

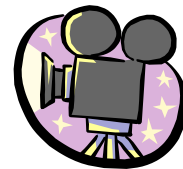


Μεταβλητή Τύποι δεδομένων

Εισαγωγή στον Προγραμματισμό

(CEID_NY131)

Variable and Data Types



Kleanthis Thramboulidis

Prof. of Software and System Engineering

University of Patras

<https://sites.google.com/site/thrambouliskleanthis/>

Βασικές Κατασκευές Γλώσσας Προγραμματισμού

- Αναπαράσταση διεργασιών
 - συνάρτηση
- Για αποθήκευση και ανάκληση πληροφορίας
 - **Μεταβλητές**
 - **Σταθερές**
 - **Τύποι δεδομένων**
- Για επεξεργασία πληροφορίας
 - τελεστές
- Έλεγχο ροής προγράμματος
 - Προτάσεις επανάληψης, διακλάδωσης, ..



Μεταβλητή Ορισμός 1/3

- **Συμβολικό όνομα**
 - που δίνει λύση στο πρόβλημα της αναφοράς σε πληροφορία της κύριας μνήμης του υπολογιστή.
- Το **όνομα της μεταβλητής** μας επιτρέπει να αναφερόμαστε
 - **στη τιμή**, χωρίς να είναι απαραίτητο να γνωρίζουμε την ακριβή διεύθυνση της μνήμης που αυτή είναι αποθηκευμένη,
 - **στην θέση της μνήμης** για να τροποποιήσουμε την τρέχουσα-τιμή της.

```
count = count + 1
```



Μεταβλητή Ορισμός 2/3

- είναι (σύμφωνα με τον Horowitz) μια τετράδα που αποτελείται από
 - ένα όνομα
 - ένα σύνολο ιδιοτήτων
 - μία αναφορά και
 - μία τιμή.



Μεταβλητή Ορισμός 3/3

- **ΟΙ ιδιότητες**
 - συνήθως εκφράζουν το είδος των τιμών που μπορεί να πάρει
 - ορίζονται
 - στη διάρκεια της μεταγλώττισης και δεν μεταβάλλονται (π.χ. C) ή
 - στο χρόνο εκτέλεσης και μεταβάλλονται
- **η αναφορά**
 - προσδιορίζει μια περιοχή αποθήκευσης που είναι μικρότερη ή μεγαλύτερη από μία θέση μνήμης



Δήλωση μεταβλητής στη C 1/2

- **προσδιορίζει** (στη διάρκεια μεταγλώττισης) τις ιδιότητες της μεταβλητής
 - τύπο δεδομένων
 - απαγόρευση αλλαγής τιμής στο χρόνο εκτέλεσης (**const**)
- **παραχωρεί** ένα συμβολικό όνομα
 - σε θέση μνήμης που θα δεσμευτεί στη διάρκεια εκτέλεσης
- **ενημερώνει** τον compiler για
 - πόσα bytes πρέπει να δεσμευθούν στη διάρκεια εκτέλεσης για την αποθήκευση της τιμής της
 - πως αυτά τα bytes πρέπει να ερμηνευθούν



Δήλωση μεταβλητής στη C 2/2

- κάθε μεταβλητή πρέπει να δηλωθεί πριν χρησιμοποιηθεί
- Παραδείγματα δηλώσεων
 - `int num1;`** // δηλώνει ακέραια μεταβλητή
 - `double result;`** // δηλώνει μεταβλητή κινητής υποδιαστολής διπλής ακρίβειας
 - `const double pi = 3.1416;`** // όπως η `result` παραπάνω αλλά σταθερά
- δήλωση με ανάθεση αρχικής τιμής
 - `int num1 = 12;`**
 - `double temp = 24.35;`**



Κανόνες δήλωσης μεταβλητών

- μέχρι 31 χαρακτήρες (ANSI C)
 - lower case, upper case, digits, underscore
 - πρώτος χαρακτήρας letter or underscore
- δήλωσε κάθε μεταβλητή πριν τη χρήση της
 - αποφεύγονται δύσκολα στην ανεύρεση λάθη
- βάλε όλες τις μεταβλητές σε μια θέση
- χρησιμοποίησε εκφραστικά ονόματα (readability)
- Προτίμησε τις τοπικές (**local**) αντί για τις γενικές μεταβλητές (**global**), εκτός και αν υπάρχει λόγος για αυτό.

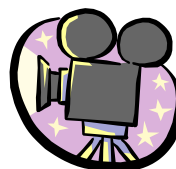
Μεταβλητή που ορίζεται έξω από συνάρτηση συνάρτηση

Μεταβλητή που ορίζεται μέσα στο σώμα συνάρτησης



Τύποι Δεδομένων

- Ένας τύπος δεδομένων **προσδιορίζει**
 - ένα σύνολο από αντικείμενα με κοινά χαρακτηριστικά τα οποία συνήθως εκφράζονται από ένα σύνολο πράξεων πάνω στα αντικείμενα.



Βασικοί τύποι δεδομένων

- **Αριθμοί**
 - βασίζονται στην υλοποίηση της αριθμητικής της μηχανής
- **Boolean**
 - με τιμές true, false
- **Χαρακτήρας**
- **Αλφαριθμητικό (String)**



Τύποι Δεδομένων

- Οι σύγχρονες προστακτικές γλώσσες
 - **περιλαμβάνουν** ορισμένους **ενσωματωμένους τύπους**,
 - **int, char, float, double ...**
 - **προσφέρουν** ένα **μηχανισμό για τον ορισμό νέων τύπων**
 - `struct student { ... };`
 - `struct rectangle { ... };`
- Η δυνατότητα ορισμού νέων τύπων αυξάνει:
 - όχι μόνο την **αναγνωσιμότητα** του προγράμματος, αλλά και
 - την **αξιοπιστία** του.



Τύποι δεδομένων στη C

- **βασικοί**
 - **char** (χαρακτήρας)
 - **int** (ακέραιος)
 - **float** (κινητής υποδιαστολής απλής ακρίβειας)
 - **double** (κινητής υποδιαστολής διπλής ακρίβειας)
- **πίνακας** (array) **[]**
- **δομή struct** - ένωση union
- **δείκτης** *****
- **απαριθμητικός τύπος** enum



Μονάδες μνήμης

- **Bit**

0	1
---	---

 - αποτελεί το βασικό δομικό στοιχείο της μνήμης του υπολογιστή. Μπορεί να έχει μια εκ των δύο τιμών: 0 ή 1
- **Byte**

0	1	0	1	1	0	1	1
---	---	---	---	---	---	---	---

ANSI requirement
 - αποτελεί τη συνηθισμένη μονάδα αποθήκευσης. Αποτελείται από 8bits
- **Word**

0	1	0	1	1	0	1	1
---	---	---	---	---	---	---	---

1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

 - αποτελεί την πληροφορία που μπορεί να διακινηθεί (I/O) σε ένα κύκλο μηχανής.
 - ταυτίζεται με το μέγεθος του data bus του επεξεργαστή
 - 8-bit, 16-bit, 32-bit, 64-bit microcomputers.



Ακέραιοι (Integers) - INT

- ANSI requirements
 - πρέπει να είναι τουλάχιστο 16 bits
 - πρέπει να είναι στο μέγεθος της λέξης (word) του υπολογιστή
- short int <=> short από -32.767 έως 32.767
- long int <=> long
- unsigned int
 - unsigned short int 0 - 65.535
- signed int <=> int (signed is default)



Εκτύπωση ακεραίων 1/2

```
int ten = 10;  
printf("%d minus %d is %d\n",ten,2,ten-2);
```

10 minus 2 is 8

■ Δεκαδικοί, Οκταδικοί, Δεκαεξαδικοί

- Προσδιοριστές %d, %o, %x

```
int num = 100;  
printf("dec=%d; , octal=%o; , hex=%x\n", num,  
num, num);
```

dec=100; octal=144; hex=64



Εκτύπωση ακεραίων 2/2

■ long, short, unsigned

- %u
- %ld, %lu l -> long
- %lx
- %lo
- %hd, %hu h -> short
- %ho
- %hx

χρησιμοποιώντας λάθος σύμβολο έχουμε απρόβλεπτα αποτελέσματα



Χαρακτήρας (char) 1/2

- κάθε χαρακτήρας έχει ένα μοναδικό αριθμητικό κωδικό
- **ASCII** code
American Standard Code for Information Interchange
- **EBCDIC** code (IBM large computers)
Extended Binary-Coded Decimal Interchange Code
- `char ch = 'A'; char ch = 65;`

```
char ch;  
printf("Enter a character:")  
scanf("%c", &ch);  
printf("Its numeric code value is : %d\n", ch);
```

Γράψτε ένα πρόγραμμα για την εμφάνιση των χαρακτήρων a-z και των αντίστοιχων ascii κωδικών τους



Χαρακτήρας (char) 2/2

```
char ch = 'A';          /* ο.k */  
ch = "A";               /* λάθος */  
ch = '\65';             /* Not portable */  
ch = 65;                /* Not portable */  
ch = A;                 /* η A θεωρείται μεταβλητή */
```

ASCII codes

A -> 65
Z -> 90
a -> 97
z -> 122

- Οι χαρακτήρες αντιμετωπίζονται ως μικροί αριθμοί

```
int j;  
j = 'A' + 'B';  
printf("%c\n", j);
```

Αποθήκευση-Ανάκληση χαρακτήρα

Δεσμεύει χώρο για χαρακτήρα

Αποδίδει τιμή στην μεταβλητή

```
int ch;
```

ch = 'A';

0	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---

printf("%c",ch);

printf("%d",ch);

Τυπώνει την τιμή της μεταβλητής ως χαρακτήρα

Τυπώνει την τιμή της μεταβλητής ως ακέραιο

A

65

© Κλεάνθης Θραμπουλίδης

Μεταβλητή – Τύποι δεδομένων

19

Μη εκτυπούμενοι χαρακτήρες

Escape character sequences

- \a alert
- \b backspace
- \f form feed
- \n newline
- \r carriage return
- \t horizontal tab
- \v vertical tab
- \\ backslash
- \' single quote

```
printf("Write, \"a \\ is a  
backslash. \\\"n");
```

Write, "a \ is a backslash."

© Κλεάνθης Θραμπουλίδης

Μεταβλητή – Τύποι δεδομένων

20



Float - Double

- float** -> 32 bits (8 για εκθέτη και πρόσημο)
(24 για βάση)
- double** -> συνήθως 64 bits
μερικά συστήματα χρησιμοποιούν τα επιπλέον 32 bits για βάση αυξάνοντας την ακρίβεια

- Εκτύπωση
 - %f για δεκαδικό
 - %e για εκθετικό

```
float plank = 6.63e-34;
```



Αλφαριθμητικό ως πίνακας χαρακτήρων

Η C **δεν έχει τύπο για το αλφαριθμητικό**, το αντιμετωπίζει ως πίνακα χαρακτήρων.

```
char name[20] = {'h','e','l','l','o','\0'};
```

```
char name[] = {'h','e','l','l','o','\0'};  
char name[] = "hello";
```

Δεσμεύουν χώρο για τον αριθμό των χαρακτήρων συν ένα για τον χαρακτήρα '\0'. Στην περίπτωση μας 6 (5+1).

Δες άσκηση 2 Κεφάλαιο 11
Διαχείριση Αλφαριθμητικού

Αναφορά στοιχείου	Μνήμη	Διεύθυνση
name[0]	h	F000
name[1]	e	F001
name[2]	l	F002
	l	F003
	o	F004
	\0	F005
		F006
...		
name[19]		F019

Το name είναι η διεύθυνση στο πρώτο στοιχείο του πίνακα



Σταθερές (const) 1/2

■ Int

- 245 αποθηκεύεται ως int
- 100.000 αποθηκεύεται ως long
- για ελεγχόμενη αποθήκευση χρησιμοποιώ l ή L

■ Αλφαριθμητική σταθερά

- ακολουθία από κανένα ή περισσότερους χαρακτήρες κλεισμένους σε διπλά εισαγωγικά.
“” “ ” “Hello world”



Σταθερές 2/2

- Οι compilers by default θεωρούν τις σταθερές κινητής υποδιαστολής, ως διπλής ακρίβειας (double)
 - + μεγαλύτερη ακρίβεια,
 - μείωση ταχύτητας εκτέλεσης
- **χρησιμοποίησε :**
 - 9.11E9**F** για float
 - 4.32e4**L** για long double



Σταθερά απαρίθμησης (enum) 1/3

■ επιτρέπει

- τη δήλωση συμβολικών ονομάτων για τη παράσταση ακέραιων σταθερών

■ δημιουργεί

- ένα **νέο τύπο** στον οποίο μπορείς να ορίσεις τις τιμές που μπορεί να πάρει μεταβλητή του τύπου αυτού.

■ Παραδείγματα ορισμού τύπου enum

```
5 enum dayOfWeek {  
6     SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY  
7 };  
8 enum labGroup {  
9     GROUP1=1, GROUP2, GROUP3  
10};
```

■ Παραδείγματα ορισμού μεταβλητών τύπου enum

```
enum dayOfWeek currentDay;  
enum labGroup yourLabGroup;
```

Δες αναλυτικά για απαριθμητικό τύπο Ενότητα 3.6 Απαριθμητικός Τύπος (κεφάλαιο 3)

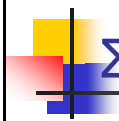
Χρήση του απαριθμητικού τύπου στην δόμηση της main με menu [Παράδειγμα 6-7 \(κεφάλαιο 6\)](#)



Σταθερά απαρίθμησης (enum) 2/3

```
5 enum dayOfWeek {  
6     SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY  
7 };  
8 enum labGroup {  
9     GROUP1=1, GROUP2, GROUP3  
10};  
11  
12 int main(int argc, char *argv[]) {  
13     enum dayOfWeek currentDay;  
14     enum labGroup yourLabGroup;  
15     char *messagePtr;  
16  
17     currentDay = WEDNESDAY; // or currentDay=3  
18     yourLabGroup = GROUP1; // or yourLabGroup = 1  
19     if(currentDay == WEDNESDAY && yourLabGroup==GROUP1)  
20         messagePtr = "Lab time is 14:00-16:00";  
21     else if(currentDay == THURSDAY && yourLabGroup==GROUP2)  
22         messagePtr = "Lab time is 15:00-17:00";  
23     else if(currentDay == FRIDAY && yourLabGroup==GROUP3)  
24         messagePtr = "Lab time is 17:00-19:00";  
25     printf("%s\n", messagePtr);  
26     return 0;  
27 }
```

Μεταβλητές του τύπου enum μπορούν να πάρουν μόνο τις τιμές που ορίζει ο τύπος enum.



Σταθερά απαρίθμησης (enum) 3/3

Ορισμός τύπου enum

```
enum spectrum {red, orange, green, blue};
```

```
enum spectrum color;
```

Ορισμός μεταβλητής τύπου enum

Ορισμός μεταβλητής τύπου enum χωρίς απόδοση ονόματος στον τύπο

```
enum {bright, medium, dark} intensity;
```

```
color = blue;      // Αποδεκτό (από τον compiler)
color = bright;    // Μη αποδεκτό-type conflict
color = 1;         // Μη αποδεκτό-type conflict
color=red + green; // Αποδεκτό αλλά προβληματικό
```

```
currentDay = MONDAY + WEDNESDAY; // Αποδεκτό αλλά προβληματικό
```