



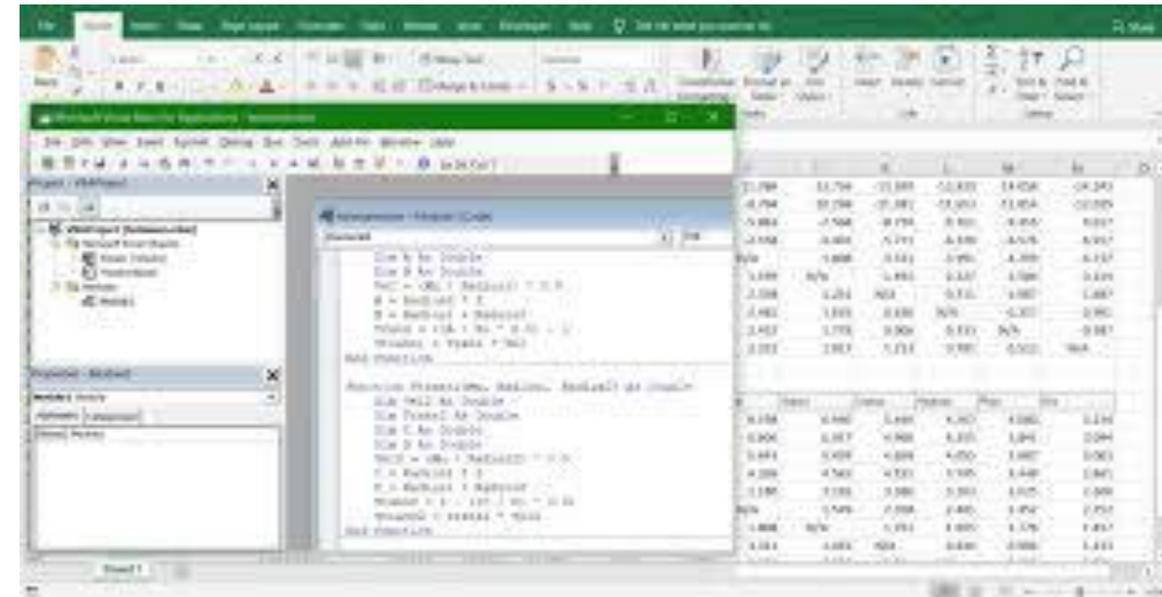
Excel VBA

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

VBA: Visual Basic for Application Γλώσσα Προγραμματισμού της Microsoft η οποία χρησιμοποιείται κυρίως στο πακέτο MS Office

Στο Ms Excel αξιοποιείται κυρίως στην αυτοματοποίηση «εργασιών» (tasks) και στην δημιουργία νέων πρόσθετων λειτουργιών.

Visual Basic with Excel



VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Η VBA είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού.

Κάθε στοιχείο του Excel θεωρείται σαν ένα αντικείμενο (object), που έχει το δικό του σύνολο ιδιοτήτων και μεθόδων και μπορεί με τη σειρά του να περιέχει άλλα αντικείμενα!!

Αντικειμενοστραφής προγραμματισμός: βάζει μαζί τα δεδομένα και τις διαδικασίες (μεθόδους) σχετικές με τα δεδομένα



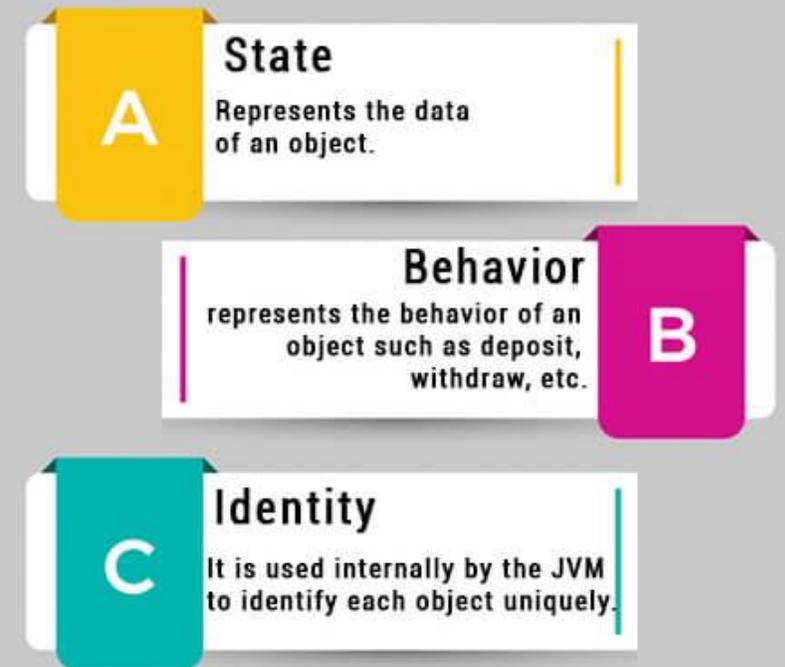
VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Αντικείμενα: ένα αντικείμενο στον κώδικα αναπαριστά μια μονάδα/οντότητα/έννοια η οποία έχει:

1. Μια κατάσταση (state), η οποία ορίζεται από ορισμένα χαρακτηριστικά
2. Μια συμπεριφορά (behavior), η οποία ορίζεται από ορισμένες ενέργειες που μπορεί να εκτελέσει το αντικείμενο
3. Μια ταυτότητα (id) που την ξεχωρίζει από τις υπόλοιπες μονάδες οντότητες έννοιες ίδιου τύπου

Παραδείγματα αντικειμένων: ένας άνθρωπος , ένα πράγμα

Characteristics of Object

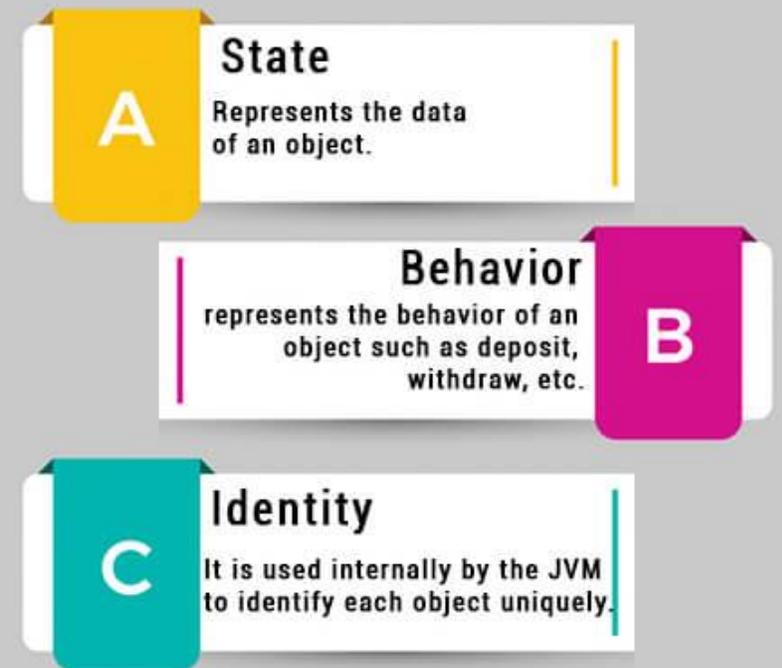


VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Κατάσταση (state): Κάθε αντικείμενο διέπεται από συγκεκριμένα χαρακτηριστικά που ορίζουν την κατάσταση του. Για παράδειγμα ένα αυτοκίνητο έχει συγκεκριμένο χρώμα, ανήκει σε συγκεκριμένο μοντέλο και κάθε χρονική στιγμή έχει μία ταχύτητα και μία κατεύθυνση κίνησης. Τα παραπάνω χαρακτηριστικά διέπουν όλα τα αυτοκίνητα, αλλά κάθε αυτοκίνητο έχει τις δικές του τιμές οι οποίες μπορεί να διαφέρουν από τις τιμές άλλων αυτοκινήτων.

Συμπεριφορά (behaviour): Κάθε αντικείμενο διέπεται από συγκεκριμένους κανόνες συμπεριφοράς. Για παράδειγμα σε ένα αυτοκίνητο εάν στρίψουμε το τιμόνι στα αριστερά το αυτοκίνητο κινείται αριστερά. Ο παραπάνω γενικός κανόνας ισχύει για όλα τα αυτοκίνητα. Η σχέση μεταξύ της γωνίας στροφής του τιμονιού ενός αυτοκινήτου και της αλλαγή πορείας που αυτή συνεπάγεται μπορεί να διαφέρει μεταξύ διαφορετικών μοντέλων.

Characteristics of Object

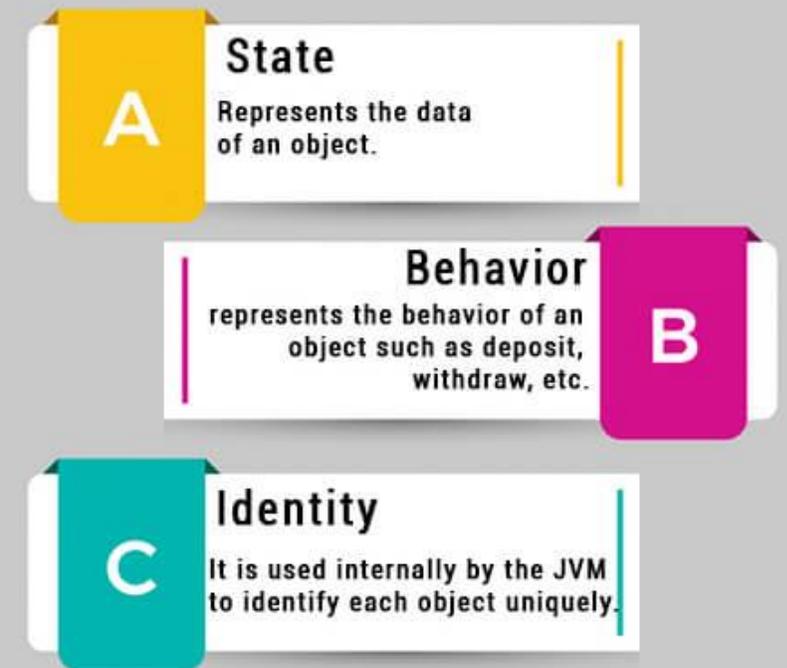


VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Κλάση: μια αφηρημένη περιγραφή αντικειμένων με κοινά χαρακτηριστικά και κοινή συμπεριφορά:

1. Μια κλάση είναι το καλούπι που μπορεί να παράγει αντικείμενα
2. Ένα αντικείμενο είναι ένα στιγμιότυπο μίας κλάσης

Characteristics of Object



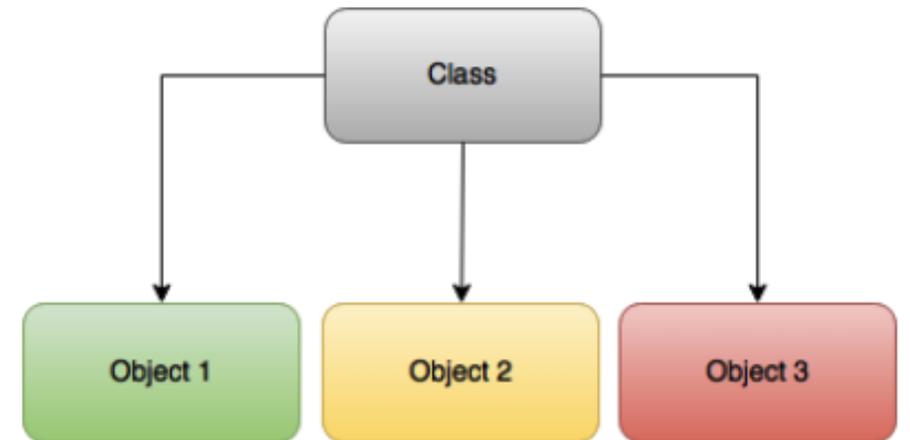
VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Κλάση: Τα παραδείγματα αντικειμένων που αναφέρθηκαν παραπάνω μπορεί κάποιος να τα δει με δύο διαφορετικούς τρόπους: α) το σχέδιο του κάθε αντικειμένου και β) το αντικείμενο αυτό καθ' αυτό.

Για παράδειγμα, όταν παράγεται ένα αυτοκίνητο, αυτό αποτελεί υλοποίηση του βασικού σχεδίου του αυτοκινήτου, έτσι όπως αυτό υλοποιείται από το εργοστάσιο κατασκευής. Μέρος του σχεδίου είναι η ύπαρξη κάποιων παραμέτρων όπως το χρώμα ή το επίπεδο του εξοπλισμού.

Στον αντικειμενοστραφή προγραμματισμό, το ρόλο του σχεδίου με την έννοια που αναφέρεται παραπάνω τον παίζει η κλάση. Η κλάση αποτελεί το σχέδιο με βάση το οποίο αντικείμενα του τύπου της συγκεκριμένης κλάσης υλοποιούνται σε ένα πρόγραμμα.

Όλα τα αντικείμενα της ίδιας κλάσης διέπονται από τις ίδιες εσωτερικές μεταβλητές (που αποτελούν τα δεδομένα ή πεδία της κλάσης) και από τις ίδιες συμπεριφορές (μεθόδους ή συναρτήσεις). Τα αντικείμενα της ίδιας κλάσης μπορεί να διαφέρουν ως προς τις τιμές των εσωτερικών τους μεταβλητών.

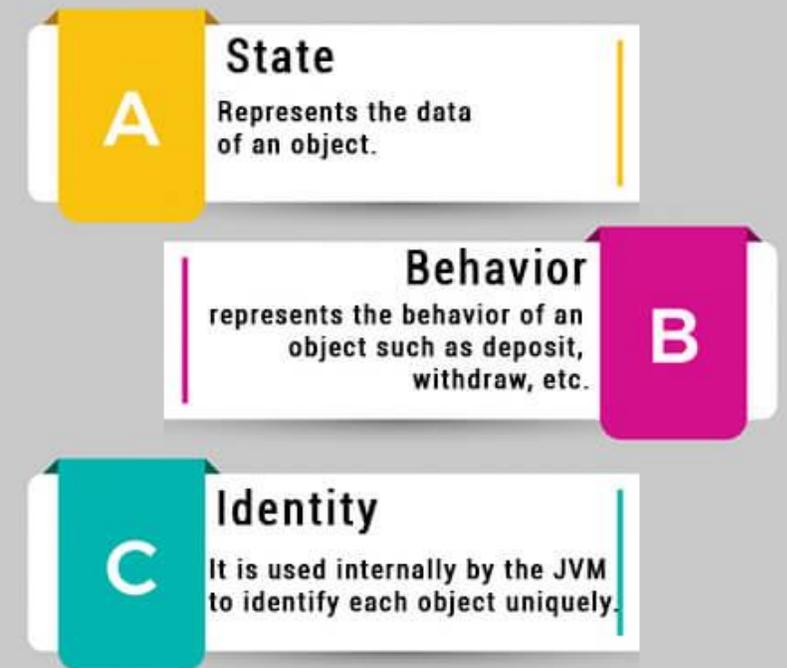


VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Παραδείγματα:

1. **Γενικό παράδειγμα:** Ένας άνθρωπος έχει τις ιδιότητες όνομα, ηλικία, φύλλο κτλ αλλά επίσης μπορεί και να εκτελεί διάφορες ενέργειες όπως π.χ. να τρέξει, να περπατήσει, να φάει κτλ.
2. η κλάση **Φοιτητής** έχει τα γενικά χαρακτηριστικά (όνομα, βαθμοί) και τη συμπεριφορά «**Εκτύπωσε**»
 - I. ο φοιτητής X είναι ένα αντικείμενο της κλάσης Φοιτητής με κατάσταση τα χαρακτηριστικά [Κώστας , 10,8]
3. η κλάση **Car** έχει τα χαρακτηριστικά (brand , color) και τη συμπεριφορά [drive , stop]
 - I. το αυτοκίνητο ZKT1234 είναι ένα αντικείμενο της κλάσης Car με κατάσταση τα χαρακτηριστικά [BMW , red]

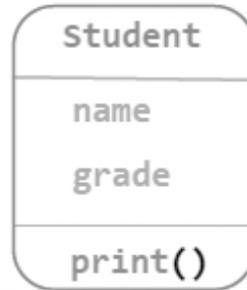
Characteristics of Object



VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Κλάση

Μια αφηρημένη περιγραφή ενός φοιτητή



Όνομα κλάσης

Πεδία κλάσης: Ιδιότητες/Χαρακτηριστικά

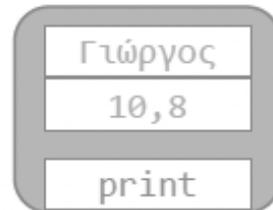
Μέθοδοι κλάσης: λειτουργίες

Αντικείμενα

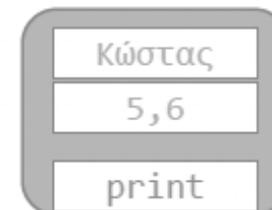
Το κάθε αντικείμενο έχει

- μια κατάσταση (name, grade)
- ενέργειες (print)
- ταυτότητα (X,Y,Z)

Student X:



Student Y:



Student Z:



VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Η κλάση γενικά ορίζει τον τύπο του αντικειμένου

1. Τα χαρακτηριστικά του αντικειμένου
2. Τις ενέργειες που μπορεί να επιτελέσει

Πρακτικά στον κώδικα: Σε αναλογία των παραπάνω, τα αντικείμενα στον αντικειμενοστραφή προγραμματισμό διαθέτουν:

1. **Δεδομένα (μεταβλητές):** Τα δεδομένα περιγράφουν την κατάσταση του κάθε αντικειμένου και μπορεί η τιμή τους να διαφέρει από αντικείμενο σε αντικείμενο.
2. **Μεθόδους (συναρτήσεις):** Οι μέθοδοι περιγράφουν την συμπεριφορά του αντικειμένου και είναι κοινές για όλα τα αντικείμενα ενός συγκεκριμένου τύπου. Για παράδειγμα όλα τα αυτοκίνητα ενός συγκεκριμένου μοντέλου διέπονται από την ίδια σχέση μεταξύ της γωνίας στροφής του τιμονιού και της γωνίας στροφής των δύο μπροστινών τροχών.

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

1. Η VBA είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού. Κάθε στοιχείο του Excel θεωρείται σαν ένα αντικείμενο (object), που έχει το δικό του σύνολο ιδιοτήτων και μεθόδων και μπορεί με τη σειρά του να περιέχει άλλα αντικείμενα.
2. Μέσω της VBA μπορούμε να διαχειριστούμε «αντικείμενα» του Excel πχ βιβλία εργασίας, φύλλα εργασίας, γραφικά, σχήματα κ.α.
3. Τα αντικείμενα διατάσσονται και προσπελούνται ιεραρχικά (Object Model). Στην κορυφή της ιεραρχίας βρίσκεται η ίδια η εφαρμογή του Excel.
4. Το Excel από μόνο του είναι ένα αντικείμενο ονόματι «application». Το αντικείμενο «application» περιέχει με την σειρά του άλλα αντικείμενα πχ το αντικείμενο «workbooks»

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

- Κάθε αντικείμενο της VBA διαθέτει το δικό του σύνολο ιδιοτήτων και μεθόδων, μέσω των οποίων το ελέγχουμε.
- Κάθε ιδιότητα (property) είναι ένα χαρακτηριστικό του αντικειμένου, όπως είναι το χρώμα ή το μέγεθός του. Για να ορίσουμε την τιμή μιας ιδιότητας, χρησιμοποιούμε μια πρόταση της μορφής

Αντικείμενο.Ιδιότητα = Τιμή

- **Παραδείγματα**

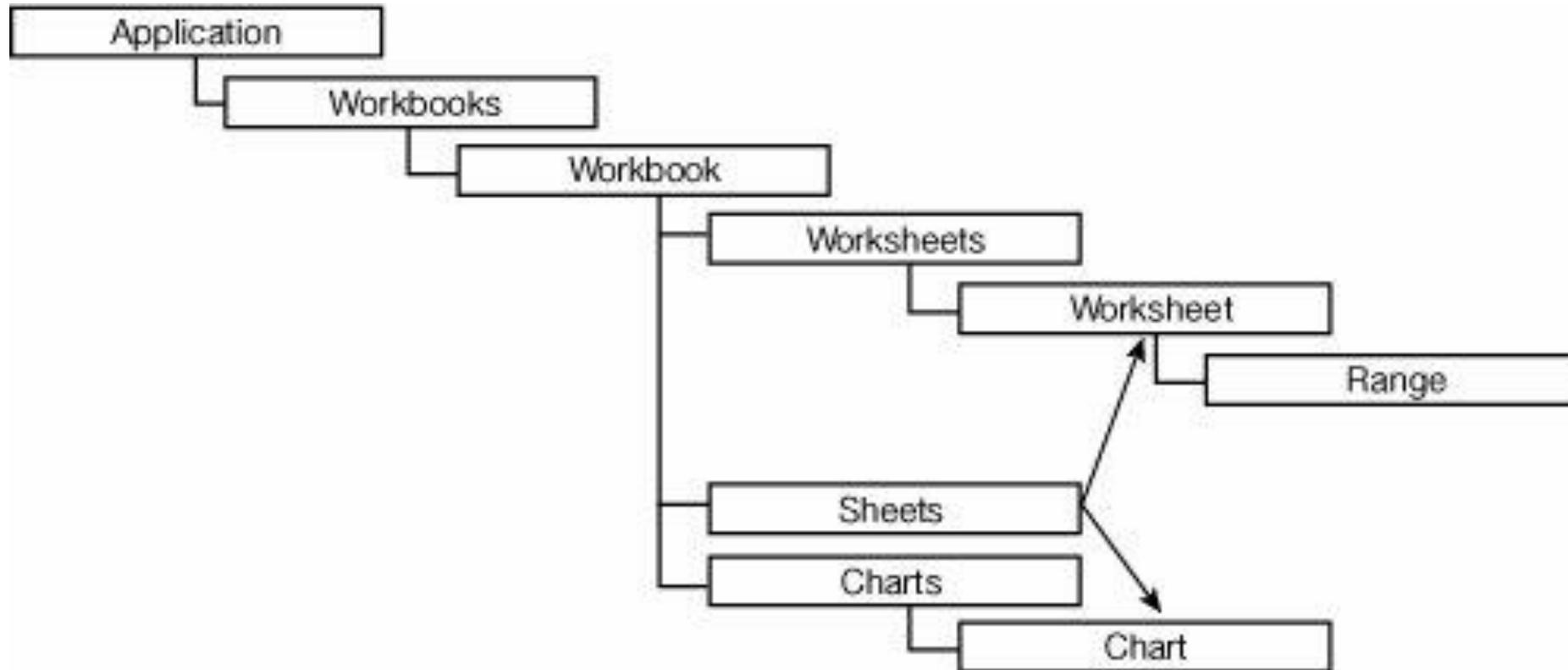
- Worksheets(“Sheet1”).Range(“A1”).Value (*Ιδιότητα*)

- Worksheets(“Sheet1”).Range(“A1”).ClearContents (*Μέθοδος*)

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

- Αναφερόμαστε σε ένα αντικείμενο χρησιμοποιώντας την σχετική του θέση στην ιεραρχία των αντικειμένων (Object Model) και τον τελεστή της τελείας.
- Παραδείγματα:
 1. `Application.Workbooks("Book1.xlsx")`
 2. `Application.Workbooks("Book1.xlsx").Worksheets("Sheet1")`
- Εάν παραλείψουμε κάποια αναφορά τότε το Excel χρησιμοποιεί το αντικείμενο «active». Έτσι αν αναφερθούμε απλά στο `Worksheets("Sheet1")` τότε εννοούμε το φύλλο εργασίας Sheet1 του βιβλίου εργασίας το οποίο την παρούσα στιγμή είναι ενεργό.

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL



VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Ένα έργο VBA περιέχει συνήθως τα αντικείμενα της εφαρμογής, όπως είναι το βιβλίο εργασίας (ThisWorkbook), τα φύλλα εργασίας (π.χ. Φύλλο1) και τυχόν φύλλα γραφημάτων.

Ένα έργο VBA μπορεί επίσης να περιέχει

1. Βασικές λειτουργικές μονάδες (Modules), οι οποίες περιέχουν κώδικα που μπορεί να χρησιμοποιηθεί από όλα τα αντικείμενα του έργου.
2. Φόρμες χρήστη (UserForm), οι οποίες είναι προσαρμοσμένα παράθυρα διαλόγου και χρησιμεύουν στην επικοινωνία μεταξύ χρήστη και της εφαρμογής,
3. Λειτουργικές μονάδες κλάσης (Class Modules), όπου ορίζονται νέα αντικείμενα.

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Sub Procedures & Function Procedures:

Η επίλυση ενός προβλήματος γίνεται πιο εύκολη εάν αυτό διαιρεθεί σε μικρότερα ανεξάρτητα τμήματα. Η επίλυση του συνολικού προβλήματος μπορεί στην συνέχεια να επιτευχθεί με διαδοχικές «κλήσεις» των ανεξάρτητων αυτών τμημάτων. Αυτά τα τμήματα, λοιπόν ονομάζονται και υποπρογράμματα. Τα υποπρογράμματα διακρίνονται σε Sub Procedures (Διαδικασίες ή Ρουτίνες) και σε Function Procedures (Συναρτήσεις).

- 1. Διαδικασία** είναι ένας τύπος υποπρογράμματος που μπορεί να εκτελέσει όλες τις λειτουργίες ενός προγράμματος. Έτσι μπορούμε να έχουμε εισαγωγή δεδομένων, να γίνουν υπολογισμοί και να τυπωθούν αποτελέσματα.
- 2. Συνάρτηση** είναι ένας τύπος υποπρογράμματος που υπολογίζει και επιστρέφει μόνο μια τιμή με το όνομα της (όπως ακριβώς και οι μαθηματικές συναρτήσεις). Η συνάρτηση επιστρέφει πάντα μία μόνο τιμή.

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Sub Procedures & Function Procedures:

- 1. Διαδικασία:** Μια διαδικασία αρχίζει με την πρόταση Sub, ακολουθούμενη από ένα όνομα και ένα ζευγάρι παρενθέσεων και τελειώνει με την πρόταση End Sub. Η σύνταξη που ακολουθούν οι διαδικασίες είναι η εξής:

*[Public][Private][Static] **Sub** [όνομα ρουτίνας]([ορίσματα])*

[Εδώ γράφουμε τον κώδικα που θέλουμε να εκτελεστεί από αυτή τη ρουτίνα]

End Sub

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Διαδικασία

Ότι βρίσκεται μέσα σε αγκύλες είναι προαιρετικό.

Μια διαδικασία μπορεί να οριστεί είτε Private (ιδιωτική) είτε Public (δημόσια).

- Οι ιδιωτικές διαδικασίες μπορούν να προσπελαστούν, να κληθούν δηλαδή μόνο από διαδικασίες της ίδιας λειτουργικής μονάδας.
- Αντίθετα, οι δημόσιες διαδικασίες μπορούν να κληθούν από παντού.
- Ο προσδιοριστής Static έχει ως συνέπεια όλες οι μεταβλητές της να παραμένουν στη μνήμη και μετά τον τερματισμό της.

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Διαδικασία

Το όνομα μιας διαδικασίας ακολουθεί τους παρακάτω κανόνες

1. Ο πρώτος χαρακτήρας πρέπει να είναι γράμμα.
2. Δεν επιτρέπεται η χρήση των χαρακτήρων ., !, @, &, \$, # καθώς και του κενού διαστήματος.
3. Το μέγιστο μήκος του ονόματος δεν πρέπει να ξεπερνά τους 255 χαρακτήρες.

Πολλές διαδικασίες δέχονται κάποιες πληροφορίες για να εκτελέσουν το έργο τους. Οι πληροφορίες αυτές φτάνουν στη διαδικασία μέσω της λίστας ορισμάτων, που είναι προαιρετική. Οι κενές παρενθέσεις δηλώνουν ότι η διαδικασία δεν δέχεται ορίσματα.

```
[Private] | [Public] | [Static] Sub όνομα ([ορίσματα])  
    [εντολές]  
[Exit Sub]  
    [εντολές]  
End Sub
```

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Διαδικασία Συμβάντων (Ειδική Περίπτωση Διαδικασίας)

- Μια διαδικασία μπορεί επίσης να κληθεί (εκτελεστεί) όταν λάβει χώρα κάποιο **συμβάν** (event). Παραδείγματα συμβάντων είναι το κλικ σ' ένα κουμπί, το άνοιγμα ενός βιβλίου
- εργασίας και η επιλογή ενός στοιχείου από μια αναπτυσσόμενη λίστα.
- Οι διαδικασίες αυτές λέγονται **διαδικασίες συμβάντων**, επειδή προϋπόθεση για να εκτελεστούν είναι να συμβεί ένα συγκεκριμένο συμβάν.
- Οι διαδικασίες συμβάντων έχουν ειδική ονομασία (το όνομα ορίζεται αυτόματα από την **VBA**) και αποθηκεύονται στη λειτουργική μονάδα κώδικα που σχετίζεται με το συγκεκριμένο αντικείμενο.
- Το όνομα μιας τέτοιας διαδικασίας αποτελείται από το όνομα του αντικειμένου το οποίο συνδέεται και με το συμβάν καθεαυτό.

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Διαδικασία Συμβάντων (Ειδική Περίπτωση Διαδικασίας)

- Παραδείγματα:

- Διαδικασία συμβάντος που εκτελείται όταν ανοίγει ένα βιβλίο εργασίας: `Workbook_Open()`
- Διαδικασία συμβάντος που εκτελείται πριν κλείσει ένα βιβλίο εργασίας: `BeforeClose()`

Για να γράψουμε κώδικα για την διαδικασία `BeforeClose()`, ακολουθούμε τα εξής βήματα:

Στο παράθυρο "Project Explorer" κάνουμε διπλό κλικ στο αντικείμενο, στο οποίο αντιστοιχεί το συμβάν και εμφανίζεται το σχετικό παράθυρο κώδικα. Για παράδειγμα, κάνουμε κλικ στο αντικείμενο `Αυτό_το_βιβλίο_εργασίας()`.

1. Από την αναδιπλούμενη λίστα αντικειμένων, που βρίσκεται στο πάνω αριστερό μέρος του παραθύρου, επιλέγουμε το αντικείμενο `Workbook`. Η VBA θα εισαγάγει αυτόματα τον ορισμό (Sub - End Sub) της διαδικασίας του **προκαθορισμένου συμβάντος**, που είναι η `Workbook_Open`.
2. Αγνοούμε τον ορισμό αυτό καθώς εμείς θέλουμε να γράψουμε κώδικα για την `BeforeClose` ή τον διαγράψουμε.
3. Από την αναδιπλούμενη λίστα των διαδικασιών, που βρίσκεται στο πάνω δεξιό μέρος του παραθύρου, επιλέγουμε το συμβάν `BeforeClose` για το οποίο θέλουμε να γράψουμε κώδικα.

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Sub Procedures & Function Procedures:

- 1. Συνάρτηση:** Μια συνάρτηση αρχίζει με την πρόταση Function, ακολουθούμενη από ένα όνομα και ένα ζευγάρι παρενθέσεων και τελειώνει με την πρόταση End Function. Η σύνταξη που ακολουθούν οι συναρτήσεις είναι η εξής:

[Public | Private | Static] Function [όνομα function]([ορίσματα]) [As τύπος_επιστροφής]

[Εδώ γράφουμε τον κώδικα που θέλουμε να εκτελεστεί από αυτή τη function]

[όνομα function] = [Αποτέλεσμα πράξεων]

End Function

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL



1. Μια οικογένεια κατανάλωσε X Kwh (κιλοβατώρες) ημερήσιου ρεύματος και Y Kwh νυχτερινού ρεύματος. Το κόστος ημερήσιου ρεύματος είναι 0.30 €. ανά Kwh και του νυχτερινού 0.15 € ανά Kwh.

Να αναπτύξετε την κατάλληλη υποδιαδικασία η οποία να υπολογίζει και να εμφανίζει το συνολικό κόστος της κατανάλωσης ρεύματος της οικογένειας .

2. Μια εταιρία αποφάσισε να πραγματοποιήσει αύξηση 12% στους μισθούς των υπαλλήλων της . Να δημιουργήσετε τις κατάλληλες υποδιαδικασίες οι οποίες:
 - I. Θα υπολογίζουν το ποσό της αύξησης που θα λάβει ο υπάλληλος.
 - II. Θα υπολογίζουν το νέο μισθό του υπαλλήλου

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

1. Μια μεταβλητή είναι μια θέση στην μνήμη του υπολογιστή και χρησιμοποιείται για να παραστήσει ένα δεδομένο στοιχείο. Σε μια μεταβλητή μπορούμε να εκχωρήσουμε μια τιμή, η οποία έχει την δυνατότητα να αλλάζει κατά τη διάρκεια εκτέλεσης του προγράμματος.
2. Τα ονόματα των μεταβλητών πρέπει να είναι περιγραφικά και να ακολουθούν τους ίδιους κανόνες ονομασίας με τις διαδικασίες.
3. ΔΕΝ μπορούμε να χρησιμοποιήσουμε για ονόματα μεταβλητών δεσμευμένες λέξεις της VBA, όπως είναι οι λέξεις-κλειδιά If, For, Next, Sub κ.λπ.
4. Για να εκχωρήσουμε μια τιμή σε μια μεταβλητή, χρησιμοποιούμε τον τελεστή ίσον (=), όπως φαίνεται και στα παραδείγματα που ακολουθούν:
 1. Price = 123
 2. VAT = 0.18

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Για να δηλώσουμε μια μεταβλητή χρησιμοποιούμε την εξής σύνταξη.

`Dim όνομα_μεταβλητής [As τύπος_δεδομένων]`

Παράδειγμα: `Dim Age As Integer, Date_of_Birth As Date`

1. Αν δηλώσουμε μια μεταβλητή χωρίς να προσδιορίσουμε τον τύπο δεδομένων της, αυτή παίρνει αυτόματα τον τύπο **Variant**.
2. Ο τύπος δεδομένων προσδιορίζει το είδος της πληροφορίας που θα αποθηκεύσουμε στη μεταβλητή.
3. Ο τύπος **Variant** είναι ο πιο ευέλικτος, επειδή μπορούμε να αποθηκεύσουμε οποιαδήποτε πληροφορία στη μεταβλητή (π.χ. αρχικά αριθμό και μετά κείμενο). Έχει όμως και το ανάλογο κόστος, γιατί δεσμεύει περισσότερη μνήμη και δημιουργεί καθυστέρηση στην επεξεργασία του.

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Εμβέλεια μεταβλητών

- Μεταβλητές που δηλώνονται με χρήση της **Dim** μέσα σε μία διαδικασία έχουν εμβέλεια μόνο εντός αυτής.
- Μεταβλητές που δηλώνονται με χρήση της **Dim** ή **Private** πριν την πρώτη διαδικασία έχουν εμβέλεια λειτουργικής μονάδας.
- Μεταβλητές που δηλώνονται με χρήση της **Public** πριν την πρώτη διαδικασία σε μία λειτουργική μονάδα έχουν εμβέλεια σε όλες τις λειτουργικές μονάδες.

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Τύπος Δεδομένων	Μέγεθος	Περιοχή Τιμών
Byte	1 byte	0 έως 255
Boolean	2 bytes	True ή False
Integer	2 bytes	-32.768 έως 32.767
Long	4 bytes	-2.147.483.648 έως 2.147.483.647
Single	4 bytes	-3,401823E38 έως -1,401298E-45 για αρνητικούς, 1,401298E-45 έως 3,401823E38 για θετικούς
Double	8 bytes	-1,79769313486231 E308 έως -4,94065645841247E-324 για αρνητικούς, 4,94065645841 247E-324 έως 1,79769313486232E308 για θετικούς
Currency	8 bytes	-922.337.203.685.477,5808 έως 922.337.203.685.477,5807
Decimal	14 bytes	+1,79.228.1 62.51 4.264.337.593.543.950.335 χωρίς δεκαδικά, +1,79228162514264337593543950335 με 28 δεκαδικά ψηφία
Date	8 bytes	1/1/100 έως 31/12/9999
Object	4 bytes	Οποιαδήποτε αναφορά αντικειμένου
String (μεταβλητού μήκους)	10 bytes + μήκος του αλφαριθμητικού	0 έως 2 δισεκατομμύρια
String (σταθερού μήκους)	Μήκος του αλφαριθμητικού	1 έως 65,400
Variant (με αριθμούς)	16 bytes	Οποιαδήποτε τιμή μέχρι το μέγεθος του Double
Variant (με χαρακτήρες)	22 bytes + μήκος αλφαριθμητικού	0 έως 2 δισεκατομμύρια
Οριζόμενος από το (User defined)	Ανάλογα με τα στοιχεία	

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Δομή With-End With

Η δομή **With-End With** επιτρέπει να εκτελούνται πολλές πράξεις σε ένα αντικείμενο, μειώνοντας τις απαιτήσεις συγγραφής κώδικα που επαναλαμβάνεται.

Παράδειγμα:

```
Sub ChangeFont()  
    Selection.Font.Bold=true  
    Selection.Font.Size=14  
    Selection.Font.Italic=true  
End Sub
```

```
Sub ChangeFont()  
    With Selection.Font  
        .Bold=true  
        .Size=14  
        .Italic=true  
    End With  
End Sub
```

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Δομές Επιλογής: Απλή Επιλογή

Οι δομές επιλογής είναι τμήματα κώδικα τα οποία εκτελούνται κάτω από συγκεκριμένες προϋποθέσεις. Ο ρόλος των συνθηκών είναι να αλλάζει η ροή του προγράμματος.

```
If Συνθήκη1=ΑΛΗΘΗΣ Then
```

```
    Ομάδα εντολών
```

```
End If
```

Η δομή αυτή εκτελεί την ομάδα εντολών εάν ισχύει η Συνθήκη1. Αν δεν ισχύει τότε το πρόγραμμα συνεχίζει την εκτέλεση των εντολών που βρίσκονται μετά το **End If**

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Δομές Επιλογής: Σύνθετη Επιλογή

```
If [Συνθήκη1=Αληθής] Then
    Ομάδα εντολών 1
Else
    Ομάδα εντολών 2
End If
```

Η δομή αυτή εκτελεί την Ομάδα εντολών 1 εάν ισχύει η Συνθήκη 1. Εάν ΔΕΝ ισχύει η συνθήκη 1 τότε εκτελεί την ομάδα εντολών 2

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Δομές Επιλογής: Πολλαπλή Επιλογή

```
If [Συνθήκη1=Αληθής] Then
    Ομάδα εντολών 1
Elseif [Συνθήκη2=Αληθής] Then
    Ομάδα εντολών 2
.....
Else
    Ομάδα εντολών Else
End If
```

Η δομή αυτή εκτελεί την ομάδα εντολών 1 εάν ισχύει η συνθήκη 1. Εάν ΔΕΝ ισχύει η συνθήκη 1 τότε εξετάζεται εάν ισχύει η συνθήκη 2 και εάν ισχύει τότε εκτελεί την ομάδα εντολών 2. Εάν ΔΕΝ ισχύει η συνθήκη 2 τότε εξετάζεται εάν ισχύει η συνθήκη N και εάν ισχύει τότε εκτελεί την ομάδα εντολών N. Εάν καμία από της N συνθήκες δεν ισχύει τότε εκτελείται η ομάδα εντολών του **Else**

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Οι δομές επανάληψης είναι τμήματα κώδικα τα οποία επαναλαμβάνονται μέχρι να ικανοποιηθεί κάποια συνθήκη. Υπάρχουν δυο βασικοί εκπρόσωποι, οι επαναλήψεις τύπου For και Do While/Until. Τα loops τύπου For είναι τμήματα κώδικα τα οποία επαναλαμβάνονται συγκεκριμένες φορές και αυτό καθορίζεται από τις τιμές "από" - "έως" που δηλώνονται κατά τη σύνταξη:

For <i>μτ=ατ</i> To <i>ττ</i> [<i>Step β</i>] Ομάδα εντολών [Exit For] Next <i>μτ</i>	<pre>sum = 0 For i = 1 To 4 sum = sum + i Next i</pre>	<pre>1^η Επαν. i=1 sum=0+1 i=1+1 2^η Επαν. i=2 sum=1+2 i=2+1 3^η Επαν. i=3 sum=3+3 i=3+1 4^η Επαν. i=4 sum=6+4 i=4+1 i=5 ->έξοδος από επανάληψη</pre>
---	--	--

Όπου: *μτ* είναι το όνομα του μετρητή, *ατ* είναι η αρχική τιμή του μετρητή, *ττ* είναι η τελική τιμή του μετρητή, *β* είναι το βήμα μεταβολής του μετρητή

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Η δομή επανάληψης Do-While Loop εκτελείται όσο η Συνθήκη Συνέχειας παραμένει ΑΛΗΘΗΣ. Μόλις η Συνθήκη Συνέχειας γίνει ΨΕΥΔΗΣ σταματά να εκτελείται η επανάληψη και συνεχίζουμε με την εκτέλεση των εντολών που βρίσκονται μετά την δομή

Do While Συνθήκη Συνέχειας Ομάδα εντολών [Exit Do] Loop	sum = 0 i=1 Do While sum<10 sum = sum + i i=i+1 Loop	1 ^η Επαν. i=1 sum=0+1 i=1+1 2 ^η Επαν. i=2 sum=1+2 i=2+1 3 ^η Επαν. i=3 sum=3+3 i=3+1 4 ^η Επαν. i=4 sum=6+4 i=4+1 sum=10 ->έξοδος από επανάληψη
Do Ομάδα εντολών Loop While Συνθήκη Συνέχειας		

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

Η δομή επανάληψης Do-Until Loop εκτελείται όσο η Συνθήκη τέλους παραμένει ΨΕΥΔΗΣ. Μόλις η Συνθήκη τέλους γίνει ΑΛΗΘΗΣ σταματά να εκτελείται η επανάληψη και συνεχίζουμε με την εκτέλεση των εντολών που βρίσκονται μετά τη δομή επανάληψης.

<p>Do Until Συνθήκη τέλους Ομάδα εντολών [Exit Do] Loop</p> <p>Do Ομάδα εντολών Loop Until Συνθήκη τέλους</p>	<pre>sum = 0 i=1 Do Until sum>=10 sum = sum + i i=i+1 Loop</pre> <p>1^η Επαν. i=1 sum=0+1 i=1+1 2^η Επαν. i=2 sum=1+2 i=2+1 3^η Επαν. i=3 sum=3+3 i=3+1 4^η Επαν. i=4 sum=6+4 i=4+1 sum=10 ->έξοδος από επανάληψη</p>
---	---

VBA (VISUAL BASIC FOR APPLICATION) & EXCEL

1. Η VBA διαθέτει μια ειδική δομή επανάληψης **For - Each**, που εφαρμόζεται σε συλλογές αντικειμένων.
2. Το χαρακτηριστικό αυτού του βρόχου είναι ότι δε χρειάζεται να γνωρίζουμε το πλήθος των στοιχείων της συλλογής. Ο βρόχος σε κάθε επανάληψη εξετάζει το επόμενο στοιχείο της συλλογής και τερματίζει μόλις εκτελέσει το μπλοκ των εντολών και για το τελευταίο στοιχείο

For Each <i>στοιχείο</i> In <i>ομάδα</i> [εντολές] [Exit For] Next [<i>στοιχείο</i>]	Dim ws As Worksheet For Each ws In ActiveWorkbook.WorkSheets MsgBox ws.Name Next ws
---	--