**Programming with B4X**

Lesson 6 – From Designer to Code

Version 1.0, February 2021

# Lesson 6 – From Designer to Code

* Class\_Globals
* Variables and Subs
* Passing Values to Code
* Events
* Attributes

What students should know

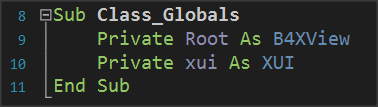
**2h**

Η σχεδίαση της εμφάνισης της εφαρμογής στον Designer είναι το πρώτο βήμα στα στάδια κατασκευής. Συχνά οι προγραμματιστές επιστρέφουν σε αυτόν προκειμένου να ξανασχεδιάσουν, να διορθώσουν ή να προσθέσουν επιμέρους πληροφορίες.

Όταν η οθόνη είναι έτοιμη ο προγραμματιστής περνά στο επόμενο στάδιο αυτό του προγραμματισμού των λειτουργιών. Δηλαδή όλα όσα στοιχεία συμπεριέλαβε στην σχεδίαση να αποκτήσουν λειτουργικότητα. Με άλλα λόγια τα πλαίσια κειμένου να μπορούν να καταγράφουν δεδομένα, τα κουμπιά να ενεργοποιούν λειτουργίες, οι λίστες να εμφανίζουν δεδομένα κλπ.

## Class\_Globals

Στην αρχή του κώδικα στην καρτέλα B4XMainPage υπάρχει ένα σύνολο δηλώσεων μεταβλητών μεταξύ Sub Class\_Globals και ΕndSub.

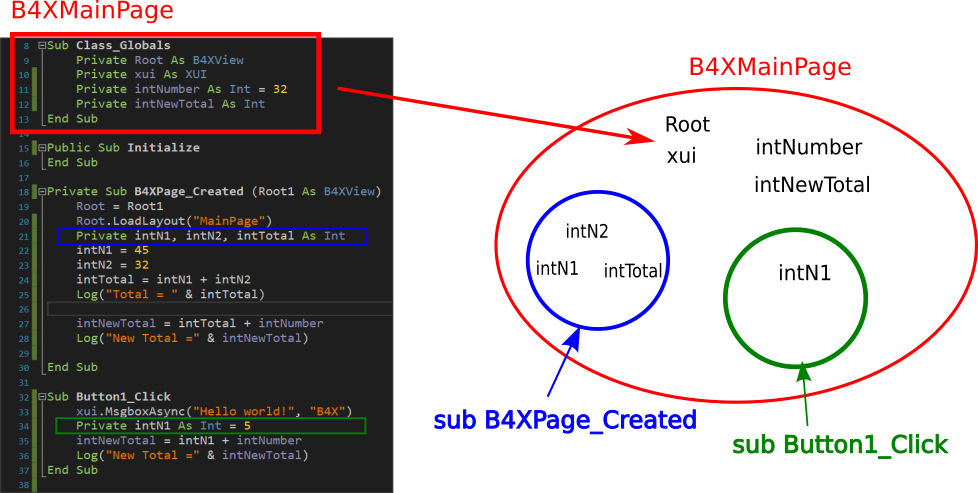
Όπως ήδη ειπώθηκε στο 3ο Μάθημα Sub είναι ένα σύνολο κώδικα που εκτελεί μια συγκεκριμένη λειτουργία. Η λειτουργία του Class\_Globals είναι να συγκεντρώνει τις δηλώσεις των μεταβλητών οι οποίες θέλουμε να είναι γνωστές σε όλο τον κώδικα της καρτέλας B4XMainPage, δηλαδή σε κάθε υποπρόγραμμα.

Picture Sub Class\_Globals

Επιπρόσθετα, αν μία δήλωση μεταβλητής ξεκινά με το public θα είναι διαθέσιμη και από άλλες «καρτέλες» προγραμμάτων. Στη συνέχεια θα μάθουμε ότι αυτές τις καρτέλες θα τις ονομάζουμε τμήματα (Modules) ή κλάσεις (Classes).

## Μια βαθύτερη ματιά στη χρήση των μεταβλητών.

Στον παρακάτω κώδικα βλέπετε τρία υποπρογράμματα



Picture Variables and Range

Οι μεταβλητές intNumber, intNewTotal, Root και xui που είναι δηλωμένες μέσα στο Class\_Globals «ζουν» μέσα στο Module B4XMainPage. Για αυτό το λόγο μπορούν να είναι διαθέσιμες σε όλα τα υποπρογράμματα που θα τις χρησιμοποιήσουν.

Αντίθετα, οι μεταβλητές intN1, intN2, intTotal «ζουν» μέσα στο υποπρόγραμμα B4XPage\_Created και για αυτό το λόγο κανένα άλλο υποπρόγραμμα δεν μπορεί να τις χρησιμοποιήσει. Παράλληλα, η μεταβλητή **intN1** που ζει μέσα στο **Button1\_Click** **δεν είναι η ίδια** με αυτή στο **B4XPage\_Created** έχει δικό της χώρο στη μνήμη και μπορεί να έχει και το ίδιο όνομα και την χρησιμοποιεί μόνο το Button1\_Click.

**Teachers tip**

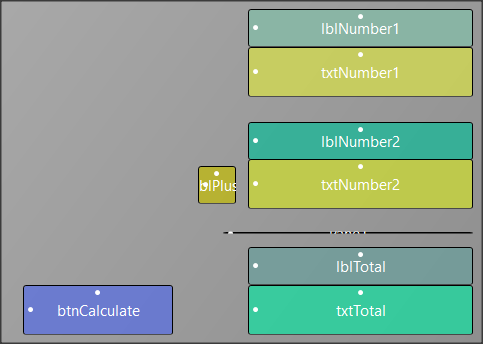
Η χρήση και το εύρος των μεταβλητών είναι κάτι που μπερδεύει τους νέους προγραμματιστές. Ανάλογα με την τάξη σας προτείνεται να γίνουν παραδείγματα εξοικείωσης στη χρήση τους.



## Passing Values to Code

Η οθόνη που έχετε ήδη προετοιμάσει περιέχει αντικείμενα που για να τα χρησιμοποιήσετε μέσα στο πρόγραμμα πρέπει να τα δηλώσετε ως μεταβλητές στο Class\_Globals.

Στο παράδειγμα με την πρόσθεση των δύο αριθμών του προηγούμενου μαθήματος κάποια από τα αντικείμενα της οθόνης δεν είναι απαραίτητο να τα συμπεριλάβετε στον κώδικα.



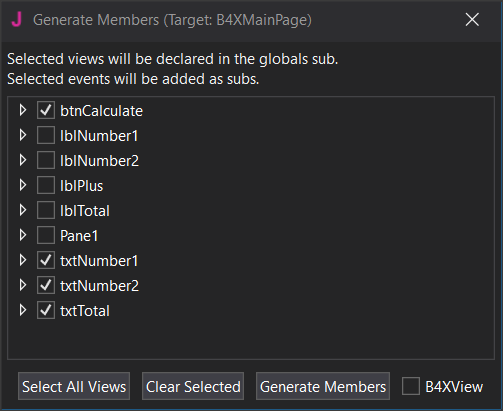
Picture Example 1 Designer View

Έτσι όλα τα στοιχεία “labels” στην συγκεκριμένη εφαρμογή δεν είναι απαραίτητο να τα διαχειριστείτε με κώδικά όπως επίσης και το στοιχείο pane. Ενώ τα αντικείμενα button και textFields πρέπει να τα προσθέσουμε ώστε να προγραμματίσουμε πάνω σε αυτά λειτουργίες.

Υπάρχουν δύο τρόποι να εισάγετε τα αντικείμενα σας στον κώδικα. Ο πρώτος γίνεται εύκολα μέσα από τον designer με τις παρακάτω λειτουργίες:

Από το μενού Tools επιλέξτε Generate Members

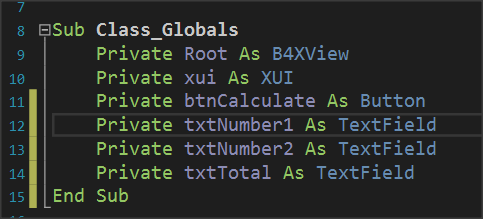
Στην οθόνη Generate Members απλά κάνετε κλικ στα αντικείμενα



Picture 4 Generate Members

* btnCalculate
* txtNumber1
* txtNumber2
* txtTotal

και στη συνέχεια κλικ στο κουμπί Generate Members.



Picture 5 Class\_Globals

Ο κώδικας σας στο υποπρόγραμμα Class\_Globals θα ενημερωθεί αυτόματα με τις μεταβλητές.

**Remember**

Κάθε αντικείμενο που εισάγουμε είναι κάποιου συγκεκριμένου τύπου όπως και τα είδη των μεταβλητών.



Δεύτερη λύση εισαγωγής είναι να γράψετε μόνοι σας τις μεταβλητές δίνοντας προσοχή ώστε τα ονόματα των αντικειμένων στην οθόνη σας να είναι ίδια με αυτά που γράφετε ως μεταβλητή.

## Events

Αφού έχετε δηλώσει τις μεταβλητές των αντικειμένων το τελικό στάδιο είναι να ενεργοποιήσετε τις λειτουργίες της φόρμας.

Αυτό εξαρτάται από τον τρόπο που έχετε αποφασίσει να λειτουργεί η κάθε εφαρμογή. Στο παράδειγμα με τους δύο αριθμούς υπάρχει ένα πλήκτρο λειτουργίας που ονομάζεται Calculate και είναι αυτό που θα ενεργοποιήσει την πρόσθεση καθώς και την εμφάνιση του αποτελέσματος στην οθόνη μας.

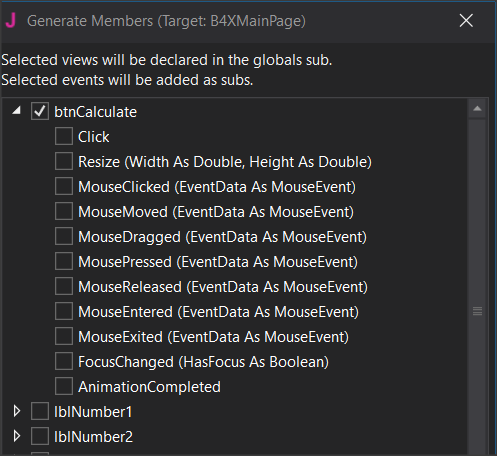
Το έναυσμα της λειτουργίας γίνεται με μια διαδικασία που ονομάζεται «**Event**» ή γεγονός. Ο προγραμματιστής πρέπει να ανιχνεύσει το γεγονός του πατήματος του πλήκτρου Calculate και όταν γίνει αυτό τότε να κάνει τους σχετικούς υπολογισμούς και την εμφάνιση του αποτελέσματος.

**Remember**

Υπάρχουν εκατοντάδες διαφορετικά γεγονότα που συμβαίνουν σε μία εφαρμογή. Ο προγραμματιστής καθορίζει τον τρόπο που θα αντιδρά το πρόγραμμα σε κάθε ένα από αυτά.

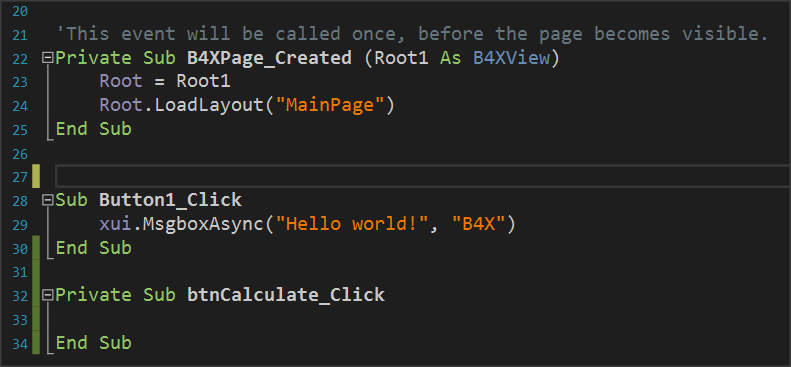


Η ανίχνευση του event είναι εύκολη και απλά πρέπει να δημιουργήσετε ένα νέο υποπρόγραμμα που θα έχει το όνομα του event. Αυτό μπορεί να γίνει ταυτόχρονα με την δήλωση των μεταβλητών των αντικειμένων μας μέσα από το designer.



Picture 6 Setting Event

Απλά ανοίγετε την λίστα του btnCalculate και θα εμφανιστούν το σύνολο των events που είναι διαθέσιμα για το συγκεκριμένο πλήκτρο. Κάντε απλά κλικ στο «Click» και «Generate Members».

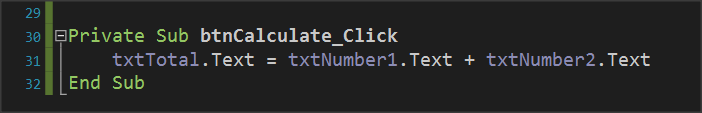


Picture 7 The Click Event

Ήδη έχει εμφανιστεί το υποπρόγραμμα για το event **btnCalculate\_Click.** Μέσα σε αυτό θα γράψετε τον κώδικα του προγράμματος για να ολοκληρωθεί. Πριν καλό είναι να διαγράψετε το υποπρόγραμμα Button1\_Click καθώς δεν χρειάζεται στην εφαρμογή μας.

## Γράφοντας κώδικα στο Event

Μέσα σε ένα υποπρόγραμμα ενός Event συνήθως γίνονται όλες οι λειτουργίες που σχετίζονται με αυτό. Στο παράδειγμα με την πρόσθεση των δύο αριθμών οι λειτουργίες αυτές είναι η πρόσθεση των αριθμών καθώς και η εμφάνιση του αποτελέσματος.



Picture 8 Calculating textFields

Ουσιαστικά και τα δύο επιτυγχάνονται με μία μόνο εντολή. Αυτή είναι η εντολή της “Picture 8 Calculating textFields” και περιγράφεται ως εξής:

*Το περιεχόμενο του textField txtTotal είναι ίσο με τα περιεχόμενα του txtNumer1 και του txtNumber2.*

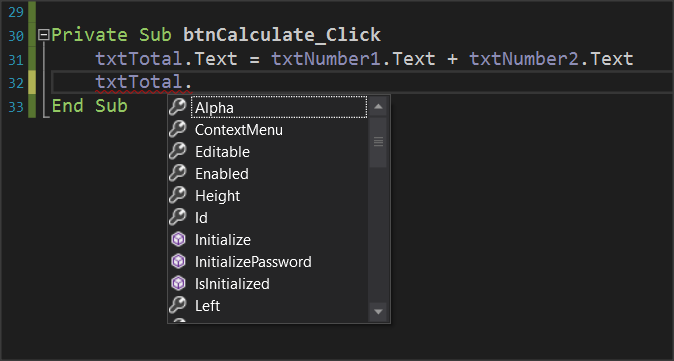
### Ιδιότητες

Κάθε αντικείμενο που εισάγετε στον κώδικά σας έχει ένα πλήθος από διαφορετικές ιδιότητες. Για παράδειγμα ιδιότητες μπορεί να είναι το χρώμα, το μέγεθος, η θέση το περιεχόμενο όπως έχουν περιγραφεί και στο προηγούμενο μάθημα. Οι ιδιότητες αυτές μπορούν να δώσουν πληροφορίες ή να αλλάξουν ζητήματα εμφάνισης. Για παράδειγμα η ιδιότητα txtNumber1.text παρέχει την πληροφορία για το περιεχόμενο του αντικειμένου txtNumber1 ή μπορεί να θέσει μία τιμή ως περιεχόμενο ανάλογα με τον τρόπο χρήσης της. Η πληροφορία αυτή είναι τύπου string.

**Remember**

Μπορείτε να κάνετε πράξεις με περιεχόμενα string όταν αυτά περιγράφουν αριθμούς. Στην περίπτωση αυτή γίνεται εσωτερική μετατροπή σε αριθμούς από την γλώσσα

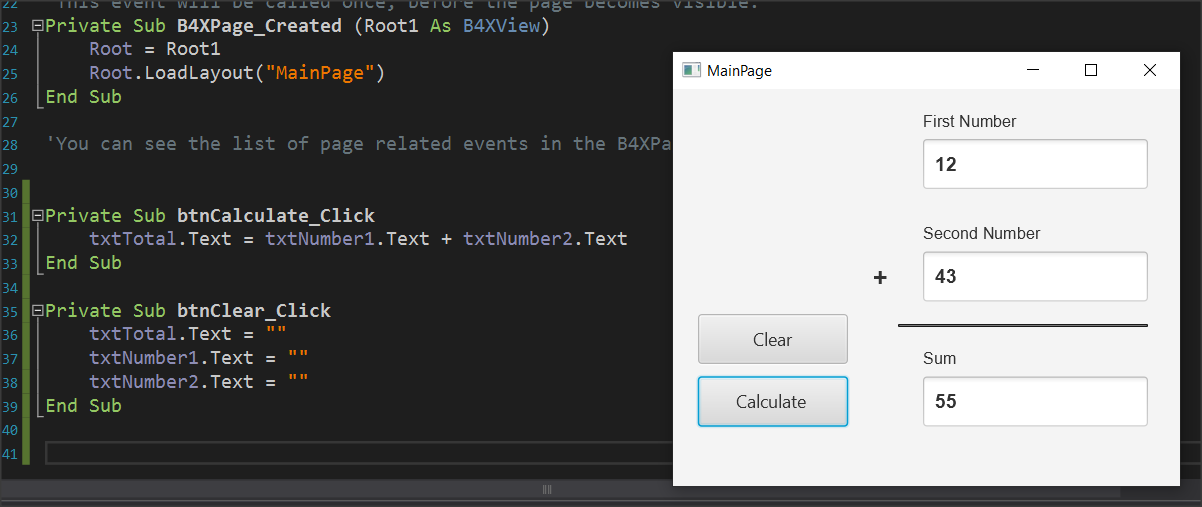




Picture 9 Ιδιότητες Αντικειμένων

Ας υποθέσουμε τώρα ότι θέλουμε να δημιουργήσουμε και μία νέα λειτουργία στο παράδειγμα όπου ένα άλλο πλήκτρο θα καθαρίζει τη φόρμα για να γράψουμε νέους αριθμούς. Οι λειτουργίες που θα κάνετε είναι οι παρακάτω:

* Ανοίγετε το Designer και προσθέτετε ένα νέο button με όνομα π.χ. btnClear
* Το ορίζετε ως μεταβλητή στο Class\_Global
* Εισάγετε το γεγονός btnClear\_Click
* Ορίζετε τις ιδιότητες text των txtNumber1, txtNumber2, txtTotal σε “”

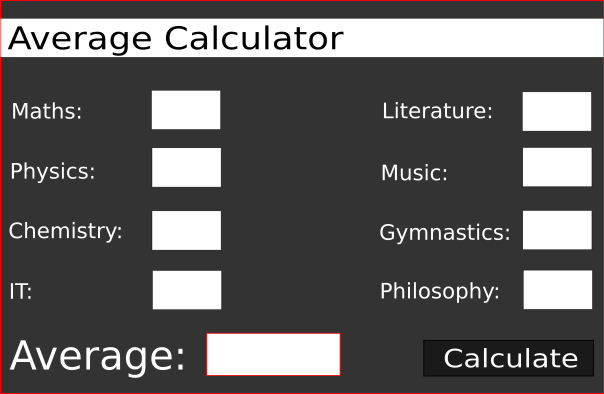


Picture 10 Clear Button and Code

Όταν θα πατάτε πλέον το πλήκτρο Clear θα καθαρίζει η φόρμα.

## Ασκήσεις

1. Να επεκτείνετε το παράδειγμα ώστε να εκτελούνται οι τέσσερεις πράξεις: Πρόσθεση, αφαίρεση, πολλαπλασιασμός και διαίρεση με το πάτημα κατάλληλου κουμπιού. Να προσθέσετε στο designer τα κατάλληλα αντικείμενα και στη συνέχεια να ολοκληρώσετε τον κώδικα.

2. Στην άσκηση 2 του προηγούμενου μαθήματος να συνεχίσετε και να ολοκληρώσετε την εφαρμογή ενεργοποιώντας τις λειτουργίες του.