

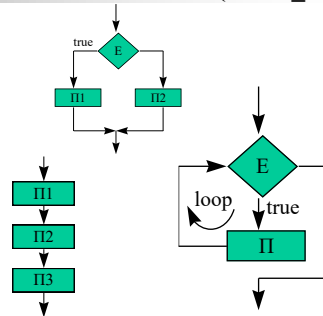


Προτάσεις ελέγχου ροής

Εισαγωγή στον Προγραμματισμό

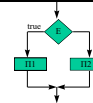
(CEID_NY13)

Control flow Statements



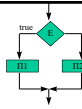
Kleanthis Thramboulidis
Prof. of Software and System Engineering
University of Patras
<https://sites.google.com/site/thramboulidiskleanthis/>

Βασικές Κατασκευές Γλώσσας Προγραμματισμού



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

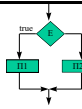
Βασικές δομές ελέγχου ροής 1/3



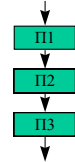
- Βασικές δομές προγραμματισμού (ανεξάρτητες γλώσσας) που χρησιμοποιούνται για την ανάπτυξη αλγορίθμων
 1. **Sequence** structure (ακολουθία)
 2. **Selection** structure (επιλογή)
 3. **Repetition/Loop** structure (επανάληψη)

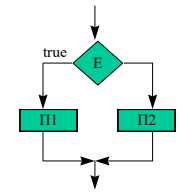
© Κλεάνθης Θραμπουλίδης Control flow Statements 3

Βασικές δομές ελέγχου ροής 2/3



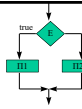
- **Sequence structure**
 - μία πρόταση ή μια ομάδα προτάσεων εκτελείτε η μία μετά την άλλη ακολουθιακά
- **Selection structure**
 - Διαφορετικές προτάσεις ή ομάδες προτάσεων εκτελούνται ανάλογα με το αν η τιμή μιας συνθήκης είναι αληθής ή ψευδής



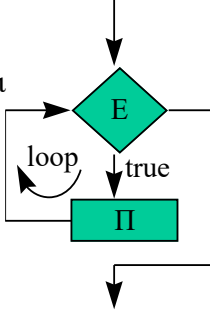


© Κλεάνθης Θραμπουλίδης Control flow Statements 4

Βασικές δομές ελέγχου ροής 3/3

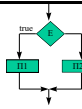


- **Repetition**
 - μία πρόταση ή ένα σύνολο προτάσεων εκτελείται επανειλημμένα, διακρίνεται σε:
 - **conditional loop**
 - η επανάληψη εκτελείται όσο μια συνθήκη είναι αληθής
 - **counting loop**
 - η πρόταση ή το σύνολο προτάσεων εκτελείται για προκαθορισμένο αριθμό επαναλήψεων



© Κλεάνθης Θραμπουλίδης Control flow Statements 5


Δομημένος Προγραμματισμός



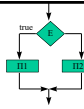
Η δομή του πηγαίου κώδικα θα πρέπει να
 μας βοηθά να κατανοήσουμε
 τι κάνει το πρόγραμμα

Ο αναγνώστης του δομημένου πηγαίου κώδικα μπορεί εύκολα να κατανοήσει τι συμβαίνει όταν το πρόγραμμα εκτελείται.

© Κλεάνθης Θραμπουλίδης Control flow Statements 6



Αναγνωσιμότητα Κώδικα



```

int max(int a, int b, int c)
{
if (a>b)
if (a>c)
return a;
else
return c;
else
if (b>c)
return b;
else
return c;
}

```

```


int max(int a, int b, int c)
{
if (a>b)
    if (a>c)
        return a;
    else
        return c;
else
    if (b>c)
        return b;
    else
        return c;
}

```

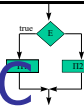
© Κλεάνθης Θραμπουλίδης

Control flow Statements

7



Προτάσεις ελέγχου ροής στη C



- Επανάληψης (looping)
 - for
 - while
 - do while
- διακλάδωσης υπό συνθήκη (conditional branching)
 - if
 - if else
 - switch
- μεταπήδησης (jumps)
 - break
 - continue
 - goto

© Κλεάνθης Θραμπουλίδης

Control flow Statements

8

Πρόταση for

**for (έκφραση1 ; έκφραση2 ; έκφραση3)
πρόταση**

**for (initialize ; test ; update)
πρόταση**

© Κλεάνθης Θραμπουλίδης Control flow Statements 9

Πρόταση for

```

for (count=1; count<max_count; count++ ) {
    printf("count is %d\n",count);
    < άλλες προτάσεις >
}
< επόμενη πρόταση >

```

© Κλεάνθης Θραμπουλίδης Control flow Statements 10

Πρόταση while 1/2

entry condition loop

**while (έκφραση)
πρόταση**

© Κλεάνθης Θραμπουλίδης Control flow Statements 11

Πρόταση while 2/2


```
while (count<limit) {
    count++;
    printf("count is %d\n",count);
}
< επόμενη πρόταση>
```

while

while

Εαν limit = 12 και count >= 12 το σώμα του loop δεν εκτελείται

© Κλεάνθης Θραμπουλίδης Control flow Statements 12



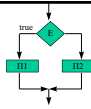
Πρόταση do while 1/2

exit condition loop


do

πρόταση

while (έκφραση);



© Κλεάνθης Θραμπουλίδης Control flow Statements 13

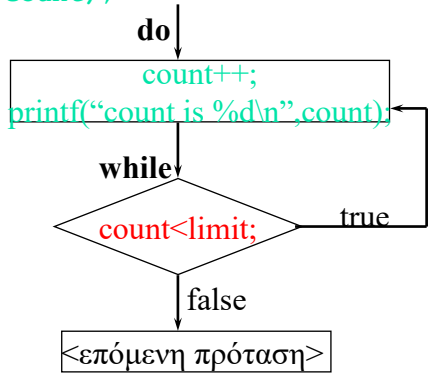
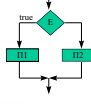


Πρόταση do while 1/2

```
do {
    count++;
    printf("count is %d\n", count);
} while (count < limit)
```

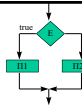
< επόμενη πρόταση >

Εαν limit = 12 και count >= 12
το σώμα του loop εκτελείται μια
φορά

© Κλεάνθης Θραμπουλίδης Control flow Statements 14

Επιλογή loop



- Αποφάσισε αν χρειάζεσαι entry-condition ή exit-condition.
- Προτίμησε γενικά το entry-condition

```

initialize;
while(test) {
    statement
    update;
}
    
```

↔

```

for(initialize;test;update)
    statement
    
```

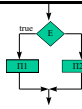
↔

```

for (;test;)
    while(test)
    
```

© Κλεάνθης Θραμπουλίδης Control flow Statements 15

Πρόταση if



```


if ( έκφραση )
    πρόταση1
else
    πρόταση2
    
```

```

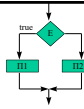
if (light == red)
    stop();
else
    go();
    
```

Άσκηση
 Δώστε τη δήλωση και τον ορισμό συνάρτησης για τον υπολογισμό του μέγιστου μεταξύ δύο ακέραιων αριθμών.

© Κλεάνθης Θραμπουλίδης Control flow Statements 16



Δήλωση της max




- Δήλωση (function prototype)


```
int max(int a, int b);
```

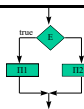
 ή εναλλακτικά


```
int max(int, int);
```

© Κλεάνθης Θραμπουλίδης Control flow Statements 17



Ορισμός της max



```
int max(int a, int b){
  int max;

  if(a>b)
    max = a;
  else
    max= b;
  return max;
}
```

```
int max(int a, int b){
  if(a>b)
    return a;
  else
    return b;
}
```

```
int max(int a, int b){
  return a>b ? a : b;
}
```

έκφραση 1

έκφραση 2

έκφραση 3

© Κλεάνθης Θραμπουλίδης Control flow Statements 18

Ενσωματωμένα (nested) if

```

if(kwh <= BREAK1)
    bill = RATE1 * kwh;
else
    if (kwh <= BREAK2)
        bill = BASE1 + RATE2 * (kwh - BREAK1);
    else
        bill = BASE2 + RATE3 * (kwh - BREAK2);
  
```

ένα else συσχετίζεται πάντα με το πλησιέστερο προηγούμενο if

© Κλεάνθης Θραμπουλίδης Control flow Statements 20

Πρόταση switch 1/2


```

switch (έκφραση) {
    case σταθερά έκφραση 1 :
        πρόταση1
    case σταθερά έκφραση 2:
        πρόταση2
    :
    :
    default :
        πρόταση3
}
  
```

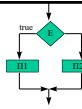
int ή char

το default a) εκτελείται όταν καμία από τις case δεν ικανοποιείται, b) δεν είναι απαραίτητα η τελευταία ετικέτα

© Κλεάνθης Θραμπουλίδης Control flow Statements 21




Πρόταση switch 2/2

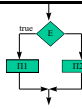


- κάθε case πρέπει να έχει μια int ή char σταθερά ή μια σταθερά έκφραση
- δύο case δεν μπορούν να έχουν την ίδια τιμή
- case και default μπορούν να μπουν με οποιαδήποτε σειρά
- η ροή συνεχίζει από το επιλεγθέν case μέχρι την εμφάνιση μιας πρότασης **break**, **return**, **goto** ή το τέλος του switch
- το break μετά τη τελευταία ετικέτα αποτελεί καλή τακτική αν και δεν είναι απαραίτητο

© Κλεάνθης Θραμπουλίδης Control flow Statements 22



Άσκηση με τη switch



Να γραφεί πρόγραμμα που να δέχεται ως είσοδο κείμενο, να απαριθμεί

- τις εμφανίσεις των ψηφίων 0-9,
- τα λευκά διαστήματα και
- τους υπόλοιπους χαρακτήρες

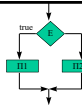
και στη συνέχεια να τυπώνει τα αποτελέσματα.

Δες Άσκηση 3 Κεφάλαιο 11
Τέσσερις Πράξεις

Δες άσκηση 4 Κεφάλαιο 11
Για ορισμό κύριας ροής προγράμματος

© Κλεάνθης Θραμπουλίδης Control flow Statements 23

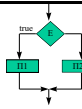
Λεκτική περιγραφή



Για κάθε χαρακτήρα του κειμένου
έλεγξε τον τύπο του χαρακτήρα
αν είναι ένας από τους ‘ ‘, \t’ ή \n’ αύξησε τον
απαριθμητή των διαστημάτων,
αν είναι ψηφίο αύξησε τον απαριθμητή που αντιστοιχεί
στο ψηφίο,
σε κάθε άλλη περίπτωση αύξησε τον απαριθμητή των
υπόλοιπων χαρακτήρων
Τύπωσε τους απαριθμητές

© Κλεάνθης Θραμπουλίδης Control flow Statements 24

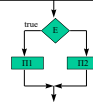
Προτάσεις διακλάδωσης χωρίς συνθήκη



- Διαχείριση ειδικών περιπτώσεων σε προτάσεις επανάληψης
 - **break**
 - **continue**
- Μεταφορά ελέγχου από συνάρτηση στην καλούσα συνάρτηση
 - **return**
- Ρητή διακλάδωση
 - **goto <ετικέτα>**

© Κλεάνθης Θραμπουλίδης Control flow Statements 25

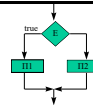
Διαχείριση ειδικών περιπτώσεων σε προτάσεις επανάληψης



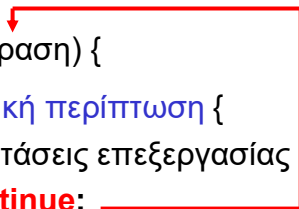
```
while (έκφραση) {  
    if ειδική περίπτωση {  
        προτάσεις επεξεργασίας ειδικής περίπτωσης  
        break;  
    }  
    προτάσεις επεξεργασίας κανονικών περιπτώσεων  
}
```



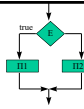
Διαχείριση ειδικών περιπτώσεων σε προτάσεις επανάληψης



```
while (έκφραση) {  
    if κανονική περίπτωση {  
        προτάσεις επεξεργασίας κανονικής περίπτωσης  
        continue;  
    }  
    προτάσεις επεξεργασίας ειδικών περιπτώσεων  
}
```



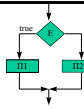
Πρόταση break



- προκαλεί την έξοδο μόνο από το πιο εσωτερικό βρόχο
- καταστρέφει τη δόμηση του κώδικα
- υπάρχει πάντα τρόπος να γραφεί κώδικας χωρίς τη χρήση της


© Κλεάνθης Θραμπουλίδης Control flow Statements 28

Παράδειγμα χρήσης break

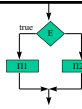


```
for( i = 0 ; i < MAX ; i++) {  
    if (num[i] < 0)      /* end loop */  
        break;  
    : /* process positive elements */  
}  
printf(.....);
```

© Κλεάνθης Θραμπουλίδης Control flow Statements 29



Αποφυγή του break



```

while (( ch = getchar() ) != '\n' ) {
    if (ch == '\t')
        break;
    putchar(ch);
}


```

```

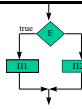
while ((ch=getchar()) != '\n' && ch != '\t')
    putchar(ch);

```

© Κλεάνθης Θραμπουλίδης Control flow Statements 30



Πρόταση continue



- προκαλεί την έναρξη της επόμενης επανάληψης **for**, **while** ή **do**
- επηρεάζει μόνο το πιο εσωτερικό βρόχο

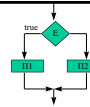
```

for( i = 0 ; i < MAX ; i++) {
    if (num[i] < 0) /* skip negative elements */
        continue;
    : /* process positive elements */
}
printf(.....);

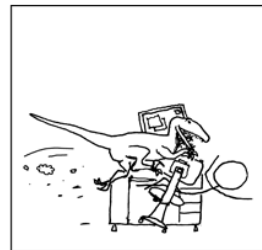
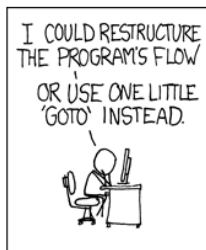
```

© Κλεάνθης Θραμπουλίδης Control flow Statements 31

Πρόταση goto



- καταστρέφει την δόμηση του κώδικα
- είναι πάντα εύκολο να γράφεις κώδικα χωρίς τη χρήση της



Source: <http://xkcd.com/292/>