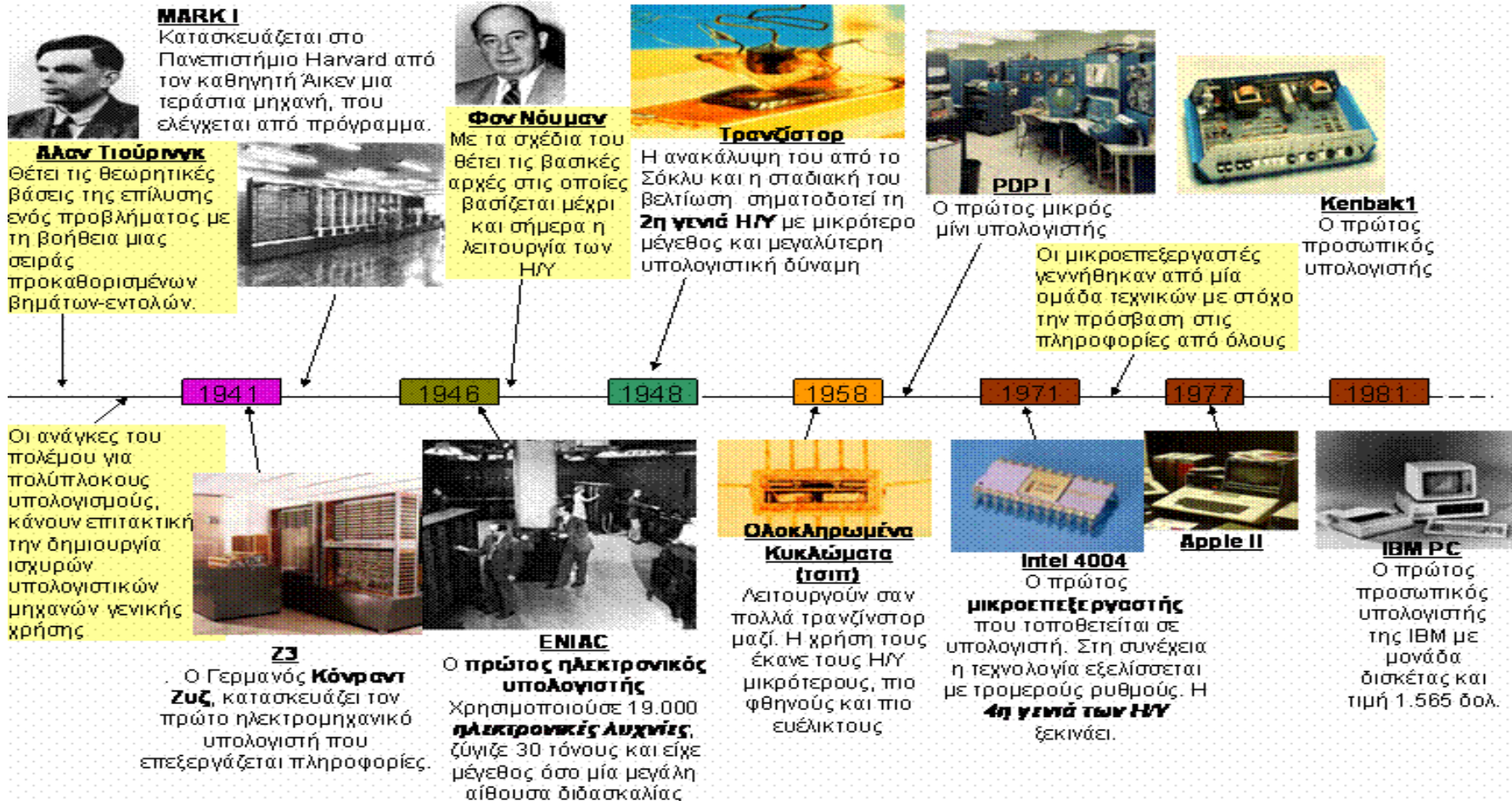


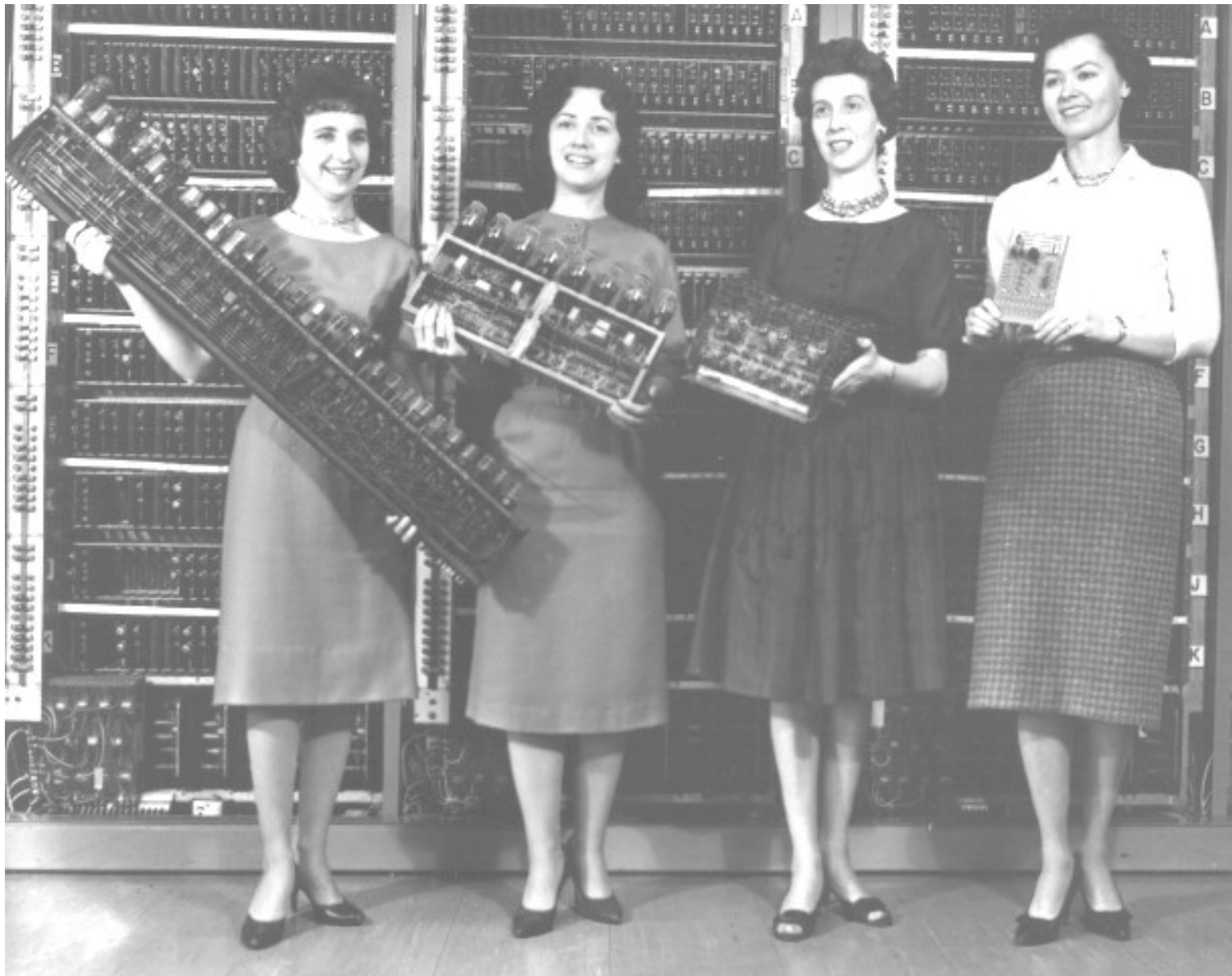
# 6.1 Η Έννοια του Προγράμματος



Ο προγραμματισμός ασχολείται με το σύνολο των εντολών που δίνονται στον υπολογιστή ώστε να υλοποιείται ο αλγόριθμος.

# 6.2 Ιστορία-Γλώσσες προγραμματισμού





**Εικόνα 4.3: Διάφορες πλακέτες υπολογιστών. Μέσα σε δύο δεκαετίες το μέγεθος μικραίνει εντυπωσιακά. (από τα αριστερά): ENIAC, EDVAC (ηλεκτρονικές λυχνίες), ORDVAC (τρανζίστορ), BRLESC-I (1962, ολοκληρωμένα κυκλώματα).**

# Γλώσσες προγραμματισμού 1ης Γενεάς Γλώσσα Μηχανής



Ένα πρόγραμμα σε γλώσσα μηχανής είναι μία ακολουθία δυαδικών ψηφίων που αποτελούν εντολές προς τον επεξεργαστή για στοιχειώδεις λειτουργίες-πράξεις.

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

## 2η Γενεά

### Συμβολική γλώσσα ή γλώσσα χαμηλού επιπέδου



Οι εντολές αποτελούνταν από συμβολικά ονόματα που αντιστοιχούσαν σε εντολές της γλώσσας μηχανής. Τη μετατροπή των συμβολικών ονομάτων σε γλώσσα μηχανής την κάνει ένα ειδικό πρόγραμμα ο **συμβολομεταφραστής (assembler)**

	INDEX = \$01		10101000	00001010
	SUM = \$02		10001100	00000001
	LDA #10		00111100	
	STA INDEX		01010001	00000001
	CLA		01000011	00000001
LOOP	ADD INDEX		11000000	11111010
	DEC INDEX		10001100	00000010
	BNE LOOP		11111111	
	STA SUM			
	BRK			

# 3η Γενεά

## Γλώσσες υψηλού επιπέδου

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

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

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

Πρόγραμμα σε  
γλώσσα υψηλού  
επιπέδου



Πρόγραμμα σε γλώσσα Μηχανής

# Γλώσσες υψηλού επιπέδου

## **FORTRAN**

Γλώσσα για την επίλυση μαθηματικών επιστημονικών προβλημάτων.

## **LISP**

Γλώσσα στον τομέα της Τεχνητής Νοημοσύνης Χειρισμός λίστας συμβόλων

## **PROLOG**

Γλώσσα στον τομέα Τεχνητής Νοημοσύνης

## **C**

Γλώσσα για δομημένο προγραμματισμό αλλά και με πολλές δυνατότητες χαμηλού επιπέδου.

**Οπτικός Προγραμματισμός οδηγούμενος από γεγονότα**

1957

1960

1970

1972

## **COBOL**

Γλώσσα για ανάπτυξη εμπορικών εφαρμογών

## **BASIC**

Γλώσσα γενικού σκοπού για εκπαίδευση αρχάριων

## **PASCAL**

Γλώσσα κατάλληλη για εκπαίδευση αλλά και για ανάπτυξη ισχυρών προγραμμάτων. Χαρακτηρίζεται από το **δομημένα προγράμματα**.

## **C++**

Γλώσσα **αντικειμενοστραφή προγραμματισμού**

## **SQL**

Γλώσσα Ερωτήσεων απαντήσεων.

## **ALGOL**

Αλγοριθμική γλώσσα Αρχικά δημιουργήθηκε για γενικής φύσεως εφαρμογές.

## **JAVA**

Αντικειμενοστραφή γλώσσα για εφαρμογές του Δυαδικτύου.

# Πλεονεκτήματα γλωσσών υψηλού επιπέδου έναντι γλωσσών χαμηλού επιπέδου



Τα προγράμματα είναι πιο κοντά στα προβλήματα που επιλύουν.



Η ανεξαρτησία από τον τύπο του υπολογιστή.  
Δυνατότητα **μεταφερσιμότητας**.



Ευκολία στην εκμάθηση και εκπαίδευση.



Ευκολότερη διόρθωση λαθών και συντήρηση προγραμμάτων.



Μείωση του κόστους και του χρόνου ανάπτυξης προγραμμάτων.



# Κατηγορίες γλωσσών ως προς τις προγραμματιστικές ιδέες που αντιπροσωπεύουν



Αλγοριθμικές ή διαδικασιών (π.χ Pascal)



Αντικειμενοστραφείς (π.χ. C++)



Συναρτησιακές (π.χ. LISP)



Μη διαδικασιακές (π.χ. PROLOG)



Γλώσσες Ερωταπαντήσεων (π.χ. SQL)

# Κατηγορίες γλωσσών ως προς το χώρο χρήσης τους



Γενικής χρήσεως (Θεωρητικά όλες)



Επιστημονικής κατεύθυνσης (π.χ. FORTRAN)



Προγραμματισμού συστημάτων (π.χ. C)



Τεχνητής Νοημοσύνης (π.χ. LISP, PROLOG)



Ειδικής χρήσης. (π.χ. ρομποτική, συστήματα διοίκησης βάσεων δεδομένων, στην εκπαίδευση μέσω υπολογιστή κ.α.)

## 6.3 Φυσικές και Τεχνητές γλώσσες



**Αλφάβητο** τα σύμβολα – στοιχεία που χρησιμοποιεί.



**Λεξιλόγιο** δηλαδή τις λέξεις που χρησιμοποιεί



**Γραμματική:**

**Τυπικό** δηλαδή το σύνολο των κανόνων που ορίζει με ποια μορφή γίνεται μία λέξη αποδεκτή.

**Συντακτικό** δηλαδή οι κανόνες που καθορίζουν τη διάταξη και τη σύνδεση των λέξεων για τη δημιουργία προτάσεων (εντολών, εκφράσεων )



**Σημασιολογία.** Το σύνολο των κανόνων που καθορίζουν το νόημα των λέξεων, των εκφράσεων και των προτάσεων.

# Διαφορές σε Φυσικές και Τεχνητές γλώσσες

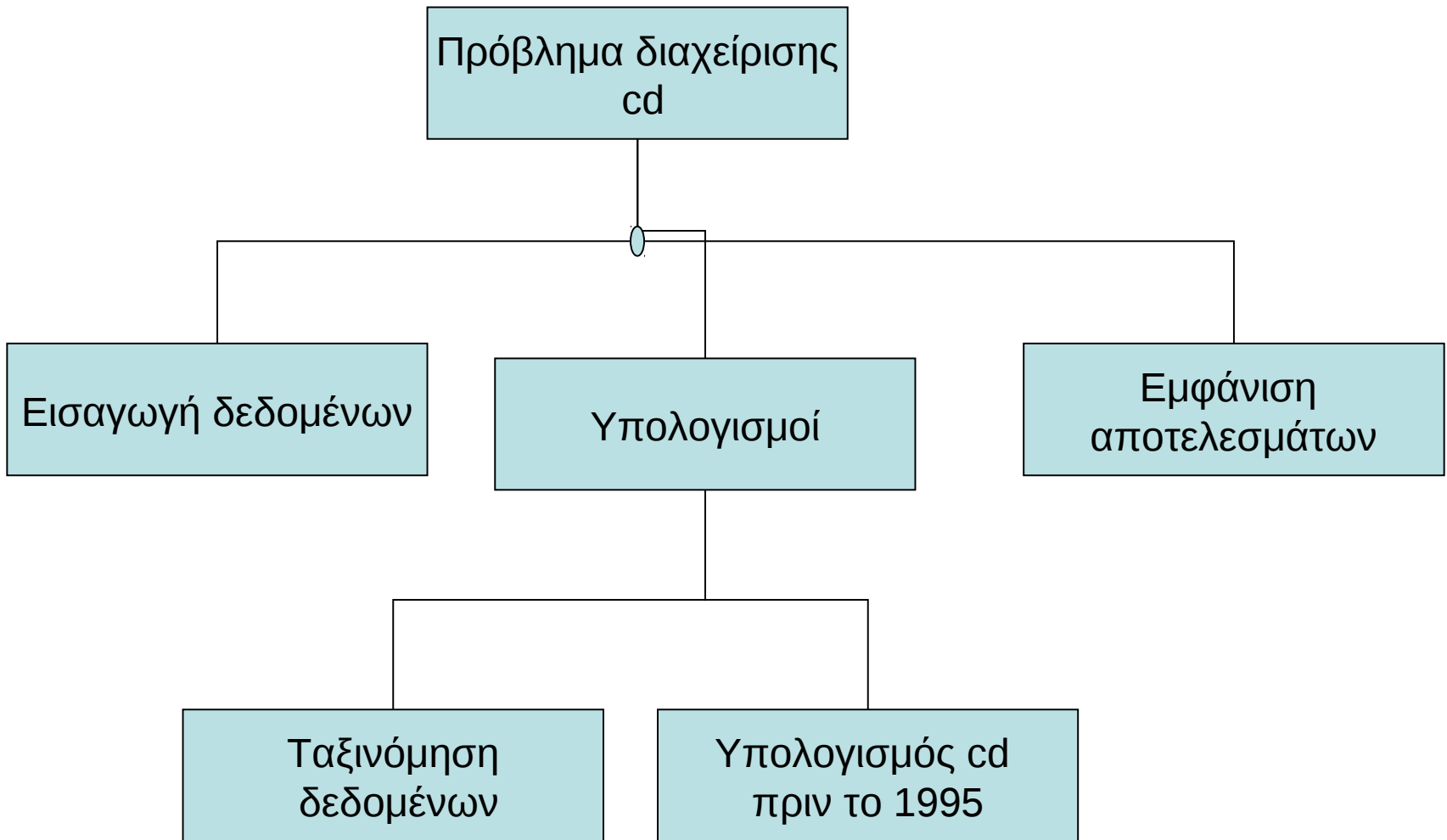
- Οι φυσικές γλώσσες εξελίσσονται συνεχώς.
- Οι Τεχνητές γλώσσες χαρακτηρίζονται από στασιμότητα. Οι κατασκευαστές όμως δημιουργούν νέες βελτιωμένες εκδόσεις τεχνητών γλωσσών ώστε να ακολουθήσουν τις τεχνολογικές εξελίξεις και να καλύψουν μεγαλύτερο εύρος εφαρμογών.

## 6.4 Τεχνικές σχεδίασης προγραμμάτων

### **Ιεραρχική σχεδίαση ή σχεδίαση Top-Down**

Χρησιμοποιεί τη στρατηγική της συνεχούς διαίρεσης του προβλήματος σε απλούστερα υποπροβλήματα.

**Τμηματικός προγραμματισμός** είναι η υλοποίηση του προγράμματος από την Ιεραρχική σχεδίαση. Κάθε υποπρόβλημα αποτελεί μία ανεξάρτητη ενότητα η οποία γράφεται ξεχωριστά από τα υπόλοιπα τμήματα του προγράμματος.





Ο **Δομημένος προγραμματισμός** χρησιμοποιεί μόνο τις τρεις βασικές δομές της ακολουθίας, της επανάληψης και της επιλογής. Όλα τα προγράμματα μπορούν να γραφούν χρησιμοποιώντας μόνο αυτές καθώς και συνδυασμούς αυτών των δομών.



Δημιουργία απλούστερων προγραμμάτων.



Άμεση μεταφορά των αλγορίθμων σε προγράμματα.



Διευκόλυνση της ανάλυσης του προγράμματος σε τμήματα.



Περιορισμός των λαθών κατά την ανάπτυξη του προγράμματος.



Διευκόλυνση στην ανάγνωση και την κατανόηση του προγράμματος από άλλους.



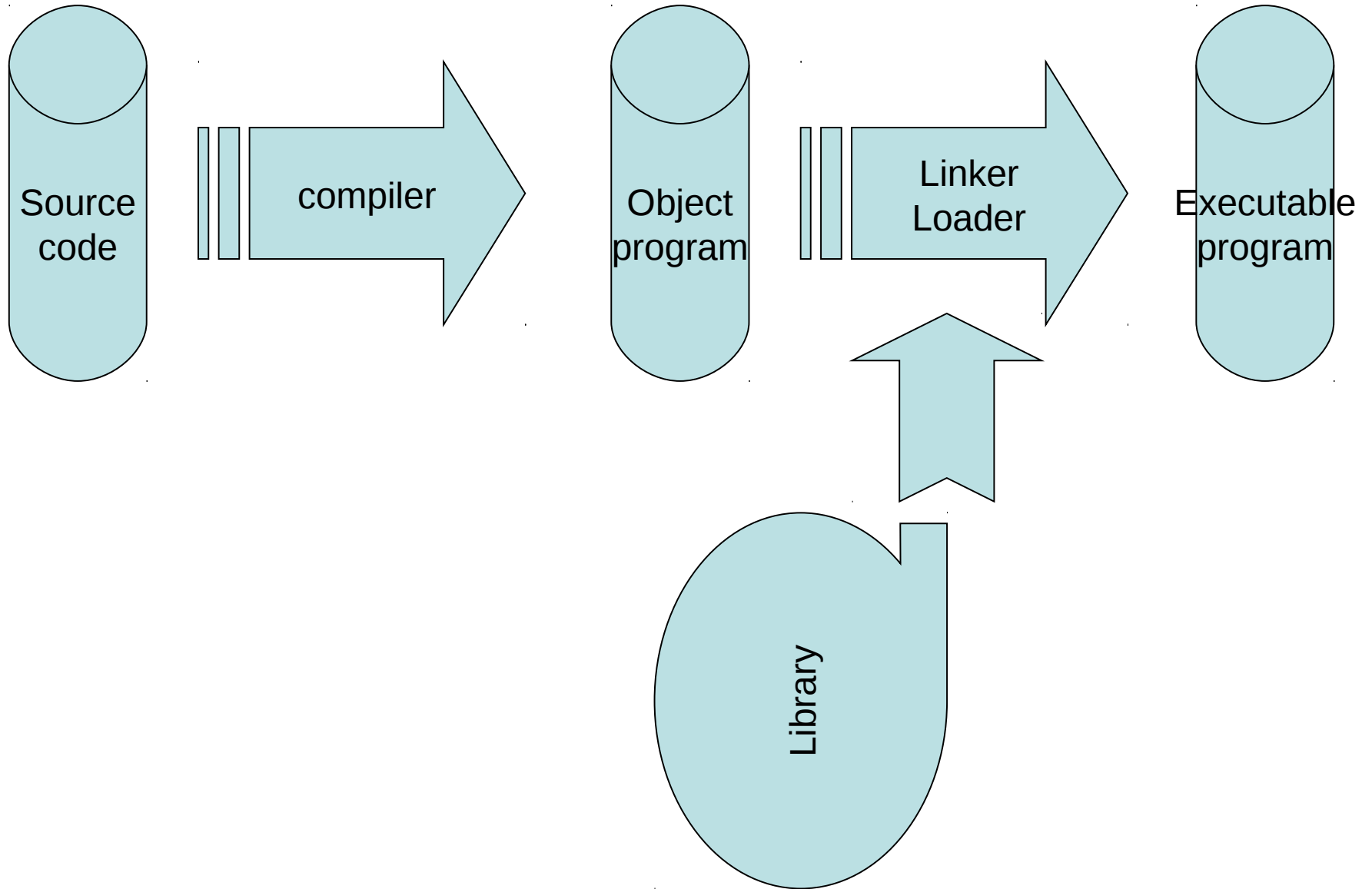
Ευκολότερη διόρθωση και συντήρηση.

## 6.7 Προγραμματιστικά περιβάλλοντα

- Συντάκτης (editor)
- Πηγαίος κώδικας (source code)
- Αντικείμενο πρόγραμμα (object program)
- Εκτελέσιμο πρόγραμμα (Executable)
- Μεταγλωττιστές (compilers)
- Διερμηνείς (Interpreters)
- Συνδετής – φορτωτής (Linker – Loader)
- Βιβλιοθήκες (Libraries)

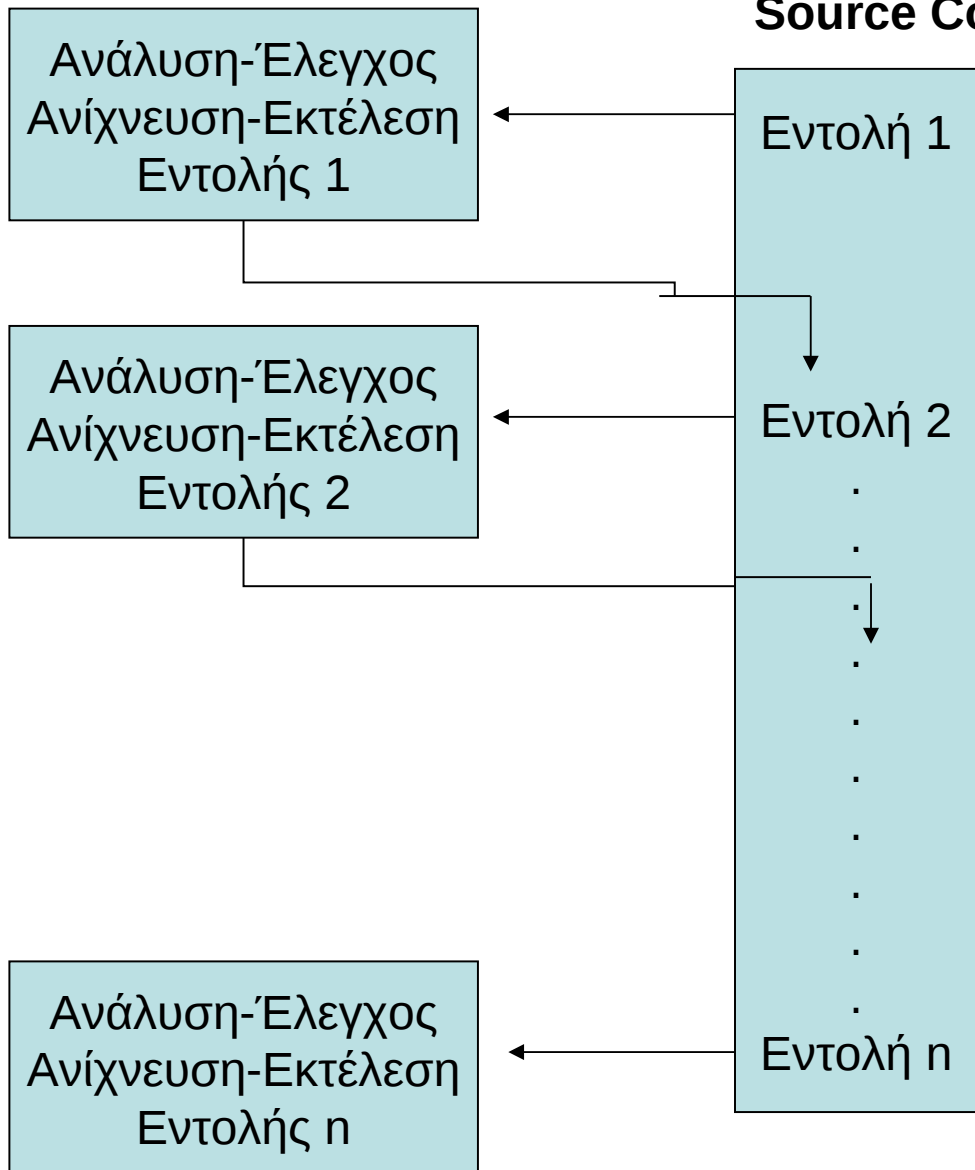


# ΜΕΤΑΓΛΩΤΤΙΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ



## Interpreter

## Source Code



```

//Ορισμός κλάσης αντικειμένου
Class Rectangle{
    public : unsigned width, height ;
    void setsize(unsigned x, unsigned y)
    unsigned area();
}
// Ορισμός μεθόδου του αντικειμένου
Void Rectangle::setsize(unsigned x, unsigned y) {
    width=x;
    height=y;
}
// Ορισμός μεθόδου του αντικειμένου
Unsigned Rectangle::area() {
    return (width * height);
}
// Κύριο πρόγραμμα
Int main() {
    unsigned A=25;
    unsigned B=14;
    Rectangle Tetragwno, parallilogramo;
    Tetragwno.setsize(A, B);
    parallilogramo.setsize(4,10);
    std::cout << Tetragwno.area() << "\n";
    std::cout << parallilogramo.area();
    return 0
}

```