Δομές Δεδομένων

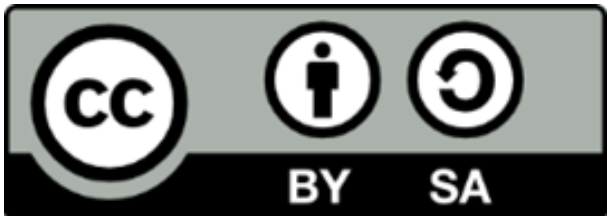
Ενότητα 1: Εισαγωγή-Υλοποίηση του ΑΤΔ Σύνολο με Πίνακα

Καθηγήτρια Μαρία Σατρατζέμη

Τμήμα Εφαρμοσμένης Πληροφορικής

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 Πανεπιστήμιο Μακεδονίας» έχει χρηματοδοτήσει μόνο τη 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.

Σκοποί ενότητας

Σκοπός της ενότητας είναι

- Να παρουσιάσει τον ΑΤΔ Σύνολο πως ορίζεται, τις πράξεις που περιέχει και πως υλοποιείται στη γλώσσα C.
- Υλοποίηση του ΑΤΔ σύνολο στη C χρησιμοποιώντας το τύπο δεδομένων πίνακα.
- Υλοποίηση των βασικών πράξεων το ΑΤΔ Σύνολο με συναρτήσεις στη C.

Περιεχόμενα ενότητας

- Ορισμός συνόλου - παραδείγματα
- Πράξεις στα σύνολα
- Το σύνολο ως Αφηρημένος Τύπος Δεδομένων (ΑΤΔ)
- Ορισμός του ΑΤΔ Σύνολο
- Βασικές λειτουργίες/πράξεις του ΑΤΔ Σύνολο
- Υλοποίηση του ΑΤΔ σύνολο με πίνακα
- Παράδειγμα
- Βασικές πράξεις/λειτουργίες
- Πακέτο για τον ΑΤΔ ΣΥΝΟΛΟ με πίνακα

Ορισμός συνόλου – παραδείγματα[1]

- Σε ένα πρόβλημα με σύνολα, τα στοιχεία επιλέγονται από ένα δεδομένο σύνολο που ονομάζεται καθολικό σύνολο (universal set) για το πρόβλημα αυτό.
- Παράδειγμα: για το σύνολο των φωνηέντων ή το σύνολο {χ, ψ, ζ} το καθολικό σύνολο μπορεί να είναι το σύνολο των γραμμάτων του Ελληνικού αλφαβήτου..

Ορισμός συνόλου – παραδείγματα[2]

- Ένα σύνολο καθορίζεται από τη σχέση μέλους (membership relation). Δεδομένου ενός συνόλου Α και ενός οποιουδήποτε αντικειμένου x του καθολικού συνόλου, θα πρέπει να είμαστε σε θέση να καθορίσουμε αν:
- το x ανήκει στο σύνολο Α: $x \in A$
- το x δεν ανήκει στο σύνολο Α: $x \notin A$

Πράξεις στα σύνολα - Ένωση

- **Ένωση** (union)
- **A** $\cup$ **B**
- η ένωση των συνόλων **A** και **B** είναι το σύνολο των στοιχείων που ανήκουν:
- στο σύνολο **A** ή
- στο σύνολο **B** ή
- και στα δύο σύνολα.

Πράξεις στα σύνολα- Τομή

- **Τομή** (intersection)
- **A ∩ B**
- η τομή των συνόλων **A** και **B** είναι το σύνολο των στοιχείων που ανήκουν και στα δύο σύνολα.

Πράξεις στα σύνολα- Διαφορά

- **Διαφορά** (difference)
- **A - B**
- η διαφορά των συνόλων **A - B** είναι το σύνολο που περιλαμβάνει τα στοιχεία του συνόλου **A** που δεν ανήκουν στο σύνολο **B**

Το σύνολο ως Αφηρημένος Τύπος Δεδομένων (ΑΤΔ) [1]

- Ένα σύνολο, από τη σκοπιά των δομών δεδομένων, είναι μια μη διατεταγμένη συλλογή αντικειμένων που ονομάζονται στοιχεία και στα οποία ορίζονται οι βασικές πράξεις του μέλους, της ένωσης, της τομής και της διαφοράς.
- Εφόσον τα στοιχεία ενός συνόλου είναι μη διατεταγμένα δεν έχει νόημα να μιλάμε για το 1ο, το 2ο στοιχείο κ.τ.λ. Για παράδειγμα, το σύνολο {2, 4, 6, 8} είναι το ίδιο με το σύνολο {4, 2, 8, 6} ή {8, 6,2, 4}.

Το σύνολο ως Αφηρημένος Τύπος Δεδομένων (ΑΤΔ) [2]

- Επομένως, σε αντίθεση με τα στοιχεία ενός πίνακα, τα στοιχεία ενός συνόλου δεν είναι άμεσα προσπελάσιμα.
- Επίσης, τα σύνολα έχουν 2 βασικές διαφορές σε σχέση με τις εγγραφές. Τα δεδομένα που αποθηκεύονται σε μια εγγραφή είναι άμεσα προσπελάσιμα και μπορεί να είναι διαφορετικού τύπου.

Το σύνολο ως Αφηρημένος Τύπος Δεδομένων (ΑΤΔ) [3]

- Μια συλλογή μοναδικών στοιχείων του ίδιου τύπου, τα οποία δεν έχουν καμία σχέση μεταξύ τους.
- Βασικές λειτουργίες/πράξεις:
  - Δημιουργία ενός κενού συνόλου
  - Δημιουργία καθολικού συνόλου
  - Εισαγωγή στοιχείου
  - Διαγραφή στοιχείου
  - Μέλος
  - Κενό σύνολο
  - Ίσα σύνολα
  - Υποσύνολο
  - Ένωση συνόλων
  - Τομή συνόλων

Λειτουργίες του ΑΤΔ Σύνολο [1]

Δημιουργία ενός κενού συνόλου

- Λειτουργία: Δημιουργεί ένα σύνολο χωρίς στοιχεία, δηλαδή το κενό σύνολο.
- Επιστρέφει: Το κενό σύνολο.

Δημιουργία καθολικού συνόλου

- Δέχεται: Ένα σύνολο.
- Λειτουργία: Δημιουργεί ένα σύνολο με όλα τα στοιχεία.
- Επιστρέφει: Το καθολικό σύνολο που δημιουργήθηκε.

Λειτουργίες του ΑΤΔ Σύνολο [2]

Εισαγωγή στοιχείου

- Δέχεται: Ένα σύνολο και ένα στοιχείο.
- Λειτουργία: Εισάγει το στοιχείο στο σύνολο.
- Επιστρέφει: Το τροποποιημένο σύνολο.

Διαγραφή στοιχείου

- Δέχεται: Ένα σύνολο και ένα στοιχείο.
- Λειτουργία: Διαγράφει το στοιχείο από το σύνολο.
- Επιστρέφει: Το τροποποιημένο σύνολο.

Λειτουργίες του ΑΤΔ Σύνολο [3]

Μέλος

- Δέχεται: Ένα σύνολο και ένα στοιχείο.
- Λειτουργία: Ελέγχει αν το στοιχείο ανήκει στο σύνολο.
- Επιστρέφει: Επιστρέφει TRUE αν το στοιχείο ανήκει στο σύνολο και FALSE διαφορετικά. Διαγραφή στοιχείου

Λειτουργίες του ΑΤΔ Σύνολο [4]

Κενό σύνολο

- Δέχεται: Ένα σύνολο.
- Λειτουργία: Ελέγχει αν το σύνολο είναι κενό.
- Επιστρέφει: Επιστρέφει TRUE αν το σύνολο είναι κενό και FALSE διαφορετικά.

Ίσα σύνολα

- Δέχεται: Δύο σύνολα.
- Λειτουργία: Ελέγχει αν τα σύνολα είναι ίσα.
- Επιστρέφει:Επιστρέφει TRUE αν τα δύο σύνολα έχουν τα ίδια στοιχεία και FALSE διαφορετικά.

Λειτουργίες του ΑΤΔ Σύνολο [5]

Υποσύνολο

- Δέχεται: Δύο σύνολα S1 και S2.
- Λειτουργία: Ελέγχει αν το σύνολο S1 είναι υποσύνολο του S2.
- Επιστρέφει: Επιστρέφει TRUE αν το σύνολο S1 είναι ένα υποσύνολο του S2, δηλαδή αν κάθε στοιχείο του S1 είναι και στοιχείο του S2

Λειτουργίες του ΑΤΔ Σύνολο [6]

Ένωση συνόλων

- Δέχεται: Δύο σύνολα S1 και S2.
- Λειτουργία: Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν ή στο S1 ή στο S2 ή και στα δύο σύνολα.
- Επιστρέφει: Επιστρέφει το σύνολο που προκύπτει από την ένωση των συνόλων S1 και S2.

Λειτουργίες του ΑΤΔ Σύνολο [7]

Τομή συνόλων

- Δέχεται: Δύο σύνολα S1 και S2.
- Λειτουργία: Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν και στα δύο σύνολα S1 και S2.
- Επιστρέφει: Επιστρέφει το σύνολο που προκύπτει από την τομή των συνόλων S1 και S2.

Λειτουργίες του ΑΤΔ Σύνολο [8]

Διαφορά συνόλων

- Δέχεται: Δύο σύνολα S1 και S2.
- Λειτουργία: Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν στο σύνολο S1 και δεν ανήκουν στο S2.
- Επιστρέφει: Επιστρέφει το σύνολο που προκύπτει από την διαφορά

Υλοποίηση του ΑΤΔ σύνολο με πίνακα

- Στην παρούσα ενότητα θα παρουσιάσουμε τον τρόπο υλοποίησης του ΑΤΔ σύνολο στη C χρησιμοποιώντας το τύπο δεδομένων πίνακα.
- Συγκεκριμένα, θα χρησιμοποιήσουμε ένα λογικό πίνακα S, όπου S[i] είναι TRUE αν το στοιχείο που αντιστοιχεί στο i-οστό στοιχείο του καθολικού συνόλου ανήκει στο σύνολο S, διαφορετικά είναι FALSE.

Παράδειγμα

θέση	τιμή
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE
10	TRUE

Αν θεωρήσουμε το σύνολο

OddNumbers = {1, 3, 5, 7,9},

όπου το καθολικό σύνολο είναι τα ψηφία 0..9, τότε αυτό μπορεί να παρασταθεί με τον πίνακα 10 θέσεων που παρουσιάζεται αριστερά.

Η 1η θέση του πίνακα αντιστοιχεί στο 1ο στοιχείο του καθολικού συνόλου, δηλαδή στο ψηφίο 0, η 2η στο ψηφίο 1 κ.τ.λ.

Δηλώσεις για υλοποίηση του ΑΤΔ Σύνολο

Για την υλοποίηση του ΑΤΔ σύνολο με πίνακα χρησιμοποιούνται οι παρακάτω δηλώσεις:

```
#define megisto_plithos 10          / *μέγιστο πλήθος
                                      στοιχείων συνόλου*/

typedef enum {
        FALSE, TRUE
} boolean;
typedef boolean typos_synolou[megisto_plithos+1];
typedef int stoixeio_synolou;
```

Βασικές πράξεις/λειτουργίες [1]

- Οι βασικές πράξεις/λειτουργίες που συνδέονται με τα σύνολα υλοποιούνται εύκολα χρησιμοποιώντας τη δομή δεδομένων του πίνακα.
- Στη συνέχεια περιγράφεται ο τρόπος υλοποίησής τους, θεωρώντας ότι οι μεταβλητές synolo, s1, s2, enosi, tomi και diafora είναι μεταβλητές πίνακα τύπου typos_synoloy.

Βασικές πράξεις/λειτουργίες [2]

- **Δημιουργία ενός κενού συνόλου** (Dimiourgia): για τη δημιουργία του κενού συνόλου, δηλαδή ενός συνόλου που δεν έχει καθόλου στοιχεία, εκχωρείται σε όλες τις θέσεις του πίνακα synolo η τιμή FALSE.
- **Δημιουργία καθολικού συνόλου** (Katholiko): για τη δημιουργία του καθολικού συνόλου, δηλαδή του συνόλου που περιλαμβάνει όλα τα στοιχεία του συγκεκριμένου τύπου βάσης που έχει δηλωθεί, εκχωρείται σε όλες τις θέσεις του πίνακα synolo η τιμή TRUE.

Βασικές πράξεις/λειτουργίες [3]

- **Εισαγωγή στοιχείου (Eisagogi)**: για την εισαγωγή, σε ένα σύνολο, του στοιχείου που βρίσκεται στην i-οστή θέση του καθολικού συνόλου απλά εκχωρούμε στην i-οστή θέση του πίνακα synolo την τιμή TRUE..
- **Διαγραφή στοιχείου (Diagrafi)**: για την διαγραφή ενός στοιχείου από ένα σύνολο εκχωρούμε στην θέση του πίνακα synolo που βρίσκεται το στοιχείο την τιμή FALSE.

Βασικές πράξεις/λειτουργίες [4]

- **Μέλος (Melos)**: για να διαπιστώσουμε αν ένα στοιχείο είναι μέλος ενός συνόλου, ελέγχουμε την τιμή που υπάρχει στην θέση του πίνακα synolo στην οποία αντιστοιχεί το στοιχείο. Αν η τιμή της συγκεκριμένης θέσης του πίνακα είναι TRUE τότε το στοιχείο είναι μέλος του συνόλου, διαφορετικά όχι.
- **Κενό (KenoSynolo)**: για να ελέγξουμε αν ένα σύνολο είναι κενό εξετάζουμε τα στοιχεία του πίνακα synolo μέχρι να βρούμε:
  - ότι κάποιο στοιχείο έχει τιμή TRUE, γεγονός που σημαίνει ότι το σύνολο δεν είναι κενό, ή
  - να εξαντληθούν όλα τα στοιχεία του πίνακα, γεγονός που σημαίνει ότι το σύνολο είναι κενό.

Βασικές πράξεις/λειτουργίες [5]

- **΄Ενωση (EnosiSynolou)**: για να βρούμε την ένωση δύο συνόλων s1 και s2 εξετάζουμε τα αντίστοιχα στοιχεία των s1 και s2 (δηλαδή τα στοιχεία που βρίσκονται στις ίδιες θέσεις των πινάκων αυτών) και αν ένα τουλάχιστον από αυτά έχει την τιμή TRUE τότε εκχωρούμε την τιμή TRUE και στην αντίστοιχη θέση του συνόλου της τομής, έστω enosi, διαφορετικά εκχωρούμε την τιμή FALSE.

Βασικές πράξεις/λειτουργίες [6]

- **Τομή (TomiSynolou)**: για να βρούμε την τομή δύο συνόλων s1 και s2, εξετάζουμε τα αντίστοιχα στοιχεία των s1 και s2 και αν και τα 2 έχουν την τιμή TRUE τότε εκχωρούμε την τιμή TRUE και στην αντίστοιχη θέση του συνόλου της τομής, έστω tomi, διαφορετικά εκχωρούμε την τιμή FALSE.

Βασικές πράξεις/λειτουργίες [7]

- **Διαφορά (DiaforaSynolou)**: για να βρούμε τη διαφορά s1 - s2 δύο συνόλων, εξετάζουμε τα αντίστοιχα στοιχεία των s1 και s2 και αν και για κάθε στοιχείο που είναι μέλος του s1 και δεν είναι μέλος του s2 εκχωρούμε την τιμή TRUE στην αντίστοιχη θέση του συνόλου της διαφοράς, έστω diafora, και σε κάθε άλλη περίπτωση την τιμή FALSE.

Πακέτο για τον ΑΠΔ Σύνολο με Πίνακα [1]

```
//  Filename: SetADT.h
#define megisto_plithos 10          / *μέγιστο πλήθος
                                      στοιχείων συνόλου*/
typedef enum {
                FALSE, TRUE
} boolean;
typedef boolean
   typos_synolou[megisto_plithos+1];
typedef int stoixeio_synolou;.
```

Πακέτο για τον ΑΠΔ Σύνολο με Πίνακα [2]

```
void Dimiourgia(typos_synolou synolo);
void Katholiko(typos_synolou synolo);
void Eisagogi(stoixeio_synolou stoixeio,
                typos_synolou synolo);
void Diagrafi(stoixeio_synolou stoixeio,
                typos_synolou synolo);
boolean Melos(stoixeio_synolou stoixeio,
              typos_synolou synolo);
```

Πακέτο για τον ΑΠΔ Σύνολο με Πίνακα [3]

```
boolean KenoSynolo(typos_synolou synolo);
boolean IsaSynola(typos_synolou s1,
    typos_synolou s2);
boolean Yposynolo(typos_synolou s1,
    typos_synolou s2);
void EnosiSynolou(typos_synolou s1,
    typos_synolou s2, typos_synolou enosi);
```

Πακέτο για τον ΑΠΔ Σύνολο με Πίνακα [4]

```
void TomiSynolou(typos_synolou s1,
    typos_synolou s2, typos_synolou tomi);
void DiaforaSynolou(typos_synolou s1,
    typos_synolou s2, typos_synolou diafora);
```

Δημιουργία Συνόλου

```
void Dimiourgia(typos_synolou synolo)
/* Λειτουργία: Δημιουργεί ένα σύνολο χωρίς στοιχεία,
δηλαδή το κενό σύνολο.
   Επιστρέφει: Το κενό σύνολο.*/
{
    stoixeio_synolou i;

    for (i = 1; i <= megisto_plithos; i++)
        synolo[i] = FALSE;
}
```

Καθολικό Σύνολο

```
void Katholiko(typos_synolou synolo)
/*  Δέχεται: Ένα σύνολο.
    Λειτουργία: Δημιουργεί ένα σύνολο με όλα τα στοιχεία
    παρόντα, έτσι όπως ορίστηκε στο τμήμα δηλώσεων του
    προγράμματος.
    Επιστρέφει:Το καθολικό σύνολο που δημιουργήθηκε.*/
{
    stoixeio_synolou i;

    for (i = 1; i <= megisto_plithos; i++)
        synolo[i] = TRUE;
}
```

Εισαγωγή[1]

```
void Eisagogi(stoixeio_synolou stoixeio,
                        typos_synolou synolo)
/*Δέχεται: Ένα σύνολο και ένα στοιχείο.
   Λειτουργία: Εισάγει το στοιχείο στο σύνολο.
   Επιστρέφει: Το τροποποιημένο σύνολο.*/
{
    synolo[stoixeio] = TRUE;
}
```

Εισαγωγή[2]

void Diagrafi(stoixeio_synolou stoixeio,
typos_synolou synolo)

```
/*Δέχεται: Ένα σύνολο και ένα στοιχείο.
   Λειτουργία: Διαγράφει το στοιχείο από το
    σύνολο.
   Επιστρέφει: Το τροποποιημένο σύνολο.*/
{
   synolo[stoixeio] =FALSE;
}
```

Μέλος

boolean Melos(stoixeio_synolou stoixeio,
typos_synolou synolo)

/*Δέχεται: Ένα σύνολο και ένα στοιχείο.

Λειτουργία: Ελέγχει αν το στοιχείο είναι μέλος του συνόλου.

Επιστρέφει: Επιστρέφει TRUE αν το στοιχείο είναι μέλος του και FALSE διαφορετικά.*/

```
{
   return synolo[stoixeio];
}
```

Κενό Σύνολο [1]

boolean KenoSynolo(typos_synolou synolo)

/*Δέχεται: Ένα σύνολο.

Λειτουργία: Ελέγχει αν το σύνολο είναι κενό.

Επιστρέφει: Επιστρέφει TRUE αν το σύνολο είναι κενό και FALSE διαφορετικά.*/

```
{
    stoixeio_synolou i; boolean keno;
    keno=TRUE; i = 1;
```

Κενό Σύνολο [2]

```
while (i <= megisto_plithos  && keno)
{
        if (Melos(i, synolo))
                keno = FALSE;
        else
                i++;
    }
    return keno;
}
```

Ίσα Σύνολα [1]

boolean IsaSynola(typos_synolou s1,
typos_synolou s2)

/*Δέχεται: Δύο σύνολα s1 και s2.

Λειτουργία: Ελέγχει αν τα δύο σύνολα είναι ίσα.

Επιστρέφει: Επιστρέφει TRUE αν τα δύο σύνολα έχουν τα ίδια στοιχεία και FALSE διαφορετικά.*)/

{

stoixeio_synolou i; boolean isa;

isa = TRUE; i=1;

Ισα Σύνολα [2]

```
while ((i <= megisto_plithos) && isa) {
        if (Melos(i,s1) != Melos(i,s2))
            isa = FALSE;
        else
            i++;
   }
    return isa;
}
```

Υποσύνολο [1]

boolean Yposynolo(typos_synolou s1,
typos_synolou s2)

/*Δέχεται: Δύο σύνολα s1 και s2.

Λειτουργία: Ελέγχει αν το σύνολο s1 είναι υποσύνολο του s2.

Επιστρέφει: Επιστρέφει TRUE αν το σύνολο s1 είναι ένα υποσύνολο του s2, δηλαδή αν κάθε στοιχείο του s1 είναι και στοιχείο του s2.*/

```
{
    stoixeio_synolou i;  boolean yposyn;
    yposyn = TRUE; i=1;
```

Υποσύνολο [2]

```
while (i <= megisto_plithos && yposyn) {
        if (Melos(i, s1) && !Melos(i, s2))
            yposyn = FALSE;
        else
            i++;
    }
    return yposyn;
}
```

Ένωση [1]

void EnosiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou enosi)

/*Δέχεται: Δύο σύνολα s1 και s2.

Λειτουργία: Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν ή στο s1 ή στο s2 ή και στα δύο σύνολα.

Επιστρέφει: Επιστρέφει το σύνολο enosi που προκύπτει από την ένωση των συνόλων s1 και s2.*/

΄Ενωση [2]

```
{
    stoixeio_synolou i;
    for (i = 1; i <= megisto_plithos; i++)
        enosi[i] = Melos(i, s1) || Melos(i, s2);
}
```

Τομή [1]

void TomiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou tomi)

/*Δέχεται: Δύο σύνολα s1 και s2.

Λειτουργία: Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν και στα δύο σύνολα s1 και s2.

Επιστρέφει: Επιστρέφει το σύνολο tomi που προκύπτει από την τομή των συνόλων s1 και s2.*/

Τομή [2]

```
{
    stoixeio_synolou i;
    for (i = 1; i <= megisto_plithos; i++)
                tomi[i] = Melos(i, s1) && Melos(i, s2);
}
```

Διαφορά [1]

void DiaforaSynolou(typos_synolou s1, typos_synolou s2, typos_synolou diafora)

/*Δέχεται:Δύο σύνολα s1 και s2.

Λειτουργία: Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν στο σύνολο s1 και δεν ανήκουν στο s2.

Επιστρέφει: Επιστρέφει το σύνολο diafora που προκύπτει από την διαφορά των συνόλων s1-s2.*/

Διαφορά [2]

```
{
    stoixeio_synolou i;

    for (i = 1; i <= megisto_plithos; i++)
        diafora[i] = Melos(i, s1) && (!Melos(i, s2));
}
```

Τέλος Ενότητας

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