

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Λειτουργικά Συστήματα

Φροντιστηριακή ενότητα # 1: Εισαγωγή στη
γλώσσα C για γνώστες της γλώσσας Java

Τμήμα: Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Σκοποί ενότητας

- Διαφορές, ομοιότητες και «ιδιαιτερότητες» σε σχέση με τη Java.
- Αναφορές και στη C++ όπου κρίνεται σκόπιμο.
- Δείκτες για διαχείριση της μνήμης.
- Πλεονεκτήματα της C για το προγραμματισμό συστημάτων.
- Βασικά παραδείγματα κώδικα.
- Μεταγλώττιση προγραμμάτων σε C.

Περιεχόμενα ενότητας

- Εισαγωγή
- Βασικές διαφορές από τη Java
- Στοιχεία της C με παραδείγματα
- Μεταγλώττιση

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Εισαγωγή

Μάθημα: Λειτουργικά Συστήματα, **Φροντιστηριακή Ενότητα # 1:**
Προγραμματισμός με τη γλώσσα C για γνώστες της γλώσσας Java

Τμήμα: Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Προτεινόμενα συγγράμματα (1 από 2)

- **Βιβλία**

- **The C Programming Language** από τους **Brian W. Kernighan, Dennis M. Ritchie**
- **C Programming: A Modern Approach** από τον **K. N. King**
- **The C Puzzle Book** από τον **Feuer**
- **C for Dummies Vol. I** του **Dan Gookin**
 - **C for Dummies Vol. II**

Προτεινόμενα συγγράμματα (2 από 2)

- **Παραδείγματα online**
 - Ίσως σας βοηθήσουν ερωτήματα σύγκρισης με τη Java
 - **Συνιστώνται ιδιαίτερα:**
 - [Wikibooks](#)
 - C for Java programmers
 - [του J. Maasen](#)
 - [του Niranjan Nagarajan](#)
 - [Top tips to help the Java programmer code in C](#)

Ιστορική αναδρομή (1 από 2)

- Δημιουργήθηκε το 1972 (Dennis Ritchie)
- Προήλθε από τις γλώσσες B και BCPL
- Τυποποιήθηκε στο διάστημα 1983-1989
 - “ANSI C” ή “C89”
 - “C99”, “C11” και “C17”
- Η σύνταξη της C αποτελεί τη βάση για τις γλώσσες C++, C#, Objective C, Java
 - Η C++ είναι ουσιαστικά επέκταση της C στον αντικειμενοστρεφή προγραμματισμό

Ιστορική αναδρομή (2 από 2)

- Οι ειδικοί/έμπειροι προγραμματιστές θεωρούν ότι η C++ και η C θα πρέπει να αντιμετωπίζονται ως διαφορετικές γλώσσες
 - Όχι μεικτός κώδικας C και C++
 - Όχι πρακτικές της C στη C++

Σχέση με το Unix/Linux

- Η C αναπτύχθηκε παράλληλα με το Unix
- Αρχικά η ανάπτυξη του Unix ήταν σε assembly
 - Πολύ αργή ανάπτυξη κώδικα
 - Κώδικας γραμμένος για συγκεκριμένη αρχιτεκτονική (μηχάνημα)
- Πλεονεκτήματα της ανάπτυξης του Unix σε C
 - Φορητότητα (portability)
 - Ευκολία και ταχύτητα ανάπτυξης, π.χ:
 - Δομές επανάληψης, κλήσεις συναρτήσεων και έτοιμων βιβλιοθηκών ..

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Βασικές διαφορές από τη Java

Μάθημα: Λειτουργικά Συστήματα, **Φροντιστηριακή Ενότητα # 1:**
Προγραμματισμός με τη γλώσσα C για γνώστες της γλώσσας Java

Τμήμα: Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Διαδικαστικές και Αντικειμενοστραφείς γλώσσες (1 από 2)

- Η γλώσσα C **δεν** είναι **αντικειμενοστραφής**
- Δεν υπάρχουν
 - Κλάσεις, άρα ούτε αντικείμενα
 - Διεπαφές (interfaces)
 - Πακέτα βιβλιοθηκών κλάσεων
 - Πολυμορφισμός (polymorphism)
 - Ενθυλάκωση (encapsulation)
 - Μέθοδοι (?), Κληρονομικότητα, σχέσεις Is-A, Has-A, κ.ο.κ.

Διαδικαστικές και Αντικειμενοστραφείς γλώσσες (2 από 2)

- Η γλώσσα C είναι διαδικαστική (procedural)
 - Τα προγράμματα δομούνται μόνο από συναρτήσεις (C functions)
 - Οι συναρτήσεις δρουν (ρητά) μόνο επί των δεδομένων πάνω στα οποία καλούνται
 - *Ερώτηση: Ποια η διαφορά από τις μεθόδους μιας αντικειμενοστραφούς γλώσσας;*
- Υπάρχουν δομές ή πρακτικές που δεν απαντώνται στη Java: structs, unions, δείκτες (pointers) ακόμη και void δείκτες ..

Χειρισμός «εξαιρέσεων» / λαθών

- Δεν υπάρχει μηχανισμός διαχείρισης εξαιρέσεων (exceptions)
 - Αυτό είναι χαρακτηριστικό των αντικειμενοστραφών γλωσσών όπως η Java και η C++
- Υπάρχει βιβλιοθήκη αναγνώρισης λαθών
 - http://www.tutorialspoint.com/cprogramming/c_error_handling.htm

Διαφορές στην οργάνωση και τη δομή (1 από 3)

Java

1. Κώδικας σε **κλάσεις**
 - Κατά προτίμηση ένα αρχείο ανά κλάση
 - Πεδία, μέθοδοι, και αναφορές οργανωμένα σε κλάσεις

C

1. Κώδικας σε διάφορες **συναρτήσεις**
 - Δηλώσεις και ορισμοί συναρτήσεων οργανώνονται σε (πολλά) αρχεία κεφαλίδας και πηγαίου κώδικα (**header and source files**)
 - Ίδια λογική, θέματα και λύσεις με τη C++

Διαφορές στην οργάνωση και τη δομή (2 από 3)

Java

2. Εκτέλεση βασικού νήματος μέσα σε μια κλάση

```
public static void  
main(String []args){  
System.out.println("Hello World");  
}
```

C

2. Εκτέλεση μιας main συνάρτησης από ένα αρχείο κώδικα

```
int main(int argc, char**  
argv){  
printf("Hello World\n");  
return 0;  
}
```

Διαφορές στην οργάνωση και τη δομή (3 από 3)

Java

3. Σύνδεση + Κλήσεις βιβλιοθηκών κώδικα από jar αρχεία
- jar: java archive σε μορφή zip
 - Συμπιεσμένος εκτελέσιμος κώδικας

```
import java.io.*;
import my.external.lib.*;
```

C

3. Σύνδεση (linking) με εξωτερικό κώδικα
- Αρχεία κεφαλίδας
 - Ο εκτελέσιμος κώδικας συνδέεται χωριστά

```
#include <stdio.h>
#include "my_header.h"
```


Βασικά αρχεία κεφαλίδας

Αρχείο	Εξήγηση / αντιστοιχία
<code>stdio.h</code>	<code>java.io.*</code>
<code>stdlib.h</code>	<code>std.</code> συναρτήσεις και μακροεντολές
<code>math.h</code>	<code>java.lang.Math</code>
<code>string.h</code>	<code>java.lang.String</code>
<code>time.h</code>	<code>System.</code> <code>currentTimeMillis();</code>

Μήκη αριθμητικών τύπων (1 από 2)

Java

Το μέγεθος των βασικών τύπων είναι γνωστό:

- `int`: 32 bits
- `short`: 16 bits
- `long`: 64 bits
- `float`: 32 bits
- `double`: 64 bits
- `byte`: 8 bits
- `char`: 16 bits

C

Το μέγεθος **δεν** είναι δεδομένο.
Εξαρτάται από:

- Το μεταγλωττιστή
- Την πλατφόρμα για την οποία έχει γίνει η μεταγλώττιση
- Βλ. παράδειγμα κώδικα με `sizeof()` (Εργαστήριο)

Μήκη αριθμητικών τύπων (2 από 2)

- long long: υπάρχει ανεπίσημα: 64 bits
- long double: υπάρχει ανεπίσημα: 96 bits
- Χρήση του unsigned
 - int i; // -2^{31} έως $2^{31}-1$
 - signed int i; // -2^{31} έως $2^{31}-1$
 - unsigned int i; // 0 έως $2^{32}-1$

Θέματα διαχείρισης μνήμης

- **Τίποτα** δεν διαχειρίζεται/ελέγχεται **αυτόματα** σε ότι αφορά τη μνήμη!
 - Όπως και η C++, η C **δεν** διαθέτει:
 - Διαχειρίσιμη «σκουπιδιών» μνήμης αντίστοιχου του java garbage collector System.gc()
 - Όρια πινάκων, έλεγχο προσπέλασης μνήμης με δείκτες, έλεγχο καλουπώματος (casting) τύπων
 - Επιπλέον, σε **αντίθεση** με τη C++, **δεν** έχει:
 - Λογικές μεταβλητές, δηλαδή, τύπου *bool* (πριν τη C99)
 - Διαχείριση strings ως πίνακες χαρακτήρων

Αναφορές Java και δείκτες της C/C++ (1 από 2)

- Προσοχή! Η έννοια της αναφοράς (reference) στη Java είναι κοντά στην έννοια του C pointer και **όχι** στην έννοια της αναφοράς στη C++
 - «Ατυχής» ομοιότητα στην ορολογία
- Ο πιο απλός τρόπος να ορίσεις τους δείκτες της C/C++:
 - Παρόμοιοι με αναφορές στη Java (δείχνουν κάπου στη μνήμη)
 - Με επιπλέον τη δυνατότητα των αριθμητικών πράξεων (βλ. συνέχεια)

Αναφορές Java και δείκτες της C/C++ (2 από 2)

- Δυνατότητα να αλλάξεις τη τιμή του δεικτη
 - Δηλαδή, την αριθμητική διεύθυνση στη μνήμη
 - Αύξηση του δείκτη, πολλαπλασιασμός, διαίρεση..
 - Πολύ σύνηθες, πχ στους πίνακες
 - Όλα επιτρέπονται! (δυστυχώς και ευτυχώς)
 - Πολλές δυνατότητες και..
 - Πολλές «ευκαιρίες» για λάθη που δύσκολα εκσφαλματώνονται (debugging)

C ή Java;

- Προτιμούμε τη C όταν πρέπει να γίνουν όλα στο χαμηλότερο επίπεδο λεπτομέρειας
 - Πλήρης έλεγχος και ανάγκη για αποδοτικό / ταχύ κώδικα σε κρίσιμα τμήματα, π.χ. για επικοινωνία
 - Χειρισμός θέσεων μνήμης ανά bi
 - Χρήσιμο για καταχωρητές συσκευών
 - Δυνατότητα κλήσεων κώδικα σε assembly
- Παραδείγματα:
 - Οδηγοί συσκευών (drivers)
 - Λειτουργικά συστήματα όπως το Linux

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Στοιχεία της C με παραδείγματα

Μάθημα: Λειτουργικά Συστήματα, **Φροντιστηριακή Ενότητα # 1:**
Προγραμματισμός με τη γλώσσα C για γνώστες της γλώσσας Java

Τμήμα: Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Δεσμευμένες λέξεις-κλειδιά (1 από 3)

double	case
float	while
int	do
short	for
long	break
if	continue
else	void
switch	char
default	return

Δεσμευμένες λέξεις-κλειδιά (2 από 3)

struct	typedef
union	extern
signed	static
unsigned	sizeof
enum	

- struct <> class
 - *όχι* πολυμορφισμός
 - [C++: public vs. private](#)
- [union Vs. struct](#)

Δεσμευμένες λέξεις-κλειδιά (3 από 3)

auto	goto
register	volatile

- [Why is **goto** bad?](#)
 - E. Dijkstra's [“A case against the goto statement”](#)

«Ιδιαιτερότητες»

- **byte**: Δεν υπάρχει. Χρήση `char` => 8 bits
- **bool**: Δεν υπάρχει. Χρήση `int 0` για `false`. Οποιαδήποτε άλλη `int` τιμή εκτός του 0, εκλαμβάνεται ως `true`.
 - `if (0) { // δεν εκτελείται }`.
 - `while(!0) { /* εκτελείται για πάντα */ }`
 - `if (a = 0) { /* δεν εκτελείται ποτέ */ }`
 - Παραδείγματα με αληθείς αποτιμήσεις:
 - `1 && 0 == 0`
 - `1 && -1 == 1`
 - `7 || 25 == 1`
 - `(!3 == 0) == !0`


«Ιδιαιτερότητες»

- `if (y = -1) { /* εκτελείται πάντα */ }`
 - Προσοχή: είναι εκχώρηση, όχι σύγκριση!
- `int x = 1; int y = 2;`
 - `while (x && y) { /* εκτελείται για πάντα */ }`
 - `while (x & y) { /* ποτέ δεν εκτελείται, γιατί; */ }`
 - Περισσότερα περί τελεστών [εδώ](#)
- `int i = 3; while (--i) { /* εκτελείται 2 φορές */ }`
- `int j /* σκουπίδια */; while (--j) { /* εκτελείται άγνωστες φορές */ }`

«Ιδιαιτερότητες»

Λάθος δήλωση μεταβλητών

```
1. void foo(){
2.     int i;
3.     i=5;
4.     int j=9; // η δήλωση
               // πρέπει να είναι
               // στην αρχή!
5. }
```



Ορθή δήλωση μεταβλητών

```
1. void loop (){
2.     int lim, i;
3.     lim = 10;
4.     for (i=0;
           i<lim;i++) {...}
5. }
```

Προσοχή! Το παραπάνω ισχύει γενικά, π.χ. για τις δομές επανάληψης..

Εμβέλεια μεταβλητών (1 από 2)

```
1. int global; // εντός του αρχείου .c
2. void foo(){
3.     int local;
4. }
```

Εμβέλεια μεταβλητών (2 από 2)

```
1. extern int global; // δηλωμένο σε
2.                    // άλλο αρχείο .c
3. static int prvtGlobal; // τοπικά
4.                    // global στο παρόν αρχείο .c
5. void foo(){
6.     static int local;
7.     // τοπική, αλλά διατηρεί την τιμή της
8. }
```


**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Μεταγλώττιση

Μάθημα: Λειτουργικά Συστήματα, **Φροντιστηριακή Ενότητα # 1:**

Προγραμματισμός με τη γλώσσα C για γνώστες της γλώσσας Java

Διδάσκων φροντιστηρίου: Ξενοφών Βασιλάκος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Hello World!

```
/* HelloWorld.c */
```

```
1. #include <stdio.h>  
2.  
3. int main(int argc, char** argv){  
4.     printf("Hello %s\n", "World!");  
5.     return 0;  
6. }
```

GCC

- gcc -Wall HelloWorld.c
- Το εκτελέσιμο που προκύπτει είναι το
 - a.out
- Για την εκτέλεση του προγράμματος:
 - ./a.out
- Πηγές εκπαιδευτικού υλικού GCC
 - <http://matrixprogramming.com/2008/03/compilelink>
 - http://www.givemefish.com/ArticlesAndBook/CheatSheets/GiveMeFish_gcc_CheatSheet.php

`./a.out Hello World!`

/ PrintArgs.c */*

```
1. #include <stdio.h>
2.
3. int main(int argc, char** argv){
4.     int i;
5.     for (i=0; i<argc; i++){
6.         printf("arg[%d]:%s\n", i, argv[i]);
7.     }
8.     return 0;
9. }
```

`./a.out Hello World!`

/ PrintArgs.c */*

```
1. #include <stdio.h>
2.
3. int main(int argc, char** argv){
4.     int i;
5.     for (i=0; i<argc; i++){
6.         printf("arg[%d]:%s\n", i, argv[i]);
7.     }
8.     return 0;
9. }
```

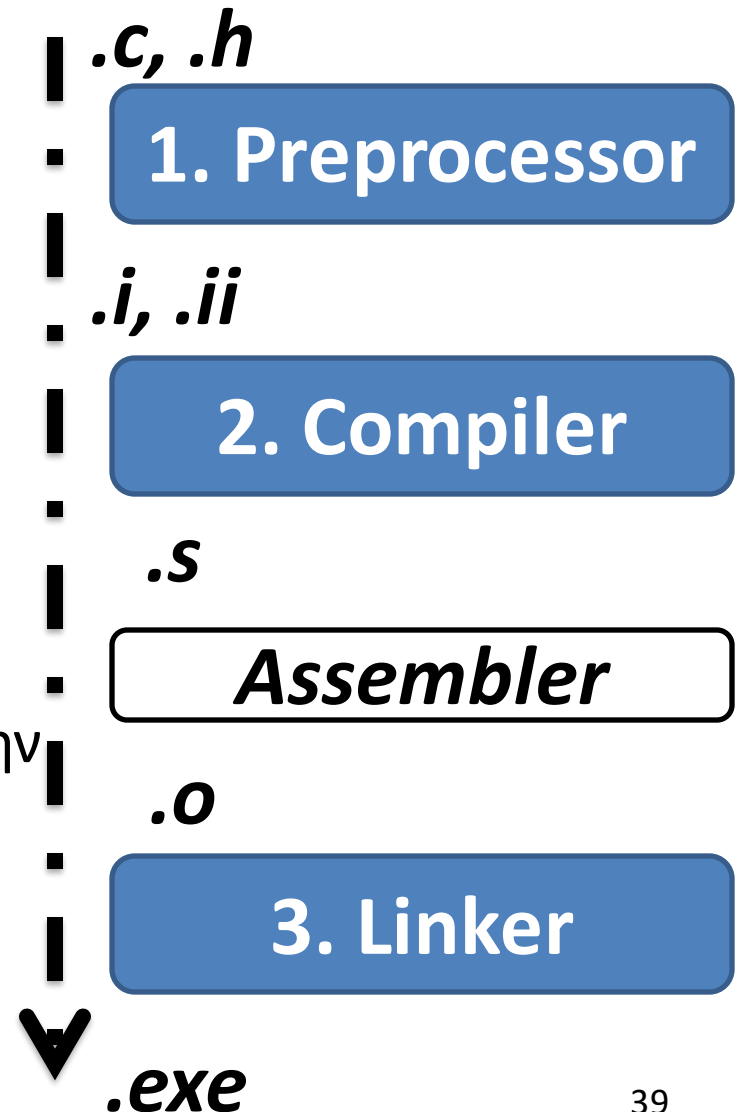
```
> gcc -Wall PrintArgs.c
> ./a.out Hello World!
```

GCC

- `gcc -Wall PrintArgs.c`
- Για την εκτέλεση του προγράμματος:
`./a.out Hello World!`
- Αποτέλεσμα εκτέλεσης προγράμματος
 - `arg[0]: ./a.out`
 - `arg[1]: Hello`
 - `arg[2]: World!`

Προεπεξεργαστής (Preprocessor)

- `#include`
 - Για να γίνει εισαγωγή από τον προεπεξεργαστή των δηλώσεων των συναρτήσεων.
- Τα τρία βήματα της μεταγλώττισης
 1. Προεπεξεργαστής (πχ `#include`, `#if`, `#elif`, `#ifndef`)
 2. Μεταγλωττιστής (*.o αρχείο στην έξοδο)
 3. Σύνδεση. Ο linker συνδέει με βιβλιοθήκες



**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Τέλος Φροντιστηριακής Ενότητας # 1

Μάθημα: Λειτουργικά Συστήματα, **Φροντιστηριακή Ενότητα # 1:**
Προγραμματισμός με τη γλώσσα C για γνώστες της γλώσσας Java,
Τμήμα: Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ