ΚΕΦΑΛΑΙΟ 1Ο ΕΙΣΑΓΩΓΗ

1. **Ποιους τομείς περιλαμβάνει η Επιστήμη των Υπολογιστών;**

* Θεωρητική Επιστήμη Υπολογιστών
* Εφαρμοσμένη Επιστήμη Υπολογιστών

1. **Με τι ασχολείται η Θεωρητική Επιστήμη Υπολογιστών;**

Ασχολείται με υπολογιστικές μεθόδους και αλγόριθμους με στόχο την συλλογή, επεξεργασία ανάλυση και αποθήκευση των πληροφοριών.

* Ανάλυση αλγορίθμων
* Θεωρία Υπολογισιμότητας
* Θεωρία Πολυπλοκότητας
* Θεωρία Γλωσσών Προγραμματισμού

1. **Με τι ασχολείται η** **Εφαρμοσμένη Επιστήμη Υπολογιστών;**

Ασχολείται με την εφαρμογή της θεωρίας των Υπολογιστών στην λύση πραγματικών προβλημάτων.

* Σχεδιασμός υλικού για Η/Υ πχ. Δίσκοι, επεξεργαστές, κλπ
* Ανάπτυξη Λογισμικού πχ. Λειτουργικά συστήματα, διάφορα προγράμματα
* Σχεδιασμός πληροφοριακών συστημάτων για συλλογή, επεξεργασία και αποθήκευση δεδομένων
* Τεχνητή νοημοσύνη
* Βάσεις δεδομένων
* Δίκτυα υπολογιστών
* Ασφάλεια υπολογιστών και πληροφοριών

ΚΕΦΑΛΑΙΟ 2Ο ΠΡΟΒΛΗΜΑ (2.1.1, 2.1.2, 2.1.3)

**1. Τι είναι πρόβλημα;**

**Πρόβλημα** είναι κάθε ζήτημα που πρέπει να σκεφτούμε για να λύσουμε.

Πχ. Ο Όμηρος στην Οδύσσεια περιγράφει τα προβλήματα που αντι-

μετώπιζε ο Οδυσσέας για να φτάσει στην Ιθάκη.

 Το πρόβλημα που κλήθηκε να επιλύσει ο Αρχιμήδης με τη βασιλική

κορώνα που οδήγησε στη γνωστή φράση του «Εύρηκα-Εύρηκα».

 Τα προβλήματα επιδημιών στην ανθρωπότητα και η αντιμετώπι-

σή τους με εμβόλια.

 Το πρόβλημα της μόλυνσης του περιβάλλοντος

**2. Ποιες κατηγορίες προβλημάτων υπάρχουν ανάλογα με την λύση τους;**

Α) **Επιλύσιμα** είναι εκείνα τα προβλήματα για τα οποία υπάρχει λύση.

Πχ. η επίλυση της δευτεροβάθμιας εξίσωσης

Β) **Μη επιλύσιμα** χαρακτηρίζονται εκείνα τα προβλήματα τα οποία ότι δεν λύνονται.

Πχ. Να γυρίσουμε τον χρόνο πίσω, δηλ. να γίνουμε πάλι παιδιά

Γ) **Ανοικτά** ονομάζονται τα προβλήματα για τα οποία η λύση τους

δεν έχει ακόμα βρεθεί, αλλά ίσως λυθούν στο μέλλον.

Πχ. Πρόβλεψη σεισμών, θεραπεία όλων των μορφών καρκίνου

**3.Τι ονομάζεται υπολογιστικό πρόβλημα;**

*Οποιοδήποτε πρόβλημα μπορεί να λυθεί και μέσω του υπολογιστή, χαρακτηρίζεται* ***υπολογιστικό*** *πρόβλημα.*

Παραδείγματα υπολογιστικών προβλημάτων είναι:

 Η επίλυση της δευτεροβάθμιας εξίσωσης.

 Η ταξινόμηση των μαθητών σε αλφαβητική σειρά.

 Η αναζήτηση και ο υπολογισμός της χιλιομετρικά συντομότερης διαδρομής που θα κάνει ένας ταχυδρόμος για να επισκεφθεί δέκα χωριά και να επιστρέψει στο χωριό από όπου ξεκίνησε περνώντας μόνο μία φορά από κάθε χωριό, με βάση έναν δεδομέ-

νο χάρτη των χωριών και των δρόμων που συνδέουν τα χωριά.

 Η εύρεση λέξης που να ξεκινά από ένα γράμμα και να τελειώνει σε ένα άλλο γράμμα.

**Παράδειγμα 2.3.** Δίνεται ο ακόλουθος χάρτης διαδρομών που συνδέει ορισμένες πόλεις. Ο χάρτης δείχνει το χρόνο που απαιτείται για τη μετακίνηση από πόλη σε πόλη. Ποια διαδρομή είναι η συντομότερη από την πόλη Α στην πόλη Β;

**Κεφ. 2, 2.1.4 Διαδικασίες επίλυσης υπολογιστικού προβλήματος**

**1. Ποια είναι η βασική προϋπόθεση για την επίλυση ενός προβλήματος;**

Βασική προϋπόθεση είναι α) η σωστή διατύπωσή του από τον δημιουργό του και β) σωστή κατανόηση από αυτόν που θα το λύσει.

**2. Ποια είναι τα στάδια επίλυσης ενός προβλήματος;**

* Κατανόηση
* Ανάλυση
* Σύνθεση
* Κατηγοριοποίηση
* Γενίκευση

**2α. Τι είναι κατανόηση ενός προβλήματος;**

**Κατανόηση** είναι να καταλάβουμε ποια είναι τα δεδομένα του προβλήματος και ποια είναι τα ζητούμενα, δηλαδή τι πρέπει να βρούμε.

**2β. Τι είναι η ανάλυση- αφαίρεση ενός προβλήματος;**

**Ανάλυση** ενός προβλήματος είναι μοίρασμά του σε μικρότερα προβλήματα. Αφαίρεση είναι η διάκριση των κύριων σημείων από τα δευτερεύοντα.

**2γ. Τι είναι η σύνθεση;**

**Σύνθεση** είναι η ένωση των λύσεων των μικρότερων προβλημάτων έτσι ώστε να λύσουμε το αρχικό.

**2δ. Τι είναι η κατηγοριοποίηση;**

**Κατηγοριοποίηση** είναι η ένταξη του προβλήματος σε μια κατηγορία παρόμοιων προβλημάτων έτσι ώστε να το λύσουμε ευκολότερα.

**2ε. Τι είναι η γενίκευση;**

**Γενίκευση** είναι να μπορέσουμε να χρησιμοποιήσουμε την λύση που βρίκαμε για ένα πρόβλημα για να λύσουμε και άλλα παρόμοια.

**3. Τι είναι η επεξεργασία δεδομένων;**

**Επεξεργασία δεδομένων** είναι η συστηματική εκτέλεση πράξεων στα δεδομένα.

**4. Με ποιους τρόπους μπορούμε να αναλύσουμε ένα πρόβλημα;**

* **Φραστική ανάλυση**
* **Διαγραμματική ανάλυση**

Παράδειγμα: Να διερευνηθεί η εξίσωση αx + β = 0 ως προς x, για

τις διάφορες τιμές του α και β.

**Απάντηση (**Φραστική ανάλυση)

Υπάρχουν 2 περιπτώσεις: Αν ο συντελεστής της μεταβλητής x είναι δι-

άφορος του μηδενός (α ≠ 0) ή αν ο συντελεστής της μεταβλητής x εί-

ναι ίσος με μηδέν (α = 0).

Περίπτωση 1: Αν α ≠ 0, τότε η εξίσωση έχει

μοναδική λύση την x =-β/α

Περίπτωση 2: Αν α = 0 τότε υπάρχουν δύο υποπεριπτώσεις: Αν ο στα-

θερός όρος β είναι διάφορος του μηδενός (β ≠ 0) ή αν εί-

ναι ίσος με μηδέν (β = 0).

Περίπτωση 2.1. Αν β ≠ 0, η εξίσωση είναι αδύνατη.

Περίπτωση 2.2. Αν β = 0, η εξίσωση είναι αόριστη.

Άσκηση : Προσπαθήστε να αναλύσετε διαγραμματικά το παραπάνω πρόβλημα.

**ΚΕΦΑΛΑΙΟ 2: Αλγόριθμοι**

**2.2.1**

**Τι είναι αλγόριθμος;**

**Αλγόριθμος** είναι μια σειρά καθορισμένων ενεργειών που μπορούν να γίνουν σε συγκεκριμένο χρόνο και έχουν στόχο την λύση ενός προβλήματος.

Παράδειγμα: 1ο Αλγόριθμος δεσίματος γραβάτας

****

**2.2.2**

**Ποιες είναι οι ιδιότητες ενός αλγορίθμου;**

**Καθοριστικότητα:** Κάθε εντολή ενός αλγορίθμου χρειάζεται να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της.

**Περατότητα:** Κάθε αλγόριθμος πρέπει να τελειώνει μετά από συγκεκριμένα βήματα εκτέλεσης των εντολών του.

**Αποτελεσματικότητα:** Κάθε εντολή ενός αλγορίθμου χρειάζεται να είναι απλή και κατανοητή, ώστε να μπορεί να εκτελεστεί επακριβώς και σε συγκεκριμένο χρόνο.

**Είσοδος:** Κάθε αλγόριθμος πρέπει να παίρνει δεδομένα πχ. Κάποιους αριθμούς.

**Έξοδος:** Κάθε αλγόριθμος πρέπει να δημιουργεί κάποιο αποτέλεσμα.

Παράδειγμα:

**2.2.5 ΑΝΑΠΑΡΑΣΤΑΣΗ ΑΛΓΟΡΙΘΜΟΥ**

**Με ποιους τρόπους μπορεί να γίνει η αναπαράσταση ενός αλγορίθμου;**

Α. Με **Φυσική Γλώσσα** περιγράφουμε τα βήματα για την λύση του προβλήματος. Συνήθως υπάρχουν ασάφειες.

Β. Με **Ψευδογλώσσα** δηλαδή μια υποθετική γλώσσα που έχει στοιχεία από γλώσσες προγραμματισμού, χωρίς πολλές λεπτομέρειες.

Γ. Με **Γλώσσα Προγραμματισμού (**είτε **οπτική** είτε **κειμενική)** που είναι μια τεχνητή γλώσσα που καταλαβαίνει ο υπολογιστής.

Δ. **Διάγραμμα ροής** που τα βήματα περιγράφονται με σχήματα.

Πχ. Να αναπτυχθεί αλγόριθμος με φυσική γλώσσα, με διάγραμμα ροής και με ψευδογλώσσα, ο οποίος θα διαβάζει τις τιμές δύο μεταβλητών και θα αντιμεταθέτει το περιεχόμενό τους.

Στη συνέχεια θα εμφανίζει ως αποτέλεσμα το περιεχόμενο των μεταβλητών μετά την αντιμετάθεση.

Να εκτελεστεί ο αλγόριθμος για τις τιμές 8 και 12.

Απάντηση

**Φυσική γλώσσα**: Αφού εισαχθούν οι τιμές δύο μεταβλητών α και β, να δώσετε το περιεχόμενο της μεταβλητής α και σε μία νέα μεταβλητή temp (προσωρινή). Στη συνέχεια, να δώσετε το περιεχόμενο της μεταβλητής β στη μεταβλητή α και τέλος να δώσετε το περιεχόμενο της μεταβλητής temp και στη μεταβλητή β.

**Ψευδογλώσσα** :

1 . Αλγόριθμος Αντιμετάθεση

2. Διάβασε α, β

3. temp ← α

4. α ← β

5. β ← temp

6. Εμφάνισε α, β

7. Τέλος Αντιμετάθεση

**Διάγραμμα ροής**

**2.2.6 ΔΕΔΟΜΕΝΑ ΚΑΙ ΑΝΑΠΑΡΑΣΤΑΣΗ ΔΕΔΟΜΕΝΩΝ**

**1.Τι είναι τα δεδομένα;**

Τα δεδομένα τα δίνουμε ως είσοδο σε έναν αλγόριθμο, εκεί γίνεται η επεξεργασία τους, δηλαδή πράξεις σε συγκεκριμένο αριθμό βημάτων και παίρνουμε ως έξοδο τα αποτελέσματα που θέλουμε.

***Ο ΑΛΓΟΡΙΘΜΟΣ ΜΕΤΑΤΡΕΠΕΙ ΤΑ ΔΕΔΟΜΕΝΑ ΣΕ ΠΛΗΡΟΦΟΡΙΕΣ***

**2. Ποιοι είναι οι πιο συνηθισμένοι τύποι δεδομένων;**

* Ακέραιος τύπος, π.χ. -1234567890
* Πραγματικός τύπος, π.χ. 3.14
* Αλφαριθμητικός τύπος, π.χ. "Ν" ή 'Μυρτώ'
* Λογικός τύπος, δηλαδή ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ

*Σε κάθε τύπο δεδομένων γίνονται διαφορετικές πράξεις*

**3. Τι μορφή μπορούν να έχουν τα δεδομένα;**

Τα δεδομένα μπορούν να είναι απλές μεταβλητές πχ. Α=2, Η=3.56 δηλ. να παίρνουν μία τιμή κάθε φορά, ή να αποθηκεύονται ως μια **δομή δεδομένων.**

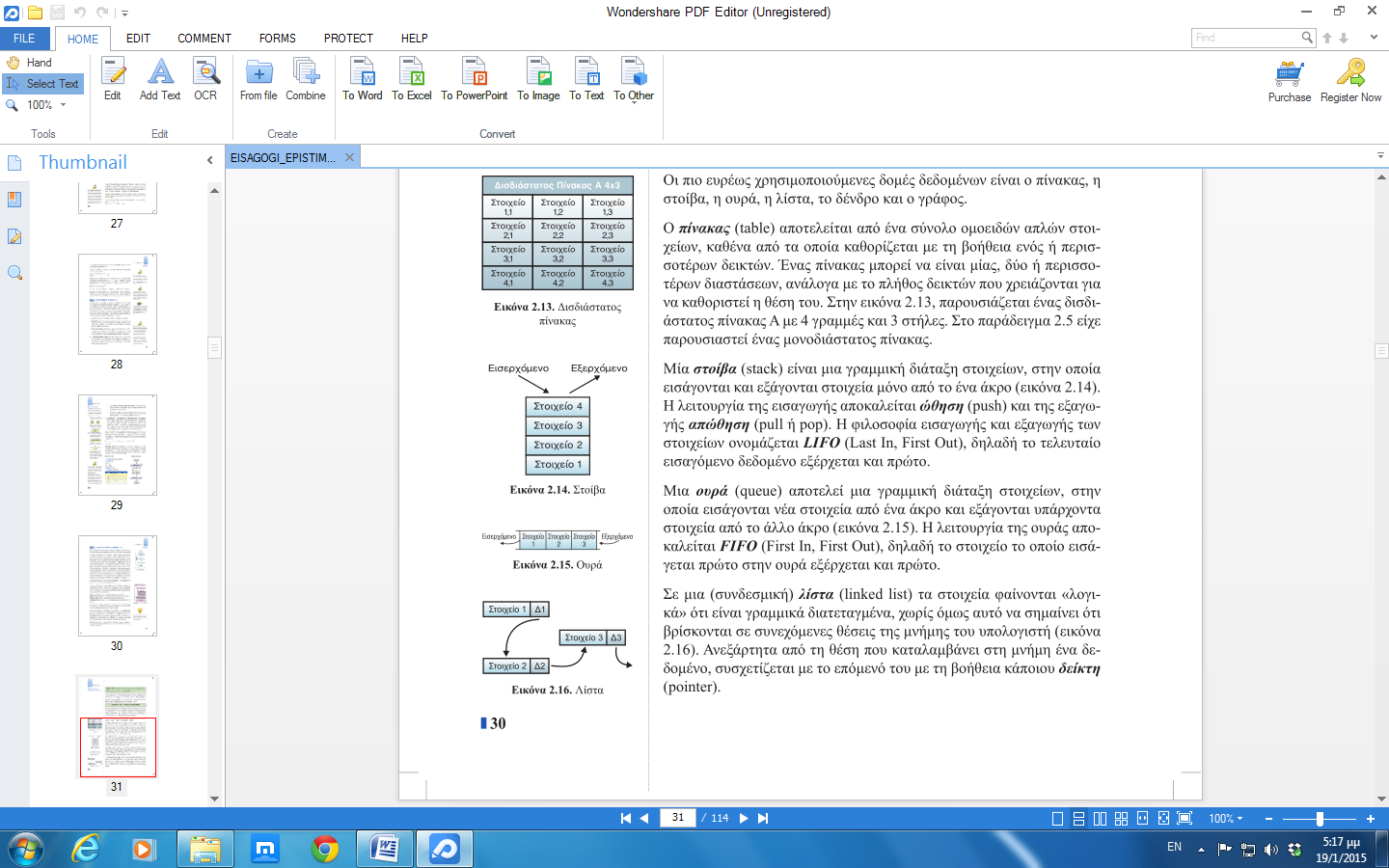
**4. Τι είναι δομή δεδομένων;**

Δομή δεδομένων είναι ένα οργανωμένο σύνολο δεδομένων.

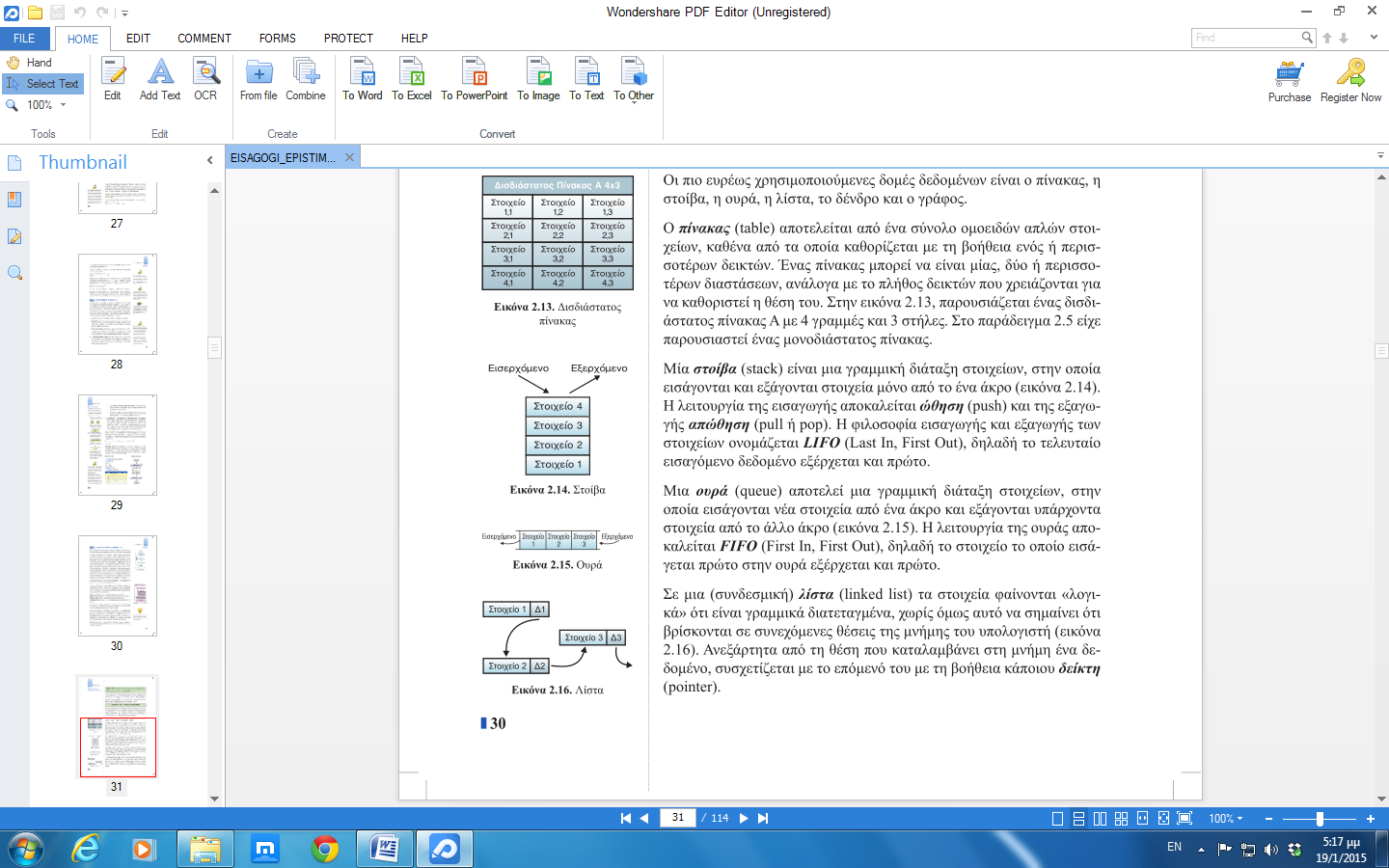
***ΑΛΓΟΡΙΘΜΟΙ + ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ=ΠΡΟΓΡΑΜΜΑΤΑ***

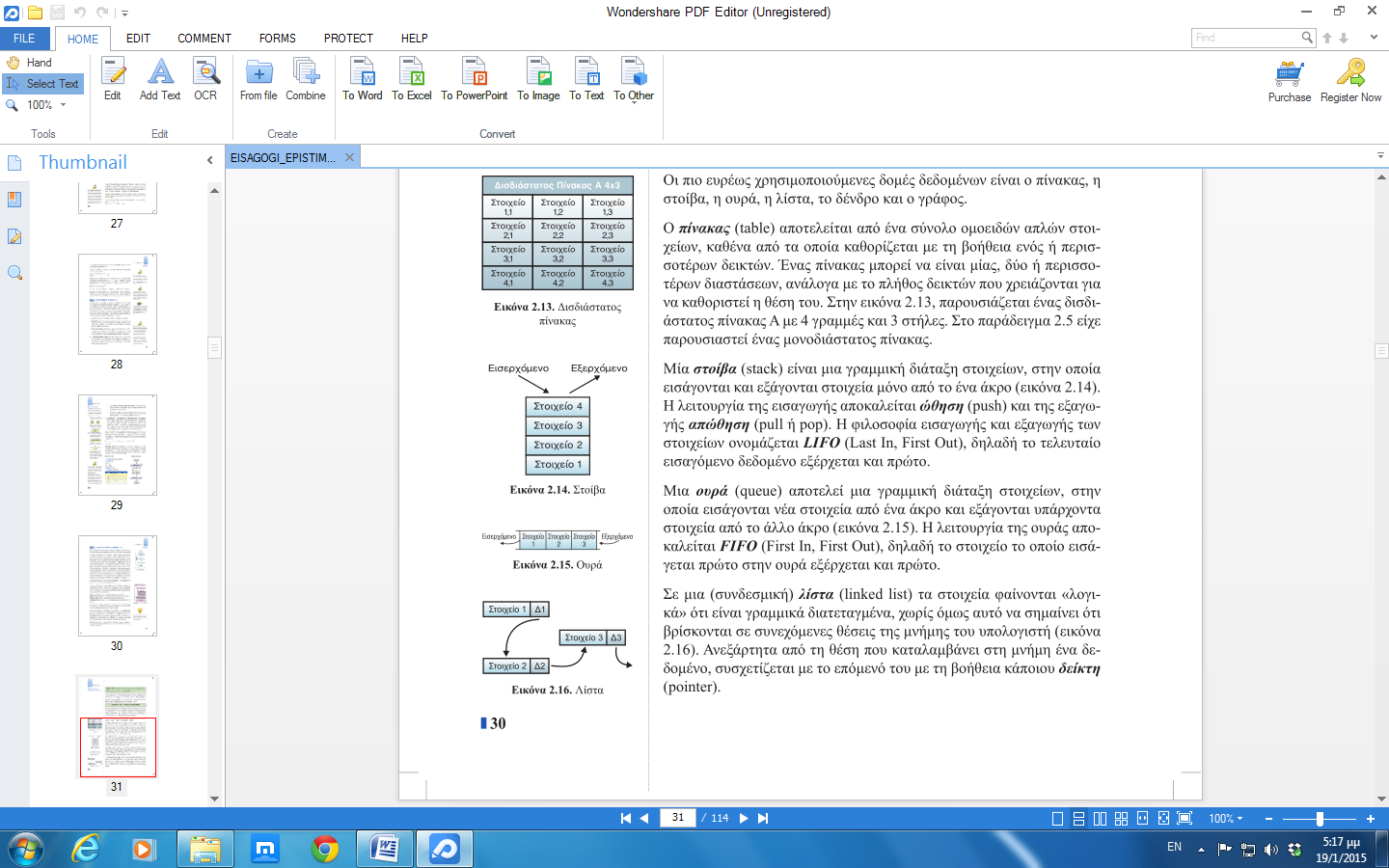
**5. Ποιες δομές δεδομένων υπάρχουν;**

* **Πίνακας :** Αποτελείται από ομοειδή στοιχεία τοποθετημένα σε γραμμές και στήλες.

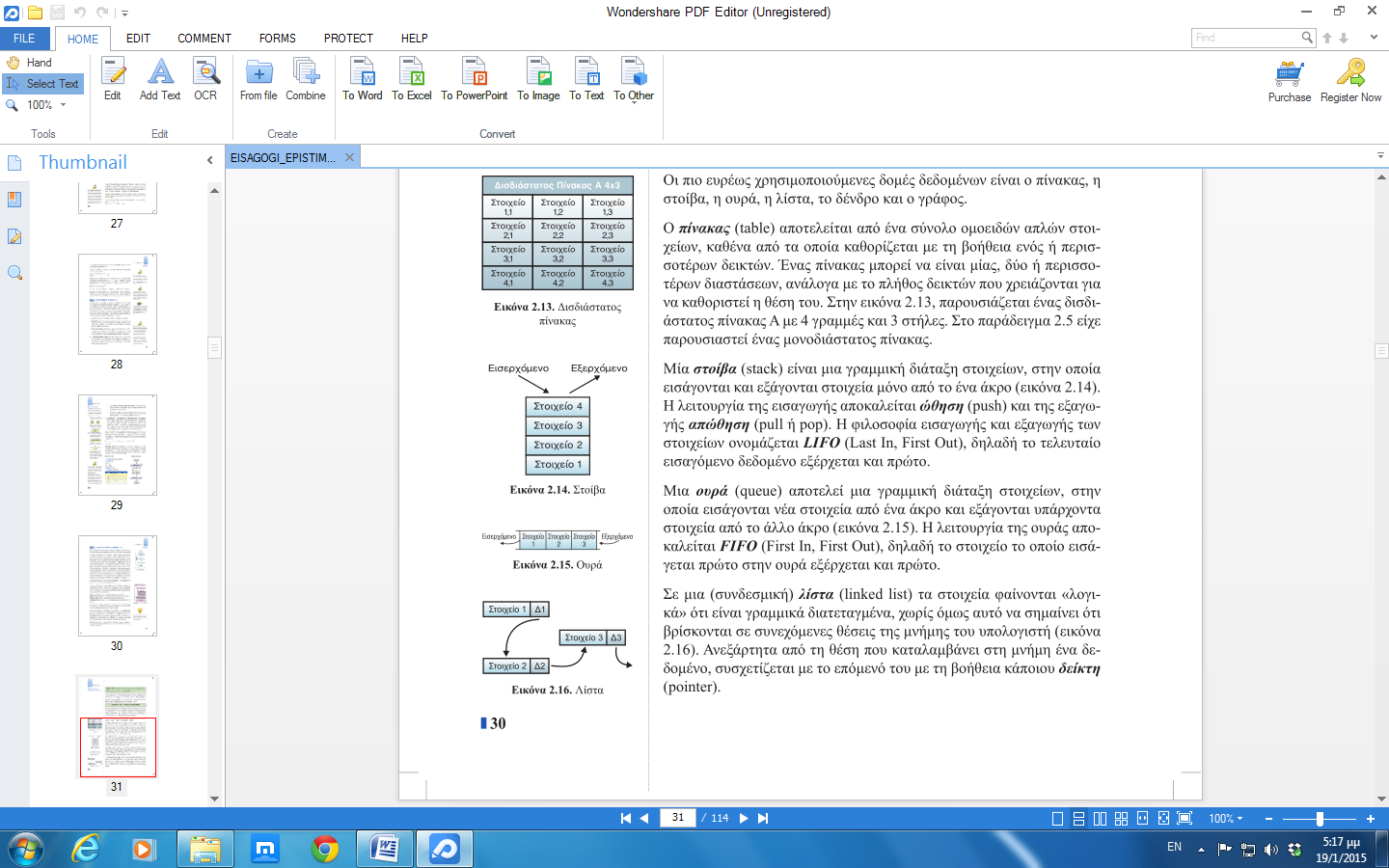


1. **Στοίβα:** Αποτελείται από μια σειρά στοιχείων που μπορούμε να προσθέσουμε (ώθηση) ή να αφαιρέσουμε (απώθηση) στοιχεία από το ένα άκρο μόνο (LIFO)

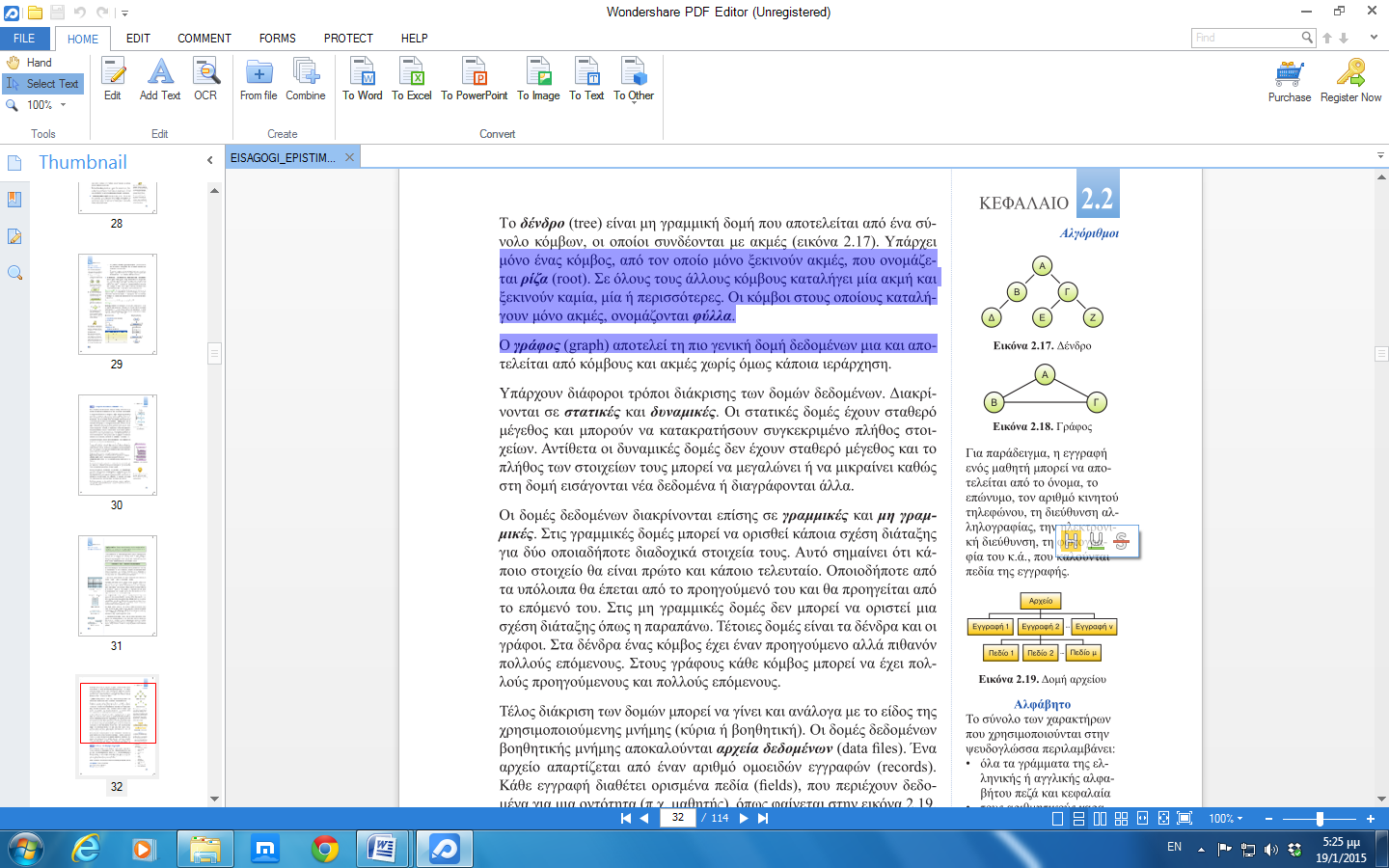


1. **Ουρά:** Αποτελείται από μια σειρά στοιχείων που προσθέτουμε νέα στοιχεία από το ένα άκρο και αφαιρούμε από το άλλο άκρο (FIFO)

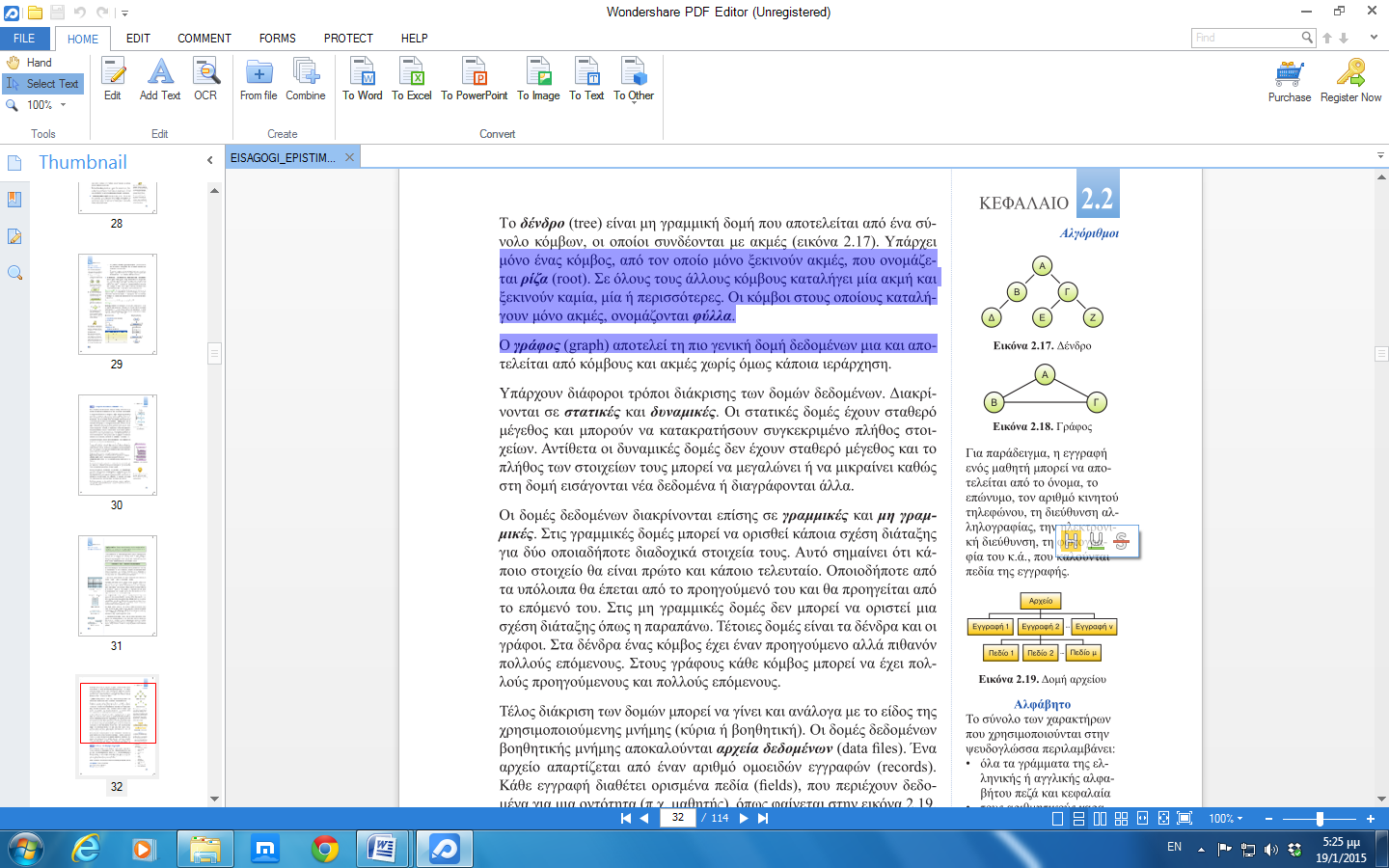
**Λίστα:** Αποτελείται από στοιχεία που δεν είναι πάντα αποθηκευμένα σε συνεχόμενες θέσεις μνήμης. Κάθε στοιχείο όμως ¨δείχνει ¨ το επόμενό του με τη βοήθεια κάποιου δείκτη.

****

1. **Δένδρο:** αποτελείται από κόμβους που συνδέονται με γραμμές. Ο κόμβος από τον οποίο ξεκινούν μόνο οι γραμμές λέγεται **ρίζα** και οι κόμβοι όπου καταλήγουν μόνο γραμμές ονομάζονται **φύλλα**.

****

1. **Γράφος:** Αποτελείται από κόμβους και γραμμές χωρίς όμως σειρά.

****

**6. Τι είναι α) στατικές, β) δυναμικές δομές;**

**Σταθερές** λέμε τις δομές που μπορούν να αποθηκεύσουν συγκεκριμένο αριθμό στοιχείων πχ. Πίνακας

**Δυναμικές** λέμε τις δομές που το μέγεθός τους αλλάζει ανάλογα αν προθθέτουμε ή αφαιρούμε στοιχεία. Πχ. Ουρά, στοίβα.

**7. Ποιες δομές λέγονται γραμμικές και ποιες μη γραμμικές;**

**Γραμμικές** ονομάζουμε τις δομές που τα στοιχεία τους βρίσκονται σε κάποια σειρά. Πχ. Πίνακας, στοίβα, ουρά και λίστα.

**Μη γραμμικές** ονομάζουμε τις δομές που τα στοιχεία τους δεν βρίσκονται σε κάποια σειρά. Πχ. Δένδρα και γράφοι.

**8. Τι είναι τα αρχεία δεδομένων;**

Είναι αρχεία αποθηκευμένα σε δίσκους , CD, DVD. Το καθένα αποτελείται από εγγραφές και κάθε εγγραφή αποτελείται από πεδία τα οποία περιγράφουν μια οντότητα πχ. Μαθητής, καθηγητής.

**2.2.7 ΕΝΤΟΛΈΣ ΚΑΙ ΔΟΜΕΣ ΑΛΓΟΡΙΘΜΩΝ**

Κάθε αλγόριθμος σε ψευδογλώσσα ξεκινά με τη γραμμή

**Αλγόριθμος** όνομα\_αλγορίθμου

και τελειώνει με τη γραμμή

**Τέλος** όνομα\_αλγορίθμου

Επεξηγηματικά σχόλια μπορούν να γράφονται οπουδήποτε στο σώμα του αλγορίθμου. Ένα σχόλιο αρχίζει με το χαρακτήρα θαυμαστικό (!)

***2.2.7.1 Εκχώρηση, Είσοδος και Έξοδος τιμών***

Η γενική μορφή της εντολής εκχώρησης είναι:

**Μεταβλητή ← Έκφραση**

**Πχ. Κ<- 3+2-Α**

Η είσοδος τιμών επιτυγχάνεται και με τις εντολές εισόδου.

**Πχ. Διάβασε**  α,α1,β1

Για την έξοδο τιμών (αποτελεσμάτων) μπορούν να χρησιμοποιηθούν οι εντολές **Γράψε, Εμφάνισε.**

Π.χ. **Εμφάνισε**''Τιμή:'', αξία.

***2.2.7.2 Δομή ακολουθίας [οπτικοποίηση](http://photodentro.edu.gr/v/item/ds/8521/10555)***

Η δομή ακολουθίας χρησιμοποιείται για την αντιμετώπιση προβλημάτων στα οποία οι εντολές εκτελούνται η μία μετά την άλλη από πάνω προς τα κάτω.

**Παράδειγμα 2.8.** **Είσοδος και έξοδος αριθμών [οπτικοποίηση](http://photodentro.edu.gr/v/item/ds/8521/10520)**

Να διαβαστούν δύο αριθμοί και να υπολογιστεί και να εμφανιστεί το άθροισμά τους.

**Αλγόριθμος** Άθροισμα

**Διάβασε** α, β

Σ ← α + β

**Εμφάνισε** Σ

**Τέλος** Άθροισμα

**Αλγόριθμος** Άθροισμα\_2

**Δεδομένα** // α, β //

Σ ← α + β

**Αποτελέσματα** // Σ //

**Τέλος** Άθροισμα\_2

#### *2.2.7.3 Δομή επιλογής*

Η διαδικασία επιλογής περιλαμβάνει τον έλεγχο μιας συνθήκης που μπορεί να έχει δύο τιμές (Αληθής ή Ψευδής) και ακολουθεί η απόφαση εκτέλεσης εντολών με βάση την τιμή αυτής της συνθήκης

Α) **Απλή εντολή επιλογής**

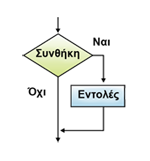
**Αν** Συνθήκη **τότε**

     Εντολές

**Τέλος\_αν**

Αν η συνθήκη είναι αληθής, τότε εκτελούνται οι εντολές.

Οι εντολές μπορούν να είναι μία ή περισσότερες.



**Πχ.**

Β) **Σύνθετη εντολή επιλογής**

**Αν** Συνθήκη **τότε**

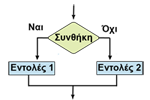
     Εντολές\_1

**αλλιώς**

     Εντολές\_2

**Τέλος\_αν**

Αν η συνθήκη είναι αληθής, τότε εκτελούνται οι εντολές 1, αλλιώς (δηλαδή αν η συνθήκη είναι ψευδής) εκτελούνται οι εντολές 2.



πχ.

#### *2.2.7.4 Δομή επανάληψης*

Την χρησιμοποιούμε όταν μια σειρά εντολών να επαναληφθεί πολλές φορές.

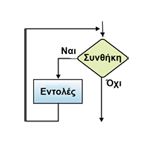
Α) **Εντολή  
Όσο ... επανάλαβε**

**Όσο** Συνθήκη **επανάλαβε**

     Εντολές

**Τέλος\_επανάληψης**

Εκτελούνται οι εντολές όσο η συνθήκη είναι αληθής



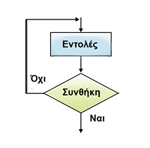
Β) **Εντολή  
Επανάλαβε ... Μέχρις\_ότου**

**Επανάλαβε**

     Εντολές

**Μέχρις\_ότου** Συνθήκη

Εκτελούνται οι εντολές μέχρις ότου η συνθήκη γίνει αληθής.



Γ) **Εντολή Για ... από ... μέχρι**

**Για** μεταβλητή **από** τ1 **μέχρι** τ2 [**με\_βήμα** β]

     Εντολές

**Τέλος\_επανάληψης**

**Παράδειγμα.** Να φτιάξετε αλγόριθμο ο οποίος θα διαβάζει 4 αριθμούς και θα υπολογίζει και θα εμφανίζει το άθροισμά τους. [προσομοίωση](http://photodentro.edu.gr/v/item/ds/8521/10581)

**Αλγόριθμος** Άθροισμα\_Αριθμών

Σ ← 0

**Για** i **από** 1 **μέχρι** 4

**Διάβασε** α

     Σ ← Σ + α

**Τέλος\_επανάληψης**

**Εμφάνισε** ''Άθροισμα:'', Σ

**Τέλος** Άθροισμα\_Αριθμών

### 2.2.9 Διόρθωση λογικών λαθών σε αλγοριθμο

Ο αλγοριθμος εκτελείται αλλά μας δίνει λάθος αποτελέσματα. Εμείς βρίσκουμε που είναι αυτά τα λάθη και τα διορθώνουμε.

πχ. ο παρακάτω αλγόριθμος βρίσκει άθροισμα; Χρειάζεται διόρθωση

**Αλγόριθμος** Άθροισμα

**Διάβασε** α, β

Σ ← α - β

**Εμφάνισε** Σ

**Τέλος** Άθροισμα

**3.2.1 Πρόγραμμα** είναι το σύνολο των εντολών που χρειάζεται να δοθούν στον υπολογιστή, ώστε να υλοποιηθεί ο αλγόριθμος για την επίλυση του προβλήματος.

Η εργασία σύνταξης των προγραμμάτων σε κάποια γλώσσα προγραμματισμού ονομάζεται ***προγραμματισμός*** και τα άτομα που γράφουν και συντάσσουν ένα πρόγραμμα ονομάζονται ***προγραμματιστές***.

Οι γλώσσες προγραμματισμού αναπτύχθηκαν με σκοπό την επικοινωνία του ανθρώπου (προγραμματιστή) με τη μηχανή (υπολογιστή). Ο υπολογιστής πράξεις σε σειρές των δύο ψηφίων 0 και 1 (***δυαδικά ψηφία***, bits), με μεγάλη ταχύτητα.

1. Αρχικά τα προγράμματα γράφονταν σε ***γλώσσα μηχανής***,

2. Στη συνέχεια αναπτύχθηκαν οι ***συμβολικές γλώσσες***

3. Τέλος δημιουργήθηκαν ***γλώσσες υψηλού επιπέδου διότι:***

* υπάρχει δυνατότητα μεταφοράς, εκτέλεσης των προγραμμάτων σε οποιοδήποτε υπολογιστή.
* είναι εύκολη η εκμάθηση, η διόρθωση των λαθών και η συντήρηση των προγραμμάτων.

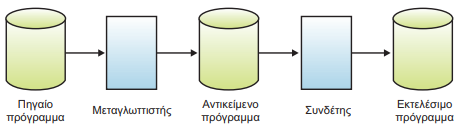
**Τέτοιες γλώσσες είναι**:

* **FORTRAN,** **COBOL,** **LISP,** **PROLOG,**  **BASIC,**  **C++,** **JAVA,**. **SCRATCH , BYOB,**  **Google AppInvertor,**  **LOGO** ,  **GameMaker** ,  **LabView**

### 2.3.2 Σχεδίαση και συγγραφή κώδικα

Για τη σύνταξη του***πηγαίου προγράμματος*** χρησιμοποιείται ένα ειδικό πρόγραμμα το οποίο ονομάζεται ***συντάκτης*** (editor). Στη συνέχεια το πηγαίο πρόγραμμα πρέπει να μεταφραστεί σε εντολές γλώσσας μηχανής. Το έργο της μετάφρασης το αναλαμβάνουν δύο προγράμματα ο ***μεταγλωττιστής*** ή ο ***διερμηνευτής****.*

Για να μεταφραστεί το πηγαίο πρόγραμμα σε εντολές γλώσσας μηχανής δεν θα πρέπει να ανιχνευθούν λάθη. Τα λάθη που εμφανίζονται κατά τη μετάφραση ονομάζονται ***συντακτικά***. Ο μεταφραστής ανιχνεύει τα λάθη και εμφανίζει κατάλληλα διαγνωστικά μηνύματα. Στη συνέχεια ακολουθεί η διόρθωσή τους από τον προγραμματιστή.



### 3.1.1 Λογισμικό και Υπολογιστικό Σύστημα

Ένα σύγχρονο υπολογιστικό σύστημα αποτελείται από: το ***υλικό***, τα ηλεκτρονικά μέρη του υπολογιστή και το ***λογισμικό***, το σύνολο των προγραμμάτων που αξιοποιούν και διαχειρίζονται τις λειτουργίες του υλικού του υπολογιστή. Το λογισμικό χωρίζεται στο ***λειτουργικό σύστημα***, το οποίο θα μελετηθεί στη συνέχεια και στο ***λογισμικό εφαρμογών***, τα οποία αποτελούν το σύνολο των προγραμμάτων που επιλύουν τα προβλήματα των χρηστών.

### 3.1.2 Το Λειτουργικό Σύστημα και οι Αρμοδιότητές του

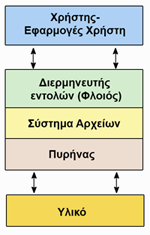
**Λειτουργικό Σύστημα (Λ.Σ.)** (Operating System – OS) είναι το σύνολο των προγραμμάτων ενός υπολογιστικού συστήματος το οποίο λειτουργεί ως σύνδεσμος ανάμεσα στα προγράμματα του χρήστη και το υλικό. Λειτουργεί ως ***ενδιάμεσος μεταξύ του ανθρώπου και της μηχανής,*** μεταφέροντας εντολές ή απαιτήσεις του χρήστη στο υπολογιστικό σύστημα.

* ***Διαχειρίζεται τους διαθέσιμους πόρους*** και να τους κατανέμει στις διάφορες διεργασίες.
* Ελέγχει την ***εκτέλεση των προγραμμάτων.***

Διαχειρίζεται τη ***λειτουργία των συσκευών εισόδου και εξόδου*** και να ελέγχει τη ροή των δεδομένων και την έξοδο των πληροφοριών.

* ***Οργανώνει*** και να ***διαχειρίζεται τα αρχεία*** του συστήματος.
* ***Ανιχνεύει***και να***εντοπίζει***πιθανά λάθη ή δυσλειτουργίες του υπολογιστικού συστήματος και να ενημερώνει τον χρήστη.
* Εφαρμόζει μηχανισμούς που βελτιώνουν την ***ασφάλεια*** του υπολογιστικού συστήματος από διάφορους κινδύνους.

### 3.1.3 Η Δομή και η Ιεραρχία του Λειτουργικού Συστήματος

Σε ένα Λ.Σ. υπάρχουν τα ακόλουθα επίπεδα 

* Ο **Πυρήνας** (Kernel), βρίσκεται πλησιέστερα προς το υλικό και αποτελεί τον ενδιάμεσο για να επιτευχθεί η επικοινωνία των προγραμμάτων με το υλικό. Ο πυρήνας «φορτώνεται» πρώτος στην κύρια μνήμη όταν ξεκινάει ο υπολογιστής.
* Το **Σύστημα Αρχείων** (File System) διαχειρίζεται τα αρχεία (δίνοντάς τους ονομασία, καταχωρώντας τα, κτλ.) και φροντίζει για τη διάθεσή τους στους χρήστες.
* Ο **Διερμηνευτής Εντολών** (Command Interpreter) ή **Φλοιός** (Shell) είναι το σύνολο των προγραμμάτων, το οποίο επιτρέπει στο χρήστη και τις εφαρμογές του να επικοινωνεί με το Λ.Σ. Η επικοινωνία γίνεται είτε με ***απευθείας εντολές*** (command mode - Εικόνα 3.2) είτε μέσω ενός ***γραφικού περιβάλλοντος διεπαφής***(GUI - Graphical User Interface – Εικόνα 3.3).

### 3.1.5Γνωστά Λειτουργικά Συστήματα

Έχουν αναπτυχθεί διάφορα Λ.Σ. τα οποία αξιοποιούν κατάλληλα την αρχιτεκτονική των επεξεργαστών. Στη συνέχεια παρατίθενται τα πιο γνωστά από αυτά.

Το **Unix ,** το **MS-DOS**, τα **MS τ**ο **Linux** ,το **Mac OS και Αndroid**

### 3.2.1 Τι είναι τα Πληροφοριακά Συστήματα

**Πληροφοριακό σύστημα (Π.Σ.) ονομάζεται ένα σύνολο οντοτήτων που συλλέγουν, επεξεργάζονται, αποθηκεύουν, βρίσκουν και μοιράζουν πληροφορίες για την υποστήριξη των αποφάσεων και ελέγχου σε μια επιχείρηση ή σε έναν οργανισμό.πχ TAXISNET, MYSCHOOL**

**Βάση Δεδομένων (Β.Δ.) είναι μία οργανωμένη συλλογή από συσχετιζόμενα δεδομένα, επεξεργασμένα και αποθηκευμένα με τέτοιο τρόπο, ώστε να μπορούν να χρησιμοποιούνται σε όλες τις εφαρμογές ενός οργανισμού ή μιας επιχείρησης.**

**Δίκτυα**

Στόχοι του κεφαλαίου είναι:

* Οι μαθητές να εντάξουν τις γνώσεις τους για θέματα επικοινωνίας και δικτύωσης συστημάτων στο σχήμα της Εφαρμοσμένης Επιστήμης των Υπολογιστών.
* Οι μαθητές να οργανώσουν σε νοητικό μοντέλο τα βασικά θέματα που αφορούν τα δίκτυα επικοινωνίας.

### 3.3.1 Τι είναι ένα Δίκτυο Υπολογιστών

**Δίκτυο Υπολογιστών** ή Δίκτυο Επικοινωνιών είναι ένα σύνολο συνδεδεμένων μεταξύ τους υπολογιστών και άλλων συσκευών (εκτυπωτές, τερματικά, δρομολογητές και δορυφόροι) που επικοινωνούν μεταξύ τους και ανταλλάσουν πληροφορίες (απλά δεδομένα, ήχο, εικόνα και βίντεο).

Πχ. (σταθερή/κινητή τηλεφωνία), η καλωδιακή τηλεόραση, το ηλεκτρονικό ταχυδρομείο, η τηλεγραφία, η τηλεδιάσκεψη), οι οικονομικές υπηρεσίες (π.χ. ηλεκτρονική συναλλαγή με την εφορία), οι πωλήσεις, το μάρκετινγκ και η διαφήμιση.

### 3.3.3 Κατηγοριοποίηση δικτύων

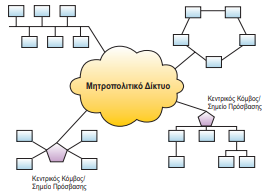
Ένα δίκτυο μπορεί να ανήκει σε διάφορες κατηγορίες ανάλογα με τη γεωγραφική του έκταση, το μέγεθός του και την τεχνολογία μετάδοσης και προώθησης που χρησιμοποιεί.

#### *3.3.3.3 Είδη δικτύων βάσει περιοχής που καλύπτουν*

Τα**Τοπικά ∆ίκτυα**(LAN – Local Area Networks) καλύπτουν μία μικρή έκταση, συνδέοντας συσκευές που βρίσκονται σε ένα δωμάτιο ή σε ένα κτίριο ή συγκρότημα κτιρίων.

Τα **Μητροπολιτικά ∆ίκτυα**(MAN – Metropolitan Area Networks) εκτείνονται στο περιβάλλον μιας ολόκληρης πόλης και χρησιμοποιούνται για την διασύνδεση δικτύων LAN ή σαν δίκτυα κορμού. Παραδείγματα τέτοιων δικτύων είναι ένα τηλεπικοινωνιακό δίκτυο υποδομής και ένα δίκτυο καλωδιακής τηλεόρασης (Εικόνα 3.15).

Τα δίκτυα**Ευρείας Περιοχής**(WAN – Wide Area Networks) επεκτείνονται σε μεγάλες γεωγραφικές περιοχές που αποτελούνται από διάφορες χώρες ή ακόμα και ηπείρους. Χρησιμοποιούν πλήθος ενδιάμεσων συσκευών, όπως π.χ. δορυφόρους.



**Εικόνα 3.15.**Παράδειγμα ενός Μητροπολιτικού δικτύου που αποτελείται από 4 δίκτυα.

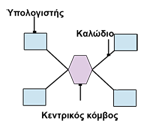
### 3.3.4 Τοπολογίες δικτύων

Ο τρόπος με τον οποίο συνδέονται οι σταθμοί σε ένα δίκτυο ονομάζεται ***τοπολογία δικτύου***.

Υπάρχουν τρεις βασικές τοπολογίες δικτύων, οι οποίες διαφέρουν ως προς την αξιοπιστία, το κόστος, την ανοχή τους σε σφάλματα και την ταχύτητα ανάνηψης μετά από κατάρρευση.

**Τοπολογία Αστέρα**: Υπάρχει ένας κεντρικός κόμβος για τον έλεγχο της κυκλοφορίας και όλες οι συσκευές συνδέονται με αυτόν με μία φυσική σύνδεση (Εικόνα 3.16).

3.3

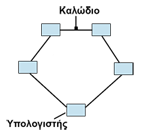


**Εικόνα 3.16.**Τοπολογία Αστέρα.

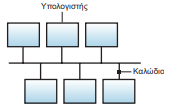
**Τοπολογία Δακτυλίου**: Σε αυτην την τοπολογία, η κάθε συσκευή συνδέεται με μια γραμμή (point-to-point link) με τις δύο διπλανές συσκευές, δημιουργώντας ένα δακτύλιο. Ένα μήνυμα μεταφέρεται από τον κάθε κόμβο στον διπλανό του προς την ίδια κατεύθυνση μέχρι να φθάσει στον προορισμό του (εικόνα 3.17). Η τοπολογία αυτή χρησιμοποιείται τόσο σε τοπικά όσο και ευρείας περιοχής δίκτυα.

**Τοπολογία Αρτηρίας**: Στην τοπολογία αυτή, υπάρχει μια γραμμή (καλώδιο) που αποτελεί τη ραχοκοκαλιά του δικτύου και όλες οι συσκευές είναι συνδεμένες σε αυτήν (Εικόνα 3.20). Η τοπολογία αυτή συναντάται κυρίως σε τοπικά δίκτυα.

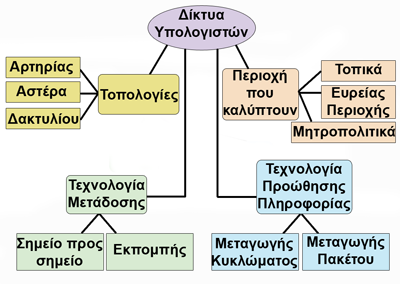
Όλες οι κατηγοριοποιήσεις που έχουν αναφερθεί μέχρι τώρα συνοψίζονται στον παρακάτω νοητικό χάρτη (Εικόνα 3.19):



**Εικόνα 3.17.**Τοπολογία Δακτυλίου.

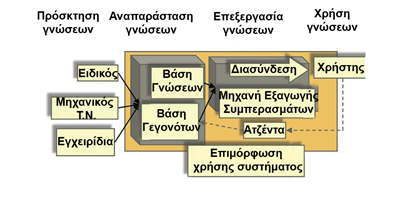


**Εικόνα 3.18.**Τοπολογία Αρτηρίας.



### 3.4.1 Τι είναι η Τεχνητή Νοημοσύνη

.4



**Εικόνα 3.22.** Αναπαράσταση Γνώσεων από την μεριά της Τεχνητής Νοημοσύνης.

.

Η Τ.Ν. είναι ο τομέας της επιστήμης των υπολογιστών, που ασχολείται, εκτός των άλλων, με τη σχεδίαση ευφυών (νοημόνων) υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζονται με τη νοημοσύνη στην ανθρώπινη συμπεριφορά. Τα προβλήματα που επιλύονται αφορούν την αντίληψη μέσω της όρασης, την μηχανική μάθηση, την εξαγωγή συμπερασμάτων από βάσεις δεδομένων κλπ. Η βασική γλώσσα προγραμματισμού της είναι η Prolog.