ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά
Τεχνολογικού Τομέα

ανοικτάμαθήματα
opencourses

Εισαγωγή στην Πληροφορική & τον Προγραμματισμό

Ενότητα 4η: Αλγόριθμος & Πρόγραμμα

Ι. Ψαρομήλιγκος – Χ. Κυτάγιας
Τμήμα Διοίκησης Επιχειρήσεων

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά Τεχνολογικού Τομέα**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σκοποί ενότητας

Στην συγκεκριμένη ενότητα παρουσιάζονται ορισμένα θέματα που αφορούν την έννοια του αλγόριθμου, της απόδοσης ή πολυπλοκότητάς του, της μετάβασης από τον αλγόριθμο σε πρόγραμμα γραμμένο σε μια γλώσσα προγραμματισμού καθώς και της διαδικασίας μεταγλώττισης και δημιουργίας του τελικού εκτελέσιμου προγράμματος από τον Η/Υ.

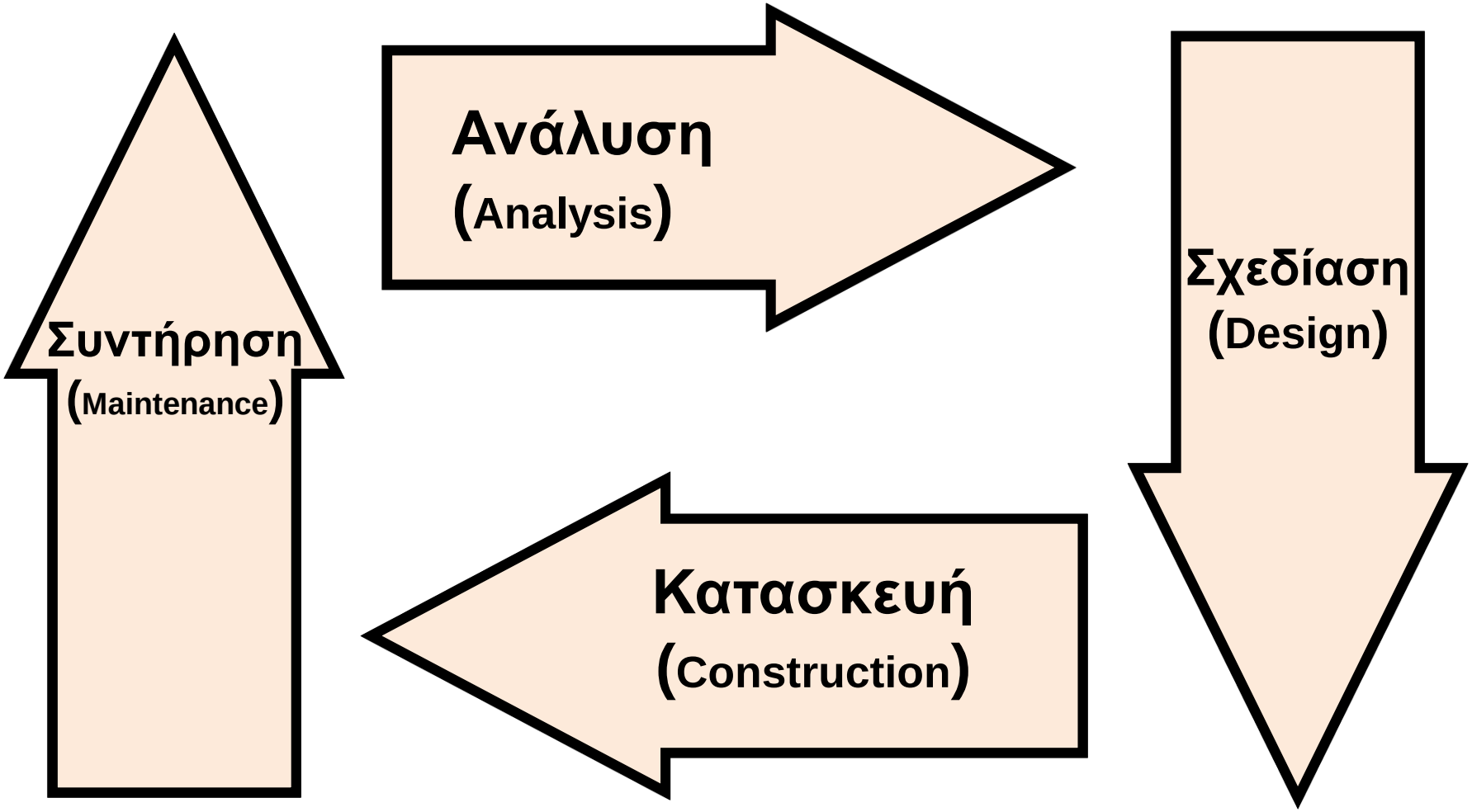
Περιεχόμενα

- Προγραμματισμός & Κύκλος Ζωής Λογισμικού
- Η έννοια του Αλγόριθμου
- Αλγοριθμική γλώσσα
- Θέματα απόδοσης αλγόριθμου
- Πρόγραμμα σε Γλώσσα Προγραμματισμού
- Μεταγλωττιστές – Διερμηνείς
- Βασικά συστατικά ενός προγράμματος

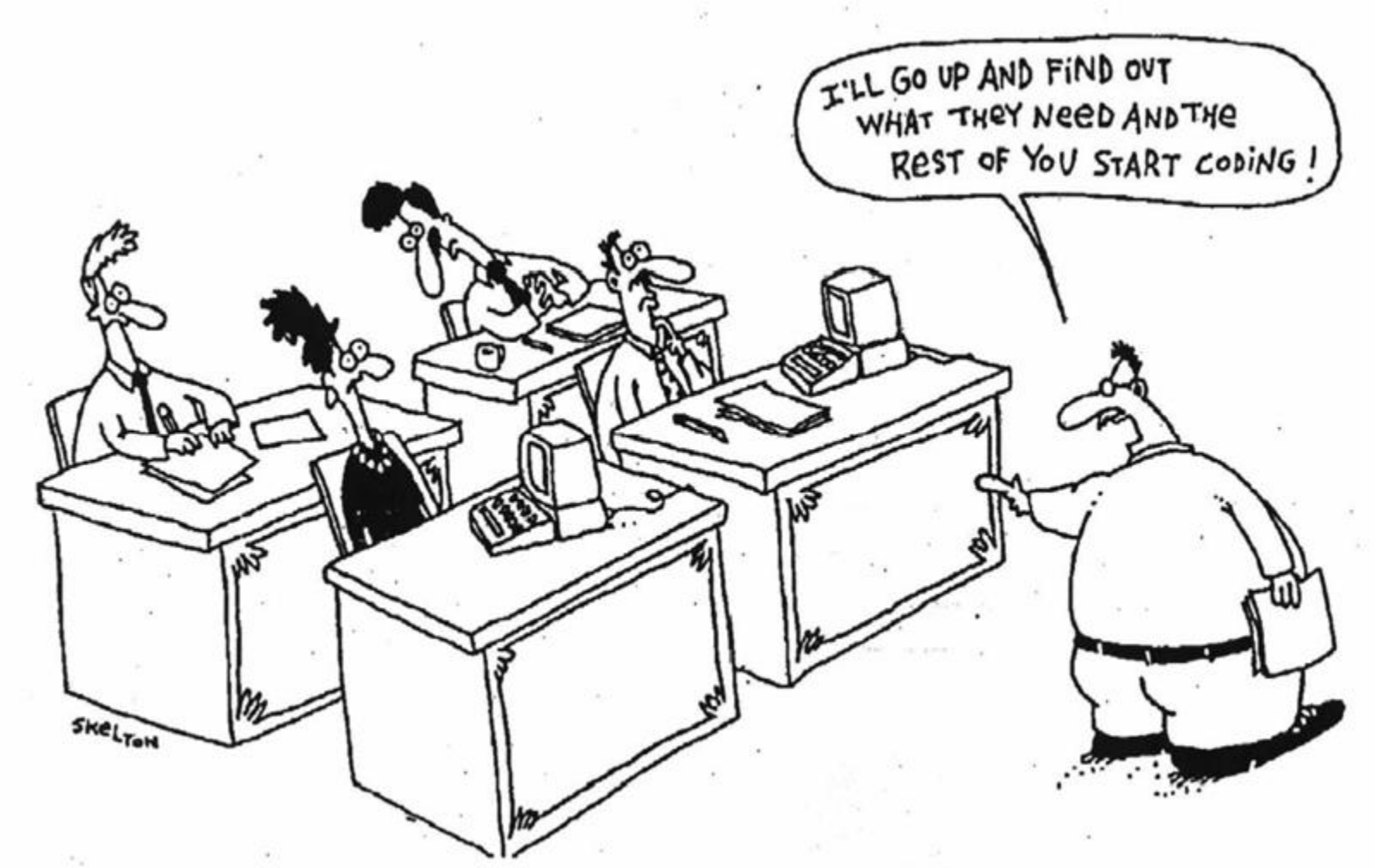
Θυμόμαστε ότι …

- Η **εργασία ανάπτυξης λογισμικού** είναι η εργασία που γίνεται κατά την κατασκευή των **λογισμικών συστημάτων** (software systems)
- Τα λογισμικά συστήματα **αυτοματοποιούν** (με τη βοήθεια του Η/Υ) την εκτέλεση εργασιών που περιλαμβάνουν **δεδομένα** (**data**) (επεξεργασία ή/και παραγωγή νέων δεδομένων)
- Η μηχανή που **εκτελεί** αυτόματα τις εργασίες αυτές είναι ο Ηλεκτρονικός Υπολογιστής

Θυμόμαστε… τις βασικές εργασίες κατά την Ανάπτυξη Λογισμικού



Μια άποψη για την εργασία του Προγραμματισμού



Προγραμματισμός

Προγραμματιστής

Εντολές σε Γλώσσα Προγραμματισμού

Προγράμματα Μετάφρασης
Γλώσσες Προγραμματισμού

Η/Υ

Εντολές σε Γλώσσα Μηχανής

Αλγόριθμος

- Ένας τρόπος για να περιγράψουμε με λεπτομέρειες τις εργασίες – ενέργειες που πρέπει να γίνουν για να επιλύσουμε ένα πρόβλημα είναι η χρήση Αλγορίθμων
- Στα μαθηματικά έχουμε χρησιμοποιήσει πολλές φορές τον όρο αυτό για να περιγράψουμε τη λύση ενός προβλήματος πχ Αλγόριθμος του Ευκλείδη
- **Αλγόριθμος** είναι μια συστηματική διαδικασία που περιλαμβάνει αυστηρά καθορισμένα βήματα για την επίλυση ενός προβλήματος σε ένα πεπερασμένο αριθμό βημάτων
  - Διαδικασία βήμα-βήμα
  - Πεπερασμένος αριθμός βημάτων

Αλγόριθμος – Παράδειγμα (1)

Αλγόριθμος για την κατασκευή ενός φλυτζανιού στιγμιαίου καφέ χωρίς ζάχαρη

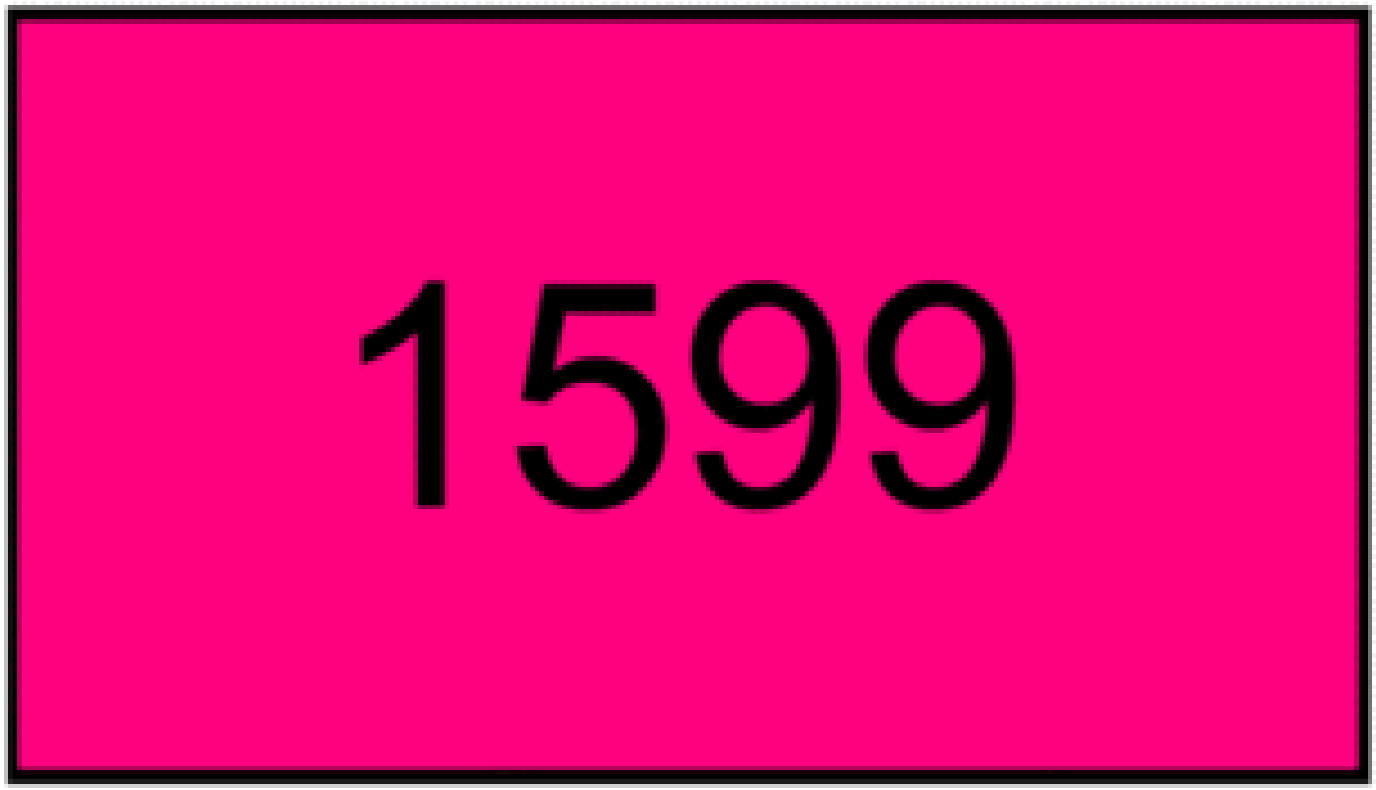
- **Βήμα 1:** Βάλε το μπρίκι κάτω από τη βρύση
- **Βήμα 2:** Περίμενε μέχρι να γεμίσει το μπρίκι
- **Βήμα 3:** Κλείσε τη βρύση
- **Βήμα 4:** Άναψε την κουζίνα;
- **Βήμα 5:** Περίμενε μέχρι να βράσει το νερό
- **Βήμα 6:** Σβήσε την κουζίνα
- **Βήμα 7:** Πάρε το δοχείο του στιγμιαίου καφέ από το ράφι
- **Βήμα 8:** Βγάλε το καπάκι από το δοχείο του στιγμιαίου καφέ
- **Βήμα 9:** Βγάλε ένα κουταλάκι του καφέ
- **Βήμα 10:** Βάλε το κουταλάκι του καφέ στο φλυτζάνι
- **Βήμα 11:** Βάλε το καπάκι στο δοχείο του στιγμιαίου καφέ
- **Βήμα 12:** Ξαναβάλε το δοχείο του στιγμιαίου καφέ στο ράφι
- **Βήμα 13:** Βάλε νερό στο φλυτζάνι μέχρι να γεμίσει

Αλγόριθμος - Παράδειγμα (2)

Αλγόριθμος του Ευκλείδη (ΜΚΔ των α, β)

- **Βήμα 1:** Βρες το πηλίκο **Π = α / β**
- **Βήμα 2:** Βρες το υπόλοιπο **Υ = α - (Π * β)**
- **Βήμα 3:** Θέσε στο **α = β**
- **Βήμα 4:** Θέσε στο **β = Υ**
- **Βήμα 5:** Αν το υπόλοιπο **Υ** δεν είναι **0**, πήγαινε στο **Βήμα 1**
- **Βήμα 6:** Ο ΜΚΔ είναι ο **α**

Αλγόριθμος του Ευκλείδη



ΜΚΔ(1599,650)

1599 = **650*2 + 299**
650 = **299*2 + 52**
299 = 52*5 + 39
52 = 39*1 + 13
39 = 13*3 + 0

http://en.wikipedia.org/wiki/Algorithm#/media/File:Euclids-algorithm-example-1599-650.gif

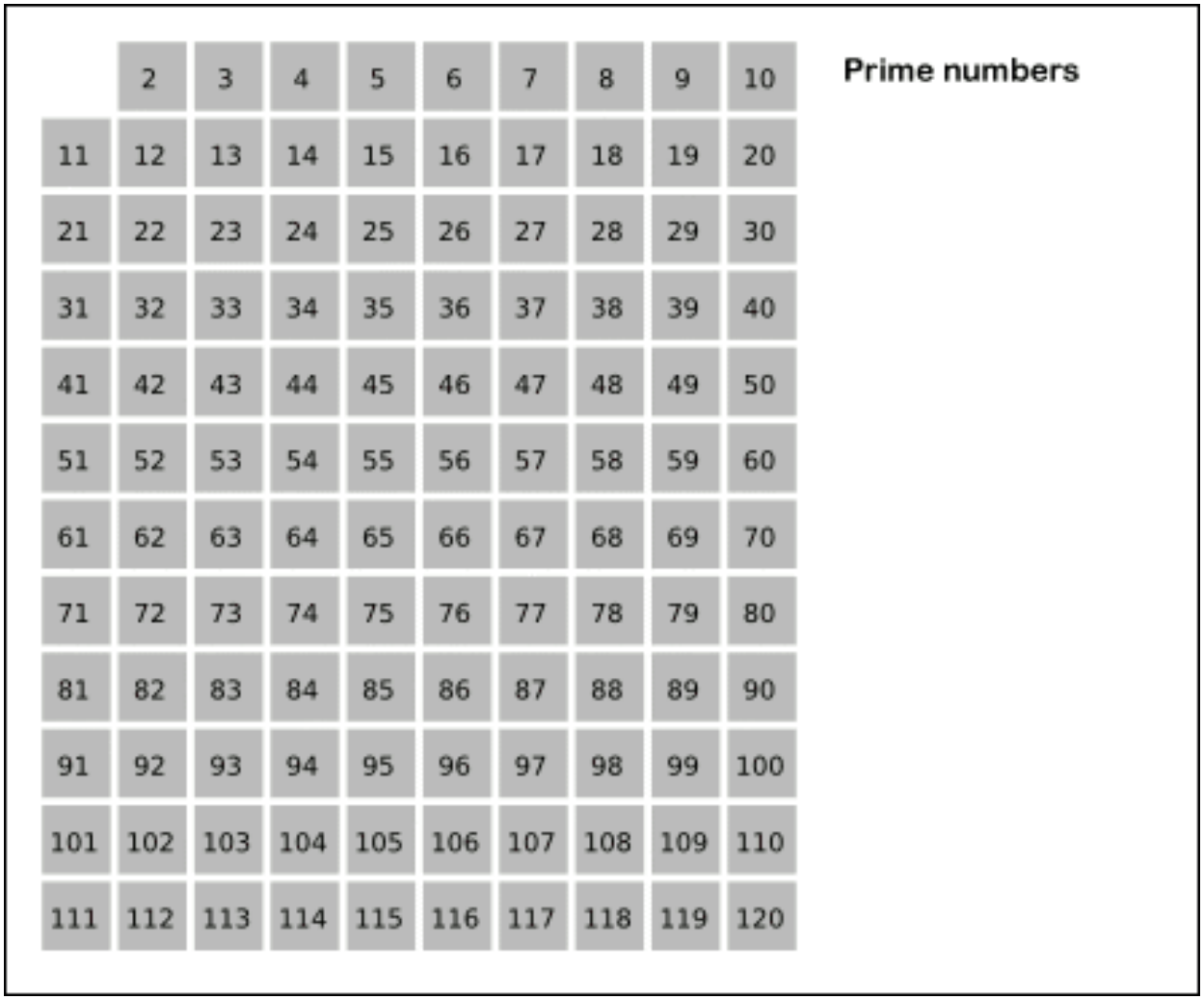
Κόσκινο του Ερατοσθένη

Η εύρεση όλων των **πρώτων αριθμών** που είναι **μικρότεροι ή ίσοι** από έναν ακέραιο ***n***, γίνεται ως εξής:

1. Δημιουργούμε μια λίστα από διαδοχικούς ακέραιους από το 2 μέχρι το *n*: **(2, 3, 4, ..., *n*)**
2. Αρχικά, έστω ότι το **p** είναι ίσο με **2**, τον 1ο πρώτο αριθμό.
3. Διαγράφουμε από τη λίστα όλα τα πολλαπλάσια του p που είναι μικρότερα ή ίσα με n. **(2p, 3p, 4p, κτλ)**
4. Βρίσκουμε τον 1ο αριθμό που απομένει στη λίστα μετά τον p (αυτός ο αριθμός είναι ο επόμενος πρώτος αριθμός) και αντικαθιστούμε το p με αυτόν τον αριθμό.
5. Επαναλαμβάνουμε τα βήματα 3 και 4 μέχρι το p^2 να είναι μεγαλύτερο από n.
6. Όλοι οι αριθμοί που απομένουν στη λίστα είναι πρώτοι αριθμοί.

http://el.wikipedia.org/wiki/%CE%9A%CF%8C%CF%83%CE%BA%CE%B9%CE%BD%CE%BF_%CF%84%CE%BF%CF%85_%CE%95%CF%81%CE%B1%CF%84%CE%BF%CF%83%CE%B8%CE%AD%CE%BD%CE%B7

Κόσκινο του Ερατοσθένη (2)



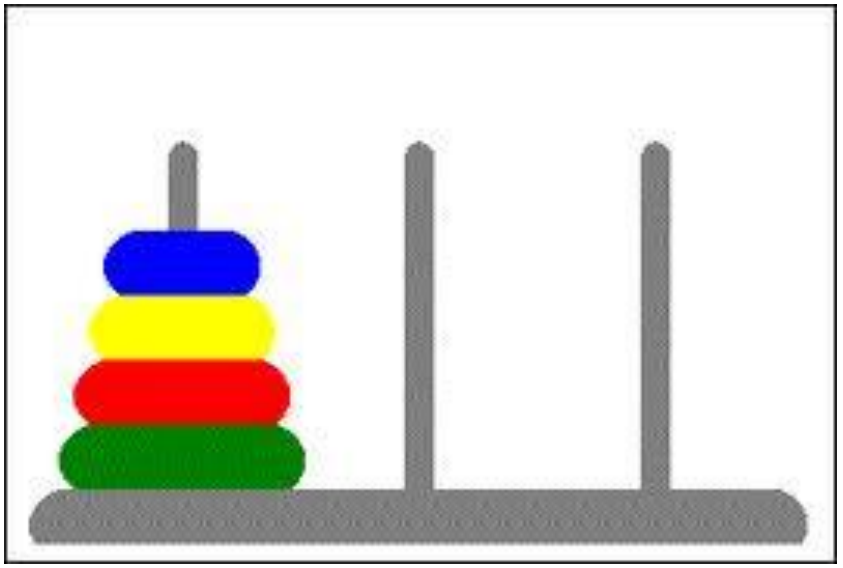
	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

http://el.wikipedia.org/wiki/%CE%9A%CF%8C%CF%83%CE%BA%CE%B9%CE%BD%CE%BF_%CF%84%CE%BF%CF%85_%CE%95%CF%81%CE%B1%CF%84%CE%BF%CF%83%CE%B8%CE%AD%CE%BD%CE%B7

6-disc Tower of Hanoi.mp4

Αλγόριθμος - Παράδειγμα (3)

Οι Πύργοι του Annoi



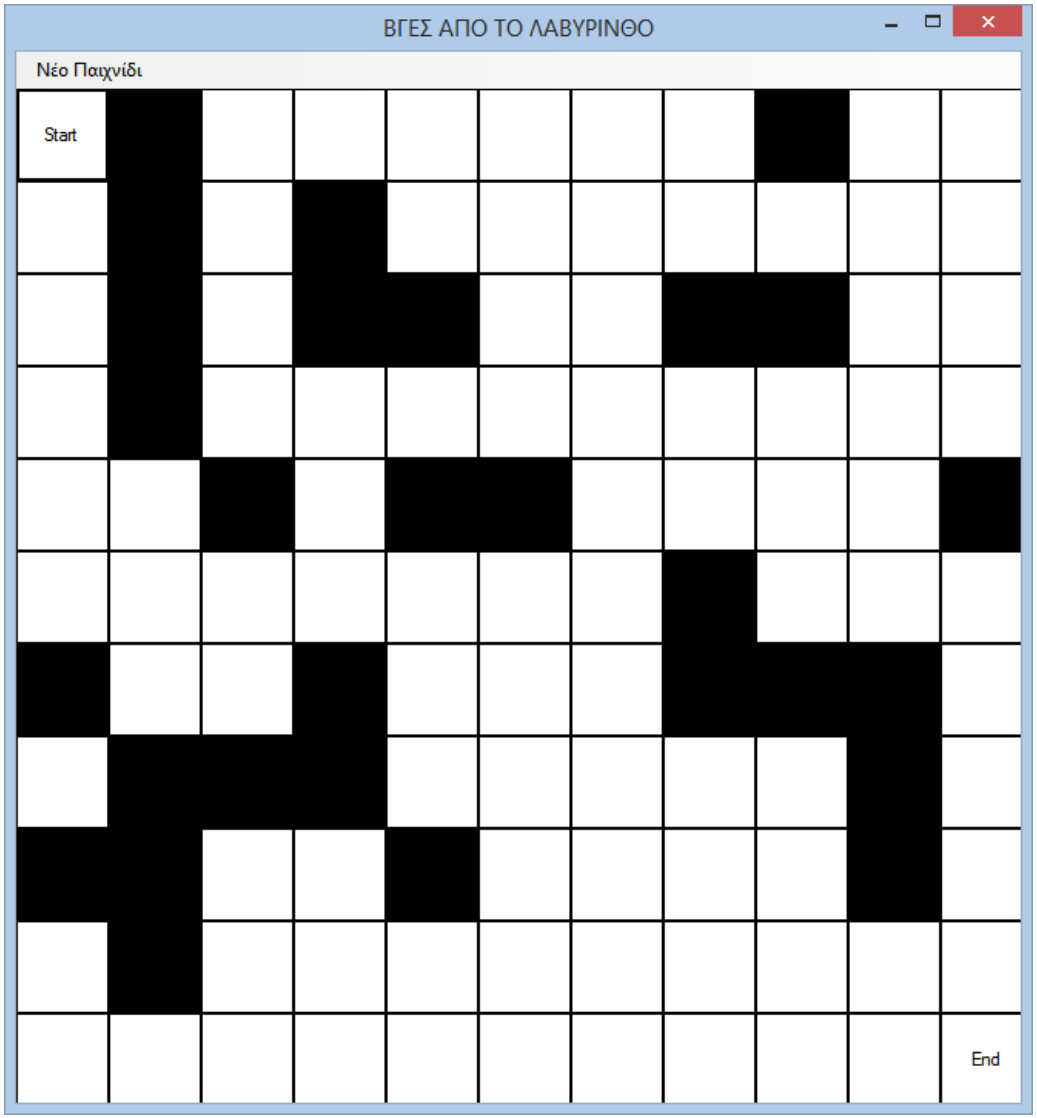
Επινοήθηκε από τον Γάλλο μαθηματικό François Édouard Anatole Lucas to 1883.

Ο θρύλος λέει ότι στην Ινδία στο μεγάλο ναό του Βράχμα στις Μπενάρες, πάνω σ' ένα μπρούτζινο βάθρο που είναι το κέντρο του κόσμου βρίσκονται 64 χρυσοί δίσκοι που σχηματίζουν ένα πύργο.

Οι δίσκοι μπορούν να μεταφερθούν σε οποιοδήποτε άξονα αλλά μόνο ένας δίσκος κάθε φορά. Κανένας δίσκος δεν μπορεί να τοποθετηθεί πάνω από μικρότερό του. Αρχικά, όταν ο Βράχμα δημιούργησε τον κόσμο όλοι οι δίσκοι ήταν περασμένοι στον ίδιο άξονα.

Σήμερα, με την μεταφορά των δίσκων από τους ιερείς βρισκόμαστε περίπου στα μισά της προσπάθειας. … *Όταν όλοι οι δίσκοι έχουν μεταφερθεί στον τρίτο άξονα θα έρθει το τέλος του κόσμου και όλα θα γίνουν σκόνη*.

Αλγόριθμος - Παράδειγμα (4)

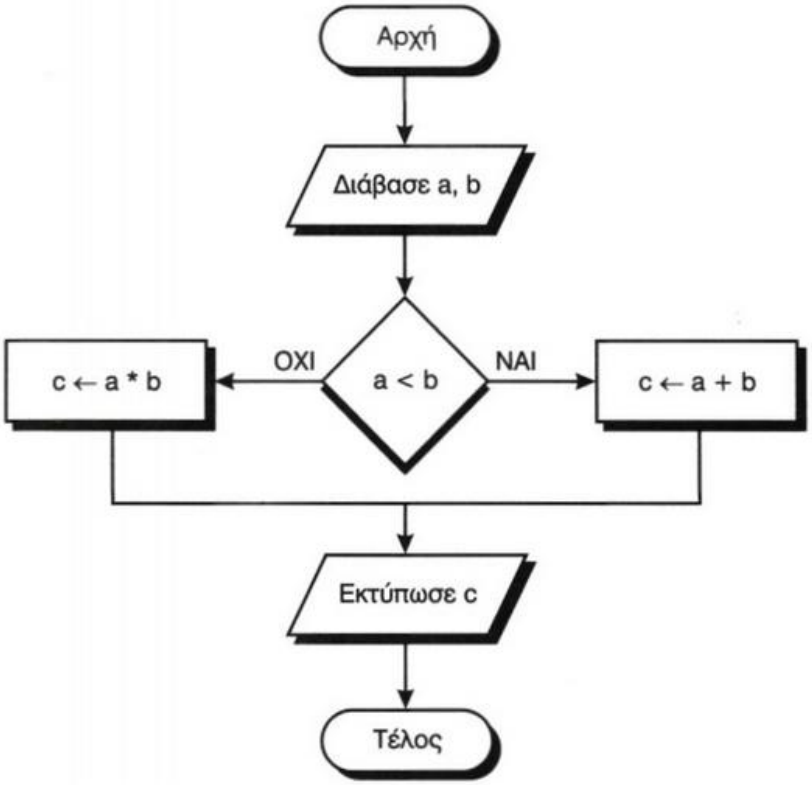


Περιγραφή Αλγόριθμου

- **Ελεύθερο κείμενο** (ο πιο αδόμητος τρόπος…)
- **Διαγραμματικές τεχνικές** (πχ διαγράμματα ροής - flow charts)
- **Φυσική γλώσσα** με βήματα
- **Ψευδογλώσσα** (δηλαδή με κάποιο είδος κωδικοποίησης)

Περιγραφή Αλγόριθμου

```
Αλγόριθμος Παράδειγμα_3
Διάβασε a, b
Αν a < b τότε
    c ← a + b
αλλιώς
    c ← a * b
Τέλος_αν
Εκτύπωσε c
Τέλος Παράδειγμα_3
```



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον - Βιβλίο Μαθητή, Γ΄ Γενικού Λυκείου, [http://ebooks.edu.gr/modules/ebook/show.php/DSGL-C101/36/198,1060/]

Αλγοριθμική Γλώσσα

- Για την περιγραφή των αλγορίθμων έχει προταθεί και χρησιμοποιείται μία «γλώσσα», η λεγόμενη **αλγοριθμική γλώσσα** η οποία είναι αποδεσμευμένη από τις λεπτομέρειες μιας κανονικής γλώσσας προγραμματισμού.
- Ο στόχος είναι να **περιγράφεται η λύση ενός προβλήματος** με τέτοιο τρόπο ώστε να μπορεί να μεταφερθεί αργότερα σε οποιαδήποτε γλώσσα προγραμματισμού.
- Στην αλγοριθμική γλώσσα αποφεύγονται εκφράσεις που υπάρχουν σε κάποια γλώσσα προγραμματισμού

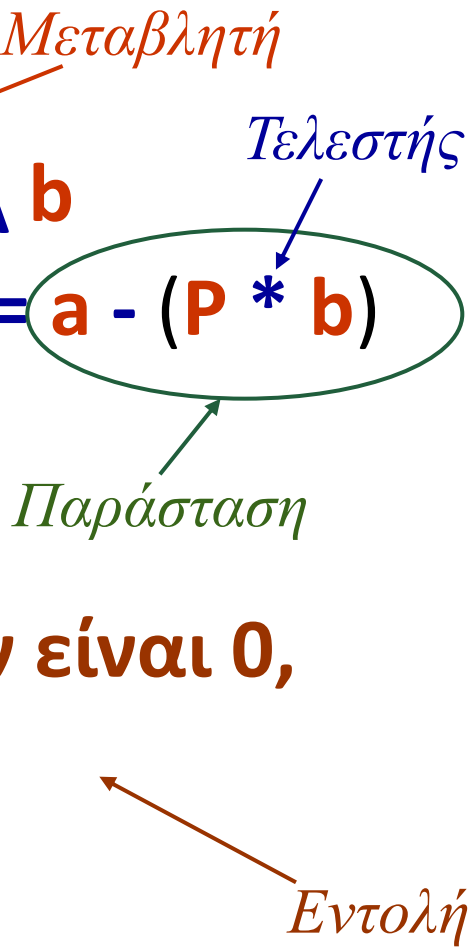
Στοιχεία Αλγοριθμικής Γλώσσας (1)

- **Δεδομένα** (Μεταβλητές, Σταθερές ποσότητες)
- **Μεταβλητές** (ανάλογα με το είδος της τιμής που μπορούν να λάβουν διακρίνονται σε αριθμητικές, αλφαριθμητικές και λογικές)
- **Τελεστές** (αριθμητικοί, λογικοί, συγκριτικοί)
- **Παραστάσεις** (Εκχώρηση Τιμής, Κριτήρια ή Συνθήκες)

Παράδειγμα (1)

Εύρεση του ΜΚΔ

1. Βρες το πηλίκο **P = a \ b**
2. Βρες το υπόλοιπο **Y = a - (P * b)**
3. Θέσε στο **a = b**
4. Θέσε στο **b = Y**
5. **Αν το υπόλοιπο Y δεν είναι 0, πήγαινε στο Βήμα 1**
6. Ο ΜΚΔ είναι ο a



Στοιχεία Αλγοριθμικής Γλώσσας (2)

- **Έλεγχος Ροής** (πχ Αν…Τότε…Αλλιώς)
- **Επαναληπτικές Δομές** (Για…από…μέχρι, Όσο…επανέλαβε, κλπ)
- **Διαδικασίες** (procedure, όνομα, σύνολο παραμέτρων)
- **Συναρτήσεις** (function, όνομα, σύνολο παραμέτρων, επιστρεφόμενη τιμή)

Παραδείγματα (2)

```
Αλγόριθμος Παράδειγμα_1
Διάβασε a
Διάβασε b
c ← a + b
Εκτύπωσε c
Τέλος Παράδειγμα_1
```

```
Αλγόριθμος Παράδειγμα_3
Διάβασε a, b
Αν a < b τότε
    c ← a + b
αλλιώς
    c ← a * b
Τέλος_αν
Εκτύπωσε c
Τέλος Παράδειγμα_3
```

```
Αλγόριθμος Παράδειγμα_10
Sum ← 0
Για i από 1 μέχρι 100
    Sum ← Sum + i
Τέλος_επανάληψης
Εκτύπωσε Sum
Τέλος Παράδειγμα_10
```

```
Αλγόριθμος Παράδειγμα_7
i ← 1
Όσο i ≤ 100 επανάλαβε
    Εμφάνισε i
    i ← i + 1
Τέλος_επανάληψης
Τέλος Παράδειγμα_7
```

Αλγόριθμος & Πρόγραμμα

```
ΠΡΟΓΡΑΜΜΑ Κόστος_Υπολογιστών
! Πρόγραμμα υπολογισμού κόστους παραγγελίας υπολογιστών
ΣΤΑΘΕΡΕΣ
ΦΠΑ=0.18
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Ποσότητα, Τιμή_μονάδας, Κόστος
    ΠΡΑΓΜΑΤΙΚΕΣ: Αξία_ΦΠΑ, Συνολικό_κόστος
ΑΡΧΗ
! Εισαγωγή δεδομένων
ΓΡΑΨΕ 'Δώσε την ποσότητα της παραγγελίας'
ΔΙΑΒΑΣΕ Ποσότητα
ΓΡΑΨΕ 'Δώσε την τιμή του υπολογιστή'
ΔΙΑΒΑΣΕ Τιμή_μονάδας
! Υπολογισμοί
Κόστος <- Ποσότητα* Τιμή_μονάδας
Αξία_ΦΠΑ <- Κόστος*ΦΠΑ
Συνολικό_κόστος <- Κόστος+Αξία_ΦΠΑ
! Εμφάνιση αποτελεσμάτων
ΓΡΑΨΕ 'Το κόστος των',Ποσότητα,'υπολογ. είναι ',Κόστος
ΓΡΑΨΕ  ' Η αξία του ΦΠΑ  είναι', Αξία_ΦΠΑ
ΓΡΑΨΕ 'Το συνολικό κόστος είναι', Συνολικό_κόστος
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Προγραμματιστικό περιβάλλον Basic

```
' Κόστος υπολογιστών
fpa = .18
INPUT "Δώσε την ποσότητα : ", Quantity
INPUT "Δώσε την τιμή του υπολογιστή : ", Value
Cost = Quantity * Value
CostFpa = Cost * fpa
Total = Cost + CostFpa
PRINT "Το κόστος των"; Quantity; "υπολογιστών είναι :";
Cost
PRINT USING "Η αξία ΦΠΑ είναι : ######"; CostFpa
PRINT USING "Το συνολικό κόστος είναι : #######"; Total
END
```

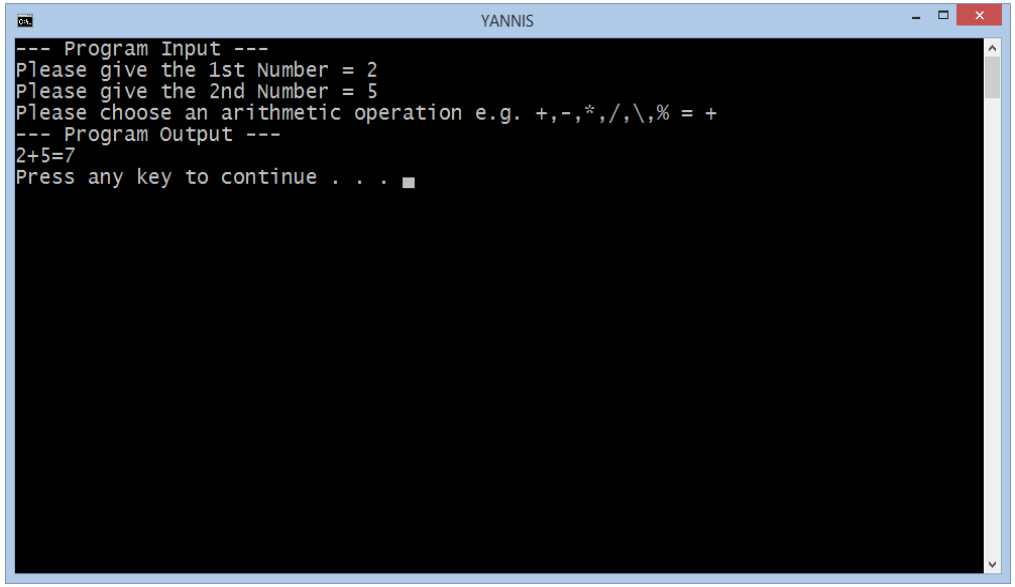
Αλγόριθμος & Πρόγραμμα (2)

```
Sub Main()
    'Num1 ο 1ος αριθμός, Num2 ο 2ος αριθμός, symbol το σύμβολο της πράξης
    'result το αποτέλεσμα, msg ένα μήνυμα κειμένου
    Dim Num1, Num2, result As Double, symbol As Char, msg As String

    'Εισοδος δεδομένων
    Console.CursorSize = 46
    Console.Title = "YANNIS"
    Console.WriteLine("--- Program Input ---")
    Console.Write("Please give the 1st Number = ")
    Num1 = Console.ReadLine()
    Console.Write("Please give the 2nd Number = ")
    Num2 = Console.ReadLine()
    Console.Write("Please choose an arithmetic operation e.g. +,-,*,/,\,% = ")
    symbol = Console.ReadLine()

    'Επεξεργασία Δεδομένων
    Select Case symbol
        Case "+" : result = Num1 + Num2
        Case "-" : result = Num1 - Num2
        Case "*" : result = Num1 * Num2
        Case "/"
            If Num2 = 0 Then msg = "Η πράξη δεν είναι δυνατή! Διαίρεση διά 0!" Else result = Num1 / Num2
        Case "%"
            If Num2 = 0 Then msg = "Η πράξη δεν είναι δυνατή! Διαίρεση διά 0!" Else result = Num1 Mod Num2
        Case "\"
            If Num2 = 0 Then msg = "Η πράξη δεν είναι δυνατή! Διαίρεση διά 0!" Else result = Num1 \ Num2
        Case Else
            msg = "Sorry!! This operation isn't supported!"
    End Select

    'Αποτελέσματα
    Console.WriteLine("--- Program Output ---")
    If msg = "" Then
        Console.WriteLine(Num1 & symbol & Num2 & "=" & result)
    Else
        Console.WriteLine(Num1 & symbol & Num2 & "=" & msg)
    End If
End Sub
```



Απόδοση Αλγόριθμου

- Συνήθως ο **χρόνος εκτέλεσής του** (πόσο χρόνο κάνει για να λύσει το πρόβλημα τρέχοντας σε μια συγκεκριμένη υπολογιστική μηχανή)
- Ο αλγόριθμος πρέπει να κάνει **αποδοτική χρήση των πόρων του υπολογιστή**
  - CPU, αποθηκευτικών χώρων, πόρων δικτύου, κ.ά
- Γενικά το κόστος μιας λύσης είναι το ποσό των πόρων που αυτή η λύση δαπανά
- Αναζητούμε αλγόριθμους που επιλύουν ένα πρόβλημα με αποδοτικό τρόπο ως προς τον **χρόνο** και τον **χώρο**
- Για τους περισσότερους αλγόριθμους, ο χρόνος τρεξίματος **T** εξαρτάται από το '**μέγεθος**' **n** της **εισόδου**

Μέτρηση Απόδοσης Αλγόριθμου

- **Best case scenario** (καλύτερη περίπτωση): *μικρότερη* τιμή που μπορεί να πάρει ο χρόνος εκτέλεσης **T(n)** για οποιαδήποτε είσοδο (input) με δεδομένο μέγεθος **n**.
- **Worse case scenario** (χειρότερη περίπτωση): *μεγαλύτερη* τιμή που μπορεί να πάρει ο χρόνος εκτέλεσης **T(n)** για οποιαδήποτε είσοδο (input) με δεδομένο μέγεθος **n**.
- **Expected/Average case scenario** (αναμενόμενη/μέση περίπτωση): *αναμενόμενη/μέση τιμή* που μπορεί να πάρει ο χρόνος εκτέλεσης **T(n)** για οποιαδήποτε είσοδο (input) με δεδομένο μέγεθος n.

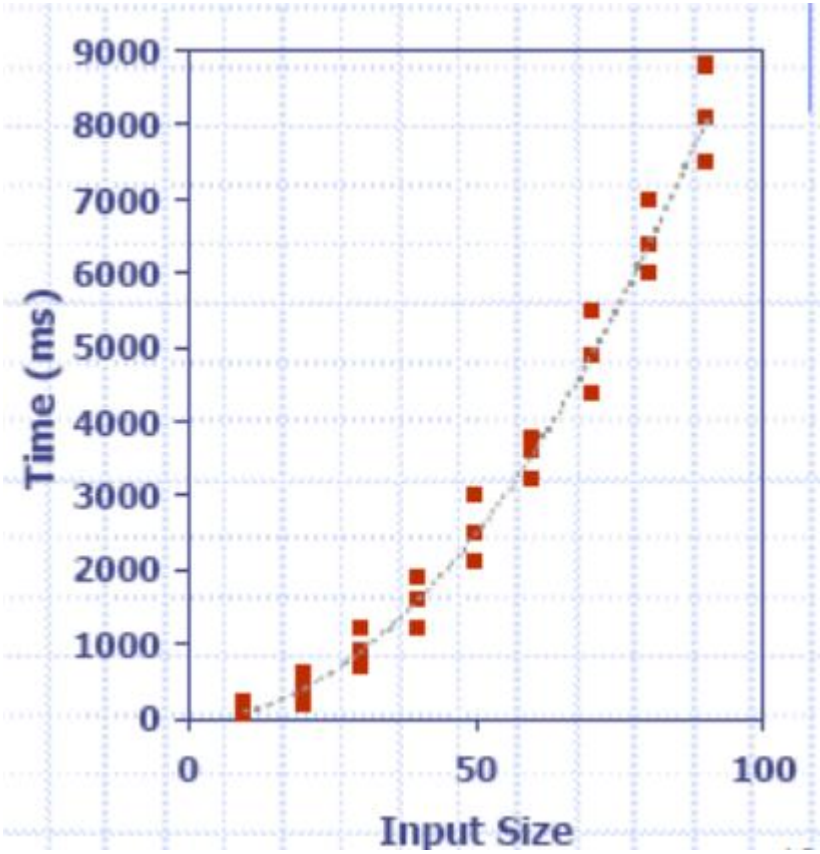
Μέτρηση Απόδοσης Αλγόριθμου

- Χρησιμοποιούνται δύο προσεγγίσεις:
  - Εμπειρική
  - Θεωρητική

- Δομές Δεδομένων
- Θεωρία αλγορίθμων

Εμπειρική προσέγγιση

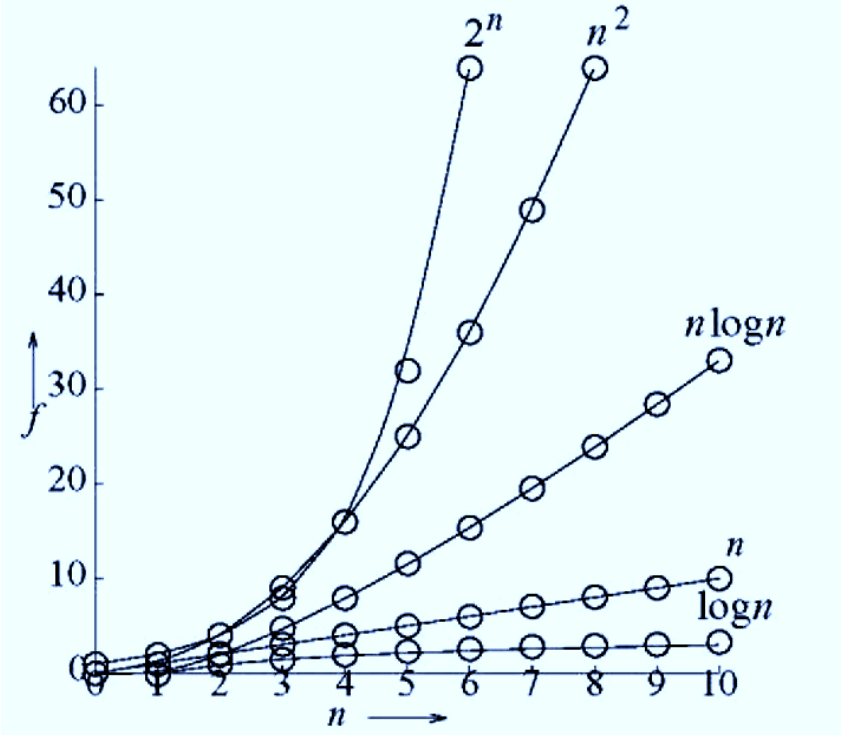
- Υλοποίηση προγράμματος
- Δημιουργία εισόδων διαφορετικού μεγέθους και σύνθεσης
- Καταγραφή του χρόνου εκτέλεσης για κάθε στιγμιότυπο
- Απεικόνιση σε γραφική παράσταση



Θεωρητική προσέγγιση

- Χρησιμοποιείται μια αναπαράσταση του αλγόριθμου (π.χ. ψευδοκώδικας)
- Εκτίμηση υπολογιστικών πόρων που απαιτεί η εκτέλεση του αλγόριθμου ως συνάρτηση του μεγέθους εισόδου
- Βασίζεται σε μαθηματικά εργαλεία και δεν εξαρτάται από συγκεκριμένη υλοποίηση
- Ανάλυση (χειρότερης, μέσης, καλύτερης περίπτωσης)

Ρυθμός αύξησης



Παράδειγμα

```
MAX = A[0]
Για i από 0 μέχρι N
  Αν A[i] >= MAX τότε
     MAX = A[i]
  Τέλος_αν
```

Ανάλυση χειρότερης περίπτωσης

- όταν τα στοιχεία του Πίνακα είναι σε αύξουσα σειρά
- $f(N) = 4 + 2N + 4N = 4 + 6N$

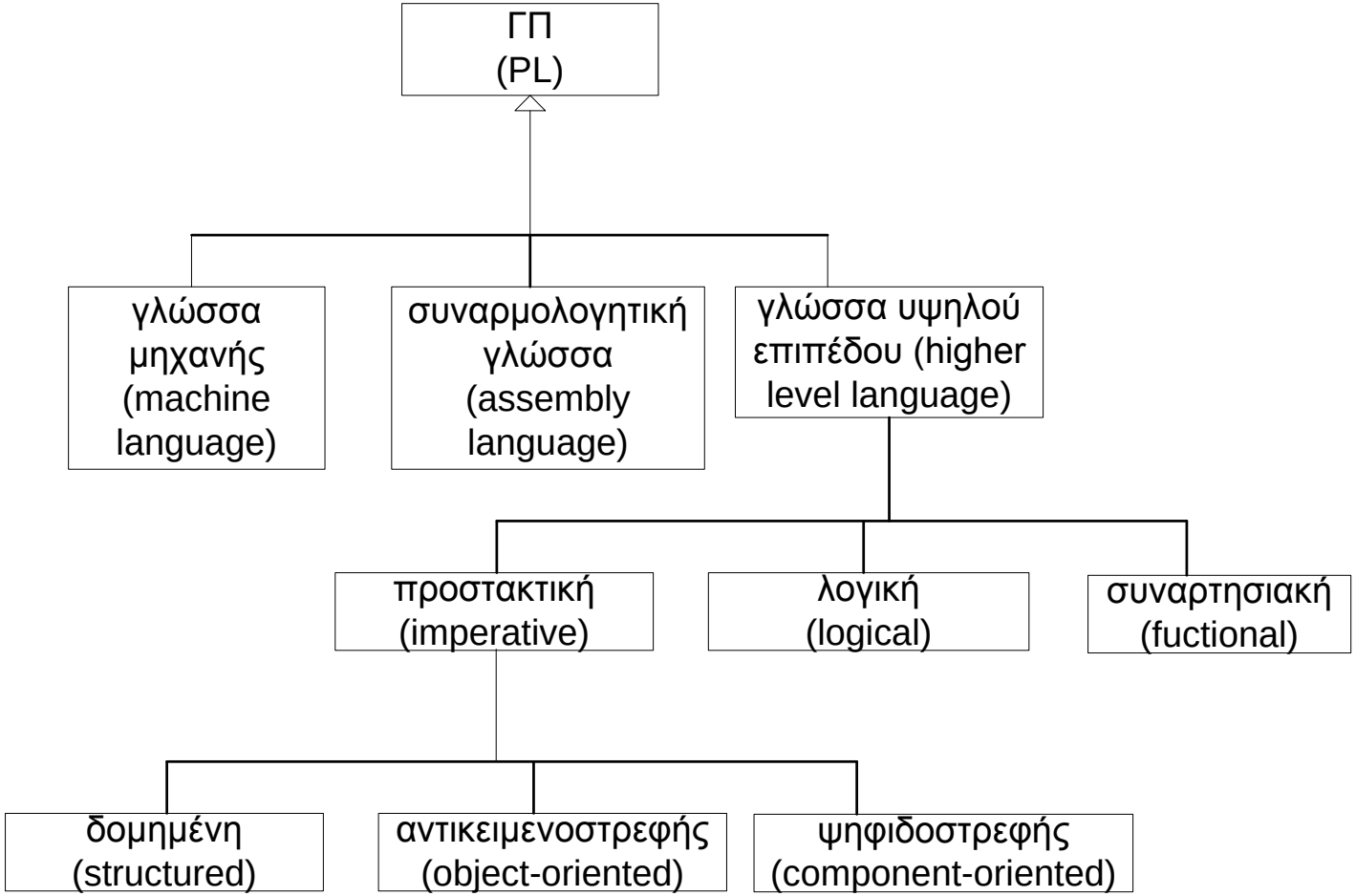
Ασυμπτωτική συμπεριφορά

- λαμβάνουμε υπόψη μόνο εκείνους τους όρους που μεγαλώνουν γρήγορα όσο το N μεγαλώνει.
- $f(N) = 6N \rightarrow f(N)=N$
- άρα είναι γραμμική η πολυπλοκότητά του

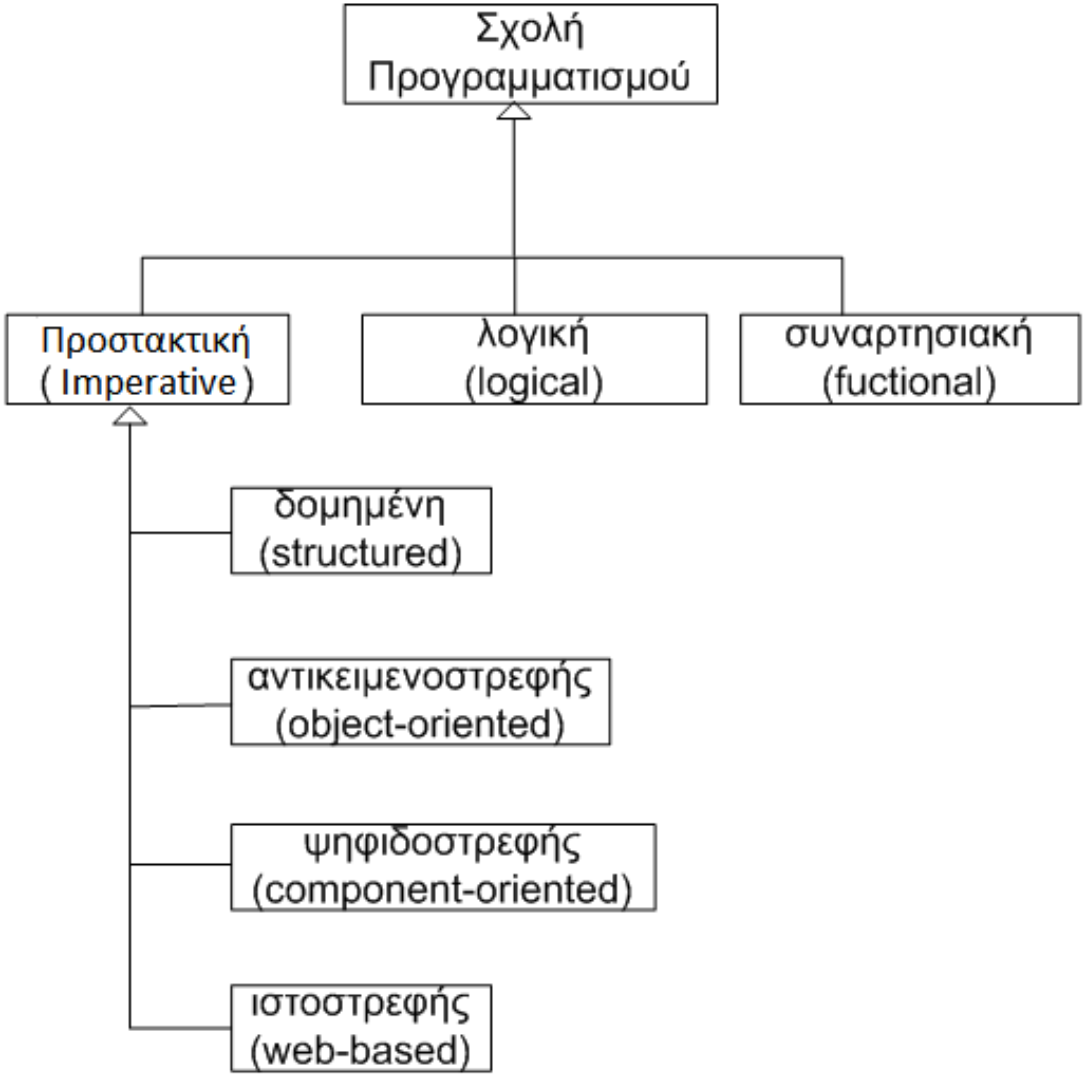
Γλώσσες Προγραμματισμού

- Η συγγραφή προγραμμάτων σε γλώσσα μηχανής είναι μια πολύ επίπονη εργασία. Οι **Γλώσσες Προγραμματισμού Υψηλού Επιπέδου** (ΓΠΥΕ) δημιουργήθηκαν από αυτή την ανάγκη.
- Ένα πρόγραμμα γραμμένο σε μια ΓΠΥΕ περιέχει εντολές που χρησιμοποιούν λέξεις που περιγράφουν την εργασία που πρόκειται να εκτελεστεί με σαφώς πιο κατανοητό τρόπο απ' ότι οι εντολές σε γλώσσα μηχανής ή οι συμβολικές γλώσσες (assembly)
- Παραδείγματα τέτοιων γλωσσών είναι οι C, C++, JAVA, Visual Basic, κ.λπ

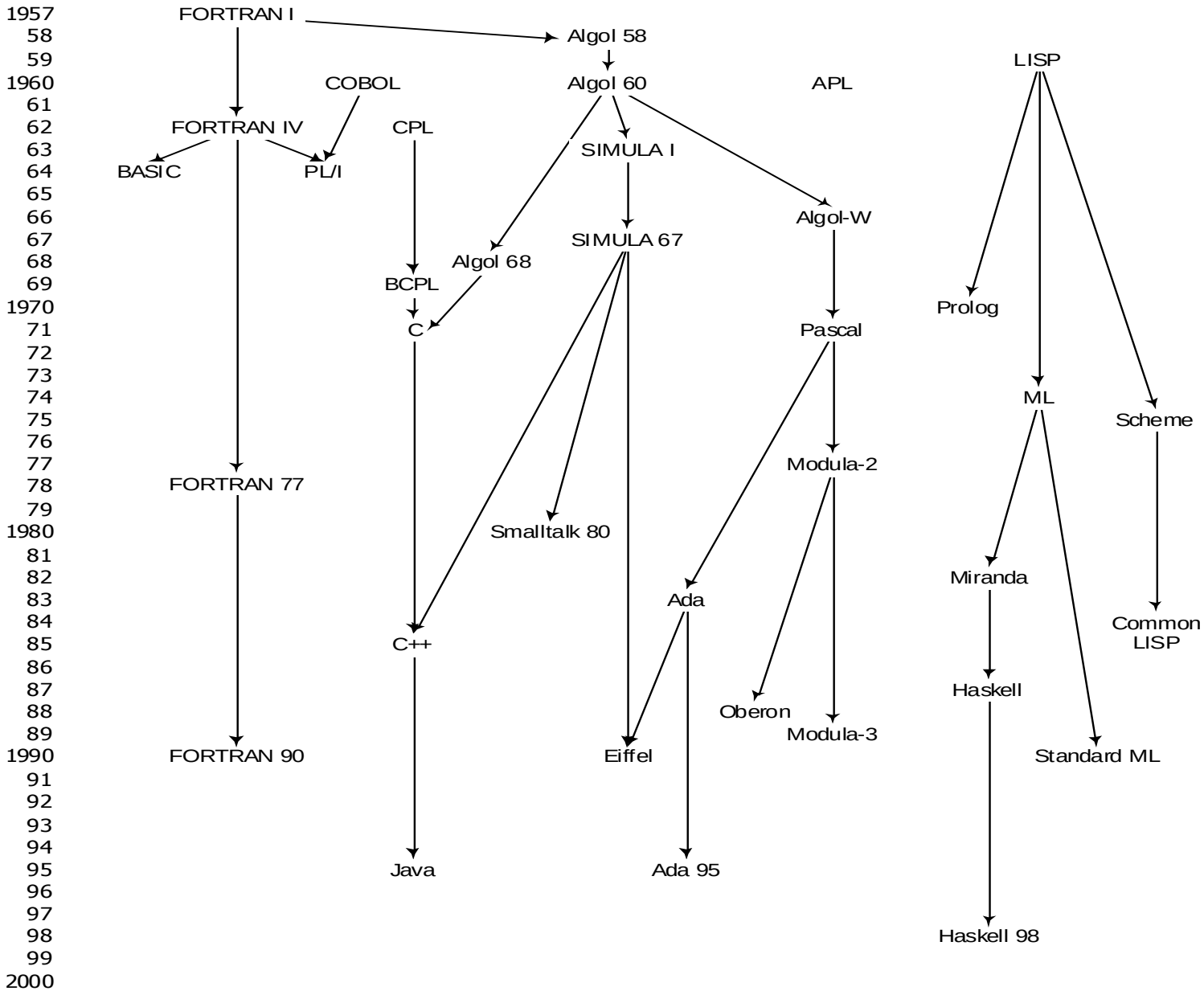
Μια κατηγοριοποίηση των Γλωσσών Προγραμματισμού



Σχολές Προγραμματισμού



Ιστορία Γλωσσών Προγραμματισμού



Παραδείγματα Προγραμματισμού

- Οι περισσότερες Γλώσσες Προγραμματισμού που έχουν χρησιμοποιηθεί και χρησιμοποιούνται σήμερα για την ανάπτυξη λογισμικών συστημάτων ανήκουν στο παράδειγμα του ***προστακτικού προγραμματισμού*** (**imperative programming**)
- η εργασία που πρόκειται να αυτοματοποιηθεί περιγράφεται ως μια ακολουθία εντολών (**προσταγών**) που πρέπει να εκτελεστούν.
- Εξέχουσα θέση στον προστακτικό προγραμματισμό έχουν οι μεταβλητές και η ανάθεση τιμών σε αυτές.
- Σύμφωνα με το παράδειγμα του ***συναρτησιακού προγραμματισμού*** (**functional programming**) η εργασία που πρόκειται να αυτοματοποιηθεί περιγράφεται ως ένα σύνολο μαθηματικών συναρτήσεων
- Σύμφωνα με το παράδειγμα του ***λογικού προγραμματισμού*** (**logic programming**) η εργασία που πρόκειται να αυτοματοποιηθεί περιγράφεται ως ένα σύνολο γεγονότων και συλλογιστικών κανόνων.

Μεταγλωττιστές – Διερμηνείς

- Ο Μεταγλωττιστής (**compiler**) είναι ένα πρόγραμμα ή ένα σύνολο προγραμμάτων που μετατρέπει τον **αρχικό κώδικα** (**source code**) που είναι γραμμένος σε μια γλώσσα υψηλού επιπέδου σε μια γλώσσα χαμηλότερου επιπέδου (συμβολική γλώσσα ή γλώσσα μηχανής). Ο βασικότερος λόγος της μεταγλώττισης είναι η δημιουργία ενός **εκτελέσιμου** (**executable**) προγράμματος.
- Ο Διερμηνέας (**interpreter**) είναι ένα πρόγραμμα που μεταφράζει και **εκτελεί** τις εντολές ενός προγράμματος γραμμένου σε μια γλώσσα υψηλού επιπέδου.

Η Διαδικασία της Μεταγλώττισης

Προγραμματιστής

Η/Υ

Γλώσσα Προγραμματισμού

Αρχικός Κώδικας (**source code**)

Εκτελέσιμος Κώδικας (**executable code**)

Εντολές σε Γλώσσα Προγραμματισμού

Εντολές σε Γλώσσα Μηχανής

Βασικά συστατικά ενός προγράμματος

- **Μεταβλητές (Variables):** Για την αποθήκευση και παράσταση δεδομένων
- **Επαναλήψεις (Loops):** Επιτρέπουν την επανάληψη της εκτέλεσης μιας ομάδας εντολών μέχρι να ικανοποιηθεί μια συνθήκη
- **Συνθήκες (Conditionals):** Επιτρέπουν την εκτέλεση μιας ομάδας εντολών που εξαρτάται από το αν ικανοποιείται μια συνθήκη (είναι δηλαδή αληθής ή ψευδής)
- **Input/Output:** Επιτρέπουν την επικοινωνία του προγράμματος με εξωτερικές οντότητες
- **Υπορουτίνες & Συναρτήσεις (Subroutines & functions):** Επιτρέπουν την ομαδοποίηση εντολών για τη διαχείριση πολυπλοκότητας (τοποθετούμε & δίνουμε ένα όνομα στην ομάδα αυτή των εντολών και την χρησιμοποιούμε όσες φορές θέλουμε στο πρόγραμμά μας χωρίς να την ξαναγράφουμε)

Το «interface» ενός προγράμματος

- Στην επιστήμη των υπολογιστών το **interface** (διεπαφή ή διασύνδεση) γενικά δηλώνει το σημείο αλληλεπίδρασης μεταξύ συστατικών (είτε σε επίπεδο υλικού είτε σε επίπεδο λογισμικού)
- Σε επίπεδο υλικού (hardware) υπάρχουν διάφορα interfaces μεταξύ των συστατικών υλικού ενός υπολογιστικού συστήματος
  - USB interface, SCSI, I/O, κ.λπ.
- Στο λογισμικό υπάρχουν επίσης διάφορα επίπεδα interface π.χ. μεταξύ του λειτουργικού συστήματος και συστατικών υλικού, μεταξύ ενός προγράμματος και του λειτουργικού συστήματος, μεταξύ ενός προγράμματος και του τελικού χρήστη (**user interface**), κ.λπ.

Το «interface» ενός προγράμματος (2)

- Σε μια μονάδα προγράμματος το interface αφορά όλα εκείνα τα στοιχεία της μονάδας που είναι «**ορατά**» έξω από τη μονάδα αυτή.
  - π.χ. σε ένα Module της VB είναι όλες οι Public (δημόσιες) μεταβλητές και ρουτίνες (μέθοδοι). Ότι δηλώνεται ως Private δεν είναι ορατό έξω από τη μονάδα αυτή.

Το προγραμματιστικό περιβάλλον

- Αποτελείται από όλα εκείνα τα εργαλεία που είναι απαραίτητα στον προγραμματιστή κατά την ανάπτυξη του προγράμματός του
  - Συνήθως πλαισιώνεται από έναν (έξυπνο) συντάκτη προγράμματος, το μεταγλωττιστή / διερμηνέα, διάφορες βιβλιοθήκες, εργαλεία ελέγχου και αποσφαλμάτωσης του προγράμματος.

Τέλος Ενότητας

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης