

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΜΕΡΟΣ ΔΕΥΤΕΡΟ	130
EXCEL ΚΑΙ VBA 2007	130
ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ	131
VBA ΣΤΟ EXCEL	131
4.1 Μακροεντολές.....	131
4.1.1 Τι είναι Μακροεντολή	131
4.1.2 Δημιουργία Μακροεντολής	131
4.1.3 Εκτέλεση Μακροεντολής.....	133
4.1.4 Αλλαγή Κώδικα σε Μακροεντολή.....	135
4.1.5 Ανάθεση Μακροεντολής σε Κουμπί	135
4.2 Τα Βασικά Στοιχεία της VBA	139
4.2.1 Επεξεργασία της VBA	139
4.2.2 Αντικείμενα, Συλλογές, Ιδιότητες και Μέθοδοι.....	143
4.2.3 Διαδικασίες.....	144
4.2.4 Τύποι Δεδομένων και Μεταβλητές.....	149
4.2.6 MsgBox - InputBox	153
4.2.7 Εντολές Επιλογής.....	156
4.2.8 Εντολές Επανάληψης (Βρόχοι).....	159
4.2.9 Πίνακες.....	162
4.3 Προσαρμοσμένες Συναρτήσεις.....	163
4.3.1 Δημιουργία Προσαρμοσμένης Συνάρτησης	163
4.3.2 Χρήση Προσαρμοσμένης Συνάρτησης	165
4.3.3 Παραδείγματα Προσαρμοσμένων Συναρτήσεων	166
4.3.4 Δημιουργία και Φόρτωση Προσθέτων.....	169

ΜΕΡΟΣ ΔΕΥΤΕΡΟ

EXCEL ΚΑΙ VBA 2007



ΚΕΦΑΛΑΙΟ ΤΕΤΑΡΤΟ

VBA ΣΤΟ EXCEL

4.1 Μακροεντολές

4.1.1 Τι είναι Μακροεντολή

Μακροεντολή (macro) είναι η καταγραφή μιας σειράς οδηγιών του Excel, οι οποίες ομαδοποιούνται και αποθηκεύονται σ' ένα ξεχωριστό φύλλο. Στην πραγματικότητα είναι εντολές της γλώσσας προγραμματισμού Visual Basic for Application (VBA). Δε χρειάζεται όμως να ξέρουμε προγραμματισμό για να δημιουργήσουμε μία μακροεντολή. Το Excel καταγράφει τις οδηγίες μας με τη σειρά και τις μεταφράζει αυτόματα. Φυσικά μπορούμε να δημιουργήσουμε μακροεντολές γράφοντας κώδικα σ' ένα ειδικό φύλλο εργασίας, που λέγεται λειτουργική μονάδα (module).

Ο σκοπός των μακροεντολών είναι η αυτοματοποίηση καθημερινών εργασιών, ώστε να κερδίζουμε χρόνο. Συχνά πρέπει να επαναλαμβάνουμε κάποιες εργασίες, όπως, για παράδειγμα, μία ειδική μορφοποίηση ενός τίτλου, εισαγωγή ίδιων δεδομένων κλπ. Αρκεί να καταγράψουμε μία φορά αυτές τις ενέργειες. Στη συνέχεια τις εκτελούμε, όποτε και όσες φορές θέλουμε.

Μπορούμε να δημιουργήσουμε κουμπιά μακροεντολών, για να κάνουμε την προσπάσιή τους εύκολη. Επίσης, έχουμε τη δυνατότητα να προσθέσουμε μία εντολή σ' ένα μενού, με την οποία θα εκτελούμε τη μακροεντολή. Ακόμα μπορούμε να αποφασίσουμε, αν θα είναι διαθέσιμες σε ορισμένα μόνο φύλλα εργασίας ή σε όλα.

Πριν ξεκινήσουμε την καταγραφή μιας μακροεντολής, σχεδιάζουμε προσεκτικά τα βήματα και τις εντολές που θέλουμε να εκτελεί. Τα λάθη κατά την καταγραφή της μακροεντολής θα καταγραφούν, όπως επίσης και οι διορθώσεις που θα κάνουμε. Ακόμα, θα πρέπει να τονίσουμε ότι το Excel καταγράφει εξ ορισμού **απόλυτες** αναφορές κελιών. Δηλαδή, αν κάνουμε κλικ σ' ένα κελί κατά τη διάρκεια της καταγραφής, κάθε φορά που την εκτελούμε θα επιλέγεται αυτό το κελί. Θα δούμε παρακάτω, στη δημιουργία μιας μακροεντολής, πως μπορούμε να καταγράψουμε σχετικές αναφορές κελιών.

4.1.2 Δημιουργία Μακροεντολής

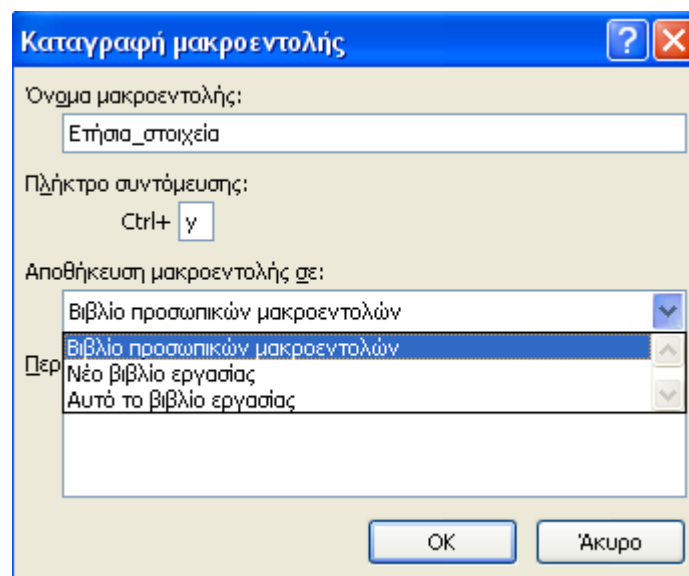
Δημιουργούμε μια μακροεντολή, η οποία θα εισάγει στο επιλεγμένο κελί το κείμενο **ΕΤΗΣΙΑ ΣΤΟΙΧΕΙΑ**, θα αυξάνει το μέγεθος των χαρακτήρων σε 12 στιγμές και θα εφαρμόζει έντονη γραφή. Στη συνέχεια θα το συγχωνεύει με το διπλανό κελί και θα στοιχίζει το κείμενο στο κέντρο θα προσθέτει περίγραμμα γύρω από το κελί και θα επιλέγει το κίτρινο για χρώμα γεμίσματος. Τέλος, θα εισάγει στα κελιά, που βρίσκονται κάτω και αριστερά από τον τίτλο, τους δώδεκα μήνες του χρόνου και θα μορφοποιεί τα διπλανά 12 κελιά με τη μορφή Νομισματική μονάδα.

Πριν ξεκινήσουμε την καταγραφή της μακροεντολής, πρέπει να εμφανίσουμε την καρτέλα **Προγραμματιστής** (αν δεν είναι ήδη ορατή):

- Κάνουμε κλικ στο **Κουμπί Office** και, στη συνέχεια, κάνουμε κλικ στο κουμπί **Επιλογές του Excel**.
- Στην κατηγορία **Δημοφιλείς**, κάτω από την ενότητα **Κύριες επιλογές του Excel**, επιλέγουμε το πλαίσιο ελέγχου **Εμφάνιση της καρτέλας "Προγραμματιστής" στην κορδέλα**.
- Κάνουμε κλικ στο κουμπί OK.

Για να αρχίσουμε την, καταγραφή της μακροεντολής, ακολουθούμε τα παρακάτω βήματα:

- Επιλέγουμε ένα οποιοδήποτε κελί.
- Ενεργοποιούμε την καρτέλα **Προγραμματιστής** και κάνουμε κλικ στο κουμπί **Καταγραφή μακροεντολής**, που βρίσκεται στην ομάδα **Κώδικας**. Τότε εμφανίζεται το παράθυρο διαλόγου "Καταγραφή μακροεντολής".



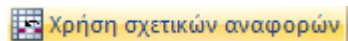
- Στο πλαίσιο κειμένου "Όνομα μακροεντολής" διαγράφουμε το προτεινόμενο όνομα Μακροεντολή1 και πληκτρολογούμε ένα δικό μας όνομα, για παράδειγμα **Ετήσια_Στοιχεία**.

Ο πρώτος χαρακτήρας του ονόματος πρέπει να είναι γράμμα. Οι υπόλοιποι χαρακτήρες μπορεί να είναι γράμματα, αριθμοί ή χαρακτήρες υπογράμμισης. Δεν επιτρέπονται τα κενά διαστήματα.

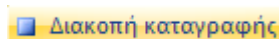
- Έχουμε τη δυνατότητα να αναθέσουμε την εκτέλεση της μακροεντολής σ' ένα συνδυασμό πλήκτρων. Στο πλαίσιο "Πλήκτρο συντόμευσης" πληκτρολογούμε το χαρακτήρα της αρεσκείας μας, για να μπορούμε να εκτελέσουμε τη μακροεντολή και με συνδυασμό πλήκτρων.
- Από τη λίστα "Αποθήκευση μακροεντολής σε" επιλέγουμε **Βιβλίο προσωπικών μακροεντολών**, ώστε η μακροεντολή να είναι διαθέσιμη όποτε χρησιμοποιούμε το

Excel. Το Excel θα δημιουργήσει ένα κρυφό βιβλίο προσωπικών μακροεντολών με όνομα **Personal.xlsb**.

- Στο πλαίσιο "Περιγραφή" πληκτρολογούμε μια περιγραφή της μακροεντολής, που ενώ είναι προαιρετικό, θα φανεί αργότερα χρήσιμη.
- Κάνουμε κλικ στο κουμπί OK.
- Πριν κάνουμε οποιαδήποτε ενέργεια, κάνουμε κλικ στο κουμπί **Χρήση σχετικών αναφορών** της ομάδας **Κώδικας**, ώστε να καταγράφονται μόνο σχετικές αναφορές κελιών. Μ' αυτό τον τρόπο η μακροεντολή θα εκτελείται σε σχέση με τη θέση του ενεργού κελιού.



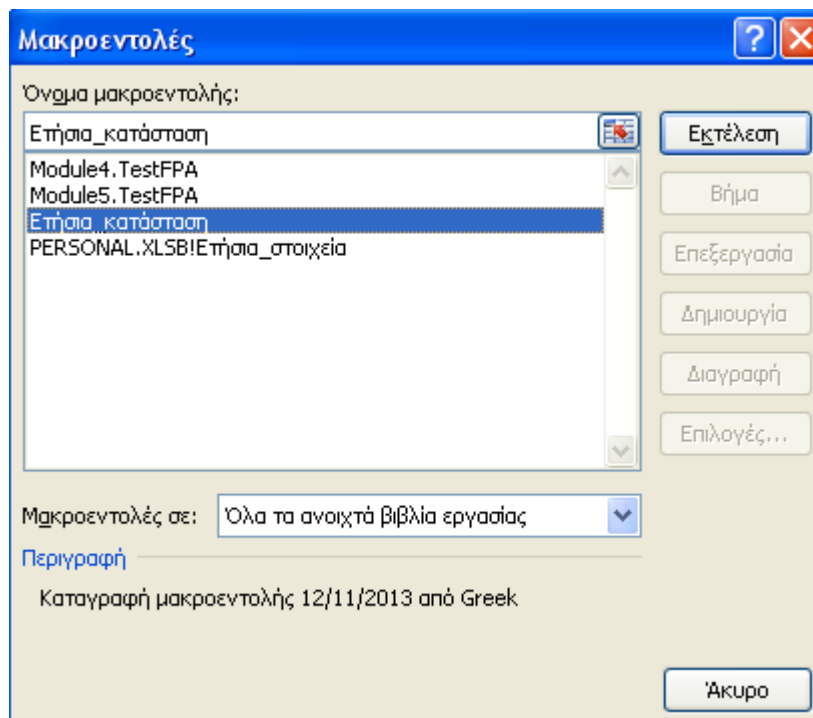
- Στο ενεργό κελί μας πληκτρολογούμε ΕΤΗΣΙΑ ΚΑΤΑΣΤΑΣΗ, αλλάζουμε το μέγεθος των χαρακτήρων σε 12 στιγμές και εφαρμόζουμε έντονη γραφή.
- Στη συνέχεια, το συγχωνεύουμε με το διπλανό κελί, στοιχίζοντας το κείμενο στο κέντρο του ενιαίου κελιού.
- Προσθέτουμε περίγραμμα γύρω από το κελί και διαλέγουμε χρώμα γεμίσματος το κίτρινο.
- Στο κελί που βρίσκεται κάτω και αριστερά πληκτρολογούμε Ιανουάριος και σύρουμε τη λαβή συμπλήρωσης μέχρι το μήνα Δεκέμβριο.
- Αυξάνουμε το πλάτος της στήλης των μηνών.
- Επιλέγουμε τα δώδεκα κελιά που βρίσκονται δεξιά των μηνών και τα μορφοποιούμε με Νομισματική μονάδα.
- Κάνουμε κλικ στο κελί που βρίσκεται δεξιά από τον Ιανουάριο.
- Σταματάμε την καταγραφή, κάνοντας κλικ στο κουμπί **Διακοπή καταγραφής** της ομάδας **Κώδικας** ή στο ομώνυμο κουμπί που βρίσκεται στη γραμμή κατάστασης.



- ☞ Όταν τερματίσουμε την εφαρμογή, το Exce! θα μας ρωτήσει αν θέλουμε να αποθηκεύσουμε τις αλλαγές που κάναμε στο "Βιβλίο εργασίας προσωπικών μακροεντολών". Κάνουμε κλικ στο κουμπί **Ναι**.

4.1.3 Εκτέλεση Μακροεντολής

Όλες οι αποθηκευμένες μακροεντολές μπορούν να εκτελεστούν από το παράθυρο διαλόγου "Μακροεντολές", που εμφανίζεται αν κάνουμε κλικ στο κουμπί **Μακροεντολές** της ομάδας **Κώδικας** ή πατήσουμε το συνδυασμό πλήκτρων **[Alt] + [F8]**.



Επιλέγουμε το όνομα της μακροεντολής που θέλουμε να εκτελέσουμε και κάνουμε κλικ στο κουμπί **Εκτέλεση**. Εναλλακτικά μπορούμε να κάνουμε διπλό κλικ πάνω στο όνομά της. Το φύλλο εργασίας θα πάρει την παρακάτω μορφή:

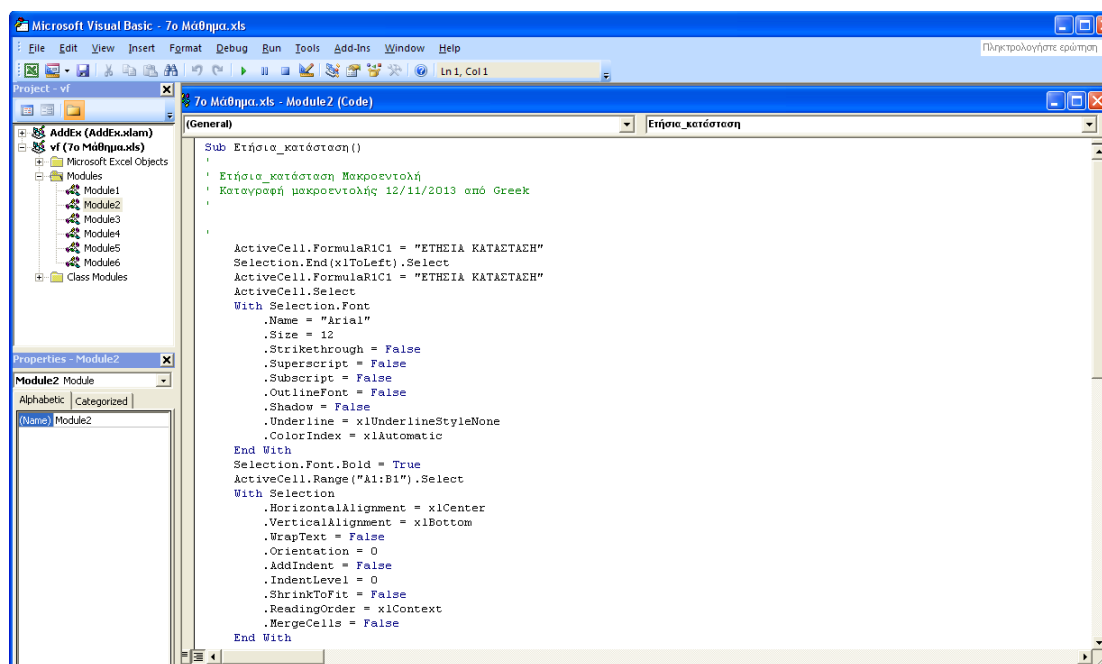
	A	B	C	D	E
1	ΕΤΗΣΙΑ ΚΑΤΑΣΤΑΣΗ				
2	Ιανουάριος				
3	Φεβρουάριος				
4	Μάρτιος				
5	Απρίλιος				
6	Μάιος				
7	Ιούνιος				
8	Ιούλιος				
9	Αύγουστος				
10	Σεπτέμβριος				
11	Οκτώβριος				
12	Νοέμβριος				
13	Δεκέμβριος				
14					

Διαγράφουμε μια μακροεντολή, αν την επιλέξουμε στο παράθυρο διαλόγου "Μακροεντολές" και πατήσουμε το κουμπί **Διαγραφή**.

4.1.4 Αλλαγή Κώδικα σε Μακροεντολή

Το κουμπί **Επεξεργασία**, του παραθύρου διαλόγου "Μακροεντολές", ανοίγει την επιλεγμένη μακροεντολή στην **Επεξεργασία Visual Basic**, όπου μπορούμε να δούμε και να επεξεργαστούμε τον κώδικά της.

καθώς το Excel κατέγραφε τις ενέργειές μας, ταυτόχρονα τις μετέτρεπε σε κώδικα της Visual Basic. Ο κώδικας αυτός φαίνεται στο κυρίως παράθυρο της εφαρμογής "Module1" (Λειτουργική μονάδα1), που λειτουργεί σαν ένας επεξεργαστής κειμένου. Οι λέξεις **Sub** και **End Sub** οριοθετούν τις εντολές της μακροεντολής.



Αν, για παράδειγμα, θέλουμε να αλλάξουμε το μέγεθος της γραμματοσειράς από 12 στιγμές σε 13 και να καταργήσουμε την έντονη γραφή, δε χρειάζεται να καταγράψουμε τη μακροεντολή από την αρχή. Αρκεί να αλλάξουμε την εντολή **.Size = 12** σε **.Size = 13** και να κάνουμε σχόλιο την εντολή **Selection.Font.Bold = True**, εισάγοντας μια (') απόστροφο πριν από την εντολή. Δηλαδή, η εντολή γίνεται **'Selection.Font.Bold = True** (με πράσινο χρώμα).

Για να γίνουν οι αλλαγές μόνιμες, πρέπει να αποθηκεύσουμε τη μακροεντολή κάνοντας κλικ στο κουμπί **Αποθήκευση** (Save) της γραμμής εργαλείων ή στην ομώνυμη εντολή από το μενού **Αρχείο** (File). Με το συνδυασμό πλήκτρων **[Alt] + [Q]** κλείνουμε την Επεξεργασία Visual Basic και επιστρέφουμε στο Microsoft Excel.

4.1.5 Ανάθεση Μακροεντολής σε Κουμπί

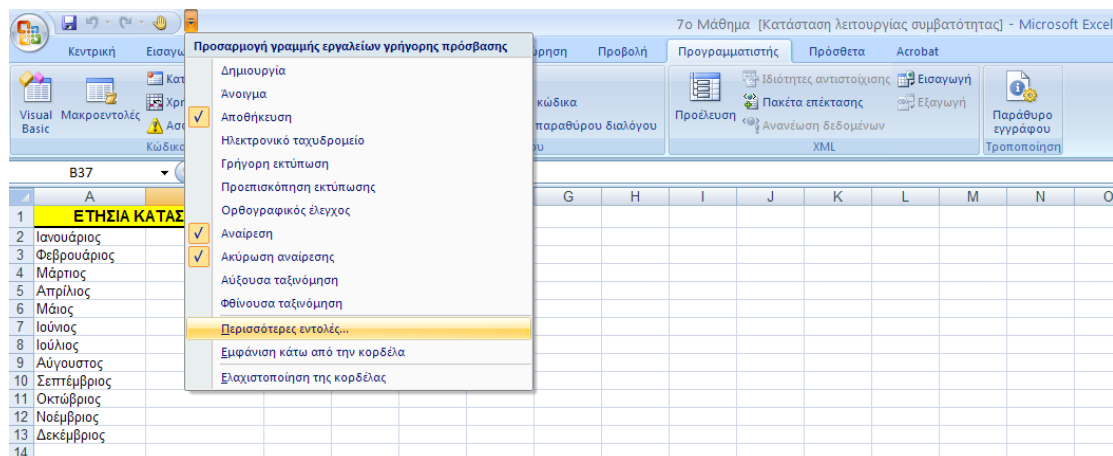
Κάποιες από τις μακροεντολές που δημιουργήσαμε, σίγουρα θα τις χρησιμοποιούμε συχνά. Για να τις εκτελούμε γρήγορα, μπορούμε να τις αναθέσουμε σε ένα κουμπί της **Γραμμής εργαλείων γρήγορης πρόσβασης**.

Η γραμμή εργαλείων γρήγορης πρόσβασης είναι μια προσαρμοζόμενη γραμμή εργαλείων, η οποία περιέχει εντολές που είναι ανεξάρτητες από την τρέχουσα καρτέλα.

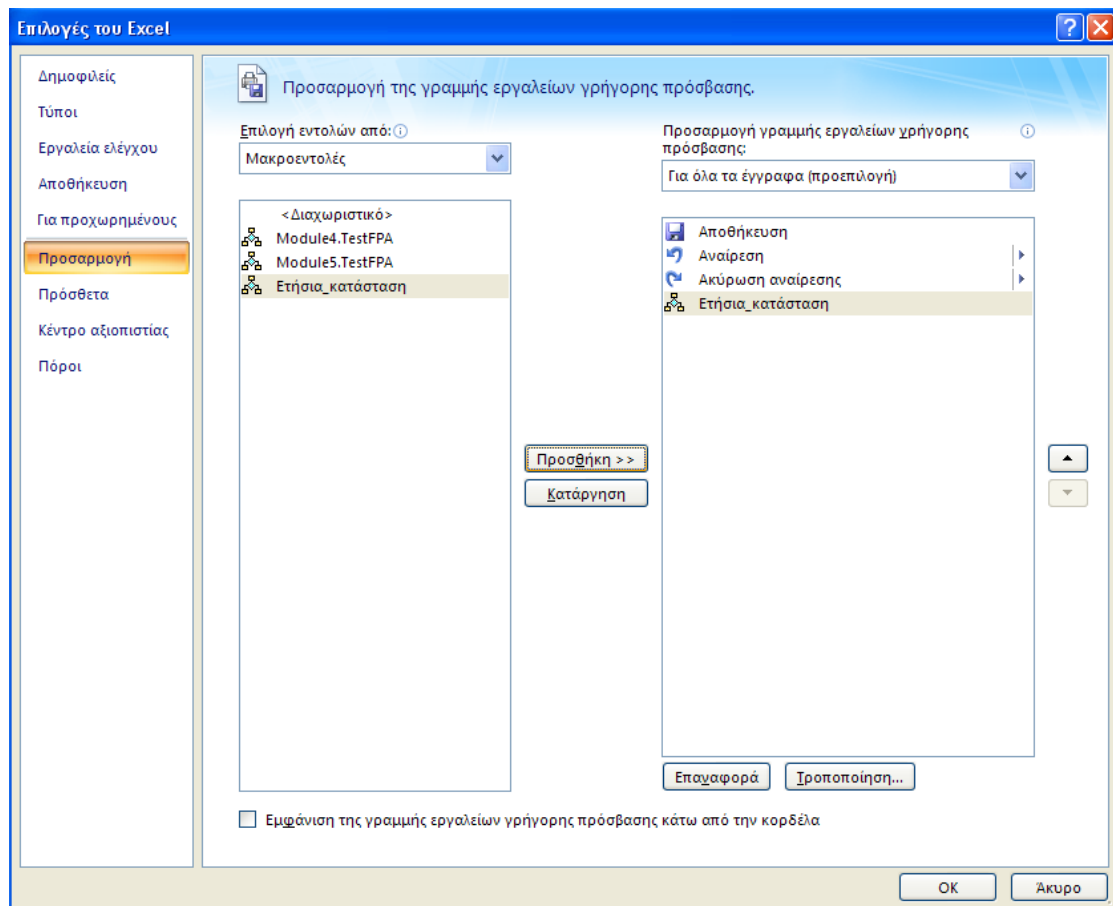
Μπορούμε να προσθέσουμε ή να αφαιρέσουμε κουμπιά εντολών του Excel αλλά και δικών μας μακροεντολών.

Η δημιουργία ενός κουμπιού στη γραμμή εργαλείων γρήγορης πρόσβασης και η ανάθεση σ' αυτό μιας μακροεντολής γίνεται με τα παρακάτω βήματα:

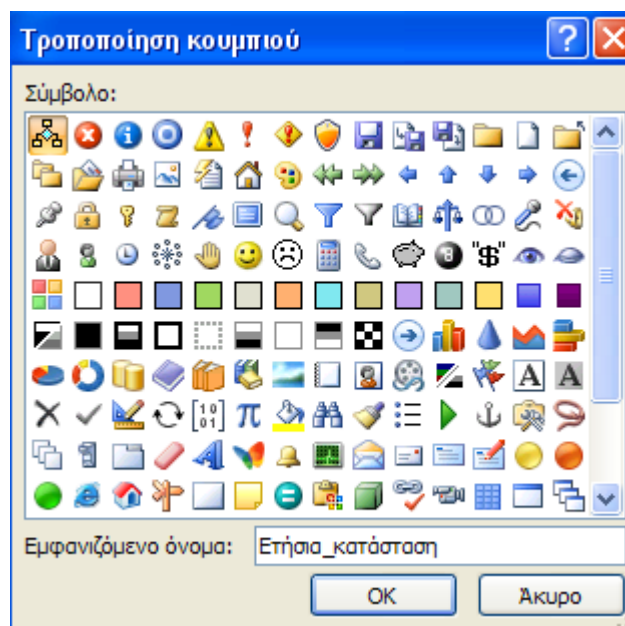
- Κάνουμε κλικ στην επιλογή **Προσαρμογή της γραμμής εργαλείων γρήγορης πρόσβασης**.
- Στη λίστα που εμφανίζεται, κάνουμε κλικ στην επιλογή **Περισσότερες εντολές**.



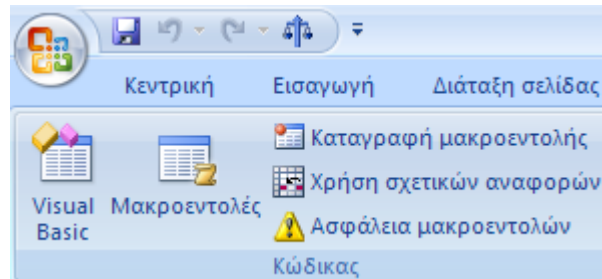
- Εμφανίζεται το παράθυρο διαλόγου "Επιλογές του Excel" με ενεργοποιημένη την κατηγορία **Προσαρμογή**.
- Στην αναδιπλούμενη λίστα **Επιλογή εντολών από** κάνουμε κλικ στην κατηγορία **Μακροεντολές**.



- Αν η εικόνα του κουμπιού που προσθέσαμε στη γραμμή εργαλείων δεν μας ικανοποιεί, κάνουμε κλικ στο κουμπί **Τροποποίηση**.
- Εμφανίζεται το παράθυρο διαλόγου "Τροποποίηση κουμπιού", απ' όπου επιλέγουμε ένα νέο εικονίδιο.



- Αν θέλουμε, μπορούμε να αλλάξουμε και την επεξήγηση πληκτρολογώντας στο πλαίσιο κειμένου "Εμφανιζόμενο όνομα" μια διαφορετική επεξήγηση.
- Κάνουμε κλικ στο κουμπί **OK**, για να επιβεβαιώσουμε τις επιλογές μας και να επιστρέψουμε πάλι στο παράθυρο διαλόγου "Επιλογές του Excel", όπου κάνουμε πάλι κλικ στο κουμπί **OK**.



- ☞ Για να αφαιρέσουμε ένα κουμπί από τη γραμμή εργαλείων γρήγορης πρόσβασης, κάνουμε δεξί κλικ πάνω στο κουμπί και από το μενού που εμφανίζεται, επιλέγουμε την εντολή Κατάργηση από τη γραμμή εργαλείων γρήγορης πρόσβασης.

4.2 Τα Βασικά Στοιχεία της VBA

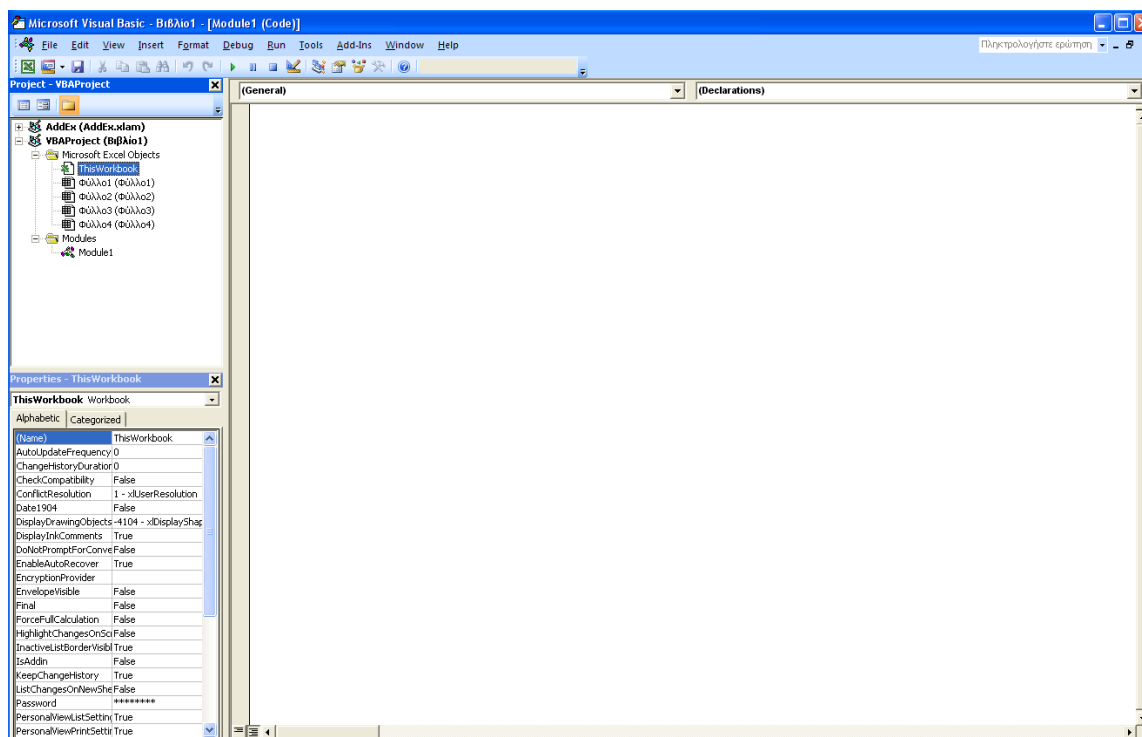
Στην προηγούμενη ενότητα μάθαμε πώς να καταγράψουμε μια απλή μακροεντολή. Αν και με τις μακροεντολές μπορούμε να αυτοματοποιήσουμε σε μεγάλο βαθμό τις καθημερινές μας εργασίες, θα ανακαλύψουμε σύντομα ότι πολλές φορές θα χρειαστεί να γράψουμε κώδικα προγράμματος, για να επεκτείνουμε τις δυνατότητές τους.

Αν και υπάρχουν στην αγορά χιλιάδες προγράμματα που εκτελούν τους πιο σύνθετους υπολογισμούς, θα δυσκολευτούμε πολύ να βρούμε το πρόγραμμα που κάνει ακριβώς αυτό που χρειαζόμαστε. Εδώ έρχεται να μας βοηθήσει η Microsoft, η οποία μας παρέχει για τις εφαρμογές του Office μία έκδοση της γλώσσας προγραμματισμού Visual Basic, τη **VBA (Visual Basic for Applications)**. Η VBA είναι μια γλώσσα προγραμματισμού φιλική προς το χρήστη, που μαθαίνεται σχετικά εύκολα. Όπως όλες οι γλώσσες προγραμματισμού απαιτεί πολύ πρακτική εξάσκηση για την εκμάθησή της.

Το Microsoft Office Excel είναι ένα ισχυρό εργαλείο υπολογισμών και ανάλυσης πληροφοριών. Από τη στιγμή που ενσωματώθηκε η VBA στην εφαρμογή, οι δυνατότητες που μας παρέχει είναι σχεδόν απεριόριστες. Μπορούμε να προγραμματίσουμε από απλές συναρτήσεις μέχρι ολοκληρωμένες εφαρμογές λογιστικής διαχείρισης.

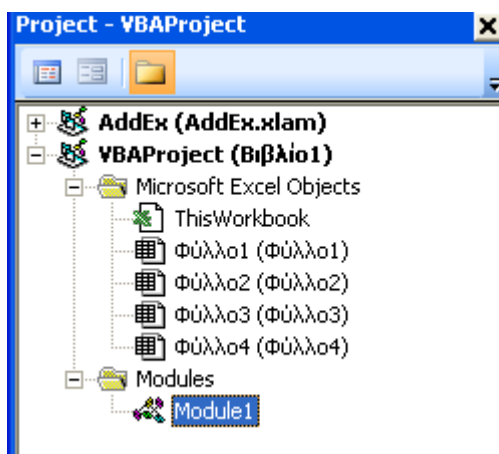
4.2.1 Επεξεργασία της VBA

Για να γράψουμε κώδικα σε VBA για ένα βιβλίο εργασίας, πρέπει να χρησιμοποιήσουμε την Επεξεργασία της Visual Basic, μια εφαρμογή ανεξάρτητη που εκτελείται μέσα από το Excel. Για να ανοίξουμε την **Επεξεργασία της Visual Basic**, κάνουμε κλικ στο κουμπί **Visual Basic** της ομάδας **Κώδικας**, στην καρτέλα **Προγραμματιστής** ή για συντομία πατάμε το συνδυασμό πλήκτρων **[Alt] + [F11]**.



Η παραπάνω εικόνα δείχνει τον επεξεργαστή της VBA. Όπως οι περισσότερες εφαρμογές των Windows, διαθέτει γραμμή τίτλου, γραμμή μενού και βασική (standard) γραμμή εργαλείων.

Στο αριστερό μέρος του VBE (Visual Basic Editor) διακρίνουμε το παράθυρο **Project Explorer**, που εμφανίζει σε δένδροειδή μορφή το έργο vBA για κάθε ανοιχτό βιβλίο εργασίας. Αν το παράθυρο δεν είναι ορατό, επιλέγουμε την εντολή **Project Explorer** από το μενού **View** ή πατάμε το συνδυασμό πλήκτρων **[Ctrl]+[R]**.



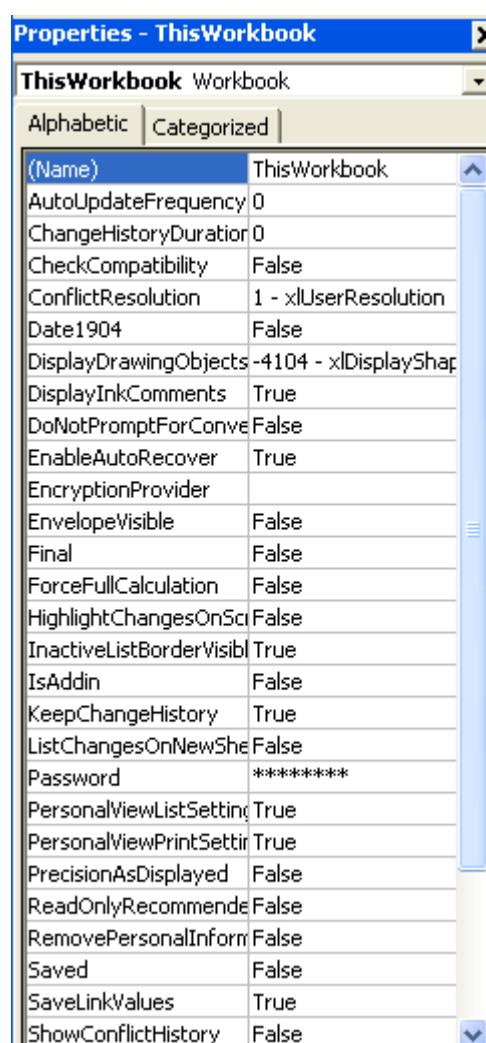
Ένα έργο VBA περιέχει συνήθως τα αντικείμενα της εφαρμογής, όπως είναι το βιβλίο εργασίας (ThisWorkbook), τα φύλλα εργασίας (π.χ. Φύλλο1) και τυχόν φύλλα γραφημάτων.

Ένα έργο VBA μπορεί επίσης να περιέχει βασικές λειτουργικές μονάδες (**Modules**), οι οποίες περιέχουν κώδικα που μπορεί να χρησιμοποιηθεί από όλα τα αντικείμενα του έργου. Επίσης μπορεί να περιέχει φόρμες χρήστη (**UserForm**), οι οποίες είναι προσαρμοσμένα παράθυρα διαλόγου και χρησιμεύουν στην επικοινωνία μεταξύ χρήστη

και της εφαρμογής, καθώς και λειτουργικές μονάδες κλάσης (**Class Modules**), όπου ορίζονται νέα αντικείμενα.

Αρχικά ένα έργο VBA περιέχει μόνο τα αντικείμενα της εφαρμογής. Για να προσθέσουμε στο έργο, για παράδειγμα, μία βασική λειτουργική μονάδα, κάνουμε δεξί κλικ στο όνομα του έργου στο παράθυρο "Project Explorer". Στο μενού συντόμευσης που εμφανίζεται, δείχνουμε με το δείκτη του ποντικιού στην εντολή **Insert** και, στη συνέχεια, κάνουμε κλικ στην επιλογή **Module**. Εναλλακτικά μπορούμε να επιλέξουμε την ίδια εντολή από το μενού **Insert**.

Κάτω από το παράθυρο "Project Explorer" διακρίνουμε το παράθυρο ιδιοτήτων (**Properties**). Εάν το παράθυρο δεν είναι ορατό, πατάμε το πλήκτρο **[F4]** για να το εμφανίσουμε (**View → Properties Window**).



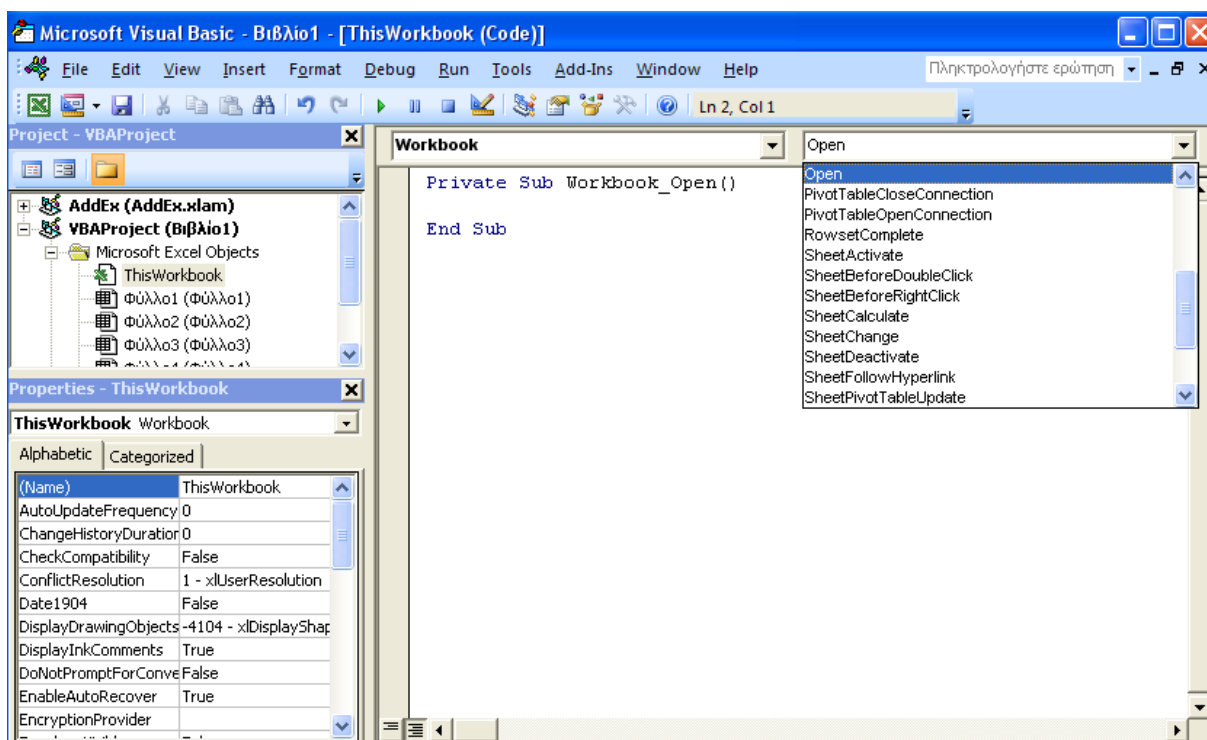
Το παράθυρο αυτό το χρησιμοποιούμε για να δούμε ή να τροποποιήσουμε τις ιδιότητες του επιλεγμένου αντικειμένου.

Κάθε αντικείμενο διαθέτει τη δική του λειτουργική μονάδα κώδικα, που περιέχει τον σχετικό κώδικα VBA. Για να εμφανίσουμε το παράθυρο κώδικα ενός αντικειμένου, κάνουμε διπλό κλικ στο αντικείμενο στο παράθυρο "Project Explorer".

Η παρακάτω εικόνα δείχνει ένα παράθυρο κώδικα βιβλίου εργασίας.



Όλα τα παράθυρα κώδικα έχουν τα ίδια χαρακτηριστικά και διαθέτουν στην κορυφή τις δύο αναπτυσσόμενες λίστες. Η αριστερή λίστα περιέχει το αντικείμενο που σχετίζεται με τη συγκεκριμένη λειτουργική μονάδα, ενώ η δεξιά περιέχει τις διαδικασίες που σχετίζονται με το αντικείμενο αυτό. Για παράδειγμα, αν ανοίξουμε το παράθυρο κώδικα του βιβλίου εργασίας, στην αριστερή λίστα θα υπάρχει το αντικείμενο Workbook και, στη δεξιά λίστα βρίσκονται όλες οι διαδικασίες του.



Η εισαγωγή κώδικα VBA γίνεται με καταγραφή μιας μακροεντολής, όπως είδαμε στο προηγούμενο κεφάλαιο, είτε με απευθείας πληκτρολόγηση στο παράθυρο κώδικα. Μπορούμε, επίσης, να αντιγράψουμε κώδικα από ένα παράθυρο σ' ένα άλλο, χρησιμοποιώντας τις γνωστές λειτουργίες αντιγραφής και επικόλλησης.

Στη βασική γραμμή εργαλείων θα βρούμε τις εντολές Run  και Reset  με τις οποίες ξεκινάμε την εκτέλεση μιας διαδικασίας ή τερματίζουμε τη λειτουργία της.

4.2.2 Αντικείμενα, Συλλογές, Ιδιότητες και Μέθοδοι

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. Για παράδειγμα, μία φόρμα χρήστη αποτελεί ένα αντικείμενο και μπορεί να περιέχει άλλα αντικείμενα, όπως είναι ένα κουμπί διαταγής και ένα πλαίσιο κειμένου. Στην κορυφή της ιεραρχίας του μοντέλου των αντικειμένων βρίσκεται η ίδια η εφαρμογή του Excel (**Application**), που περιέχει όλα τα άλλα αντικείμενα.

Αναφερόμαστε σ' ένα αντικείμενο με βάση τη θέση του στην ιεραρχία χρησιμοποιώντας την **τελεία** σαν διαχωριστή. Για παράδειγμα, για να αναφερθούμε στο ενεργό βιβλίο εργασίας, χρησιμοποιούμε την έκφραση.

Application.ActiveWorkbook

Πολλές φορές είναι βολικό να χρησιμοποιήσουμε **συλλογές (collection)** αντικειμένων, δηλαδή ομάδες όμοιων αντικειμένων. Για παράδειγμα, η συλλογή όλων των φύλλων εργασίας ενός βιβλίου εργασίας είναι η **Worksheets()**.

Αν θέλουμε να αναφερθούμε στο πρώτο φύλλο εργασίας, χρησιμοποιούμε την έκφραση.

Worksheets(1)

ή την ισοδύναμη

Worksheets ("Φύλλο1")

εφόσον ονομάζεται *Φύλλο 1*.

Σε αρκετές περιπτώσεις είναι αναγκαίο να χρησιμοποιήσουμε την πλήρη ιεραρχία του μοντέλου αντικειμένων. Για παράδειγμα, αν έχουμε ανοιχτά δύο βιβλία εργασίας που έχουν και τα δύο, ένα φύλλο εργασίας με το ίδιο όνομα *Φύλλο1*, θα πρέπει πριν από το όνομα του φύλλου εργασίας να αναφερθούμε στο όνομα του βιβλίου εργασίας του, με μια τελεία ανάμεσά τους

Workbooks (1).Worksheets ("Φύλλο1")

ή ισοδύναμα αν το βιβλίο εργασίας ονομάζεται *Βιβλίο1*:

Workbooks("Βιβλίο1.Xls").Worksheets("Φύλλο1")

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**. Για παράδειγμα, αν θέλουμε να αλλάξουμε το όνομα του ενεργού φύλλου εργασίας, θα χρησιμοποιήσουμε την πρόταση:

```
ActiveSheet.Name = "Προϋπολογισμός 2005"
```

Αντίστοιχα, η εντολή για να εκχωρήσουμε το όνομα του ενεργού φύλλου εργασίας σε μια μεταβλητή που δηλώθηκε με όνομα `sheetName` είναι:

```
sheetName = ActiveSheet.Name
```

Μέθοδος (method) είναι μια ενέργεια που μπορεί να εκτελέσει ένα αντικείμενο. Για παράδειγμα, η επόμενη εντολή κλείνει το τρέχον βιβλίο εργασίας

```
Workbooks.Close
```

Οι περισσότερες μέθοδοι χρειάζονται επιπλέον πληροφορίες για να εκτελέσουν το σκοπό τους. Οι πληροφορίες αυτές ονομάζονται **ορίσματα**. Για να εκτυπώσουμε το ενεργό φύλλο εργασίας στον προκαθορισμένο εκτυπωτή του συστήματος, αρκεί να χρησιμοποιήσουμε τη μέθοδο:

Η μέθοδος **PrintOut** διαθέτει οκτώ προαιρετικά ορίσματα. Τα δύο πρώτα ορίσματα καθορίζουν την αρχική και την τελική σελίδα εκτύπωσης. Έτσι για να εκτυπώσουμε τις τρεις πρώτες σελίδες του ενεργού φύλλου εργασίας, μπορούμε να χρησιμοποιήσουμε την πρόταση:

```
ActiveSheet.PrintOut 1,3
```

Ένας άλλος τρόπος κλήσης της ίδιας εντολής είναι η χρήση ονομαστικών ορισμάτων:

```
ActiveSheet.PrintOut From=1 To=3
```

Η τρόπος αυτός είναι πιο κατανοητός και δεν απαιτεί τη χρήση του χαρακτήρα κόμμα (,) για να δείξουμε την παράλειψη κάποιου ορίσματος, όπως φαίνεται στην εντολή που ακολουθεί:

```
ActiveSheet.PrintOut,,5
```

Εδώ θέλουμε να εκτυπώσουμε όλο το ενεργό φύλλο εργασίας σε πέντε αντίτυπα. Η παράλειψη των δύο πρώτων ορισμάτων απαιτεί τη χρήση δύο χαρακτήρων κόμμα πριν το τρίτο όρισμα. Η χρήση ονομαστικών ορισμάτων κάνει σίγουρα την εντολή περισσότερο κατανοητή:

```
ActiveSheet.PrintOut Copies:=5
```

4.2.3 Διαδικασίες

Μια διαδικασία είναι ένα σύνολο εντολών της VBA, που εκτελούν μια συγκεκριμένη λειτουργία και χρησιμεύει στην οργάνωση του κώδικα σύμφωνα με τις αρχές του δομημένου προγραμματισμού. Μια διαδικασία βρίσκεται μέσα σε μια λειτουργική μονάδα κώδικα και όταν καλείται, εκτελεί με τη σειρά όλες τις εντολές που περιέχει, από την πρώτη έως την τελευταία (εκτός εάν ο προγραμματιστής έχει προβλέψει διαφορετικά). Μπορεί να περιέχει χιλιάδες εντολές, συνήθως όμως αποφεύγουμε το μεγάλο κώδικα και προτιμάμε να *"σπάμε"* τον κώδικα σε μικρότερες διαδικασίες για την ευκολότερη συντήρησή του.

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:

[Private] | [Public] | [Static] Sub όνομα ([ορίσματα])

[εντολές]

[Exit Sub]

[εντολές]

End Sub

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.

Το όνομα μιας διαδικασίας ακολουθεί τους παρακάτω κανόνες

- Ο πρώτος χαρακτήρας πρέπει να είναι γράμμα.
- Δεν επιτρέπεται η χρήση των χαρακτήρων **., !, @, &, \$, #** καθώς και του κενού διαστήματος.
- 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.

Η πρόταση **Exit Sub** έχει ως συνέπεια τον άμεσο τερματισμό της, χωρίς δηλαδή την εκτέλεση των υπόλοιπων εντολών που ακολουθούν.

Στη συνέχεια, ακολουθούν δύο παραδείγματα δημοσίων (Public) διαδικασιών, η πρώτη είναι χωρίς ορίσματα και η δεύτερη δέχεται ένα όρισμα.

```
Public Sub LikeExcel()
```

```
    MsgBox "Μου αρέσει το Excel"
```

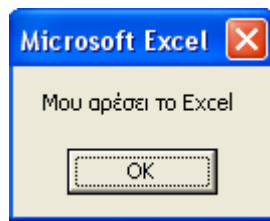
```
End Sub
```

```
Public Sub Hello(user)
```

```
    MsgBox "Γειά σας" & user
```

```
End Sub
```

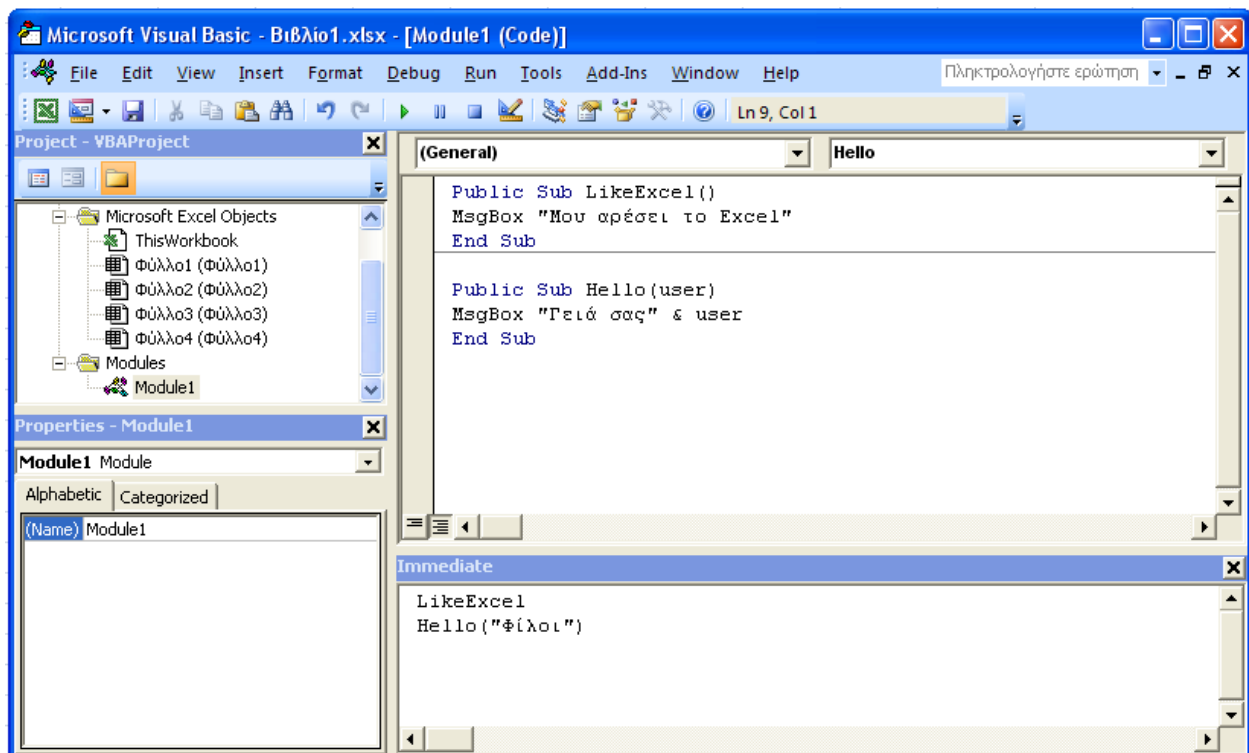
Η πρώτη με όνομα **LikeExcel**, καλεί τη συνάρτηση MsgBox της Visual Basic, η οποία και εμφανίζει ένα μήνυμα σ' ένα παράθυρο διαλόγου.



Η δεύτερη διαδικασία, με όνομα **Hello**, δέχεται μια τιμή, την οποία και εμφανίζει με την κλήση της συνάρτησης MsgBox.

Για να εκτελέσουμε τις δύο αυτές διαδικασίες, εισάγουμε στο έργο μία βασική λειτουργική μονάδα κώδικα (Module) και πληκτρολογούμε τον παραπάνω κώδικα. Στη συνέχεια, εμφανίζουμε το παράθυρο "Immediate" (**View → Immediate Window**), πληκτρολογούμε το όνομα της διαδικασίας και πατάμε το πλήκτρο **[Enter]**.

Το παράθυρο Immediate χρησιμεύει στην άμεση εκτέλεση εντολών για την αποσφαλμάτωση (ανίχνευση λαθών) του κώδικά μας.



Το μήνυμα όπως είναι φυσικό θα εμφανιστεί στο φύλλο εργασίας. Στην κλήση της δεύτερης διαδικασίας, μετά το όνομά της αφήνουμε ένα κενό διάστημα, πληκτρολογούμε την τιμή που θέλουμε να περάσουμε στη διαδικασία και μετά πατάμε το πλήκτρο **[Enter]**. Το αποτέλεσμα είναι ένα μήνυμα με ένα παράθυρο διαλόγου, όπως αυτό που φαίνεται στην εικόνα που ακολουθεί:



Πολλές φορές καλούμε μια διαδικασία μέσα από κάποια άλλη διαδικασία. Η κλήση αυτή γίνεται με δύο τρόπους:

- Πληκτρολογούμε στο σημείο που θέλουμε να την καλέσουμε το όνομά της και αν διαθέτει ορίσματα, αφήνουμε ένα κενό διάστημα και, στη συνέχεια, εισάγουμε και τα ορίσματά της χωρισμένα με κόμμα.
- Πριν το όνομά της, πληκτρολογούμε τη λέξη-κλειδί **Call** και στη συνέχεια, μέσα σε ένα ζευγάρι παρενθέσεων, τα ορίσματά της διαχωρισμένα με κόμμα.

Αν θέλουμε να καλέσουμε τις δύο αυτές διαδικασίες μέσα από κάποια άλλη, για παράδειγμα την **Test**, θα γράψουμε τον παρακάτω κώδικα:

```
Sub Test ()  
LikeExcel  
Hello " Φίλοι"  
End Sub
```

```
ή ισοδύναμα  
Sub Test ()  
    Call LikeExcel  
    Call Hello " Φίλοι"  
End Sub
```

Μια διαδικασία μπορεί να εκτελεστεί με κλικ σ' ένα κουμπί γραμμής εργαλείων ή με επιλογή μιας εντολής ενός μενού, όπως ακριβώς και μια μακροεντολή. Άλλωστε οι διαδικασίες είναι μακροεντολές που δημιουργούνται χωρίς καταγραφή, αλλά με προγραμματισμό. Μια διαδικασία μπορεί επίσης να εκτελεστεί όταν λάβει χώρα κάποιο συμβάν (**event**).

Παραδείγματα συμβάντων είναι το κλικ σ' ένα κουμπί διαταγών, 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.

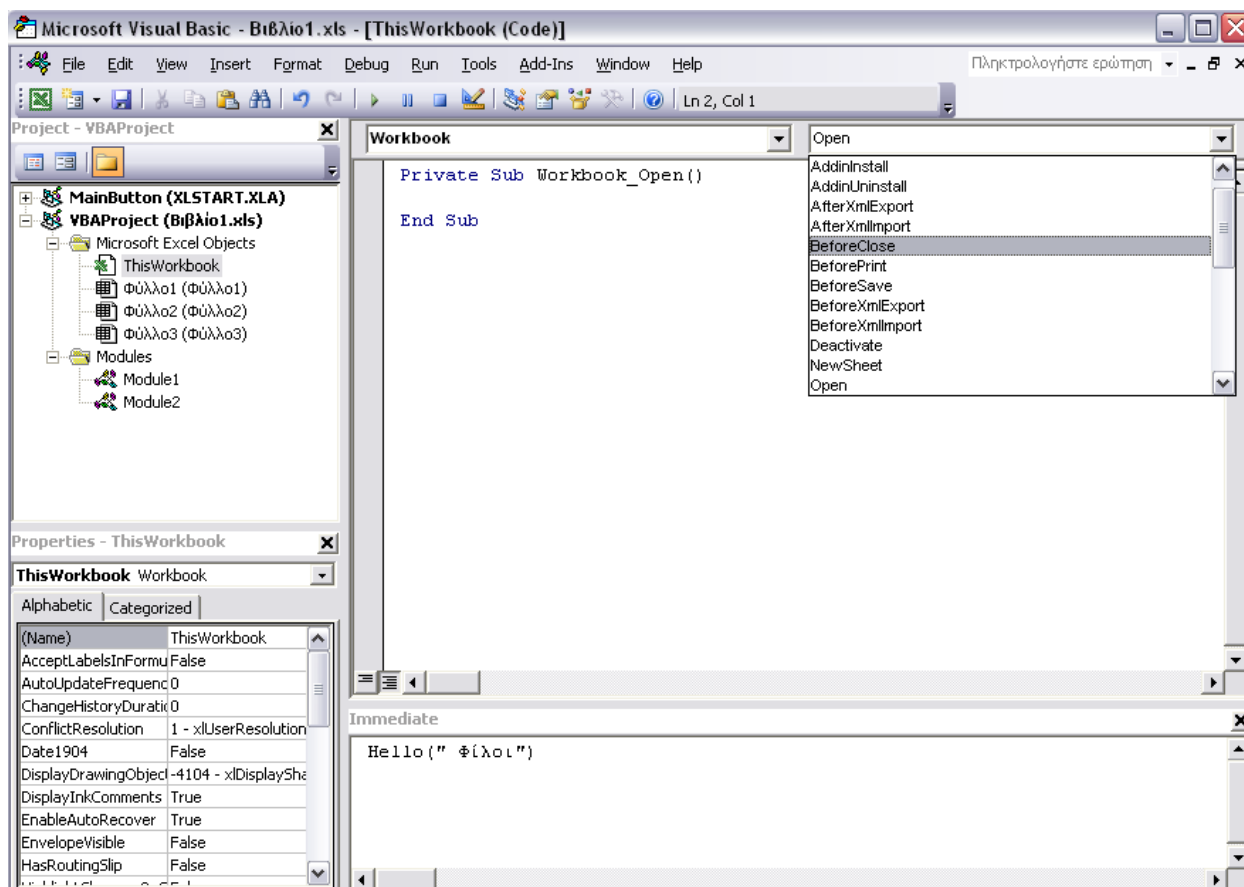
Οι διαδικασίες συμβάντων έχουν ειδική ονομασία (το όνομα ορίζεται αυτόματα από την VBA) και αποθηκεύονται στη λειτουργική μονάδα κώδικα που σχετίζεται με το συγκεκριμένο αντικείμενο. Το όνομα μιας τέτοιας διαδικασίας αποτελείται από το όνομα του αντικειμένου που συνδέεται με το συμβάν, το χαρακτήρα υπογράμμισης και το όνομα του συμβάντος. Έτσι, η διαδικασία συμβάντος που εκτελείται όταν ανοίγει ένα βιβλίο εργασίας είναι η:

Workbook_Open

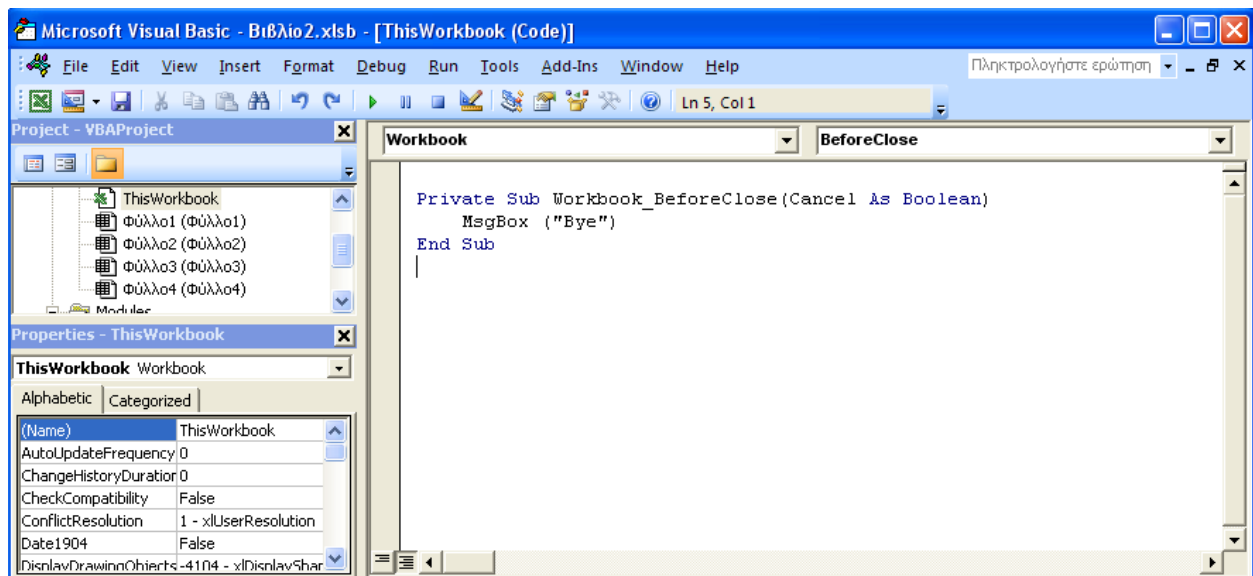
- ☞ Δεν επιτρέπεται η αλλαγή του ονόματος μιας διαδικασίας συμβάντων και η μετακίνησή της σε άλλη λειτουργική μονάδα κώδικα.

Για να γράψουμε κώδικα για την παραπάνω διαδικασία, ακολουθούμε τα εξής βήματα:

- Στο παράθυρο "Project Explorer" κάνουλα διπλό κλικ στο αντικείμενο, στο οποίο αντιστοιχεί το συμβάν και εμφανίζεται το σχετικό παράθυρο κώδικα. Για παράδειγμα, κάνουμε κλικ στο αντικείμενο **ThisWorkbook**.
- 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. Αγνοούμε τον ορισμό αυτό ή τον διαγράφουμε.
- Από την αναδιπλούμενη λίστα των διαδικασιών, που βρίσκεται στο πάνω δεξιό μέρος του παραθύρου, επιλέγουμε το συμβάν για το οποίο θέλουμε να γράψουμε κώδικα.

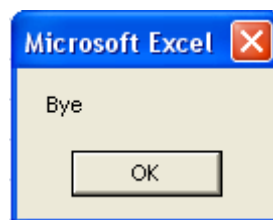


- Για παράδειγμα, επιλέγουμε **BeforeClose**.
- Στη συνέχεια, πληκτρολογούμε τον κώδικα που θέλουμε να εκτελείται πριν κλείσει το ενεργό βιβλίο εργασίας.



- Για να αποθηκεύσουμε τις αλλαγές και να επιστρέψουμε στο Excel, επιλέγουμε πρώτα την εντολή **Save** και, στη συνέχεια, την εντολή **Close and Return to Microsoft Excel** από το μενού **File** του VBE.

Όταν κλείσουμε το βιβλίο εργασίας, το Excel θα εμφανίσει το παρακάτω παράθυρο διαλόγου:



4.2.4 Τύποι Δεδομένων και Μεταβλητές

Οι μεταβλητές είναι επώνυμες θέσεις μνήμης και χρησιμεύουν στην προσωρινή αποθήκευση πληροφοριών. Τα ονόματα των μεταβλητών πρέπει να είναι περιγραφικά και να ακολουθούν τους ίδιους κανόνες ονομασίας με τις διαδικασίες. Είναι αυτονόητο, ότι δεν μπορούμε να χρησιμοποιήσουμε για ονόματα μεταβλητών δεσμευμένες λέξεις της VBA, όπως είναι οι λέξεις-κλειδιά If, For, Next, Sub κ.λπ.

Για να εκχωρήσουμε μια τιμή σε μια μεταβλητή, χρησιμοποιούμε τον τελεστή ίσον (=), όπως φαίνεται και στα παραδείγματα που ακολουθούν:

Timi = 123

fpa = 0.18

Telos = False

Onoma = "Σούλα"

Hmerominia = #8/28/2008#

Η τιμή που βρίσκεται δεξιά του τελεστή εκχωρείται στη μεταβλητή που βρίσκεται στα αριστερά του.

Ο παραπάνω τρόπος δημιουργίας μεταβλητών δεν συνιστάται. Αν και φαίνεται εύκολος, μπορεί να προκαλέσει λάθη τα οποία ακόμα και έμπειροι προγραμματιστές δύσκολα μπορούν να εντοπίσουν. Για να δηλώσουμε μια μεταβλητή με τον "επίσημο" τρόπο, πρέπει να χρησιμοποιήσουμε την ακόλουθη σύνταξη:

Dim όνομα_μεταβλητής [As τύπος_δεδομένων]

Αν δηλώσουμε μια μεταβλητή χωρίς να προσδιορίσουμε τον τύπο δεδομένων της, αυτή παίρνει αυτόματα τον τύπο **Variant**. Ο τύπος δεδομένων προσδιορίζει το είδος της πληροφορίας που θα αποθηκεύσουμε στη μεταβλητή. 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.

Για να αναγκάσουμε την VBA να δέχεται μόνο τη ρητή δήλωση μεταβλητών με την πρόταση **Dim**, πρέπει να εισαγάγουμε στο τμήμα δηλώσεων (**Declarations**) της λειτουργικής μονάδας κώδικα την πρόταση:

Option Explicit

Τώρα αν δηλώσουμε σε μια διαδικασία μια μεταβλητή χωρίς την πρόταση Dim και εκτελέσουμε τη διαδικασία, η VBA θα εμφανίσει.

Οι μεταβλητές που δηλώνονται μέσα σε μια διαδικασία ονομάζονται **τοπικές μεταβλητές**, μπορούν να χρησιμοποιηθούν μόνο στη συγκεκριμένη διαδικασία και καταστρέφονται μόλις τερματιστεί η διαδικασία. Οι μεταβλητές αυτές δηλώνονται με την πρόταση Dim αμέσως μετά την πρόταση Sub, όπως δείχνει και το παράδειγμα που ακολουθεί:

Sub Test ()

Dim Onoma As String Dim x

Dim Ilikia As Integer, Hmerominia As Date

Ακολουθεί ο κώδικας της διαδικασίας

End Sub

Η πρώτη πρόταση Dim δηλώνει μια μεταβλητή τύπου **String**, στην οποία μπορούμε να αποθηκεύσουμε αλφαριθμητικά (κείμενο). Με την επόμενη πρόταση δηλώνουμε μια μεταβλητή με όνομα x, η οποία θα είναι τύπου Variant, αφού δηλώθηκε χωρίς τύπο δεδομένων. Στην τρίτη πρόταση δηλώθηκαν δύο μεταβλητές μαζί, η μια ως ακέραια μεταβλητή (**Integer**) και η άλλη ως μεταβλητή ημερομηνίας (**Date**). Η επόμενη πρόταση αποτελεί σχόλιο. Τα σχόλια δεν εκτελούνται από την VBA και χρησιμεύουν στην τεκμηρίωση του κώδικα. Ένα σχόλιο αρχίζει με απόστροφο και εμφανίζεται με πράσινο χρώμα.

Αν θέλουμε μια μεταβλητή να είναι προσπελάσιμη από όλες τις διαδικασίες της λειτουργικής μονάδας, πρέπει να τη δηλώσουμε στο τμήμα δηλώσεων της λειτουργικής μονάδας κώδικα, πριν από οποιαδήποτε διαδικασία. Το επόμενο παράδειγμα δείχνει τη χρήση μιας μεταβλητής λειτουργικής μονάδας από δύο διαδικασίες της.

Dim Timi As Single *'Μεταβλητή Λειτ. Μονάδας*

Sub Arxikopoihsi()

Timi = 10.5

End Sub

Sub Υπολογισμος(Posotita)

Dim TelikiTimi As Single ' Τοπική μεταβλητή TelikiTimi = Timi *Posotita

MsgBox TelikiTimi

End Sub

Στην πρώτη διαδικασία αρχικοποιούμε με χρήση του τελεστή εκχώρησης την μεταβλητή Timi, που δηλώθηκε στο τμήμα δηλώσεων της λειτουργικής μονάδας. Στη διαδικασία Υπολογισμος δηλώνουμε μια τοπική μεταβλητή TelikiTimi και, στη συνέχεια, της εκχωρούμε το αποτέλεσμα ενός υπολογισμού, στο οποίο παίρνει μέρος η μεταβλητή Timi.

☞ Στο επίπεδο λειτουργικής μονάδας μπορούμε αντί για τη λέξη-κλειδί Dim να χρησιμοποιήσουμε την ισοδύναμη λέξη-κλειδί **Private** (ιδιωτική).

Για να δηλώσουμε μια μεταβλητή, η οποία θα είναι προσπελάσιμη από όλες τις διαδικασίες όλων των λειτουργικών μονάδων κώδικα του έργου πρέπει να δηλώσουμε τη μεταβλητή στο τμήμα δηλώσεων χρησιμοποιώντας τη λέξη-κλειδί **Public** (δημόσια) αντί για Dim ή Private.

Public User As String

Μια ειδική κατηγορία μεταβλητών είναι οι στατικές μεταβλητές. Αυτές δηλώνονται με τη χρήση της λέξης-κλειδί **Static** σε επίπεδο διαδικασίας και διατηρούν την τιμή τους και μετά το τερματισμό της διαδικασίας. Στο παράδειγμα που ακολουθεί, η μεταβλητή i αυξάνει κατά 1 κάθε φορά που καλούμε τη διαδικασία MyCount.

Sub MyCount ()

Static i As Integer

i = i + 1

End Sub

Σε αντίθεση με μια μεταβλητή, μια σταθερά (**constant**) αρχικοποιείται με μια τιμή, η οποία δεν μπορεί να αλλάξει στη συνέχεια. Η σύνταξη για τη δημιουργία μιας σταθεράς έχει ως εξής:

[Public | Private] Const όνομα_σταθεράς [As τύπος]=τιμή

Ακολουθούν τρία παραδείγματα δήλωσης σταθερών:

Const ORIO = 50

Const PASSWORD = "12345"

Public Const FPA As Single = 0.23

Μια καλή πρακτική είναι να γράφουμε τα ονόματα των σταθερών με κεφαλαία γράμματα, για να ξεχωρίζουν εύκολα από τις μεταβλητές.

Η χρήση σταθερών αντί κυριολεκτικών τιμών κάνουν τον κώδικα ευανάγνωστο και διευκολύνει τη μελλοντική αλλαγή τιμών. Για παράδειγμα, αν χρησιμοποιήσουμε την τιμή 0.23 (23% ΦΠΑ) για υπολογισμούς και κάποια στιγμή αλλάξει η τιμή αυτή, θα πρέπει να εντοπίσουμε όλες τις εμφανίσεις της τιμής αυτής στον κώδικα και να κάνουμε τις κατάλληλες τροποποιήσεις. Αν όμως έχουμε χρησιμοποιήσει σταθερά, αρκεί να αλλάξουμε μόνο την τιμή της σταθεράς.

Εκτός από τις δικές μας σταθερές υπάρχουν πάρα πολλές ενσωματωμένες σταθερές της VBA, τις οποίες μπορούμε να χρησιμοποιήσουμε χωρίς να τις δηλώσουμε. Για παράδειγμα, στη συνάρτηση MsgBox μπορούμε να χρησιμοποιήσουμε τη σταθερά **vbYesNo** (με τιμή 4), για να εμφανίσουμε στο παράθυρο διαλόγου τα δύο κουμπιά **Ναι** και **Όχι** αντί του προκαθορισμένου **OK**.

☞ Μπορούμε να δούμε όλες τις ενσωματωμένες σταθερές (καθώς και τις ιδιότητες και τις μεθόδους) από το παράθυρο του Object Browser **[F2]**.

Ο πίνακας που ακολουθεί δείχνει τους τύπους δεδομένων της VBA, το μέγεθος σε bytes που καταλαμβάνουν στη μνήμη και το εύρος τιμών τους.

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	

Οι οριζόμενοι από το χρήστη τύποι δεδομένων (user defined types) μπορούν να περιέχουν ένα ή περισσότερα στοιχεία οποιουδήποτε τύπου δεδομένων και ορίζονται με την πρόταση Type, όπως στο επόμενο παράδειγμα:

```
Private Type myType
    LastName As String
    BirthDay As Date
End Type
```

4.2.6 MsgBox - InputBox

Στα προηγούμενα παραδείγματα είδαμε τη συνάρτηση MsgBox, με την οποία μπορούμε να εμφανίσουμε μηνύματα στο χρήστη. Η συνηθισμένη σύνταξη της συνάρτησης έχει ως εξής

MsgBox(μήνυμα[, κουμπιά] [, τίτλος])

- *μήνυμα*: Είναι το κείμενο που εμφανίζεται στο παράθυρο διαλόγου και είναι το υποχρεωτικό όρισμα της συνάρτησης.
- *κουμπιά*: Αριθμητική τιμή που προσδιορίζει το πλήθος, το εικονίδιο, και το είδος των κουμπιών που θα εμφανιστούν στο παράθυρο διαλόγου, καθώς και το προεπιλεγμένο κουμπί. Αν παραλειφθεί, η προκαθορισμένη τιμή είναι μηδέν και εμφανίζεται το κουμπί OK.
- *τίτλος*: Ο τίτλος του παραθύρου διαλόγου. Αν παραληφθεί, εμφανίζεται το όνομα της εφαρμογής.

Ο πίνακας που ακολουθεί περιγράφει τις τιμές και τις σταθερές που μπορούμε να χρησιμοποιήσουμε για το δεύτερο όρισμα, ώστε να προσαρμόσουμε κατάλληλα το παράθυρο διαλόγου.

Σταθερά	Τιμή	Περιγραφή
vbOKOnly	0	Εμφανίζει το κουμπί OK.
vbOKCancel	1	Εμφανίζει τα κουμπιά OK και Cancel.
vbAbortRetryIgnore	2	Εμφανίζει τα κουμπιά Abort, Retry και Ignore.
vbYesNoCancel	3	Εμφανίζει τα κουμπιά Yes, No και Cancel.
vbYesNo	4	Εμφανίζει τα κουμπιά Yes και No.
vbRetryCancel	5	Εμφανίζει τα κουμπιά Retry και Cancel.
vbCritical	16	Εμφανίζει το εικονίδιο Κρίσιμου Μηνύματος.
vbQuestion	32	Εμφανίζει το εικονίδιο Προειδοποιητικού Ερωτήματος.
vbExclamation	48	Εμφανίζει το εικονίδιο Προειδοποιητικού Μηνύματος.
vbInformation	64	Εμφανίζει το εικονίδιο Πληροφοριακού Μηνύματος.
vbDefaultButton1	0	Το πρώτο κουμπί είναι το προεπιλεγμένο.
vbDefaultButton2	256	Το δεύτερο κουμπί είναι το προεπιλεγμένο.

Σταθερά	Τιμή	Περιγραφή
vbDefaultButton3	512	Το τρίτο κουμπί είναι το προεπιλεγμένο.
vbApplicationModal	0	Ο χρήστης πρέπει να ανταποκριθεί στο παράθυρο διαλόγου πριν συνεχίσει την εργασία με οποιαδήποτε εφαρμογή.
vbSystemModal	4096	Ο χρήστης πρέπει να ανταποκριθεί στο παράθυρο διαλόγου πριν συνεχίσει την εργασία με οποιαδήποτε εφαρμογή.

Οι πρώτες έξι τιμές (0-5) του πίνακα καθορίζουν το πλήθος και το είδος των κουμπιών που θα εμφανιστούν στο παράθυρο διαλόγου. Οι επόμενες τέσσερις τιμές (16, 32, 48, 64) καθορίζουν το είδος του εικονιδίου. Ακολουθούν τρεις τιμές (0, 256, 512) που καθορίζουν το προεπιλεγμένο κουμπί και τέλος, οι δύο επόμενες τιμές (0, 4096) καθορίζουν την υποχρέωση (modality) για ανταπόκριση στο παράθυρο διαλόγου. Για να υπολογίσουμε την τελική τιμή του ορίσματος, προσθέτουμε έναν μόνο αριθμό από κάθε ομάδα.

Η σύνταξη της MsgBox για να εμφανίσουμε ένα παράθυρο διαλόγου που θα έχει τα κουμπιά Ναι και Όχι, προεπιλεγμένο το δεύτερο κουμπί και θα εμφανίζει το εικονίδιο Πληροφοριακού Μηνύματος, έχει ως εξής

MsgBox "Είσαι σίγουρος;", 4+64+256, "Η εφαρμογή μου"

ή ακόμα πιο σύντομα:

MsgBox "Είσαι σίγουρος;", 320, "Η εφαρμογή μου"

Χρησιμοποιώντας τις ενσωματωμένες σταθερές αντί για αριθμητικές τιμές, θα μπορούσαμε να γράψουμε ισοδύναμα:

MsgBox "Είσαι σίγουρος;", vbYesNo + vbDefaultButton2 + vbInformation, "Η εφαρμογή μου"

☞ Μια πρόταση της VBA μπορεί να επεκταθεί σε δύο ή περισσότερες γραμμές, χρησιμοποιώντας στο τέλος κάθε γραμμής το κενό διάστημα και τον χαρακτήρα υπογράμμισης (_).

Αν το μήνυμα που θέλουμε να εμφανίσουμε με την MsgBox είναι μεγάλο και θέλουμε να το "σπάσουμε" σε περισσότερες γραμμές, μπορούμε να χρησιμοποιήσουμε τον τελεστή συνένωσης αλφαριθμητικών (&) σε συνδυασμό με τη σταθερά αλλαγής γραμμής **vbCrLf**, όπως φαίνεται στο επόμενο παράδειγμα:

MsgBox "Είσαι σίγουρος για την επιλογή σου;" & vbCrLf & "Αν αμφιβάλλεις κάνε κλικ στο κουμπί Όχι", vbYesNo

Η MsgBox είναι στην πραγματικότητα μια συνάρτηση, αυτό σημαίνει ότι επιστρέφει κάποια τιμή. Αν μας ενδιαφέρει η ανταπόκριση του χρήστη, δηλαδή σε ποιο κουμπί έκανε κλικ, θα πρέπει να αποθηκεύσουμε την τιμή που επιστρέφει η συνάρτηση σε μια μεταβλητή.

Στο παράδειγμα που ακολουθεί, γίνεται έλεγχος για την τιμή επιστροφής που εκχωρείται στη μεταβλητή intRes και στη συνέχεια, με μια δομή If - End If εκτελείται ο ανάλογος κώδικας.

```

Dim intRes As Integer
intRes = MsgBox("Να συνεχίσω;", vbYesNo)
If intRes = vbYes Then
    'Κώδικας αν η μεταβλητή intRes έχει τιμή vbYes ή 6
Else
    'Κώδικας αν η μεταβλητή intRes έχει τιμή vbNo ή 7
End If

```

Ο πίνακας που ακολουθεί δείχνει τις επιστρεφόμενες σταθερές και τις αντίστοιχες αριθμητικές τιμές από την συνάρτηση MsgBox"

Σταθερά	Τιμή	Περιγραφή
vbOK	1	Έγινε κλικ στο κουμπί OK
vbCancel	2	Έγινε κλικ στο κουμπί Cancel
vbAbort	3	Έγινε κλικ στο κουμπί Abort
vbRetry	4	Έγινε κλικ στο κουμπί Retry
vbIgnore	5	Έγινε κλικ στο κουμπί Ignore
vbYes	6	Έγινε κλικ στο κουμπί Yes
vbNo	7	Έγινε κλικ στο κουμπί No

Η συνάρτηση **InputBox** χρησιμοποιείται για την εισαγωγή δεδομένων από το χρήστη. Εμφανίζει ένα παράθυρο διαλόγου, που προβάλλει ένα μήνυμα και περιέχει ένα πλαίσιο κειμένου για την πληκτρολόγηση των δεδομένων.

Η μερική σύνταξή της έχει ως εξής

```

InPutBox(μηνυμα[, τίτλος] [, προεπιλογή])

```

- *μήνυμα:* Είναι το κείμενο που εμφανίζεται στο παράθυρο διαλόγου. Είναι το υποχρεωτικό όρισμα της συνάρτησης.
- *τίτλος ()* τίτλος του παραθύρου διαλόγου. Αν παραληφθεί, εμφανίζεται το όνομα της εφαρμογής.
- *προεπιλογή:* Η προεπιλεγμένη τιμή στο πλαίσιο κειμένου. Συνήθως αφήνουμε το πλαίσιο κειμένου κενό.

Η συνάρτηση **InputBox** επιστρέφει το κείμενο που πληκτρολόγησε ο χρήστης στο πλαίσιο κειμένου, εφόσον πατηθεί το κουμπί **OK**. Αν ο χρήστης δεν πληκτρολογήσει κάτι ή πατήσει το κουμπί **Cancel**, τότε η συνάρτηση επιστρέφει το κενό αλφαριθμητικό ("").

Στο παράδειγμα που ακολουθεί, ζητείται το όνομα του χρήστη και η απάντησή του αποθηκεύεται στη μεταβλητή strName. Στη συνέχεια, χρησιμοποιούμε τον τελεστή συνένωσης αλφαριθμητικών (&), για να εμφανίσουμε το μήνυμα με τη χρήση της MsgBox.

```
Dim strName As String
strName = InputBox ("Πως σε λένε; ", "Η εφαρμογή μου")
MsgBox "Γειά σου" & strName
```

4.2.7 Εντολές Επιλογής

Οι εντολές σε μια διαδικασία εκτελούνται με τη σειρά, η μια μετά την άλλη. Τις περισσότερες φορές όμως η λύση ενός προβλήματος απαιτεί τη λήψη αποφάσεων ανάλογα με τις συνθήκες που επικρατούν. Η εντολή **If - End If** επιτρέπει σ' ένα πρόγραμμα να πάρει αποφάσεις και να εκτελέσει κάποιον συγκεκριμένο κώδικα ανάλογα με το αποτέλεσμα της σύγκρισης.

Η απλούστερη μορφή σύνταξής της έχει ως εξής

```
If συνθήκη Then
    [εντολές]
```

```
End If
```

Η εντολή If αποτιμά το αποτέλεσμα της συνθήκης και αν είναι αληθές (True) τότε θα εκτελεστούν οι εντολές που βρίσκονται μέσα στη δομή If - End If. Αν το αποτέλεσμα της συνθήκης είναι ψευδές (False), τότε δεν θα εκτελεστούν οι εντολές αυτές. Στη συνέχεια και στις δύο περιπτώσεις θα εκτελεστεί ο κώδικας που ακολουθεί το End If.

```
Ekptosi = 0
```

```
If Posotita > 100 Then
```

```
    Ekptosi = 0.1
```

```
End If
```

Στο παραπάνω παράδειγμα ελέγχουμε αν η μεταβλητή Posotita έχει τιμή μεγαλύτερη του 100. Αν αυτό αληθεύει, εκχωρούμε στη μεταβλητή Ekptosi την τιμή 0.1, διαφορετικά η τιμή της μεταβλητής παραμένει μηδέν.

Αν θέλουμε να εκτελεστούν εντολές στην περίπτωση που το αποτέλεσμα είναι ψευδές, μπορούμε να χρησιμοποιήσουμε την προαιρετική πρόταση Else:

```
If συνθήκη Then
```

```
    [εντολές]
```

```
Else
```

```
    [εντολές]
```

```
End If
```

Στο παράδειγμα που ακολουθεί, ζητείται το όνομα του χρήστη και σύμφωνα με την απάντησή του, εμφανίζεται ένα ανάλογο μήνυμα καλωσορίσματος.

```
Dim UserName As String
```

```
UserName = InputBOX("Πως λέγεσαι:")
```

```
UserName = "" Then
```

```
    MsgBox "Γεια σου ανώνυμε χρήστη."
```

```
Else
```

```
    MsgBox "Γεια σου" & UserNarne
```

```
End If
```

Αν οι επιλογές είναι περισσότερες από δύο, μπορούμε να χρησιμοποιήσουμε την πλήρη δομή της If, που έχει ως εξής:

```
If συνθήκη-1 Then
    [εντολές]
[Elseif συνθήκη-2 Then
    [εντολές]]
...
[Elseif συνθήκη-n Then
    [εντολές]]
[Else
    [εντολές] ]
End If
```

Με την προαιρετική πρόταση **Elseif** μπορούμε να ελέγξουμε πολλαπλές συνθήκες. Αν και μπορούμε να χρησιμοποιήσουμε όσες προτάσεις Elseif θέλουμε, θα εκτελεστούν οι εντολές της πρώτης συνθήκης που θα αποτιμηθεί ως αληθής. Ακολουθεί ένα παράδειγμα, όπου χρησιμοποιούνται δύο προτάσεις Elseif και η Else:

```
Dim posotita As Integer
Dim ekptosi As Single
posotita = InputBOX("Δώσε ποσότητα: ")
If posotita < 10 Then
    ekptosi = 0
Elseif posotita < 50
    ekptosi = 0.1
Else
    If posotita < 100
        ekptosi = 0.2
    Else
        ekptosi = 0.3
    End If
End If
```

Το παραπάνω τμήμα κώδικα μπορεί να γίνει πιο κατανοητό, αν χρησιμοποιήσουμε **λογικούς τελεστές** (And, Or, Not):

```
If posotita < 10 Then
    ekptosi = 0
Elseif posotita >= 10 And posotita < 50
    ekptosi = 0.1
Elseif posotita >= 50 And posotita < 100
    ekptosi = 0.2
Else
    ekptosi = 0.3
End If
```

Όπως έχουμε ήδη μάθει, ο λογικός τελεστής **And** παράγει αποτέλεσμα True, αν και οι δύο πλευρές του είναι αληθείς. Ο τελεστής Or παράγει **True**, αν τουλάχιστον μια από τις πλευρές του είναι αληθής. Ο τελεστής Not παράγει το αντίθετο αποτέλεσμα. Εάν η συνθήκη είναι αληθής, το *Not συνθήκη* παράγει False και το αντίστροφο.

Η VBA διαθέτει μια εναλλακτική δομή της **If**, που λέγεται **Select Case** και χειρίζεται καλύτερα τις πολλαπλές επιλογές. Η σύνταξή της έχει ως εξής:

Select Case *έκφραση*

Case *τιμή-1*

[εντολές]

[Case *τιμή-2*

[εντολές]]

...

[Case *τιμή-ν*

[εντολές]]

[Case Else

[εντολές]]

End Select

Το παράδειγμα που ακολουθεί, χρησιμοποιεί τη δομή Select Case για να κάνει αποτίμηση της τιμής μιας μεταβλητής

```
Dim intNum As Integer
```

```
intNum = 7
```

```
Select Case intNum
```

```
Case 0 To 5
```

```
    MsgBox "Μεταξύ 0 και 5"
```

```
Case 6,7,8
```

```
    MsgBox "6, 7 ή 8"
```

```
Case 9
```

```
    MsgBox "9"
```

```
Case Is >= 10
```

```
    MsgBox "Μεγαλύτερος ή ίσος του 10"
```

```
Case Else
```

```
    MsgBox "Άρνητικός αριθμός"
```

```
End Select
```

Η λέξη-κλειδί **To** προσδιορίζει ένα εύρος τιμών, ενώ η λέξη-κλειδί **Is** χρησιμοποιείται με έναν από τους παρακάτω τελεστές σύγκρισης για τον προσδιορισμό του εύρους τιμών:

Τελεστής	Περιγραφή
>	Μεγαλύτερο από
>=	Μεγαλύτερο ή ίσο από
<	Μικρότερο από
<=	Μικρότερο ή ίσο από
<>	Όχι ίσο (διάφορο από)
=	Ίσο

Όταν γίνεται σύγκριση αλφαριθμητικών τιμών, η VBA χρησιμοποιεί τον κώδικα ASCII για τη σύγκριση των χαρακτήρων. Σύμφωνα με τον κώδικα ASCII, το λατινικό "A" με τιμή 65 είναι μικρότερο από το "B" με τιμή 66. Αν οι πρώτοι χαρακτήρες των αλφαριθμητικών είναι ίδιοι, γίνεται σύγκριση των δεύτερων χαρακτήρων κ.ο.κ. Έτσι για παράδειγμα, η συνθήκη "Γιώργος" > "Γιάννης" είναι αληθής, επειδή οι πρώτοι δύο χαρακτήρες είναι ίδιοι και το "ώ" έχει μεγαλύτερη τιμή (233) στον κώδικα ASCII από το χαρακτήρα "ά" με τιμή (225).

4.2.8 Εντολές Επανάληψης (Βρόχοι)

Βρόχος είναι μια δομή που εκτελεί επαναληπτικά ένα μπλοκ εντολών μέχρι να ικανοποιηθεί μια συνθήκη ή κάποιος συγκεκριμένος αριθμός επαναλήψεων. Ο απλούστερος τύπος βρόχου είναι η δομή **For - Next**, που απαιτεί να ξέρουμε πόσες φορές θα εκτελεστούν οι εντολές του, και έχει την παρακάτω σύνταξη:

For *μετρητής* = *αρχική_τιμή* To *τελική_τιμή* [Step βήμα]

[εντολές]

[Exit For]

[εντολές]

Next [*μετρητής*]

Ο *μετρητής* είναι μια αριθμητική μεταβλητή που αρχικοποιείται στην κεφαλή του βρόχου και μετά από κάθε επανάληψη αυξάνει κατά ένα, εκτός αν προβλεφθεί διαφορετικά με την προαιρετική πρόταση Step. Όταν ο μετρητής ξεπεράσει την τελική τιμή, τερματίζεται η επανάληψη του μπλοκ των εντολών. Μπορούμε να τερματίσουμε πρόωρα ένα βρόχο For - Next με χρήση της πρότασης **Exit For**.

Στο παράδειγμα που ακολουθεί, υπολογίζουμε το άθροισμα των ακεραίων 1 έως 100:

```
Dim i As Integer, sum As Integer
```

```
sum = 0
```

```
For i = 1 To 100
```

```
    sum = sum + i
```

```
Next i
```

Ο βρόχος του επόμενου παραδείγματος ξεκινάει από το 100 και μειώνεται σε κάθε επανάληψη κατά 5, μέχρι ο μετρητής να πάρει την τιμή 0.

```
For i = 100 To 0 Step -5
```

```
    sum = sum + i
```

```
Next i
```

Όταν χρησιμοποιούμε ως βήμα αρνητική τιμή, πρέπει η αρχική τιμή του μετρητή να είναι μεγαλύτερη από την τελική.

Ο βρόχος **Do - While** είναι μια άλλη δομή επανάληψης, που εκτελείται όσο ικανοποιείται μια συνθήκη και έχει δύο μορφές σύνταξης:

Do While *συνθήκη*

```
    [εντολές]
```

```
    [Exit Do]
```

```
    [εντολές]
```

Loop

Do

```
    [εντολές]
```

```
    [Exit Do]
```

```
    [εντολές]
```

Loop While *συνθήκη*

Η διαφορά στις δύο μορφές σύνταξης είναι η τοποθέτηση της συνθήκης στην αρχή ή στο τέλος του βρόχου. Στην πρώτη περίπτωση οι εντολές του βρόχου μπορεί να μην εκτελεστούν ποτέ (αν η συνθήκη είναι ψευδής). Στη δεύτερη περίπτωση οι εντολές του βρόχου θα εκτελεστούν τουλάχιστον μια φορά. Η πρόταση **Exit Do** τερματίζει πρόωρα ένα βρόχο Do - While.

Το παράδειγμα που ακολουθεί υπολογίζει το άθροισμα των ακεραίων 1 έως 100 με ένα βρόχο Do - While:

```
Dim i As Integer, sum As Integer
```

```
sum = 0
```

```
i = 1
```

```
Do While i < 100
```

```
    sum = sum + i
```

```
    i = i + 1
```

```
Loop
```

Παρατηρούμε ότι δεν πρέπει να ξεχάσουμε να αρχικοποιήσουμε το μετρητή πριν το βρόχο και μέσα στο βρόχο να τον αυξήσουμε κατά ένα, κάτι που γίνεται αυτόματα στο βρόχο For - Next.

Μια παραλλαγή αυτού του βρόχου είναι η δομή **Do - Until**. Η διαφορά τους είναι ότι ένας βρόχος Do - Until εκτελείται όσο η συνθήκη είναι ψευδής και τερματίζει μόλις η συνθήκη γίνει αληθής (True). Ο βρόχος Do - Until έχει και αυτός δύο μορφές σύνταξης

Do Until *συνθήκη*

[εντολές]

[Exit Do]

[εντολές]

Loop

Do

[εντολές]

[Exit Do]

[εντολές]

Loop Until *συνθήκη*

Στην πρώτη μορφή σύνταξης ο βρόχος μπορεί να μην εκτελεστεί ποτέ, ενώ στη δεύτερη περίπτωση θα εκτελεστεί τουλάχιστον μια φορά. Και εδώ, η πρόταση Exit Do τερματίζει πρόωρα το βρόχο. Το παράδειγμα που ακολουθεί υπολογίζει πάλι το άθροισμα των ακεραίων από 1 έως 100.

sum = 0

i = 1

Do Until i= 101

sum = sum + i

i = i + 1

Loop

Το επόμενο παράδειγμα δείχνει έναν ατέρμονα βρόχο, ο οποίος τερματίζεται μόνο αν δοθεί ο σωστός κωδικός πρόσβασης, που είναι η λέξη *secret*.

Dim strPasswd As String

Do

strPasswd = InputBox("Δώσε τον κωδικό πρόσβασης:")

Loop Until strPasswd = "12345"

Η VBA διαθέτει μια ειδική δομή επανάληψης **For - Each**, που εφαρμόζεται σε συλλογές αντικειμένων και έχει την παρακάτω σύνταξη:

For Each *στοιχείο* In *ομάδα*

[εντολές]

Exit For

[εντολές]

Next [*στοιχείο*]

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.

Το επόμενο παράδειγμα εμφανίζει τα ονόματα των φύλλων εργασίας του ενεργού βιβλίου εργασίας

Dim ws As Worksheet
For Each ws In ActiveWorkbook.WorkSheets
 MsgBox ws.Name
Next ws

4.2.9 Πίνακες

Ένας πίνακας είναι μια ομάδα στοιχείων του ίδιου τύπου που έχουν το ίδιο όνομα. Είναι πολύ χρήσιμοι επειδή μας επιτρέπουν να επεξεργασθούμε μεγάλες ποσότητες δεδομένων με τη βοήθεια βρόχων, κάτι που είναι αδύνατο με τη χρήση απλών μεταβλητών.

Οι πίνακες δηλώνονται με μια πρόταση Dim, Private ή Public, όπως ακριβώς και μια απλή μεταβλητή με την προσθήκη ενός ακέραιου αριθμού σε παρενθέσεις, που δηλώνει το πλήθος των στοιχείων του. Για παράδειγμα, για να δηλώσουμε έναν πίνακα 10 ακεραίων γράφουμε:

```
Dim matrix(100) As Integer
```

Το πρώτο στοιχείο του πίνακα είναι το matrix(0) και το τελευταίο είναι το matrix(100). Αν θέλουμε το πρώτο στοιχείο του πίνακα να έχει δείκτη ένα και όχι μηδέν, πρέπει να συμπεριλάβουμε την πρόταση

```
Option Base 1
```

στο τμήμα δηλώσεων της λειτουργικής μονάδας, πριν από οποιαδήποτε διαδικασία ή δήλωση μεταβλητών. Εναλλακτικά, μπορούμε να δηλώσουμε έναν πίνακα καθορίζοντας το πρώτο και το τελευταίο στοιχείο του, όπως φαίνεται στην πρόταση που ακολουθεί:

```
Dim matrix(1 To 100) As Integer
```

Στο παράδειγμα που ακολουθεί εκχωρούμε τυχαίες τιμές από 1 το έως και το 49 σ' έναν πίνακα 6 ακεραίων, με τη χρήση του βρόχου For και της συνάρτησης **Rnd**, που επιστρέφει τυχαίες τιμές στο διάστημα **[0, 1)**.

```
Dim luckynumbers(5) As Integer
```

```
Randomize
```

```
For i = 0 To 5
```

```
    luckynumbers(i) = Int((49 - 1 + 1) * Rnd + 1)
```

```
Next i
```

H Randomize αρχικοποιεί τη γεννήτρια «ψευδοτυχαίων» αριθμών της Rnd, ενώ η συνάρτηση **Int** επιστρέφει το ακέραιο μέρος ενός πραγματικού αριθμού.

Τα παραπάνω παραδείγματα αφορούν πίνακες μιας διάστασης. Αν χρειαζόμαστε πίνακες με περισσότερες διαστάσεις, θα πρέπει για κάθε διάσταση να δηλώσουμε το πρώτο και το τελευταίο στοιχείο, όπως φαίνεται στην επόμενη πρόταση:

```
Dim matrix(1 To 4, 1 To 5) As Integer
```

Ο παραπάνω πίνακας αποτελείται από τέσσερις γραμμές και πέντε στήλες.

Για να αναφερθούμε σ' ένα στοιχείο αυτού του πίνακα, πρέπει να αναφερθούμε ταυτόχρονα και στον αριθμό της γραμμής και στον αριθμό της στήλης που ανήκει. Για παράδειγμα, με την επόμενη πρόταση εκχωρήσουμε μια τιμή στο στοιχείο της 1ης γραμμής και της 2ης στήλης του πίνακα:

matrix(1,2) = 63

Η VBA υποστηρίζει πίνακες μέχρι 60 διαστάσεων, στην πράξη όμως χρησιμοποιούμε μόνο τις πρώτες τρεις.

Όλοι οι προηγούμενοι πίνακες έχουν σταθερό πλήθος στοιχείων. Πολλές φορές δεν γνωρίζουμε το πλήθος των στοιχείων που θέλουμε να επεξεργαστούμε. Στην περίπτωση αυτή, μπορούμε να δηλώσουμε ένα **δυναμικό** πίνακα ως εξής:

```
Dim matrix() As Integer
```

Πριν τη χρήση ενός δυναμικού πίνακα, πρέπει να ορίσουμε με την πρόταση **ReDim** το πλήθος των στοιχείων του. Στο παράδειγμα που ακολουθεί, ζητείται πρώτα το πλήθος των δεδομένων από το χρήστη, μετά ορίζουμε το μέγεθος του πίνακα με την ReDim και στη συνέχεια μ' ένα βρόχο For εκχωρούμε τιμές σ' όλα τα στοιχεία του. Οι συναρτήσεις **LBound** και **UBound** μας επιστρέφουν το δείκτη του μικρότερου και μεγαλύτερου στοιχείου του πίνακα αντίστοιχα.

```
Dim intArray() As Integer
```

```
Dim elements As Integer, i As Integer
```

```
elements = InputBox("Πόσες τιμές θα δώσεις;")
```

```
ReDim intArray(elements)
```

```
For i = LBound(intArray) To UBound(intArray)
```

```
    intArray(i) = InputBox("Δώσε την" & i & " τιμή:")
```

```
Next i
```

4.3 Προσαρμοσμένες Συναρτήσεις

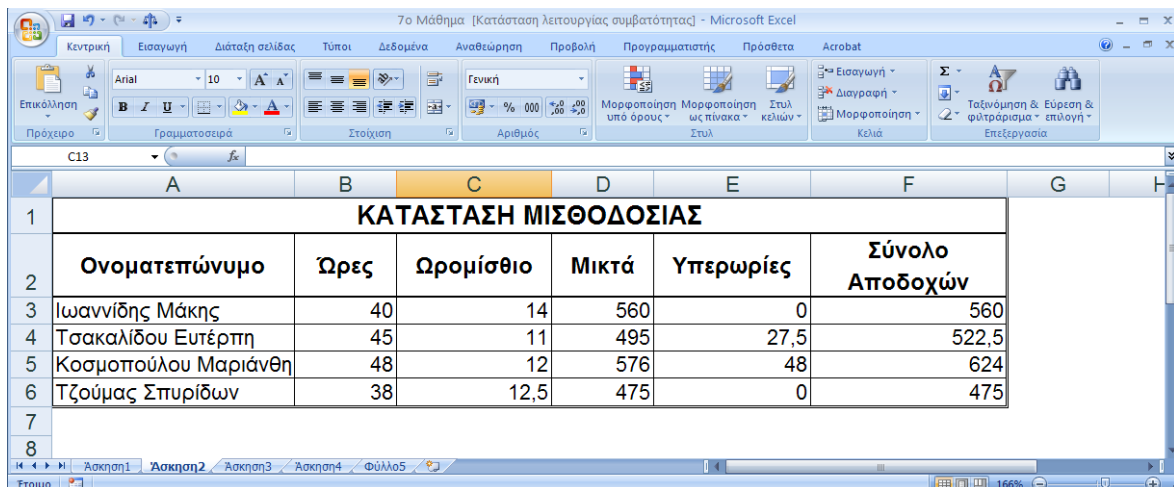
Παρ' ότι το Excel διαθέτει έναν μεγάλο αριθμό ενσωματωμένων συναρτήσεων, κάποια στιγμή είναι σίγουρο ότι θα χρειαστούμε μία συνάρτηση που δεν υπάρχει έτοιμη. Ευτυχώς το Excel μας δίνει τη δυνατότητα να προγραμματίσουμε τις δικές μας συναρτήσεις. Οι συναρτήσεις που δημιουργούνται από το χρήστη, λέγονται προσαρμοσμένες ή οριζόμενες από το χρήστη - **User Defined Functions (UDF)** και χρησιμοποιούνται ακριβώς όπως και οι ενσωματωμένες συναρτήσεις. Οι προσαρμοσμένες συναρτήσεις δέχονται τιμές από το φύλλο εργασίας, κάνουν κάποιους υπολογισμούς και επιστρέφουν μία (συνήθως αριθμητική) τιμή πίσω στο φύλλο εργασίας.

Δυστυχώς δεν μπορούμε να δημιουργήσουμε μία προσαρμοσμένη συνάρτηση με καταγραφή, όπως γίνεται με τις μακροεντολές αλλά πρέπει να γράψουμε κώδικα σε VBA. Ο κώδικας των συναρτήσεων γράφεται σε μια υπάρχουσα ή μια νέα λειτουργική μονάδα του βιβλίου εργασίας. Για να είναι οι συναρτήσεις μας προσπελάσιμες από όλα τα βιβλία εργασίας, μπορούμε να δημιουργήσουμε ένα πρόσθετο που θα τις περιέχει και να το εγκαταστήσουμε στο Excel.

4.3.1 Δημιουργία Προσαρμοσμένης Συνάρτησης

Θα δημιουργήσουμε μια απλή συνάρτηση, η οποία θα υπολογίζει το ποσό των υπερωριών των εργαζομένων μιας επιχείρησης.

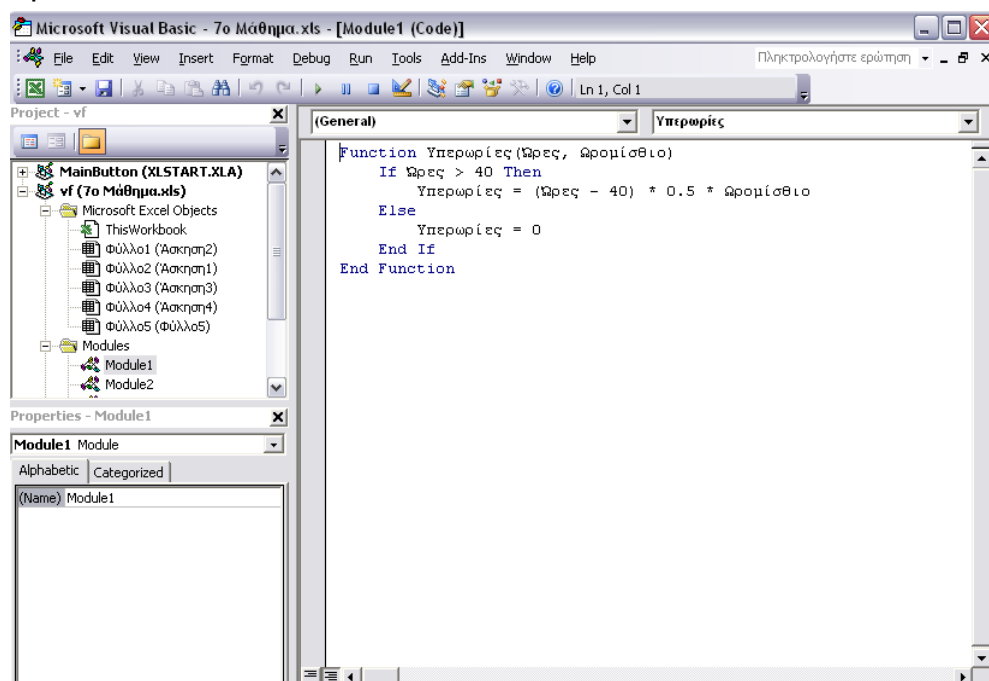
Για το σκοπό αυτό, δημιουργούμε το παρακάτω βιβλίο εργασίας, όπου θεωρούμε ότι οι κανονικές ώρες εργασίας για μια εβδομάδα είναι 40. Κάθε παραπάνω ώρα, θα πληρώνεται με επιπλέον 50% του ωρομίσθιου κάθε εργαζόμενου.



	A	B	C	D	E	F	G	H
1	ΚΑΤΑΣΤΑΣΗ ΜΙΣΘΟΔΟΣΙΑΣ							
2	Ονοματεπώνυμο	Ώρες	Ωρομίσθιο	Μικτά	Υπερωρίες	Σύνολο Αποδοχών		
3	Ιωαννίδης Μάκης	40	14	560	0	560		
4	Τσακαλίδου Ευτέρπη	45	11	495	27,5	522,5		
5	Κοσμοπούλου Μαριάνθη	48	12	576	48	624		
6	Τζούμας Σπυρίδων	38	12,5	475	0	475		
7								
8								

Για να δημιουργήσουμε την προσαρμοσμένη συνάρτηση, η οποία θα δέχεται το σύνολο των ωρών, το ωρομίσθιο και θα υπολογίζει το ποσό των υπερωριών, ακολουθούμε τα παρακάτω βήματα:

- Πατάμε το συνδυασμό πλήκτρων **[Alt]+[F11]** ή **Προγραμματιστής → Visual Basic**, για να εμφανίσουμε την Επεξεργασία της Visual Basic.
- Επιλέγουμε την εντολή **Module** από το μενού **Insert** για να εισαγάγουμε μία νέα κενή βασική λειτουργική μονάδα (Module1), όπου και πληκτρολογούμε τον παρακάτω κώδικα.



- Αποθηκεύουμε την εργασία μας, κάνοντας κλικ στο κουμπί **Save** ΑΛΦΑ.XLSX της γραμμής εργαλείων. Με την εντολή **Close and Return to Microsoft Excel** από το μενού **File**, επιστρέφουμε πίσω στο φύλλο εργασίας.

☞ Τα βιβλία εργασίας που περιέχουν κώδικα VBA αποθηκεύονται ως **βιβλία εργασίας του Excel με δυνατότητες μακροεντολών** (.xlsm).

4.3.2 Χρήση Προσαρμοσμένης Συνάρτησης

Μετά τον ορισμό (καταχώριση) της συνάρτησης σε μία λειτουργική μονάδα, είμαστε έτοιμοι να τη χρησιμοποιήσουμε στο φύλλο εργασίας. Στο κελί **E3** καταχωρίζουμε **=Υπερωρίες(B3;C3)** και πατάμε το πλήκτρο **[Enter]**. Το πρώτο όρισμα **B5** αντιστοιχεί στην παράμετρο **ώρες** της συνάρτησης, ενώ το όρισμα **C3** αντιστοιχεί στην παράμετρο **ωρομίσθιο** της συνάρτησης.

Το Excel θα κάνει τους κατάλληλους υπολογισμούς και θα εμφανίσει την τιμή 0,00 € το κελί **E3** γιατί πράγματι ο Ιωαννίδης Μάκης δεν έχει εργαστεί περισσότερες από 40 ώρες. Αντιγράφουμε τη συνάρτηση και στα υπόλοιπα κελιά, σύροντας τη λαβή συμπλήρωσης ως το κελί **E6**, οπότε το φύλλο εργασίας μας παίρνει την τελική του μορφή. Στη περίπτωση του κελιού **E4** ή τιμή που υπολογίζεται είναι 27,5 αφού η Τσακαλίδου Ευτέρπη, η οποία έχει 5 ώρες επιπλέον των 40, το ποσό που θα πρέπει να της καταβληθεί είναι 27,50 € (5 x 5,5 €/ώρα).

Σύνταξη Συναρτήσεων και Επεξήγηση του Κώδικα

Όπως είδαμε προηγουμένως, μία συνάρτηση ορίζεται σε μία λειτουργική μονάδα κώδικα. Θα πρέπει πάντα να έχουμε υπόψη μας, ότι οι διαδικασίες εκτελούν κάτι, σε αντίθεση με τις συναρτήσεις που επιστρέφουν μια τιμή. Μοιάζουν δηλαδή, με τις συναρτήσεις του Excel.

Η σύνταξη για τη δήλωση μιας συνάρτησης έχει ως εξής:

```
Function όνομα [(ορίσματα)] [As τύπος_επιστροφής]  
    [εντολές]  
    [όνομα = τιμή_επιστροφής]  
    [Exit Function]  
    [εντολές]  
    [όνομα = τιμή_επιστροφής]  
End Function
```

Ο ορισμός της ξεκινάει με την πρόταση **Function** και ακολουθεί το όνομα της συνάρτησης μ' ένα ζευγάρι παρενθέσεων. Το όνομα της συνάρτησης ακολουθεί τους κανόνες ονομασίας διαδικασιών. Μέσα στις παρενθέσεις υπάρχουν συνήθως παράμετροι, χωρισμένες με κόμμα μεταξύ τους, οι οποίες δέχονται τις τιμές των ορισμάτων. Τα ορίσματα, όπως είναι γνωστό, είναι τιμές ή αναφορές κελιών που τροφοδοτούν τη συνάρτηση. Με τη λέξη-κλειδί **As** δηλώνουμε τον τύπο δεδομένων της τιμής επιστροφής. Αν και η δήλωση αυτή είναι προαιρετική, συνιστάται η χρήση της. Η πρόταση **Exit Function** τερματίζει πρόωρα μια συνάρτηση. Πριν τον πρόωρο τερματισμό ή πριν τον κανονικό τερματισμό της με **End Function**, πρέπει να εκχωρήσουμε στο όνομα της συνάρτησης την τιμή που θέλουμε να επιστρέψουμε.

Μια συνάρτηση μπορεί όπως ακριβώς και μια διαδικασία, να είναι ιδιωτική ή δημόσια. Αν δεν καθορίσουμε με τους προσδιοριστές **Private** ή **Public** πριν τη λέξη-κλειδί **Function** την εμβέλεια της, θα είναι δημόσια.

Μια συνάρτηση μπορεί να κληθεί από ένα φύλλο εργασίας, όπως στην περίπτωση μας ή μέσα από μια διαδικασία με τον παρακάτω τρόπο:

Synolo = Υπερωρίες(90, 10)

Τα ορίσματα μπορούν να είναι κυριολεκτικά αριθμητικά ή μεταβλητές.

Ας δούμε αναλυτικά τον κώδικα της συνάρτησης *Υπερωρίες*:

Η συνάρτησή μας δέχεται δύο τιμές, τις οποίες και χρησιμοποιεί για τον υπολογισμό του ποσού των υπερωριών.

Η πρόταση

If ώρες > 40 Then

ελέγχει αν η παράμετρος ώρες έχει τιμή μεγαλύτερη του 40. Στην περίπτωση που αυτό αληθεύει, υπολογίζει το ποσό των υπερωριών

Υπερωρίες = (ώρες - 40) * 0.5 * ωρομίσθιο

(υπερωρίες επί το 50% του ωρομισθίου). Το αποτέλεσμα το εκχωρεί σε μία μεταβλητή που έχει το ίδιο όνομα με τη συνάρτηση.

Στην περίπτωση που η παράμετρος ώρες **δεν είναι μεγαλύτερη του 40**, εκτελείται η εντολή μετά τη λέξη-κλειδί **Else**

Υπερωρίες = 0

Δηλαδή, στη μεταβλητή **Υπερωρίες** εκχωρείται η τιμή **μηδέν**.

Ο κώδικας μιας συνάρτησης τερματίζεται με την πρόταση **End Function**. Το αποτέλεσμα των υπολογισμών επιστρέφεται μέσω της μεταβλητής, που πρέπει υποχρεωτικά να έχει το ίδιο όνομα με τη συνάρτηση.

4.3.3 Παραδείγματα Προσαρμοσμένων Συναρτήσεων

Παράδειγμα συνάρτησης χωρίς ορίσματα

Στο παράδειγμα αυτό, θα δούμε μια συνάρτηση χωρίς ορίσματα, με το όνομα *Ημερομηνία*, που επιστρέφει την τρέχουσα ημερομηνία στη μορφή:

Ημέρα, ηη/μμ/εεεε

Ο τύπος της τιμής επιστροφής θα είναι αλφαριθμητικό (string). Πληκτρολογούμε τον ακόλουθο κώδικα σε μια λειτουργική μονάδα κώδικα:

Function Ημερομηνία As String

Dim Ημέρα As String

Select Case Weekday(Date)

Case 1

Ημέρα = "Κυριακή"

Case 2

Ημέρα = "Δευτέρα"

Case 3

Ημέρα "Τρίτη"

Case 4

Ημέρα "Τετάρτη"

Case 5

Ημέρα = "Πέμπτη"

Case 6

Ημέρα = "Παρασκευή"

Case 7

Ημέρα = Σάββατο"

End Select

Ημερομηνία = Ημέρα & ", " & Date

End Function

Η συνάρτηση **Date** επιστρέφει την ημερομηνία του συστήματος στη μορφή ηη/μμ/εεεε. Για να πάρουμε την ημέρα, χρησιμοποιούμε τη συνάρτηση **Weekday**, η οποία επιστρέφει έναν ακέραιο αριθμό σύμφωνα με τον παρακάτω πίνακα:

Σταθερά	Τιμή	Περιγραφή
vbSunday	1	Κυριακή
vbMonday	2	Δευτέρα
vbTuesday	3	Τρίτη
vbWednesday	4	Τετάρτη
vbThursday	5	Πέμπτη
vbFriday	6	Παρασκευή
vbSaturday	7	Σάββατο

Στη συνέχεια, με μια δομή Select - Case παίρνουμε την ημέρα σαν κυριολεκτικό αλφαριθμητικό και συνενώνουμε τις δύο αυτές τιμές και τις εκχωρούμε στο όνομα της συνάρτησης.

Για να καλέσουμε τη συνάρτηση από ένα φύλλο εργασίας του Excel, εισάγουμε στο επιθυμητό κελί τον τύπο:

=Ημερομηνία ()

και πατάμε το πλήκτρο **[Enter]**.

	A	B	C
1			
2			
3	=Ημερομηνία()		20/11/2013
4			
5	Ορίσματα συνάρτησης Ημερομηνία = "Τρίτη, 26/11/2013" Δεν υπάρχει διαθέσιμη βοήθεια. Αυτή η συνάρτηση δεν δέχεται ορίσματα. Αποτέλεσμα = Τρίτη, 26/11/2013 Βοήθεια για αυτήν τη συνάρτηση OK Άκυρο		
6			
7			
8			
9			
10			
11			
12			

Παράδειγμα συνάρτησης με δύο ορίσματα

Η συνάρτηση του παραδείγματος αυτού δέχεται δύο ορίσματα. Το πρώτο όρισμα είναι ένα ποσό και το δεύτερο μια αριθμητική ποσότητα που αναπαριστά την κατηγορία του προϊόντος. Ανάλογα με την κατηγορία υπολογίζει και το ποσό του ΦΠΑ.

Function ΦΠΑ(Ποσό, Κατηγορία) As Currency

Dim Συντελεστής As Single

Select Case Κατηγορία

Case 1

Συντελεστής = 0.23

Case 2

Συντελεστής = 0.16

Case 3

Συντελεστής = 0.13

End Select

ΦΠΑ = Ποσό * Συντελεστής

End Function

Για να καλέσουμε τη συνάρτηση από ένα φύλλο εργασίας του Excel, εισάγουμε στο επιθυμητό κελί τον τύπο:

=ΦΠΑ(100,2)

και πατάμε το πλήκτρο **[Enter]**.

Το παράδειγμα που ακολουθεί δείχνει μια διαδικασία, η οποία καλεί τη συνάρτηση ΦΠΑ και εμφανίζει την τελική τιμή με τη χρήση της συνάρτησης MsgBox:

```
Sub TestFPA()
```

```
    Dim Price As Currency, Fprice As Currency
```

```
    Dim Category As Integer
```

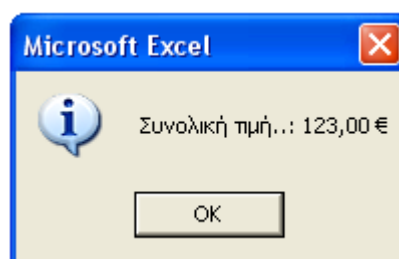
```
    Price = InputBox("Δώσε την τιμή του προϊόντος...: ")
```

```
    Category = InputBox("Δώσε την κατηγορία του προϊόντος...: ")
```

```
    Fprice = Price + ΦΠΑ(Price, Category)
```

```
    MsgBox "Συνολική τιμή...: " & Format(Fprice, "#,##0.00 €"), vbInformation
```

```
End Sub
```



Η συνάρτηση **Format** αλλάζει τη μορφή εμφάνισης μιας έκφρασης, ανάλογα με το αλφαριθμητικό μορφοποίησης που δίνουμε.

Η μορφή **"#,##0.00 €"** καθορίζει να εμφανιστεί η τιμή με διαχωριστή χιλιάδων, ένα τουλάχιστον ψηφίο πριν την υποδιαστολή και στη συνέχεια, δύο ψηφία μετά και το σύμβολο του Ευρώ (€). Η συνάρτηση μπορεί επίσης να χρησιμοποιηθεί για τη μορφοποίηση ημερομηνιών, ώρας και αλφαριθμητικών.

4.3.4 Δημιουργία και Φόρτωση Προσθέτων

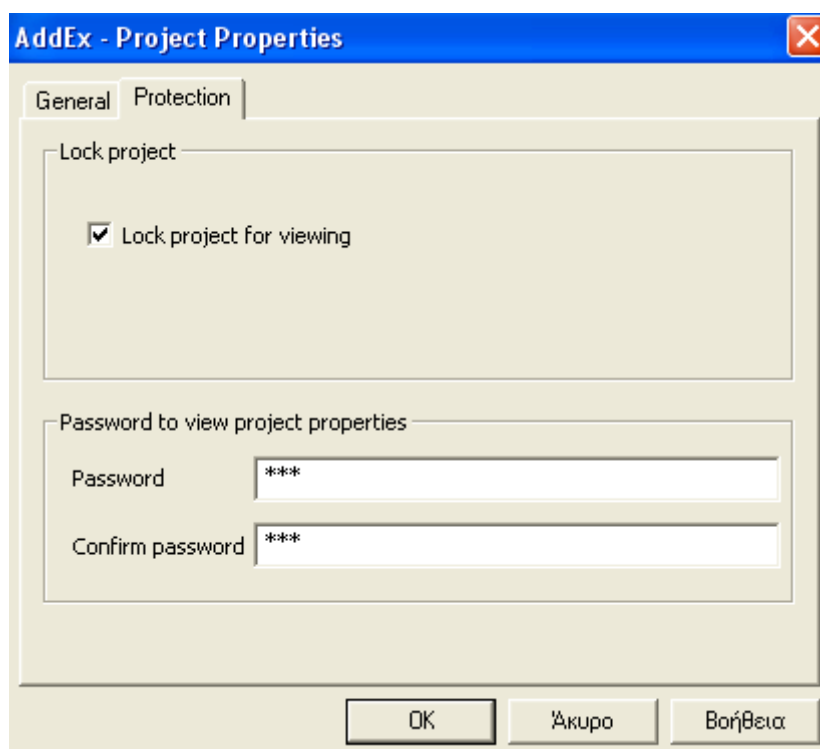
Τα πρόσθετα είναι προγράμματα που προσθέτουν επιπλέον δυνατότητες στο Microsoft Excel. Για παράδειγμα, το "Πακέτο εργαλείων ανάλυσης" παρέχει ένα σύνολο βοηθημάτων για πολύπλοκες αναλύσεις δεδομένων. Για την ανάπτυξη προσθέτων δεν απαιτείται ειδικό λογισμικό. Οποιοδήποτε βιβλίο εργασίας του Excel που περιέχει μακροεντολές και προσαρμοσμένες συναρτήσεις, μπορεί να μετατραπεί σε πρόσθετο.

Τα πρόσθετα είναι πολύ χρήσιμα, κυρίως για δύο λόγους:

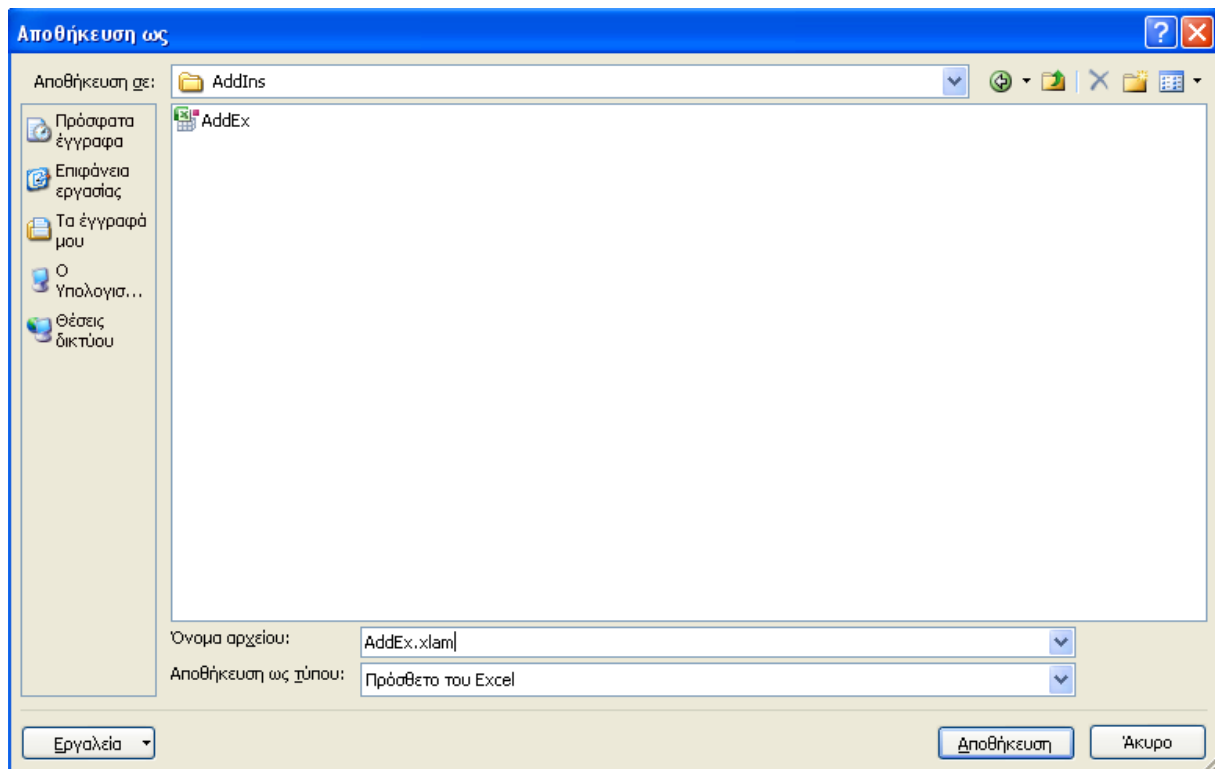
- ✓ Απλοποιούν την πρόσβαση στις συναρτήσεις άλλου βιβλίου εργασίας. Για να καλέσουμε μια συνάρτηση που βρίσκεται σε μια λειτουργική μονάδα άλλου αρχείου, πρέπει πριν το όνομά της να αναφέρουμε και το όνομα του αρχείου. Αν όμως η συνάρτηση βρίσκεται σ' ένα φορτωμένο πρόσθετο, δεν χρειάζεται να αναφέρουμε το όνομά του.
- ✓ Προστατεύουν το κώδικά μας μέσω κωδικού πρόσβασης, αποτρέποντας έτσι την αντιγραφή του.

Για να δημιουργήσουμε ένα πρόσθετο με τις συναρτήσεις της προηγούμενης ενότητας, ακολουθούμε τα παρακάτω βήματα:

- Ανοίγουμε το βιβλίο εργασίας **Προσαρμοσμένες Συναρτήσεις**.
- Πατάμε το συνδυασμό πλήκτρων **[Alt]+[F11]**, για να εμφανίσουμε την Επεξεργασία της Visual Basic.
- Στο παράθυρο έργων κάνουμε δεξί κλικ στο όνομα του έργου και από το μενού συντόμευσης επιλέγουμε **VBAProject Properties**.
- Εμφανίζεται το παράθυρο διαλόγου "Project Properties", όπου στην καρτέλα "General" πληκτρολογούμε ένα όνομα και μια περιγραφή για το έργο.
- Κάνουμε κλικ στην καρτέλα "Protection" και επιλέγουμε το πλαίσιο ελέγχου **lock Project for viewing**. Στα δύο πλαίσια κειμένου, που βρίσκονται στο κάτω τμήμα της καρτέλας πληκτρολογούμε έναν κωδικό πρόσβασης και κάνουμε κλικ στο κουμπι **OK**.

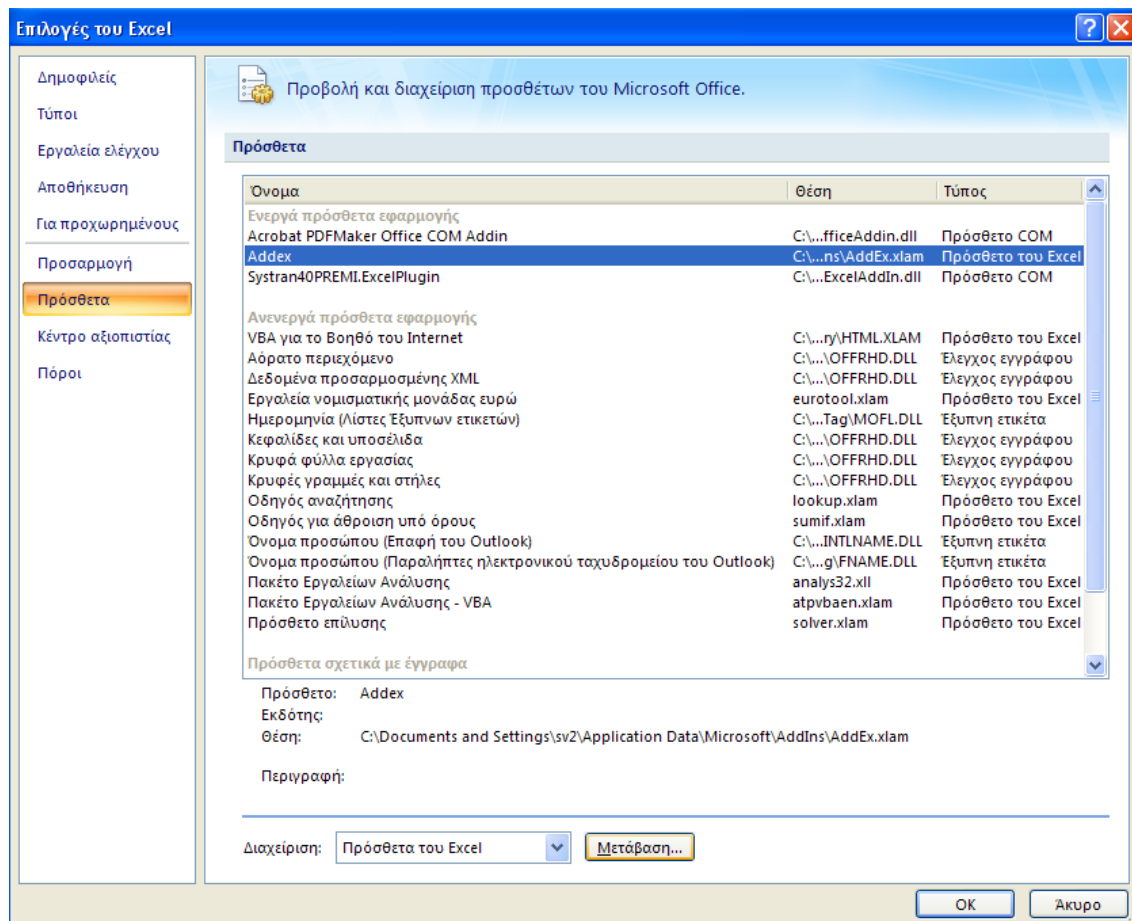


- Κλείνουμε το VBE και στο Excel αποθηκεύουμε το βιβλίο εργασίας. Στη συνέχεια, κάνουμε κλικ στο **Κουμπι Office** και από το μενού που εμφανίζεται, επιλέγουμε την εντολή **Αποθήκευση ως**. Εμφανίζεται το ομώνυμο παράθυρο διαλόγου.
- Από την αναπτυσσόμενη λίστα "Αποθήκευση ως" επιλέγουμε **Πρόσθετο του Excel**. Αν θέλουμε να δημιουργήσουμε ένα πρόσθετο για προηγούμενη έκδοση, επιλέγουμε **Πρόσθετο του Excel 97-2003**.
- Στο πλαίσιο "Όνομα αρχείου" πληκτρολογούμε ένα όνομα για το πρόσθετο, για παράδειγμα **AddEx** και κάνουμε κλικ στο κουμπι **Αποθήκευση**.

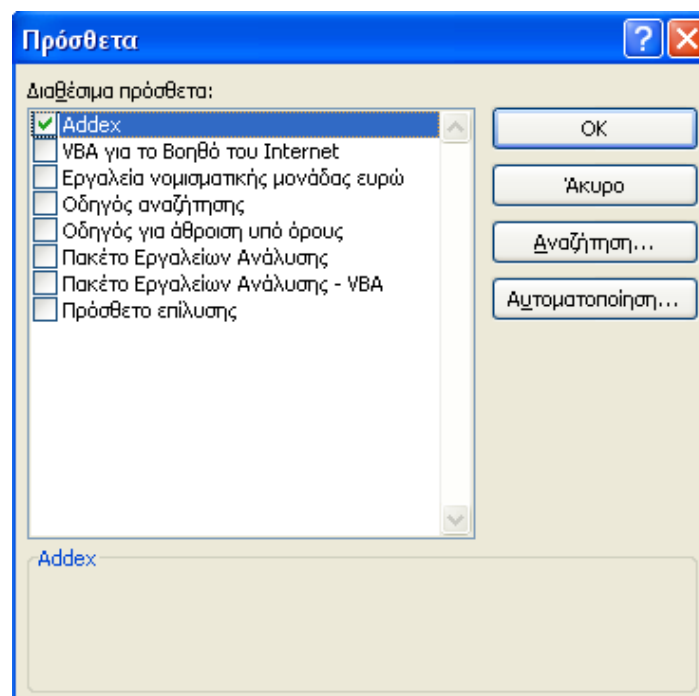


Για να φορτώσουμε ένα πρόσθετο του Excel, ακολουθούμε τα παρακάτω βήματα:

- Ανοίγουμε ένα οποιοδήποτε βιβλίο εργασίας.
- Κάνουμε κλικ στο **Κουμπί Office** και στο μενού που ανοίγει κάνουμε κλικ στο κουμπί **Επιλογές του Excel**.
- Από το αριστερό τμήμα του παραθύρου επιλέγουμε την κατηγορία **Πρόσθετα**.
- Από την αναδιπλούμενη λίστα **Διαχείριση**, που βρίσκεται στο κάτω δεξιό τμήμα του παραθύρου, επιλέγουμε **Πρόσθετα του Excel** και στη συνέχεια, κάνουμε κλικ στο κουμπί **Μετάβαση**.

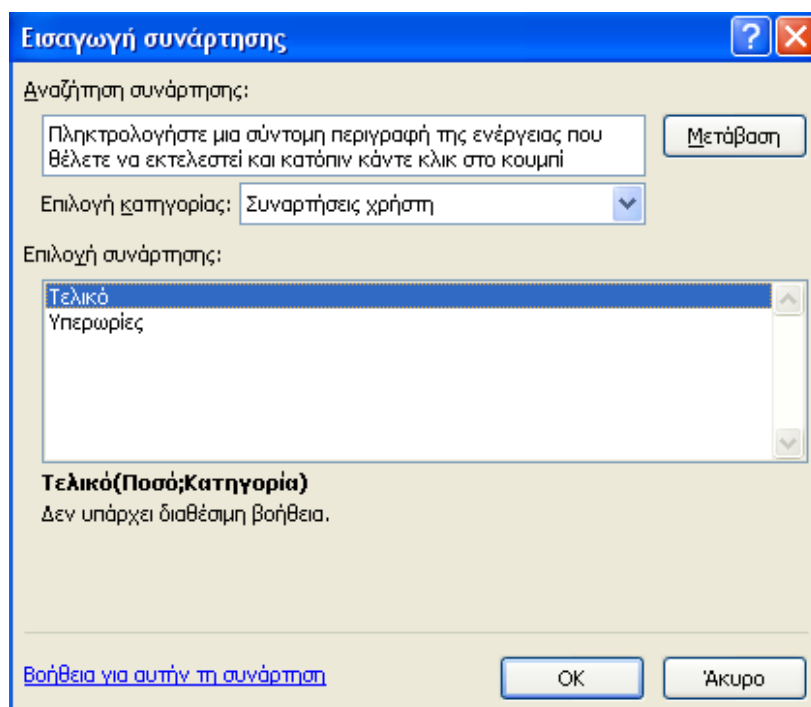


- Το παράθυρο διαλόγου "Πρόσθετα" εμφανίζει μια λίστα με τα διαθέσιμα πρόσθετα.
- Στη λίστα **Διαθέσιμα πρόσθετα**, επιλέγουμε το πλαίσιο ελέγχου που βρίσκεται αριστερά από το πρόσθετο που θέλουμε να φορτώσουμε.



- Κάνουμε κλικ στο κουμπί **OK**, για κλείσει το παράθυρο διαλόγου και να ενεργοποιηθεί το πρόσθετο.

Οι συναρτήσεις που περιέχει ένα πρόσθετο μπορούν να προσπελαστούν όπως και οι ενσωματωμένες συναρτήσεις του Excel. Θα τις βρούμε στο παράθυρο διαλόγου "Εισαγωγή συνάρτησης" στην κατηγορία **Συναρτήσεις χρήστη**.



- ☞ Το Excel αποθηκεύει εξ ορισμού τα πρόσθετα στον υποφάκελο **Application Data\Roaming\Microsoft\Πρόσθετα** του προσωπικού φακέλου του χρήστη.