

Πρόγραμμα Μεταπτυχιακών Σπουδών  
Πληροφορική και Υπολογιστική Βιοϊατρική

# Θέματα Προγραμματισμού Η/Υ

## Ενότητα 7:

Θεματική Ενότητα: Δομές επανάληψης

# ΘΕΜΑΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Η/Υ

Θεματική Ενότητα 7

Δομές επανάληψης

Πληροφορική και Υπολογιστική Βιοϊατρική  
Α. Κακαρούντας, Γ. Σπαθούλας, Π. Κοντού

# Δομές επανάληψης

- Μία δομή επανάληψης κάνει μία εντολή ή ένα σύνολο εντολών να εκτελούνται επαναλαμβανόμενα
  - Π.χ. θέλετε το παραπάνω πρόγραμμα ελέγχου κατανάλωσης αλκοόλ να εκτελείται για 20 άτομα (ή όσα καθορίσει ο χρήστης) χωρίς επανεκκίνηση.
  - Επανάληψη του κώδικα είναι:
    - ▣ Χρονοβόρα
    - ▣ Δεσμεύει πολύ χώρο στη μνήμη
    - ▣ Απαιτεί πολλαπλή αποσφαλμάτωση
    - ▣ Λειτουργεί για συγκεκριμένο αριθμό επαναλήψεων
  - Π.χ. θέλετε το παραπάνω πρόγραμμα να εκτελείται για πολλαπλά άτομα μέχρι ο χρήστης ή κάποιο κριτήριο να το τερματίσει.
- a. βρόχος ελεγχόμενος με μετρητή (count-controlled loop)**  
επαναλαμβάνεται ένα συγκεκριμένο αριθμό φορές (εντολή **for**)
- b. βρόχος ελεγχόμενος με συνθήκη (condition-controlled loop)** χρησιμοποιεί μία συνθήκη αληθούς/ψευδούς για να ελέγξει το πόσες φορές θα επαναληφθεί (εντολή **while**)

# for

- Ένας βρόχος ελεγχόμενος με μετρητή, εκτελείται για ένα συγκεκριμένο αριθμό επαναλήψεων.

for μεταβλητή\_στόχος in [τιμή1, τιμή2, ...]:

εντολή  
εντολή  
...

μπλοκ των προς  
επανάληψη  
εντολών

αλληλουχία τιμών  
(λίστα)

# Πρόγραμμα απλού for βρόχου με χρήση λίστας αριθμών.  
print('Θα εμφανίσω τους αριθμούς 1 έως 3.')

for num in [1, 2, 3]:

print(num)

Θα εμφανίσω τους αριθμούς 1 έως 3

1  
2  
3

# for - range

- Η συνάρτηση `range` απλουστεύει τη διαδικασία σύνταξης των `for` βρόχων (ελεγχόμενων με μετρητή).
- Δημιουργεί έναν τύπο αντικειμένου γνωστό ως *λίστα τιμών βρόχου* (iterable), παραπλήσιο της λίστας.
- Π.χ.

```
for num in range(4):  
    print(num)           ==           for num in [0, 1, 2, 3]:  
                                print(num)
```

Ξεκινά από το 0 έως το 4 (μη συμπεριλαμβανομένου)

```
for num in range(1, 4):  
    print(num)
```

1  
2  
3

```
for num in range(1, 8, 2):  
    print(num)
```

1  
3  
5  
7

# Παράδειγμα for

```
for rep in range(20) :
    age = int( input('Δώσε την ηλικία σου: ')
    goniki_egrisi = input('Έχεις γονική έγκριση? Δώσε N για ναι ή O για όχι: ')
    if goniki_egrisi == 'N' :
        goniki_egrisi = True          # Boolean μεταβλητή
    else :
        goniki_egrisi = False

    if age >=18 :
        print('Μπορείς να αγοράσεις ποτό')
    else :
        if goniki_egrisi == True :
            print('Μπορείς να αγοράσεις μικρό ποτό')
        else :
            print('Μπορείς να αγοράσεις αναψυκτικό')
```

# Χρήση της Μεταβλητής Στόχου μέσα στη for

- Η μεταβλητή στόχος συχνά χρησιμοποιείται σε κάποιον υπολογισμό ή κάποια άλλη διεργασία μέσα στο σώμα του βρόχου.
- Π.χ. γράψετε ένα πρόγραμμα που να εμφανίζει τους αριθμούς 1 έως 4 και τα τετράγωνά τους, σε μορφή πίνακα

```
print('Αριθμός\tΤετράγωνο')
```

```
print('-----')
```

```
for number in range(1, 5):
```

```
    square = number**2
```

```
    print(number, '\t', square)
```

Αριθμός  
Τετράγωνο

```
-----  
1          1  
2          4  
3          9  
4         16
```

# Έλεγχος επαναλήψεων for-βρόχου από το χρήστη

- Συχνά πρέπει να δίνεται στο χρήστη η δυνατότητα να ελέγξει τον αριθμό επαναλήψεων ενός βρόχου.
- Π.χ. τροποποίηση του προηγούμενου παραδείγματος για υπολογισμό των τετραγώνων μέχρι του αριθμού που θα δώσει ο χρήστης.

```
end = int(input('Μέχρι ποιον αριθμό; '))
print('Αριθμός\tΤετράγωνο')
print('-----')
for number in range(1, end+1) :
    square = number**2
    print(number, '\t', square)
```



# Υπολογισμός Τρέχοντος Αθροίσματος

- *Τρέχον άθροισμα* ονομάζεται το άθροισμα αριθμών το οποίο αυξάνεται (συσσωρεύεται) με κάθε επανάληψη ενός βρόχου.
- Η μεταβλητή που χρησιμοποιείται για την αποθήκευση του τρέχοντος αθροίσματος ονομάζεται *αθροιστής ή συσσωρευτής*.

```
total = 0          # Αρχικοποίηση της μεταβλητής του αθροιστή.
```

```
max = 5           # Το πλήθος των προς άθροιση αριθμών
```

```
# Εισαγωγή των αριθμών και πρόσθεσή τους.
```

```
for counter in range(max) :
```

```
    number = int( input('Δώστε έναν ακέραιο αριθμό: ') )
```

```
    total = total + number    # γράφεται και total += number
```

```
# Εμφάνιση του αθροίσματος των αριθμών.
```

```
print('Το άθροισμα είναι', total)
```

- **+=, -=, \*=, /=** Τελεστές Επαυξημένης Εκχώρησης

Για άθροιση μόνο των τιμών που ικανοποιούν κάποιο κριτήριο, η έκφραση συσσώρευσης πρέπει να περικλείεται σε κάποια if εντολή

# for-Ένθετοι Βρόχοι (παράδειγμα)

Ανάγκη για ύπαρξη βρόχου επανάληψης μέσα σε βρόχο. Π.χ.

- Εκτύπωση δευτερολέπτων

```
for seconds in range(60):
```

```
    print(seconds)
```

- Εκτύπωση λεπτών και δευτερολέπτων

```
for minutes in range(60):
```

```
    for seconds in range(60):
```

```
        print(minutes, ':', seconds)
```

- Εκτύπωση ωρών, λεπτών και δευτερολέπτων

```
for hours in range(24):
```

```
    for minutes in range(60):
```

```
        for seconds in range(60):
```

```
            print(hours, ':', minutes, ':', seconds)
```

# while

- Ένας βρόχος ελεγχόμενος με συνθήκη εκτελείται επαναλαμβανόμενα όσο κάποια συνθήκη είναι αληθής.

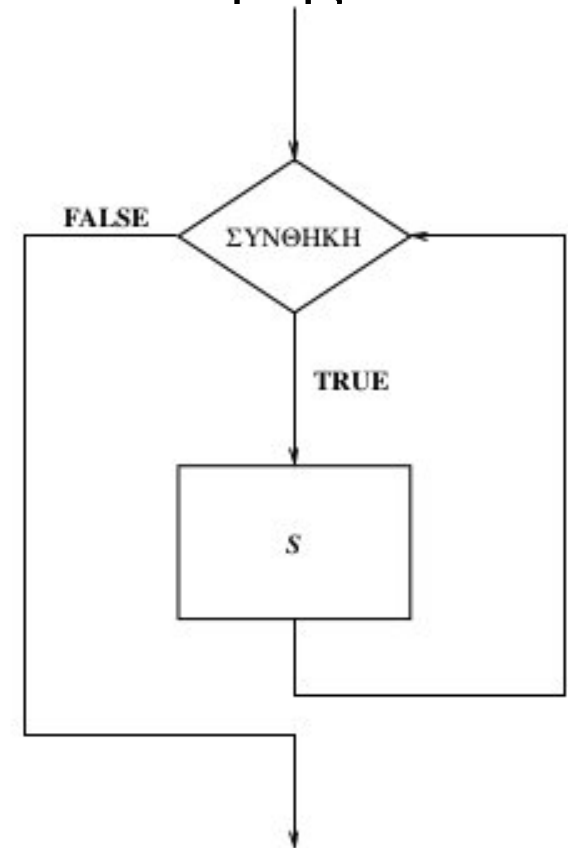
while συνθήκη :

εντολή

εντολή

...

- Ο Βρόχος while είναι Βρόχος Προελέγχου
  - ▣ Ελέγχει τη συνθήκη του πριν πραγματοποιήσει μία επανάληψη
  - ▣ Συνήθως είναι απαραίτητο να γίνουν κάποια βήματα πριν από το βρόχο για να εξασφαλίσουν ότι θα εκτελεστεί τουλάχιστον μία φορά
  - ▣ Αν η συνθήκη ικανοποιείται πάντα: *Ατέρμων Βρόχος (Infinite Loop)*
    - Άρα πρέπει εντός του βρόχου να επηρεάζεται το αποτέλεσμα της συνθήκης.



# Παράδειγμα

```
rep = 0
```

```
while rep < 20 :
```

```
    age = int( input('Δώσε την ηλικία σου: ')
```

```
    goniki_egrisi = input('Έχεις γονική έγκριση? Δώσε N για ναι ή O για όχι: ')
```

```
    if goniki_egrisi == 'N' :
```

```
        goniki_egrisi = True           # Boolean μεταβλητή
```

```
    else :
```

```
        goniki_egrisi = False
```

```
    if age >= 18 :
```

```
        print('Μπορείς να αγοράσεις ποτό')
```

```
    else :
```

```
        if goniki_egrisi == True :
```

```
            print('Μπορείς να αγοράσεις μικρό ποτό')
```

```
        else :
```

```
            print('Μπορείς να αγοράσεις αναψυκτικό')
```

```
    rep = rep + 1           # ή rep += 1
```

# Παράδειγμα `while` - Τιμή Τερματισμού

- *Τιμή τερματισμού* (*sentinel*) ονομάζεται μία ειδική τιμή που σηματοδοτεί το τέλος μίας αλληλουχίας τιμών.
- Όταν το πρόγραμμα διαβάσει την τιμή αυτή, ξέρει ότι έχει φτάσει στο τέλος της αλληλουχίας και έτσι ο βρόχος τερματίζεται.
- Π.χ. στο παράδειγμα επαναλαμβανόμενου ελέγχου κατανάλωσης αλκοόλ να τερματίζεται αν ο χρήστης δώσει ηλικία 0.
- Η τιμή τερματισμού θα πρέπει να είναι αρκετά χαρακτηριστική ώστε να μη συγχέεται με τις κανονικές τιμές της αλληλουχίας.

# Παράδειγμα



```
age = int( input('Δώσε την ηλικία σου (0 για έξοδο): ')  
while age != 0 :
```

```
    goniki_egrisi = input('Έχεις γονική έγκριση? Δώσε N για ναι ή O για όχι: ')  
    if goniki_egrisi == 'N' :
```

```
        goniki_egrisi = True                # Boolean μεταβλητή
```

```
    else :
```

```
        goniki_egrisi = False
```

```
    if age >=18 :
```

```
        print('Μπορείς να αγοράσεις ποτό')
```

```
    else :
```

```
        if goniki_egrisi == True :
```

```
            print('Μπορείς να αγοράσεις μικρό ποτό')
```

```
        else :
```

```
            print('Μπορείς να αγοράσεις αναψυκτικό')
```



```
age = int( input('Δώσε την ηλικία σου (0 για έξοδο): ')
```

# While - Βρόχος Επαλήθευσης Εισόδου

- Επαλήθευση εισόδου: η διαδικασία ελέγχου των δεδομένων που εισάγονται στο πρόγραμμα, για να διασφαλισθεί η εγκυρότητά τους πριν χρησιμοποιηθούν σε κάποιο υπολογισμό.
- Η επαλήθευση εισόδου πραγματοποιείται συνήθως με ένα βρόχο που επαναλαμβάνεται όσο κάποια μεταβλητή εισόδου αναφέρεται σε μη αποδεκτά δεδομένα.

# Παράδειγμα

```
age = int( input('Δώσε την ηλικία σου (0 για έξοδο): ') )
while age < 0 or age > 120 :
    print("Μάλλον έδωσες λάθος ηλικία, προσπάθησε ξανά!")
    age = int( input('Δώσε την ηλικία σου (0 για έξοδο): ') )
while age != 0 :
    goniki_egrisi = input('Έχεις γονική έγκριση? Δώσε N για ναι ή O για όχι: ')
    if goniki_egrisi == 'N' :
        goniki_egrisi = True                # Boolean μεταβλητή
    else :
        goniki_egrisi = False
    if age >= 18 :
        print('Μπορείς να αγοράσεις ποτό')
    else :
        if goniki_egrisi == True :
            print('Μπορείς να αγοράσεις μικρό ποτό')
        else :
            print('Μπορείς να αγοράσεις αναψυκτικό')
    age = int( input('Δώσε την ηλικία σου (0 για έξοδο): ') )
while age < 0 or age > 120 :
    print("Μάλλον έδωσες λάθος ηλικία, προσπάθησε ξανά!")
    age = int( input('Δώσε την ηλικία σου (0 για έξοδο): ') )
```



# Τεχνικές εισόδου πολλών δεδομένων - Ανακεφαλαίωση δομών επανάληψης

- Για γνωστό (ή καθορισμένο από το χρήστη πλήθος δεδομένων  $n$ )
  - ▣ Αρκεί μία απλή επαναληπτική διαδικασία, αφού το πλήθος των δεδομένων είναι γνωστό.
  - ▣ Μπορεί να γίνει είτε με `for` είτε με `while`, αλλά συνήθως προτιμάται το `for`.

```
n = int( input("Δώσε αριθμό επαναλήψεων") )      # είσοδος αριθμού επαναλήψεων
for i in range(n)                                # ή (0, n, 1) ή (1, n+1, 1)
    x = input("Δώσε το ", n, "ο δεδομένο", sep="") # είσοδος δεδομένου
```

- Για άγνωστο πλήθος δεδομένων
  - ▣ Είναι βασικό να ορισθεί κάποιο *flag* που να σηματοδοτεί τη λήξη της διαδικασίας εισαγωγής δεδομένων.
  - ▣ Συνήθως το *flag* τερματισμού είναι μια απίθανη τιμή (π.χ., για εισαγωγή έτους: η τιμή 0).
  - ▣ Εκτελείτε μόνο με `while`

```
x = input("Δώσε το δεδομένο")                # είσοδος 1ου δεδομένου
while x != flag
    x = input("Δώσε το δεδομένο")            # είσοδος δεδομένου
```

# ΘΕΜΑΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Η/Υ

Θεματική Ενότητα 7

Δομές επανάληψης

Πληροφορική και Υπολογιστική Βιοϊατρική  
Α. Κακαρούντας, Γ. Σπαθούλας, Π. Κοντού