

ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

ΚΥΚΛΟΣ ΑΝΑΠΤΥΞΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Κύκλος Ανάπτυξης Προγράμματος

2

Κύκλος Ανάπτυξης Προγράμματος

Κύκλος Επεξεργασίας Δεδομένων (δεδομένα / είσοδος, επεξεργασία, πληροφορίες / έξοδος, αποθήκευση και πρόγραμμα)

Φάσεις Κύκλου Ανάπτυξης Προγράμματος (Περιγραφή Προβλήματος / Καθορισμός Απαιτήσεων, Ανάλυση Προβλήματος, Σχεδίαση Αλγορίθμου, Υλοποίηση Αλγορίθμου σε Πρόγραμμα, Δοκιμή και Επαλήθευση Προγράμματος (Αποσφαλμάτωση), Συντήρηση και Ενημέρωση Προγράμματος)

Πρόγραμμα Πηγαίου Κώδικα, Πρόγραμμα αντικειμένου κώδικα

Γλώσσα μηχανής

Μεταγλωττιστής και Διερμηνέας

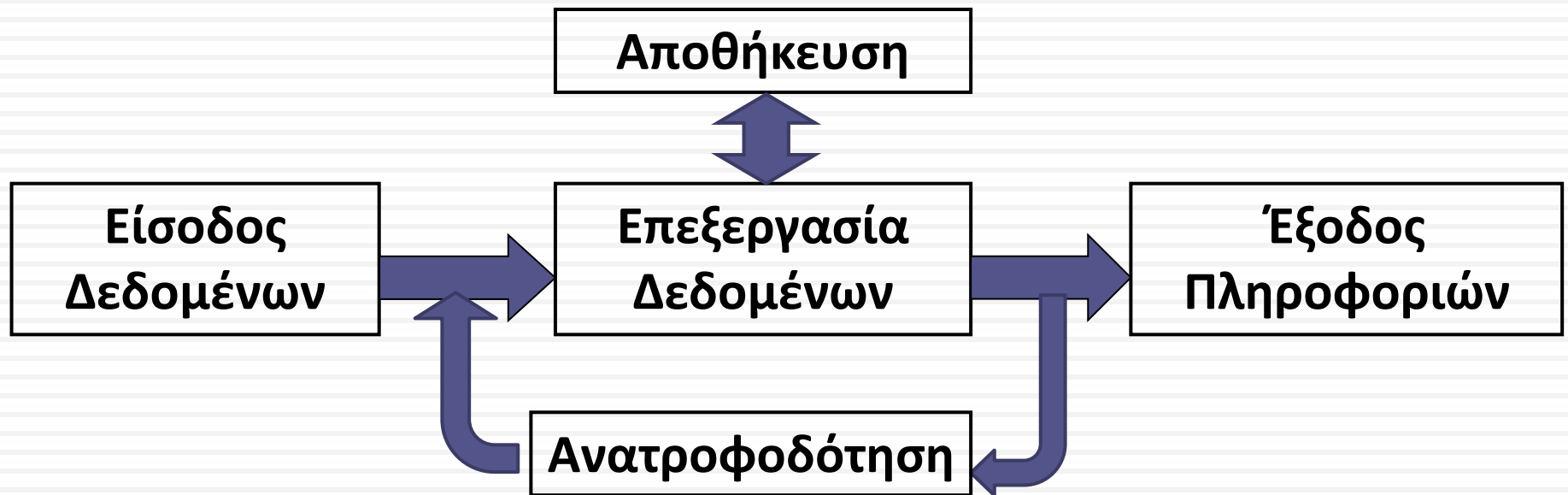
Στόχοι: Στο τέλος αυτού του μαθήματος ο μαθητή πρέπει:

3

- Να μπορεί να αναφέρει τα κύρια χαρακτηριστικά του Ηλεκτρονικού Υπολογιστή.
- Να σχεδιάζει το λειτουργικό διάγραμμα του Ηλεκτρονικού Υπολογιστή, να απαριθμεί και να επεξηγεί τις μονάδες από τις οποίες αποτελείται και να εξηγεί σε συντομία τη λειτουργία του.
- Να αναφέρει τα κύρια χαρακτηριστικά και τις κατηγορίες της Κύριας και Βοηθητικής Μνήμης.
- Να αναφέρει τις κύριες περιφερειακές συσκευές και τα χαρακτηριστικά τους και να περιγράφει σε συντομία την αρχή λειτουργίας τους.

Κύκλος επεξεργασίας δεδομένων

Ο Η/Υ δέχεται στην είσοδο τα δεδομένα, τα αποθηκεύει και τα επεξεργάζεται με μια προκαθορισμένη σειρά εντολών και στην συνέχεια μας δίνει στην έξοδο τις πληροφορίες.



Εισαγωγή - έννοιες

Εισαγωγή - έννοιες

6

- **Αλγόριθμος**: Μια σειρά από οδηγίες που οδηγούν στη λύση του προβλήματος
- **Πρόγραμμα**: Η αναπαράσταση ενός αλγορίθμου σε μορφή κατανοητή από τον υπολογιστή.
- **Προγραμματισμός**: Η διαδικασία ανάπτυξης ενός αλγορίθμου και η συγγραφή του προγράμματος
- **Γλώσσα προγραμματισμού**: Το σύνολο των γραμματικών και συντακτικών κανόνων που μας επιτρέπει να δίνουμε εντολές στον Η/Υ μέσω ενός προγράμματος.
- **Κύκλος ανάπτυξης προγράμματος**: Η διαδικασία που ακολουθούμε για την ανάπτυξη ενός προγράμματος.

Φάσεις Κύκλου Ανάπτυξης Προγράμματος

1. Περιγραφή του προβλήματος, καθορισμός απαιτήσεων,
2. Ανάλυση προβλήματος, προσδιορισμός της λύσης
3. Σχεδίαση της λύσης του προβλήματος
 - Ανάπτυξη αλγορίθμου
 - Σχεδιασμός διαγράμματος ροής
 - Δημιουργία ψευδοκώδικα
4. Κωδικοποίηση σε γλώσσα προγραμματισμού,
5. Έλεγχος, διόρθωση λαθών,
6. Συντήρηση προγράμματος, τεκμηρίωση.

1. Περιγραφή του προβλήματος & καθορισμός απαιτήσεων

- Με απλά βήματα απομονώνουμε και καταγράφουμε τις πραγματικές συνιστώσες ενός προβλήματος, τοποθετώντας τις σε λογική σειρά μεταξύ τους.
 - ▣ Ποια είναι τα δεδομένα εισόδου;
 - ▣ Ποιο είναι το αποτέλεσμα που πρέπει να προκύψει από την επίλυση του προβλήματος;
- Αποσαφηνίζουμε τους στόχους που επιδιώκουμε να υλοποιήσουμε με αναλυτικό τρόπο προκειμένου να καταγραφεί το πλαίσιο απαιτήσεων της όλης προσπάθειας.
 - ▣ Μήπως οι απαιτήσεις για την επίλυση του προβλήματος καθιστούν την αυτοματοποίηση της διαδικασίας μη συμφέρουσα;

2. Ανάλυση του προβλήματος & προσδιορισμός της λύσης

- Σκιαγραφούμε ένα προσχέδιο της επίλυσης του προβλήματος.
- Ελέγχουμε αν η λύση καλύπτει τους στόχους που έχουν τεθεί και αν παράγει τα επιθυμητά δεδομένα εξόδου.
- Διερευνούμε την πιθανότητα ύπαρξης περισσότερων λύσεων.
- Επιλέγουμε την βέλτιστη λύση με βάση τις προδιαγραφές που έχουν τεθεί.

ΠΡΟΣΟΧΗ: ΛΑΘΟΣ ΑΝΑΛΥΣΗ => ΛΑΘΟΣ ΛΥΣΗ

2. Ανάλυση του προβλήματος & προσδιορισμός της λύσης

- Ως μέρος της ανάλυσης απαντούμε και στις πιο κάτω ερωτήσεις:
 - ▣ Ποια είναι τα δεδομένα (inputs)
 - ▣ Ποια πρέπει να είναι τα εξαγόμενα αποτελέσματα (outputs)
 - ▣ Τι χρειάζεται να γίνει (επεξεργασία)

ΠΡΟΣΟΧΗ: ΛΑΘΟΣ ΑΝΑΛΥΣΗ => ΛΑΘΟΣ ΛΥΣΗ

3. Σχεδίαση της λύσης του προβλήματος

- Σκιαγραφούμε ένα προσχέδιο της επίλυσης του προβλήματος
- Αναπτύσσουμε τον αλγόριθμο επίλυσης του προβλήματος
 - ακολουθία αυστηρά δομημένων βημάτων προκειμένου να επιλύσουμε το πρόβλημα.
- Για την περιγραφή της λύσης ενός προγράμματος χρησιμοποιούμε
 - τον ψευδοκώδικα. ή/και
 - το λογικό διάγραμμα

3. Σχεδίαση της λύσης του προβλήματος

- Σκιαγραφούμε ένα προσχέδιο της επίλυσης του προβλήματος
- Αναπτύσσουμε τον αλγόριθμο επίλυσης του προβλήματος
 - ▣ Δημιουργία μιας σειράς μη διαφορούμενων βημάτων που οδηγούν στη λύση του προβλήματος.
- Για την περιγραφή της λύσης ενός προγράμματος χρησιμοποιούμε
 - ▣ λογικό διάγραμμα
 - ▣ ψευδοκώδικα.

3.1 Αλγόριθμος

Ορισμός Αλγορίθμου

Αλγόριθμος είναι μια πεπερασμένη σειρά εκτελέσιμων και σαφών εντολών που οδηγεί στην επίλυση ενός προβλήματος

3.1 Αλγόριθμος

Παραδείγματα αλγορίθμων

- Οδηγίες πλυντηρίου
- Οδηγίες για συναρμολόγηση επίπλου
- Στην Πληροφορική
 - Ταξινόμηση Λίστας Αριθμών
 - Εύρεση Μέσου Όρου Μιας Λίστας Αριθμών
 -

3.1 Αλγόριθμος

- Αρχικά προδιαγράφεται τι είναι το δεδομένο και τι το επιδιωκόμενο αποτέλεσμα.
- Αυτό σταδιακά εκλεπτύνεται.
- Η σταδιακή διάσπαση (εκλέπτυνση) συνεχίζεται μέχρις ότου φτάσουμε σε ατομικά υποπροβλήματα, δηλαδή προβλήματα που δεν είναι δυνατό να διασπαστούν περαιτέρω.
- • Στο χαμηλότερο επίπεδο ο αλγόριθμος διατυπώνει με σαφήνεια την ακριβή διαδικασία παραγωγής της λύσης του προβλήματος

3.1 Αλγόριθμος

Γιατί να καταγράψω τον αλγόριθμο;

- **Για ιδίαν χρήση** – Δεν χρειάζεται να ξανασκεφτείτε το πρόβλημα
- Όστε άλλοι να μπορούν να το επιλύσουν, χωρίς να ξέρουν πολλά γύρω από αυτό
- Δεν χρειάζεται να καταλάβουν τις αρχές πίσω από αυτό το πρόβλημα.
- Απλώς ακολουθούν τις εντολές.
- Η νοημοσύνη είναι «κωδικοποιημένη στον αλγόριθμο»
- **Για να σας βοηθήσει να καταλάβετε αν επιλύει ορθά το πρόβλημα, βρείτε αν είναι αποδοτικός.**

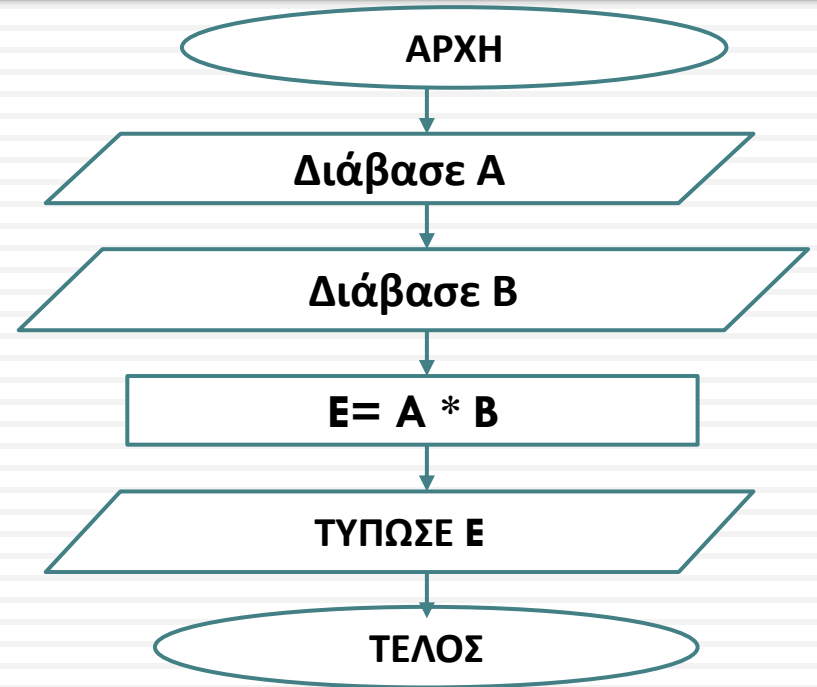
3.2 Λογικό Διάγραμμα ή Διάγραμμα Ροής

Λογικό Διάγραμμα: Μέθοδος σχηματικής αναπαράστασης της ροής των οδηγιών που συνθέτουν έναν αλγόριθμο.

Λεκτική Περιγραφή

1. Διάβασε μήκος πλευράς a
2. Διάβασε μήκος πλευράς β
3. Υπολόγισε το γινόμενο των δύο αριθμών που ισοούνται με το εμβαδόν ($E=a*\beta$)
4. Τύπωσε το Εμβαδόν (E)

Λογικό Διάγραμμα



3.3 Ψευδοκώδικας

Ψευδοκώδικας: Είναι ένα μίγμα κοινών σε διάφορες γλώσσες προγραμματισμού όρων (εντολών), που χρησιμοποιούνται για να εκφράσουμε τη λύση.

Λεκτική Περιγραφή

1. Άνοιξε το αρχείο με τα στοιχεία των μαθητών
2. Διάβασε το όνομα του πρώτου μαθητή
3. Διάβασε το τμήμα του μαθητή
4. Έλεγξε αν το τμήμα του μαθητή είναι ΘΗΨ1
5. Τύπωσε το όνομα του μαθητή
6. Επανάλαβε τη διαδικασία από το 3^ο μέχρι το 5^ο βήμα μέχρι να φτάσεις στο τέλος του αρχείου
7. Κλείσε το αρχείο
8. Τέλος προγράμματος.

Ψευδοκώδικας

1. open STUDENT.DAT
2. read student name
3. while not EOF do
4. read student name
5. read student class
6. if class="ΘΗΨ1" then print student name
7. end while
8. close STUDENT.DAT
9. End

4. Κωδικοποίηση

- Πραγματοποιείται η συγγραφή του προγράμματος σε μια γλώσσα προγραμματισμού.
- Μέσω **μεταγλωττιστή** (compiler) ή **μεταφραστή** (interpreter) το πρόγραμμα μετατρέπεται σε **γλώσσα μηχανής** η οποία είναι αναγνωρίσιμη από τον υπολογιστή.
- Στο στάδιο αυτό γίνεται ο έλεγχος **συντακτικών λαθών**

5. Έλεγχος λαθών

Διόρθωση προγράμματος

- Διορθώνονται πιθανά λογικά σφάλματα (σφάλματα που σχετίζονται με τον σχεδιασμό της λύσης).
- Λύνουμε με το χέρι ή άλλους τρόπους το πρόβλημα σε ένα σύνολο δεδομένων και συγκρίνουμε το αποτέλεσμα με τις εξόδους του προγράμματος
 - ▣ Σφάλματα σύνταξης
 - ▣ Σφάλματα που σχετίζονται με το αν χρησιμοποιήσαμε σωστά τη γλώσσα προγραμματισμού στη διάρκεια της υλοποίησης
 - ▣ Σφάλματα Run-time
 - ▣ Σφάλματα κατά τη διάρκεια εκτέλεσης του προγράμματος

6. Συντήρηση προγράμματος

- Συγγραφή τεκμηρίωσης
- Το λογισμικό εγκαθίσταται και ξεκινά η λειτουργία του.
- Οι χρήστες επιθυμούν (ή χρειάζονται) περισσότερα από το πρόγραμμα
- Όταν το περιβάλλον αλλάζει πρέπει να αλλάξει και το πρόγραμμα

6.2 Αξιολόγηση Λύσεων

- Ορθότητα Λύσεων
 - ▣ Αναλυτικές Μέθοδοι –Αποδείξεις
 - ▣ Εμπειρικές Μέθοδοι -Δοκιμές
- Τεκμηρίωση Λύσεων
 - ▣ Σχόλια στο κώδικα
 - ▣ Ευκολία κατανοήσεως
- Εκτίμηση Απόδοσης
 - ▣ Ταχύτητα, ανάγκη σε μνήμη
 - ▣ Αναλυτικές/Εμπειρικές Μέθοδοι
- Επεκτασιμότητα

Παράδειγμα Προγραμματισμού

Γράψετε ένα πρόγραμμα το οποίο να μετατρέπει μίλια σε χιλιόμετρα.

Βήμα1. Κατανόηση

- Τι μας ζητά;
 - Τι μίλια;
- αγγλικά μίλια, ναυτικά μίλια;
 - Από πού παίρνουμε τις πληροφορίες;
- Από τον χρήστη, από αρχείο;

Βήμα2. Ανάλυση

- Δεδομένα (εισόδου): μίλια
- Δεδομένα (εξόδου): χιλιόμετρα
- Άλλα δεδομένα: σχέση 1mile:1.609Km
- Επεξεργασία: $\text{χιλιόμετρα} = \text{μίλια} * 1.609$
- Παράδειγμα: Πόσα Km είναι 10 μίλια?
 $1.609 * 10 \text{ μίλια} = 16.09 \text{ χιλιόμετρα}$

Βήμα 3. Σχεδιασμός Λύσης

- Δεδομένα (εισόδου): μίλια
- Δεδομένα (εξόδου): χιλιόμετρα
- Άλλα δεδομένα: σχέση 1mile:1.609Km
- Επεξεργασία: $\text{χιλιόμετρα} = \text{μίλια} * 1.609$
- Παράδειγμα: Πόσα Km είναι 10 μίλια?
 $1.609 * 10 \text{ μίλια} = 16.09 \text{ χιλιόμετρα}$

Βήμα 3. Σχεδιασμός Λύσης

Αλγόριθμος

1. Πάρε τα δεδομένα εισόδου
2. Κάνε τη μετατροπή
3. Παρουσίασε το αποτέλεσμα

1η Εκλέπτυνση

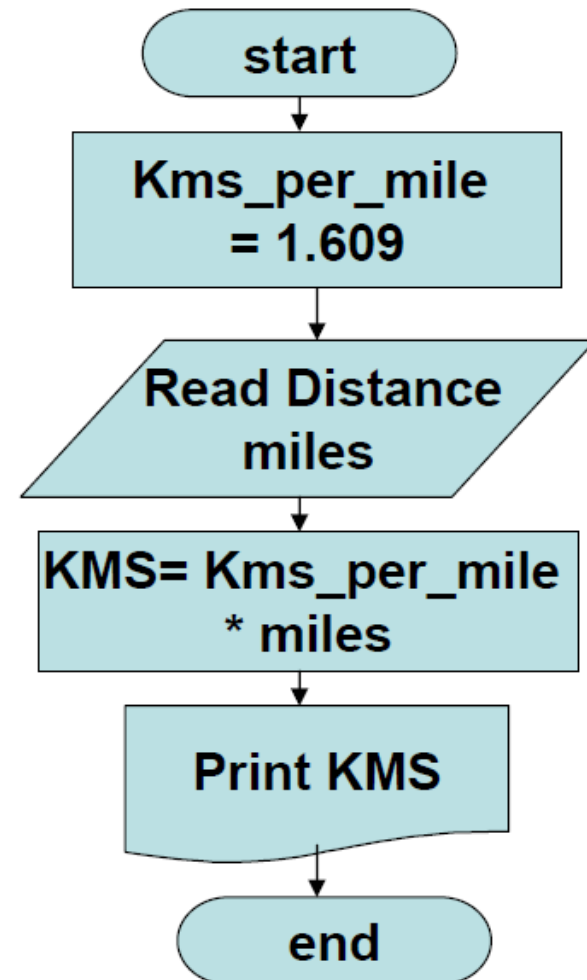
1. Διάβασε τα Μίλια
2. Υπολόγισε:
 $\text{Χλμ.} = \text{Μίλια} * 1.609$
3. Δείξε το αποτέλεσμα στην οθόνη

Βήμα 3. Σχεδιασμός Λύσης

2η Εκλέπτυνση

1. Διάβασε τα Μίλια
2. Αποθήκευσε τα Μίλια στη μεταβλητή miles
3. Υπολόγισε:
 $Kms = Miles * 1.609$
4. Εμφάνισε τη μεταβλητή Kms στην οθόνη

Διάγραμμα Ροής



Βήμα4. Υλοποίηση

```
1.  /*
2.   * Converts distance in miles to kilometers.
3.   */
4.  #include <stdio.h>           /* printf, scanf definitions */
5.  #define KMS_PER_MILE 1.609   /* conversion constant      */
6.
7.  int
8.  main(void)
9.  {
10.     double miles, /* input - distance in miles.      */
11.           kms;    /* output - distance in kilometers */
12.
13.     /* Get the distance in miles. */
14.     printf("Enter the distance in miles> ");
15.     scanf("%lf", &miles);
16.
17.     /* Convert the distance to kilometers. */
18.     kms = KMS_PER_MILE * miles;
19.
20.     /* Display the distance in kilometers. */
21.     printf("That equals %f kilometers.\n", kms);
22.
23.     return (0);
24. }
```

Sample Run

```
Enter the distance in miles> 10.00
That equals 16.090000 kilometers.
```