



ACM AUTH
Student Chapter

Programming Basics

ACM Student Chapter Auth



Στόχος Σεμιναρίου

- Το σεμινάριο απευθύνεται **κυρίως...**
 - σε φοιτητές θετικής κατεύθυνσης του λυκείου που θέλουν να ασχοληθούν με τον προγραμματισμό
- Στόχος είναι...
 - Η εξοικίωσή τους με τις πιο βασικές έννοιες **διαδικασιακού** προγραμματισμού και **αλγοριθμικής** λογικής
 - Η Απόκτηση της ικανότητας σκέψης ώστε να μπορέσει να εφαρμοστεί η λογική σε κάποια γλώσσα προγραμματισμού στη συνέχεια (πχ C/C++)



Περιεχόμενα

- Τι είναι αλγόριθμος
- Μεταβλητές και εντολή εκχώρησης
- Δομές επιλογής
- Βασική μεθοδολογία
- Δομές δεδομένων – Πίνακες
- Μεθοδολογία πινάκων
- Γλώσσες Προγραμματισμού

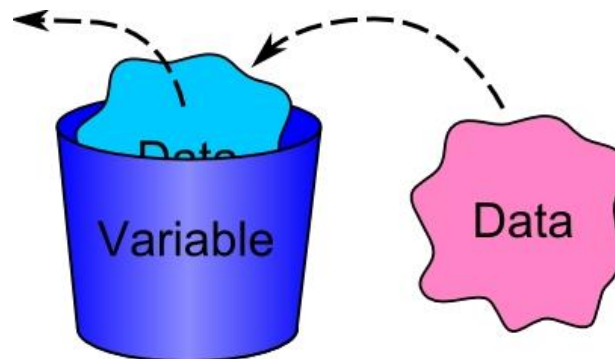


Τι είναι αλγόριθμος

- **Αλγόριθμος** είναι μία πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων, σε πεπερασμένο χρόνο, με σκοπό την επίλυση κάποιου προβλήματος
- Χαρακτηριστικά
 - Είσοδος
 - Έξοδος
 - Περατότητα
 - Καθοριστικότητα
 - Αποτελεσματικότητα

Μεταβλητές

- Οι μεταβλητές παριστάνουν ποσότητες που η τιμή τους μπορεί να μεταβάλλεται κατά τη διάρκεια εκτέλεσής του προγράμματος
 - Καθορίζονται από
 - Όνομα (κανόνες ονοματολογίας)
 - Τύπο δεδομένων (ακέραιος, χαρακτήρας κλπ)





Εντολή εκχώρησης

- $\alpha \leftarrow 1$



- $\alpha \leftarrow 2$



Ανταλλαγή μεταβλητών?

$A \leftarrow 1$

$B \leftarrow 2$

Temp $\leftarrow A$

$A \leftarrow B$

$B \leftarrow \text{Temp}$





Άλλες εντολές που θα χρησιμοποιήσουμε

- **Εμφάνιση** (έξοδος - οθόνη)
 - Εμφάνισε 5
- **Διάβασε** (είσοδος - πληκτρολόγιο)
 - Διάβασε x
- **Τελεστές:**
 - +, -, *, /, Mod, Div
 - =, >, <, >=, <=, !=
 - ή, και
- **Σχόλια:**
 - ! Αυτό είναι ένα σχόλιο

Σειριακή εκτέλεση

Αλγόριθμος

```
!...  
!Εντολές  
!...  
  
}
```

Παράδειγμα

```
Αλγόριθμος  
  Διάβασε vasi, upsos  
  Emv ← vasi * upsos  
  Εμφάνισε Emv  
}
```



Δομές επιλογής

- **Απλή επιλογή**

```
Αν( <συνθήκη> )  
{  
    !εντολές  
}
```

- **Σύνθετη επιλογή**

```
Αν( <συνθήκη> )  
{  
    !εντολές1  
}  
Αλλιώς{  
    !εντολές2  
}
```

- **Πολλαπλή επιλογή**

```
Αν( <συνθήκη1> )  
{  
    !εντολές1  
}  
Αλλιώς_αν( <συνθήκη2 >  
{  
    εντολές2  
}  
...  
Αλλιώς_αν( <συνθήκηN> )  
{  
    εντολέςN  
}  
Αλλιώς{  
    εντολέςN+1  
}
```




Απλά παραδείγματα

Αλγόριθμος{

Διάβασε x

Αν $(x \bmod 2 = 0)$

{

 Εμφάνισε “Άρτιος αριθμός”

}

Αλλιώς [$\text{Αλλιώς_αν } (x \bmod 2 == 1)$]

{

 Εμφάνισε “Περιττός Αριθμός”

}

}

Αλγόριθμος{

 Διάβασε x

 Αν $(x > 0)$

 {

 Εμφάνισε “Θετικός
 αριθμός”

 }

 Αλλιώς_αν $(x < 0)$

 {

 Εμφάνισε “Αρνητικός
 Αριθμός”

 }

 Αλλιώς{

 Εμφάνισε “μηδέν”

 }

}



Εμφώλευση και στοίχιση

```
Αν( <συνθήκη1> ){  
    !εντολές1  
    Αν(<συνθήκη2> ){  
        !εντολές2  
        Αν(<συνθήκη3> ){  
            !εντολές3  
            !... .. ΚΟΚ ... ..  
        }  
    }  
}
```

Παράδειγμα κλιμακωτής χρέωσης

Εταιρεία παραγωγής προϊόντων

- ... - 0 : error
- 1-100kg : 5€/τεμαχιο
- 101-250kg : 4,6€/τεμάχιο
- 251-500kg : 4,2€/τεμαχιο
- 501 - ...kg : error

```
Αλγόριθμος{
  Διάβασε tem
  Αν(tem <= 0 ή tem > 500)
  {
    Εμφάνισε "error"
  }
  Αλλιώς{
    Αν(tem <= 100)
    {
      xr ← tem * 5
    }
    Αλλιώς_Αν(tem <= 250)
    {
      xr ← 100*5 + (tem-100)*4,6
    }
    Αλλιώς{
      xr ← 100*5+150*4,6+(tem-250)*4,2
    }
    Εμφάνισε xr
  }
}
```



Δομές επανάληψης – Για

- Για var από 1 μέχρι c με_βήμα 1
{
 !block εντολών
}
- Παρατηρήσεις
 - Γνωστός αριθμός επαναλήψεων
 - Χρησιμοποιούμε την μεταβλητή var αλλά δεν την μεταβάλλουμε μέσα στο block εντολών
 - Οι επαναλήψεις εκτελούνται όταν η συνθήκη είναι αληθής
 - Αν το βήμα δεν μεταβάλλεται, έχουμε άπειρες επαναλήψεις
 - Αριθμός των επαναλήψεων = $[(\text{Τιμή2} - \text{Τιμή1})/\text{βήμα}] + 1$



Δομές επανάληψης - Όσο

Όσο(<συνθήκη>)

{

!block εντολών

}

- Άγνωστο πλήθος επαναλήψεων
- Εκτελείται όσο το συνθήκη είναι αληθής
- Δεν ξεχνάμε να μεταβάλουμε την μεταβλητή που εξαρτάται η συνθήκη μέσα στο block εντολών

Αρχή_Επανάληψης

{

!block εντολών

Όσο(<συνθήκη>)

- Όμοια με την Όσο
- Εκτελείται πάντα η πρώτη επανάληψη και στην συνέχεια πραγματοποιείται ο πρώτος έλεγχος
- Συνήθως μας βολεύει να την χρησιμοποιούμε για έλεγχο εισόδου



Παραδείγματα κατανόησης

```
Για var από 1 μέχρι 10{  
    Εμφάνισε var  
}
```



```
Var ← 1 !(από?)  
Όσο(var<10){  
    Εμφάνισε var  
    var ← var + 1 !(βήμα?)  
}
```

```
Για var από 1 μέχρι 10 με_βήμα 2{  
    Εμφάνισε var  
}
```



```
Var ← 1 !(από?)  
Όσο(var<10){  
    Εμφάνισε var  
    var ← var + 2 !(βήμα?)  
}
```



Παραδείγματα κατανόησης

Προσοχή!

```
Για i από 1 μέχρι 5 με_βήμα 1{  
    !εντολές...}
```

```
}
```

Εμφάνισε i

Τι εμφανίζεται στην οθόνη?





Βασική Μεθοδολογία

- Αν είναι γνωστός ο αριθμός των επαναλήψεων επιλέγουμε
 - Για
- Αν είναι άγνωστος ο αριθμός των επαναλήψεων επιλέγουμε:
 - Όσο
 - Αν υπάρχει πιθανότητα να μη γίνει καμία επανάληψη
 - Αρχή_Επανάληψης...Όσο
 - Αν πρέπει να γίνει τουλάχιστον μία επανάληψη



Βασική Μεθοδολογία

Επαναληπτική εισαγωγή με έλεγχο τιμής

```
Αρχή_Επανάληψης{  
  Διάβασε mark  
  Όσο(mark<0 ή mark>10)
```



```
Διάβασε mark  
While(mark < 0 ή mark > 10){  
  Εμφάνισε "Error – try again"  
  Διάβασε mark  
}
```



Βασική Μεθοδολογία

Μέσος όρος **100** ακεραίων

```
Αλγόριθμος{  
  Σ ← 0  
  Για i από 0 μέχρι 100 με_βήμα 1 {  
    Διάβασε x  
    Σ ← Σ + x  
  }  
  ΜΟ ← Σ / 100  
  Εμφάνισε ΜΟ  
}
```

Μέσος όρος ακεραίων (**άγνωστο πλήθος**)

```
Αλγόριθμος{  
  Σ ← 0  
  Ν ← 0  
  Διάβασε x  
  Όσο(x!=0){  
    Ν ← Ν + 1  
    Σ ← Σ + x  
    Διάβασε x  
  }  
  Αν(N>0){  
    ΜΟ ← Σ / Ν  
    Εμφάνισε ΜΟ  
  }  
}
```

Βασική μεθοδολογία

Γνωστός αριθμός επαναλήψεων

Άγνωστός αριθμός επαναλήψεων

```

Αλγόριθμος{
  Διάβασε x
  max ← x
  min ← x
  Για i από 1 μέχρι 100
    Διάβασε x
    Αν(x>max){
      max ← x
    }
    Αν (x<min){
      min ← x
    }
  }
  Εμφάνισε max, " - ", min
}

```

```

Αν(x>max){
  max ← x
}
Αλλιώς_αν(x<min){
  min ← x
}

```

```

Αλγόριθμος{
  Διάβασε x
  max ← x
  min ← x
  Όσο (x != 0){
    Αν(x>max){
      max ← x
    }
    Αν (x<min){
      min ← x
    }
    Διάβασε x
  }
  Εμφάνισε max, " - ", min
}

```



Βασική Μεθοδολογία

Εύρεση αθροίσματος σειράς αριθμών

Η τιμή του κάθε όρου ταυτίζεται με την τιμή του μετρητή των επαναλήψεων

$\Sigma 1 \leftarrow 0$

Για i από 1 μέχρι N {

$\Sigma 1 \leftarrow \Sigma 1 + 1$

}

$\Sigma 2 \leftarrow 0$

Για i από 1 μέχρι N με_βήμα 3 {

$\Sigma 2 \leftarrow \Sigma 2 + i$

}

Άθροισμα $1 + 3 + 6 + 10 + 15 + 21 + \dots$ των N πρώτων όρων

$\Sigma 1 \leftarrow 0$

$X \leftarrow 1$

$\beta \leftarrow 2$

Για i από 1 μέχρι N {

$\Sigma \leftarrow \Sigma + X$

$X \leftarrow X + \beta$

$\beta \leftarrow \beta + 1$

}



Εμφώλευση

Εμφάνιση συντεταγμένων ενός τετραγωνικού πίνακα 5x5

Αλγόριθμος{

 Για i από 0 μέχρι 5 με_βήμα 1{

 Για j από 0 μέχρι 5 με_βήμα 1{

 Εμφάνισε i,j

 }

 }

}



Ασκήσεις

Είσοδος – a, b

Έξοδος – ακέραιοι αριθμοί στο διάστημα (a, b)

```
Αλγόριθμος{
  Διάβασε  $a, b$ 
  Αν( $a=b$ ){
    Εμφάνισε "Error"
  }
  Αν( $a>b$ ){
     $temp \leftarrow a$ 
     $a \leftarrow b$ 
     $b \leftarrow temp$ 
  }
  Για  $i$  από  $a+1$  μέχρι  $b$ {
    Εμφάνισε  $i$ 
  }
}
```



Ασκήσεις

1	2	3	4
*	**	***	****
*	**	***	****
*	**	***	****
*	**	***	****

```
*  
**  
***  
...  
** .....* (N αστεράκια)
```

Αλγόριθμος{

Διάβασε α

Για j από 0 μέχρι 4

{

Για i από 0 μέχρι α

{

Εμφάνισε "*"

}

Εμφάνισε "\n"

}

}

Αλγόριθμος{

Διάβασε α

Για i από 0 μέχρι α

{

Για j από 0 μέχρι i

{

Εμφάνισε "*"

}

Εμφάνισε "\n"

}

}



Ασκήσεις

Υπολογισμός δύναμης

```
Αλγόριθμος{  
    Διάβασε vasi  
    Διάβασε ekthetis  
     $c \leftarrow 1$   
    Για i από 0 μέχρι ekthetis{  
         $c \leftarrow c * vasi$   
    }  
    Εμφάνισε c  
}
```

Υπολογισμός παραγοντικού

```
Αλγόριθμος{  
    Διάβασε a  
     $N \leftarrow 1$   
    Για i από 1 μέχρι a+1{  
         $N \leftarrow N * i$   
    }  
    Εμφάνισε N  
}
```




Ασκήσεις

Να γραφεί αλγόριθμος που θα διαβάζει έναν αριθμό και θα εμφανίζει το άθροισμα των ψηφίων του (π.χ. 345 \rightarrow 3+4+5=12).

Αλγόριθμος{

$x \leftarrow 10, \text{sum} \leftarrow 0, \gamma \leftarrow 10$

Διάβασε a !integer

$\text{sum} \leftarrow \text{sum} + (a \bmod x)$

Όσο($a \geq x$){

$x \leftarrow x * 10;$

$\text{sum} \leftarrow \text{sum} + (a \bmod x) / \gamma$

$\gamma \leftarrow \gamma * 10$

}

Εμφάνισε sum

}



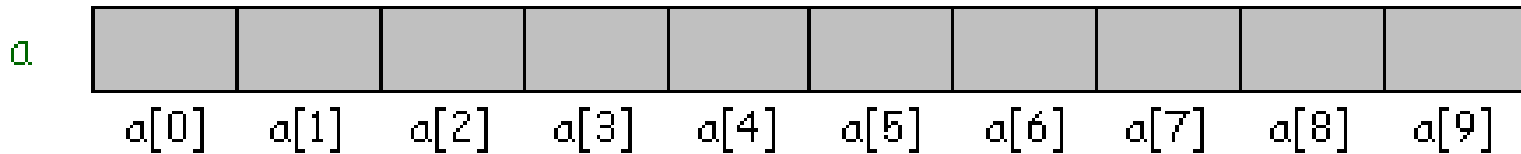
Δομές Δεδομένων – Πίνακες

- **Wirth**
 - Προγράμματα = αλγόριθμος + δομές δεδομένων
- **Δομή δεδομένων** είναι ένα σύνολο αποθηκευμένων δεδομένων που μπορούν να υποστούν λειτουργία από ένα σύνολο λειτουργιών
- Οι **πίνακες** αποτελούν στατικές δομές δεδομένων, που δεσμεύουν συνεχόμενες θέσεις μνήμης.
 - Όλα τα στοιχεία είναι του ίδιου τύπου.
 - Μπορεί να είναι μονοδιάστατοι, δυσδιάστατοι κλπ.
- **Βασικές λειτουργίες πινάκων**
 - Εισαγωγή, Διαγραφή, Αναζήτηση, Ταξινόμηση, Αντιγραφή Συγχώνευση και διαχωρισμός.



Αναπαράσταση ενός πίνακα

1D



2D

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]



Παραδείγματα - Διαχείριση πινάκων

Προσπέλαση των στοιχείων ενός μονοδιάστατου πίνακα

```
Για i από 0 μέχρι N{  
  Διάβασε Π[i]  
}
```

```
Για i από 0 μέχρι N{  
  Γράψε Π[i]  
}
```



Παραδείγματα - Διαχείριση πινάκων

Άθροισμα και μέσος όρος στοιχείων ενός πίνακα

```
Σ ← 0
Για i από 0 μέχρι N{
    Σ ← Σ + Π[i]
}
ΜΟ ← Σ/N
```



Παραδείγματα - Διαχείριση πινάκων

Προσπέλαση των στοιχείων ενός δισδιάστατου πίνακα

```
Για i από 0 μέχρι N{  
    Για j από 0 μέχρι M{  
        Διάβασε Π[i][j]  
    }  
}
```

Εμφώλευση



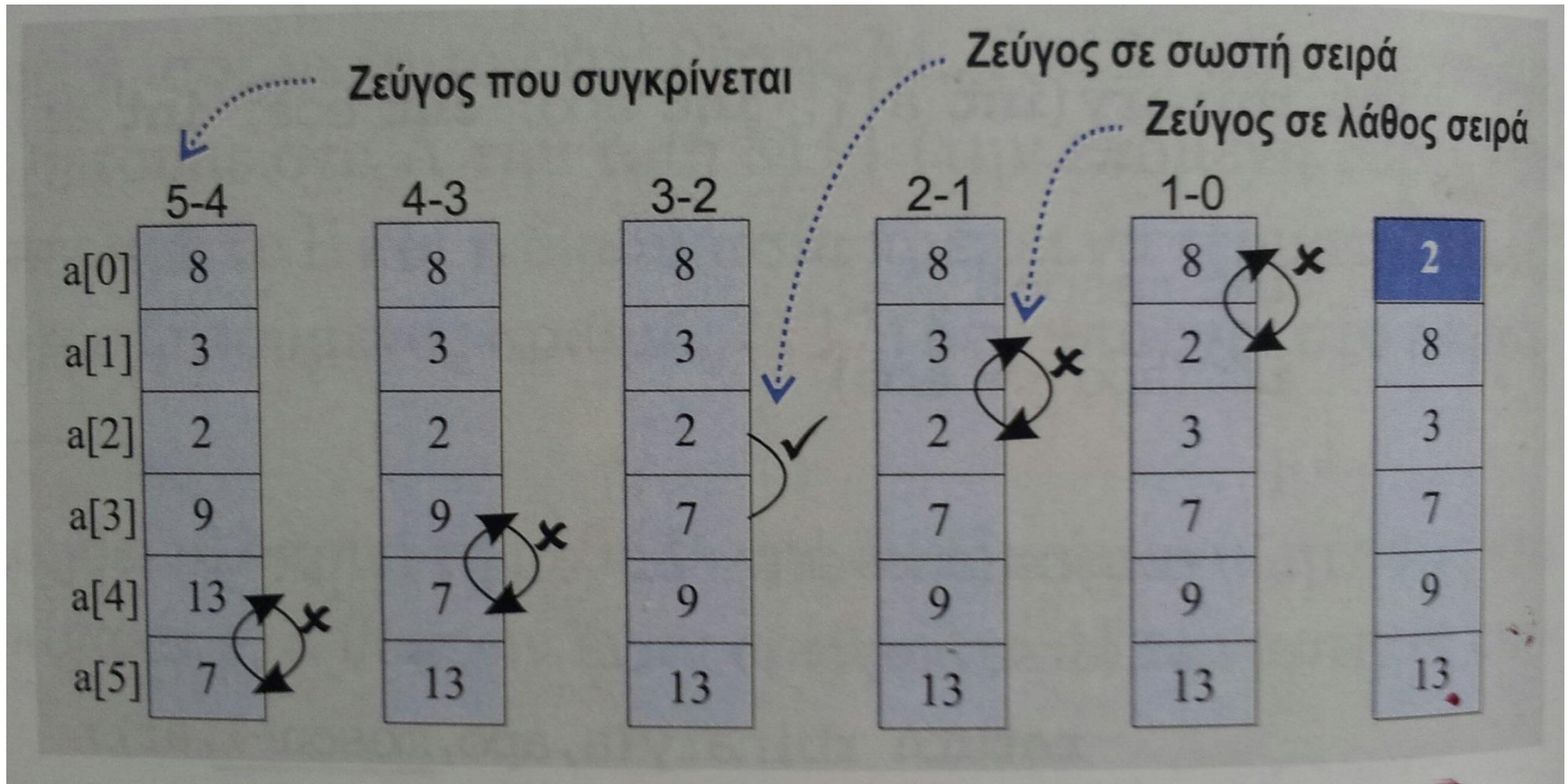
```
Για i από 0 μέχρι N{  
    Για j από 0 μέχρι M{  
        Γράψε Π[i][j]  
    }  
}
```



Παραδείγματα - Μεθοδολογία πινάκων

```
max ← Π[0]
min ← Π[0]
max_pos ← 0
min_pos ← 0
Για i από 1 μέχρι N{
    Αν(Π[i] > max){
        max ← Π[i]
        max_pos ← i
    }
    Αν(Π[i] < min){
        min ← Π[i]
        min_pos ← i
    }
}
```

Ταξινόμηση φυσαλίδας





Ταξινόμηση φυσαλίδας

```
Αλγόριθμος{
  !δεδομένα N,Π
  Για i από 1 μέχρι N{ !N-1 φορές
    Για j από N-1 μέχρι i-1 με_βήμα -1{
      Αν (Π[j] < Π[j-1]){ !αύξουσα
        Αντιμετάθεσε(Π[j-1],Π[j])
      }
    }
  }
  !Αποτελέσματα Π
}
```



Σειριακή Αναζήτηση

Αλγόριθμος{

 Διάβασε key

 pos \leftarrow -1

 Για i από 0 μέχρι N{

 Αν($\Pi[i] = \text{key}$){

 pos \leftarrow i

 }

 }

 Αν (pos = -1){

 Εμφάνισε “Δε βρέθηκε”

 }

 Αλλιώς{

 !εντολές...

 }

}



Συγχώνευση ταξινομημένων πινάκων

Array A

1	3	5	5	100
---	---	---	---	-----

Array B

2	4	5	6	7	8	9
---	---	---	---	---	---	---

Resulting Array: Merge (A, B)

1	2	3	4	5	5	5	6	7	8	9	100
---	---	---	---	---	---	---	---	---	---	---	-----



Συγχώνευση ταξινομημένων πινάκων

Αλγόριθμος{

$i \leftarrow 0, j \leftarrow 0, k \leftarrow 0$

Όσο($i < N$ και $j < M$){

Αν($A[i] < B[j]$){

$\Gamma[k] \leftarrow A[i]$

$i \leftarrow i+1$

}

Αλλιώς{

$\Gamma[k] \leftarrow B[j]$

$j \leftarrow j+1$

}

$k \leftarrow k+1$

}

!συνέχεια δεξιά \rightarrow

Αν($i > N-1$){ !τελείωσε ο πίνακας A

Για λ από j μέχρι M {

$\Gamma[k] \leftarrow B[\lambda]$

$k \leftarrow k+1$

}

}

Αλλιώς{ !τελείωσε ο πίνακας B

Για λ από i μέχρι N {

$\Gamma[k] \leftarrow A[\lambda]$

$k \leftarrow k+1$

}

}

}!τέλος αλγόριθμου



Βασικά θέματα γλωσσών προγραμματισμού

- Γλώσσες προγραμματισμού υψηλού επιπέδου
 - Διαδικασιακές (ή αλγοριθμικές) (οι περισσότερες)
 - Αντικειμενοστραφείς (C++,Java)
 - Συναρτησιακές (Lisp)
 - Μη διαδικασιακές (Prolog)
 - Ερωτοαπαντήσεων (SQL)
 - Παράλληλου προγραμματισμού
 - Οπτικού και οδηγούμενου από τα γεγονότα προγραμματισμού(Visual C++, delphi, Java κ.α)



ACM AUTH
Student Chapter

Τέλος

Thank you!