

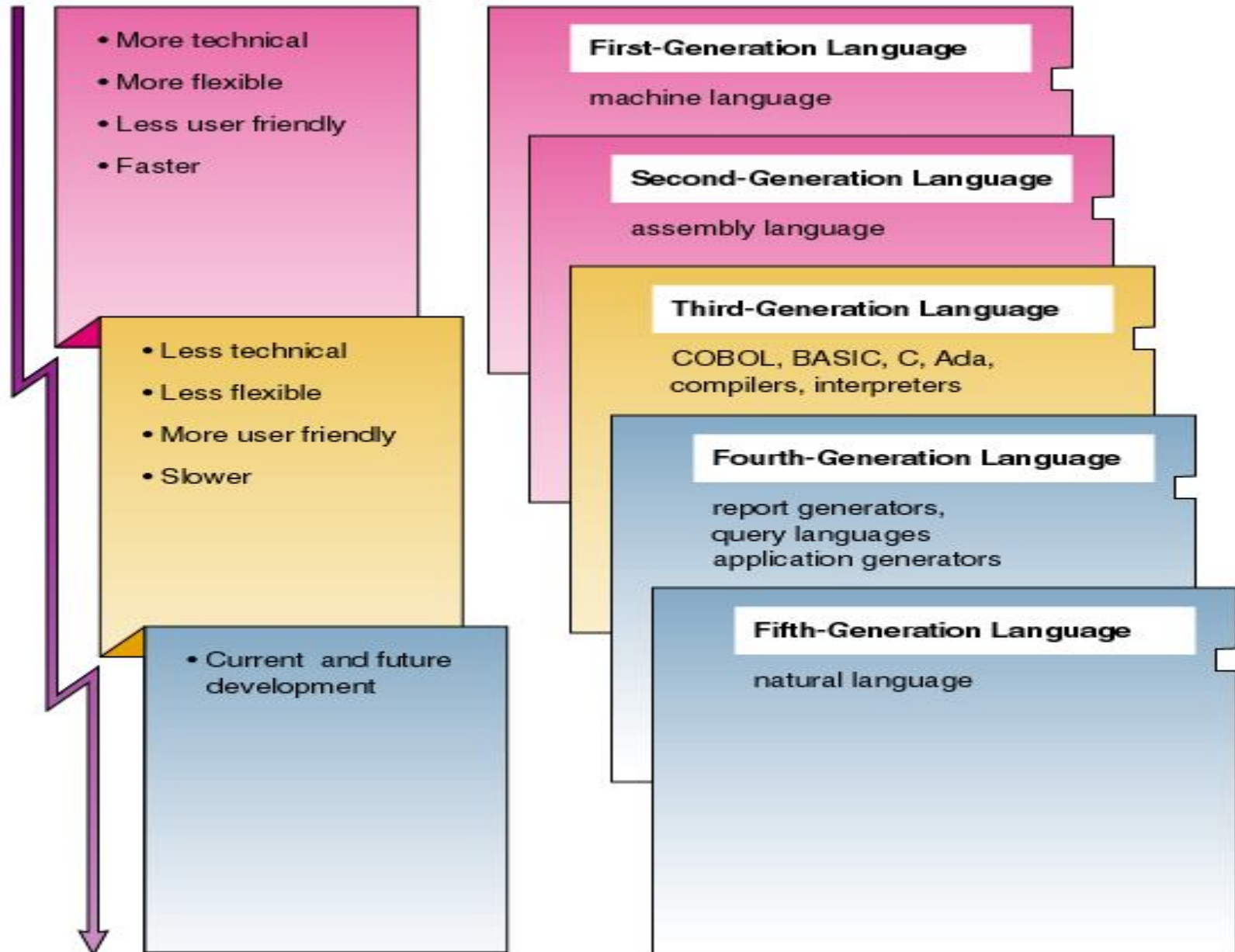
Προγραμματισμός Η/Υ



6η Διάλεξη

Γενειές Γλωσσών Προγραμματισμού

2.2



Οι 3 γενιές γλωσσών προγραμματισμού

2.3

Γλώσσα μηχανής (0 & 1)
υπερβολικά δύσκολος
προγραμματισμός

Συμβολική γλώσσα – χρήση
απλών εντολών
ADD, SUB. Ο Προγραμματισμός
παραμένει δύσκολος

COBOL, μία γλώσσα 3ης γενιάς
Χρησιμοποιεί αγγλικές λέξεις ως
εντολές

First generation
Machine language

```
11110010 01110011 1101 001000010000 0111 000000101011  
11110010 01110011 1101 001000011000 0111 000000101111  
11111100 01010010 1101 001000010010 1101 001000011101  
11110000 01000101 1101 001000010011 0000 000000111110  
11110011 01000011 0111 000001010000 1101 001000010100  
10010110 11110000 0111 000001010100
```



Second generation
Assembly language

```
PACK 210(8,13),02B(4,7)  
PACK 218(8,13),02F(4,7)  
MP 212(6,13),21D(3,13)  
SRP 213(5,13),03E(0),5  
UNPK 050(5,7),214(4,13)  
OI 054(7),XFO'
```



Third generation
COBOL

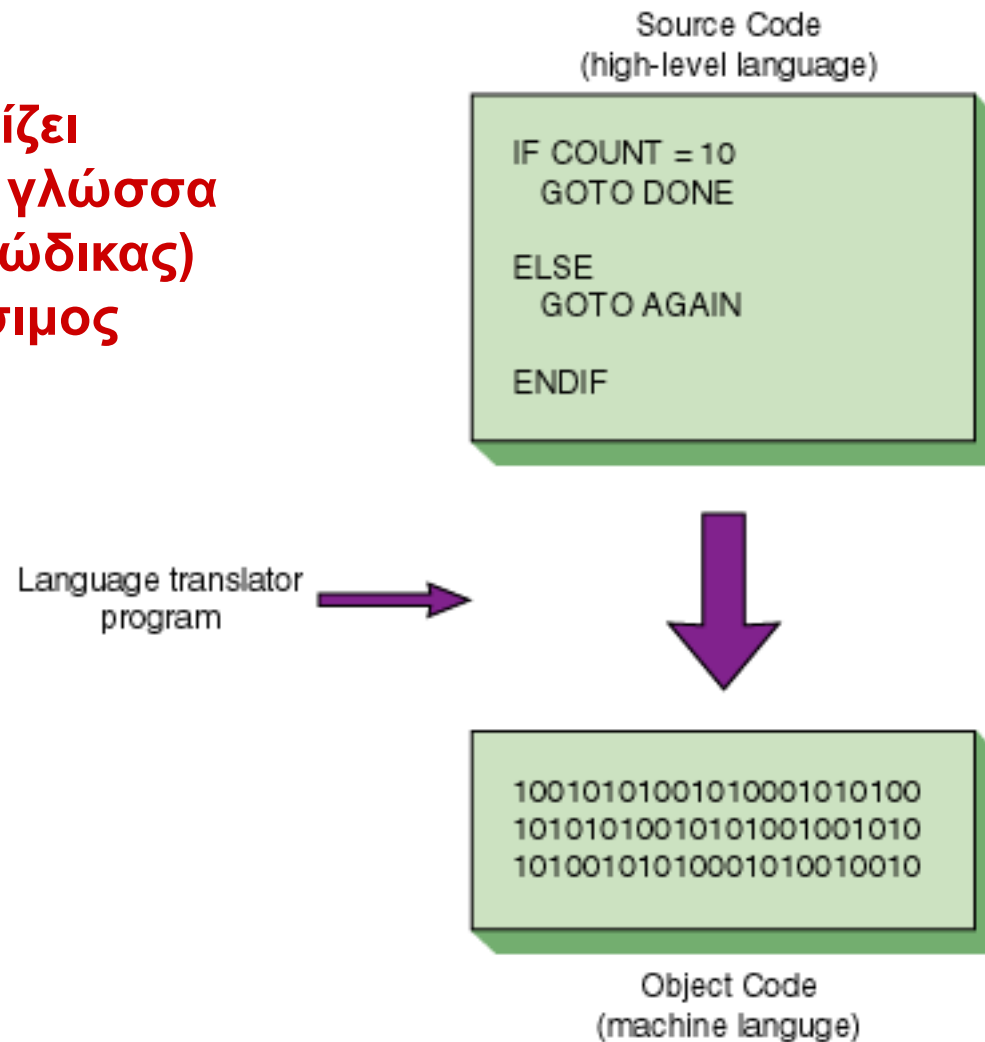
```
MULTIPLY HOURS-WORKED BY PAY-RATE GIVING GROSS-PAY ROUNDED
```

Η έννοια του Μεταφραστή

2.4

Quit

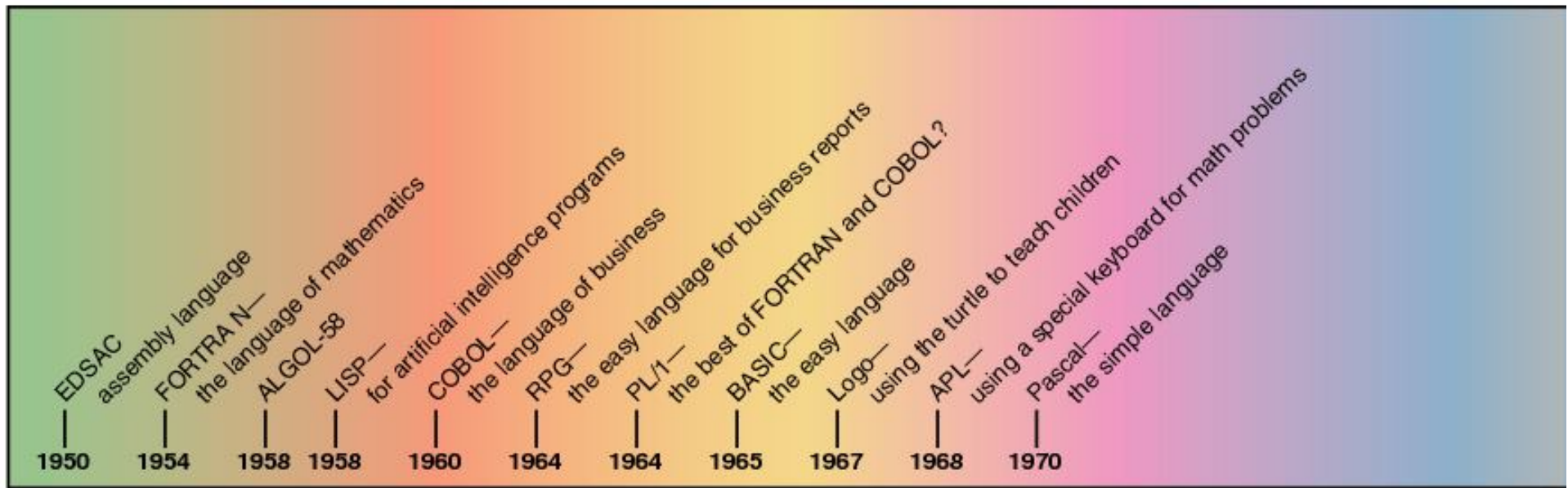
Λογισμικό το οποίο μεταγλωττίζει
ένα πρόγραμμα γραμμένο σε γλώσσα
Υψηλού επιπέδου (πηγαίος κώδικας)
σε γλώσσα μηχανής (εκτελέσιμος
κώδικας)



Η Ιστορική Εξέλιξη των Γλωσσών Προγραμματισμού

2.5

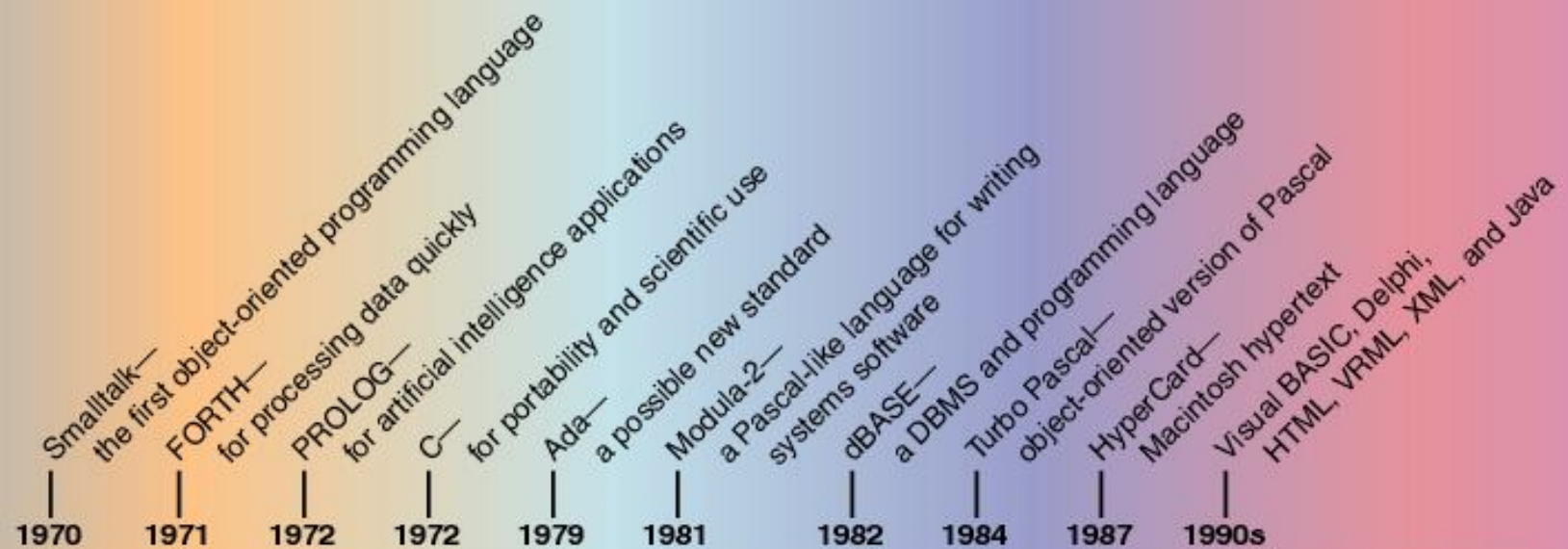
Quit



Η Ιστορική Εξέλιξη των Γλωσσών Προγραμματισμού

2.6

Quit



Παραδείγματα Γλωσσών Προγραμματισμού

2.7

Quit

FORTRAN

```
IF (XINVO .GT. 500.00) THEN
    DISCNT = 0.07 * XINVO
ELSE
    DISCNT = 0.0
ENDIF
XINVO = XINVO - DISCNT
```

COBOL

```
OPEN-INVOICE-FILE.
  OPEN I-O INVOICE FILE.

READ-INVOICE-PROCESS.
  PERFORM READ-NEXT-REC THROUGH READ-NEXT-REC-EXIT UNTIL END-OF-FILE.
  STOP RUN.

READ-NEXT-REC.
  READ INVOICE-REC
  INVALID KEY
    DISPLAY 'ERROR READING INVOICE FILE '
    MOVE 'Y' TO EOF-FLAG
    GOTO READ-NEXT-REC-EXIT.
  IF INVOICE-AMT > 500
    COMPUTE INVOICE-AMT = INVOICE-AMT - (INVOICE-AMT * .07)
    REWRITE INVOICE-REC.

READ-NEXT-REC-EXIT.
  EXIT.
```

Όταν το ποσό είναι μεγαλύτερο των 500 Ευρώ υπάρχει έκπτωση 7%.
Αλλιώς δεν υπάρχει έκπτωση

BASIC

```
10 REM This Program Calculates a Discount Based on the Invoice Amount
20 REM If Invoice Amount is Greater Than 500, Discount is 7%
30 REM Otherwise Discount is 0
40 REM
50 INPUT "What is the Invoice Amount"; INV.AMT
60 IF INV.AMT > 500 THEN LET DISCOUNT = .07 ELSE LET DISCOUNT = 0
70 REM Display results
80 PRINT "Original Amt", "Discount", "Amt after Discount"
90 PRINT INV.AMT, INV.AMT * DISCOUNT, INV.AMT - INV.AMT * DISCOUNT
100 END
```

Pascal

```
if INVOICEAMOUNT > 500.00 then
    DISCOUNT := 0.07 * INVOICEAMOUNT
else
    DISCOUNT := 0.0;
INVOICEAMOUNT := INVOICEAMOUNT - DISCOUNT
```

C

```
if (invoice_amount > 500.00)
    DISCOUNT = 0.07 * invoice_amount;
else
    discount = 0.00;
invoice_amount = invoice_amount - discount;
```


2.9

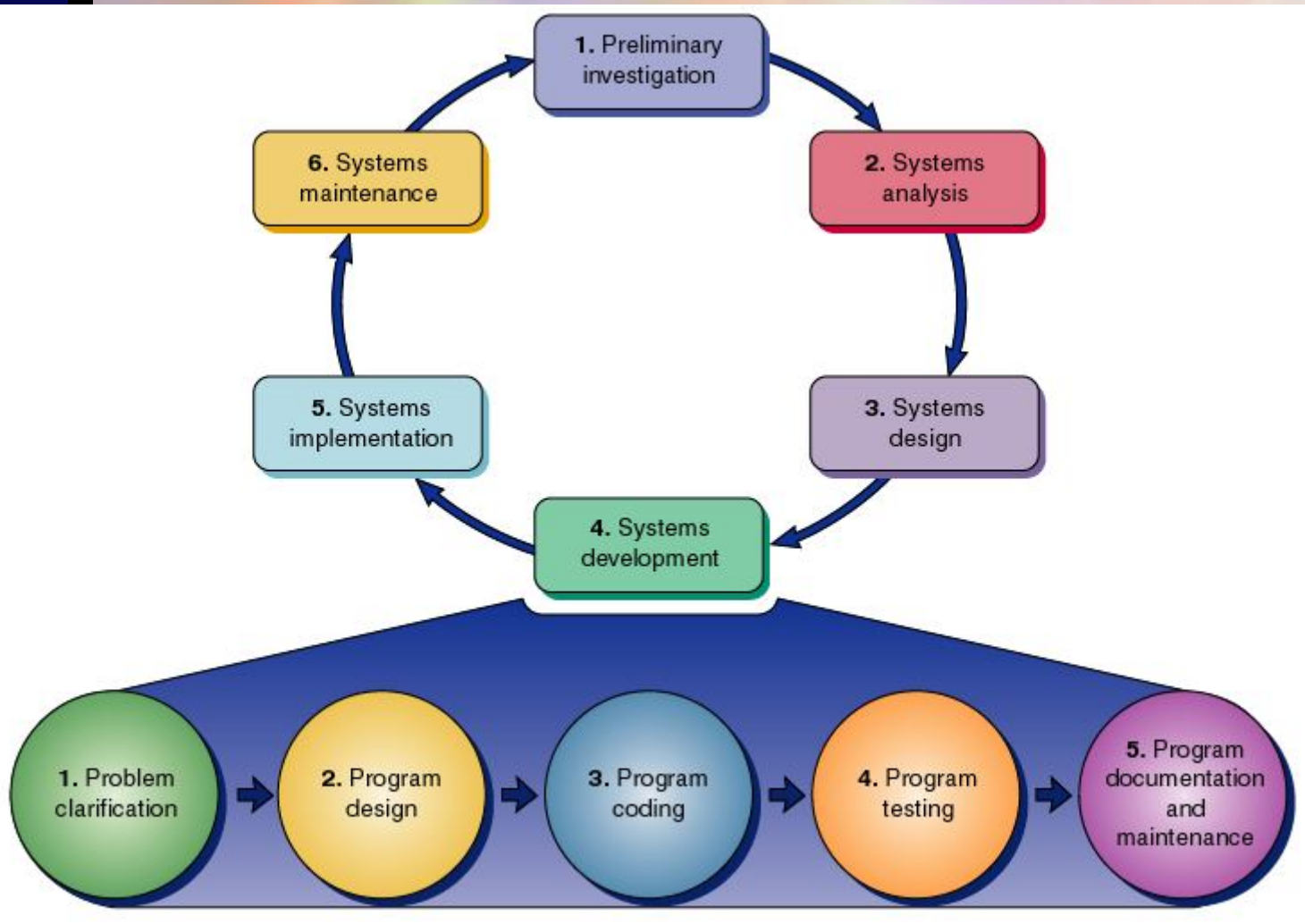
Quit

Ανάπτυξη Λογισμικού

Τα 5 βήματα ανάπτυξης προγραμμάτων

2.10

Quit



Βήμα 1ο : Κατανόηση προβλήματος

2.11

Quit



1. Τι ζητάει το πρόβλημα
2. Τι εισόδους δέχεται
3. Τι έξοδο απαιτείται να παράγει
4. Μπορεί να υλοποιηθεί;

Βήμα 2ο : Σχεδίαση/γραφική αναπαράσταση προγράμματος

2.12

Quit

1. Εύρεση λογικής προγράμματος
2. Εύρεση Αλγορίθμου
3. Γραφική αναπαράσταση ή κωδικοποίηση με χρήση ψευδογλώσσας



2.13

Quit

START

DO WHILE (so long as) there are records

 Read a customer billing account record

 IF today 's date is greater than 30 days from
 date of last customer payment

 Calculate total amount due

 Calculate 5% interest on amount due

 Add interest to total amount due to calculate
 grand total

 Print on invoice overdue amount

 ELSE

 Calculate total amount due

 ENDIF

 Print out invoice

END DO

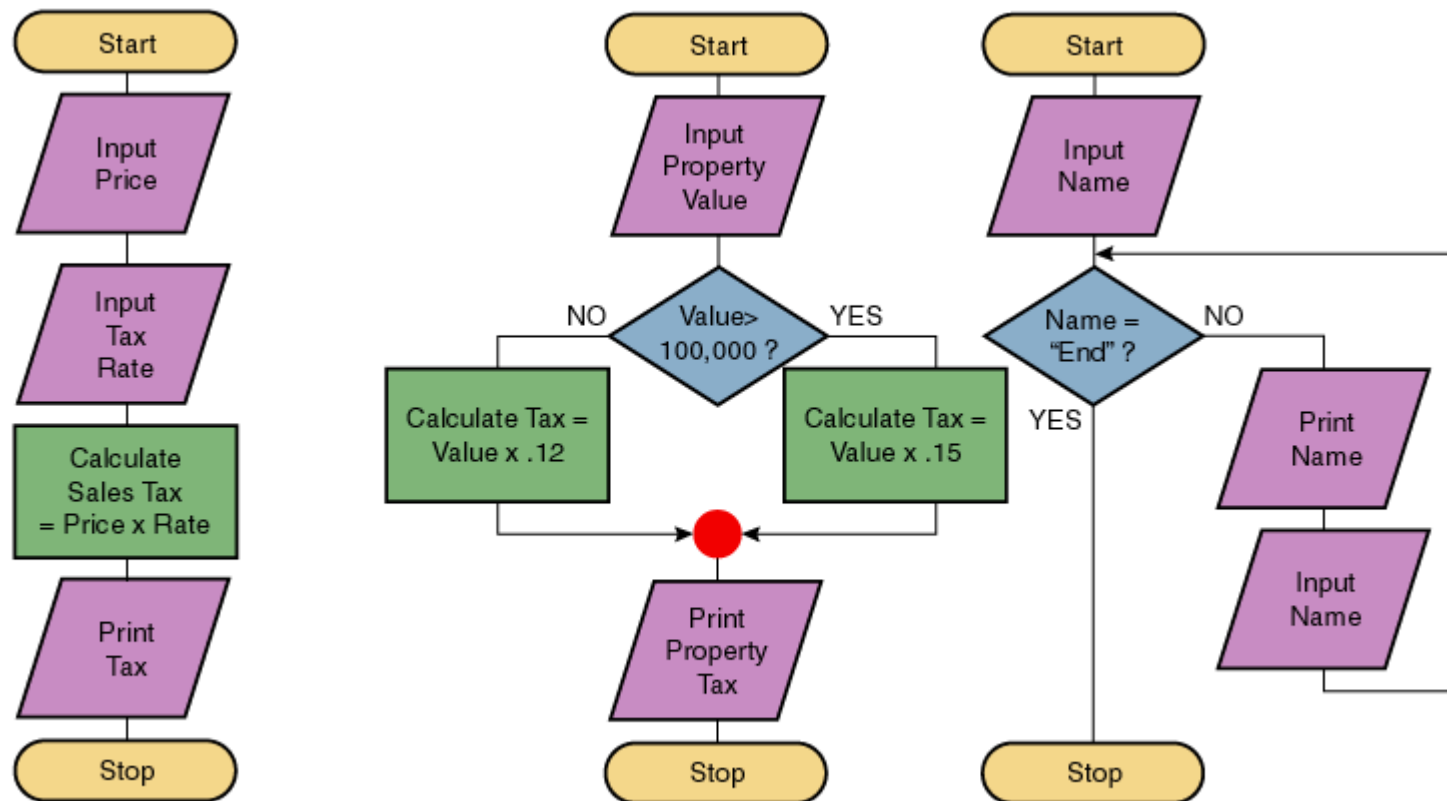
END

Διαγράμματα Ροής

2.14

Quit







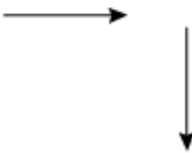
Παραδείγματα:



Περιγραφή των Συμβόλων σε ένα διάγραμμα ροής

2.15

Quit

Symbol	Name	Use
	Terminal	Indicates the beginning and end of a program.
	Process	A calculation or assigning of a value to a variable.
	Input/Output (I/O)	Any statement that causes data to be input to a program (INPUT, READ) or output from the program, such as printing on the display screen or printer.
	Decision	Program decisions. Allows alternate courses of action based on a condition. A decision indicates a question that can be answered <i>yes</i> or <i>no</i> (or <i>true</i> or <i>false</i>).
	Predefined Process	A group of statements that together accomplish one task. Used extensively when programs are broken into modules.
	Connector	Can be used to eliminate lengthy flowlines. Its use indicates that one symbol is connected to another.
	Flowlines and Arrowheads	Used to connect symbols and indicate the sequence of operations. The flow is assumed to go from top to bottom and from left to right. Arrowheads are only required when the flow violates the standard direction.

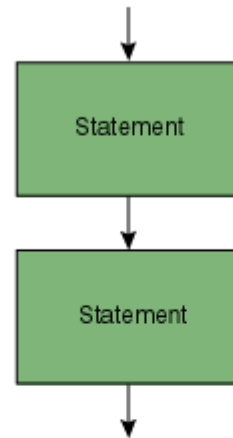
Οι τρεις δομές ελέγχου

2.16

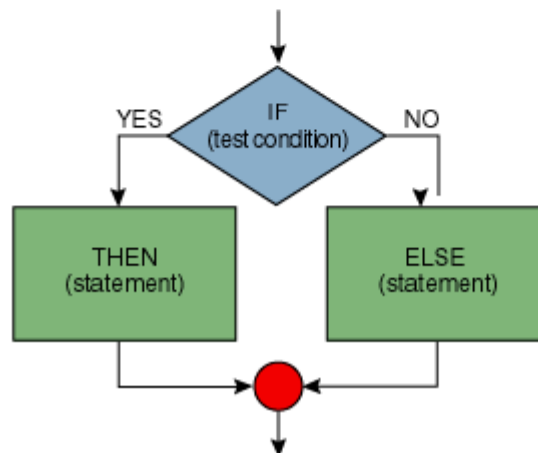
Quit

Οι τρεις δομές ελέγχου:
Ακολουθία, επιλογή και επανάληψη

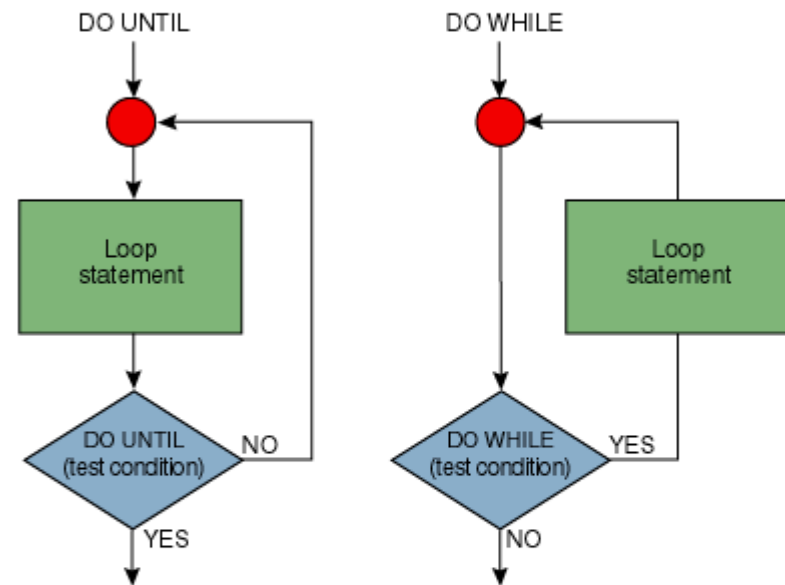
Sequence control structure
(one program statement follows another in logical order)



Selection control structure
(IF-THEN-ELSE)



Iteration control structures:
DO UNTIL and DO WHILE



Το 3ο Βήμα: Κωδικοποίηση

2.17

Quit

1. Επιλογή γλώσσας υψηλού επιπέδου.
2. κωδικοποίηση στη γλώσσα αυτή.



Το 4ο Βήμα: Έλεγχος του Προγράμματος

2.18

Quit

1. Έλεγχος και απομάκρυνση λαθών.
2. εκτέλεση προγράμματος με απλά δεδομένα.
3. εκτέλεση προγράμματος με αληθινά δεδομένα

7/9

0800 Action started
1000 stopped - action ✓
1300 1000 HP - MC
1300 1000 PRG 2 12047675
1300 1000 Const 2.13067405
Relays are on ass failed special speed test
in relay
Relays changed
1120 Started Cassio Tape (Sine check)
1500 Checked Radio Receiver Test
1545 Relay to Panel F
(Moth in relay)



First actual case of bug being found.
1130 in relay started.
1200 closed form

Το 5ο Βήμα: Τεκμηρίωση/Συντήρηση του Προγράμματος

2.19

Quit

1. Συγγραφή εγχειριδίων για τον χρήστη.
2. Συγγραφή εγχειριδίων για τον προγραμματιστή
3. Συντήρηση προγράμματος.

