

4. Επιλογή και Επανάληψη

4.1 Η Εντολή Επιλογής if .. then

Η εντολή If .. Then .. χρησιμοποιείται για την λήψη λογικών αποφάσεων σε ένα πρόγραμμα. Η εντολή αυτή έχει διάφορες μορφές σύνταξης οι οποίες περιγράφονται στις παρακάτω ενότητες.

Η μορφή If .. Then .. Else .. End If

Η εντολή **If <λογική συνθήκη> Then <τμήμα 1> Else <τμήμα 2> End If** εκτελεί το πρώτο τμήμα προγράμματος εφόσον η λογική συνθήκη της εντολής είναι αληθής διαφορετικά εκτελεί το δεύτερο τμήμα προγράμματος το οποίο ορίζεται μεταξύ των λέξεων else και end if.

```
Public Sub main()  
    Dim i As Integer  
    Console.WriteLine("Δώστε ένα ακέραιο")  
    i = Console.ReadLine()  
    'Έλεγχος αν ο αριθμός είναι θετικός ή αρνητικός  
    If i <= 0 Then  
        Console.WriteLine("Ο αριθμός είναι αρνητικός")  
    Else  
        Console.WriteLine("Ο αριθμός είναι θετικός")  
    End If  
    Console.ReadLine()  
End Sub
```

Παράδειγμα 4.1

Στο παραπάνω παράδειγμα ελέγχεται εάν ο αριθμός ο οποίος εισάγεται από τον χρήστη είναι μεγαλύτερος ή ίσος του μηδενός και αναλόγως εκτυπώνεται το αντίστοιχο μήνυμα. Το αποτέλεσμα της εκτέλεσης του προγράμματος εάν εισαχθεί ο αριθμός -45 είναι:

```
Δώστε ένα ακέραιο -45  
Ο αριθμός είναι αρνητικός
```

Η μορφή If .. Then .. End If

Η εντολή **If <λογική συνθήκη> Then <τμήμα προγράμματος>** εκτελεί ένα τμήμα προγράμματος εφόσον ικανοποιείται κάποια λογική συνθήκη.

Στο παρακάτω παράδειγμα η εντολή αυτή χρησιμοποιείται για να ελέγξει εάν ο αριθμός ο οποίος εισάγεται από τον χρήστη είναι θετικός. Στην περίπτωση που ο αριθμός είναι αρνητικός, τότε εκτυπώνεται το μήνυμα

ότι ο αριθμός είναι αρνητικός και σταματά η εκτέλεση της διαδικασίας με την εντολή exit sub.

```
Public Sub main()
    Dim i As Integer
    Console.WriteLine("Δώστε ένα θετικό ακέραιο")
    i = Console.ReadLine()
    'Έλεγχος αν ο αριθμός είναι θετικός
    If i <= 0 Then
        Console.WriteLine("Λάθος, ο αριθμός είναι αρνητικός")
        Exit Sub ' Τέλος της διαδικασίας
    End If
    'Εκτύπωση τετραγωνικής ρίζας
    Console.WriteLine(
        "Η τετραγωνική ρίζα είναι {0:N4}", Math.Sqrt(i))
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.1

Στην περίπτωση όπου το τμήμα προγράμματος το οποίο εκτελείται αποτελείται από μία μόνο εντολή, η εντολή μπορεί να συνταχθεί σε μία μόνο γραμμή.

π.χ.

```
if i <= 0 then Exit Sub
```

Ένθεση εντολών If .. Then

Η εντολή επιλογής If .. Then μπορεί να εμπεριέχει ή ενθέτει και άλλες εντολές επιλογής.

Η χρήση ένθετων (εμφωλευμένων) εντολών If .. Then γίνεται όταν υπάρχουν περισσότερες από δύο εναλλακτικές περιπτώσεις διακλάδωσης. Εάν για παράδειγμα χρειάζεται να γίνει έλεγχος ενός αριθμού εάν είναι αρνητικός, μηδενικός ή θετικός, τότε γίνεται χρήση ένθετων εντολών επιλογής if .. then.

```
Public Sub main()
    Dim i As Integer
    Console.WriteLine("Δώστε ένα ακέραιο ")
    i = Console.ReadLine()
    'Έλεγχος του αριθμού
    If i < 0 Then
        Console.WriteLine("Ο αριθμός είναι αρνητικός")
    Else
        If i = 0 Then
            Console.WriteLine("Ο αριθμός είναι μηδενικός")
        Else
            Console.WriteLine("Ο αριθμός είναι θετικός")
        End If
    End If
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.2

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος εάν δοθεί ο αριθμός 0 είναι:

Δώστε ένα ακέραιο 0
0 αριθμός είναι μηδενικός

Ένθεση με τη μορφή If .. then .. elseif .. else .. end if

Η σύνταξη των ένθετων εντολών if .. then μπορεί να απλοποιηθεί με την χρήση της λέξης elseif. Με την χρήση της ένθετης μορφής if .. then .. elseif .. else .. end if το προηγούμενο παράδειγμα μπορεί να απλοποιηθεί όπως φαίνεται στο παρακάτω παράδειγμα:

```
Public Sub main()  
    Dim i As Integer  
    Console.WriteLine("Δώστε ένα ακέραιο ")  
    i = Console.ReadLine()  
    'Έλεγχος του αριθμού  
    If i < 0 Then  
        Console.WriteLine("Ο αριθμός είναι αρνητικός")  
    ElseIf i = 0 Then  
        Console.WriteLine("Ο αριθμός είναι μηδενικός")  
    Else  
        Console.WriteLine("Ο αριθμός είναι θετικός")  
    End If  
    Console.ReadLine()  
End Sub
```

Παράδειγμα 4.3

4.2 Η εντολή πολλαπλής επιλογής Select

Η πολλαπλή επιλογή χρησιμοποιείται για την λήψη λογικών αποφάσεων όταν οι εναλλακτικές περιπτώσεις είναι περισσότερες από δύο. Η εντολή πολλαπλής επιλογής Select εξετάζει την τιμή κάποιας μεταβλητής (Integer, String κλπ) και αναλόγως της τιμής της μεταβλητής εκτελεί διαφορετικό τμήμα προγράμματος.

Το παρακάτω παράδειγμα με την εντολή Select εξετάζει την τιμή ενός μονοψήφιου θετικού ακέραιου ο οποίος εισάγεται από τον χρήστη, υπολογίζει την κατάλληλη αλφαριθμητική τιμή που αντιστοιχεί στον αριθμό και τον εκτυπώνει ολογράφως.

```
Public Sub main()  
    'Πρόγραμμα το οποίο ζητά έναν αριθμό από 0 έως 9  
    'και τον τυπώνει ολογράφως  
    Dim i As Integer 'Ο αριθμός που θα μετατραπεί  
    Dim s As String 'Ο αριθμός ολογράφως  
    Console.WriteLine("Δώστε ένα ακέραιο από 0 έως 9 : ")  
    i = Console.ReadLine()  
    'Έλεγχος του αριθμού  
    Select Case i  
        Case 0  
            s = "μηδέν"  
        Case 1  
            s = "ένα"  
        Case 2  
            s = "δύο"  
        Case 3  
            s = "τρία"  
        Case 4
```

```

        s = "τέσσερα"
    Case 5
        s = "πέντε"
    Case 6
        s = "έξι"
    Case 7
        s = "επτά"
    Case 8
        s = "οκτώ"
    Case 9
        s = "εννέα"
    Case Else
        'Εάν ο αριθμός δεν είναι από 0 έως 9
        s = "*****"
    End Select
    Console.WriteLine("Πληκτρολογήθηκε ο αριθμός {0}", s)
    Console.ReadLine()
End Sub

```

Παράδειγμα 4.4

Εάν δοθεί ο αριθμός 4, τότε το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

```

Δώστε ένα ακέραιο από 0 έως 9 : 4
Πληκτρολογήθηκε ο αριθμός τέσσερα

```

Στην περίπτωση όπου η ίδια επιλογή γίνεται για δύο ή περισσότερες τιμές, μπορούμε να παραθέσουμε τις τιμές μετά την λέξη case με την χρήση κόμματος.

Π.χ. αν i είναι η μεταβλητή πολλαπλής επιλογής

```
Case 1,3,5 'ισοδυναμεί με i=1 or i=2 or i=3
```

Εάν η ίδια επιλογή γίνεται για ένα εύρος τιμών, το εύρος τιμών μπορεί να ορισθεί με την χρήση της αρχικής τιμής, της λέξης To και της τελική τιμής του εύρους
πχ αν i είναι η μεταβλητή πολλαπλής επιλογής

```
Case 0 To 5 'ισοδυναμεί με i>=0 and i<=5
```

Επίσης με την χρήση του τελεστή Is μπορούμε να συγκρίνουμε την τιμή της μεταβλητής πολλαπλής επιλογής με άλλες τιμές ή μεταβλητές.

πχ αν i είναι η μεταβλητή πολλαπλής επιλογής

```
Case Is < 0 'ισοδυναμεί με i<0
```

Η μεταβλητή η οποία χρησιμοποιείται στην πολλαπλή επιλογή μπορεί να είναι και άλλος τύπος δεδομένων, όπως πραγματικός αριθμός, χαρακτήρας ή αλφαριθμητική μεταβλητή.

```

Public Sub main()
    'Πρόγραμμα το οποίο διαβάζει ένα χαρακτήρα
    'και εκτυπώνει την περιγραφή του

```

```

Dim c As Char 'Χαρακτήρας ο οποίος εισάγεται
Dim s As String 'Περιγραφή του χαρακτήρα
Console.WriteLine("Δώστε ένα ελληνικό χαρακτήρα : ")
c = Console.ReadLine()
'Πολλαπλή επιλογή ανάλογα με το χαρακτήρα
Select Case c
    Case "α" To "ω"
        s = "ελληνικός πεζός χαρακτήρας"
    Case "ά", "έ", "ό", "ή", "ύ", "ί", "ώ"
        s = "ελληνικός πεζός χαρακτήρας με τόνο"
    Case "Α" To "Ω"
        s = "ελληνικός κεφαλαίος χαρακτήρας"
    Case Is < " "
        s = "ειδικός χαρακτήρας"
    Case Else
        s = "κάποιος άλλος χαρακτήρας"
End Select
Console.WriteLine("Πληκτρολογήθηκε {0}", s)
Console.ReadLine()
End Sub

```

Παράδειγμα 4.5

Το παραπάνω παράδειγμα χρησιμοποιεί ως μεταβλητή πολλαπλής επιλογής ένα χαρακτήρα ο οποίος εισάγεται από το πληκτρολόγιο και κάνοντας χρήση πολλαπλών τιμών, εύρους τιμών και συγκρίσεων τιμών για την κάθε επιλογή εκτυπώνει την περιγραφή του χαρακτήρα που εισάγεται.

Το αποτέλεσμα της εκτέλεσης του προγράμματος εάν δοθεί ο χαρακτήρας "δ" είναι:

```

Δώστε ένα ελληνικό χαρακτήρα : δ
Πληκτρολογήθηκε ελληνικός πεζός χαρακτήρας

```

4.3 Η εντολή επιλογής IIF

Η εντολή IIF υπολογίζει μια λογική συνθήκη και αναλόγως επιστρέφει μια τιμή. Η σύνταξη της είναι:

```
<μεταβλητή> = IIF(<συνθήκη>, <τιμή 1>, <τιμή 2>)
```

Στο παρακάτω παράδειγμα η εντολή IIF χρησιμοποιείται για να ελέγξει εάν ένας αριθμός είναι άρτιος ή περιττός. Εάν το υπόλοιπο της διαίρεσης με τον αριθμό 2 είναι 0, τότε ο αριθμός είναι άρτιος διαφορετικά είναι περιττός.

```

Public Sub main()
    'Πρόγραμμα το οποίο ελέγχει εάν ένας αριθμός είναι περιττός
    Dim i As Integer
    Dim s As String
    Console.WriteLine("Δώστε ένα ακέραιο : ")
    i = Console.ReadLine()
    s = IIf(i Mod 2 = 0, "άρτιος", "περιττός")
    Console.WriteLine("Ο αριθμός {0} είναι {1}", i, s)
    Console.ReadLine()
End Sub

```

Παράδειγμα 4.6

Το αποτέλεσμα της εκτέλεσης του προγράμματος εάν δοθεί ο αριθμός 35 είναι:

```
Δώστε ένα ακέραιο : 35
Ο αριθμός 35 είναι περιττός
```

Η εντολή IIf επιστρέφει μια τιμή η οποία είναι τύπου Object. Εάν χρησιμοποιείται η επιλογή Option Strict On, τότε πρέπει να χρησιμοποιείται και η κατάλληλη συνάρτηση ή μέθοδος για την μετατροπή της επιστρεφόμενης τιμής.

Στο προηγούμενο παράδειγμα, εάν οριστεί η επιλογή Option Strict On, πρέπει να χρησιμοποιηθεί η συνάρτηση μετατροπής CStr.

```
s = CStr(IIf(i Mod 2 = 0, "άρτιος", "περιττός"))
```

4.4 Η εντολή πολλαπλής επιλογής Choose

Η εντολή Choose επιστέφει μια τιμή ανάλογα με την τιμή κάποιας μεταβλητής η οποία χρησιμοποιείται ως δείκτης. Η σύνταξη της εντολής αυτής είναι:

```
<μεταβλητή> = Choose(δείκτης, <τιμή 1>, <τιμή 2>, .. , <τιμή N>)
```

Εάν ο δείκτης έχει την τιμή 1, τότε επιστρέφεται η πρώτη τιμή, εάν έχει τιμή 2 επιστρέφεται η δεύτερη τιμή κ.ο.κ.

Το παρακάτω παράδειγμα χρησιμοποιεί την εντολή Choose για να εκτυπώσει ολογράφως την τιμή ενός ακεραίου αριθμού από 0 έως 9.

```
Public Sub main()
    'Πρόγραμμα το οποίο μετατρέπει αριθμό από 0 έως 9
    'και τον τυπώνει ολογράφως με την εντολή Choose
    Dim i As Integer 'Ο αριθμός που θα μετατραπεί
    Dim s As String 'Ο αριθμός ολογράφως
    Console.Write("Δώστε ένα ακέραιο από 0 έως 9 : ")
    i = Console.ReadLine()
    'Μετατροπή του αριθμού
    s = Choose(i + 1, "μηδέν", "ένα", "δύο", "τρία", "τέσσερα", _
               "πέντε", "έξι", "επτά", "οκτώ", "εννέα")
    Console.WriteLine("Πληκτρολογήθηκε ο αριθμός {0}", s)
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.7

Το αποτέλεσμα της εκτέλεσης του προγράμματος εάν δοθεί ο αριθμός 0 είναι:

```
Δώστε ένα ακέραιο από 0 έως 9 : 0
Πληκτρολογήθηκε ο αριθμός μηδέν
```

4.5 Η εντολή επανάληψης For

Η εντολή επανάληψης For επαναλαμβάνει την εκτέλεση ενός τμήματος του προγράμματος κάνοντας χρήση μιας μεταβλητής η οποία χρησιμοποιείται ως δείκτης των επαναλήψεων. Η γενική σύνταξη της εντολής είναι:

```
For <μεταβλητή> [As <Τύπος>]= <αρχική τιμή> To <τελική τιμή> [Step <βήμα>]
    [if <συνθήκη> then Continue For]
    [if <συνθήκη> then Exit For]
    <Τμήμα του προγράμματος το οποίο επαναλαμβάνεται>
Next
```

Ο ορισμός του βήματος των επαναλήψεων (Step) είναι προαιρετικός και μπορεί να παραλειφθεί. Σε περίπτωση που δεν έχει ορισθεί βήμα, τότε το βήμα είναι ίσο με την μονάδα.

Ο δείκτης των επαναλήψεων στην πρώτη επανάληψη παίρνει την αρχική τιμή και αυξάνεται διαδοχικά με το βήμα των επαναλήψεων μέχρι να φτάσει την τελική τιμή.

```
Public Sub main()
    Dim i As Integer
    For i = 1 To 5
        Console.WriteLine(i)
    Next
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.8

Στο παραπάνω παράδειγμα η μεταβλητή *i* χρησιμοποιείται ως δείκτης των επαναλήψεων με αρχική τιμή 1 και τελική τιμή 5. Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

```
1
2
3
4
5
```

Το βήμα των επαναλήψεων μπορεί να είναι αρνητικό. Στην περίπτωση αυτή η τελική τιμή πρέπει να είναι μικρότερη από την τελική τιμή.

```
Public Sub main()
    Dim i As Integer
    For i = 5 To 1 Step -1
        Console.WriteLine(i)
    Next
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.9

Στο παραπάνω παράδειγμα η μεταβλητή *i* μειώνεται από την αρχική τιμή 5 στην τελική τιμή 1.

Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

```
5
4
3
2
1
```

Ο δείκτης των επαναλήψεων μπορεί να είναι οποιοσδήποτε τύπος δεδομένων για τον οποίο ορίζονται οι τελεστές \leq , \geq , + και -. Μπορεί για παράδειγμα να χρησιμοποιηθεί πραγματικός τύπος μεταβλητής διπλής ακρίβειας με βήμα 0.1.

```
Public Sub main()
    For x As Double = 10 To 10.5 Step 0.1
        Console.WriteLine(x)
    Next
    Console.ReadLine()
End sub
```

Παράδειγμα 4.10

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι:

```
10
10,1
10,2
10,3
10,4
10,5
```

Η δήλωση της μεταβλητής του δείκτη των επαναλήψεων μπορεί να γίνει προαιρετικά στην ίδια γραμμή με την εντολή For.

```
Public Sub main()
    For i As Integer = 1 To 5
        Console.WriteLine(i)
    Next
End Sub
```

Παράδειγμα 4.11

Με την προαιρετική χρήση της εντολής Continue For σταματά η εκτέλεση της τρέχουσας επανάληψης και οι επαναλήψεις συνεχίζονται με την επόμενη τιμή του δείκτη των επαναλήψεων.

```
Public Sub main()
    For i As Integer = 1 To 5
        If i=3 then Continue For
        Console.WriteLine(i)
    Next
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.12

Στο παραπάνω παράδειγμα σταματά η εκτέλεση της επανάληψης όταν ο δείκτης έχει την τιμή 3. Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

```
1
2
4
5
```

Με την προαιρετική εντολή Exit For διακόπτεται η εντολή επανάληψης

```
Public Sub main()
    For i As Integer = 1 To 5
        If i=3 then Exit For
        Console.WriteLine(i)
    Next
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.13

Στο παραπάνω παράδειγμα διακόπτονται όλες οι επαναλήψεις όταν ο δείκτης έχει την τιμή 3. Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

```
1
2
```

Ο δείκτης των επαναλήψεων μπορεί να χρησιμοποιηθεί σε διάφορους υπολογισμούς όπως για παράδειγμα την εύρεση του αθροίσματος ή του παραγοντικού ενός ακεραίου.

```
Public Sub main()
    'Πρόγραμμα το οποίο υπολογίζει
    'το άθροισμα των αριθμών από 1 έως 100
    Dim sum As Integer = 0
    For i As Integer = 1 To 100
        sum += i 'Πρόσθεση του i στο sum
    Next
    Console.WriteLine(sum)
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.14 Εύρεση του αθροίσματος των αριθμών από 1 έως 100

Το παραπάνω παράδειγμα χρησιμοποιεί μια μεταβλητή sum ως αθροιστή με αρχική τιμή 0. Σε κάθε επανάληψη ο δείκτης επανάληψης ο οποίος παίρνει τιμές από 1 έως 100 προστίθεται στην μεταβλητή sum. Το sum μετά το τέλος των επαναλήψεων είναι ίσο με το άθροισμα των ακεραίων από 1 έως 100.

```
Public Sub main()
    'Πρόγραμμα το οποίο υπολογίζει
    'το παραγοντικό ενός αριθμού
    'n = 1*2*3*..n
    Dim n As Integer ' Ο ακεραίος αριθμός
    Dim p As Integer = 1 ' Το παραγοντικό
    Console.Write("Δώστε έναν ακεραίο: ")
```

```

n = Console.ReadLine()
For i As Integer = 2 To n
    p *= i ' p = p * i
Next
Console.WriteLine
    ("Το παραγοντικό του {0} είναι {1}", n, p)
Cosnole.Readline()
End Sub

```

Παράδειγμα 4.15 Εύρεση του παραγοντικού ενός ακέραιου

Το παραπάνω παράδειγμα χρησιμοποιεί την μεταβλητή *p* με αρχική τιμή 1 για να υπολογίσει το παραγοντικό ενός ακέραιου *n* ο οποίος εισάγεται από τον χρήστη. Σε κάθε επανάληψη το *p* γίνεται ίσο με γινόμενο της προηγούμενης τιμής του επί τον δείκτη των επαναλήψεων *i* ο οποίος παίρνει τις τιμές από 2 έως *n*.

Επανάληψη	Τιμή του <i>n</i>	Τιμή του <i>p</i>
1	2	1*2
2	3	1*2*3
3	4	1*2*3*4
..		
<i>n</i> -1	<i>n</i>	1*2*3*4*...* <i>n</i>

Το αποτέλεσμα της εκτέλεσης του προγράμματος εάν δοθεί ο αριθμός 6 είναι:

```

Δώστε έναν ακέραιο: 6
Το παραγοντικό του 6 είναι 720

```

Ένθεση εντολών For

Το τμήμα του προγράμματος το οποίο επαναλαμβάνεται με μία εντολή For μπορεί να εμπεριέχει κάποια άλλη εντολή For. Είναι δηλαδή δυνατή η ένθεση εντολών For και γενικότερα εντολών επανάληψης μέσα σε άλλες εντολές επανάληψης.

```

Public Sub main()
    Dim i, j As Integer
    For i = 1 To 3
        For j = 1 To 2
            Console.WriteLine("i={0} j={1}", i, j)
        Next
    Next
    Cosnole.Readline()
End sub

```

Παράδειγμα 4.16

Το παραπάνω πρόγραμμα ενθέτει μια εντολή επανάληψης με δείκτη *j* ο οποίος παίρνει τις τιμές από 1 έως 2 σε μια εντολή επανάληψης με δείκτη *i* ο οποίος παίρνει τις τιμές από 1 έως 3. Η ένθεση έχει ως αποτέλεσμα η εντολή επανάληψης με δείκτη *j* να εκτελείται για κάθε αλλαγή της τιμής του *i*. Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος έχει σαν αποτέλεσμα:

```
i=1 j=1
i=1 j=2
i=2 j=1
i=2 j=2
i=3 j=1
i=3 j=2
```

Το παρακάτω παράδειγμα χρησιμοποιεί ένθετες εντολές For για να εκτυπώσει ένα πλαίσιο 5 γραμμών και 6 στηλών με τους χαρακτήρες "*". Στο τέλος της ένθετης εντολής επανάληψης με δείκτη j η οποία εκτυπώνει 6 χαρακτήρες "*" στην ίδια γραμμή, εκτελείται η εντολή Console.WriteLine() με την οποία γίνεται η αλλαγή γραμμής για κάθε διαφορετική τιμή του i.

```
Public Sub main()
    Dim i, j As Integer
    For i = 1 To 5
        For j = 1 To 6
            Console.Write("*")
        Next
        Console.WriteLine()
    Next
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.17

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι:

```
*****
*****
*****
*****
*****
```

4.6 Η εντολή επανάληψης Do Loop

Η εντολή επανάληψης Do Loop επαναλαμβάνει διαρκώς την εκτέλεση ενός τμήματος του προγράμματος. Οι επαναλήψεις συνεχίζουν ή σταματούν ανάλογα με την τιμή κάποιας λογικής συνθήκης. Εάν η λογική συνθήκη ελέγχεται στην αρχή τότε η εντολή έχει την μορφή:

```
Do [ While ή Until <συνθήκη> ]
    <εντολή>
    [if <συνθήκη> then Exit Do]
    [if <συνθήκη> then Continue Do]
    <εντολή>
Loop
```

Εάν η λογική συνθήκη ελέγχεται στην τέλος τότε η εντολή έχει την μορφή:

```
Do
    <εντολή>
    [if <συνθήκη> then Exit Do]
    [if <συνθήκη> then Continue Do]
```

```
<εντολή>
Loop [ While ή Until <συνθήκη> ]
```

Εάν χρησιμοποιείται η δήλωση While οι επαναλήψεις διαρκούν όσο η τιμή της λογικής συνθήκης είναι αληθής. Αντίθετα όταν χρησιμοποιείται η δήλωση Until οι επαναλήψεις διακόπτονται όταν η τιμή της λογικής συνθήκης γίνει αληθής.

Με την προαιρετική εντολή Exit Do διακόπτονται οι επαναλήψεις, ενώ με την προαιρετική εντολή Continue Do δεν εκτελούνται οι επόμενες εντολές του τμήματος του προγράμματος που επαναλαμβάνεται.

Η εντολή μπορεί επίσης να συνταχθεί χωρίς της χρήση λογικής συνθήκης στην αρχή ή το τέλος της εντολής.

```
Do
  <εντολή>
  [if <συνθήκη> then Exit Do]
  [if <συνθήκη> then Continue Do]
  <εντολή>
Loop
```

Στην περίπτωση αυτή οι επαναλήψεις θα πρέπει να διακόπτονται με την εντολή Exit Do διαφορετικά οι επαναλήψεις θα συνεχίζουν διαρκώς.

Οι διάφορες μορφές της εντολής Do Loop δίνουν μεγάλη ευελιξία στην σύνταξη τμημάτων του προγράμματος τα οποία επαναλαμβάνονται. Μπορούν να χρησιμοποιηθούν στην θέση της εντολής For, αλλά και για την επανάληψη τμημάτων του προγράμματος όταν δεν είναι γνωστός εκ των προτέρων ο αριθμός των επαναλήψεων.

Τα παρακάτω παραδείγματα είναι ισοδύναμα με το παράδειγμα 4.8 όπου χρησιμοποιείται εντολή For για να εκτυπώσει του αριθμούς από ένα έως 5.

```
Public Sub main()
  Dim i As Integer = 1           'Αρχική τιμή
  Do While i <= 5               'Τελική τιμή
    Console.WriteLine(i)
    i += 1                      'Βήμα
  Loop
  Console.ReadLine()
End Sub
```

Παράδειγμα 4.18

```
Public Sub main()
  Dim i As Integer = 1           'Αρχική τιμή
  Do Until i > 5                 'Τελική τιμή
    Console.WriteLine(i)
    i += 1                      'Βήμα
  Loop
  Console.ReadLine()
End Sub
```

Παράδειγμα 4.19 Έλεγχος συνθήκης στην αρχή με Until

```
Public Sub main()
    Dim i As Integer = 1      'Αρχική τιμή
    Do
        Console.WriteLine(i)
        i += 1                'Βήμα
    Loop While i <= 5        'Τελική τιμή
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.20 Έλεγχος συνθήκης στην τέλος με While

```
Public Sub main()
    Dim i As Integer = 1      'Αρχική τιμή
    Do
        Console.WriteLine(i)
        i += 1                'Βήμα
    Loop Until i > 5         'Τελική τιμή
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.21 Έλεγχος συνθήκης στην τέλος με Until

```
Public Sub main()
    Dim i As Integer = 1      'Αρχική τιμή
    Do
        Console.WriteLine(i)
        i += 1                'Βήμα
        If i > 5 Then Exit Do 'Τελική τιμή
    Loop
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.22 Χρήση Do Loop χωρίς έλεγχο αρχικής ή τελικής συνθήκης

Σε αρκετές περιπτώσεις ο αριθμός των επαναλήψεων δεν είναι εκ των προτέρων γνωστός. Στο παρακάτω παράδειγμα το πρόγραμμα διαβάζει διαδοχικά ακέραιους αριθμούς τους οποίους εισάγει ο χρήστης. Η εισαγωγή των αριθμών σταματά όταν ο χρήστης πληκτρολογήσει τον αριθμό 0. Το πρόγραμμα υπολογίζει και εκτυπώνει τον αριθμό των αρνητικών αριθμών που δόθηκαν από τον χρήστη.

```
Public Sub main()
    'Πρόγραμμα το οποίο μετρά τον αριθμό
    'των αρνητικών αριθμών που εισάγει ο χρήστης
    Dim counter As Integer = 0 'Μετρητής αρνητικών αριθμών
    Dim i As Integer 'Αρχική τιμή
    Do
        Console.Write("Δώστε ένα ακέραιο ή 0 για τέλος: ")
        i = Console.ReadLine()
        If i < 0 Then counter += 1
    Loop While i <> 0
    Console.WriteLine
    ("Δώσατε {0} αρνητικούς αριθμούς ", counter)
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.23

Το αποτέλεσμα της εκτέλεσης του προγράμματος είναι:

```
Δώστε ένα ακέραιο ή 0 για τέλος: -2
Δώστε ένα ακέραιο ή 0 για τέλος: 4
Δώστε ένα ακέραιο ή 0 για τέλος: -9
Δώστε ένα ακέραιο ή 0 για τέλος: 6
Δώστε ένα ακέραιο ή 0 για τέλος: 0
Δώσατε 2 αρνητικούς αριθμούς
```

4.7 Η εντολή επανάληψης While

Η εντολή επανάληψης while είναι αντίστοιχη της εντολής Do While Loop. Η σύνταξη της είναι:

```
While <συνθήκη>
  <εντολή>
  [if <συνθήκη> then Exit While]
  [if <συνθήκη> then Continue While]
  <εντολή>
End While
```

και είναι αντίστοιχη της:

```
Do While <συνθήκη>
  <εντολή>
  [if <συνθήκη> then Exit Do]
  [if <συνθήκη> then Continue Do]
  <εντολή>
Loop
```

Στο παρακάτω παράδειγμα η εντολή επανάληψης While χρησιμοποιείται για να υπολογίσει και εκτυπώσει πόσα χρόνια χρειάζονται για τον διπλασιασμό ενός αρχικού κεφαλαίου το οποίο κατατίθεται στην τράπεζα με σταθερό επιτόκιο και ανατοκισμό στο τέλος κάθε έτους.

```
Public Sub main()
  'Πρόγραμμα το οποίο υπολογίζει τα χρόνια που
  'χρειάζονται για το διπλασιασμό ενός κεφαλαίου
  Dim arxikoKefalaio As Single 'Το αρχικό κεφάλαιο
  Dim kefalαιο As Single 'Το κεφάλαιο στο τέλος κάθε έτους
  Dim epitokio As Single 'Το επιτόκιο
  Console.WriteLine("Δώστε αρχικό κεφάλαιο: ")
  arxikoKefalaio = Console.ReadLine()
  Console.WriteLine("Δώστε επιτόκιο: ")
  epitokio = Console.ReadLine()
  Dim metritis As Integer = 0 ,Μετρητής ετών
  kefalαιο = arxikoKefalaio
  While kefalαιο < (2 * arxikoKefalaio)
    metritis += 1 'αύξηση έτους
    'Πρόσθεση τόκου στο κεφάλαιο
    kefalαιο += kefalαιο * epitokio / 100
    Console.WriteLine(
      "Έτος {0} Κεφάλαιο:{1:N2}", metritis, kefalαιο)
  End While
  Console.ReadLine()
End Sub
```

Το αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος με αρχικό κεφάλαιο 1000 και επιτόκιο 15 % είναι:

```
Δώστε αρχικό κεφάλαιο: 1000
Δώστε επιτόκιο: 15
Έτος 1 Κεφάλαιο: 1.150,00
Έτος 2 Κεφάλαιο: 1.322,50
Έτος 3 Κεφάλαιο: 1.520,88
Έτος 4 Κεφάλαιο: 1.749,01
Έτος 5 Κεφάλαιο: 2.011,36
Το κεφάλαιο διπλασιάζεται σε 5 έτη
```

4.8 Η εντολή επανάληψης For Each

Η εντολή επανάληψης For Each χρησιμοποιείται για να απαριθμήσει όλα τα μέλη ενός πίνακα ή άλλων συλλογών (Collections) δεδομένων.

Στο παρακάτω παράδειγμα η εντολή For Each χρησιμοποιείται για να εκτυπώσει τους χαρακτήρες μια αλφαριθμητικής μεταβλητής. Οι αλφαριθμητικές μεταβλητές για τις περισσότερες γλώσσες προγραμματισμού είναι πίνακες χαρακτήρων.

```
Public Sub main()
    Dim onoma As String
    Console.WriteLine("Δώστε ένα όνομα: ")
    onoma = Console.ReadLine()
    Dim c As Char
    Console.WriteLine("Οι χαρακτήρες του ονόματος είναι:")
    For Each c In onoma
        Console.WriteLine(c)
    Next
    Console.ReadLine()
End Sub
```

Παράδειγμα 4.25

```
Δώστε ένα όνομα: Άρης
Οι χαρακτήρες του ονόματος είναι:
Α
ρ
η
ς
```

4.9 Η εντολή goto

Η εντολή goto αν και δεν είναι εντολή επανάληψης ή επιλογής, μπορεί να χρησιμοποιηθεί για την επιλογή ή επανάληψη υποπρογραμμάτων. Τα προγράμματα που χρησιμοποιούν εντολές goto γίνονται με δυσκολία κατανοητά και είναι ιδιαίτερα εύκολα να χαθεί ο έλεγχος της ροής του προγράμματος.

Η εντολή goto δίνει την εντολή στο πρόγραμμα να συνεχίσει την εκτέλεση του σε κάποιο άλλη γραμμή η σημείο του προγράμματος. Η εντολή goto παίρνει ως παράμετρο τον αριθμό γραμμής ή το όνομα ετικέτας (label) του προγράμματος στο οποίο θα γίνει η μετάβαση.

Ετικέτες μπορούμε να ορίσουμε σε οποιοδήποτε σημείο του προγράμματος με την χρήση του ονόματος της ετικέτας και τον χαρακτήρα ":".

Το παρακάτω πρόγραμμα εκτυπώνει τους αριθμούς από 1 έως 5 στην οθόνη.

```
Public Sub main()  
    Dim i As Integer = 1  
arxi:  
    Console.WriteLine(i)  
    i += 1  
    If i <= 5 Then GoTo arxi  
    Console.ReadLine()  
End Sub
```

Η χρήση της εντολής αυτής δεν ενδείκνυται παρά μόνο σε ελάχιστες περιπτώσεις όπως όταν χρειάζεται να διακοπούν ένθετες εντολές επανάληψης και ο έλεγχος να δοθεί σε συγκεκριμένο σημείο του προγράμματος για να αντιμετωπισθούν λάθη κατά την εκτέλεση του προγράμματος.

4.10 Ασκήσεις

Ασκηση 4.1

Δημιουργήστε εφαρμογή lab_4_ask_1 η οποία:

- α) ζητά από τον χρήστη να εισάγει ένα πραγματικό αριθμό x και στην συνέχεια εξετάζει και εκτυπώνει εάν ο αριθμός x ανήκει στο διάστημα [1 .. 10]
- β) ζητά από τον χρήστη να εισάγει δύο ακέραιους αριθμούς m και n και στην συνέχεια εκτυπώνει την τιμή του μεγαλύτερου από αυτούς.

Ασκηση 4.2

Δημιουργήστε εφαρμογή lab_4_ask_2 η οποία ζητά από τον χρήστη να εισάγει έναν ακέραιο από 0 έως 9999 και στην συνέχεια κάνοντας χρήση της εντολής επιλογής Select εκτυπώνει τον αριθμό των ψηφίων του αριθμού (μονοψήφιος, διψήφιος, τριψήφιος ή τετραψήφιος).

Ασκηση 4.3

Δημιουργήστε εφαρμογή lab_4_ask_3 η οποία ζητά από τον χρήστη να εισάγει τα συνολικά εισόδημα του και στην συνέχεια υπολογίζει και εκτυπώνει το φόρο τον οποίο θα πρέπει να πληρώσει. Ο φορολογικός συντελεστής είναι:

- Για τα πρώτα 8.400 Ευρώ 0%
- Για τα επόμενα 5.000 Ευρώ 15%
- Για τα επόμενα 10.000 Ευρώ 30%
- Για επιπλέον εισόδημα 40%

Πχ. Για εισόδημα 20.000 Ευρώ ο φόρος είναι: $8400*0\%+5000*15\%+(20000-13400)*30\%=2730$

Ασκηση 4.4

Δημιουργήστε εφαρμογή lab_4_ask_4 η οποία ζητά από τον χρήστη να εισάγει τρεις πραγματικούς αριθμούς και την συνέχεια τους εκτυπώνει στην οθόνη σε αύξουσα σειρά.

Υπόδειξη: Χρησιμοποιήστε ένθετες εντολές If για να ελέγξετε όλες τις περιπτώσεις

Ασκηση 4.5

Δημιουργήστε εφαρμογή lab_4_ask_5 η οποία κάνοντας χρήση εντολών επανάληψης εκτυπώνει τις αντιστοιχία σε Ευρώ των 100, 200, 300 έως και 1000 δραχμών

Ασκηση 4.6

Δημιουργήστε εφαρμογή lab_4_ask_6 η οποία υπολογίζει και εκτυπώνει το άθροισμα των περιττών

ακεραίων από 1 έως 199.

Άσκηση 4.7

Δημιουργήστε εφαρμογή lab_4_ask_7 η οποία ζητά από τον χρήστη να εισάγει ένα ακέραιο αριθμό N και στην συνέχεια εκτυπώνει τις τιμές των παρακάτω παραστάσεων:

- $1+2+3+...+N$
- $1^2+2^2+3^2+...+N^2$
- $1*2*3*...*N$

Άσκηση 4.8

Δημιουργήστε εφαρμογή lab_4_ask_8 η οποία με την χρήση ένθετων εντολών επανάληψης εκτυπώνει τον παρακάτω πίνακα:

1	2	3	4	5
2	3	4	5	6
3	4	5	6	7
4	5	6	7	8

Άσκηση 4.9

Δημιουργήστε εφαρμογή lab_4_ask_9 η οποία ζητά από τον χρήστη να εισάγει διαδοχικά 30 ακεραίους αριθμούς και στην συνέχεια εκτυπώνει το άθροισμα τους, τον μέσο όρο τους, τον μεγαλύτερο αριθμό και το πλήθος των αρνητικών αριθμών.

Άσκηση 4.10

Δημιουργήστε εφαρμογή lab_4_ask_10 η οποία κάνοντας χρήση εντολής επανάληψης Do Loop, υπολογίζει και εκτυπώνει τα ημίτονα και συνημίτονα των γωνιών από 10 έως 20 μοίρες με βήμα 0.1

Άσκηση 4.11

Δημιουργήστε εφαρμογή lab_4_ask_11 η οποία διαβάζει διαδοχικά ακεραίους αριθμούς τους οποίους εισάγει ο χρήστης. Η εισαγωγή των αριθμών σταματά όταν ο χρήστης πληκτρολογήσει τον αριθμό 0 και στην συνέχεια το πρόγραμμα εκτυπώνει στην οθόνη το άθροισμα τους, τον μικρότερο αριθμό, το πλήθος των περιττών αριθμών και το πλήθος των αριθμών οι οποίοι είναι μεγαλύτεροι του 20.

Άσκηση 4.12

Δημιουργήστε εφαρμογή lab_4_ask12 η υπολογίζει την μικρότερη τιμή του N για την οποία η παρακάτω παράσταση είναι μεγαλύτερη του 65000.

$$1^2 + 2^2 + 3^2 + \dots + N^2$$

Άσκηση 4.13

Σφαίρα αφήνεται να πέσει από αρχικό ύψος x . Σε κάθε αναπήδηση η σφαίρα χάνει το 10% του προηγούμενου ύψους. Η σφαίρα σταματά την κίνηση της όταν το ύψος της γίνεται ≤ 0.01 . Γράψτε εφαρμογή `lab_4_ask_13` ζητά από τον χρήστη να δώσει το αρχικό ύψος x και στην συνέχεια υπολογίζει και εκτυπώνει τον αριθμό των αναπηδήσεων που θα κάνει η σφαίρα.

Άσκηση 4.14

Δημιουργήστε εφαρμογή `lab_4_ask_14` η οποία ζητά από τον χρήστη να εισάγει δύο αριθμούς x και y και στην συνέχεια υπολογίζει το υπόλοιπο της διαίρεσης x / y με την χρήση διαδοχικών αφαιρέσεων του y από το x .

Άσκηση 4.15

Δημιουργήστε εφαρμογή `lab_4_ask_15` η οποία εκτυπώνει τους Πυθαγόρειους αριθμούς από το 1 μέχρι το 100. Τρεις ακέραιοι αριθμοί a, b, c λέγονται Πυθαγόρειοι αν ισχύει: $a^2 + b^2 = c^2$

Υπόδειξη: Χρησιμοποιήστε τρεις ένθετες εντολές επανάληψης με μεταβλητές a, b και c .

Άσκηση 4.16

Δημιουργήστε εφαρμογή `lab_4_ask_16` η οποία ζητά από τον χρήστη να εισάγει έναν ακέραιο αριθμό, εκτυπώνει τους διαιρέτες του αριθμού και εκτυπώνει εάν ο αριθμός αυτός είναι πρώτος ή όχι.

Υπόδειξη: Πρώτος αριθμός είναι ο αριθμός ο οποίος διαιρείται ακριβώς μόνο με την μονάδα και τον εαυτό του. Χρησιμοποιήστε εντολή επανάληψης για να εκτυπώσετε και να μετρήσετε τους διαιρέτες του αριθμού. Ανάλογα με το αποτέλεσμα της μέτρησης εκτυπώστε εάν ο αριθμός είναι πρώτος ή όχι.