



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά
Τεχνολογικού Τομέα



Εισαγωγή στην Πληροφορική & τον Προγραμματισμό

Ενότητα 7^η: Εντολές Επανάληψης

Ι. Ψαρομήλιγκος – Χ. Κυτάγιας
Τμήμα Διοίκησης Επιχειρήσεων



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 Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά Τεχνολογικού Τομέα**» έχει χρηματοδοτήσει μόνο την 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.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σκοποί ενότητας

Στην συγκεκριμένη ενότητα παρουσιάζονται οι εντολές επανάληψης:

- For... Next
- Do... Loop

της γλώσσας προγραμματισμού Visual Basic.NET.

Περιεχόμενα

- Η Δομή **For ... Next**
- Η Δομή **Do ... Loop**

Η Δομή For...Next

- Χρησιμοποιούμε τη δομή "**For...Next**" για να επαναλάβουμε ένα συγκεκριμένο αριθμό φορών την εκτέλεση ενός συνόλου εντολών (το σύνολο των εντολών το δηλώνουμε μεταξύ του **For** και του **Next**).
- Η συγκεκριμένη δομή χρησιμοποιεί μια μεταβλητή ελέγχου ως "**μετρητή**" των επαναλήψεων. Εμείς μπορούμε να καθορίσουμε την αρχική και την τελική τιμή του μετρητή καθώς και το "**βήμα**" της αύξησης ή μείωσης (το βήμα μπορεί να είναι και αρνητικό).

```
For Μετρητής = Αρχική_τιμή to Τελική_τιμή [Step βήμα]
    εκτελέσιμη εντολή 1
    εκτελέσιμη εντολή 2
    ...
    εκτελέσιμη εντολή K
Next [Μετρητής]
```

Παραδείγματα (1)

```
For X=1 to 100  
    Console.WriteLine(X)  
Next X
```

```
S = 0  
For X=1 to 100 Step 2  
    S = S + X  
Next X
```

```
S = 0  
For X=0 to 100  
    If X Mod 2 = 0 Then S = S + X^2  
Next X
```

```
Dim Index, S as Integer  
For Index=1 to 100  
    S=S+Index  
    If (S>10) Then Exit For  
Next Index
```

Παραδείγματα (2)

Στο παρακάτω τμήμα προγράμματος υπολογίζεται με τη βοήθεια της μεταβλητής **s** το άθροισμα:

$$1^2 + (0.9)^2 + (0.8)^2 + \dots + (0.1)^2 [=3.85]$$

```
Dim i as double, s as double
s = 0
For i=1 to 0.1 step -0.1
    s = s + i^2
Next i
```


Η Δομή Do...Loop (1)

- Όταν ο αριθμός των επαναλήψεων δεν είναι γνωστός και εξαρτάται από κάποια συνθήκη χρησιμοποιούμε τη δομή "**Do ... Loop**"
- Η δομή "**Do ... Loop**" παρουσιάζεται με τις παρακάτω μορφές:

--- 1η Μορφή ---

```
Do {While | Until} συνθήκη  
    [εντολές]  
Loop
```

--- 2η Μορφή ---

```
Do  
    [εντολές]  
Loop {While | Until} συνθήκη
```

Η Δομή Do...Loop (2)

--- 1η Μορφή ---

Do {While | Until} συνθήκη
[εντολές]
Loop

--- 2η Μορφή ---

Do
[εντολές]
Loop {While | Until} συνθήκη

- Στην 1η μορφή η συνθήκη εξετάζεται **πρώτα** ενώ στη 2η μορφή εξετάζεται στο **τέλος**. Συνεπώς, στην 1η μορφή η συνθήκη μπορεί να αποτρέψει εντελώς τις επαναλήψεις ενώ στη 2η μορφή θα έχουμε τουλάχιστον μια επανάληψη.
- Η συνθήκη συνοδεύεται με το **While** ή το **Until**. Αν έχουμε "**While συνθήκη**" σημαίνει "**Όσο** η συνθήκη=true **επανάλαβε**" ενώ αν έχουμε "**Until συνθήκη**" σημαίνει "**Μέχρι** η συνθήκη να γίνει ίση με true **επανάλαβε**". Αρα με While τερματίζουμε τις επαναλήψεις όταν γίνει "συνθήκη=false" ενώ με Until όταν γίνει "συνθήκη=true"

Παράδειγμα Do While...Loop

Στο επόμενο τμήμα προγράμματος εισάγεται ένας αριθμός στη μεταβλητή **number** τύπου **Integer**. Όσο η τιμή που πληκτρολογούμε δε βρίσκεται εντός του διαστήματος **[1,10]**, εμφανίζεται σχετικό μήνυμα και ξαναζητείται τιμή για τη **number**.

```
Dim number As Integer
Console.Write("Δώσε έναν αριθμό από το 1 έως το 10=")
number = Console.ReadLine()
Do While (number < 1 Or number > 10)
    Console.WriteLine("Ο αριθμός δεν είναι στο διάστημα [1,10]")
    Console.Write("Δώσε έναν αριθμό από το 1 έως το 10=")
    number = Console.ReadLine()
Loop
```

Παράδειγμα Do Until...Loop

- Το επόμενο τμήμα προγράμματος κάνει ακριβώς ότι και το προηγούμενο χρησιμοποιώντας "**Until συνθήκη**".
- Εισάγεται ένας αριθμός στη μεταβλητή **number** τύπου **Integer**. Οσο η τιμή που πληκτρολογούμε δε βρίσκεται εντός του διαστήματος **[1,10]**, εμφανίζεται σχετικό μήνυμα και ξαναζητείται τιμή για τη number.

```
Dim number As Integer
Console.Write("Δώσε έναν αριθμό από το 1 έως το 10=")
number = Console.ReadLine()
Do Until (number >= 1 And number <= 10)
    Console.WriteLine("Ο αριθμός δεν είναι στο διάστημα [1,10]")
    Console.Write("Δώσε έναν αριθμό από το 1 έως το 10=")
    number = Console.ReadLine()
Loop
```

Παράδειγμα Do...Loop While

- Το επόμενο τμήμα προγράμματος κάνει ακριβώς ό,τι και τα προηγούμενα δύο χρησιμοποιώντας "**While συνθήκη**" στο τέλος.
- Εισάγεται ένας αριθμός στη μεταβλητή **number** τύπου **Integer**. Όσο η τιμή που πληκτρολογούμε δε βρίσκεται εντός του διαστήματος **[1,10]**, εμφανίζεται σχετικό μήνυμα και ξαναζητείται τιμή για τη **number**.

```
Dim number As Integer
```

```
Do
```

```
    Console.WriteLine("Δώσε έναν αριθμό από το 1 έως το 10=")
```

```
    number = Console.ReadLine()
```

```
    If (number < 1 Or number > 10) Then
```

```
        Console.WriteLine("Ο αριθμός δεν είναι στο διάστημα [1,10]")
```

```
    End If
```

```
Loop While (number < 1 Or number > 10)
```

Παράδειγμα Do...Loop Until

- Το επόμενο τμήμα προγράμματος κάνει ακριβώς ότι και το προηγούμενο χρησιμοποιώντας "**Until συνθήκη**" στο τέλος.
- Εισάγεται ένας αριθμός στη μεταβλητή **number** τύπου **Integer**. Όσο η τιμή που πληκτρολογούμε δε βρίσκεται εντός του διαστήματος **[1,10]**, εμφανίζεται σχετικό μήνυμα και ξαναζητείται τιμή για τη **number**.

```
Dim number As Integer
```

```
Do
```

```
    Console.WriteLine("Δώσε έναν αριθμό από το 1 έως το 10=")
```

```
    number = Console.ReadLine()
```

```
    If (number < 1 Or number > 10) Then
```

```
        Console.WriteLine("Ο αριθμός δεν είναι στο διάστημα [1,10]")
```

```
    End If
```

```
Loop Until (number >= 1 And number <= 10)
```

Ατέρμονες επαναλήψεις

- Είναι επαναλήψεις στις οποίες δεν υπάρχει τρόπος διαφυγής. Δείτε το παρακάτω τμήμα κώδικα:

```
Dim number As Integer
```

```
Do
```

```
    Console.Write("Δώσε έναν αριθμό ή -1 για ΤΕΡΜΑΤΙΣΜΟ. ")
```

```
    number = Console.ReadLine()
```

```
    number = number ^ 2
```

```
    Console.WriteLine("Το τετράγωνό του είναι " & number)
```

```
Loop While number <> -1
```

Ατέρμονες επαναλήψεις παράδειγμα

- Μια βελτίωση του προηγούμενου προγράμματος για να έχουμε τρόπο διαφυγής είναι και η παρακάτω:

```
Dim number As Integer
```

```
Do
```

```
    Console.Write("Δώσε έναν αριθμό ή -1 για ΤΕΡΜΑΤΙΣΜΟ. ")
```

```
    number = Console.ReadLine()
```

```
    if number = -1 Then Exit Do
```

```
    number = number ^ 2
```

```
    Console.WriteLine("Το τετράγωνό του είναι " & number)
```

```
Loop
```


Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

