



Πανεπιστήμιο Αιγαίου

# Εισαγωγή στον Προγραμματισμό Η/Υ

Ενότητα 3: Προγραμματιστικές δομές απόφασης  
και επανάληψης

Μιχάλης Βαΐτης  
Τμήμα Γεωγραφίας



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

# 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 Αιγαίου**» 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.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Σκοποί ενότητας

- Να γνωρίζετε τη χρήση και σύνταξη της εντολής `if` στη γλώσσα R.
- Να γνωρίζετε τη χρήση και σύνταξη της εντολής `for` στη γλώσσα R.
- Να γνωρίζετε τη χρήση και σύνταξη της εντολής `while` στη γλώσσα R.
- Να γνωρίζετε τη χρήση και σύνταξη της εντολής `repeat` στη γλώσσα R.
- Να μπορείτε να χρησιμοποιείτε αποτελεσματικά τη γλώσσα R για την επίλυση υπολογιστικών προβλημάτων.

# Δομές Επανάληψης

Χρησιμοποιούνται για την επανάληψη ομάδας εντολών (βρόχος – loop). Ο αριθμός των επαναλήψεων είτε είναι γνωστός κατά την έναρξη των επαναλήψεων, είτε είναι άγνωστος και καθορίζεται από τον επαναλαμβανόμενο έλεγχο μιας συνθήκης.

Στη R:

- for
- while
- repeat

# for

```
for (μεταβλητή in sequence)
    { εντολές }
```

## Σημεία προσοχής

- Η τιμή της μεταβλητής, καθώς και οι τιμές αρχής, τέλους και βήματος δεν πρέπει να μεταβάλλονται από τις εντολές του βρόχου

### Παράδειγμα 1

```
for (i in 1:10) {
    print (i)
}
# το i αυξάνεται κατά ένα
# και γίνονται δέκα επαναλήψεις
```

### Παράδειγμα 2

```
for(i in seq(0, 10, by = 2)){
    print(i)
}
# το i αυξάνεται κατά δύο (από το 0)
# και γίνονται έξι επαναλήψεις
```

# while

```
while ( συνθήκη )  
  { εντολές }
```

## Σημεία προσοχής

- Έλεγχος της συνθήκης στην αρχή του βρόχου
- Όσο η συνθήκη είναι αληθής, οι εντολές εκτελούνται επαναληπτικά.
- Αν στην αρχή η συνθήκη είναι ψευδής, οι εντολές δεν εκτελούνται καθόλου.

## Παράδειγμα

```
while ( (i <- as.numeric(readline())) != 0 )  
  { print (i) }
```

# repeat

```
repeat
{
    εντολές
    if ( συνθήκη ) break
    εντολές
}
```

## Σημεία προσοχής

- Έλεγχος της συνθήκης οπουδήποτε μέσα στο βρόχο (συνήθως ο έλεγχος γίνεται στο τέλος).
- Αν η συνθήκη είναι αληθής, σταματά η εκτέλεση του βρόχου.



# Υπολογισμός $n!$

Το  $n!$  (παραγοντικό) ενός θετικού ακεραίου αριθμού  $n$ , είναι το γινόμενο:

$$1 * 2 * 3 * \dots * n$$

Π.χ.

$$3! = 1 * 2 * 3 = 6$$

$$5! = 1 * 2 * 3 * 4 * 5 = 120$$

Γενικά ισχύει:  $n! = (n-1)! * n$

# Υπολογισμός $n!$

```
print('Δώσε θετικό ακέραιο')  
n <- as.numeric(readline())  
np <- 1  
for (i in 1:n)  
    np <- np * i  
print(np)
```

# Υπολογισμός δύναμης

Η ύψωση σε δύναμη είναι μαθηματική πράξη, που συμβολίζεται ως  $a^n$  και περιλαμβάνει δύο αριθμούς, την βάση  $a$  και τον εκθέτη  $n$ .

Αν το  $n$  είναι θετικός ακέραιος, η ύψωση σε δύναμη αντιστοιχεί σε επαναλαμβανόμενο πολλαπλασιασμό, με άλλα λόγια είναι το γινόμενο  $n$  παραγόντων  $a$ :

$$a^n = \underbrace{a \times \cdots \times a}_n,$$

Αν το  $n$  είναι μηδέν, το αποτέλεσμα είναι ο αριθμός 1.

Δηλ.  $a^0 = 1$

# Υπολογισμός δύναμης

Αν  $a$  και  $n$  είναι θετικοί ακέραιοι ή μηδέν, γράψτε πρόγραμμα στην R που να υπολογίζει το  $a^n$ , χωρίς τη χρήση του τελεστή  $^$ .

```
print('Δώσε βάση')
a <- as.numeric(readline())
print('Δώσε εκθέτη')
n <- as.numeric(readline())
power <- 1
if (n == 0) {
  print(power)
} else {
  for (i in 1:n) power <- power * a
  print(power)
}
```

# Υπολογισμός δύναμης

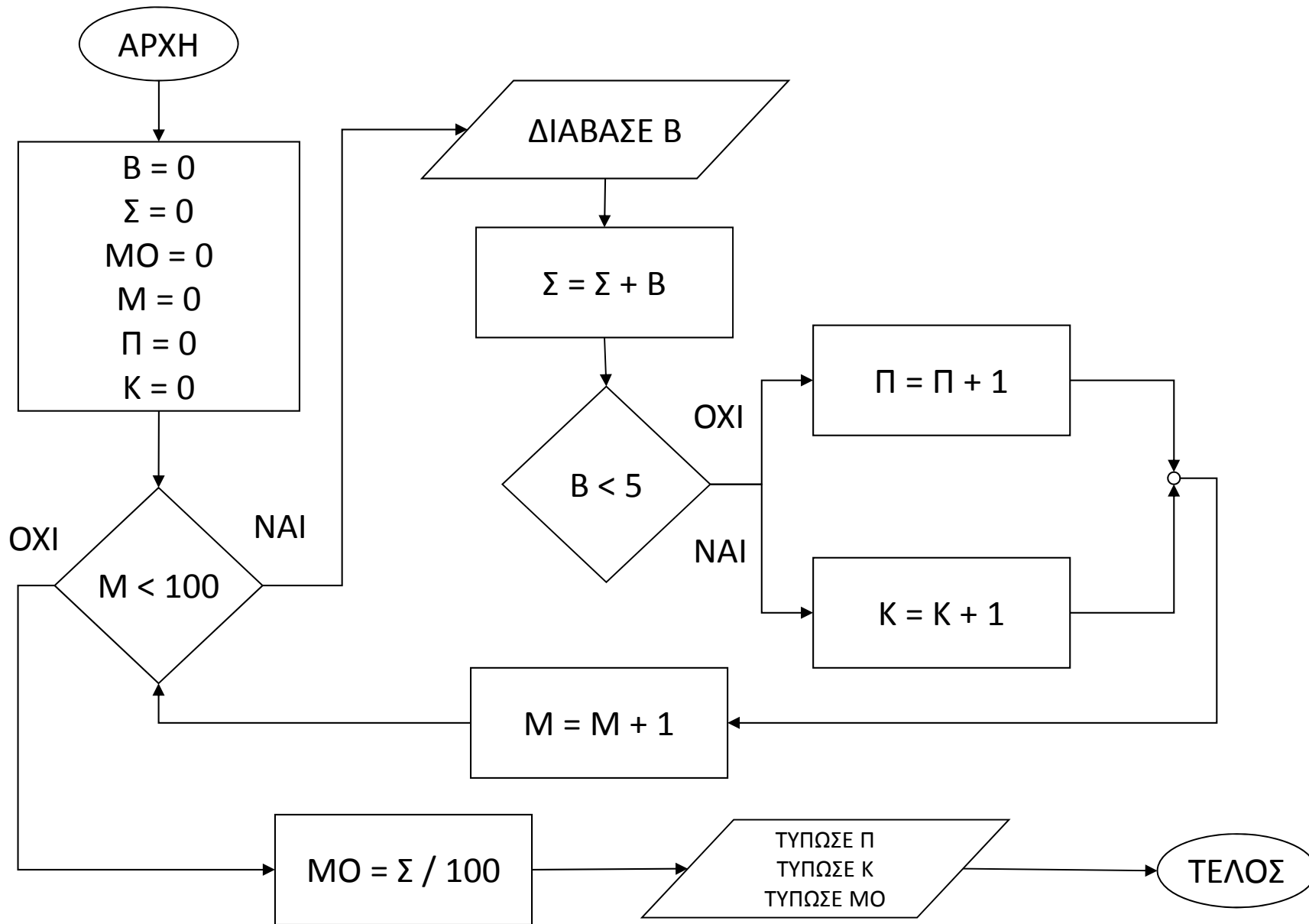
Εναλλακτικά, με μόνο μία εντολή print για το αποτέλεσμα. Δεν εκτελούνται εντολές στο if αν η συνθήκη είναι αληθής. Στο else υπάρχει μόνο μία εντολή – το for, το οποίο με τη σειρά του εκτελεί επαναληπτικά μόνο ένα υπολογισμό.

```
print('Δώσε βάση')
a <- as.numeric(readline())
print('Δώσε εκθέτη')
n <- as.numeric(readline())
power <- 1
if (n == 0) {
} else for (i in 1:n) power <- power * a
print(power)
```

# Στατιστικά εξετάσεων

Μετά την εξέταση 100 φοιτητών σε ένα μάθημα, ζητούνται:

- Ο αριθμός των φοιτητών που «πέρασαν» (δηλ. ο βαθμός τους ήταν στο  $[5, 10]$ ).
- Ο αριθμός των φοιτητών που «κόπηκαν» (δηλ. ο βαθμός τους ήταν στο  $[0, 5)$ ).
- Ο μέσος όρος της βαθμολογίας των 100 φοιτητών.



```
sum <- 0
metritis <- 0
pass <- 0
fail <- 0

repeat {
  vathmos <- as.numeric(readline())
  sum <- sum + vathmos
  if (vathmos < 5) fail <- fail + 1 else pass <- pass + 1
  metritis <- metritis + 1
  if (metritis = 100) break
}
mo <- sum/100

print(`Πέρασαν:`); print(pass)
print(`Κόπηκαν:`); print(fail)
print(`Μέσος όρος:`); print(mo)
```



# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

