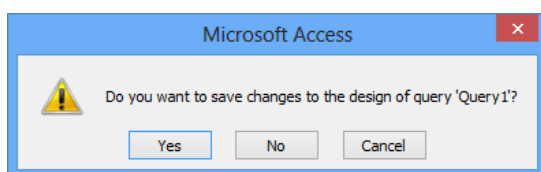
(*Σημείωση: Εάν προσθέσουμε λάθος πεδίο, κάνουμε κλικ στο πλήκτρο  για να το διαγράψουμε.*)

- (7) Συνεχίζουμε με τον ίδιο τρόπο όπως πιο πάνω μέχρι να **προσθέσουμε** όλα τα πεδία που θέλουμε να χρησιμοποιήσουμε στο ερώτημά μας.

- (8) Ακολουθώντας, από την καρτέλα **Query Tools**→**Design** και την ομάδα **Results** επιλέγουμε το εικονίδιο **Run** για να εκτελέσει το ερώτημα η Microsoft Access και να εμφανίσει το αποτέλεσμα.



- (9) Τέλος, κλείνουμε το ερώτημα, του καθορίζουμε ένα **περιγραφικό όνομα** και αποθηκεύουμε το ερώτημα μας.



4. Τροποποίηση Ερωτήματος

- (1) Από την περιοχή όλων των αντικειμένων της εφαρμογής (**All Access Objects**) κάτω από τον τίτλο **Queries**, επιλέγουμε το ερώτημα που θέλουμε να τροποποιήσουμε και κάνουμε **διπλό κλικ** πάνω του για να το ανοίξουμε πρώτα σε προβολή φύλλου δεδομένων (**Datasheet View**).

- (2) Ακολουθώντας, από την καρτέλα **Home** και την ομάδα **Views** επιλέγουμε το εικονίδιο **Design View** για να ανοίξουμε το ερώτημά μας σε προβολή σχεδίασης.

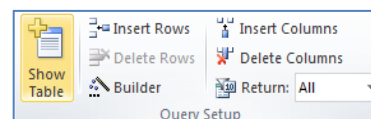


ή

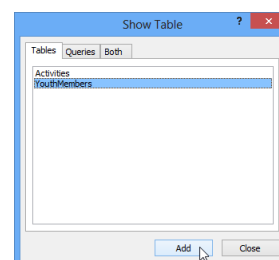
Επιλέγουμε το ερώτημα από τη συγκεκριμένη περιοχή, κάνουμε δεξί κλικ πάνω του και επιλέγουμε την εντολή **Design View** για να το ανοίξει σε προβολή σχεδίασης.

- (3) Στην περίπτωση που θέλουμε να **προσθέσουμε** ακόμη έναν πίνακα ή ακόμη ένα ερώτημα στο νέο μας ερώτημα, κάνουμε τις πιο κάτω ενέργειες:

- Από την καρτέλα **Query Tools**→**Design** και την ομάδα **Query Setup** επιλέγουμε το εικονίδιο **Show Table** για να ανοίξουμε το πλαίσιο διαλόγου εμφάνισης πινάκων.



- Στη συνέχεια, από το πλαίσιο διαλόγου εμφάνισης πινάκων επιλέγουμε τον/τους **πίνακα/ες** ή το/τα **ερώτημα/τα** που θέλουμε να χρησιμοποιήσουμε και κάνουμε κλικ στο κουμπί **Add** για να το/τα **προσθέσουμε** στην προβολή σχεδίασης του ερωτήματός μας.

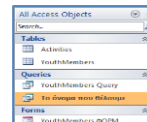


- Ακολουθώντας, κάνουμε κλικ στο κουμπί  για να **κλείσουμε** το πλαίσιο διαλόγου εμφάνισης πινάκων.

(4) Στη συνέχεια, κλείνουμε το ερώτημά, αποθηκεύοντας όμως **ξανά** τις νέες αλλαγές που κάναμε στο ερώτημά μας.

5. Διαγραφή Ερωτήματος (Delete Query)

(1) Από την περιοχή όλων των αντικειμένων της εφαρμογής (**All Access Objects**) κάτω από τον τίτλο **Queries**, επιλέγουμε το ερώτημα που θέλουμε να διαγράψουμε.

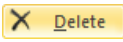


(2) Ακολουθώντας, κάνουμε δεξί κλικ και επιλέγουμε την εντολή **Delete** για να το διαγράψουμε από τη βάση δεδομένων μας.

ή

Από την καρτέλα **Home** και την ομάδα **Records** επιλέγουμε το εικονίδιο **Delete**.



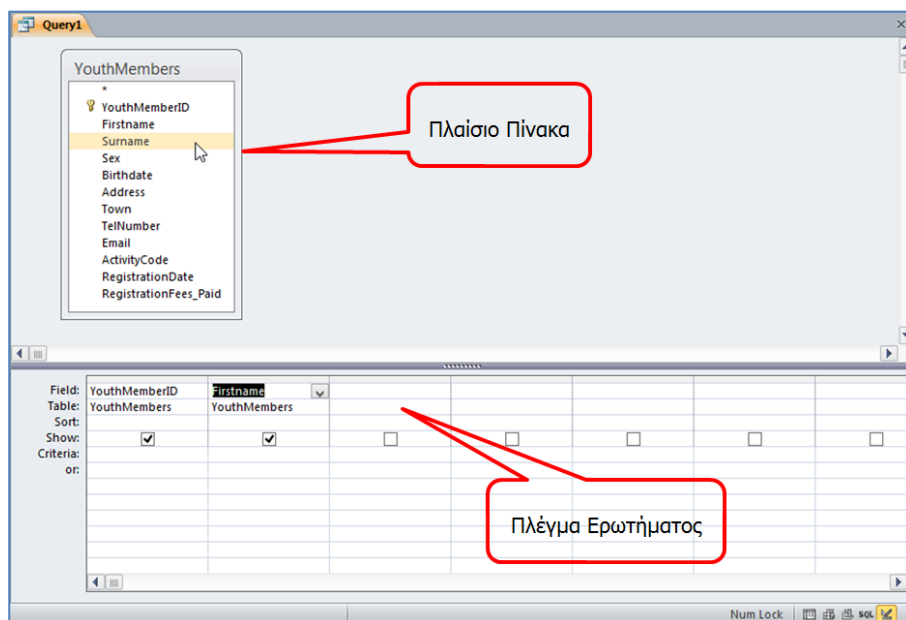
Ακολουθώντας, κάνουμε κλικ στο κουμπί  για να το **διαγράψουμε** από τη βάση δεδομένων μας.

ή

Μπορούμε, επίσης, να χρησιμοποιήσουμε το πλήκτρο **Delete**.

6. Το Παράθυρο της Προβολής Σχεδίασης ενός Ερωτήματος

Το παράθυρο της προβολής σχεδίασης ενός ερωτήματος χωρίζεται σε **δύο τμήματα**:



(1) Στο πάνω μισό τμήμα του παραθύρου μπορούμε να δούμε το **πλαίσιο πίνακα** με τα πεδία του στο οποίο βασίζεται το ερώτημα. Το ερώτημα του δικού μας παραδείγματος βασίζεται σε έναν μόνο πίνακα, γι' αυτό και βλέπουμε μόνο μια λίστα πεδίων. (Το * στην αρχή του καταλόγου αντιπροσωπεύει όλα τα πεδία.)

(2) Στο κάτω μισό τμήμα του παραθύρου βρίσκεται το πλέγμα ενός πίνακα – ονομάζεται **πλέγμα ερωτήματος**, στο οποίο μπορούμε να δημιουργήσουμε και να τροποποιήσουμε το ερώτημά μας με οπτικό τρόπο όπως πιο κάτω:

- **Field:** Παρουσιάζει τα πεδία που θα χρησιμοποιήσουμε.
- **Table:** Παρουσιάζει τον/τους πίνακα/ες που θα χρησιμοποιήσουμε.
- **Sort:** Καθορίζουμε την ταξινόμηση που θα χρησιμοποιήσουμε όπως:
 - **Ascending** (ΑΥΞΟΥΣΑ ταξινόμηση),
 - **Descending** (ΦΘΙΝΟΥΣΑ ταξινόμηση),
 - **(not sorted)** (ΔΙΧΩΣ ταξινόμηση).
- **Show:** Καθορίζουμε εάν το πεδίο θα είναι ορατό ή όχι όταν εκτελέσουμε το ερώτημα.
- **Criteria:** Καθορίζουμε τα κριτήρια, δηλαδή τους περιορισμούς που ορίζουμε στα ερωτήματα, ώστε να μας βρίσκουν μόνο τις εγγραφές που μας ενδιαφέρουν.

7. Κριτήρια Ερωτημάτων (Criteria)

Τα κριτήρια είναι οι **περιορισμοί** που ορίζουμε στα ερωτήματα, ώστε να βρίσκουν από τη Βάση Δεδομένων **μόνο τις εγγραφές** που μας ενδιαφέρουν όπως:

Σταθερές Τιμές	Έκφραση	Περιγραφή
Λέξη ή Γράμμα	Λεμεσός	Ίση με ...
Αριθμός	2012	Ίση με ...
Ημερομηνία	31/1/1967	Ίση με ...

Συγκριτικοί Τελεστές	Έκφραση	Περιγραφή
Between ...and...	Between 1/1/2013 and 31/12/2013	Μεταξύ ... και ...
=	=35	Ίση με ...
>	>36	Μεγαλύτερη από ...
<	<40	Μικρότερη από ...
<>	<>0	Διάφορο του ... (άνισο)
>=	>=50	Μεγαλύτερη ή ίση με ...
<=	<=46	Μικρότερη ή ίση με ...

Λογικοί Τελεστές	Έκφραση	Περιγραφή
AND	<50 AND >40	<u>και</u> ο ένας περιορισμός <u>και</u> ο άλλος
OR	48 OR 52	<u>ή</u> ο ένας περιορισμός <u>ή</u> ο άλλος
NOT	NOT Ελένη	Εκτός από ...

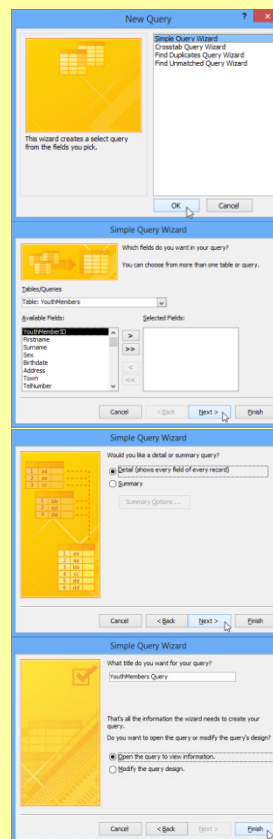
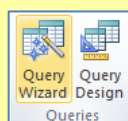
Χαρακτήρες Μπαλαντέρ	Έκφραση	Περιγραφή
*	N*	Αντιστοιχεί σε κανένα ή περισσότερους χαρακτήρες
?	N???ς	Αντιστοιχεί με οποιονδήποτε μεμονωμένο χαρακτήρα
[]	[XK]ριστίνα	Αντιστοιχεί σε κάποιο από τους χαρακτήρες στις αγκύλες
!	[!XK]ριστίνα	Αντιστοιχεί με οποιονδήποτε χαρακτήρα <u>εκτός</u> από τους χαρακτήρες στις αγκύλες
-	[A-E]	Αντιστοιχεί με κάθε χαρακτήρα που βρίσκεται στην περιοχή μέσα στις αγκύλες
#	2#5	Αντιστοιχεί με οποιονδήποτε μεμονωμένο αριθμητικό χαρακτήρα

Υπόμνημα

Περιγραφή
(Description)**Καρτέλα**
(Tab)**Εικονίδιο**
(Icon)**Ομάδα**
(Group)**Παράθυρο Διαλόγου**
(Dialog Box)**Πλήκτρα**
(Keys)

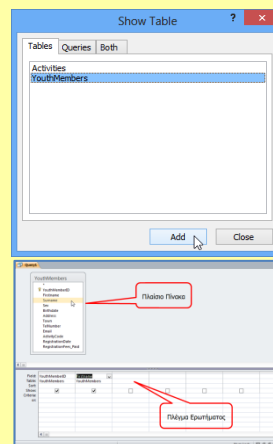
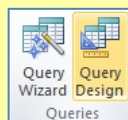
Δημιουργία
Ερωτήματος με τη
χρήση του
Οδηγού Ερωτημάτων
(**Query Wizard**)

Create

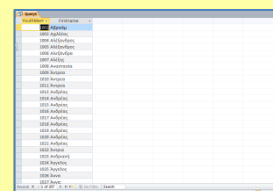
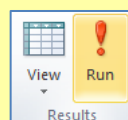


Δημιουργία
Ερωτήματος με τη
χρήση της
Προβολής Σχεδίασης
(**Query Design**)

Create



Εκτέλεση
Ερωτήματος
(**Run**)

Query Tools
Design

Εμφάνιση Πινάκων σε Ερώτημα (Show Table)

Query Tools
Design

Show Table

Insert Rows, Insert Columns, Delete Rows, Delete Columns, Builder, Return: All, Query Setup

Show Table

Tables, Queries, Both, Activities, YouthMembers, Add, Close

Τροποποίηση Ερωτήματος

Home

View

Views

Field: Firstname, Surname, Sex, Table: YouthMembers, Sort: YouthMembers, Show: Ascending, Criteria: (not sorted)

Ταξινόμηση Αποτελεσμάτων των Ερωτημάτων (Sort)

Field: Firstname, Surname, Sex, Table: YouthMembers, Sort: YouthMembers, Show: Ascending, Criteria: (not sorted)

Διαγραφή Ερωτήματος (Delete Query)

Home

Delete

Refresh All, New, Save, Spelling, Records, Delete, More

Delete, Delete Record, Delete Column, Delete

Τι θα μάθουμε σήμερα:

- ✦ Να δημιουργούμε ερωτήματα από δύο συσχετιζόμενους πίνακες.

1. Ερωτήματα από δύο Συσχετιζόμενους Πίνακες (Queries in Related Tables)

Τα ερωτήματα από **δύο συσχετιζόμενους πίνακες** αποτελούν έναν από τους πιο αποτελεσματικούς τρόπους εξαγωγής συγκεκριμένων πληροφοριών από μια συσχετιζόμενη βάση δεδομένων.

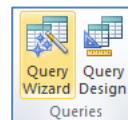
Όταν υπάρχουν συσχετισμοί μεταξύ δύο πινάκων, οι πληροφορίες που μπορούν να παραχθούν από ένα ερώτημα περιέχουν ήδη τους συσχετισμούς ανάμεσα στα δεδομένα των δύο πινάκων, έχοντας έτσι το πλεονέκτημα της **ταυτόχρονης παρουσίασης συσχετιζόμενων πληροφοριών** από δύο ή περισσότερους πίνακες.

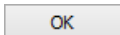
Αυτό είναι και το **τεράστιο πλεονέκτημα** της δημιουργίας ερωτημάτων από δύο συσχετιζόμενους πίνακες: η ταυτόχρονη συγκέντρωση, αντιπαράβολή και παρουσίαση συσχετιζόμενων πληροφοριών από μια βάση δεδομένων για περαιτέρω επεξεργασία ή αποθήκευση.

2. Δημιουργία ερωτήματος με τη χρήση του Οδηγού Ερωτημάτων (Query Wizard)

(1) Επιλέγουμε τον πίνακα.

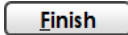
(2) Από την καρτέλα **Create** και την ομάδα **Queries** επιλέγουμε το εικονίδιο **Query Wizard**.



(3) Στη συνέχεια, επιλέγουμε **Simple Query Wizard** και κάνουμε κλικ στο κουμπί  για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.

(4) Ακολουθώντας, κάνουμε κλικ στο κουμπί  για να μεταφέρουμε ΟΛΑ τα πεδία στο πλαίσιο **Selected Fields:** ή κάνουμε κλικ στο κουμπί  για να μεταφέρουμε ΕΝΑ προς ΕΝΑ τα πεδία στο πλαίσιο **Selected Fields:** και μετά κάνουμε κλικ στο κουμπί  για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.

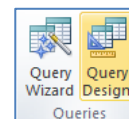
(5) Στη συνέχεια, επιλέγουμε την **επιθυμητή διάταξη** (Αναλυτικά ή Περιληπτικά) για το ερώτημά μας και κάνουμε ξανά κλικ στο κουμπί  για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.

(6) Τέλος, καθορίζουμε ένα **περιγραφικό όνομα** και κάνουμε κλικ στο κουμπί  για να **αποθηκεύσουμε** το ερώτημά μας.

3. Δημιουργία ερωτήματος με τη χρήση της Προβολής Σχεδίασης (Query Design)

(1) Επιλέγουμε τον πίνακα.

(2) Από την καρτέλα **Create** και την ομάδα **Queries** επιλέγουμε το εικονίδιο **Query Design**.



(3) Στη συνέχεια, από το πλαίσιο διαλόγου εμφάνισης πινάκων (**Show Table**) επιλέγουμε τον/τους **πίνακα/ες** που θέλουμε να χρησιμοποιήσουμε και κάνουμε κλικ στο κουμπί **Add** για να τον/τους **προσθέσουμε** στην **προβολή σχεδίασης (Design View)** του ερωτήματός μας.

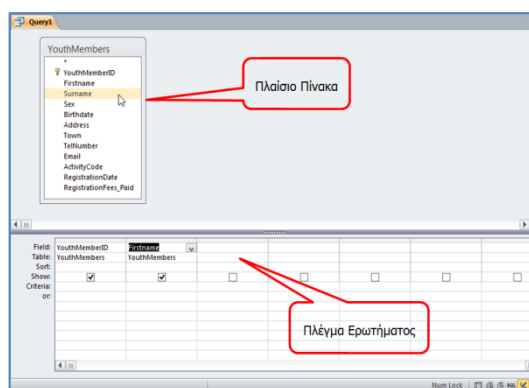
(4) Ακολούθως, κάνουμε κλικ στο κουμπί **Close** για να **κλείσουμε** το πλαίσιο διαλόγου εμφάνισης πινάκων.

(5) Στο **πλαίσιο πίνακα**, κάνουμε **διπλό κλικ** στο **πρώτο πεδίο** για να το **προσθέσουμε** στην **πρώτη στήλη** του πλέγματος ερωτημάτων μας.

(Σημείωση: Στο πάνω μισό τμήμα του παραθύρου μπορούμε να δούμε το **πλαίσιο πίνακα** με τα πεδία του.

(Το * στην αρχή του καταλόγου αντιπροσωπεύει όλα τα πεδία.)

Στο κάτω μισό τμήμα του παραθύρου βρίσκεται το πλέγμα ενός πίνακα – ονομάζεται **πλέγμα ερωτήματος**, στο οποίο μπορούμε να δημιουργήσουμε και να τροποποιήσουμε το ερώτημά μας με οπτικό τρόπο.)

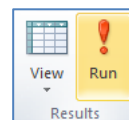


(6) Στη συνέχεια, κάνουμε **διπλό κλικ** στο **δεύτερο πεδίο** για να το **προσθέσουμε** στη **δεύτερη στήλη** του πλέγματος ερωτημάτων μας με τον ίδιο τρόπο.

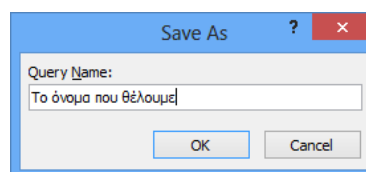
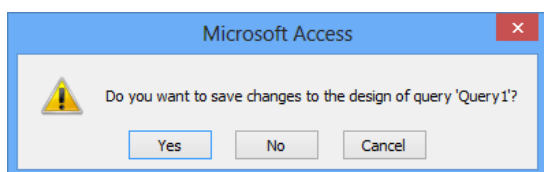
(Σημείωση: Εάν προσθέσουμε λάθος πεδίο, κάνουμε κλικ στο πλήκτρο **Delete** για να το διαγράψουμε.)

(7) Συνεχίζουμε με τον ίδιο τρόπο όπως πιο πάνω μέχρι να **προσθέσουμε** όλα τα πεδία που θέλουμε να χρησιμοποιήσουμε στο ερώτημά μας.

(8) Ακολούθως, από την καρτέλα **Query Tools**→**Design** και την ομάδα **Results** επιλέγουμε το εικονίδιο **Run** για να εκτελέσει το ερώτημα η **Microsoft Access** και να εμφανίσει το αποτέλεσμα.



(9) Τέλος, κλείνουμε το ερώτημα, του καθορίζουμε ένα **περιγραφικό όνομα** και αποθηκεύουμε το ερώτημά μας.

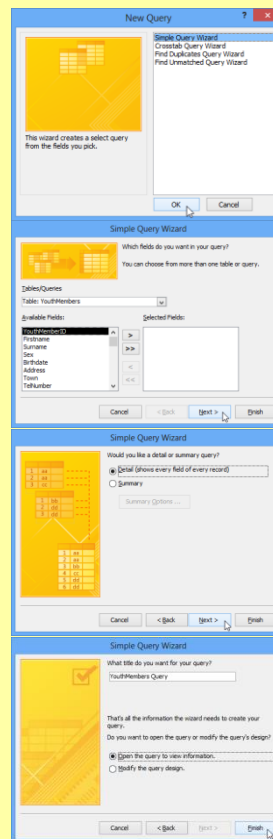
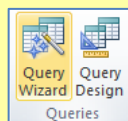


Υπόμνημα

Περιγραφή
(Description)**Καρτέλα**
(Tab)**Εικονίδιο**
(Icon)**Ομάδα**
(Group)**Παράθυρο Διαλόγου**
(Dialog Box)**Πλήκτρα**
(Keys)

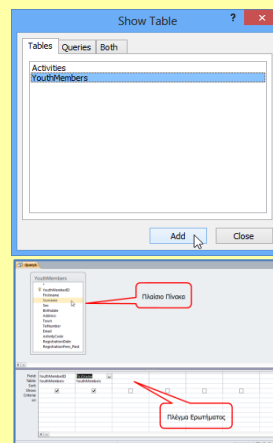
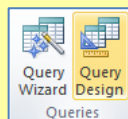
Δημιουργία
Ερωτήματος με τη
χρήση του
Οδηγού Ερωτημάτων
(**Query Wizard**)

Create

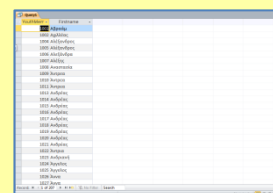
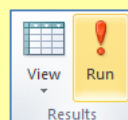


Δημιουργία
Ερωτήματος με τη
χρήση της
Προβολής Σχεδίασης
(**Query Design**)

Create



Εκτέλεση
Ερωτήματος
(**Run**)

Query Tools
Design

Εμφάνιση Πινάκων
σε Ερώτημα
(**Show Table**)

Query Tools
Design

Show Table

Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return: All
Query Setup

Show Table

Tables | Queries | Both
Activities
You@Members
Add Close

Γ6.10 Εκθέσεις

Τι θα μάθουμε σήμερα:

- Να περιγράφουμε την ανάγκη χρήσης των εκθέσεων σε μια βάση δεδομένων τόσο από πίνακες όσο και από ερωτήματα (*πλεονεκτήματα*)
- Να δημιουργούμε, να αποθηκεύουμε, να εκτυπώνουμε και να διαγράφουμε απλές εκθέσεις.

1. Εκθέσεις (Reports)

Συχνά, χρειάζεται να επιλέξουμε, να οργανώσουμε και να τυπώσουμε συγκεκριμένα στοιχεία της βάσης δεδομένων μας. Για αυτό τον λόγο χρειαζόμαστε τις **εκθέσεις ή αναφορές** οι οποίες είναι εργαλεία που έχουν ως βασικό σκοπό τους την αποτελεσματική παρουσίαση των δεδομένων μας σε **έντυπη μορφή**, έτοιμα προς εκτύπωση.

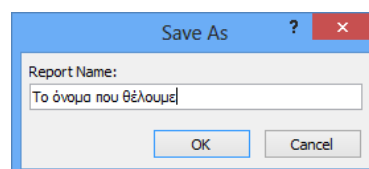
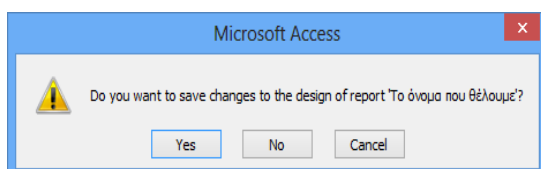
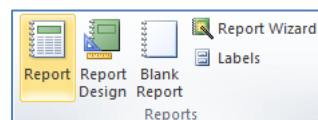
Μια έκθεση αφού τη δημιουργήσουμε και την αποθηκεύσουμε, μπορούμε να την εκτελέσουμε (τρέξουμε). Η ενέργεια αυτή έχει πάντα ως αποτέλεσμα την απεικόνιση των δεδομένων που υπάρχουν στη βάση δεδομένων τη συγκεκριμένη στιγμή.

Κάθε έκθεση μπορεί να μορφοποιηθεί έτσι ώστε να παρουσιάζει τις πληροφορίες με τον καλύτερο δυνατό τρόπο σε ό,τι αφορά την ανάγνωση και τη χρήση τους.

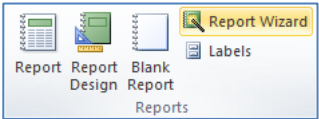
Οι εκθέσεις δημιουργούνται συνήθως για **εκτύπωση** ή για την **προβολή** τους στην οθόνη. Μπορούμε επίσης, το αποτέλεσμα τους να το αποθηκεύσουμε σε ένα αρχείο ή να το κάνουμε εξαγωγή σε κάποιο άλλο πρόγραμμα ή να το αποστείλουμε ως μήνυμα ηλεκτρονικού ταχυδρομείου.

2. Δημιουργία Έκθεσης με το Εργαλείο Αυτόματης Έκθεσης (Report)

- (1) Επιλέγουμε το ερώτημα ή τον πίνακα για το/τον οποίο θέλουμε να δημιουργήσουμε μια έκθεση.
- (2) Από την καρτέλα **Create** και την ομάδα **Reports** επιλέγουμε το εικονίδιο **Report** και η έκθεσή μας δημιουργείται με ένα **αυτόματο** σχετικά τρόπο.
- (3) Τέλος, κλείνουμε την έκθεση, της καθορίζουμε ένα **περιγραφικό όνομα** και αποθηκεύουμε την έκθεσή μας.



3. Δημιουργία Έκθεσης με τη χρήση του Οδηγού Εκθέσεων (Report Wizard)

- (1) Επιλέγουμε το ερώτημα ή τον πίνακα για το/τον οποίο θέλουμε να δημιουργήσουμε μια έκθεση και ...
- (2) Από την καρτέλα **Create** και την ομάδα **Reports** επιλέγουμε το εικονίδιο **Report Wizard**. 
- (3) Ακολουθώντας, κάνουμε κλικ στο κουμπί  για να μεταφέρουμε ΟΛΑ τα πεδία στο πλαίσιο **Selected Fields:** ή κάνουμε κλικ στο κουμπί  για να μεταφέρουμε ΕΝΑ προς ΕΝΑ τα πεδία στο πλαίσιο **Selected Fields:** και μετά κάνουμε κλικ στο κουμπί  για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- (4) Στη συνέχεια, επιλέγουμε ένα από τα πεδία βάσει του οποίου θέλουμε να **ομαδοποιήσουμε** την έκθεσή μας και κάνουμε κλικ στο κουμπί  για να το μεταφέρουμε στο πλαίσιο στα δεξιά και κάνουμε ξανά κλικ στο κουμπί  για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- (5) Στη συνέχεια, μπορούμε επιλέγοντας μέχρι και τέσσερα πεδία, να **ταξινομήσουμε** την έκθεσή μας ανάλογα, σε αύξουσα ή φθίνουσα αλφαβητική/αριθμητική σειρά βάσει του πεδίου και κάνουμε ξανά κλικ στο κουμπί  για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- (6) Στη συνέχεια, επιλέγουμε την **επιθυμητή διάταξη** μαζί με τον **επιθυμητό προσανατολισμό** της έκθεσής μας και κάνουμε ξανά κλικ στο κουμπί  για να εμφανίσουμε το επόμενο πλαίσιο διαλόγου.
- (7) Τέλος, καθορίζουμε ένα **περιγραφικό όνομα** και κάνουμε κλικ στο κουμπί  για να **αποθηκεύσουμε** την έκθεσή μας.

4. Προβολές για Έκθεση (Views)

4.1 Προβολή Σχεδίασης (Design View)

- (1) Από την περιοχή όλων των αντικειμένων της εφαρμογής (**All Access Objects**) κάτω από τον τίτλο **Reports**, όπου εμφανίζονται όλες οι εκθέσεις που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε την έκθεση που θέλουμε και κάνουμε **διπλό κλικ** πάνω της για να την ανοίξουμε πρώτα σε προβολή έκθεσης (**Report View**).
- (2) Ακολουθώντας, από την καρτέλα **Home** και την ομάδα **Views** επιλέγουμε το εικονίδιο **Design View** για να ανοίξουμε την έκθεσή μας σε προβολή σχεδίασης. 

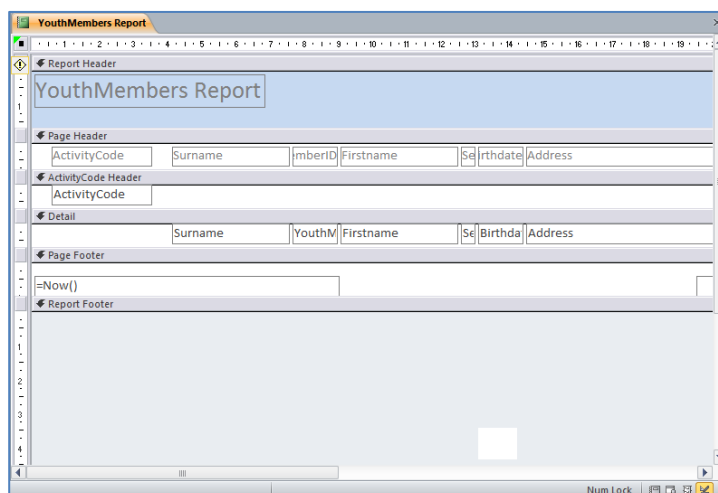
ή

Κάνουμε κλικ στο κουμπί  από τη γραμμή κατάστασης στην κάτω δεξιά γωνιά της οθόνης μας.

ή

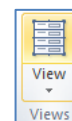
Από την περιοχή όλων των αντικειμένων της εφαρμογής (**All Access Objects**) κάτω από τον τίτλο **Reports**, όπου εμφανίζονται όλες οι εκθέσεις που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε την έκθεση που θέλουμε

και κάνουμε **δεξί κλικ** πάνω της και επιλέγουμε την εντολή **Design View** για να την ανοίξει σε προβολή σχεδίασης.



4.2 Προβολή Διάταξης Έκθεσης (Layout View)

- (1) Από την περιοχή όλων των αντικειμένων της εφαρμογής (All Access Objects) κάτω από τον τίτλο **Reports**, όπου εμφανίζονται όλες οι εκθέσεις που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε την έκθεση που θέλουμε και κάνουμε **διπλό κλικ** πάνω της για να την ανοίξουμε πρώτα σε προβολή έκθεσης (Report View).
- (2) Ακολουθώντας, από την καρτέλα Home και την ομάδα Views επιλέγουμε το εικονίδιο Layout View για να ανοίξουμε την έκθεσή μας σε προβολή διάταξης έκθεσης.



ή

Κάνουμε κλικ στο κουμπί  από τη γραμμή κατάστασης στην κάτω δεξιά γωνιά της οθόνης μας.

ή

Από την περιοχή όλων των αντικειμένων της εφαρμογής (All Access Objects) κάτω από τον τίτλο **Reports**, όπου εμφανίζονται όλες οι εκθέσεις που έχουν δημιουργηθεί στη βάση δεδομένων μας, επιλέγουμε την έκθεση που θέλουμε και κάνουμε **δεξί κλικ** πάνω της και επιλέγουμε την εντολή **Layout View** για να την ανοίξει σε προβολή διάταξης έκθεσης.

ActivityCode	Surname	mberID	Firstname	Se	irthdate	Address
1	Κωνσταντίνου	1126	Λάζαρος	A	#####	Μόρφου 21
	Ερρίκου	1215	Ζήνα	Θ	#####	Ριζοκαρπάσου 35
	Δήμου	1059	Χρίστος	A	#####	Αμμογώστου 17
	Νίκου	1026	Άννα	Θ	#####	Καντάρας 25
	Πολύκαρπου	1102	Ελένη	Θ	#####	Κυρηνείας 37
	Γιώργου	1101	Ελένη	Θ	#####	Μόρφου 6
	Άγγελου	1100	Ελένη	Θ	#####	Καντάρας 11
	Αλέξη	1002	Αχιλλέας	A	#####	Καντάρας 15
	Κυριάκου	1009	Άντρεα	Θ	#####	Κυρηνείας 7
	Γκία	1127	Λένα	Θ	#####	Ριζοκαρπάσου 901
	Ρομπέρτ	1128	Λιούσι	Θ	#####	Καντάρας 10
	Γκέλα	1158	Ναταλία	Θ	#####	Κυρηνείας 38
	Σάββα	1164	Νικόλαος	A	#####	Καντάρας 36
	Νικόλαου	1004	Αλέξανδρος	A	#####	Αμμογώστου 37
	Γεώργιου	1214	Βέρα	Θ	#####	Ριζοκαρπάσου 38
	Γκέλα	1046	Χρήστος	A	#####	Καντάρας 7
	Προκόπη	1150	Μιχάλης	A	#####	Καντάρας 3

5. Τροποποίηση Έκθεσης

(1) Από την περιοχή όλων των αντικειμένων της εφαρμογής (**All Access Objects**) κάτω από τον τίτλο **Reports**, επιλέγουμε την έκθεση που θέλουμε να τροποποιήσουμε και κάνουμε **διπλό κλικ** πάνω της για να την ανοίξουμε πρώτα σε προβολή έκθεσης (**Report View**).

(2) Ακολούθως, από την καρτέλα **Home** και την ομάδα **Views** επιλέγουμε το εικονίδιο **Design View** για να ανοίξουμε την έκθεσή μας σε προβολή σχεδίασης και κάνουμε τις αλλαγές που θέλουμε ακριβώς με τον ίδιο τρόπο που κάναμε και σε μια φόρμα.



ή

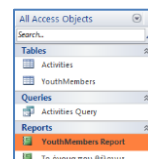
Επιλέγουμε την έκθεση από τη συγκεκριμένη περιοχή, κάνουμε δεξί κλικ πάνω της και επιλέγουμε την εντολή **Design View** για να την ανοίξει σε προβολή σχεδίασης και κάνουμε τις αλλαγές που θέλουμε ακριβώς με τον ίδιο τρόπο που κάναμε και σε μια φόρμα.

(3) Στη συνέχεια, κλείνουμε την έκθεση, αποθηκεύοντας όμως **ξανά** τις νέες αλλαγές που κάναμε στην έκθεσή μας.

6. Διαγραφή Έκθεσης (Delete Report)

(1) Από την περιοχή όλων των αντικειμένων της εφαρμογής (**All Access Objects**) κάτω από τον τίτλο **Reports**, επιλέγουμε την έκθεση που θέλουμε να διαγράψουμε.

(2) Ακολούθως, κάνουμε δεξί κλικ και επιλέγουμε την εντολή **Delete** για να τη διαγράψουμε από τη βάση δεδομένων μας.



ή

Από την καρτέλα **Home** και την ομάδα **Records** επιλέγουμε το εικονίδιο **Delete**.



Ακολούθως, κάνουμε κλικ στο κουμπί  για να τη **διαγράψουμε** από τη βάση δεδομένων μας.

ή

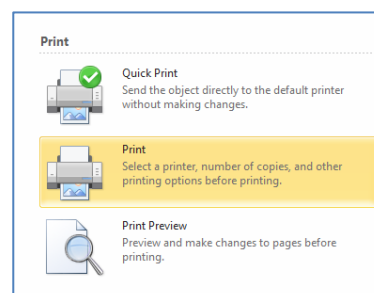
Μπορούμε, επίσης, να χρησιμοποιήσουμε το πλήκτρο **Delete**.

7. Εκτύπωση Έκθεσης (Print Report)

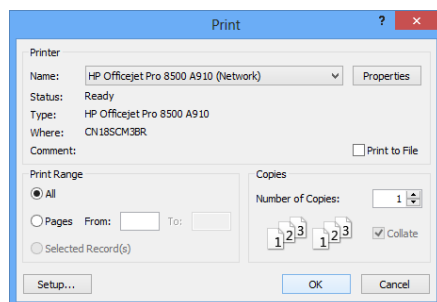
(1) Επιλέγουμε την έκθεση που θέλουμε να εκτυπώσουμε.

(2) Από την καρτέλα **File** επιλέγουμε την εντολή **Print**.

- Κάνουμε κλικ στο εικονίδιο **Quick Print** για να αποστείλουμε την έκθεση απευθείας στον προεπιλεγμένο εκτυπωτή χωρίς να ορίσουμε επιλογές εκτυπωτή.
- Κάνουμε κλικ στο εικονίδιο **Print** για να



ανοίξουμε ένα παράθυρο διαλόγου από το οποίο μπορούμε να επιλέξουμε εκτυπωτή, να καθορίσουμε τον αριθμό των αντιγράφων κ.λπ.

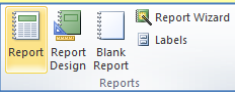
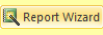
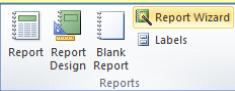
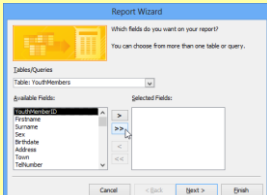
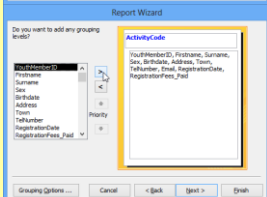
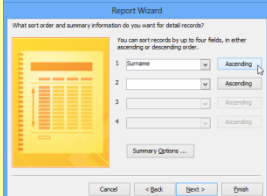


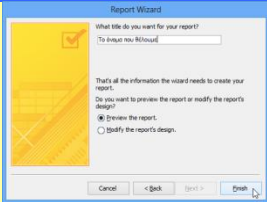
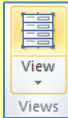
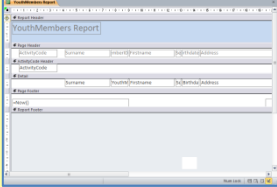
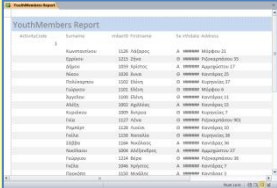
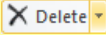
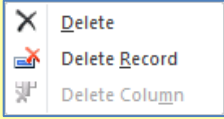
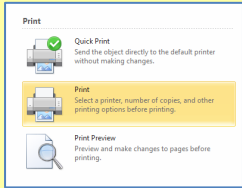
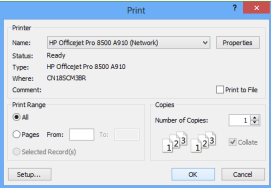
ή

Μπορούμε επίσης να χρησιμοποιήσουμε τον συνδυασμό πλήκτρων

CTRL + P.

- Κάνουμε κλικ στο εικονίδιο **Print Preview** για να ανοίξουμε την έκθεσή μας σε Προβολή Προεπισκόπησης της Εκτύπωσης.

Υπόμνημα					
<u>Περιγραφή</u> (Description)	<u>Καρτέλα</u> (Tab)	<u>Εικονίδιο</u> (Icon)	<u>Ομάδα</u> (Group)	<u>Παράθυρο Διαλόγου</u> (Dialog Box)	<u>Πλήκτρα</u> (Keys)
Δημιουργία Έκθεσης με το Εργαλείο Αυτόματης Έκθεσης (Report)					
Δημιουργία Έκθεσης με τη χρήση του Οδηγού Εκθέσεων (Report Wizard)				   	

Προβολή Σχεδίασης (Design View)	Home			
Προβολή Διάταξης Έκθεσης (Layout View)	Home			
Τροποποίηση Έκθεσης	Home			
Διαγραφή Έκθεσης (Delete Report)	Home			 
Εκτύπωση Έκθεσης (Print Report)	File			

ΕΝΟΤΗΤΑ Γ7 Αλγοριθμική Σκέψη, Προγραμματισμός και Σύγχρονες Εφαρμογές Πληροφορικής

Για την προετοιμασία και τη συγγραφή του υλικού αυτής της ενότητας εργάστηκαν οι ακόλουθοι/θες Καθηγητές/τριες Πληροφορικής και Επιστήμης Ηλεκτρονικών Υπολογιστών που παρακολούθησαν το πρόγραμμα επιμόρφωσης μάχιμων εκπαιδευτικών για το Νέο Αναλυτικό Πρόγραμμα:

Αλκείδου Μαρίνα
Ανδρέου Δημήτρης
Αντωνίου Αντώνης Φ.
Αχιλλέως Κλειώ
Γεωργίου Αντώνης
Δημητρίου Λευτέρης
Δημητρίου Παναγιώτα
Διονυσίου Μιχάλης (Σύμβουλος)
Ζαχαρίου Κυριάκος
Θεοδώρου Ηλίας (Σύμβουλος)
Κουρούσης Αντώνης
Μάκκουλα Πόλα
Μακρή Σωτικοπούλου Ελένη
Μέσσιου Παρασκευή

Νικολάου Νικολέττα
Οικονομίδης Γιώργος
Παναγίδης Μιχάλης
Παπαδοπούλου Άνθη
Παπαλυσάνδρου Αλέξανδρος (Σύμβουλος)
Παρπέρης Νικόλας
Παυλικκάς Παύλος (Σύμβουλος)
Πετρή Ιωσήφ
Συμεωνίδης Κωνσταντίνος
Τσιμούρης Χαράλαμπος
Φωτίου Παύλος
Χατζηγιάννης Γιώργος
Χατζηλοΐζου Μαρία

Γ7.1

Εισαγωγή στους Αλγορίθμους

Τι θα μάθουμε σήμερα:

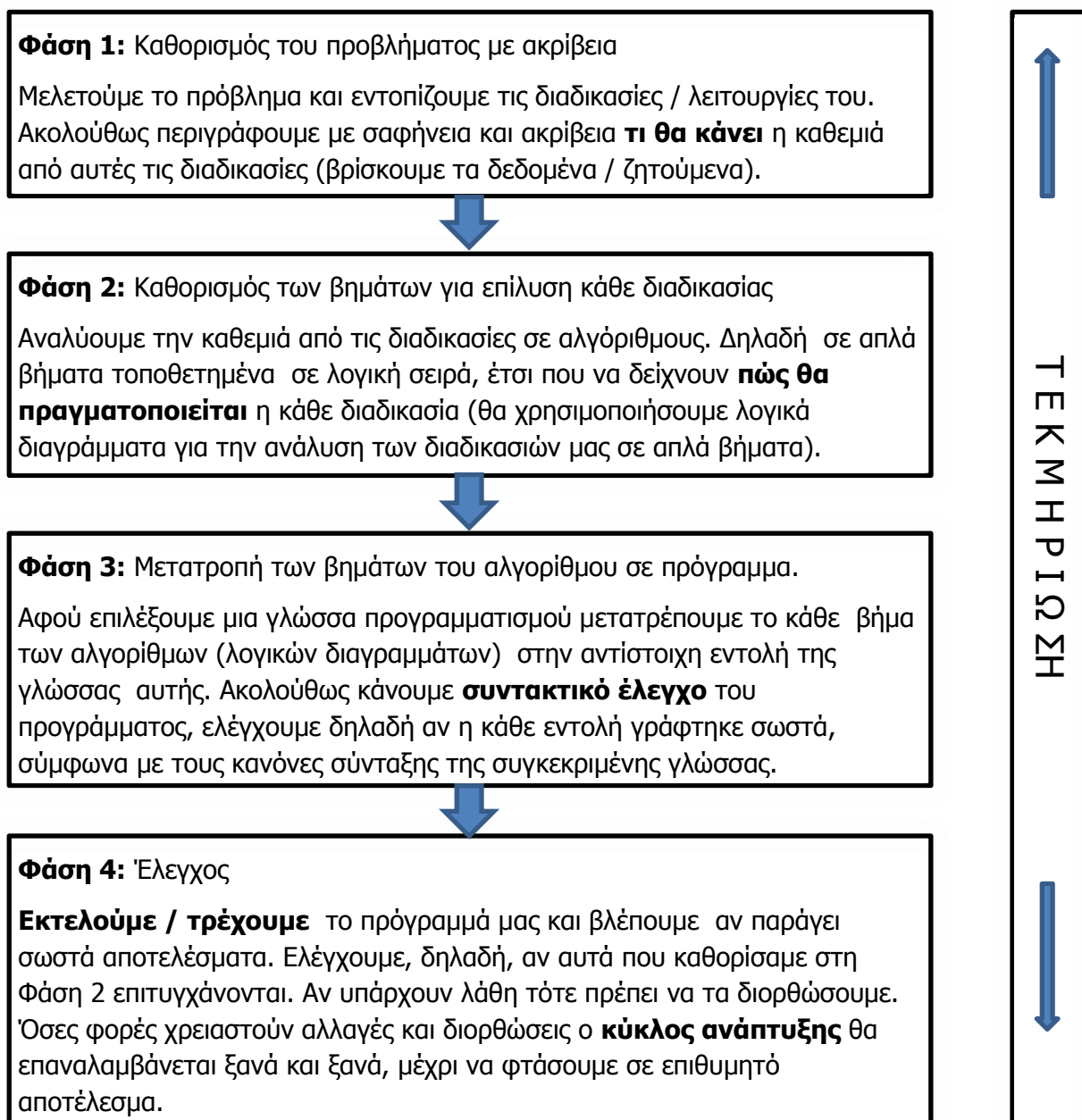
- ❖ Να αναφέρουμε και να εξηγούμε τα στάδια ανάπτυξης ενός προγράμματος
- ❖ Να εξηγούμε την έννοια του αλγορίθμου
- ❖ Να αναφέρουμε απλούς αλγόριθμους από την καθημερινότητά μας
- ❖ Να αξιολογούμε μια σειρά από βήματα/εντολές και να εξηγούμε αν αποτελούν ή όχι αλγόριθμο.
- ❖ Να αξιολογούμε έτοιμα παραδείγματα και να διακρίνουμε σε ποια από αυτά παρατηρούνται ή όχι τα χαρακτηριστικά ενός σωστού αλγορίθμου (Σαφήνεια/καθοριστικότητα, αποτελεσματικότητα, περατότητα)
- ❖ Να αναφέρουμε τρόπους διατύπωσης ενός αλγορίθμου (λεκτική περιγραφή/ψευδοκώδικας, λογικό διάγραμμα)
- ❖ Να αναλύουμε απλά προβλήματα σε αλγόριθμο χρησιμοποιώντας τη λεκτική μέθοδο
- ❖ Να βρίσκουμε τον πιο αποδοτικό αλγόριθμο σε ένα πρόβλημα (π.χ. ταξινόμηση αριθμών με διαφορετικούς τρόπους - πιο σύντομη διαδρομή για ένα οδηγό να φτάσει στον προορισμό του).

1. Ο Κύκλος Ανάπτυξης Εφαρμογής – Ποια βήματα/φάσεις πρέπει να ακολουθήσουμε για να δημιουργήσουμε ένα πρόγραμμα

Τα προγράμματα δημιουργούνται από ανθρώπους και όπως όλες οι τεχνολογικές κατασκευές (π.χ. κτίρια, αυτοκίνητα, γέφυρες, ηλεκτρικές συσκευές), για να λειτουργούν σωστά θα πρέπει να σχεδιαστούν προσεκτικά, να κατασκευαστούν όπως έχουν σχεδιαστεί και να ελεγχθούν πριν τις χρησιμοποιήσουμε. Σκεφτείτε την κατασκευή ενός μεγάλου κτιρίου. Χωρίς διάφορες λεπτομερείς μελέτες, σχεδιαγράμματα από πολιτικούς μηχανικούς, αρχιτέκτονες, ηλεκτρολογικά και πολλά άλλα σχέδια, ο εργολάβος δεν μπορεί καν να ξεκινήσει δουλειά. Για να διορθωθεί μια κατασκευή που δεν έχει σχεδιαστεί σωστά, συνήθως απαιτεί την εξασφάλιση νέων υλικών και πολύ κόπο για την επιδιόρθωσή της, συχνά με ψηλό κόστος, πράγμα που αναγκάζει τον κατασκευαστή να τη σχεδιάσει πολύ προσεκτικά. Το λογισμικό, σε αντίθεση με άλλες κατασκευές, δεν απαιτεί την αγορά υλικών και συχνά δίνεται η εντύπωση ότι δεν απαιτείται σχεδιασμός. Στην πραγματικότητα όμως, το κόστος εντοπισμού και επίλυσης λαθών στο λογισμικό, ιδιαίτερα εάν δεν έχει σχεδιαστεί σωστά, είναι μια δύσκολη, μακροχρόνια και δαπανηρή διαδικασία.

Βλέπουμε λοιπόν ότι όλες οι σοβαρές κατασκευές γίνονται με καλή προεργασία και σχεδιασμό. Το ίδιο συμβαίνει και στην πληροφορική. Για τη δημιουργία ενός προγράμματος (λογισμικού) χρειάζεται να περάσουμε μέσα από μια διαδικασία που περιλαμβάνει διάφορες φάσεις / στάδια τα οποία ονομάζονται Κύκλος Ανάπτυξης Αλγοριθμικής Εφαρμογής.

Στη συνέχεια του κεφαλαίου θα δούμε αναλυτικά όλες τις φάσεις του κύκλου ανάπτυξης και στο τέλος θα μπορούμε να δημιουργήσουμε απλά προγράμματα.



Εικόνα 21 Οι φάσεις / στάδια του Κύκλου Ανάπτυξης Προγράμματος

Στην Εικόνα 21 γίνεται αναφορά σε πολλές διαδικασίες/λειτουργίες. Τα μεγάλα, σύνθετα προγράμματα αποτελούνται από πολλές ανεξάρτητες λειτουργίες. Όλα όμως τα προγράμματα που θα μελετήσουμε στη συνέχεια θα αποτελούνται από μία μόνο διαδικασία/λειτουργία.

Τεκμηρίωση

Τεκμηρίωση είναι το σύνολο του γραπτού υλικού που συνοδεύει ένα πρόγραμμα. Δεν είναι ακόμα μια φάση στην ανάπτυξη ενός προγράμματος, αλλά μία παράλληλη διαδικασία που χρησιμοποιείται σε όλα τα στάδια ανάπτυξης ενός προγράμματος. Μεταξύ άλλων, η τεκμηρίωση, περιλαμβάνει:

Για τον χρήστη: το εγχειρίδιο χρήσης (user manual), το οποίο μας καθοδηγεί με βήμα προς βήμα οδηγίες, ώστε να χρησιμοποιήσουμε το πρόγραμμα.

Για τον προγραμματιστή: περιλαμβάνει όλα τα εργαλεία τα οποία θα κάνουν την ανάπτυξη ενός νέου προγράμματος ή την τροποποίηση ενός υφισταμένου προγράμματος ευκολότερη.

Παράδειγμα

Ένα από τα δημοφιλέστερα παιχνίδια είναι το FIFA που κατασκευάζεται από την εταιρεία *EA Sports*. Στις εκδόσεις του παιχνιδιού πριν από το 2012 παρουσιαζόταν το εξής πρόβλημα. Σε περίπτωση δυναμικής διεκδίκησης της μπάλας (π.χ. τάκλιν) φαινόταν ότι ο ένας παίκτης «έμπαινε» μέσα στον άλλο. Επίσης ο τρόπος με τον οποίον αλληλεπιδρούσαν οι παίκτες στις διάφορες διεκδικήσεις ήταν προβλέψιμος. Τα προβλήματα διορθώθηκαν στο FIFA 2012 με τη δημιουργία του **Impact Engine**, ενός προγράμματος (μιας διαδικασίας/λειτουργίας) το οποίο να διαχειρίζεται την αλληλεπίδραση μεταξύ των παικτών.



Εικόνα 22 Το impact engine του FIFA 2012

Στον παρακάτω πίνακα περιγράφονται με πολύ συνοπτικό τρόπο τα βήματα του κύκλου ανάπτυξης του Impact Engine.

Φάση 1: Η εταιρεία (EA Sports), αναγνώρισε το πρόβλημα. Αυτό έγινε τόσο από το προσωπικό της εταιρείας όσο και από τους χρήστες που άσκησαν κριτική. Αφού έγινε αναγνώριση του προβλήματος, οι αναλυτές της εταιρείας κατέγραψαν με σαφήνεια το **τι** έπρεπε να γίνει.

Φάση 2: Έχοντας μπροστά τους την ανάλυση του προβλήματος, τα άτομα της ομάδας ανάπτυξης του παιχνιδιού άρχισαν να ψάχνουν και να σχεδιάζουν τις λύσεις.

Φάση 3: Σε αυτή τη φάση οι προγραμματιστές της εταιρείας υλοποίησαν τις λύσεις που δόθηκαν από την ομάδα ανάπτυξης.

Φάση 4: Τέλος, ειδικοί gamers ανέλαβαν τον έλεγχο. Έπαιξαν το παιχνίδι πολλές ώρες προσπαθώντας να εντοπίσουν τυχόν λάθη στο Impact Engine.

Είναι φανερό ότι αν μια από τις φάσεις του κύκλου ανάπτυξης δεν σχεδιαστεί σωστά, τότε ολόκληρο το πρόγραμμα θα αποτύχει.

2. Τι είναι αλγόριθμος

Αλγόριθμος ονομάζουμε έναν αριθμό από βήματα γραμμένα σε λογική σειρά, τα οποία χρειάζονται για την εκτέλεση μιας εργασίας ή για την επίλυση ενός προβλήματος.

Η λέξη «αλγόριθμος» προέρχεται από μία μελέτη του Πέρση μαθηματικού του 8ου αιώνα μ.Χ. Αλ Χουαρίζμι (Abu Ja'far Mohammed ibn Musa Al-Khowarismi). Πέντε αιώνες αργότερα η μελέτη μεταφράστηκε στα Λατινικά και άρχισε με τη φράση "Algorithmus dixit" (ο Αλγόριθμος είπε...). Έτσι το όνομά του ταυτίστηκε με τη συστηματική, λεπτομερή και ακριβή περιγραφή του τρόπου επίλυσης κάποιου προβλήματος, η οποία είναι απαραίτητη στη διαδικασία δημιουργίας ενός προγράμματος για τον υπολογιστή.



Εικόνα 23 Ο Abu Ja'far Mohammed ibn Musa Al-Khowarismi σε αιγυπτιακό γραμματόσημο.

2.1 Παραδείγματα αλγορίθμων

Στη ζωή μας, σε καθημερινή βάση, εκτελούμε δεκάδες απλές ή σύνθετες διαδικασίες που μπορούν να περιγραφούν με αλγόριθμους. Οι διαδικασίες αυτές δεν είναι απαραίτητο να σχετίζονται με ηλεκτρονικούς υπολογιστές. Μερικές τέτοιες διαδικασίες (αλγόριθμοι) είναι οι ακόλουθες:

- Ο υπολογισμός του εμβαδού ενός γεωμετρικού σχήματος (π.χ. ενός ορθογωνίου)
- Εύρεση του μεγαλύτερου ή μικρότερου αριθμού από μία ομάδα αριθμών
- Η αποστολή ηλεκτρονικού μηνύματος μέσω του ηλεκτρονικού μας υπολογιστή
- Ο έλεγχος των φώτων της τροχαίας σε μία πόλη.
- Η πορεία που ακολουθεί καθημερινά ένας διανομέας εφημερίδων για να διανείμει τις εφημερίδες
- Ένα ηλεκτρονικό παιχνίδι με πολλά στάδια.

Τρία από τα πιο πάνω παραδείγματα αλγορίθμων αναλύονται σε απλά βήματα πιο κάτω:

Υπολογισμός εμβαδού ορθογωνίου

1. Διάβασε το μήκος ορθογωνίου
2. Διάβασε το πλάτος ορθογωνίου
3. Υπολόγισε το εμβαδόν (Εμβαδόν = μήκος $\times$ πλάτος)
4. Τύπωσε το εμβαδόν

Εύρεση του μεγαλύτερου αριθμού από μια ομάδα 3 διαφορετικών αριθμών

1. Διάβασε τους 3 αριθμούς (a,b,c)
2. Αν $a > b$ και $a > c$ τότε
3. Τύπωσε a
4. Αλλιώς αν $b > a$ και $b > c$ τότε
5. Τύπωσε b
6. Αλλιώς
7. Τύπωσε c

Αποστολή ηλεκτρονικού μηνύματος (e-mail)

1. Μπαίνω στην ιστοσελίδα του παροχέα μου.
2. Πληκτρολογώ το User Name μου
3. Πληκτρολογώ το Password μου
4. Επιλέγω «Νέο Μήνυμα»
5. Πληκτρολογώ ή επιλέγω την ηλεκτρονική διεύθυνση του παραλήπτη
6. Γράφω το μήνυμα
7. Επιλέγω «Αποστολή»
8. Επιλέγω «Έξοδος»

Πρέπει όμως να τονίσουμε ότι μια σειρά από βήματα δεν αποτελούν απαραίτητα αλγόριθμο. Για παράδειγμα:

Οι κανόνες σχολικής συμπεριφοράς στην τάξη.

1. Ακούω προσεκτικά τον/την καθηγητή/τριά μου!
2. Προσφέρω τη βοήθειά μου στους συμμαθητές μου!
3. Σέβομαι και αγαπώ τους συμμαθητές μου!
4. Σηκώνω το χέρι μου όταν θέλω να μιλήσω!
5. Ακούω με προσοχή τον συμμαθητή μου!
6. Ακολουθώ τις οδηγίες του/της καθηγητή/τριάς μου!
7. Φροντίζω για την καθαριότητα της τάξης μου!

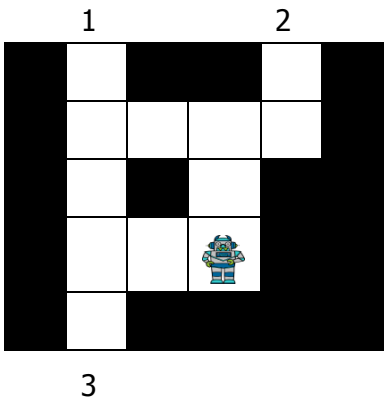
Οι οδηγίες αυτές δεν αποτελούν αλγόριθμο, είναι ασύνδετες μεταξύ τους, δεν έχουν κάποια λογική σειρά και δεν οδηγούν στην παραγωγή κάποιου συγκεκριμένου αποτελέσματος.

3. Τα χαρακτηριστικά ενός αλγορίθμου

Είναι ξεκάθαρο ότι τα βήματα του προηγούμενου παραδείγματος δεν αποτελούν αλγόριθμο. Ένας αλγόριθμος δεν αποτελείται από μια οποιαδήποτε σειρά βημάτων ή εντολών αλλά μία σειρά από βήματα ή εντολές σε **λογική σειρά** και πρέπει επιπρόσθετα να έχει τα πιο κάτω χαρακτηριστικά:

- **Σαφήνεια / Καθοριστικότητα:** Κάθε βήμα (εντολή) του αλγορίθμου πρέπει να είναι απλό και να είναι διατυπωμένο με σαφήνεια, ώστε να μην υπάρχει αμφιβολία για τον τρόπο με τον οποίο θα εκτελεσθεί. Πρέπει, επίσης, να μπορεί να εκτελεσθεί με ακρίβεια μέσα σε συγκεκριμένα χρονικά πλαίσια (πεπερασμένος). Για παράδειγμα, η εντολή «ηροθερμαίνω τον φούρνο» δεν είναι σαφής, διότι δεν προσδιορίζει για πόσο χρονικό διάστημα προθερμαίνω τον φούρνο.
- **Περατότητα:** Κάθε αλγόριθμος πρέπει πάντα να τελειώνει ύστερα από την εκτέλεση ενός αριθμού βημάτων, με άλλα λόγια δεν μπορεί να συνεχίζει την εκτέλεσή του για πάντα.
- **Αποτελεσματικότητα:** Να δίνει συγκεκριμένα αποτελέσματα, τα οποία να μπορούν να εξακριβωθούν από έναν άνθρωπο, εκτελώντας με «μολύβι και χαρτί» τις εντολές / βήματα του αλγορίθμου. Επίσης οι εντολές αυτές θα πρέπει να είναι διατυπωμένες έτσι ώστε να μπορούν να εκτελεστούν από αυτόν στον οποίο απευθύνεται ο αλγόριθμος.

Ο πιο κάτω πίνακας παρουσιάζει μερικά παραδείγματα «αλγορίθμων» που δεν έχουν όλα τα πιο πάνω χαρακτηριστικά.

Βρες τον μικρότερο από 3 αριθμούς 1. Διαβάζω αριθμούς 2. Βρίσκω τον μικρότερο 3. Τυπώνω τον μικρότερο	Πρόβλημα σαφήνειας/καθοριστικότητας (Πόσους αριθμούς διαβάζω; Πώς βρίσκω τον μικρότερο;)
Το ρομπότ που βλέπετε προσπαθεί να βγει από τον λαβύρινθο σε μια από τις εξόδους 1, 2 ή 3. Αν καταφέρει να βγει τότε ο αλγόριθμος τερματίζεται. Μπορεί να κινείται μόνο μέσα από άσπρα κουτάκια και είναι προγραμματισμένο να κινείται ως εξής: 1. Κάνε επιτόπου στροφή δεξιά 2. Προχώρα 2 κουτιά ευθεία μπροστά 3. Αν βγεις σε κάποια από τις 3 εξόδους σταμάτα. Διαφορετικά επανέλαβε τα βήματα 1 και 2. 	Πρόβλημα περατότητας (Ο αλγόριθμος δεν θα σταματήσει ποτέ γιατί το ρομπότ θα κάνει κύκλους μέσα στον λαβύρινθο και δεν θα καταφέρει να βγει σε μια από τις εξόδους).
Υπολογισμός μέσου όρου 20 αριθμών 1. Διάβασε 20 αριθμούς 2. Υπολόγισε τον μέσο όρο 3. Τύπωσε τον μέσο όρο.	Πρόβλημα αποτελεσματικότητας Η 2^η εντολή δεν είναι απλή και δεν μπορεί να εκτελεστεί απευθείας από τον υπολογιστή. Πρέπει να υπολογιστεί πρώτα το άθροισμα των 20 αριθμών και μετά ο μέσος τους όρος.

4. Τρόποι διατύπωσης αναπαράστασης αλγορίθμων

Έναν αλγόριθμο μπορούμε να τον διατυπώσουμε με τρεις τρόπους: Λεκτική Περιγραφή, Ψευδοκώδικα (pseudocode), και Λογικό Διάγραμμα (flowchart).

- Λεκτική περιγραφή: Περιγραφή του αλγορίθμου σε βήματα χρησιμοποιώντας λέξεις από την καθημερινή μας ζωή.
- Ψευδοκώδικας: Πιο αυστηρή διατύπωση του αλγορίθμου κατά την οποία χρησιμοποιούνται καθορισμένες εντολές/λέξεις και συμβολισμοί.
- Λογικό Διάγραμμα: Ένας γραφικός τρόπος παρουσίασης ενός αλγορίθμου, που αποτελείται από σύμβολα με συγκεκριμένη σημασία, που συνδέονται μεταξύ τους με βέλη, που δείχνουν τη ροή της εκτέλεσης.

Παράδειγμα : Σε πόσα χρόνια ένας ανήλικος θα δικαιούται να βγάλει δίπλωμα οδήγησης;

Λεκτική Περιγραφή	Ψευδοκώδικας	Λογικό Διάγραμμα
1. Διάβασε την ηλικία 2. Αφαίρεσε την ηλικία από τον αριθμό 18 3. Τύπωσε τη διαφορά	1. Διάβασε age 2. $X = 18 - \text{age}$ 3. Τύπωσε X	<pre> graph TD A([Αρχή]) --> B[/Διάβασε age/] B --> C[X = 18 - age] C --> D[/Τύπωσε X/] D --> E([Τέλος]) </pre>

5. Αποδοτικός Αλγόριθμος

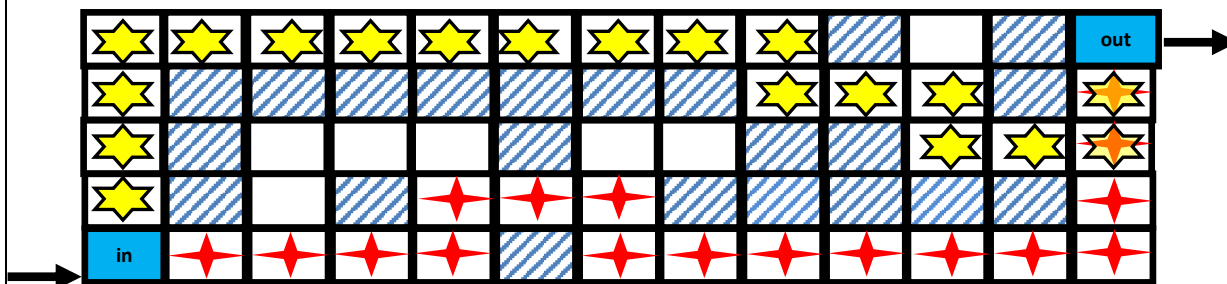
Συνήθως παρατηρούμε ότι πολλά προβλήματα της καθημερινότητάς μας επιλύονται με περισσότερους από έναν τρόπους. Για παράδειγμα:

- Για να περπατήσουμε από το σπίτι μέχρι το σχολείο μας. Συνήθως υπάρχουν περισσότερες από μια διαδρομές που μπορούμε να ακολουθήσουμε.
- Η λύση ενός προβλήματος στα μαθηματικά. Πολύ συχνά βλέπουμε ότι κάποιος συμμαθητής μας έλυσε την ίδια άσκηση με διαφορεικό τρόπο από τον δικό μας.

Σε όλες τις περιπτώσεις είναι λογικό να ψάχνουμε για την καλύτερη και αποδοτικότερη λύση. Έτσι, θα ακολουθήσουμε τη συντομότερη διαδρομή για να πάμε στο σχολείο μας και θα λύσουμε την άσκηση των μαθηματικών με τον τρόπο που θεωρούμε ευκολότερο. Παρομοίως στην πληροφορική μπορεί να υπάρχουν περισσότεροι από ένας αλγόριθμοι για τη λύση του ίδιου προβλήματος. Ένας αλγόριθμος είναι πιο **αποδοτικός** από κάποιον άλλον αν εκτελείται γρηγορότερα και κάνει χρήση όσο το δυνατόν λιγότερου χώρου μνήμης.

Πιο κάτω παρουσιάζονται τρία παραδείγματα προβλημάτων. Για το κάθε πρόβλημα προτείνονται δύο διαφορετικοί αλγόριθμοι οι οποίοι αποτελούν λύσεις στο πρόβλημα. Όπως επεξηγείται, αν συγκρίνουμε τους αλγόριθμους αυτούς συχνά κάποιος είναι αποδοτικότερος από τον άλλον.

Παράδειγμα 1 : για να εισέλθει κάποιος από την είσοδο (in) και να εξέλθει από την έξοδο (out) έχει δύο επιλογές (α) ακολουθώντας τη διαδρομή με τα αστεράκια, (β) ακολουθώντας τη διαδρομή με τους σταυρούς. Ποια επιλογή / διαδρομή είναι πιο αποδοτική;



Απάντηση: η διαδρομή (β) γιατί είναι συντομότερη 17 βήματα ενώ η (α) χρειάζεται 19.

Παράδειγμα 2: Ετοιμασία μέτριου frappe. Ποιος από τους δύο αλγορίθμους είναι αποδοτικότερος;

- | | |
|--|--|
| 1. Βάζουμε λίγο νερό σε ένα ποτήρι | 1. Βάζουμε λίγο νερό σε ένα ποτήρι |
| 2. Βάζουμε 1 κουταλάκι στιγμιαίου καφέ | 2. Βάζουμε 1 κουταλάκι ζάχαρη |
| 3. Βάζουμε 1 κουταλάκι ζάχαρη | 3. Βάζουμε 1 κουταλάκι στιγμιαίου καφέ |
| 4. Ανακατεύουμε με το mixer | 4. Ανακατεύουμε με το mixer |
| 5. Προσθέτουμε και άλλο νερό | 5. Προσθέτουμε και άλλο νερό |
| 6. Βάζουμε λίγο γάλα | 6. Βάζουμε 2-3 παγάκια |
| 7. Βάζουμε 2-3 παγάκια | 7. Βάζουμε λίγο γάλα |

Απάντηση: κανένας, και οι δύο έχουν την ίδια απόδοση, απλά κάποια από τα βήματα τους δεν είναι στην ίδια σειρά.

Παράδειγμα 3: εξεύρεση του ελάχιστου κοινού πολλαπλάσιου (ΕΚΠ) δύο ακεραίων αριθμών α και β όπου $\alpha \leq \beta$ (για παράδειγμα το ΕΚΠ των αριθμών 3 και 4 είναι το 12).

1. Διάβασε έναν αριθμό α 2. Διάβασε ένα μεγαλύτερο αριθμό β 3. Όρισε τον ακέραιο X ίσο με β 4. Αν ο X διαιρείται ακριβώς και από τους δύο ακεραίους α και β τότε τύπωσε τον X (που είναι το ΕΚΠ) και σταμάτα διαφορετικά αύξησε τον X κατά μια μονάδα και επανέλαβε το βήμα 4.	1. Διάβασε έναν αριθμό α 2. Διάβασε ένα μεγαλύτερο αριθμό β 3. Όρισε τον ακέραιο X ίσο με β 4. Αν ο X διαιρείται ακριβώς και από τους δύο ακεραίους α και β τότε τύπωσε τον X (που είναι το ΕΚΠ) και σταμάτα διαφορετικά αύξησε τον X κατά β μονάδες και επανέλαβε το βήμα 4.																														
Π.χ Είσοδος $\alpha=3$ $\beta=4$ <table> <tr><td>(1) $\alpha=3$</td><td>(Βήμα 1)</td></tr> <tr><td>(2) $\beta=4$</td><td>(Βήμα 2)</td></tr> <tr><td>(3) $X=4$</td><td>(Βήμα 3)</td></tr> <tr><td>(4) 4 δεν διαιρείται με 3, αλλά με 4 αρα $X=5$</td><td>(Βήμα 4)</td></tr> <tr><td>(5) 5 δεν διαιρείται με 3, ούτε με 4, αρα $X=6$</td><td>(Βήμα 4)</td></tr> <tr><td>(6) 6 διαιρείται με 3, αλλά όχι με 4, αρα $X=7$</td><td>(Βήμα 4)</td></tr> <tr><td>...</td><td></td></tr> <tr><td>(11) 10 δεν διαιρείται με 3, ούτε με 4, αρα $X=12$</td><td>(Βήμα 4)</td></tr> <tr><td>(12) 12 διαιρείται με 3 και με 4, αρα ΕΚΠ=12</td><td>(Βήμα 4)</td></tr> </table> Εδώ χρειάζονται 12 βήματα.	(1) $\alpha=3$	(Βήμα 1)	(2) $\beta=4$	(Βήμα 2)	(3) $X=4$	(Βήμα 3)	(4) 4 δεν διαιρείται με 3, αλλά με 4 α ρα $X=5$	(Βήμα 4)	(5) 5 δεν διαιρείται με 3, ούτε με 4, α ρα $X=6$	(Βήμα 4)	(6) 6 διαιρείται με 3, αλλά όχι με 4, α ρα $X=7$	(Βήμα 4)	...		(11) 10 δεν διαιρείται με 3, ούτε με 4, α ρα $X=12$	(Βήμα 4)	(12) 12 διαιρείται με 3 και με 4, α ρα ΕΚΠ=12	(Βήμα 4)	Π.χ Είσοδος $\alpha=3$ $\beta=4$ <table> <tr><td>(1) $\alpha=3$</td><td>(Βήμα 1)</td></tr> <tr><td>(2) $\beta=4$</td><td>(Βήμα 2)</td></tr> <tr><td>(3) $X=4$</td><td>(Βήμα 3)</td></tr> <tr><td>(4) 4 δεν διαιρείται με 3, αλλά με 4 αρα $X=9$</td><td>(Βήμα 4)</td></tr> <tr><td>(5) 9 διαιρείται με 3, αλλά όχι με 4, αρα $X=12$</td><td>(Βήμα 4)</td></tr> <tr><td>(6) 12 διαιρείται με 3 και με 4, αρα ΕΚΠ=12</td><td>(Βήμα 4)</td></tr> </table> Εδώ χρειάζονται μόνο 6 βήματα. Άρα ο αλγόριθμος είναι πιο αποδοτικός. (Η διαφορά γίνεται ακόμη πιο μεγάλη εάν οι αριθμοί α και β είναι μεγαλύτεροι, αφού το βήμα 4 επαναλαμβάνεται πολύ περισσότερες φορές στον αλγόριθμο αριστερά παρά στον αλγόριθμο πιο πάνω.)	(1) $\alpha=3$	(Βήμα 1)	(2) $\beta=4$	(Βήμα 2)	(3) $X=4$	(Βήμα 3)	(4) 4 δεν διαιρείται με 3, αλλά με 4 α ρα $X=9$	(Βήμα 4)	(5) 9 διαιρείται με 3, αλλά όχι με 4, α ρα $X=12$	(Βήμα 4)	(6) 12 διαιρείται με 3 και με 4, α ρα ΕΚΠ=12	(Βήμα 4)
(1) $\alpha=3$	(Βήμα 1)																														
(2) $\beta=4$	(Βήμα 2)																														
(3) $X=4$	(Βήμα 3)																														
(4) 4 δεν διαιρείται με 3, αλλά με 4 α ρα $X=5$	(Βήμα 4)																														
(5) 5 δεν διαιρείται με 3, ούτε με 4, α ρα $X=6$	(Βήμα 4)																														
(6) 6 διαιρείται με 3, αλλά όχι με 4, α ρα $X=7$	(Βήμα 4)																														
...																															
(11) 10 δεν διαιρείται με 3, ούτε με 4, α ρα $X=12$	(Βήμα 4)																														
(12) 12 διαιρείται με 3 και με 4, α ρα ΕΚΠ=12	(Βήμα 4)																														
(1) $\alpha=3$	(Βήμα 1)																														
(2) $\beta=4$	(Βήμα 2)																														
(3) $X=4$	(Βήμα 3)																														
(4) 4 δεν διαιρείται με 3, αλλά με 4 α ρα $X=9$	(Βήμα 4)																														
(5) 9 διαιρείται με 3, αλλά όχι με 4, α ρα $X=12$	(Βήμα 4)																														
(6) 12 διαιρείται με 3 και με 4, α ρα ΕΚΠ=12	(Βήμα 4)																														

Βασικές Έννοιες

Κύκλος Ανάπτυξης μιας Εφαρμογής/ενός Προγράμματος:

Τα βήματα/φάσεις που πρέπει να ακολουθήσουμε για να δημιουργήσουμε ένα πρόγραμμα.

1^η Φάση—Καθορισμός του προβλήματος με ακρίβεια:

Μελετούμε προσεκτικά το πρόβλημά μας για να εντοπίσουμε τις διαδικασίες/λειτουργίες που το αποτελούν και περιγράφουμε με σαφήνεια και ακρίβεια **τι θα κάνει** η καθεμιά από αυτές τις διαδικασίες, ποιά **δεδομένα** δέχεται και ποιές **πληροφορίες (ζητούμενα)** δημιουργεί.

2^η Φάση—Καθορισμός των βημάτων επίλυσης της κάθε διαδικασίας:

Αναλύουμε την κάθε διαδικασία σε **αλγόριθμο** ώστε να διατυπώνεται με σαφήνεια και ακρίβεια **πώς θα πραγματοποιείται** η κάθε διαδικασία.

3^η Φάση–Μετατροπή των βημάτων σε πρόγραμμα:	Μετατρέπουμε τον αλγόριθμο σε πρόγραμμα, μεταφράζοντάς τον σε σειρές από εντολές σε μια γλώσσα προγραμματισμού.
4^η Φάση–Έλεγχος προγράμματος:	Εκτελούμε το πρόγραμμα και ελέγχουμε κατά πόσο κάνει αυτά που καταγράψαμε στη 2 ^η φάση.
Τεκμηρίωση:	Είναι το σύνολο του γραπτού υλικού που συνοδεύει ένα πρόγραμμα. Περιλαμβάνει οδηγίες για τον χρήστη και σημαντικές πληροφορίες για τον προγραμματιστή.
Αλγόριθμος:	Είναι ένας αριθμός από βήματα ή εντολές, γραμμένα σε λογική σειρά, τα οποία χρειάζονται για την εκτέλεση μιας εργασίας ή για την επίλυση ενός προβλήματος.
Χαρακτηριστικά Αλγορίθμου:	Σαφήνεια/Καθοριστικότητα: Κάθε βήμα (εντολή) του αλγορίθμου πρέπει να είναι απλό και να είναι διατυπωμένο με σαφήνεια. Επίσης, πρέπει να εκτελείται με ακρίβεια μέσα σε συγκεκριμένα πλαίσια. Περατότητα: Κάθε αλγόριθμος πρέπει πάντα να τελειώνει ύστερα από την εκτέλεση ενός αριθμού βημάτων, με άλλα λόγια δεν μπορεί να συνεχίζει την εκτέλεσή του για πάντα. Αποτελεσματικότητα: Να δίνει συγκεκριμένα αποτελέσματα, τα οποία να μπορούν να εξακριβωθούν από έναν άνθρωπο, εκτελώντας με «μολύβι και χαρτί» τις εντολές / βήματα του αλγορίθμου. Επίσης οι εντολές αυτές θα πρέπει να είναι διατυπωμένες έτσι ώστε να μπορούν να εκτελεστούν από αυτόν στον οποίο απευθύνεται ο αλγόριθμος.
Λεκτική περιγραφή:	Περιγραφή ενός αλγορίθμου σε βήματα χρησιμοποιώντας λέξεις από την καθημερινή μας ζωή.
Ψευδοκώδικας:	Πιο αυστηρή διατύπωση του αλγορίθμου όπου χρησιμοποιούνται καθορισμένες εντολές/λέξεις και συμβολισμοί.
Λογικό Διάγραμμα:	Ένας γραφικός τρόπος παρουσίασης ενός αλγορίθμου, με γεωμετρικά σχήματα.

Γ7.2

Αναπαράσταση Αλγορίθμων με Λογικό Διάγραμμα

Τι θα μάθουμε σήμερα:

- Να αναλύουμε ένα πρόβλημα και να καταγράφουμε τα Δεδομένα, τα Ζητούμενα (Πληροφορίες) και την Επεξεργασία
- Να σχεδιάζουμε τα σύμβολα που χρησιμοποιούνται σε ένα λογικό διάγραμμα και να εξηγούμε τη χρήση του καθενός από αυτά
- Να χρησιμοποιούμε το κατάλληλο σχήμα για την αναπαράσταση μιας εντολής
- Να διακρίνουμε σταθερές και μεταβλητές ανάμεσα στα βήματα του αλγορίθμου.

1. Λογικά Διαγράμματα

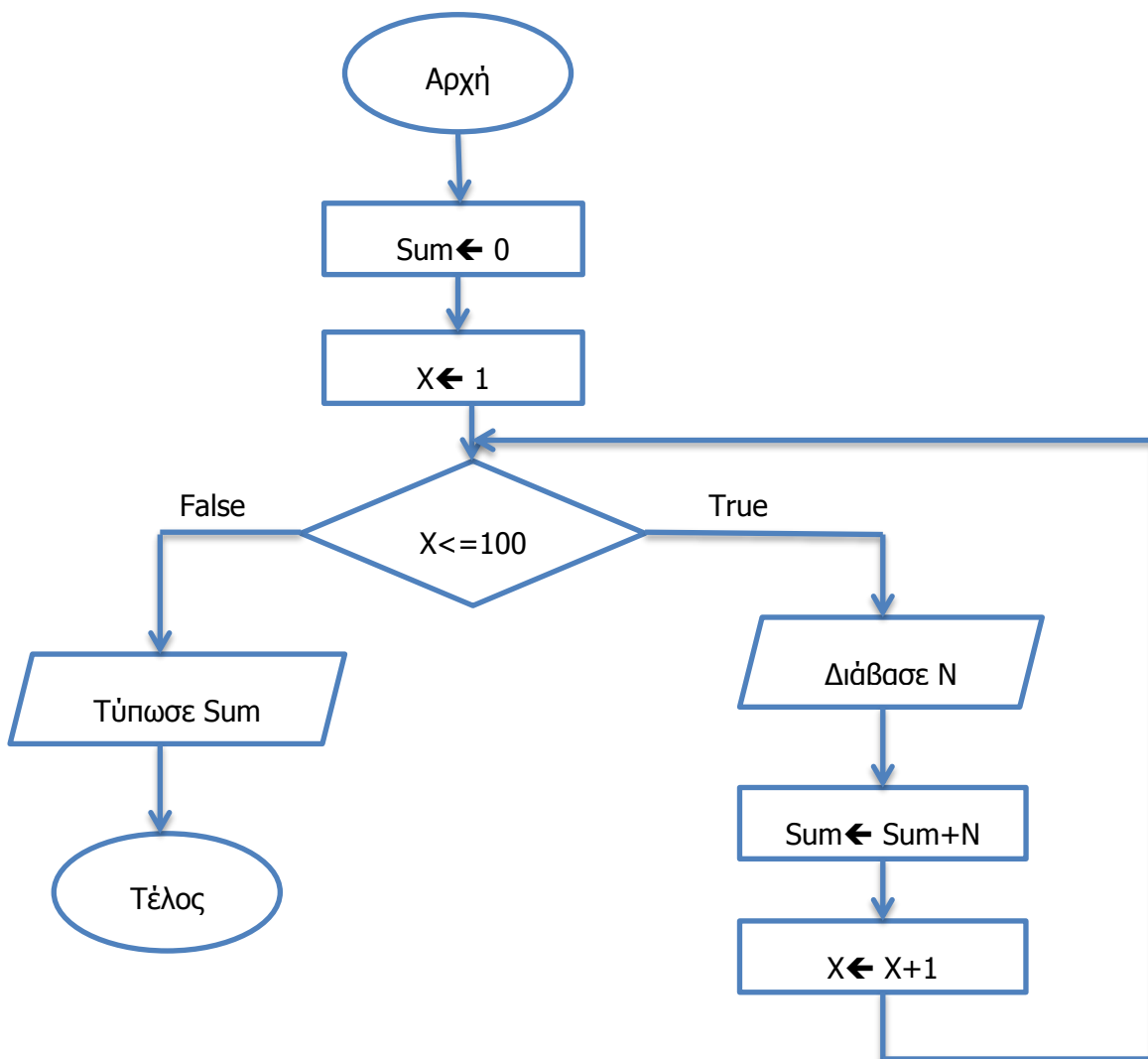
Στο προηγούμενο κεφάλαιο εξετάσαμε την έννοια του αλγορίθμου και αναφέραμε διάφορους τρόπους αναπαράστασης του. Η μέθοδος που θα χρησιμοποιήσουμε στα επόμενα μαθήματα είναι αυτή των λογικών διαγραμμάτων. Τα **λογικά διαγράμματα (flowcharts)** είναι ένας σχηματικός τρόπος για να αναπαριστούμε τη λύση ενός προβλήματος. Αποτελούνται από ένα σύνολο γεωμετρικών σχημάτων, όπου το καθένα δηλώνει μία συγκεκριμένη ενέργεια. Τα γεωμετρικά σχήματα ενώνονται μεταξύ τους με βέλη, που δηλώνουν τη σειρά εκτέλεσης των ενεργειών αυτών. Τα κυριότερα γεωμετρικά σχήματα που χρησιμοποιούμε είναι τα εξής:

Σύμβολο	Χρήση
	Αρχή / Τέλος
	Είσοδος Δεδομένων / Έξοδος Πληροφοριών
	Πράξη / Επεξεργασία
	Ερώτηση/ Απόφαση
	Πορεία Ενεργειών

Τα κυριότερα πλεονεκτήματα των λογικών διαγραμμάτων είναι η περιεκτικότητά τους, δηλαδή η επεξήγηση πολύπλοκων εννοιών με σύντομο και απλό τρόπο, καθώς και η ανεξαρτησία που παρέχουν όσον αφορά τη γλώσσα, αφού είναι εύκολο να γίνουν κατανοητά ακόμα και μεταξύ ατόμων που μιλούν διαφορετική γλώσσα. Επίσης, ο καθένας από εμάς θα περιγράψει τη λύση ενός προβλήματος με διαφορετικό τρόπο, αν χρησιμοποιήσει φυσική γλώσσα. Με τα λογικά διαγράμματα υπάρχει μια δομή που πρέπει να ακολουθείται και μας δίνει συγκεκριμένα αποτελέσματα.

Στην Εικόνα 24 παρουσιάζεται το λογικό διάγραμμα για το πρόγραμμα που δέχεται 100 αριθμούς και υπολογίζει το άθροισμά τους. Για το λογικό διάγραμμα χρειάστηκε να γράψουμε

μόλις 63 χαρακτήρες (17 λέξεις). Η περιγραφή της λύσης του ίδιου προβλήματος με τη χρήση φυσικής γλώσσας, όπως παρουσιάζεται στην Εικόνα 25, απαιτεί 407 χαρακτήρες (82 λέξεις). Επίσης, είναι φανερό ότι για να διαβάσει κάποιος την περιγραφή στη φυσική γλώσσα θα πρέπει να γνωρίζει τη γλώσσα στην οποία είναι γραμμένη, ενώ το λογικό διάγραμμα μπορεί να διαβαστεί εύκολα από τον οποιονδήποτε.



Εικόνα 24 Λογικό διάγραμμα

Για να υπολογίσουμε το άθροισμα 100 αριθμών πρέπει πρώτα να μηδενίσουμε το συνολικό άθροισμα (sum). Στη συνέχεια χρησιμοποιούμε έναν μετρητή (x) για να μετρήσουμε τους αριθμούς που θα δώσουμε. Στην αρχή θέτουμε τον μετρητή μας ίσο με 1. Όσο ο μετρητής είναι μικρότερος ή ίσος με το 100 θα διαβάζουμε έναν αριθμό (N) και θα τον προσθέτουμε στο άθροισμα. Κάθε φορά που θα δίνουμε έναν αριθμό θα αυξάνουμε τον μετρητή μας κατά 1. Όταν δώσουμε και τους 100 αριθμούς παρουσιάζουμε το άθροισμα.

Εικόνα 25 Φυσική Γλώσσα

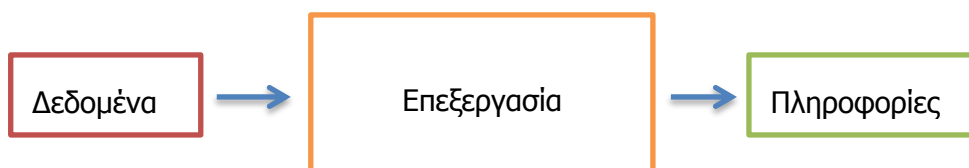
2. Δεδομένα, Επεξεργασία, Πληροφορίες (Ζητούμενα),

Έχουμε αναφέρει ότι το πρώτο στάδιο ανάπτυξης ενός προγράμματος είναι η κατανόηση του προβλήματος. Η κατανόηση του προβλήματος είναι η σωστή και πλήρης αποσαφήνιση των δεδομένων και ζητούμενων του προβλήματος.

Η πρώτη μας ενέργεια λοιπόν για να επιλύσουμε ένα πρόβλημα, είναι η καταγραφή των δεδομένων. Τα δεδομένα είναι τα στοιχεία που μας είναι γνωστά και μπορούν να μας βοηθήσουν στη λύση του προβλήματος.

Σε κάθε πρόβλημα ψάχνουμε να βρούμε την απάντηση σε μια ερώτηση. Αυτό που ψάχνουμε είναι το ζητούμενο (πληροφορίες).

Η διαδικασία μέσω της οποίας βρίσκουμε τις πληροφορίες από τα δεδομένα, ονομάζεται επεξεργασία.



Εικόνα 26 Δεδομένα – Επεξεργασία – Πληροφορίες

3. Σταθερές και Μεταβλητές

Έχουμε αναφέρει ότι ένα από τα χαρακτηριστικά ενός αλγορίθμου είναι η γενικότητα, δηλαδή να μπορεί να επιλύσει παρόμοιας φύσης προβλήματα. Αν θέλω για παράδειγμα να δημιουργήσω ένα πρόγραμμα που θα βρίσκει το εμβαδόν ενός τετραγώνου, θα πρέπει το πρόγραμμά μου να μπορεί να υπολογίσει το αποτέλεσμα ανεξάρτητα από το μήκος της πλευράς που θα του δίνω. Είναι φανερό ότι για διαφορετικές τιμές της πλευράς θα έχω και διαφορετικά αποτελέσματα. Στο παρόν στάδιο μπορούμε να ονομάσουμε μεταβλητή σε έναν αλγόριθμο τα δεδομένα/πληροφορίες που αλλάζουν (μεταβάλλονται) κατά τη διάρκεια της εκτέλεσης του αλγορίθμου. Σε μεταγενέστερο στάδιο θα δούμε ότι οι μεταβλητές είναι θέσεις στην κύρια μνήμη του υπολογιστή μας που αποθηκεύουν προσωρινά δεδομένα/πληροφορίες.

Από την άλλη μεριά, έχουμε τις σταθερές. Όπως αναφέρει και το όνομά τους, οι σταθερές δεν μπορούν να αλλάζουν περιεχόμενο. Σκεφτείτε ότι έχετε ένα πρόβλημα στο οποίο πρέπει να χρησιμοποιήσετε τον αριθμό π (π.χ. να υπολογίσετε το εμβαδόν ενός κύκλου). Ο αριθμός π είναι ένας άρρητος αριθμός του οποίου έχουμε συνηθίσει να χρησιμοποιούμε μόνο τα πρώτα 2 δεκαδικά ψηφία. Τι γίνεται όμως αν θέλουμε να χρησιμοποιήσουμε τα πρώτα 10 δεκαδικά ψηφία σε πολλές πράξεις; Για να αποφύγουμε να γράφουμε κάθε φορά ολόκληρο τον αριθμό ορίζουμε μια σταθερά (π.χ. $p = 3.1415926535$) και πλέον όταν θα αναφερόμαστε σε αυτή θα έχει την αρχική τιμή που της δώσαμε.

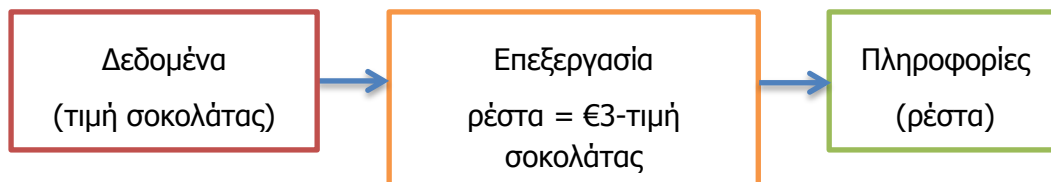
4. Αναπαράσταση αλγορίθμου με τη χρήση λογικών διαγραμμάτων

Τα βήματα που πρέπει να ακολουθήσουμε για να σχεδιάσουμε ένα λογικό διάγραμμα είναι τα ακόλουθα:

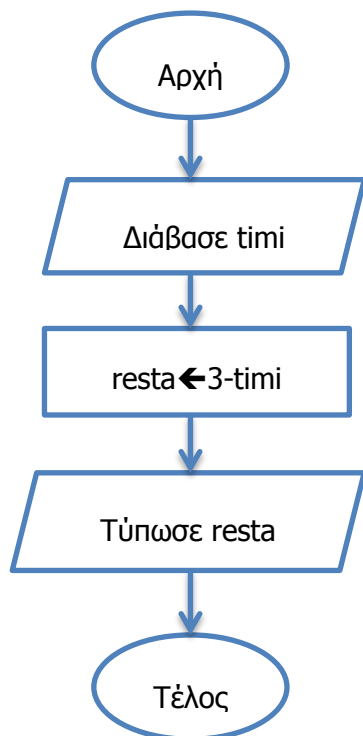
- (1) Τα λογικά διαγράμματα πρέπει πάντα να έχουν μία αρχή και ένα τέλος.
- (2) Βρίσκουμε τα δεδομένα (εισόδους) και χρησιμοποιούμε το κατάλληλο σχήμα (παραλληλόγραμμο γράφοντας μέσα τη λέξη **Διάβασε**).
- (3) Βρίσκουμε τις διαδικασίες (αριθμητικές πράξεις – αποφάσεις) που πρέπει να χρησιμοποιήσουμε για να πάρουμε τις πληροφορίες/ ζητούμενα.
- (4) Παρουσιάζουμε τις πληροφορίες (παραλληλόγραμμο γράφοντας μέσα τη λέξη **Τύπωσε**).

Παράδειγμα 1

Βρίσκεστε στην καντίνα του σχολείου σας και θέλετε να αγοράσετε μια σοκολάτα. Ο καντινιέρης σας πληροφορεί για την τιμή της σοκολάτας. Στο πορτοφόλι σας έχετε €3. Πόσα ρέστα θα πάρετε;



Μετατρέποντας το πιο πάνω σχήμα σε λογικό διάγραμμα έχουμε:



Μερικές παρατηρήσεις για το λογικό διάγραμμα:

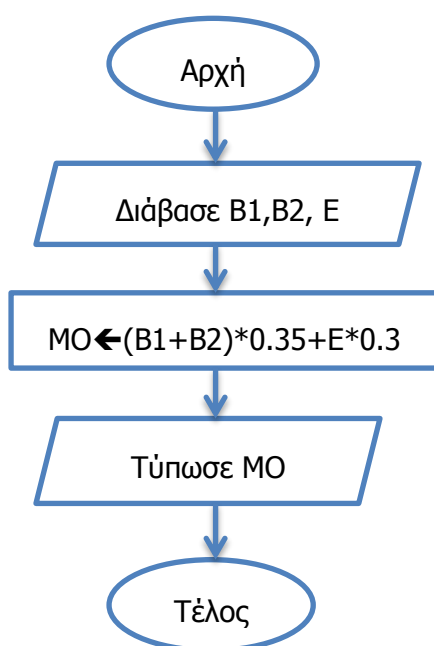
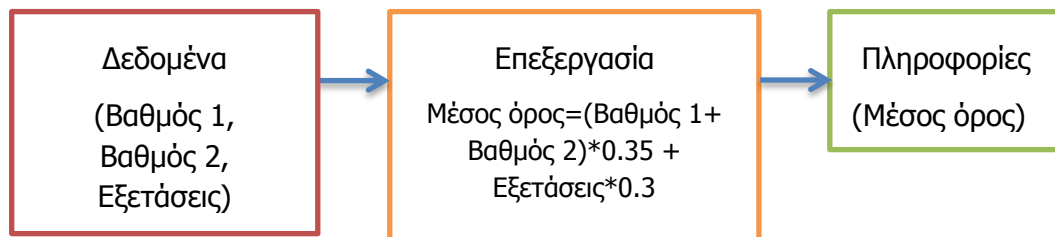
- (1) Για σκοπούς ευκολίας αντικαταστήσαμε το «τιμή σοκολάτας» με τη λέξη *timi*
- (2) Οι μόνες λέξεις στα ελληνικά που χρησιμοποιούμε είναι Αρχή, Τέλος, Διάβασε, Τύπωσε. Για οτιδήποτε άλλο χρησιμοποιούμε λατινικούς χαρακτήρες. Θα μπορούσαμε ακόμα να αντικαταστήσουμε τις ελληνικές λέξεις με τις λέξεις Start, End, Read, Print/Write.
- (3) Το $\leftarrow$ μπορεί να αντικατασταθεί από το =
- (4) Στη θέση των λέξεων *timi* και *resta* θα μπορούσαμε να βάλουμε οτιδήποτε (π.χ. X, Y). Είναι όμως καλή ιδέα να χρησιμοποιούμε λέξεις που περιγράφουν τα δεδομένα και τις πληροφορίες.
- (5) Μεταβλητές του αλγορίθμου μας είναι η *timi* και τα *resta*.

Παράδειγμα 2

Ο μέσος όρος στο μάθημα των Νέων Ελληνικών υπολογίζεται ως εξής:

Μέσος όρος = (Βαθμός 1^{ου} τετραμήνου + Βαθμός 2^{ου} τετραμήνου) * 0.35 + Βαθμός εξετάσεων * 0.3

Η καθηγήτρια σας ενημερώνει για τους βαθμούς σας. Ποιος θα είναι ο μέσος όρος σας;

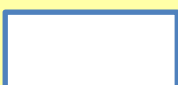


Για να σχεδιάσουμε το λογικό διάγραμμα βρίσκουμε τα δεδομένα και τα ζητούμενα. Στο πρόβλημά μας δεδομένα είναι οι βαθμοί των τετραμήνων και ο βαθμός των εξετάσεων. Αυτά είναι και τα στοιχεία που θα **διαβάσει** το πρόγραμμά μας. Ζητούμενο είναι ο μέσος όρος. Για να υπολογίσουμε τον μέσο όρο πρέπει να χρησιμοποιήσουμε τον μαθηματικό τύπο που μας έχει δοθεί στην εκφώνηση. Ο μέσος όρος είναι η πληροφορία που θα **τυπωθεί**.

Μεταβλητές του πιο πάνω αλγορίθμου είναι οι 3 βαθμοί (B1, B2, E) και ο μέσος όρος (MO).

Βασικές Έννοιες**Λογικό
Διάγραμμα:**

Είναι ένας σχηματικός τρόπος για την αναπαράσταση ενός αλγορίθμου. Στα λογικά διαγράμματα χρησιμοποιούμε τα παρακάτω σχήματα:

Σύμβολο	Χρήση
	Αρχή / Τέλος
	Είσοδος δεδομένων / Έξοδος πληροφοριών
	Πράξη / Επεξεργασία
	Ερώτηση/Απόφαση
	Πορεία εκτέλεσης

Δεδομένα:

Τα στοιχεία σε ένα πρόβλημα που μας είναι γνωστά και μπορούν να μας βοηθήσουν στη λύση του προβλήματος.

Επεξεργασία:

Η διαδικασία μέσω της οποίας βρίσκουμε τις πληροφορίες.

Πληροφορίες:

Τα ζητούμενα του προβλήματος. Προκύπτουν μετά την επεξεργασία των δεδομένων.

Μεταβλητή:

Μεταβλητή σε έναν αλγόριθμο ονομάζουμε ένα στοιχείο (δεδομένο/ πληροφορία) που δεν έχει σταθερή τιμή. Όταν αναφερόμαστε στον όρο μεταβλητή για ένα πρόγραμμα, εννοούμε τη θέση μνήμης που δεσμεύεται στην κύρια μνήμη του υπολογιστή (RAM) και χρησιμοποιείται για να αποθηκεύει προσωρινά δεδομένα και πληροφορίες του προγράμματός μας.

Σταθερά:

Σταθερά σε έναν αλγόριθμο ονομάζουμε ένα στοιχείο το οποίο δεν αλλάζει (μένει σταθερό), ανεξάρτητα από τα δεδομένα που δίνονται στο πρόβλημα (π.χ. ο αριθμός π).

Γ7.3 Προβλήματα Ακολουθιακής Δομής

Τι θα μάθουμε σήμερα:

- Να προτείνουμε τη λύση ενός προβλήματος ακολουθιακής δομής χρησιμοποιώντας λογικό διάγραμμα
- Να χρησιμοποιούμε τους μαθηματικούς τελεστές $+$, $-$, $*$, $/$, div και mod
- Να χρησιμοποιούμε την προκαταρκτική εκτέλεση για να ελέγχουμε την ορθότητα του αλγορίθμου μας.

1. Ακολουθιακή Δομή

Η ακολουθιακή δομή χρησιμοποιείται για την αντιμετώπιση απλών προβλημάτων, όπου είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών/ εντολών. Η πρώτη ενέργεια που πρέπει να εκτελέσει ένας αλγόριθμος σε αυτά τα προβλήματα είναι η είσοδος δεδομένων. Ακολουθεί η επεξεργασία των δεδομένων και τέλος γίνεται η παρουσίαση των πληροφοριών.

Ένα παράδειγμα ακολουθιακής δομής από την καθημερινή ζωή είναι η εκτέλεση μιας συνταγής για να φτιάξουμε ένα φαγητό. Παίρνουμε τα υλικά (δεδομένα), τα μαγειρεύουμε (επεξεργασία) και έχουμε έτοιμο το φαγητό (αποτέλεσμα/ πληροφορία).

2. Προκαταρκτική Εκτέλεση

Προκαταρκτική εκτέλεση είναι η διαδικασία στην οποία εκτελούμε τις εντολές του αλγορίθμου, όπως θα τις εκτελούσε ο ηλεκτρονικός υπολογιστής. Σε κάθε βήμα σημειώνουμε τις τιμές των μεταβλητών ή των αποφάσεων που αλλάζουν. Χρησιμοποιούμε συνήθως ένα πίνακα στον οποίο καταχωρούμε:

- (1) Τις αρχικές τιμές των μεταβλητών καθώς και τις τιμές τους μετά από κάθε αλλαγή.
- (2) Τις αποφάσεις και τις τιμές τους όταν αλλάζουν.
- (3) Την παρουσίαση των μηνυμάτων και των αποτελεσμάτων του αλγορίθμου. Δηλαδή τι θα δούμε στην οθόνη του υπολογιστή, τις πληροφορίες που παράγονται στην έξοδο.

Παράδειγμα

Θα γράψουμε την προκαταρκτική εκτέλεση του λογικού διαγράμματος που δόθηκε στο προηγούμενο κεφάλαιο και υπολόγιζε τον μέσο όρο στο μάθημα των Νέων Ελληνικών. Όπως αναφέραμε ήδη οι μεταβλητές μας είναι οι βαθμοί των 2 τετραμήνων, των εξετάσεων και ο μέσος όρος. Για να ελέγξουμε τον αλγόριθμό μας θέτουμε σαν αρχικές τιμές $B1=20$, $B2=19$ και $E=18$. Δημιουργούμε τον παρακάτω πίνακα και τον συμπληρώνουμε εκτελώντας τον αλγόριθμο μας βήμα προς βήμα.

Μεταβλητές				Παρουσίαση
B1	B2	E	MO	19.05
20	19	18		
			19.05	

3. Μαθηματικές εκφράσεις

Οι μαθηματικές πράξεις στην πληροφορική συμβολίζονται με συγκεκριμένα σύμβολα.

Μαθηματική πράξη	Μαθηματικός Συμβολισμός	Συμβολισμός στους αλγορίθμους	Παράδειγμα	Αποτέλεσμα
Πρόσθεση	$A + B$	$A + B$	$5+2$	7
Αφαίρεση	$A - B$	$A - B$	$7 - 3$	4
Πολλαπλασιασμός	$A \times B$	$A * B$	$5*3$	15
Διαίρεση	$A \div B$	A / B	$5/2$	2,5
Ακέραια Διαίρεση	$A \div B$	$A \text{ div } B$	$9 \text{ div } 3$	3
			$5 \text{ div } 2$	2
Υπόλοιπο Διάρθρωσης		$A \text{ mod } B$	$9 \text{ mod } 3$	0
			$5 \text{ mod } 2$	1

Κάποιες πράξεις έχουν μεγαλύτερη προτεραιότητα από άλλες. Αυτό σημαίνει ότι όταν έχουμε έναν υπολογισμό με πολλές πράξεις δεν ξεκινούμε από την πρώτη αλλά πρέπει να δούμε ποια έχει προτεραιότητα. Η προτεραιότητα πράξεων είναι η ίδια με τα μαθηματικά και είναι η ακόλουθη:

- (1) Παρενθέσεις
- (2) Πολλαπλασιασμός, Διαίρεση, Ακέραια Διαίρεση, Υπόλοιπο Διάρθρωσης
- (3) Πρόσθεση, Αφαίρεση.

Σε περίπτωση που έχουμε πράξεις με την ίδια προτεραιότητα τότε εκτελείται αυτή που βρίσκεται πιο αριστερά.

Παραδείγματα

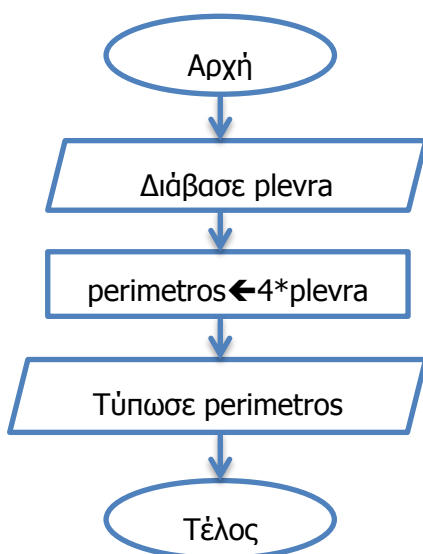
- (α) $7+12*2-1 = 7+24-1 = 30$
- (β) $6*4/2-1 = 24/2-1 = 12-1 = 11$
- (γ) $4^2*2 = 16*2 = 32$
- (δ) $19 \text{ div } 3 \text{ mod } 4 = 6 \text{ mod } 4 = 2$
- (ε) $(7+3)*2 = 10*2 = 20$

4. Προβλήματα με μία (1) είσοδο και έξοδο

Πρόκειται για τα απλούστερα προβλήματα που θα συναντήσετε. Ακολουθούν δύο παραδείγματα. Για καθένα από τα παραδείγματα θα σχεδιάζουμε το λογικό διάγραμμα και θα γράφουμε την προκαταρκτική εκτέλεση.

Παράδειγμα 1

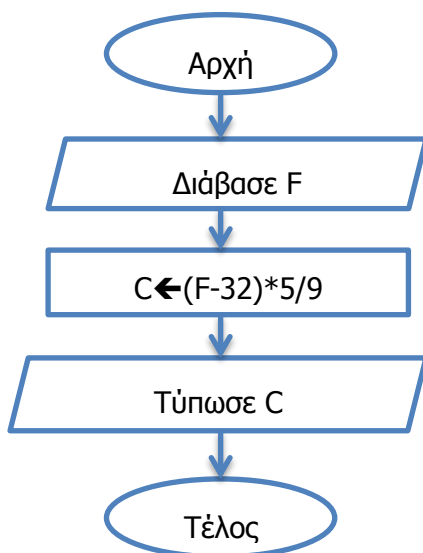
Να παρουσιάσετε τον αλγόριθμο που υπολογίζει την περίμετρο ενός τετραγώνου και να γράψετε την προκαταρκτική εκτέλεση για αρχική τιμή της πλευράς = 4



Μεταβλητές		Παρουσίαση
plevra	perimetros	16
4		
	16	

Παράδειγμα 2

Στην Κύπρο η μονάδα μέτρησης της θερμοκρασίας που χρησιμοποιούμε είναι οι βαθμοί Κελσίου (°C). Υπάρχουν όμως άλλες χώρες, όπως οι ΗΠΑ, που χρησιμοποιούν τους βαθμούς Φάρεναϊτ (°F). Ο τρόπος μετατροπής των βαθμών Φάρεναϊτ σε βαθμούς Κελσίου δίνεται από τον τύπο $C = (F - 32) * 5/9$. Να σχεδιάσετε το λογικό διάγραμμα και να γράψετε την προκαταρκτική εκτέλεση για αρχική τιμή $F = 98.6$.



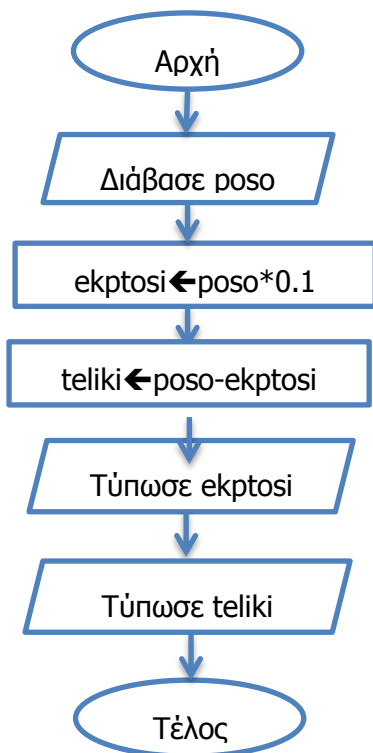
Μεταβλητές		Παρουσίαση
F	C	37
98.6		
	37	

5. Προβλήματα με μία (1) είσοδο και περισσότερες εξόδους

Η μεθοδολογία δεν αλλάζει σε σχέση με την προηγούμενη κατηγορία. Οι αλλαγές που θα παρατηρήσουμε είναι ότι θα έχουμε περισσότερες μεταβλητές και περισσότερα στοιχεία στην παρουσίαση.

Παράδειγμα 3

Ένα κατάστημα ρούχων κάνει έκπτωση 10% σε όλους τους πελάτες του. Να σχεδιάσετε το λογικό διάγραμμα που δέχεται το αρχικό ποσό και τυπώνει την έκπτωση και την τελική τιμή. Τέλος να γράψετε την προκαταρκτική εκτέλεση για αρχική τιμή $poso=200$.



Μεταβλητές		
poso	ekptosi	teliki
200		
	20	
		180

Παρουσίαση
20
180

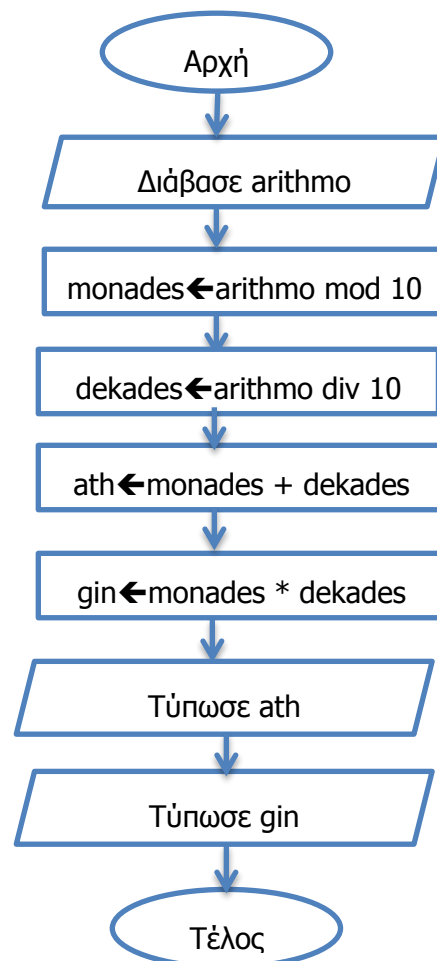
Παράδειγμα 4

Να σχεδιάσετε το λογικό διάγραμμα που δέχεται έναν διψήφιο ακέραιο αριθμό και τυπώνει το άθροισμα και το γινόμενο των ψηφίων του. Για παράδειγμα αν δοθεί ο αριθμός 53 θα τυπωθεί 8 ($5+3$) και 15 ($5*3$).

Η δυσκολία αυτής της άσκησης είναι να ξεχωρίσουμε τα ψηφία του διψήφιου αριθμού. Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε τους τελεστές div (για να απομονώσουμε τις δεκάδες) και mod (για να απομονώσουμε τις μονάδες). Π.χ. $53 \text{ div } 10 = 5$, $53 \text{ mod } 10 = 3$.

Μεταβλητές				
arithmo	monades	dekades	ath	gin
53				
	3			
		5		
			8	
				15

Παρουσίαση
8
15



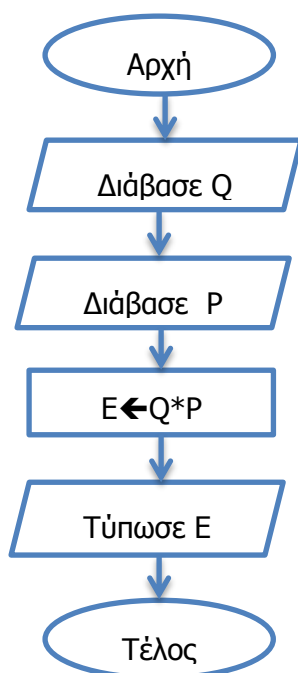
6. Προβλήματα με περισσότερες εισόδους και 1 έξοδο

Σε αυτή τη κατηγορία προβλημάτων θα έχουμε περισσότερες από μία εισόδους (περισσότερες μεταβλητές στο Διάβασε) και μόνο μία έξοδο.

Παράδειγμα 5

Ένας έμπορος πώλησε κάποια ποσότητα (Q) από πορτοκάλια προς κάποια τιμή (P).

Πόση θα είναι η εισπραξη (E); Να σχεδιάσετε το λογικό διάγραμμα και να γράψετε την προκαταρκτική εκτέλεση για Q=100, P= 1.50

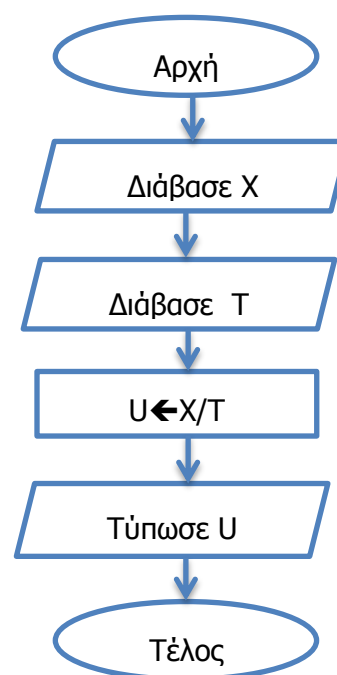


Μεταβλητές			Παρουσίαση
Q	P	E	150
100			
	1.50		
		150	

Παράδειγμα 6

Η ταχύτητα ενός σώματος δίνεται από τον τύπο $U = X/T$ (ταχύτητα = μετατόπιση/ χρόνος). Για παράδειγμα αν ένα σώμα διένυσε μια απόσταση 100 μέτρα μέσα σε 2 δευτερόλεπτα τότε σημαίνει ότι η ταχύτητα του ήταν 50 m/s. Να σχεδιάσετε το λογικό διάγραμμα για το πιο πάνω πρόβλημα.

Μεταβλητές			Παρουσίαση
X	T	U	50
100			
	2		
		50	



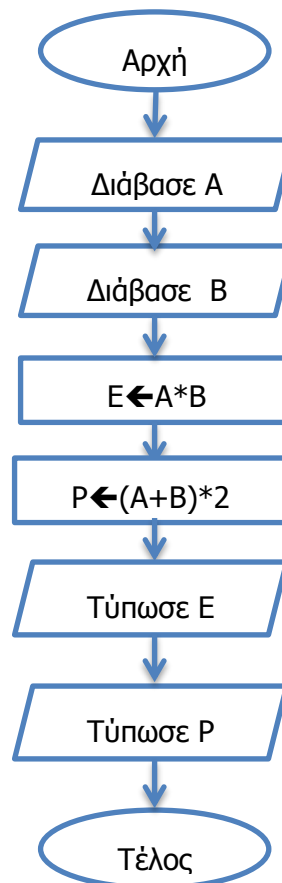
7. Προβλήματα με περισσότερες εισόδους και εξόδους

Σε αυτή την κατηγορία προβλημάτων ο αλγόριθμος μας θα δέχεται πολλαπλές εισόδους και θα παράγει πολλαπλά αποτελέσματα.

Παράδειγμα 7

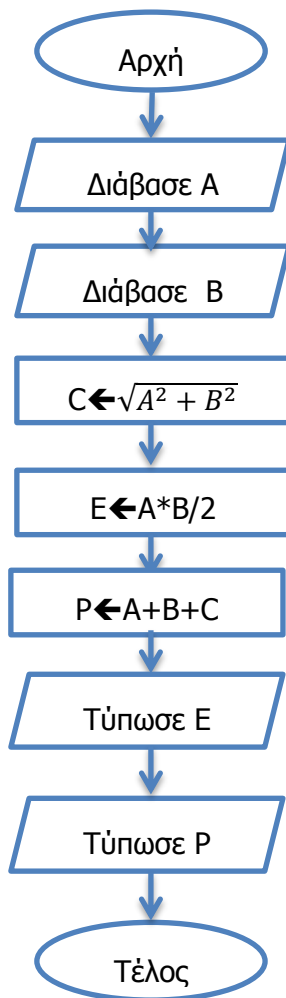
Να σχεδιάσετε το λογικό διάγραμμα που δέχεται το μήκος (A) και το πλάτος (B) ενός ορθογωνίου και υπολογίζει και τυπώνει το εμβαδόν (E) και την περίμετρο (P) του. Το εμβαδόν ενός ορθογωνίου δίνεται από τον τύπο $E=A*B$, ενώ η περίμετρος από τον τύπο $P=(A+B)*2$. Να γράψετε την προκαταρκτική εκτέλεση για αρχικές τιμές $A=10$ και $B=5$.

Μεταβλητές				Παρουσίαση
A	B	E	P	50
10				30
	5			
		50		
			30	

**Παράδειγμα 8**

Να σχεδιάσετε το λογικό διάγραμμα που υπολογίζει το εμβαδόν και την περίμετρο ενός ορθογωνίου τριγώνου όταν είναι γνωστές οι 2 κάθετες. Για να υπολογίσετε την περίμετρο πρέπει να υπολογίσετε την υποτείνουσα χρησιμοποιώντας το πυθαγόρειο θεώρημα.

(Σημείωση: $E=A*B/2$, $P=A+B+C$, $C=\sqrt{A^2 + B^2}$)



Μεταβλητές					Παρουσίαση
A	B	C	E	P	6 12
3					
	4				
		5			
			6		
				12	

Βασικές Έννοιες**Ακολουθιακή
Δομή:**

Κατηγορία προβλημάτων τα οποία δεν περιέχουν αποφάσεις. Σε αυτού του είδους τα προβλήματα η λύση ακολουθεί το μοντέλο **Είσοδος Δεδομένων → Επεξεργασία → Παρουσίαση Πληροφοριών**

**Προκαταρκτική
Εκτέλεση:**

Είναι η εκτέλεση του αλγορίθμου στο χαρτί, όπως θα τον εκτελούσε ο υπολογιστής.

**Τελεστές –
Προτεραιότητα
Πράξεων:**

Για την αναπαράσταση μαθηματικών εκφράσεων χρησιμοποιούμε τους ακόλουθους τελεστές

Μαθηματική πράξη	Μαθηματικός Συμβολισμός	Συμβολισμός στους αλγορίθμους
Πρόσθεση	$A + B$	$A + B$
Αφαίρεση	$A - B$	$A - B$
Πολλαπλασιασμός	$A \times B$	$A * B$
Διαίρεση	$A \div B$	A / B
Ακέραια Διαίρεση	$A \div B$	$A \text{ div } B$
Υπόλοιπο Διάρεσης		$A \text{ mod } B$

Η προτεραιότητα των πιο πάνω τελεστών είναι:

- (1) Παρενθέσεις
- (2) Πολλαπλασιασμός, Διαίρεση, Ακέραια Διαίρεση, Υπόλοιπο Διάρεσης
- (3) Πρόσθεση, Αφαίρεση.

Γ7.4 Προβλήματα Δομής Διακλάδωσης

Τι θα μάθουμε σήμερα:

- ◆ Να διακρίνουμε τις διαφορές μεταξύ ακολουθιακής δομής και δομής διακλάδωσης
- ◆ Να μελετούμε προβλήματα και να εξηγούμε αν απαιτείται ή όχι δομή διακλάδωσης
- ◆ Να δίνουμε παραδείγματα από την καθημερινότητα όπου γίνεται χρήση κάποιου είδους δομής διακλάδωσης
- ◆ Να σχεδιάζουμε λογικά διαγράμματα με μία συνθήκη, τόσο χωρίς το else όσο και με εντολές στο else
- ◆ Να σχεδιάζουμε λογικά διαγράμματα με σύνθετες συνθήκες (AND - OR)
- ◆ Να σχεδιάζουμε λογικά διαγράμματα με φωλιασμένες (nested) δομές διακλάδωσης
- ◆ Να χρησιμοποιούμε τη μέθοδο της προκαταρκτικής εκτέλεσης για την εξαγωγή των αποτελεσμάτων των λογικών διαγραμμάτων.

1. Δομή διακλάδωσης ή Δομή επιλογής

1.1 Δομή διακλάδωσης και καθημερινότητα

Πολλές φορές τυγχάνει να εμπλακούμε σε απλά προβλήματα τα οποία εμπεριέχουν «διλήμματα» της καθημερινής μας ζωής. Επίσης, όλοι έτυχε να πούμε κάποιες φράσεις όπως π.χ. «αν διαβάσω, θα γράψω καλά στο διαγώνισμα» ή «αν έχω περισσότερες από 50 αδικαιολόγητες απουσίες, θα μείνω στάσιμος».

Σε αυτές τις φράσεις, όπως και σε πολλές άλλες τις οποίες χρησιμοποιούμε στον καθημερινό μας λόγο, μπορούμε να αναγνωρίσουμε το γεγονός ότι ανάλογα με το τι θα γίνει ή τι θα επιλέξουμε, θα υπάρξει διαφορετικό αποτέλεσμα. Αυτή η έννοια στην οποία γίνεται κάποια σύγκριση (π.χ. να πάρουμε δύο δεδομένα και να αποφασίσουμε εάν το πρώτο είναι ίσο, μεγαλύτερο ή μικρότερο από το δεύτερο) ή κάποιος λογικός συλλογισμός (π.χ. συλλογισμός που απαντιέται με «Αληθής» ή «Ψευδής»), ονομάζεται συνθήκη.

Κάποια απτά παραδείγματα από την καθημερινή μας ζωή όπου υπάρχει κάποια συνθήκη, φαίνονται στον πιο κάτω πίνακα:

Φράση	Συνθήκη	Τιμή (Αληθής/Ψευδής)	Αποτέλεσμα
Αν βρέξει, θα πάρω ομπρέλα.	Αν βρέξει	Αληθής Ψευδής	Θα πάρω ομπρέλα Δε θα πάρω ομπρέλα
Αν αγοράσω προϊόντα αξίας πέραν των 100 ευρώ θα έχω έκπτωση 15% διαφορετικά θα έχω έκπτωση 7%	Αν το ποσό είναι μεγαλύτερο από 100 ευρώ	Αληθής Ψευδής	Θα έχω έκπτωση 15% Θα έχω έκπτωση 7%

1.2 Διαφορές ανάμεσα στη δομή ακολουθίας και στη δομή διακλάδωσης

Όλα τα προβλήματα δεν λύνονται μόνο με τη διαδοχική εκτέλεση κάποιων συγκεκριμένων εντολών. Πολλές φορές πρέπει να ελεγχθεί μια κατάσταση και ανάλογα να αποφασίσουμε αν θα συνεχίσουμε προς τη μία ή την άλλη κατεύθυνση τις ενέργειές μας.

Η ακολουθιακή δομή εντολών ή δομή ακολουθίας χρησιμοποιείται για την αντιμετώπιση απλών προβλημάτων, όπου είναι δεδομένη η σειρά εκτέλεσης ενός συνόλου ενεργειών. Ένα

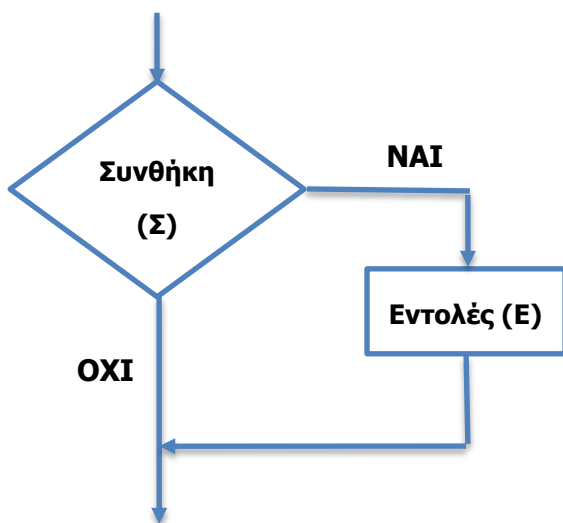
απλό παράδειγμα από την καθημερινή μας ζωή είναι η ακολουθία οδηγιών μίας συνταγής μαγειρικής με στόχο την κατασκευή ενός φαγητού. Τα βήματα και οι ποσότητες που πρέπει να ακολουθηθούν είναι συγκεκριμένα και οι οδηγίες απόλυτα καθορισμένες και σαφείς.

Στη πραγματικότητα όμως, πολύ λίγα προβλήματα μπορούν να επιλυθούν με τον τρόπο της ακολουθιακής δομής ενεργειών. Συνήθως τα προβλήματα έχουν κάποιες ιδιαιτερότητες και δεν ισχύουν τα ίδια βήματα για κάθε περίπτωση. Η πλέον συνηθισμένη περίπτωση είναι να λαμβάνονται κάποιες αποφάσεις με βάση κάποια δεδομένα κριτήρια, που μπορεί να είναι διαφορετικά για κάθε διαφορετικό στιγμιότυπο ενός προβλήματος. Οι καθημερινές απλές μας ενέργειες περιέχουν αυτή τη διαδικασία επιλογής με βάση κάποια κατάσταση. Για παράδειγμα, το πρόβλημα της προετοιμασίας μας για έξοδο σχετίζεται με τις καιρικές συνθήκες. Έτσι λέμε ότι «αν βρέχει, θα πάρω ομπρέλα, αλλιώς θα πάρω καπέλο». Η συνθήκη εδώ είναι το «αν βρέχει», ενώ η απόφαση είναι είτε να «πάρω ομπρέλα» είτε να «πάρω καπέλο» με βάση την «τιμή» της συνθήκης.

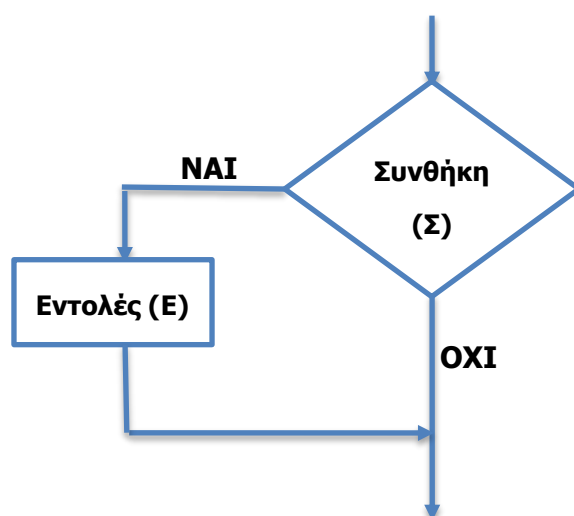
1.3 Η δομή διακλάδωσης:

Η δομή διακλάδωσης ή επιλογής χρησιμοποιείται, όταν κατά την εκτέλεση του αλγορίθμου θα πρέπει να ελέγξουμε μια κατάσταση και ανάλογα να εκτελέσουμε ή όχι κάποιες εντολές.

Στο λογικό διάγραμμα, η εντολή επιλογής εμφανίζεται ως μία διακλάδωση στην πορεία της εκτέλεσης των εντολών. Μέσα σε έναν ρόμβο διατυπώνεται μια συνθήκη, δηλαδή ένα ερώτημα, που μπορεί να απαντηθεί με ΝΑΙ (True) ή ΟΧΙ (False), και ανάλογα με την απάντηση ακολουθείται η κατάλληλη διαδρομή, όπως φαίνεται πιο κάτω. (Εικόνα 27 και Εικόνα 28)

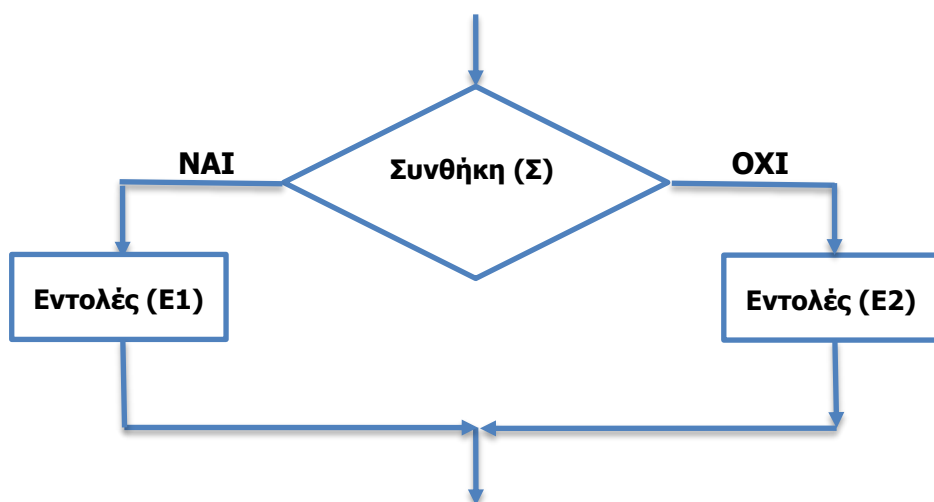


Εικόνα 27



Εικόνα 28

Υπάρχει και μια δεύτερη μορφή της δομής επιλογής, όπου ελέγχουμε μια κατάσταση και, αν ισχύει μια συνθήκη, εκτελούμε ένα σύνολο εντολών, ενώ, αν δεν ισχύει η συνθήκη, εκτελούμε ένα διαφορετικό σύνολο εντολών. Σε αυτή την περίπτωση το λογικό διάγραμμα έχει τη μορφή του παρακάτω σχήματος (Εικόνα 29).



Εικόνα 29

Γενικά η διαδικασία της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (Αληθής ή Ψευδής) και ακολουθεί η απόφαση εκτέλεσης κάποιας ενέργειας με βάση την τιμή της λογικής αυτής συνθήκης.

Εάν η συνθήκη είναι αληθής εκτελείται το σκέλος του ΝΑΙ, ενώ στην αντίθετη περίπτωση που η συνθήκη είναι ψευδής, εκτελείται το σκέλος του ΟΧΙ. Σε πολλές περιπτώσεις αντί του ΝΑΙ και του ΟΧΙ μπορεί να δείτε τους όρους ΟΡΘΟ / ΛΑΘΟΣ, ΑΛΗΘΗΣ/ΨΕΥΔΗΣ ή τους Αγγλικούς Όρους TRUE / FALSE. Για παράδειγμα θα μπορούσαμε να ρωτήσουμε «Είσαι μεγαλύτερος των 18 χρόνων»; Η απάντηση σε αυτή την ερώτηση είναι ΝΑΙ ή ΟΧΙ. Αν η απάντηση είναι ΟΧΙ, η σημασία της είναι ότι το άτομο είναι Μικρότερο ή Ίσο με 18 χρόνια. Στις συνθήκες γενικά χρησιμοποιούμε τα συγκριτικά σύμβολα, όπως φαίνεται πιο κάτω.

Πίνακας Συγκριτικών Συμβόλων

Ισότητα	=
Μεγαλύτερο	>
Μεγαλύτερο ή Ίσο	>=
Μικρότερο	<
Μικρότερο ή Ίσο	=<
Άνισο	<>

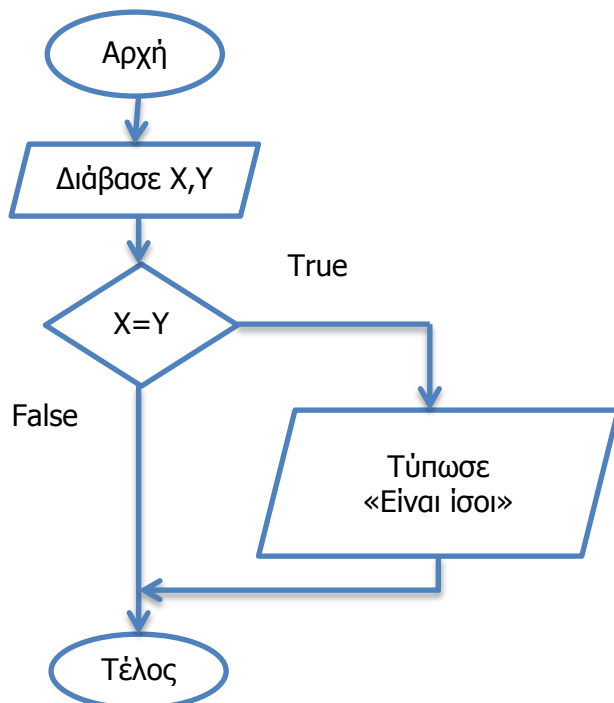
Παραδείγματα Συνθήκης χρησιμοποιώντας συγκριτικά σύμβολα

Συνθήκη	Αποτέλεσμα (Αληθής/Ψευδής)
3>2	Αληθής
10<=10	Αληθής
8<4	Ψευδής
5<>5	Ψευδής

1.4 Σχεδιασμός λογικών διαγραμμάτων με μια απλή συνθήκη (χωρίς το αλλιώς - else)

Παράδειγμα 1

Να σχεδιάσετε λογικό διάγραμμα το οποίο να διαβάζει 2 αριθμούς **X**, **Y**. Αν οι 2 αριθμοί είναι ίσοι, τότε να τυπώνει το μήνυμα «**Είναι ίσοι**». Να γράψετε την προκαταρκτική εκτέλεση για αρχικές τιμές α) $X=5, Y=5$ και β) $X=5, Y=6$



Πίνακας Προκαταρκτικής εκτέλεσης α

Μεταβλητές		Απόφαση	
X	Y	X=Y	True/False
5			
	5		
		5=5	True

Παρουσίαση
Είναι ίσοι

Πίνακας Προκαταρκτικής εκτέλεσης β

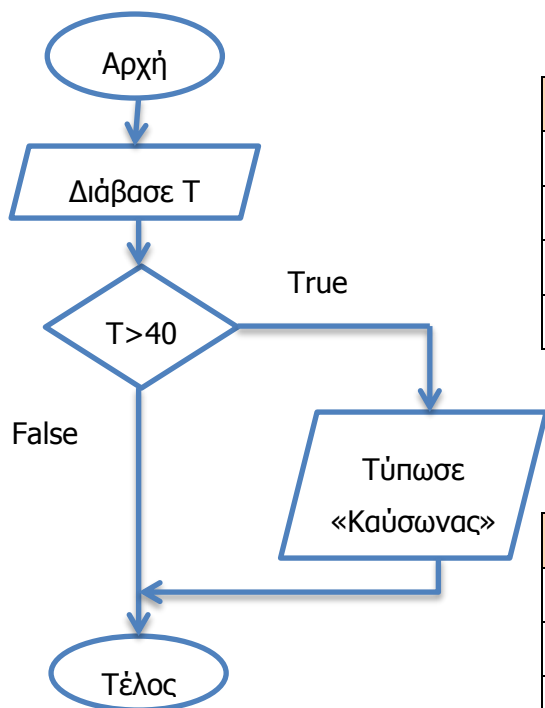
Μεταβλητές		Απόφαση	
X	Y	X=Y	True/False
5			
	6		
		5=6	False

Παρουσίαση

Παρατηρούμε ότι στην προκαταρκτική εκτέλεση έχουμε μια επιπλέον περιοχή για να γράφουμε τις αποφάσεις (δηλαδή τα αποτελέσματα των συνθηκών). Για κάθε απόφαση που υπάρχει στον αλγόριθμό μας τοποθετούμε δυο στήλες, μια για τη συνθήκη και μια για το αποτέλεσμα της συνθήκης. Προσέξτε, επίσης, ότι σε αυτή την υποκατηγορία προβλημάτων δομής διακλάδωσης (χωρίς εντολές στο Όχι/False), υπάρχει περίπτωση το πρόγραμμα που θα υλοποιήσουμε για κάποιες περιπτώσεις να μην εμφανίζει κανένα αποτέλεσμα.

Παράδειγμα 2

Να σχεδιάσετε λογικό διάγραμμα το οποίο να διαβάζει τη θερμοκρασία μιας ημέρας (**T**). Αν η θερμοκρασία είναι μεγαλύτερη από 40, τότε να τυπώνει το μήνυμα «**Καύσωνας**». Να γράψετε την προκαταρκτική εκτέλεση για αρχικές τιμές α) $T=45$ και β) $T=40$



Πίνακας Προκαταρκτικής εκτέλεσης α

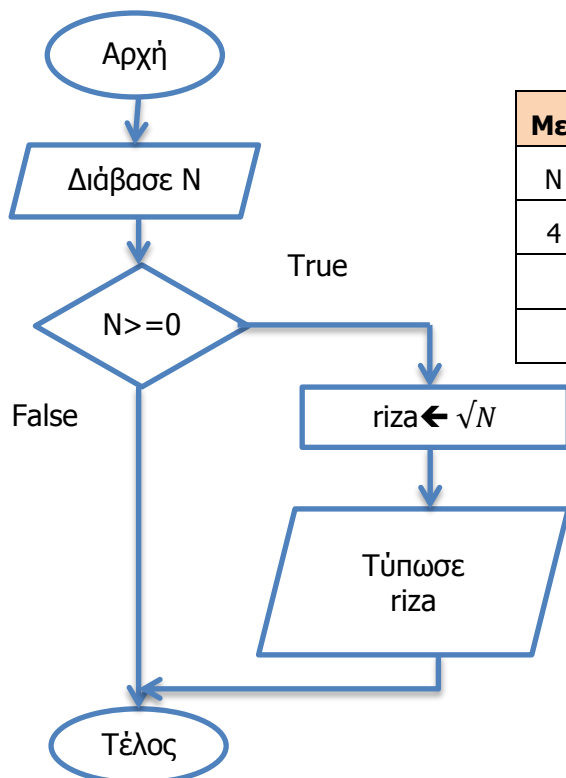
Μεταβλητές		Απόφαση		Παρουσίαση
T		$T > 40$	True/False	Καύσωνας
45				
		$45 > 40$	True	

Πίνακας Προκαταρκτικής εκτέλεσης β

Μεταβλητές		Απόφαση		Παρουσίαση
T		$T > 40$	True/False	
40				
		$40 > 40$	False	

Παράδειγμα 3

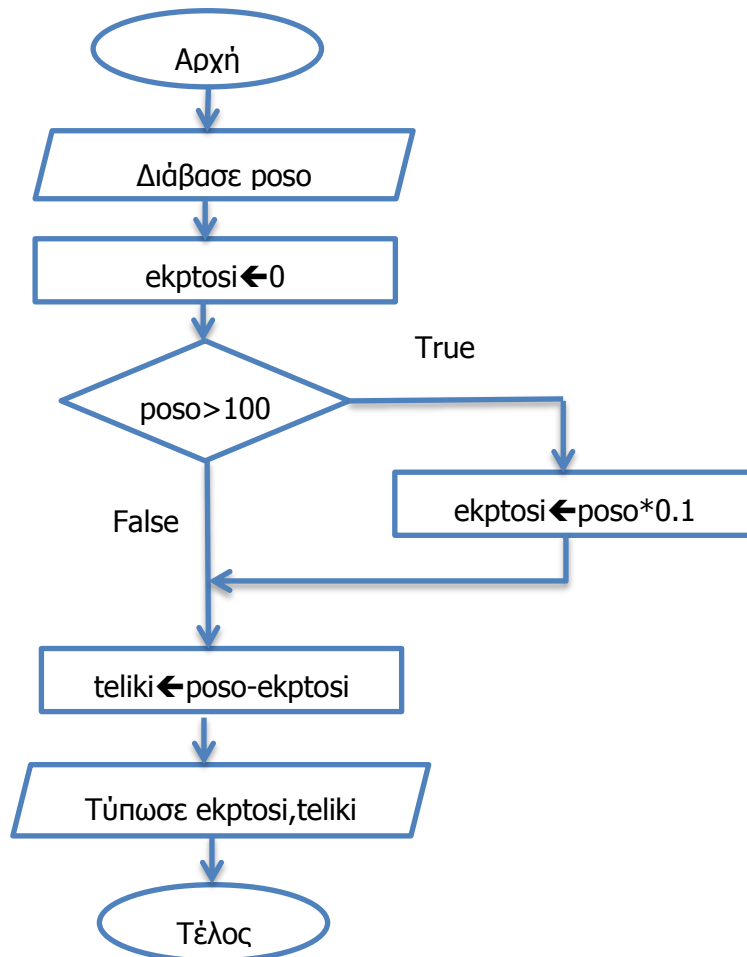
Να σχεδιάσετε λογικό διάγραμμα το οποίο να διαβάζει έναν αριθμό (N). Αν ο αριθμός είναι θετικός (≥ 0) τότε υπολογίζει και τυπώνει την τετραγωνική ρίζα του αριθμού. Να γράψετε την προκαταρκτική εκτέλεση για $N=4$.



Μεταβλητές		Απόφαση		Παρουσίαση
N	riza	$N \geq 0$	True/False	2
4				
		$4 \geq 0$	True	
	2			

Παράδειγμα 4

Σε παράδειγμα που είχαμε δώσει στην ακολουθιακή δομή, δημιουργήσαμε τον αλγόριθμο για ένα κατάστημα που έκανε έκπτωση 10% σε όλους τους πελάτες του. Να ξανασχεδιάσετε το λογικό διάγραμμα για το πιο πάνω πρόβλημα, ώστε το κατάστημα να κάνει έκπτωση μόνο σε όσους πελάτες κάνουν αγορές πέραν των 100 ευρώ.



Σε αντίθεση με τα προηγούμενα παραδείγματα εδώ βλέπουμε ότι θα έχουμε αποτέλεσμα στην παρουσίαση ανεξαρτήτως αν ισχύει ή όχι η συνθήκη. Δοκιμάστε να γράψετε την προκαταρκτική εκτέλεση για α) $poso=200$ και β) $poso=50$.

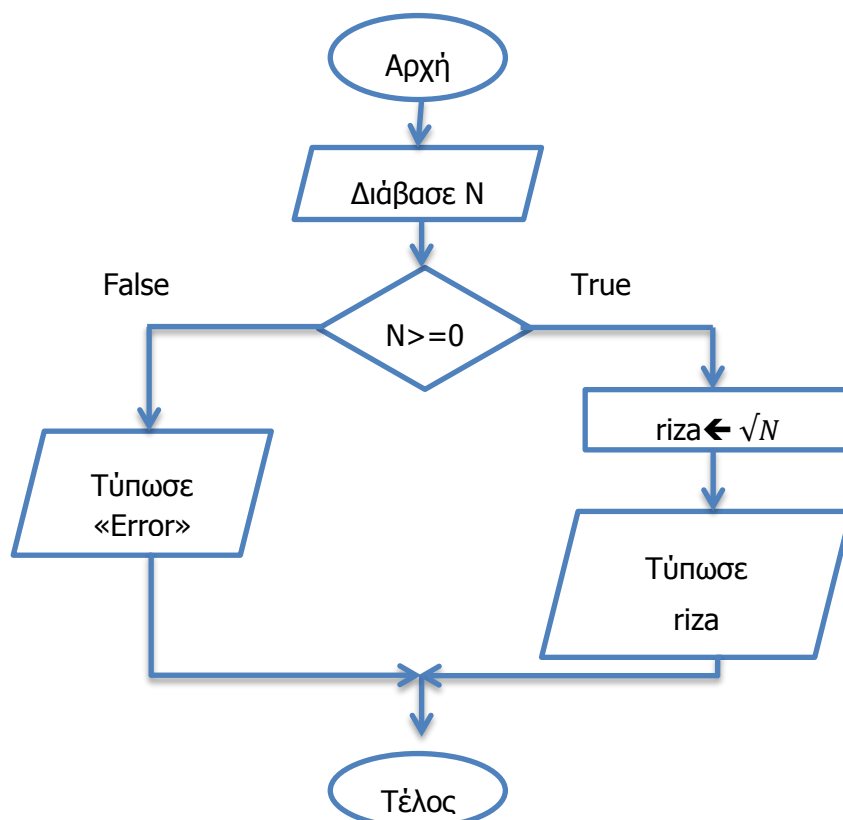
1.5 Σχεδιασμός λογικών διαγραμμάτων με μια απλή συνθήκη (με εντολές για το Όχι/False)

Η απουσία εντολών για την περίπτωση που η συνθήκη είναι ψευδής (false), δεν είναι κάτι συνηθισμένο, σε αλγορίθμους/ προγράμματα με μια απλή συνθήκη. Ο λόγος είναι ότι περιμένουμε πάντα από το πρόγραμμά μας να μας ενημερώσει για κάτι (να έχει κάποια έξοδο πληροφοριών). Στα πιο πάνω παραδείγματα θα θέλαμε να ξέρουμε ότι οι αριθμοί «δεν είναι ίσοι» (Παράδειγμα 1), «δεν είναι καύσωνας» (Παράδειγμα 2) και «δεν υπάρχει τετραγωνική ρίζα» (παράδειγμα 3).

Η αντιμετώπιση των προβλημάτων με εντολές για την περίπτωση που η συνθήκη είναι ψευδής δεν διαφέρει πολύ με τα προβλήματα που ήδη έχουμε δει. Πρέπει να εντοπίσουμε τα δεδομένα και τα ζητούμενα (πληροφορίες), να βρούμε τη συνθήκη ελέγχου και τις εντολές που θα εκτελούνται όταν θα ισχύει και όταν δεν θα ισχύει η συνθήκη.

Παράδειγμα 5

Να τροποποιήσετε το Παράδειγμα 3 του προηγούμενου υποκεφαλαίου, ώστε να τυπώνει μήνυμα «Error» στην περίπτωση που δεν υπάρχει τετραγωνική ρίζα.

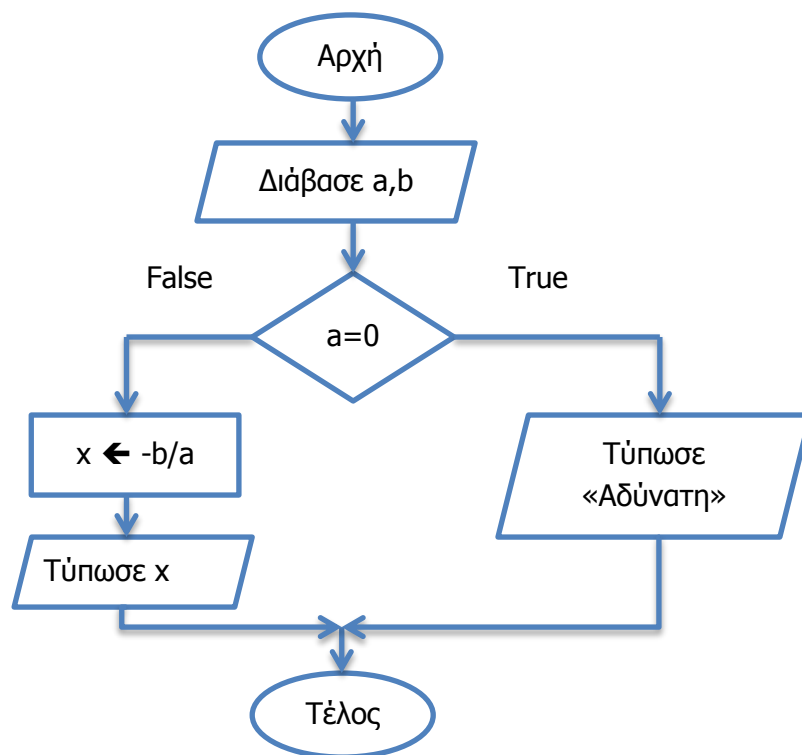


Ας δοκιμάσουμε να γράψουμε την προκαταρκτική εκτέλεση για αρχική τιμή $N=-4$. Παρατηρούμε ότι πλέον ο αλγόριθμός μας θα τυπώνει «Error» στην περίπτωση που δεν ισχύει η συνθήκη. Δοκιμάστε να υπολογίσετε την τετραγωνική ρίζα του 4 και του -4. Τι παρατηρείτε;

Μεταβλητές		Απόφαση		Παρουσίαση
N	riza	$N \geq 0$	True/False	Error
-4				
		$-4 \geq 0$	False	

Παράδειγμα 6

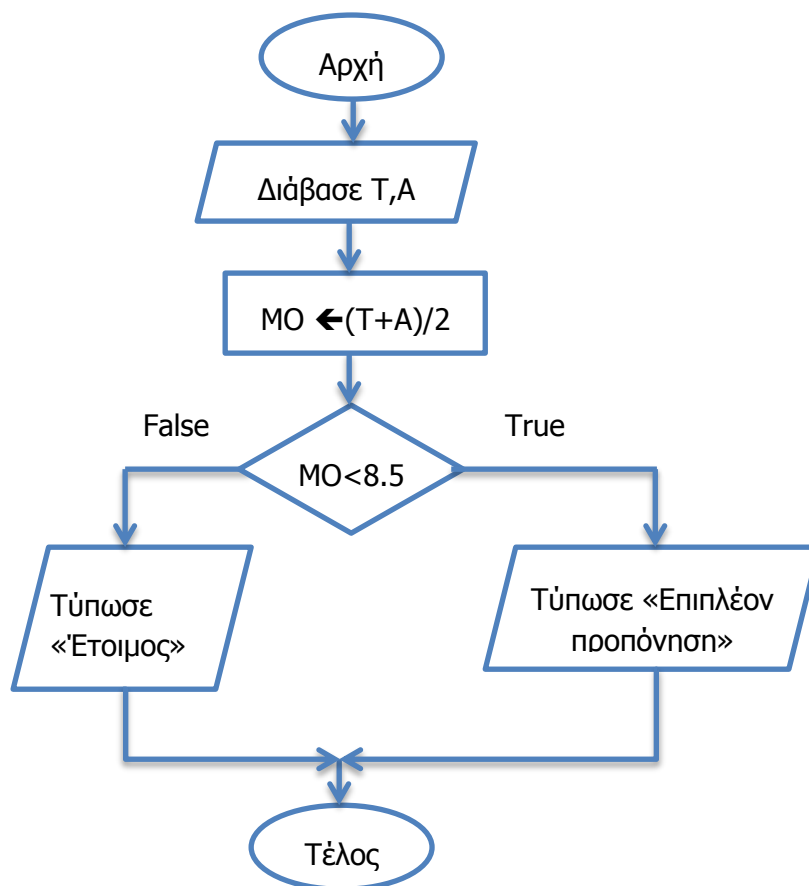
Ένας συμμαθητής σας δυσκολεύεται να βρίσκει τη λύση της εξίσωσης α' βαθμού $ax+b=0$. Να σχεδιάσετε το λογικό διάγραμμα για το πιο πάνω πρόβλημα που θα δέχεται το a και b και θα υπολογίζει το x . Σε περίπτωση που δοθεί $a=0$ να τυπώνει το μήνυμα «**Αδύνατη**». Τέλος να γράψετε την προκαταρκτική εκτέλεση για την εξίσωση $2x+4=0$. (Σημείωση: Η λύση της εξίσωσης είναι $x=-b/a$). Να γράψετε την προκαταρκτική εκτέλεση για αρχικές τιμές $a=2$ και $b=4$.



Μεταβλητές			Απόφαση		Παρουσίαση
a	B	x	a=0	True/False	-2
2					
	4				
			2=0	False	
		-2			

Παράδειγμα 7

Ο προπονητής μιας ποδοσφαιρικής ομάδας έχει παρατηρήσει ότι οι παίκτες της ομάδας του δεν είναι σε φόρμα. Αποφάσισε λοιπόν να τους εξετάσει τόσο σε θέματα ταχύτητας, όσο και σε θέματα αντοχής, υποβάλλοντάς τους σε μία άσκηση ταχύτητας (T) και σε μια άσκηση αντοχής (A). Σε καθεμιά από τις ασκήσεις ο ποδοσφαιριστής παίρνει μια βαθμολογία με άριστα το 10. Αν ο μέσος όρος των 2 ασκήσεων είναι μικρότερος του 8.5 τότε ο ποδοσφαιριστής πρέπει να περάσει από επιπλέον προπόνηση διαφορετικά θεωρείται ότι είναι έτοιμος. Να σχεδιάσετε το λογικό διάγραμμα για το πιο πάνω πρόβλημα.



2. Λογικοί τελεστές

Μέχρι τώρα οι λογικές συνθήκες που χρησιμοποιήσαμε ήταν απλές και δεν μπορούσαν να ελέγξουν περισσότερο από ένα πράγμα την ίδια στιγμή. Αυτό δεν είναι και πολύ βολικό στην πράξη και μπορούμε να φέρουμε πολλά παραδείγματα για αυτό. Αν ρωτήσουμε κάποιον τι θα γίνει αν πατήσω το κουμπί X παίζοντας το παιχνίδι FIFA στο Playstation τότε το πιο πιθανόν είναι να απαντήσει ότι ο παίκτης θα δώσει πάσα. Αυτό είναι λάθος, γιατί ο παίκτης θα δώσει πάσα εφόσον έχει στην κατοχή του μπάλα. Άρα για να δώσει πάσα το παιχνίδι πρέπει να ελέγξει δύο πράγματα ταυτόχρονα. Ότι έχει πατηθεί το κουμπί X και ότι ο παίκτης έχει στην κατοχή του μπάλα. Αυτή είναι μια σύνθετη συνθήκη. Άλλα παραδείγματα σύνθετων συνθηκών είναι:

- (1) Έλεγχος ότι ο βαθμός σε ένα μάθημα βρίσκεται μεταξύ του 1 και του 20
- (2) Έλεγχος για το ποιοι μήνες έχουν 30 μέρες
- (3) Έλεγχος για το πότε μένει στάσιμος ένας μαθητής

και πολλές άλλες

Για να δημιουργήσουμε σύνθετες συνθήκες χρειαζόμαστε τους λογικούς τελεστές. Οι λογικοί τελεστές δουλεύουν όπως και οι αριθμητικοί τελεστές (+, -, *, /), αλλά σε λογικές εκφράσεις. Μια λογική έκφραση έχει πάντα σαν αποτέλεσμα είτε την τιμή **ΑΛΗΘΗΣ (True)** είτε την τιμή **ΨΕΥΔΗΣ (False)**. Ήδη έχουμε χρησιμοποιήσει λογικές εκφράσεις στα προηγούμενα παραδείγματα της δομής διακλάδωσης (π.χ. $a=0$, $T>40$ κ.λπ.). Οι λογικοί τελεστές που

χρησιμοποιούμε στην πληροφορική είναι το **NOT**, το **AND** και το **OR**. Στο κεφάλαιο αυτό θα εξετάσουμε μόνο το AND και το OR.

2.1 Λογικός Τελεστής AND

Ο λογικός τελεστής AND εκτελεί τη λογική πράξη AND (ΚΑΙ) μεταξύ των 2 συνθηκών της. Στον πιο κάτω πίνακα αληθείας του λογικού τελεστή AND φαίνονται τα αποτελέσματα.

Συνθήκη 1	λογική πράξη	Συνθήκη 2	Αποτέλεσμα
Αληθής	AND	Αληθής	Αληθής
Αληθής	AND	Ψευδής	Ψευδής
Ψευδής	AND	Αληθής	Ψευδής
Ψευδής	AND	Ψευδής	Ψευδής

Παράδειγμα λογικού τελεστή AND από την καθημερινότητα



Αυτόματο πότισμα.

Για να λειτουργήσει σωστά το αυτόματο πότισμα που τοποθέτησα θα πρέπει και το **σημείο A** (βρύση) αλλά **και** το **σημείο B** (μηχανισμός), να είναι ανοικτά. Αν έστω και ένα από τα δυο σημεία είναι κλειστό τότε δεν θα μπορεί να περάσει το νερό.

1. εάν το σημείο A είναι κλειστό (**ψευδής**) και το σημείο B είναι κλειστό (**ψευδής**) τότε δεν θα μπορεί να περάσει το νερό, αποτέλεσμα **ψευδής**.
2. εάν το σημείο A είναι ανοικτό (**αληθής**) και το σημείο B είναι κλειστό (**ψευδής**) τότε δεν θα μπορεί να περάσει το νερό, αποτέλεσμα **ψευδής**.
3. εάν το σημείο A είναι κλειστό (**ψευδής**) και το σημείο B είναι ανοικτό (**αληθής**) τότε δεν θα μπορεί να περάσει το νερό, αποτέλεσμα **ψευδής**.
4. εάν το σημείο A είναι ανοικτό (**αληθής**) και το σημείο B είναι ανοικτό (**αληθής**) τότε θα μπορεί να περάσει το νερό, αποτέλεσμα **αληθής**.

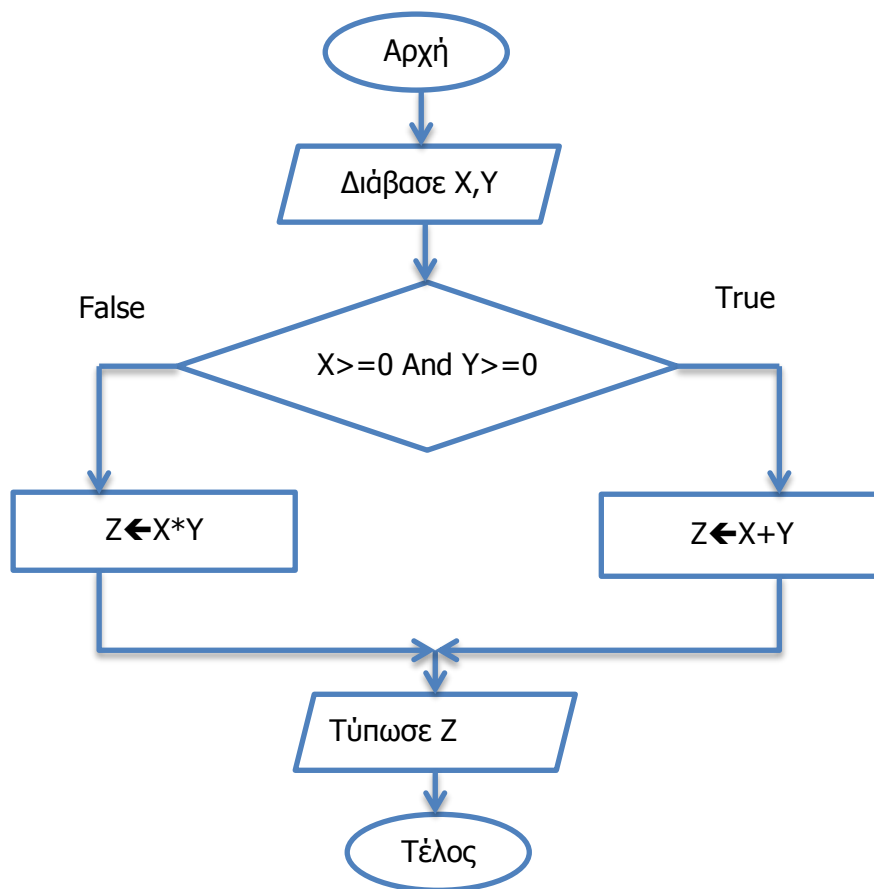
Παραδείγματα λογικών εκφράσεων με τη χρήση του τελεστή AND

Έστω ότι έχουμε τις μεταβλητές A=10, B=20, C=30

Λογική Έκφραση	Αποτέλεσμα
(A>B) And (A>C)	False
(A<B) And (A<C)	True
(B>A) And (B<C)	True
(A=B) And (C=30)	False
(B<100) And (C>100)	False

Παράδειγμα 8

Να σχεδιάσετε το λογικό διάγραμμα που δέχεται 2 αριθμούς (X,Y). Αν είναι και οι 2 αριθμοί θετικοί τότε να τυπώνεται το άθροισμά τους διαφορετικά να τυπώνεται το γινόμενο τους. Να γράψετε την προκαταρκτική εκτέλεση για αρχικές τιμές α) X=10, Y=5 και β) X=10, Y=-5.

**Πίνακας Προκαταρκτικής εκτέλεσης α**

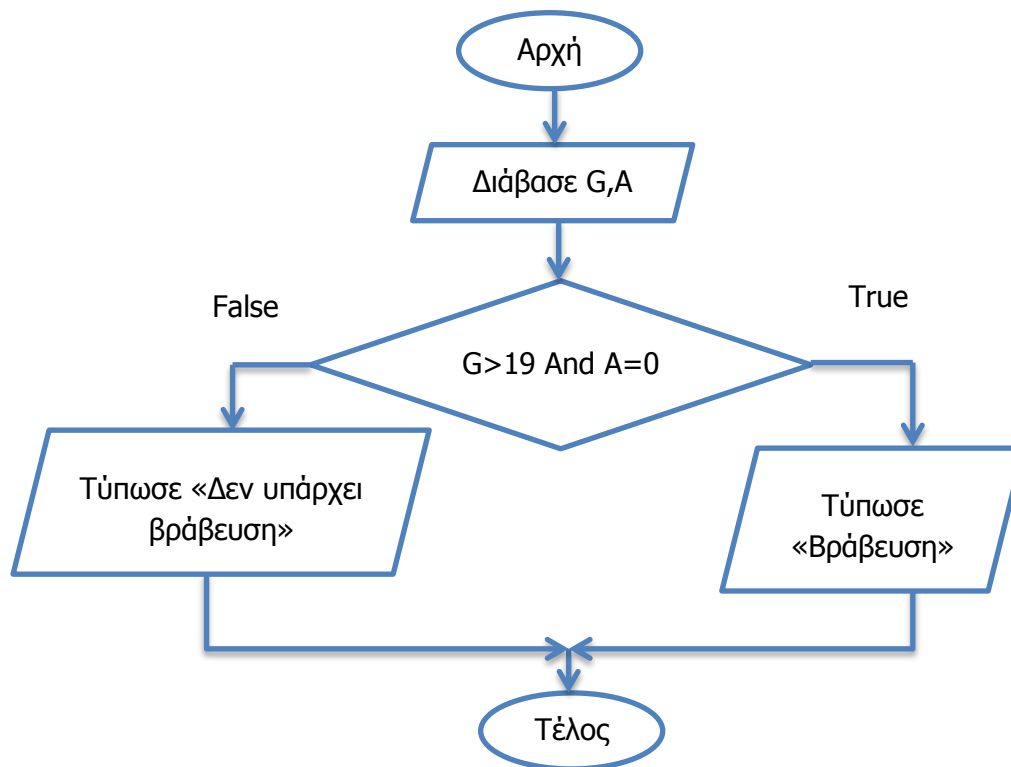
Μεταβλητές			Απόφαση		Παρουσίαση
X	Y	Z	X >= 0 And Y >= 0	True/False	15
10					
	5				
			10 >= 0 And 5 >= 0	True	
		15			

Πίνακας Προκαταρκτικής εκτέλεσης β

Μεταβλητές			Απόφαση		Παρουσίαση
X	Y	Z	X >= 0 And Y >= 0	True/False	-50
10					
	-5				
			10 >= 0 And -5 >= 0	False	
		-50			

Παράδειγμα 9

Ένα λύκειο αποφάσισε να δώσει το βραβείο «Επίδοσης & Ανελλιπούς Παρακολούθησης» σε όσους μαθητές πέτυχαν γενικό βαθμό μεγαλύτερο από το 19 και δεν είχαν ούτε μια απουσία. Να σχεδιάσετε το λογικό διάγραμμα που δέχεται τον γενικό βαθμό (G) και τις απουσίες (A) ενός μαθητή και τυπώνει το μήνυμα «Βράβευση» ή «Δεν υπάρχει βράβευση».

**2.2 Λογικός Τελεστής OR**

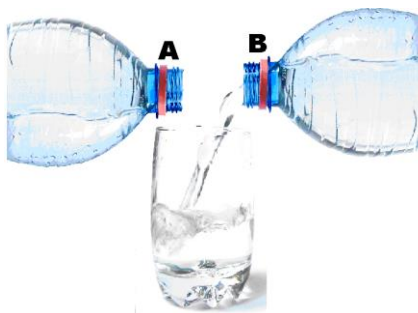
Ο λογικός τελεστής OR εκτελεί τη λογική πράξη OR ('Η) μεταξύ των συνθηκών της. Στον πιο κάτω πίνακα αληθείας του λογικού τελεστή OR φαίνονται τα αποτελέσματα.

Συνθήκη 1	λογική πράξη	Συνθήκη 2	Αποτέλεσμα
Αληθής	OR	Αληθής	Αληθής
Αληθής	OR	Ψευδής	Αληθής
Ψευδής	OR	Αληθής	Αληθής
Ψευδής	OR	Ψευδής	Ψευδής

Παραδείγματα λογικών εκφράσεων με τη χρήση του τελεστή OR

Έστω ότι έχουμε τις μεταβλητές A=10, B=20, C=30

Λογική Έκφραση	Αποτέλεσμα
(A>B) Or (A>C)	False
(A<B) Or (A<C)	True
(B>A) Or (B>C)	True
(A=B) Or (C=30)	True
(B>100) Or (C>100)	False

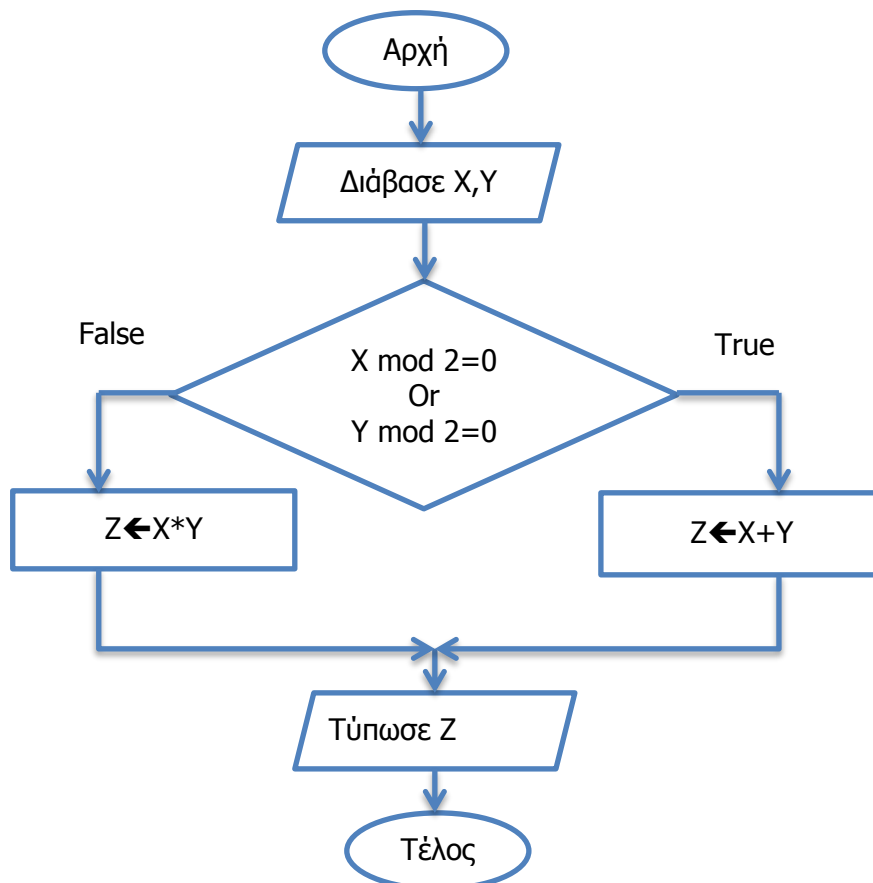
Παράδειγμα λογικού τελεστή OR από την καθημερινότητα**Νερό στο ποτήρι.**

Για να γεμίσει το ποτήρι με νερό θα πρέπει έστω και ένα από τα σημεία A **Ή** B να είναι ανοικτό

1. εάν το σημείο A είναι κλειστό (ψευδής) **Ή** το σημείο B είναι κλειστό (ψευδής) τότε δεν θα μπορεί να περάσει το νερό, αποτέλεσμα **ψευδής**.
2. εάν το σημείο A είναι ανοικτό (αληθής) **Ή** το σημείο B είναι κλειστό (ψευδής) τότε θα μπορεί να περάσει το νερό, αποτέλεσμα **αληθής**.
3. εάν το σημείο A είναι κλειστό (ψευδής) **Ή** το σημείο B είναι ανοικτό (αληθής) τότε θα μπορεί να περάσει το νερό, αποτέλεσμα **αληθής**.
4. εάν το σημείο A είναι ανοικτό (αληθής) **Ή** το σημείο B είναι ανοικτό (αληθής) τότε θα μπορεί να περάσει το νερό, αποτέλεσμα **αληθής**.

Παράδειγμα 10

Να σχεδιάσετε το λογικό διάγραμμα που δέχεται 2 ακέραιους θετικούς αριθμούς (X, Y). Αν ένας από τους δύο αριθμούς είναι άρτιος (το υπόλοιπο της διαίρεσης με το 2 είναι 0) τότε τυπώνει το άθροισμά τους, διαφορετικά τυπώνει το γινόμενό τους. Να γράψετε την προκαταρκτική εκτέλεση για αρχικές τιμές α) X=4, Y=7 και β) X=5, Y=7.



Πίνακας Προκαταρκτικής εκτέλεσης α

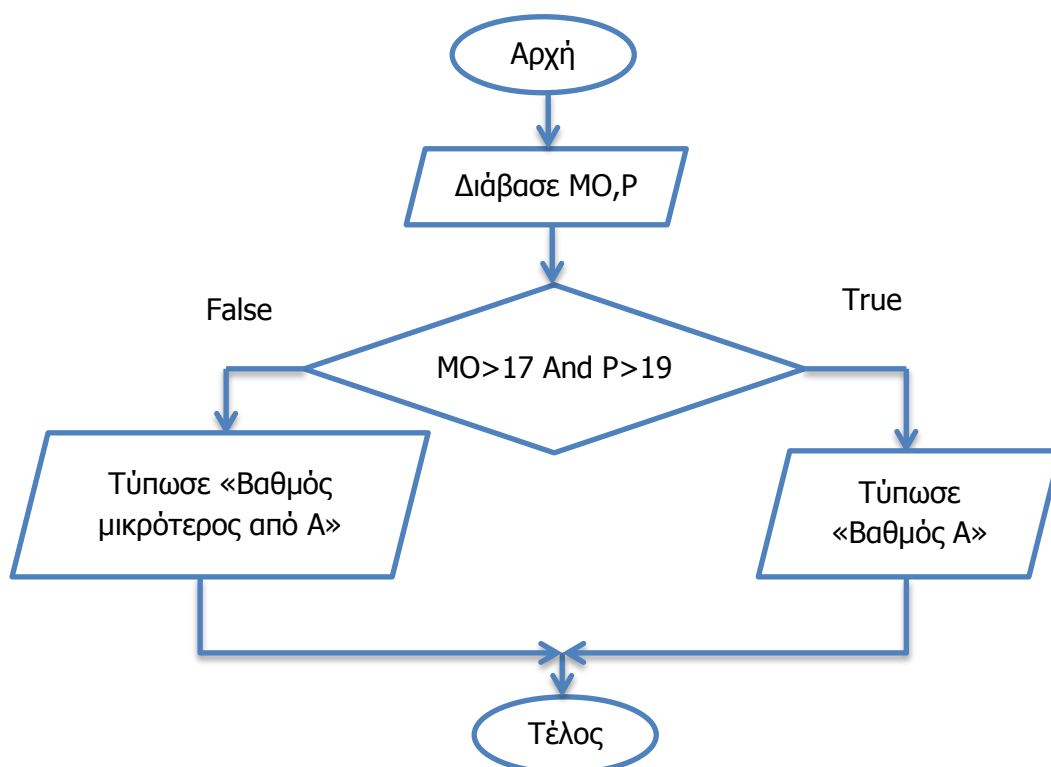
Μεταβλητές			Απόφαση		Παρουσίαση
X	Y	Z	$X \bmod 2 = 0$ Or $Y \bmod 2 = 0$	True/False	11
4					
	7				
			$4 \bmod 2 = 0$ Or $7 \bmod 2 = 0$	True	
		11			

Πίνακας Προκαταρκτικής εκτέλεσης β

Μεταβλητές			Απόφαση		Παρουσίαση
X	Y	Z	$X \bmod 2 = 0$ Or $Y \bmod 2 = 0$	True/False	35
5					
	7				
			$5 \bmod 2 = 0$ Or $7 \bmod 2 = 0$	False	
		35			

Παράδειγμα 11

Ένας καθηγητής αποφάσισε ότι για να πάρει ένας μαθητής «Α» στον βαθμό του τετραμήνου θα πρέπει ο μέσος όρος (ΜΟ) των διαγωνισμάτων του να είναι μεγαλύτερος από 17 και ο προφορικός (Ρ) του βαθμός να είναι μεγαλύτερος του 19. Να σχεδιάσετε το λογικό διάγραμμα που δέχεται τους δύο βαθμούς και τυπώνει το ανάλογο μήνυμα.

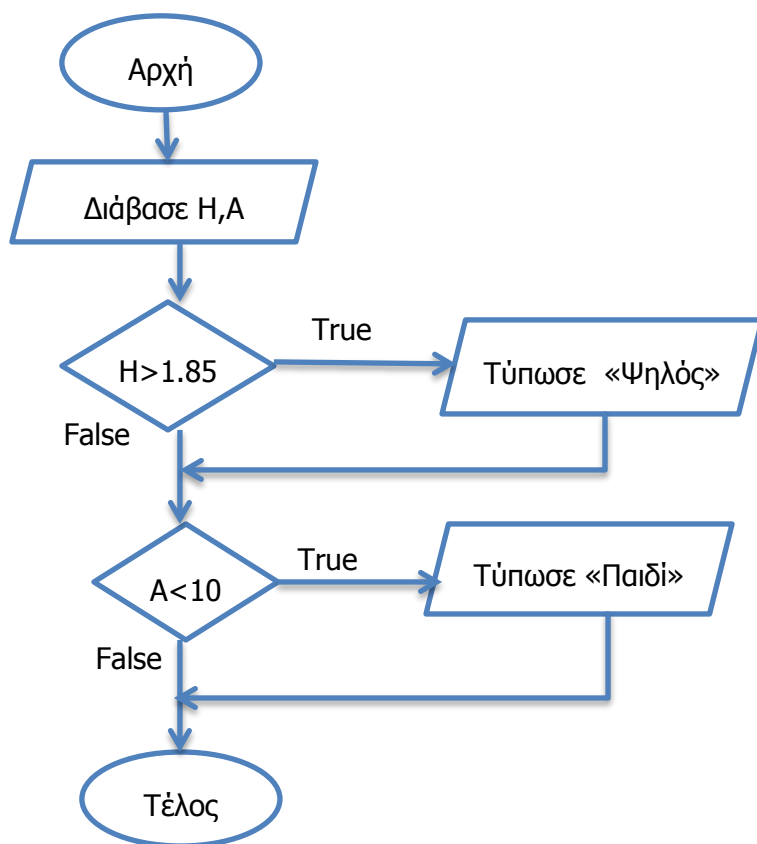


3. Προβλήματα με περισσότερες από μια ανεξάρτητες συνθήκες

Μέχρι τώρα αντιμετωπίσαμε προβλήματα με μόνο μία συνθήκη. Είναι φανερό όμως ότι αν θέλουμε να λύσουμε πιο σύνθετα προβλήματα θα χρειαστεί να χρησιμοποιήσουμε περισσότερους ελέγχους. Η φιλοσοφία αυτών των προβλημάτων παραμένει η ίδια και το μόνο που αλλάζει είναι η πολυπλοκότητα. Στα παρακάτω παραδείγματα θα δούμε προβλήματα που χρειάζονται περισσότερους ανεξάρτητους ελέγχους. Λέγοντας ανεξάρτητους ελέγχους εννοούμε ότι δεν θα έχουμε έλεγχο μέσα σε άλλον έλεγχο.

Παράδειγμα 12

Να σχεδιάσετε το λογικό διάγραμμα που δέχεται το ύψος (H) και την ηλικία (A) ενός ατόμου. Αν το ύψος είναι μεγαλύτερο από 1.85 τότε τυπώνεται το μήνυμα «Ψηλός» και αν η ηλικία είναι μικρότερη από 10 τότε τυπώνεται το μήνυμα «Παιδί».

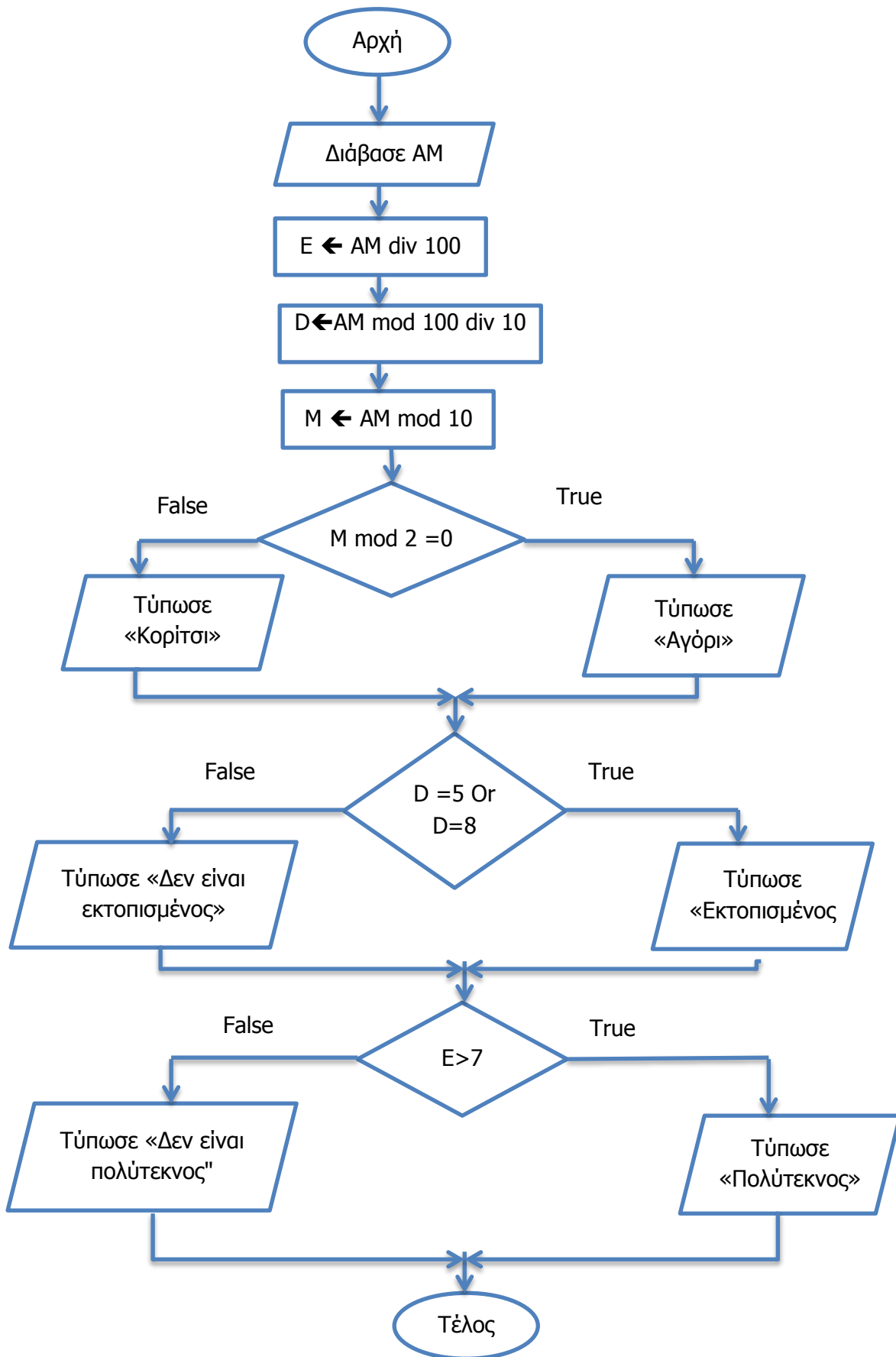


Παράδειγμα 13

Ένα σχολείο δημιουργεί τους τριψήφιους αριθμούς μητρώου των μαθητών του ως εξής:

- Αν το ψηφίο των μονάδων είναι άρτιος αριθμός τότε σημαίνει ότι ο μαθητής είναι αγόρι, διαφορετικά είναι κορίτσι.
- Αν το ψηφίο των δεκάδων είναι το 5 ή το 8 τότε σημαίνει ότι ο μαθητής είναι εκτοπισμένος, διαφορετικά δεν είναι.
- Αν ο αριθμός των εκατοντάδων είναι μεγαλύτερος του 7 τότε ο μαθητής είναι από πολύτεκνη οικογένεια, διαφορετικά δεν είναι.

Να σχεδιάσετε το λογικό διάγραμμα που δέχεται τον αριθμό μητρώου (AM) του μαθητή και τυπώνει τις πληροφορίες του. Να γράψετε, επίσης, την προκαταρκτική εκτέλεση για αρχικές τιμές α) AM=123 και β) AM=952



Πίνακας Προκαταρκτικής εκτέλεσης α

Μεταβλητές				Αποφάσεις					
AM	E	D	M	$M \bmod 2 = 0$	T/F	$D=5 \text{ or } D=8$	T/F	$E>7$	T/F
123									
	1	2	3						
				$3 \bmod 2 = 0$	F	$2=5 \text{ Or } 2=8$	F	$1>7$	F

Παρουσίαση
Κορίτσι Δεν είναι εκτοπισμένος Δεν είναι πολύτεκνος

Πίνακας Προκαταρκτικής εκτέλεσης β

Μεταβλητές				Αποφάσεις					
AM	E	D	M	$M \bmod 2 = 0$	T/F	$D=5 \text{ or } D=8$	T/F	$E>7$	T/F
952									
	9	5	2						
				$2 \bmod 2 = 0$	T	$5=5 \text{ Or } 5=8$	T	$9>7$	T

Παρουσίαση
Αγόρι Εκτοπισμένος Πολύτεκνος

4. Φωλιασμένη Δομή Διακλάδωσης

Η δομή Διακλάδωσης είναι η διαδικασία του ελέγχου κάποιας **συνθήκης** που μπορεί να έχει **δύο τιμές** (Αληθής ή Ψευδής) και ακολουθεί η απόφαση εκτέλεσης κάποιας ενέργειας με βάση την τιμή της λογικής αυτής συνθήκης. Στις περιπτώσεις που από τον έλεγχο τιμών προκύπτει αναγκαιότητα για καινούργιο έλεγχο τιμών, τότε έχουμε μια φωλιασμένη δομή διακλάδωσης.

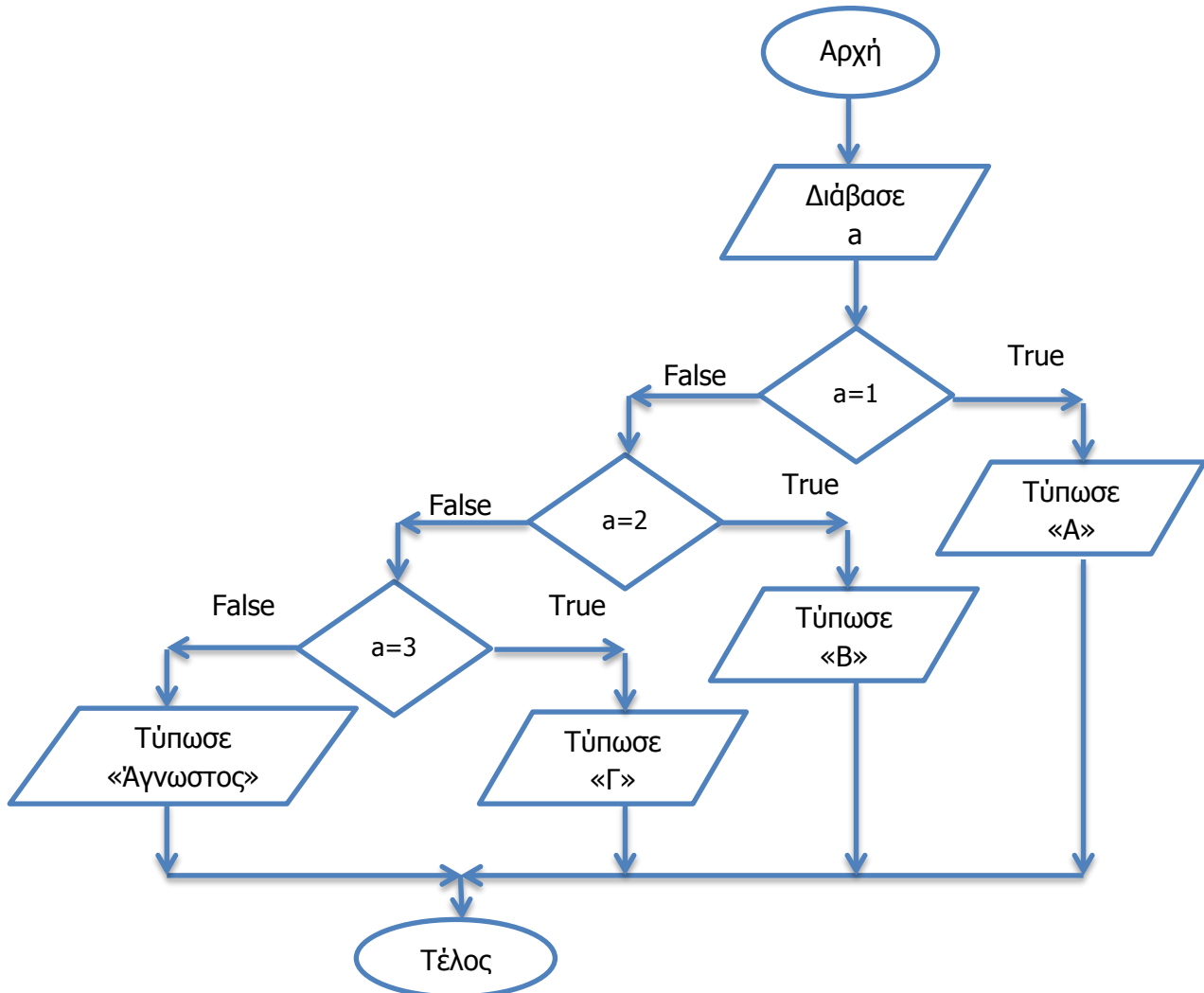
Οι διαδικασίες φωλιασμένης δομής διακλάδωσης εφαρμόζονται στα προβλήματα όπου μπορεί να ληφθούν περισσότερες από δύο διαφορετικές αποφάσεις, ανάλογα με την τιμή που παίρνει μία ή περισσότερες εκφράσεις.

Προβλήματα που απαιτούν φωλιασμένη δομή διακλάδωσης

- (1) Ένας αλγόριθμος που διαβάζει την ηλικία ενός ατόμου και ανάλογα με την ηλικία του εμφανίζει το ανάλογο μήνυμα → (1-5 «Βρεφοκομείο – νηπιαγωγείο», 6-11 «Δημοτικό», 12-14 «Γυμνάσιο», 15-17 «Λύκειο», 18-21 «Ενήλικας», κ.λπ.)
- (2) Ένας αλγόριθμος που διαβάζει τον βαθμό ενός μαθητή (1-20) και τυπώνει την βαθμολογία που αντιστοιχεί στο γυμνάσιο (0-9 «Ε», 10-12 «Δ», 13-15 «Γ», 16-18 «Β», 19-20 «Α»).
- (3) Ένας αλγόριθμος που διαβάζει έναν αριθμό από το 1-24 και αναλόγως ποιος αριθμός είναι τυπώνει το γράμμα του αλφαβήτου που αντιστοιχεί σε αυτόν.
- (4) Ένας αλγόριθμος που διαβάζει το ύψος και το βάρος ενός ατόμου, υπολογίζει το Δείκτη Μάζας Σώματος του ατόμου και ανάλογα με την τιμή του δείκτη αυτού τυπώνει κατάλληλο μήνυμα (<20 «Λιποβαρής», 20-25 «Κανονικό Βάρος», 26-29 «Υπέρβαρος», 30 «Παχύσαρκος», >30 «Σοβαρά παχύσαρκος»).
- (5) Ένας αλγόριθμος που διαβάζει τον μισθό ενός υπαλλήλου και αναλόγως του μισθού υπολογίζει ποιο φόρο θα πρέπει να του επιβάλει (<2000 «2%», 2000-3000 «3%», 3001-5000 «5%», >5000 «12%»).
- (6) Ένας αλγόριθμος που διαβάζει αριθμούς από το 1-12 και εμφανίζει το όνομα του αντίστοιχου μήνα.

Παράδειγμα 14

Να διαβάζεται ένας ακέραιος (a). Αν ο ακέραιος έχει τιμή 1 ή 2 ή 3 να εκτυπώνεται το αντίστοιχο γράμμα της αλφαβήτου (Α-Γ), διαφορετικά να εκτυπώνεται η λέξη «άγνωστος». Να γράψετε την προκαταρκτική εκτέλεση για αρχικές τιμές του a=2, 1, 3 και 4

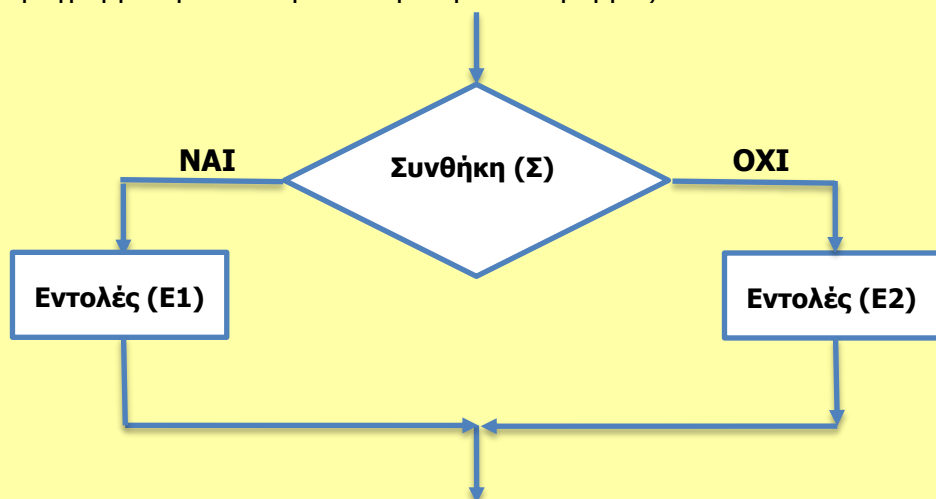


Πίνακες προκαταρκτικής εκτέλεσης για Παράδειγμα 14

Μεταβλητές	Αποφάσεις						Παρουσίαση
A	a=1	T/F	a=2	T/F	a=3	T/F	B
2	2=1	F	2=2	T			
Μεταβλητές	Αποφάσεις						Παρουσίαση
A	a=1	T/F	a=2	T/F	a=3	T/F	A
1	1=1	T					
Μεταβλητές	Αποφάσεις						Παρουσίαση
A	a=1	T/F	a=2	T/F	a=3	T/F	Γ
3	3=1	F	3=2	F	3=3	T	
Μεταβλητές	Αποφάσεις						Παρουσίαση
A	a=1	T/F	a=2	T/F	a=3	T/F	Άγνωστος
4	4=1	F	4=2	F	4=3	F	

Βασικές Έννοιες**Δομή****Διακλάδωσης:**

Δομή διακλάδωσης έχουμε όταν σε έναν αλγόριθμο πρέπει να εκτελεστεί μία από δύο εναλλακτικές σειρές εντολών με βάση μια απόφαση, ελέγχοντας μια λογική συνθήκη. Το γεωμετρικό σχήμα που χρησιμοποιούμε στα λογικά διαγράμματα για να δείξουμε ότι το πρόγραμμα πρέπει να μία απόφαση είναι ο ρόμβος.

**Λογικοί****Τελεστές:**

Χρησιμοποιούνται για να δημιουργήσουμε σύνθετες λογικές εκφράσεις. Οι λογικοί τελεστές που χρησιμοποιούμε είναι το NOT, AND και OR.

Ο λογικός τελεστής AND εκτελεί τη λογική πράξη AND (ΚΑΙ) μεταξύ δυο συνθηκών της λογικής έκφρασης. Ο πίνακας αληθείας του είναι:

Συνθήκη 1	Λογική πράξη	Συνθήκη 2	Αποτέλεσμα
Αληθής	AND	Αληθής	Αληθής
Αληθής	AND	Ψευδής	Ψευδής
Ψευδής	AND	Αληθής	Ψευδής
Ψευδής	AND	Ψευδής	Ψευδής

Ο λογικός τελεστής OR εκτελεί την λογική πράξη OR (Ή) μεταξύ δυο συνθηκών της λογικής έκφρασης. Ο πίνακας αληθείας του είναι:

Συνθήκη 1	Λογική πράξη	Συνθήκη 2	Αποτέλεσμα
Αληθής	OR	Αληθής	Αληθής
Αληθής	OR	Ψευδής	Αληθής
Ψευδής	OR	Αληθής	Αληθής
Ψευδής	OR	Ψευδής	Ψευδής

Φωλιασμένη**Δομή****Διακλάδωσης:**

Οι διαδικασίες φωλιασμένης δομής διακλάδωσης εφαρμόζονται στα προβλήματα όπου μπορεί να ληφθούν περισσότερες από δύο διαφορετικές αποφάσεις ανάλογα με την τιμή που παίρνει μία έκφραση.

Γ7.5 Γλώσσες Προγραμματισμού

Τι θα μάθουμε σήμερα:

- ❖ Να εξηγούμε τι είναι γλώσσα προγραμματισμού
- ❖ Να διακρίνουμε τη διαφορά μεταξύ αλγορίθμου και γλώσσας προγραμματισμού
- ❖ Να διακρίνουμε τις διάφορες γλώσσες προγραμματισμού και να αναφέρουμε τα κύρια χαρακτηριστικά τους
- ❖ Να περιγράφουμε τι είναι γλώσσα μηχανής
- ❖ Να περιγράφουμε τη χρήση των μεταγλωττιστών και των διερμηνέων σε μια γλώσσα προγραμματισμού
- ❖ Να ξεχωρίζουμε τον πηγαίο κώδικα από τον εκτελέσιμο.

1. Εισαγωγή

Γλώσσα προγραμματισμού είναι μια τεχνητή γλώσσα που λειτουργεί ως μέσο με το οποίο μπορούμε να μετατρέψουμε έναν αλγόριθμο σε πρόγραμμα, ώστε να εκτελείται από τον υπολογιστή. Οποιοδήποτε πρόγραμμα βλέπετε στον υπολογιστή σας, είτε πρόκειται για παιχνίδι είτε για λειτουργικό σύστημα, έχει δημιουργηθεί χρησιμοποιώντας μια γλώσσα προγραμματισμού.

Επειδή όπως θα εξηγήσουμε στη συνέχεια, η περιγραφή των εντολών με τρόπο που να μπορεί να τις εκτελεί ο υπολογιστής είναι δύσκολη για τον άνθρωπο, οι γλώσσες προγραμματισμού λειτουργούν ως ένα ενδιάμεσο στάδιο, ώστε η περιγραφή του αλγορίθμου να γίνεται με ευκολία και ακρίβεια και στη συνέχεια, με αυτόματο τρόπο, να μεταφράζεται σε μορφή που να μπορεί να εκτελεστεί από την ΚΜΕ.

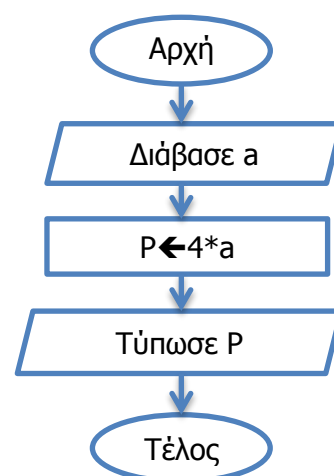
2. Γλώσσα Μηχανής

Όπως ήδη έχουμε μάθει, ο υπολογιστής, και συγκεκριμένα η Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) εκτελεί εντολές οι οποίες είναι διατυπωμένες με τρόπο που να μπορεί να τις επεξεργαστεί, δηλαδή με δυαδική μορφή. Οι εντολές στην περίπτωση αυτή περιγράφονται σε **Γλώσσα Μηχανής (Machine Language)**, δηλαδή τη γλώσσα που «καταλαβαίνει» η ΚΜΕ.

Π.χ. ο αλγόριθμος για τον υπολογισμό της περιμέτρου ενός τετραγώνου, τον οποίο είδαμε στην αρχή της ενότητας, περιγράφεται με το λογικό διάγραμμα που φαίνεται στην Εικόνα 30. Η διατύπωση του αλγορίθμου αυτού σε Γλώσσα Μηχανής φαίνεται στην επόμενη σελίδα στην Εικόνα 31(γ).

Για την ΚΜΕ αυτή είναι η «μητρική γλώσσα», αλλά για τον προγραμματιστή είναι εξαιρετικά δύσκολο να περιγράφει εντολές σε αυτή τη μορφή και η πιθανότητα να γίνουν λάθη στη μετάφραση είναι πολύ μεγάλη.

Το πλεονέκτημα που είχε η γλώσσα μηχανής ήταν ότι γινόνταν κατανοητή απευθείας από την ΚΜΕ. Τα μειονεκτήματά της, πέρα από τη δυσκολία προγραμματισμού, είναι ότι κάθε «οικογένεια» ΚΜΕ είχε τη δική της γλώσσα μηχανής (δηλαδή ποιος συνδυασμός από 0 και 1 αντιστοιχεί στην κάθε εντολή).



Εικόνα 30 Λογικό διάγραμμα για τον υπολογισμό της περιμέτρου ενός τετραγώνου

[illegible]

Εικόνα 31 Το ίδιο πρόγραμμα σε γλώσσα (α) Υψηλού επιπέδου, (β) Χαμηλού επιπέδου και (γ) Μηχανής

3. Γλώσσες Προγραμματισμού

Οι πρώτοι προγραμματιστές ήταν αναγκασμένοι να προγραμματίζουν σε Γλώσσα Μηχανής και άρχισαν να αναζητούν πιο φιλικούς προς τον άνθρωπο τρόπους περιγραφής ενός προγράμματος. Αρχικά προσπάθησαν να περιγράψουν με συμβολικό τρόπο την κάθε εντολή της Γλώσσας Μηχανής και έτσι δημιουργήθηκε η Συμβολική Γλώσσα (Assembly Language). Αντί, δηλαδή, ο άνθρωπος να γράφει το πρόγραμμα απευθείας σε δυαδική μορφή, το έγραφε χρησιμοποιώντας μικρές συμβολικές λέξεις (π.χ. `movl` σημαίνει «μετάφερε δεδομένα από/προς έναν καταχωρητή» , `mul` σημαίνει «πολλαπλασίασε» κ.λπ.). Προγράμματα γραμμένα σε αυτές τις γλώσσες προγραμματισμού χρειάζεται να μεταφραστούν, εντολή προς εντολή στις αντίστοιχες εντολές της Γλώσσας Μηχανής, όπως φαίνεται στην Εικόνα 31(β). Οι γλώσσες αυτές συχνά χαρακτηρίζονται ως **Γλώσσες Χαμηλού Επιπέδου**, αφού βρίσκονται ένα μόνο βήμα μακριά από τη Γλώσσα Μηχανής. Η μετάφραση σε γλώσσα μηχανής γίνεται αυτόματα με κατάλληλο πρόγραμμα το οποίο δέχεται στην είσοδο το αρχείο με τις συμβολικές εντολές και παράγει στην έξοδο ένα αρχείο με τις αντίστοιχες εντολές σε δυαδικό σύστημα (το «εκτελέσιμο» αρχείο).

Επειδή η Γλώσσες Χαμηλού Επιπέδου αναγκάζουν τον/την προγραμματιστή/τρια να «σκέφτεται» περισσότερο όπως η ΚΜΕ παρά με τρόπο παρόμοιο με τον αλγόριθμο, σταδιακά δημιουργήθηκαν τρόποι περιγραφής οι οποίοι μοιάζουν περισσότερο στην ανθρώπινη γλώσσα. Όπως και οι φυσικές γλώσσες των ανθρώπων, έτσι και αυτές έχουν λεξιλόγιο και σύνταξη, τα οποία πρέπει να ακολουθούνται πιστά και χαρακτηρίζονται ως **Γλώσσες Υψηλού Επιπέδου**, αφού είναι πιο κοντά στην ανθρώπινη γλώσσα παρά στη «γλώσσα» της ΚΜΕ. Ένα παράδειγμα δίνεται στην Εικόνα 31(α), ο αλγόριθμος για υπολογισμό της

περιμέτρου του τετραγώνου διατυπωμένος στη γλώσσα υψηλού επιπέδου Pascal. Μερικές από τις πιο γνωστές γλώσσες προγραμματισμού Ψηλού Επιπέδου είναι οι ακόλουθες:

- **Visual Basic:** Χρησιμοποιείται για προγράμματα που θα «τρέξουν» σε Windows. Εύκολη στην εκμάθησή της.
- **C/C++:** Πρόκειται για γλώσσες με μεγάλες δυνατότητες που στόχο έχουν τη δημιουργία πολύπλοκων προγραμμάτων (τα Windows είναι «γραμμένα» κατά κύριο λόγο σε C++).
- **Pascal:** Γλώσσα που χρησιμοποιείται κατά κύριο λόγο για εκπαιδευτικούς λόγους. Προτιμάται για το «λεξιλόγιο» της που είναι πολύ κοντά στη φυσική γλώσσα.
- **Java:** Κύριο χαρακτηριστικό αυτής της γλώσσας είναι ότι μπορεί να δημιουργήσει εφαρμογές ανεξαρτήτως πλατφόρμας (λειτουργικού συστήματος). Οι περισσότερες εφαρμογές για κινητά τηλέφωνα (π.χ. Android) έχουν δημιουργηθεί με Java.
- **Perl:** Αρχικά δημιουργήθηκε για γρήγορη ανάλυση δεδομένων και δημιουργία εκθέσεων. Παρέχει εύκολους τρόπους αναζήτησης και μετατροπής δεδομένων από μια μορφή σε άλλη. Χρησιμοποιείται κυρίως από διαχειριστές συστημάτων και για αυτοματοποιήσεις σε ιστοσελίδες.
- **Python:** Γλώσσα προγραμματισμού η οποία δημιουργήθηκε με στόχο το ξεκάθαρο και απλό συντακτικό. Χρησιμοποιείται για αυτοματοποιήσεις σε ιστοσελίδες, ως γλώσσα αυτοματισμού σε εφαρμογές και ως εκπαιδευτική γλώσσα.

Παρακάτω παρουσιάζονται οι εντολές στις πιο πάνω γλώσσες για το πρόγραμμα που υπολογίζει την περίμετρο ενός τετραγώνου.

<pre>#include <iostream> using namespace std; int main() { int a, P; cin >> a; P = 4 * a; cout << P; return 0; }</pre> (α) C++	<pre>Private Sub cmdCalculate_Click() Dim a As Integer Dim P As Integer a = InputBox("Δώσε την πλευρά") P = 4 * a MsgBox P End Sub</pre> (β) Visual Basic
<pre>Program tetragono; var a, P:integer; begin Readln(a); P := 4*a; Writeln(P); End.</pre> (γ) Pascal	<pre>Import java.util.*; public class tetragono { public static void main(String[] args){ Scanner in = new Scanner(System.in); int a, P; a = in.nextInt(); P = 4 * a; System.out.println(P); } }</pre> (δ) Java

Εικόνα 32 Πρόγραμμα το οποίο υπολογίζει την περίμετρο ενός τετραγώνου:

(α) σε γλώσσα προγραμματισμού C++, (β) σε γλώσσα προγραμματισμού Visual Basic, (γ) σε γλώσσα προγραμματισμού Pascal, (δ) σε γλώσσα προγραμματισμού Java

Οι γλώσσες υψηλού επιπέδου είναι σχετικά εύκολες στη μάθηση και επιτρέπουν στον/στην προγραμματιστή/τρια να δημιουργεί πολύπλοκα προγράμματα σε σχετικά σύντομο χρονικό διάστημα, αυξάνοντας την παραγωγικότητά του/της. Παρόλο που οι πιο πάνω γλώσσες έχουν διαφορές, όταν πρόκειται για απλά προγράμματα βλέπουμε ότι έχουν πολλές ομοιότητες.

Στις Γλώσσες Υψηλού Επιπέδου, πολλά στοιχεία που χρησιμοποιούνται συχνά (υπολογισμοί, συναρτήσεις, κουμπιά, παράθυρα, κ.λπ.) υπάρχουν ως έτοιμα τμήματα προγραμμάτων, ενώ παράλληλα παρέχονται πολλά χρήσιμα εργαλεία για την ανάπτυξη προγραμμάτων.

4. Μεταγλωττιστές και Διερμηνείς

Οι γλώσσες προγραμματισμού υψηλού επιπέδου χρησιμοποιούν συντακτικό και λεξιλόγιο που είναι πιο κοντά στη φυσική γλώσσα, αλλά τα προγράμματα που γράφονται με αυτές θα πρέπει να «μεταφραστούν» σε εντολές της Γλώσσας Μηχανής. Η «μετάφραση» γίνεται αυτόματα με κατάλληλα προγράμματα και είναι μια πολύπλοκη διαδικασία.

Υπάρχουν δύο κατηγορίες «μεταφραστών». Ανάλογα με τη γλώσσα προγραμματισμού, τα προγράμματα αυτά είναι οι **μεταγλωττιστές (compilers)** και οι **διερμηνείς (interpreters)**. Η διαφορά τους είναι ότι οι μεταγλωττιστές ελέγχουν ολόκληρο το πρόγραμμα για συντακτικά λάθη και στη συνέχεια το μετατρέπουν σε γλώσσα μηχανής και αποθηκεύουν τις εντολές σε ένα εκτελέσιμο αρχείο, το οποίο δίνεται στον χρήστη για να εκτελέσει αργότερα. Αντίθετα, οι διερμηνείς ελέγχουν μία-μία με τη σειρά τις εντολές του προγράμματος και εάν δεν υπάρχει σφάλμα, εκτελούν την κάθε εντολή πριν προχωρήσουν στην επόμενη.

Οι μεταγλωττιστές χρησιμοποιούνται συνήθως για μεγαλύτερα και πιο σύνθετα προγράμματα ή όπου η ταχύτητα εκτέλεσης είναι σημαντική. Οι γλώσσες προγραμματισμού C/C++ και Pascal χρησιμοποιούν μεταγλωττιστή. Όπου η άμεση εκτέλεση είναι πιο σημαντική από την ταχύτητα χρησιμοποιούνται διερμηνείς. Οι γλώσσες προγραμματισμού Visual Basic, Perl και Python κάνουν χρήση διερμηνέων. Η γλώσσα προγραμματισμού Java χρησιμοποιεί ένα συνδυασμό των δύο. Ένας μεταγλωττιστής μεταφράζει το πρόγραμμα σε μια ενδιάμεση μορφή η οποία μοιάζει με γλώσσα μηχανής. Στη συνέχεια αυτό το ενδιάμεσο πρόγραμμα το επεξεργάζεται ένας σχετικά απλός διερμηνέας. Για να εκτελούνται προγράμματα γραμμένα σε Java σε ένα νέο υπολογιστή ή λειτουργικό σύστημα, χρειάζεται να εγκατασταθεί μόνο ο διερμηνέας (γι αυτό και προγράμματα γραμμένα σε Java είναι διαθέσιμα για πολλές και διαφορετικές συσκευές).

Βασικές Έννοιες

Γλώσσα Προγραμματισμού:	Είναι τρόπος περιγραφής ενός αλγορίθμου σε πρόγραμμα με σκοπό να μεταφραστεί σε μορφή που να εκτελείται από την ΚΜΕ του υπολογιστή.
Γλώσσα Μηχανής:	Η γλώσσα που «καταλαβαίνει» η ΚΜΕ. Οι εντολές αναπαριστούνται με συνδυασμό δυαδικών ψηφίων.
Γλώσσα Χαμηλού Επιπέδου:	Συμβολική γλώσσα αντίστοιχη με τη γλώσσα μηχανής, αλλά πιο εύκολη στη χρήση από τους ανθρώπους.
Γλώσσα Υψηλού Επιπέδου:	Γλώσσα περιγραφής ενός αλγορίθμου παραπλήσια με την ανθρώπινη, με δικό της συντακτικό και γραμματική. Παραδείγματα: Pascal, C/C++, Visual Basic, Java, Perl, Python.
Μεταγλωττιστής (Compiler):	Πρόγραμμα μετάφρασης Γλώσσας Υψηλού Επιπέδου σε Γλώσσα Μηχανής, το οποίο ελέγχει ολόκληρο το πρόγραμμα για συντακτικά λάθη και στη συνέχεια το μετατρέπει σε γλώσσα μηχανής και αποθηκεύουν τις εντολές σε ένα εκτελέσιμο αρχείο.
Διερμηνέας (Interpreter):	Πρόγραμμα μετάφρασης Γλώσσας Υψηλού Επιπέδου σε Γλώσσα Μηχανής, το οποίο ελέγχει μία-μία εντολή για συντακτικά λάθη και στη συνέχεια την εκτελεί πριν προχωρήσει στην επόμενη.

Πηγές

1. Αράπογλου Α., Μαβόγλου Χ., Οικονομάκος Η., Φύτρος Κ., (2006) *Πληροφορική Α', Β', Γ' Γυμνασίου*, σελ. 182—184, ΟΕΔΒ.

Γ7.6 Η Γλώσσα Προγραμματισμού Pascal

Τι θα μάθουμε σήμερα:

- Να χρησιμοποιούμε το περιβάλλον προγραμματισμού του *Pame* για να ανοίγουμε (Open), να αποθηκεύουμε (Save), να αποσφαλματώνουμε και να εκτελούμε (Run) προγράμματα στη γλώσσα προγραμματισμού Pascal
- Να επιλέγουμε κατάλληλο τύπο δεδομένων για τις μεταβλητές ενός προγράμματος
- Να χρησιμοποιούμε εντολές εισόδου και εξόδου
- Να επιλέγουμε και να τοποθετούμε τις κατάλληλες εντολές με τη σωστή σειρά, για να μετατρέψουμε έναν αλγόριθμο σε πρόγραμμα
- Να εκτελούμε το πρόγραμμα και να διαπιστώνουμε κατά πόσο εκτελείται σωστά.

1. Εισαγωγή

Η γλώσσα προγραμματισμού Pascal δημιουργήθηκε το 1971 στο πανεπιστήμιο της Γενεύης από τον Ελβετό καθηγητή Nicklaus Wirth. Η γλώσσα αυτή είναι ιδιαίτερα δημοφιλής για διδασκαλία προγραμματισμού, καθώς η σύνταξη της είναι σχετικά εύκολη για μάθηση και παράλληλα προωθεί την παραγωγή δομημένων προγραμμάτων.

Το *Pame* αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης προγραμμάτων σε γλώσσα προγραμματισμού Pascal, απευθυνόμενο σε μαθητές και βασίζεται στον σύγχρονο και δημοφιλή μεταγλωττιστή FreePascal. Το συγκεκριμένο λογισμικό διατίθεται δωρεάν μέσω της ιστοσελίδας <http://users.sch.gr/samartzidis/pame/>.



Εικόνα 33 Λογισμικό Pame

2. Ενεργοποίηση του Pame

Διπλό κλικ στο εικονίδιο του Pame που βρίσκεται στην επιφάνεια εργασίας.



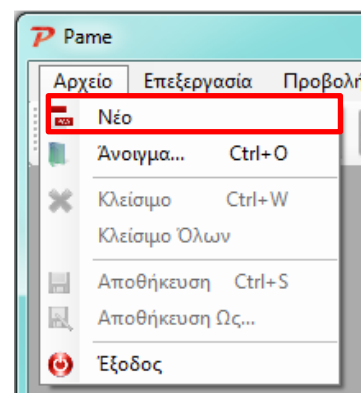
Εικόνα 34 Συντόμευση του Pame στην επιφάνεια εργασίας

3. Δημιουργία νέου αρχείου

- (1) Από το κυρίως μενού του προγράμματος κάνουμε κλικ στο **Αρχείο**.
- (2) Επιλέγουμε το **Νέο**.

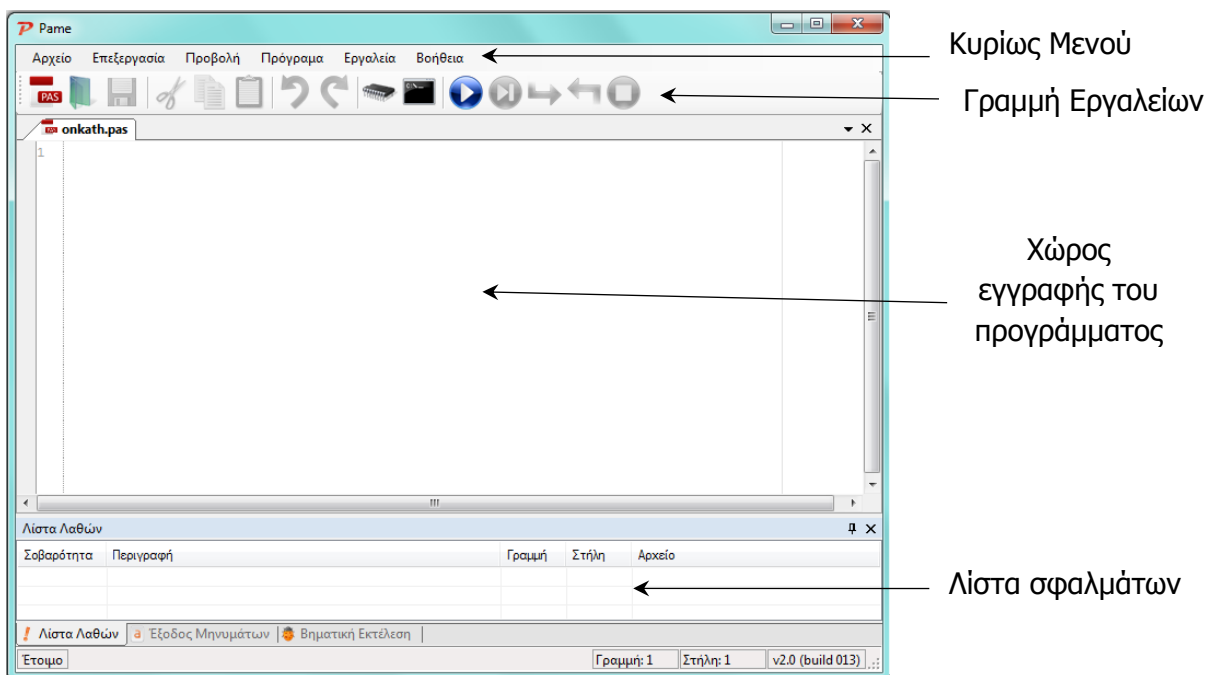
Ή

- (1) Από την γραμμή εργαλείων επιλέγουμε το εργαλείο



Εικόνα 35 Δημιουργία καινούργιου αρχείου

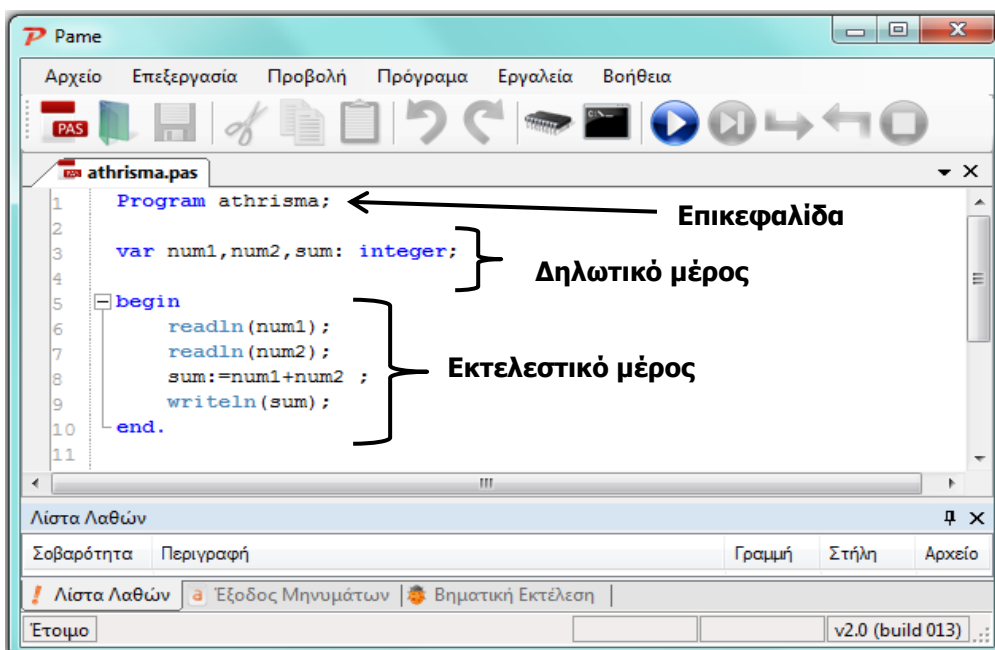
4. Το παράθυρο του περιβάλλοντος *Pame*

Εικόνα 36 Το παράθυρο *Pame*

5. Εισαγωγή Κώδικα

Ένα πρόγραμμα στην Pascal αποτελείται από τρία μέρη:

- Επικεφαλίδα
- Δηλωτικό μέρος
- Εκτελεστικό μέρος



Εικόνα 37 Δομή προγράμματος στην Pascal

Πρέπει ακόμη να έχουμε υπόψη μας ότι όλες οι εντολές στην Pascal τελειώνουν με το σύμβολο στίξης ; (ελληνικό ερωτηματικό) εκτός από την αρχή του εκτελεστικού μέρους, μετά δηλαδή τη λέξη **begin** και στο τέλος του εκτελεστικού μέρους, μετά τη λέξη **end** στο οποίο βάζουμε το σύμβολο στίξης . (τελεία) (άλλες περιπτώσεις στις οποίες εξαιρείται θα δούμε σε πιο προχωρημένες τάξεις).

5.1 Επικεφαλίδα

Η επικεφαλίδα αποτελείται από τη δεσμευμένη λέξη **program**, ακολουθούμενη από ένα αναγνωριστικό, το οποίο αντιπροσωπεύει το όνομα του προγράμματος. Το όνομα μπορεί να έχει μήκος μέχρι 64 χαρακτήρες άρα μπορούμε να το κάνουμε να είναι αρκετά «αναγνωριστικό». Π.χ.

```
program athroisma;
```

5.2 Δηλωτικό μέρος

Στο δηλωτικό μέρος ορίζουμε τις σταθερές και τις μεταβλητές που θα χρησιμοποιήσουμε αργότερα στο πρόγραμμά μας.

Η δήλωση σταθερών γίνεται με τη δεσμευμένη λέξη **const** , ακολουθούμενη από έναν κατάλογο με τα αναγνωριστικά (ονόματα των σταθερών) και την τιμή που θέλουμε να δώσουμε στην κάθε σταθερά.

```
const  
  vat   = 0.17;  
  name  = 'Nicolas';
```

Η τιμή που ανατίθεται σε μια σταθερά δεν μπορεί να αλλάξει κατά τη διάρκεια εκτέλεσης του προγράμματος.

Οι σταθερές χρησιμοποιούνται συνήθως όταν έχουμε μια τιμή στην οποία αναφερόμαστε/χρησιμοποιούμε σε όλη την έκταση του προγράμματος, η οποία όμως μπορεί να αλλάξει στο μέλλον (π.χ. ΦΠΑ). Με αυτό τον τρόπο, όταν κάποτε χρειαστεί να αλλάξει αυτή η τιμή μπορούμε να αλλάξουμε μόνο την τιμή της σταθεράς στην αρχή του προγράμματος.

Η δήλωση μεταβλητών γίνεται με τη δεσμευμένη λέξη **var** , ακολουθούμενη έναν κατάλογο με τα αναγνωριστικά (ονόματα των μεταβλητών) μαζί και με τον τύπο δεδομένων της κάθε μεταβλητής.

Στις μεταβλητές αναθέτονται τιμές διάφορων τύπων δεδομένων, όμως αυτές οι τιμές μπορούν να αλλάζουν κατά τη διάρκεια εκτέλεσης του προγράμματος. Τρεις βασικοί τύποι δεδομένων που θα χρησιμοποιήσουμε είναι οι εξής:

Integer: Αντιπροσωπεύει ακέραιους αριθμούς. Για αυτό κάθε δεδομένο που δηλώνεται ως **integer** είναι ένας ακέραιος αριθμός. Ο τύπος **integer** μπορεί να περιέχει ακέραιους αριθμούς από το -32768 μέχρι το 32767 (16bit). Για μεγαλύτερο εύρος τιμών υπάρχει ο τύπος **longint** (32bit).

Real: Αντιπροσωπεύει τους πραγματικούς (δεκαδικούς) αριθμούς. Ο τύπος **real** μπορεί να περιέχει πραγματικούς αριθμούς από το 3.4×10^{-38} μέχρι το 3.4×10^{38} για θετικούς και το ίδιο για αρνητικούς αριθμούς.

String: Αντιπροσωπεύει αλφαριθμητικά (ακολουθίες χαρακτήρων, κείμενο).

Π.χ.

```
var
  Hlikia: integer;
  Ypsos : real;
  Name   : string;
```

5.3 Εκτελεστικό μέρος

Το εκτελεστικό μέρος ξεκινά με τη δεσμευμένη λέξη **begin** και τελειώνει με τη δεσμευμένη λέξη **end**. Η λέξη **end** ακολουθείται από μια τελεία, που υποδηλώνει και το τέλος του προγράμματος. Ανάμεσα στο **begin** και το **end** βρίσκονται όλες οι εντολές του προγράμματος, οι οποίες αντιστοιχούν στον αλγόριθμο και μπορούν να κάνουν εργασίες όπως είσοδος δεδομένων, εκχώρηση τιμών και τροποποίησή τους (επεξεργασία) και εμφάνιση τους (έξοδος πληροφοριών).

Είσοδος: Για είσοδο δεδομένων χρησιμοποιούμε την εντολή **readln**, η οποία έχει την πιο κάτω μορφή:

```
readln(μεταβλητή) ;
```

Μπορούμε να διαβάσουμε περισσότερες από μία μεταβλητές στη σειρά:

```
readln(μεταβλητή_1, μεταβλητή_2, μεταβλητή_3,...) ;
```

Π.χ.

```
readln(height) ;
readln(bathmos1, bathmos2,bathmos3) ;
```

Έξοδος: Η εντολή για παρουσίαση αποτελεσμάτων είναι η εντολή **writeln**, η οποία έχει την πιο κάτω μορφή:

```
writeln(μεταβλητή) ;
```

Μπορούμε να εμφανίσουμε μαζί κάποιο μήνυμα ή περισσότερες από μία μεταβλητές στη σειρά:

```
writeln('Μήνυμα σε μονά εισαγωγικά ',μεταβλητή) ;
writeln(μεταβλητή_1, μεταβλητή_2, μεταβλητή_3,...) ;
writeln(emvdon) ;
Writeln (' Ο μέσος όρος του μαθητή είναι: ', mesos_oros:8:2) ;
```

Επεξεργασία: Μια βασική λειτουργία σε ένα πρόγραμμα είναι η πραγματοποίηση κάποιου υπολογισμού, το αποτέλεσμα του οποίου (η τιμή) να εκχωρηθεί σε κάποια μεταβλητή για να χρησιμοποιηθεί αργότερα. Το μέρος υπολογισμού της τιμής αυτής ονομάζεται **έκφραση** και γίνεται χρησιμοποιώντας αριθμούς, τις τιμές άλλων μεταβλητών ή σταθερών, σύμβολα τελεστών των διάφορων μαθηματικών πράξεων και παρενθέσεις. Η εκχώρηση τιμής σε μια μεταβλητή γίνεται με το σύμβολο **:=**. Η γενική σύνταξη της εκχώρησης τιμής είναι:

```
μεταβλητή := έκφραση;
```

Π.χ.

```
perimetros := 4*plevra;
mesos_oros := (bathmos1+bathmos2+bathmos3)/3;
```

5.4 Εκτύπωση μεταβλητών τύπου Real

Οι μεταβλητές τύπου Real με την εντολή Writeln παρουσιάζονται στην οθόνη σε εκθετική μορφή.

Αν για παράδειγμα η μεταβλητή X έχει την τιμή 40.52 τότε η εντολή Writeln(X) θα παρουσιάσει στην οθόνη τον αριθμό 4.05200000000000E+001.

Αν δεν θέλουμε το περιεχόμενο της μεταβλητής να τυπωθεί σε εκθετική μορφή τότε πρέπει να καθορίσουμε σε πόσα διαστήματα θα τυπωθεί η μεταβλητή και πόσα δεκαδικά ψηφία θα εμφανιστούν.

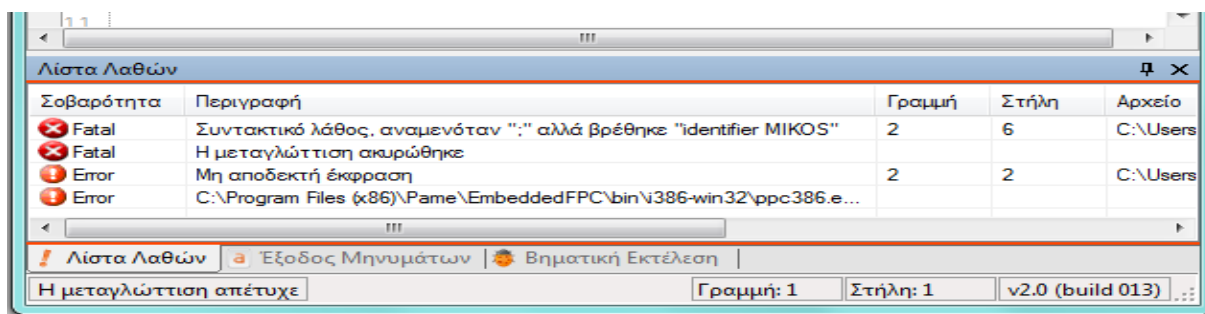
Για παράδειγμα αν X=40.52 η εντολή Writeln(X:6:2) θα τυπώσει στην οθόνη το περιεχόμενο της μεταβλητής X σε 6 διαστήματα με δύο δεκαδικά ψηφία δηλαδή 40.52.

Θα πρέπει να προσέξουμε ώστε να υπάρχουν αρκετά διαστήματα για να χωρέσει το πρόσημο, το ακέραιο μέρος, η τελεία και τα δεκαδικά ψηφία που θέλουμε.

6. Αποσφαλμάτωση (Debugging)

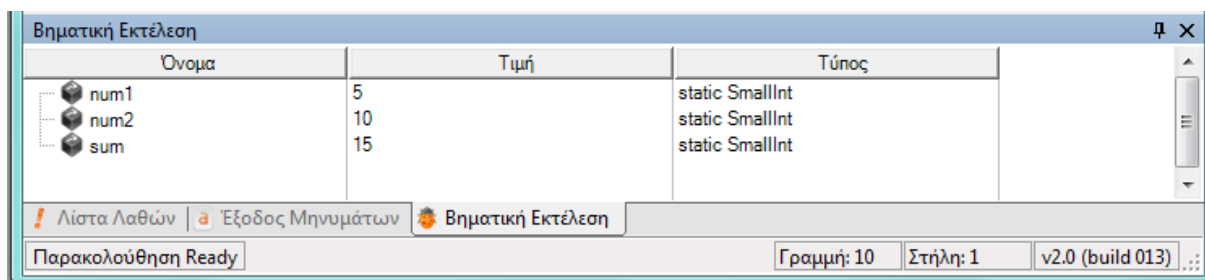
Η αποσφαλμάτωση του προγράμματος γίνεται για να βρούμε τυχόν λάθη ή παραλείψεις στη σύνταξη του προγράμματος. Στο περιβάλλον του *Pame* αυτή γίνεται χρησιμοποιώντας την

εντολή **debug** που βρίσκεται στην γραμμή εργαλείων κάνοντας κλικ στο εργαλείο . Πιθανά σφάλματα παρουσιάζονται στη **λίστα σφαλμάτων** που βρίσκεται στο κάτω μέρος του παραθύρου όπως φαίνεται πιο κάτω.



Εικόνα 38 Λίστα σφαλμάτων κατά τη διάρκεια της αποσφαλμάτωσης

Αφού τα λάθη διορθωθούν τότε αρχίζει η βηματική εκτέλεση του προγράμματος που παρουσιάζεται στη **Βηματική Εκτέλεση** που βρίσκεται στο κάτω μέρος του παραθύρου όπως φαίνεται πιο κάτω.



Εικόνα 39 Βηματική εκτέλεση του προγράμματος

7. Μεταγλώττιση (Compile)

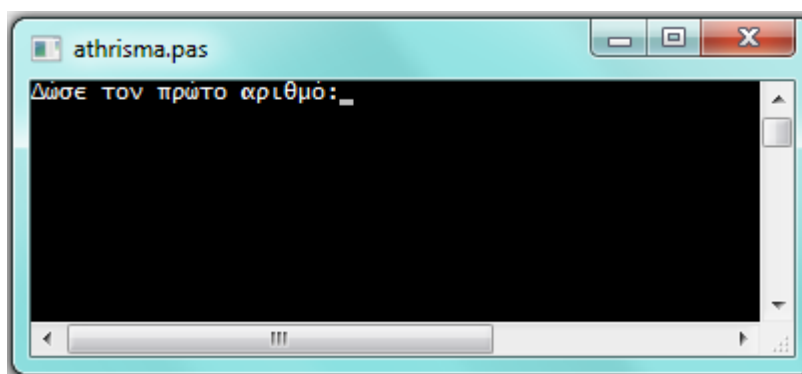
Η μεταγλώττιση του προγράμματος γίνεται χρησιμοποιώντας την εντολή **compile** που βρίσκεται στη γραμμή εργαλείων κάνοντας κλικ στο εργαλείο .

Η εντολή compile μπορεί να χρησιμοποιηθεί και για αποσφαλμάτωση του προγράμματος.

8. Εκτέλεση (Run)

Η εκτέλεση του προγράμματος γίνεται χρησιμοποιώντας την εντολή **run** που βρίσκεται στη γραμμή εργαλείων κάνοντας κλικ στο εργαλείο .

Κατά την εκτέλεση ανοίγει ένα ξεχωριστό παράθυρο στο οποίο γίνεται η πιθανή είσοδος δεδομένων και παρουσιάζεται το αποτέλεσμα του προγράμματος όπως πιο κάτω.



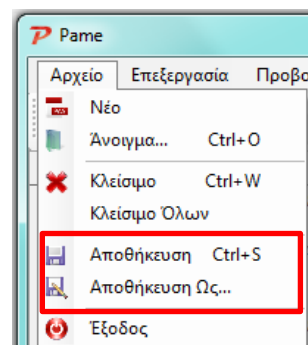
Εικόνα 40 Παράθυρο Run

9. Αποθήκευση (Save, Save as)

- (1) Από το κυρίως μενού του προγράμματος κάνουμε κλικ στο **Αρχείο**.
- (2) Επιλέγουμε την εντολή **Αποθήκευση** ή **Αποθήκευση ως...** αναλόγως με τι θέλουμε να κάνουμε.

ή

Από τη γραμμή εργαλείων επιλέγουμε το εργαλείο .



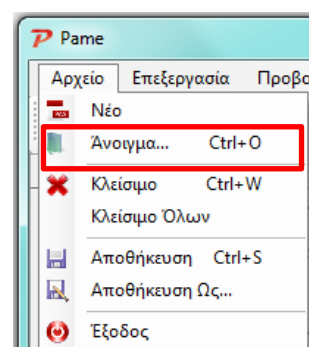
Εικόνα 41 Αποθήκευση προγράμματος

10. Άνοιγμα αρχείου (Open)

- (1) Από το κυρίως μενού του προγράμματος κάνουμε κλικ στο **Αρχείο**.
- (2) Επιλέγουμε την εντολή **Άνοιγμα**.

ή

- (1) Από τη γραμμή εργαλείων επιλέγουμε το εργαλείο .



Εικόνα 42 Άνοιγμα προγράμματος

11. Η εντολή if...then...else

Για να υλοποιήσουμε προβλήματα δομής διακλάδωσης πρέπει να χρησιμοποιήσουμε την εντολή **if...then...else**. Η εντολή ελέγχει μια συνθήκη, αν η συνθήκη είναι αληθής (**true**) τότε εκτελείται μια ομάδα εντολών, διαφορετικά (**false**) εκτελείται μια άλλη ομάδα εντολών.

Σύνταξη εντολής

```
if <συνθήκη> then
begin
    Ομάδα εντολών 1;
end
else
begin
    Ομάδα εντολών 2;
end;
```

Αν οι ομάδες εντολών αποτελούνται **μόνο** από μια εντολή, μπορούμε να παραλείψουμε το **begin** και το **end**

Πάνω από την εντολή **else** δεν βάζουμε ;

Η ομάδα εντολών 2 δεν είναι υποχρεωτική. Μπορεί να έχουμε δομή διακλάδωσης χωρίς **else**.

Παράδειγμα 1

Να δημιουργήσετε ένα πρόγραμμα που να δέχεται δυο ακέραιους αριθμούς και να τυπώνει τον μεγαλύτερο.

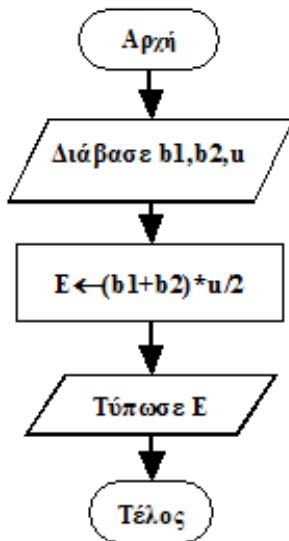
Πρόγραμμα	Επεξήγηση
<pre>program megaliteros; var a,b:integer; begin read(a,b); if a>b then write(a) else write(b); end.</pre>	Δηλώνουμε 2 μεταβλητές a και b. Διαβάζουμε τους 2 αριθμούς. Αν ο a είναι μεγαλύτερος από τον b τότε τύπωσε τον a διαφορετικά τύπωσε τον b Σημειώσεις 1) Στην εντολή που βρίσκεται πάνω από το else απαγορεύεται να βάλουμε ερωτηματικό (;) 2) Δεν χρειάζεται να χρησιμοποιήσουμε το Begin...End; για τις ομάδες εντολές, επειδή έχουμε μόνο μία εντολή σε κάθε ομάδα. 3) Το ερωτηματικό πάνω από την εντολή End είναι προαιρετικό.

12. Μετατροπή Λογικού Διαγράμματος σε Πρόγραμμα

Για να μετατρέψουμε ένα πρόβλημα σε πρόγραμμα, όπως ήδη έχουμε δει, πρέπει πρώτα να ακολουθήσουμε τα στάδια ανάπτυξης και να σχεδιάσουμε το λογικό διάγραμμα που μας δίνει τη σχηματική λύση του προβλήματος. Αν το λογικό διάγραμμα που σχεδιάσαμε είναι σωστό τότε το μόνο που μένει είναι να το μεταφράσουμε στη γλώσσα προγραμματισμού που θέλουμε.

Παράδειγμα 2

Έστω ότι έχουμε το παρακάτω λογικό διάγραμμα. Ακολουθούμε τα παρακάτω βήματα για να το μετατρέψουμε σε πρόγραμμα στη γλώσσα προγραμματισμού Pascal.



Βήμα 1°: Επιλέγουμε ένα όνομα για το πρόγραμμα.

```
program trigono;
```

Βήμα 2°: Εντοπίζουμε τις μεταβλητές του προγράμματος. Από τον καθορισμό του προβλήματος (δεδομένα και πληροφορίες) και το λογικό διάγραμμα διαπιστώνουμε ότι αυτά που «αλλάζουν» (μεταβλητές) είναι τα **b1**, **b2**, **u** και **E**. Οι μεταβλητές μας θα είναι πραγματικοί αριθμοί άρα θα τις δηλώσουμε ως **real**.

```
var
```

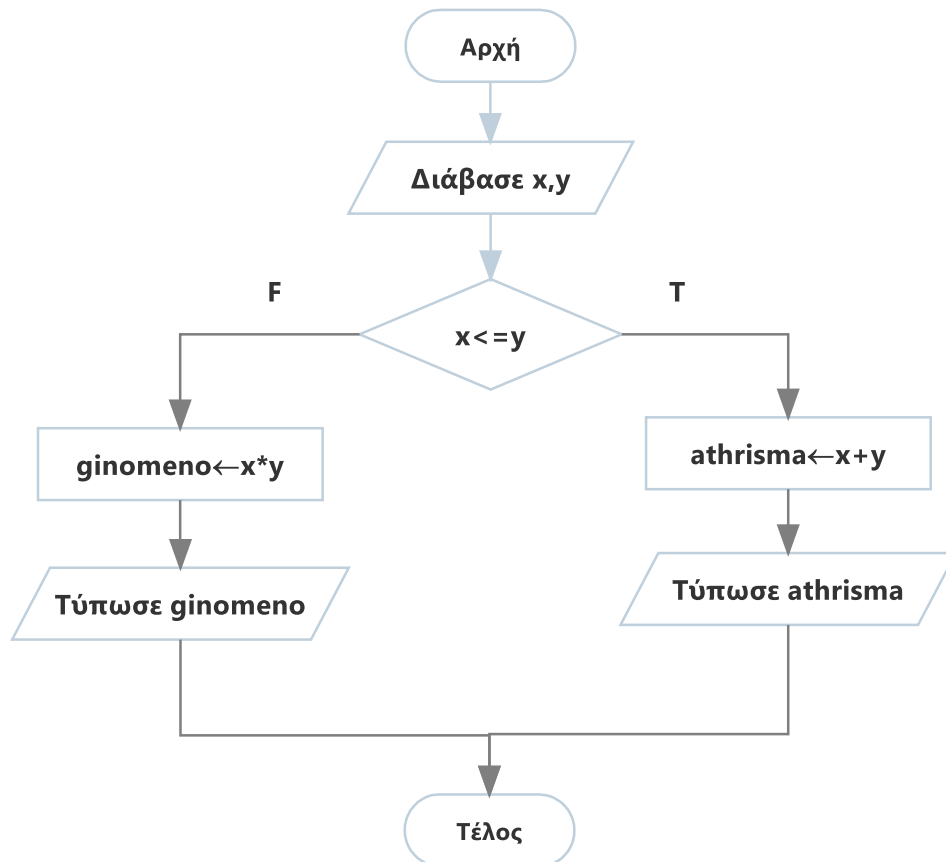
```
  b1,b2,u,E:real;
```

Βήμα 3°: Ξεκινώντας από την αρχή μετατρέπουμε το κάθε σχήμα σε εντολή, όπως φαίνεται στον πιο κάτω πίνακα.

Σχήμα	Εντολή
Αρχή	<code>begin</code>
Διάβασε b1,b2,u	<code>readln(b1,b2,u);</code>
$E \leftarrow (b1+b2) * u / 2$	<code>E := (b1+b2) * u / 2</code>
Τύπωσε E	<code>writeln(E);</code>
Τέλος	<code>end.</code>

Παράδειγμα 3

Να διαβαστούν δύο αριθμοί. Αν ο πρώτος αριθμός είναι μικρότερος ή ίσος με τον δεύτερο τότε να υπολογίζεται το άθροισμά τους και να τυπώνεται, αλλιώς να υπολογίζεται το γινόμενό τους και να τυπώνεται.



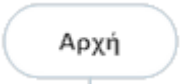
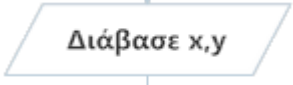
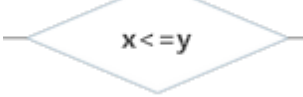
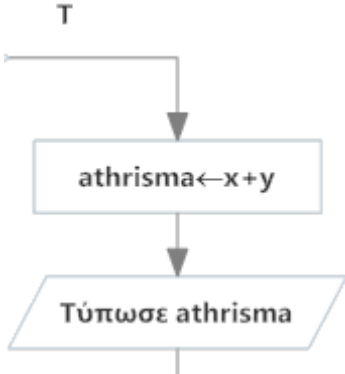
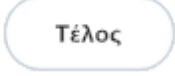
Βήμα 1°: Επιλέγουμε ένα όνομα για το πρόγραμμα.

```
program paradeigma;
```

Βήμα 2°: Εντοπίζουμε τις μεταβλητές του προγράμματος. Έχουμε 4 μεταβλητές (οι 2 αριθμοί, το άθροισμα και το γινόμενο). Οι μεταβλητές μας είναι ακέραιοι αριθμοί.

```
var
  a,b:integer;
  athrisma,ginomeno:integer;
```

Βήμα 3°: Ξεκινώντας από την αρχή μετατρέπουμε το κάθε σχήμα σε εντολή, όπως φαίνεται στον πιο κάτω πίνακα.

Σχήμα	Εντολή
	<code>begin</code>
	<code>read(x,y)</code>
	<code>if x<=y then</code>
	<pre> begin athrisma:=x+y; writeln(athrisma) end </pre>
	<pre> Else begin ginomeno:=x*y; writeln(ginomeno) end; </pre>
	<code>end.</code>

Βασικές Έννοιες

Program:	Δεσμευμένη λέξη της Pascal ακολουθούμενη από ένα αναγνωριστικό που αντιπροσωπεύει το όνομα του προγράμματος και αποτελεί την επικεφαλίδα.
Σταθερά:	Θέση μνήμης που χρησιμοποιούμε για να αναθέσουμε μια τιμή την οποία χρησιμοποιούμε σε όλη την έκταση του προγράμματος, η οποία όμως δεν αλλάζει κατά την εκτέλεση του προγράμματος.
Μεταβλητή:	Θέση μνήμης που χρησιμοποιούμε για να αναθέσουμε μια τιμή, η οποία όμως μπορεί να αλλάζει κατά τη διάρκεια εκτέλεσης του προγράμματος.
Const:	Δεσμευμένη λέξη της Pascal την οποία χρησιμοποιούμε για να ορίσουμε μια σταθερά.Π.χ. <pre>const vat = 0.17;</pre>
Var:	Δεσμευμένη λέξη της Pascal την οποία χρησιμοποιούμε για να ορίσουμε μια μεταβλητή.Π.χ. <pre>var number1 : integer;</pre>
Integer:	Τύπος δεδομένων ο οποίος αντιπροσωπεύει τους ακέραιους αριθμούς.
Real:	Τύπος δεδομένων ο οποίος αντιπροσωπεύει τους πραγματικούς αριθμούς.
String:	Τύπος δεδομένων ο οποίος δέχεται αλφαριθμητικά δεδομένα.
Readln:	Εντολή για είσοδο δεδομένων. Π.χ. <pre>readln(ypsos) ;</pre>
Writeln:	Εντολή για έξοδο δεδομένων. Π.χ. <pre>writeln(embadon) ;</pre>
:=	Σύμβολο εντολής εκχώρησης τιμής (επεξεργασία, υπολογισμός). Η τιμή της έκφρασης στο δεξί μέρος αποθηκεύεται στη μεταβλητή στο αριστερό μέρος. Π.χ. <pre>a := b*(c+d) ;</pre>
If-then-else:	Εντολή η οποία αναπαριστά τη δομή διακλάδωσης. Π.χ. <pre>if a>b then begin c := a+b; d := 4*a end else begin c := a-b; d := 4*b end;</pre>

Υπόμνημα

Λειτουργία	Γραμμή Εργαλείων	Μενού
Εκκίνηση του προγράμματος Pame	διπλό κλικ στο 	Αρχείο⇒Νέο
Νέο αρχείο	κλικ στο 	Αρχείο⇒Νέο
Αποθήκευση αρχείου	κλικ στο 	Αρχείο⇒Αποθήκευση Αρχείο⇒Αποθήκευση Ως...
Άνοιγμα αρχείου	κλικ στο 	Αρχείο⇒Άνοιγμα
Μεταγλώττιση (compile)	κλικ στο 	Πρόγραμμα⇒Μεταγλώττιση
Αποσφαλμάτωση (debugging)	κλικ στο 	Πρόγραμμα⇒Εκκίνηση Βηματικής Εκτέλεσης
Εκτέλεση (run)	κλικ στο 	Πρόγραμμα⇒Εκτέλεση

Πηγές

1. Σταυρινίδης Ε., Σωτικόπουλος Κ. (2005) *Ηλεκτρονικοί Υπολογιστές Τόμος Β (PASCAL)*, σελ. 29—71, Υπουργείο Παιδείας Και Πολιτισμού.
2. Γιώργος Σαματζίδης, *Pame Pascal Made Easy*, <http://users.sch.gr/samartzidis/pame/>.
3. ΚΕΝΤΡΟ ΠΛΗ.ΝΕ.Τ. ΠΕΡ. ΕΝΟΤΗΤΑΣ ΦΛΩΡΙΝΑΣ, <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-Pascal.html>