

# ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Ευάγγελος Γ. Ούτσιος

Θεόδωρος Γ. Λάντζος

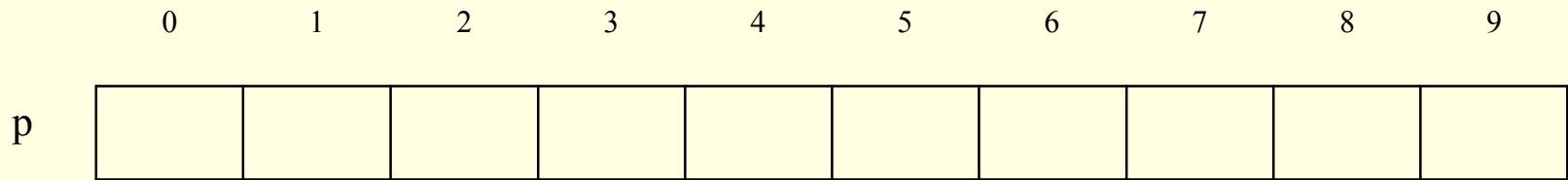
Διάλεξη Νο4

# Πίνακες

---

- Αθροιστικός τύπος, Στατική δομή
- Δομή που περιέχει στοιχεία του ίδιου τύπου
- Τα στοιχεία αποθηκεύονται σε συνεχόμενες θέσεις μνήμης
- Δηλώνεται όπως οι μεταβλητές
- Η χρήση γίνεται με το όνομα του πίνακα και με την χρήση ενός δείκτη σε παρένθεση ή αγκύλη (για την C έχουμε αγκύλη)
- Πίνακες πολλών διαστάσεων
- Ευρέως χρησιμοποιημένοι είναι οι μονοδιάστατοι και δισδιάστατοι

# Μονοδιάστατοι πίνακες & Functional programming



Ορισμός μονοδιάστατου πίνακα ακεραίων

```
#define N 10
```

```
main()
```

```
{
```

```
  int p[N];
```

```
  .
```

```
  .
```

```
  .
```

```
  .
```

```
}
```

Διάβασμα μονοδιάστατου πίνακα.

```
for (i=0; i<N; i++)
```

```
  cin >> p[i];
```

Εμφάνιση μονοδιάστατου πίνακα.

```
for (i=0; i<N; i++)
```

```
  cout << p[i];
```

Άθροισμα στοιχείων μονοδιάστατου πίνακα.

```
sum = 0;
```

```
for (i=0; i<N; i++)
```

```
  sum = sum + p[i];
```

# Παράδειγμα υλοποίησης πίνακα μιας διάστασης

```
#include <stdio.h>
#define N 10

main()
{
  int p[N];
  int i, art = 0, per = 0 ;

  for (i=0; i<N; i++)
    cin >> p[i];

  for (i=0; i<N; i++)
    if (p[i] % 2 == 0)
      art = art + 1;
    else
      per = per + 1;

  cout << "Πλήθος άρτιων = " << art << endl;
  cout << "Πλήθος περιττών = " << per << endl;
}
```

Δήλωση

Ανάγνωση Τιμών

Επεξεργασία

# Δισδιάστατοι Πίνακες & Functional Programming

P	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										

Σύνολο από μονοδιάστατους πίνακες, συνθέτοντας ένα πίνακα από γραμμές και στήλες

Χρήση με το όνομα και δύο δείκτες έναν για τις γραμμές και έναν για τις στήλες

# Εργασίες σε διδιάστατο πίνακα

Ορισμός διδιάστατου πίνακα ακεραίων

```
#define M 8
#define N 10
main()
{
    int p[M][N];
    .
    .
    .
    .
}
```

Διάβαση διδιάστατου πίνακα.

```
for (i=0; i<M; i++)
    for (j=0; j<N; j++)
        cin >> p[i][j];
```

Εμφάνιση διδιάστατου πίνακα.

```
for (i=0; i<M; i++)
    for (j=0; j<N; j++)
        cout << p[i][j];
```

Άθροισμα στοιχείων διδιάστατου πίνακα.

```
sum = 0;
for (i=0; i<M; i++)
    for (j=0; j<N; j++)
        sum = sum + p[i][j];
```

Άθροισμα στοιχείων διδιάστατου πίνακα κατά γραμμή.

```
for (i=0; i<M; i++)
{
    row[i] = 0;
    for (j=0; j<N; j++)
        row[i] = row[i] + p[i][j];
}
```

Άθροισμα στοιχείων διδιάστατου πίνακα κατά στήλη.

```
for (j=0; j<N; j++)
{
    col[j] = 0;
    for (i=0; i<M; i++)
        col[j] = col[j] + p[i][j];
}
```

# Πίνακες και αντικειμενοστραφής προγραμματισμός

---

Στον αντικειμενοστραφή επιπροσθέτως οι πίνακες εμφανίζονται

- A. Ως στοιχεία της κλάσης (κάποιο ή κάποια χαρακτηριστικό της κλάσης να είναι πίνακας)
- B. Ως πίνακας αντικειμένων

# Παράδειγμα Α.

```
#include <iostream.h>
```

```
const int N = 8;
```

```
class Student
```

```
{
```

```
private:
```

```
int am;
```

```
float vath[N];
```

```
public:
```

```
Student()
```

```
{
```

```
int i;
```

```
am = 0;
```

```
for (i=0; i<N; i++)
```

```
    vath[i] = 0;
```

Δήλωση  
χαρακτηριστικού  
ως πίνακας

Constructor

Μηδενισμός πίνακα βαθμολογιών  
κατά την δημιουργία αντικειμένου

```
};  
void readData()  
{  
    int i;  
    cout << "Δώσε Α.Μ.:";  
    cin >> am;  
    cout << "Δώσε βαθμούς:";  
    for (i=0; i<N; i++)  
        cin >> vath[i];  
}  
float getAvg()  
{  
    int i;  
    float sum = 0, avg;  
    for (i=0; i<N; i++)  
        sum += vath[i];  
    avg = sum/N;  
    return avg;  
}  
float getMin()  
{  
    int i;  
    float min;  
    min = vath[0];  
    for (i=1; i<N; i++)  
        if (vath[i] < min)  
            min = vath[i];  
    return min;  
}  
float getMax()  
{  
    int i;  
    float max;  
    max = vath[0];  
    for (i=1; i<N; i++)  
        if (vath[i] > min)  
            max = vath[i];  
    return max;  
}  
};  
main()  
{  
    Student s1;  
  
    s1.readData();  
    cout << "Μέσος όρος βαθμολογίας = " << s1.getAvg() << endl;  
    cout << "Μεγαλύτερη βαθμολογία = " << s1.getMax() << endl;  
    cout << "Μικρότερη βαθμολογία = " << s1.getMin() << endl;  
}
```

Ανάγνωση τιμών για  
τον πίνακα  
βαθμολογιών

Επεξεργασία : Υπολογισμός  
Αθροίσματος πίνακα  
βαθμολογιών

Επεξεργασία : Εύρεση  
μικρότερου βαθμού στον  
πίνακα βαθμολογιών

Επεξεργασία : Εύρεση  
μεγαλύτερου βαθμού στον  
πίνακα βαθμολογιών



# Παράδειγμα Β.

```
#include <iostream.h>
const int N = 10;
class Employee
{
private:
    int am;
    char name[20];
    float misthos;
public:
    void readData()
    {
        cout << "Δώσε Α.Μ.:";
        cin >> am;
        cout << "Δώσε όνομα:";
        cin >> name;
        cout << "Δώσε μισθό:";
        cin >> misthos;
    }
    void printData()
    {
        cout << "Α.Μ.: " << am << endl;
        cout << "Όνομα: " << name << endl;
        cout << "Μισθός: " << misthos << endl;
    }
};
```

Πίνακας ως  
χαρακτηριστικό  
κλάσης

main()

```
{
    Employee emp[N];
    int i, armit;
    bool found = false;

    cout << "Δώσε πληροφορίες για " << N << " εργαζόμενους:" << endl;
    for (i=0; i<N; i++)
    {
        cout << endl << "Εργαζόμενος " << i+1 << ":" << endl;
        emp[i].readData();

        cout << "Δώσε Α.Μ. για αναζήτηση:";
        cin >> armit;
        i = 0;
        found = false;
        while (i<N && found == false)
        {
            if (emp[i].am == armit)
                found = true;
            else
                i++;
        }
        if (found == true)
            emp[i].printData();
        else
            cout << "Ο συγκεκριμένος εργαζόμενος δεν υπάρχει." << endl;
    }
}
```

Πίνακας από N  
αντικείμενα κλάσης  
Employee

Ανάγνωση τιμών για  
κάθε αντικείμενο στο  
πίνακα

Αναζήτηση Employee  
στο πίνακα  
αντικειμένων με βάση  
το Α.Μ.