

ΘΕΜΑΤΙΚΗ ΕΝΟΤΗΤΑ 2: ΘΕΜΑΤΑ ΘΕΩΡΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

§2.3 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Στόχοι:

1. Να δημιουργείτε ευκρινές γνωσιακό και οργανωμένο νοητικό σχήμα που να περιλαμβάνει τα είδη και τεχνικές προγραμματισμού, με βάση την πρότερη εμπειρία τους..

Χρονοπρογραμματισμός

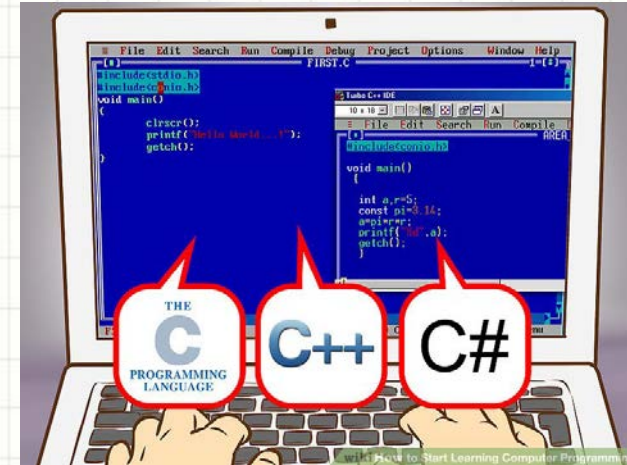
Διδακτέα Ύλη	Διάρκεια (ώρες)
§2.3.1 Αναφορά σε γλώσσες προγραμματισμού §2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού §2.3.2 Σχεδίαση και συγγραφή κώδικα	1

§2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

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

Η διαδικασία κατασκευής των προγραμμάτων σε κάποια γλώσσα προγραμματισμού ονομάζεται **προγραμματισμός**.

Τα άτομα που γράφουν ένα πρόγραμμα ονομάζονται **προγραμματιστές**.



```
import javax.swing.*;
import java.lang.*;
import java.util.*;
import java.io.*;

public class Selections extends Thread {

    public Selections() {
        Object object = null;

        try {
            Mpe3000 mpe3000 = new Mpe3000("172.24.64.78:30803:mgr/sooper_sysadmin");
            Database loadb = new Database("music.adbctest_sysadmin",5,mpe3000);
            loadb.connect("MDS");
            // This method "tumps" the Selections dataset
            TurboBuffer inv_location = null;
            inv_location = loadb.createTurboBuffer("Selections");
            inv_location.setFetchSize( 1000 );
            inv_location.setColumns("8");

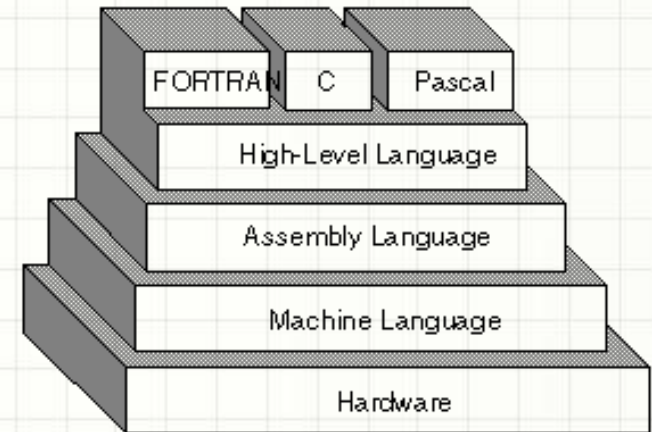
            System.out.println("Selections Details");
            System.out.println(" ");
            // Passing NO parameter to the turbobuffer.initializeRead() method will perform a serial read.
            inv_location.initializeRead();
            boolean notfound = true;
            while ( inv_location.next() ) {
                System.out.println("Album Code = " + inv_location.getString("AlbumCode"));
                System.out.println("Selection Name = " + inv_location.getString("SelectionName"));
                System.out.println("Composer Name = " + inv_location.getString("ComposerName"));
                System.out.println("Performer = " + inv_location.getString("Performer"));
                System.out.println(" ");
                System.out.println(" ");
            }
        } catch (ADBException ex) { System.out.println("TurboHUGZ error " + ex.getMessage());
            ex.printStackTrace();
        }

        public static void main( String[] args ) {
            Selections Selections = new Selections();
        }
    }
}
```


§2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

Ιστορική εξέλιξη:

- Γλώσσα μηχανής (1^ης γενιάς)
- Συμβολικές γλώσσες (2^ης γενιάς)
- Γλώσσες υψηλού επιπέδου (3^ης γενιάς)



High-level Language

```
temp = v[k];  
v[k] = v[k+1];  
v[k+1] = temp;
```

```
TEMP = V(K)  
V(K) = V(K+1)  
V(K+1) = TEMP
```

C/Java Compiler

Fortran Compiler

Assembly Language

```
lw $t0, 0($2)  
lw $r1, 4($2)  
sw $r1, 0($2)  
sw $t0, 4($2)
```

MIPS Assembler

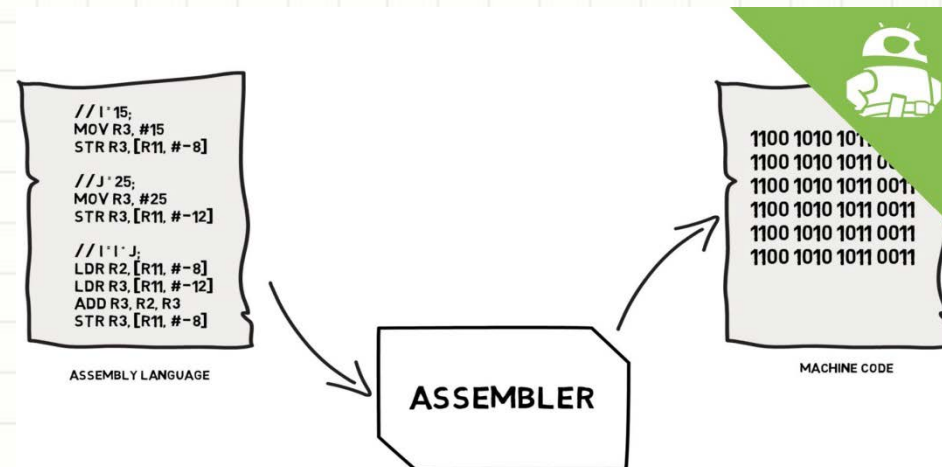
Machine Language

```
0000 1001 1100 0110 1010 1111 0101 1000  
1010 1111 0101 1000 0000 1001 1100 0110  
1100 0110 1010 1111 0101 1000 0000 1001  
0101 1000 0000 1001 1100 0110 1010 1111
```


§2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

Οι **συμβολικές γλώσσες** κάνουν χρήση εντολών (συμβολικά ονόματα), που αντιστοιχούν σε εντολές της γλώσσας μηχανής.

Τη μετάφραση των εντολών σε γλώσσα μηχανής την αναλαμβάνει ειδικό πρόγραμμα, ο **συμβολομεταφραστής**.

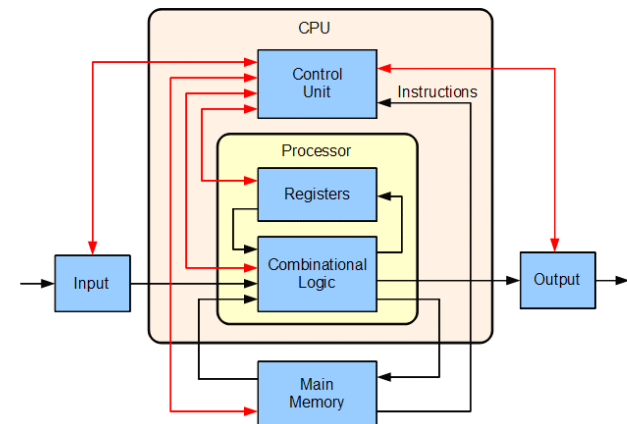


§2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

Οι **συμβολικές γλώσσες**:

1. Είναι στενά συνδεδεμένες με την αρχιτεκτονική του υπολογιστή
2. Οδηγούν σε μακροσκελή προγράμματα που είναι δύσκολο να γραφούν και να συντηρηθούν
3. Δεν μπορούν να μεταφερθούν και να εκτελεστούν σε υπολογιστή διαφορετικής αρχιτεκτονικής.

Αρχιτεκτονική υπολογιστών: το σύνολο των κανόνων και μεθόδων που περιγράφουν την λειτουργικότητα, οργάνωση και υλοποίηση των υπολογιστικών συστημάτων.



§2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

Οι **γλώσσες υψηλού επιπέδου** κάνουν χρήση εντολών, απλές λέξεις της αγγλικής γλώσσας, ακολουθώντας αυστηρούς κανόνες σύνταξης.

- Φυσικότερος τρόπος έκφρασης προβλημάτων
- Δυνατότητα μεταφερσιμότητας, δηλ. εκτέλεση των προγραμμάτων σε οποιοδήποτε υπολογιστή
- Εύκολη εκμάθηση
- Εύκολη διόρθωση λαθών
- Εύκολη συντήρηση



§2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

```
10101000 00001010
10001100 00000001
00111100
01010001 00000001
01000011 00000001
11000000 11111010
10001100 00000010
11111111
```

```
LOOP
INDEX=$01
SUM=$02
LDA #10
STA INDEX
LOOP CLA
ADD INDEX
DEC INDEX
BNE LOOP
STA SUM
BRK
```

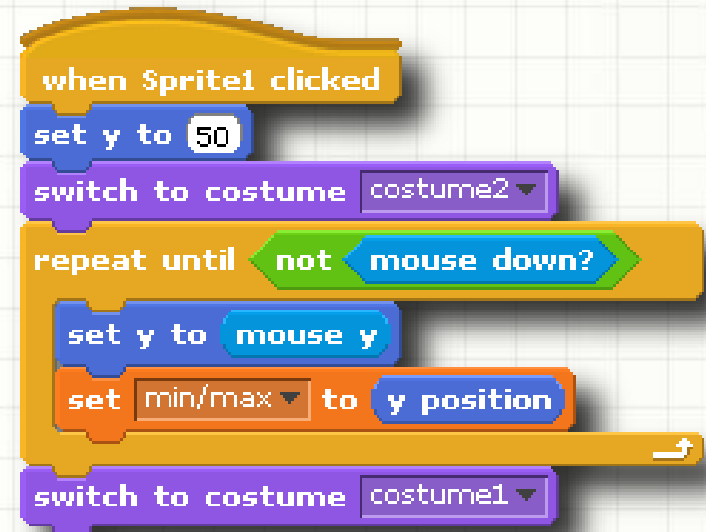
```
sum = 0
FOR index=1 TO 10
  sum=sum+index
NEXT index
END
```

Ένα πρόγραμμα σε γλώσσα μηχανής, σε συμβολική γλώσσα και σε γλώσσα υψηλού επιπέδου για τον υπολογισμό του αθροίσματος των αριθμών 1 έως 10

§2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

Η εμφάνιση των γραφικών περιβαλλόντων εργασίας, οδήγησαν στον οπτικό προγραμματισμό (visual programming):

- Visual Basic
- Visual C++
- Delphi
- Scratch
- BYOB
- Google App Inventor



Σε ένα οπτικό περιβάλλον προγραμματισμού υπάρχει η δυνατότητα να δημιουργείται γραφικά ολόκληρο το περιβάλλον της εφαρμογής (διάλογοι, μενού, κτλ)

Κεφ. 7 Βασικές Έννοιες Προγραμματισμού



Βιβλίο Μαθητή

§2.3.1 Αναφορά σε γλώσσες προγραμματισμού

§2.3.1.1 Πρόγραμμα και Γλώσσες Προγραμματισμού

Δραστηριότητες

§2.3.2 Σχεδίαση και συγγραφή κώδικα

Το πρόγραμμα που γράφεται σε κάποια γλώσσα προγραμματισμού ονομάζεται **πηγαίο πρόγραμμα** (source code)

Για την σύνταξη του πηγαίου προγράμματος χρησιμοποιείται ένα ειδικό πρόγραμμα που ονομάζεται **συντάκτης** (editor)

Κάθε πρόγραμμα γραμμένο σε οποιαδήποτε γλώσσα προγραμματισμού προκειμένου να εκτελεστεί, πρέπει να μετατραπεί σε μορφή αναγνωρίσιμη και εκτελέσιμη από τον υπολογιστή, δηλαδή σε εντολές **γλώσσας μηχανής**.

Τη μετάφραση του πηγαίου προγράμματος σε γλώσσα μηχανής αναλαμβάνουν δύο ειδικά προγράμματα: ο **μεταγλωττιστής** ή ο **διερμηνευτής**.

```
#include <stdio.h>
main() {
    int a, b, S, M
    scanf ("%d %d", &a,
    &b);
    S = a + b;
    P = a * b;
    printf (S, P);
}
```

Αλγόριθμος Πράξεις

Διάβασε a, b

S <- a + b

P <- a * b

Εμφάνισε S, P

Τέλος Πράξεις

§2.3.2 Σχεδίαση και συγγραφή κώδικα

Ο μεταγλωττιστής δέχεται ολόκληρο το πηγαίο πρόγραμμα και παράγει ισοδύναμο σε γλώσσα μηχανής



- Κατά την μετάφραση, ο μεταγλωττιστής ανιχνεύει τα συντακτικά λάθη και εμφανίζει κατάλληλα διαγνωστικά μηνύματα.
- Ακολουθεί η διόρθωση των λαθών από τον προγραμματιστή και το αρχικό πρόγραμμα υποβάλλεται ξανά για μεταγλώττιση.
- Η διαδικασία επαναλαμβάνεται μέχρι να εξαλειφθούν πλήρως τα συντακτικά λάθη.

§2.3.2 Σχεδίαση και συγγραφή κώδικα

- Το **αντικείμενο** πρόγραμμα, παρόλο που είναι σε μορφή κατανοητή από τον υπολογιστή, συνήθως δεν είναι σε θέση να εκτελεστεί.
- Χρειάζεται να συνδεθεί με άλλα τμήματα προγράμματος που είτε έχει γράψει ο προγραμματιστής είτε αποτελούν μέρος της γλώσσας προγραμματισμού, βρίσκονται στις **βιβλιοθήκες** της γλώσσας.
- Την σύνδεση αναλαμβάνει ειδικό πρόγραμμα, ο **συνδέτης-φορτωτής** (linker-loader), ο οποίος τελικά παράγει το **εκτελέσιμο** πρόγραμμα (executable).
- Το **εκτελέσιμο πρόγραμμα** είναι το πρόγραμμα που εκτελείται από τον υπολογιστή και είναι τελείως ανεξάρτητο από το πηγαίο πρόγραμμα.

§2.3.2 Σχεδίαση και συγγραφή κώδικα

Τα λάθη σε ένα πρόγραμμα είναι δύο ειδών:

■ **Συντακτικά** λάθη (syntax errors)

Εμφανίζονται στο στάδιο της **μεταγλώττισης** του αρχικού προγράμματος είτε από μεταγλωττιστή είτε από διερμηνευτή.

- Οφείλονται σε αναγραμματισμούς ονομάτων εντολών, παράλειψη δήλωσης δεδομένων, κλπ.
- Ο προγραμματιστής πρέπει να τα διορθώσει για να παραχθεί το εκτελέσιμο πρόγραμμα

■ **Λογικά** λάθη (logical errors)

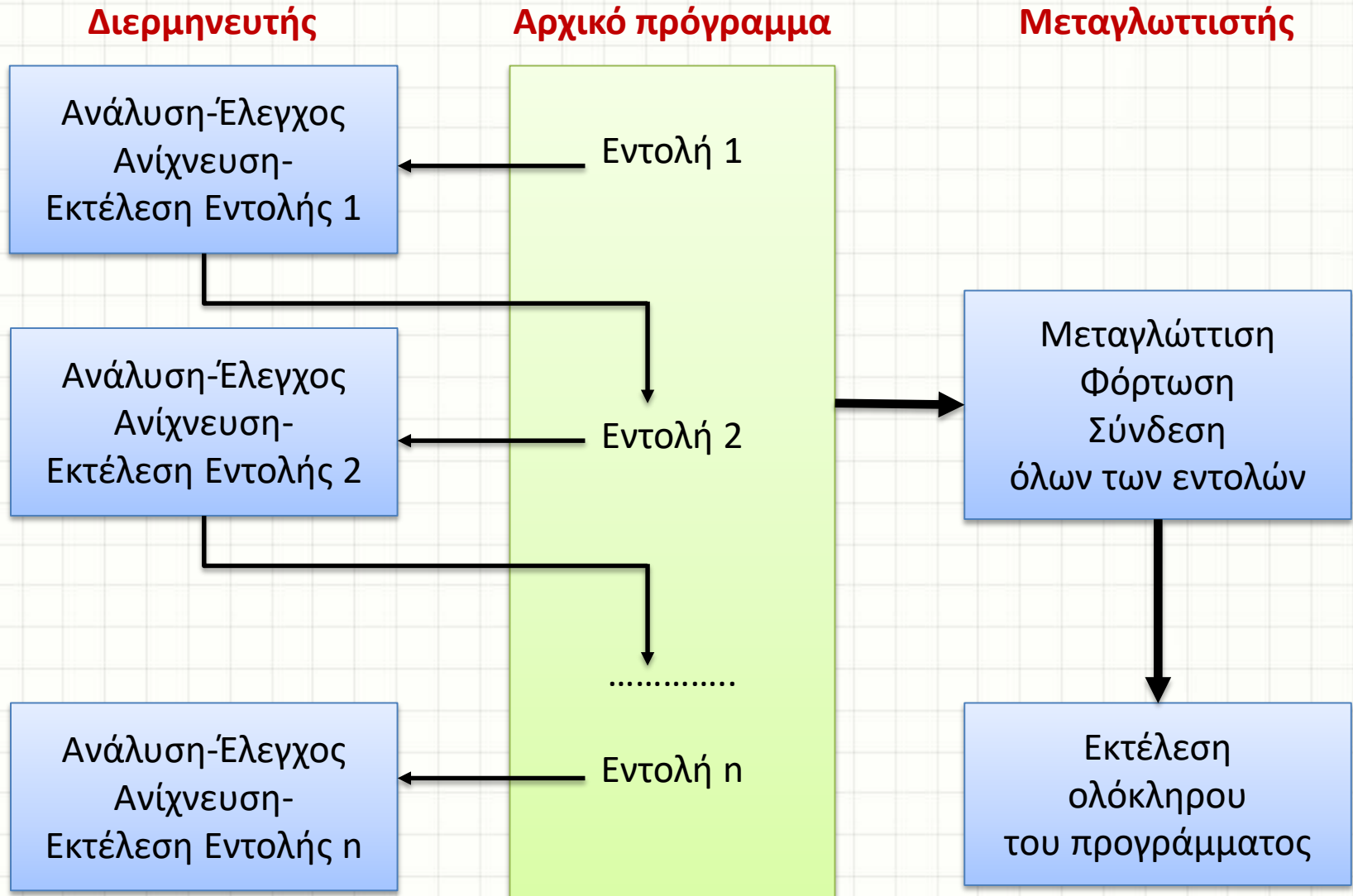
Εμφανίζονται μόνο στην **εκτέλεση** του προγράμματος

- Είναι τα πιο επικίνδυνα γιατί οφείλονται σε σφάλματα στην υλοποίηση του αλγορίθμου.

§2.3.2 Σχεδίαση και συγγραφή κώδικα

Ο **διερμηνευτής** (interpreter) διαβάζει μία προς μία τις εντολές του αρχικού προγράμματος και για καθεμία εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής.

§2.3.2 Σχεδίαση και συγγραφή κώδικα



Κεφ. 7 Βασικές Έννοιες Προγραμματισμού



Βιβλίο Μαθητή

§2.3.2 Σχεδίαση και συγγραφή κώδικα

Δραστηριότητες

Παράδειγμα 2.33 αντιμετάθεση μεταβλητών (σελ.62)
