# Lesson 8 – Subroutines

* What is a subroutine.
* Declaring a Sub
* Passing Values
* Returning Values from a sub

What students should know

**2h**

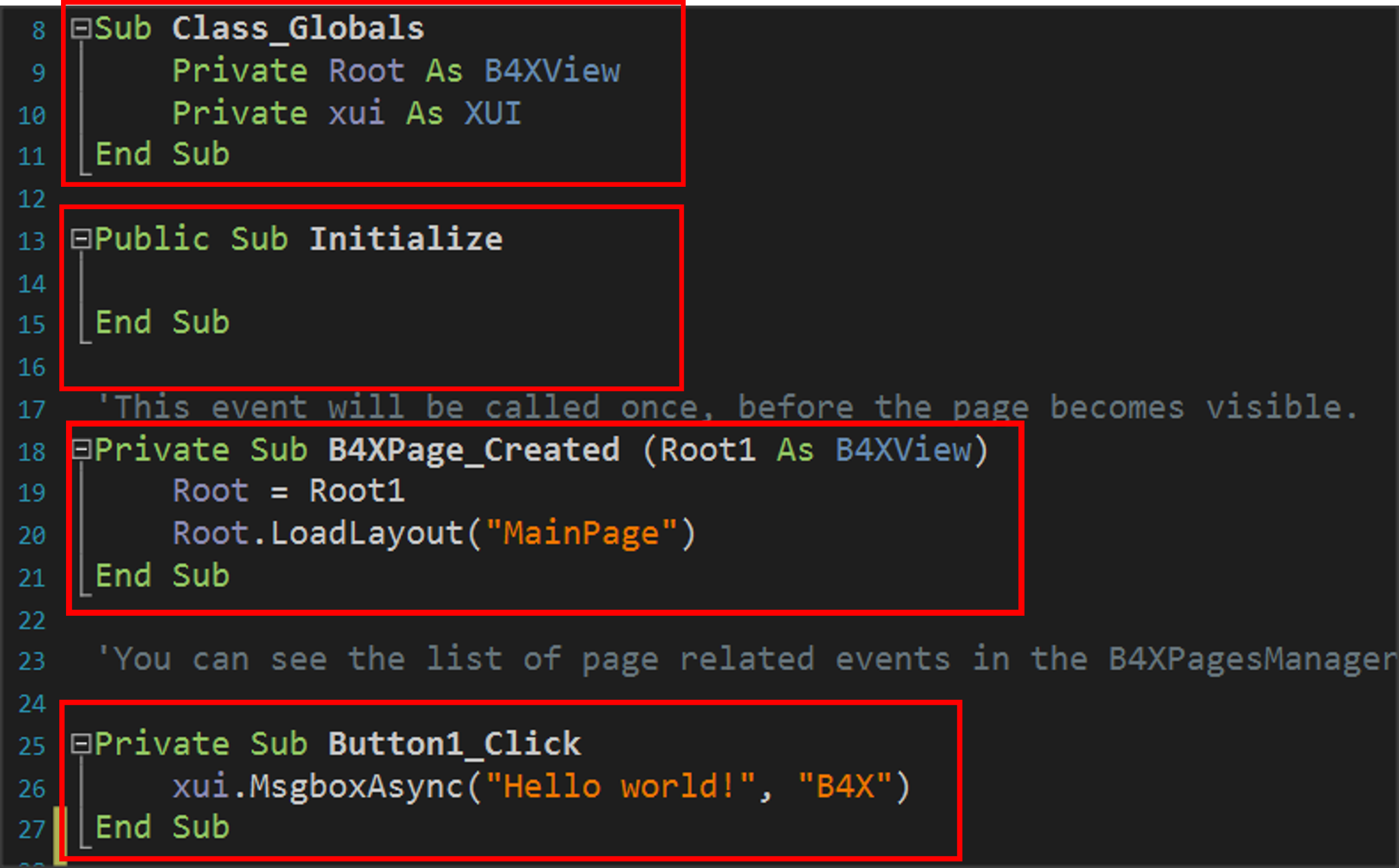
Στον προγραμματισμό, μια υπορουτίνα είναι μια ακολουθία εντολών προγράμματος που εκτελεί μια συγκεκριμένη εργασία, η οποία έχει μια ενιαία μορφή. Αυτή η μονάδα μπορεί στη συνέχεια να χρησιμοποιηθεί σε προγράμματα όπου θα πρέπει να εκτελεστεί αυτή η συγκεκριμένη εργασία.

Οι υπορουτίνες μπορούν να οριστούν μέσα σε προγράμματα ή ξεχωριστά σε ειδικά αρχεία που ονομάζονται βιβλιοθήκες και μπορούν να χρησιμοποιηθούν από πολλά προγράμματα. Σε διαφορετικές γλώσσες προγραμματισμού, μια υπορουτίνα μπορεί να ονομαστεί ρουτίνα, υποπρόγραμμα, συνάρτηση, μέθοδος ή διαδικασία.

Γενικά για την δημιουργία ενός υποπρογράμματος ο προγραμματιστής θα πρέπει να έχει υπόψη τα παρακάτω:

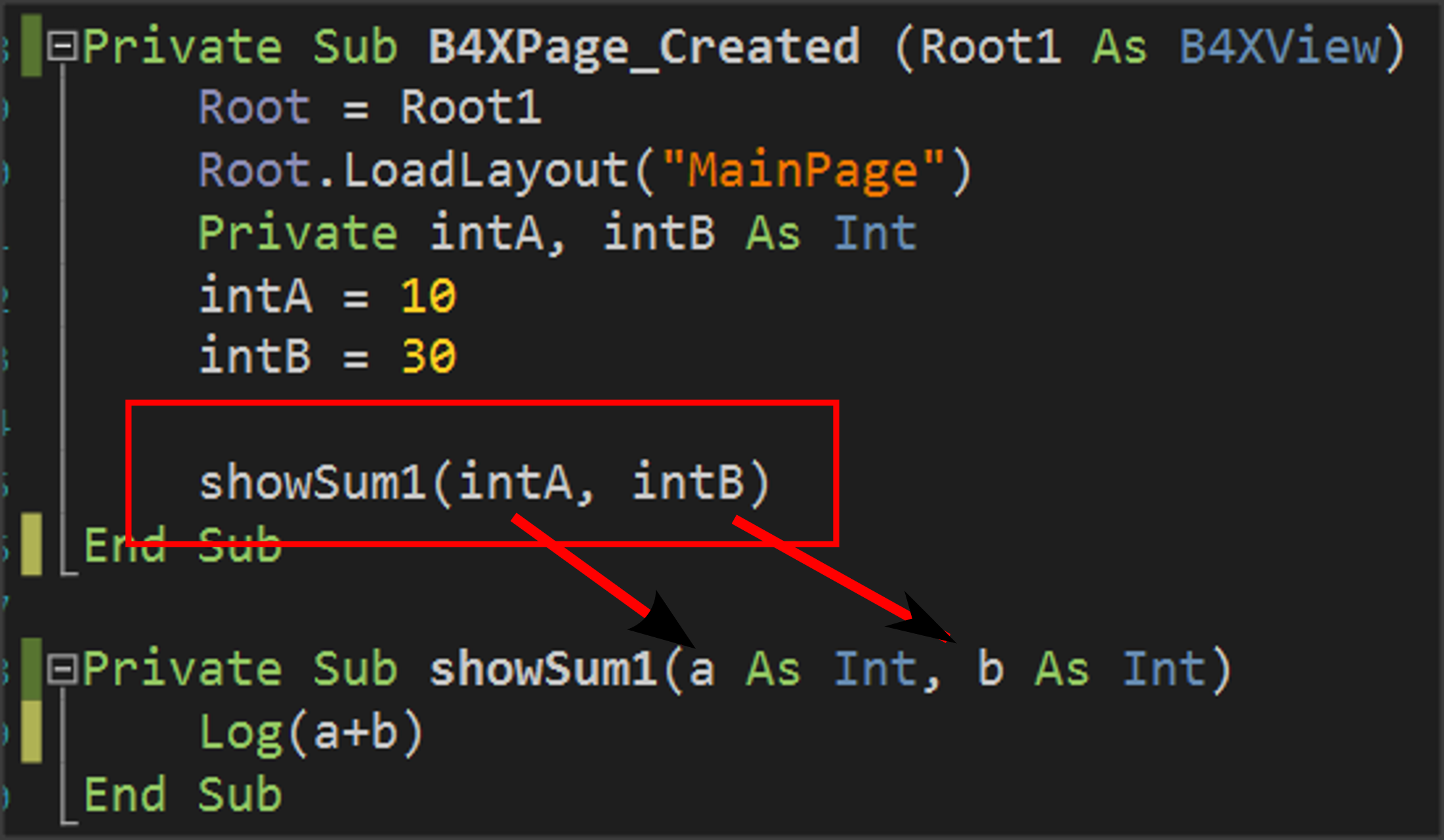
* Το υποπρόγραμμα θα πρέπει να επιτελεί μόνο μία εργασία.
* Να είναι σχετικά μικρό σε μέγεθος και ιδανικά όχι μεγαλύτερο από μία οθόνη ώστε να μπορεί να διαβάζεται εύκολα.
* Να έχει τέτοιο όνομα που να παραπέμπει στη λειτουργία του.

## Δημιουργία υποπρογράμματος στη B4J

Ήδη έχετε συναντήσει υποπρογράμματα στην B4J που έχει προετοιμάσει η γλώσσα. Τέτοια είναι το υποπρόγραμμα sub Class\_Globals, Initialize, B4XPage\_Created, Button1\_Click. Παρατηρήστε ότι και τα γεγονότα όπως το Button1\_Click είναι μία ρουτίνα με σκοπό την εξυπηρέτηση του γεγονότος.

### Παράδειγμα 1

Έστω, ότι χρειάζεται να υλοποιηθεί μία λειτουργία όπως για παράδειγμα η πρόσθεση δύο αριθμών που δίνει ο χρήστης. Σαν πρόγραμμα είναι πολύ απλό και γενικά δεν χρειάζεται υποπρόγραμμα για μία τέτοια λειτουργία. Εδώ θα χρησιμοποιήσουμε υποπρόγραμμα προκειμένου να καταλάβουμε τον τρόπο χρήσης και λειτουργίας του.

Στο πρόγραμμα της εικόνας έχουν δηλωθεί δύο ακέραιοι αριθμοί intA και intB, έχουν εκχωρηθεί τιμές και στη συνέχεια καλείται η ρουτίνα showSum1.

Αυτό γίνεται γράφοντας το όνομα της ρουτίνας (οποίο το αποφασίζει ο προγραμματιστής) και στη συνέχεια μέσα σε παρένθεση τις μεταβλητές τις τιμές των οποίων πρέπει να γνωρίζει για να λειτουργήσει.

Η συγγραφή του υποπρογράμματος γίνεται πριν ή μετά από το τρέχον υποπρόγραμμα:

* Ξεκινά πάντα με τη δήλωση Private που σημαίνει ότι είναι ένα υποπρόγραμμα που θα το γνωρίζει η τρέχουσα καρτέλα στην οποία βρισκόμαστε ή Public προκειμένου το υποπρόγραμμα να είναι γνωστό και σε άλλα τμήματα της εφαρμογής μας.
* Ακολουθεί η δήλωση Sub που σημαίνει subroutine (υπορουτίνα) και
* Μέσα σε παρένθεση αναφέρονται τα ονόματα των μεταβλητών που θα υποδεχθούν τα δεδομένα από το σημείο κλήσης.

Παρατηρήστε στην εικόνα ότι τα δεδομένα αυτά εισέρχονται με τη σειρά που γράφονται κατά την κλήση της ρουτίνας δηλαδή η τιμή της inA θα μπει στην μεταβλητή a και η τιμή της μεταβλητής intB θα μπει στην μεταβλητή b.

**Remember**

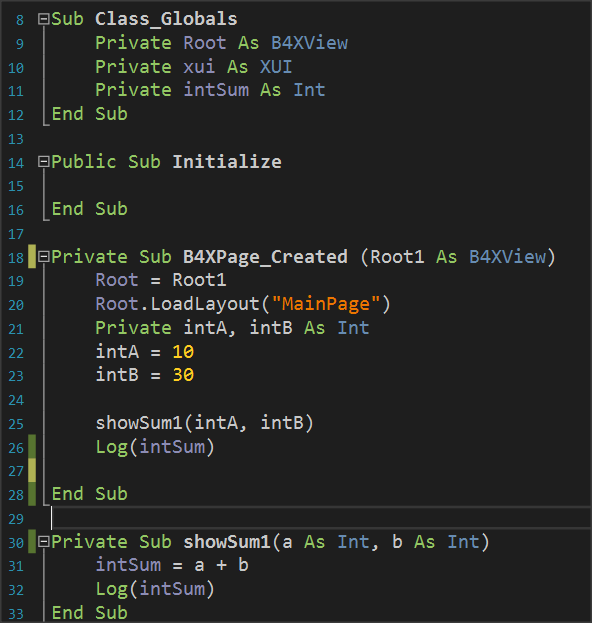
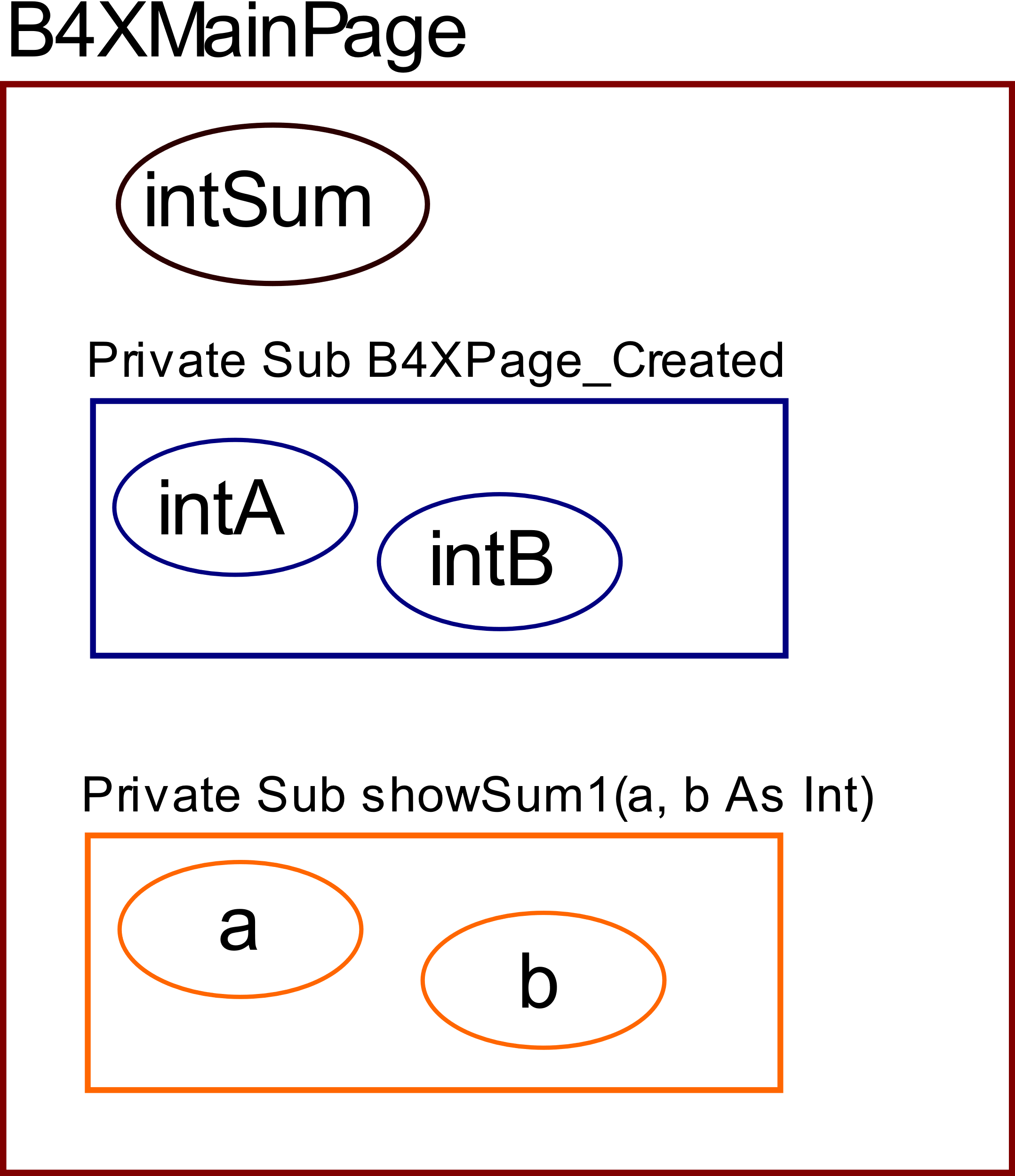
Οι μεταβλητές που ανταλλάσσονται μεταξύ υποπρογραμμάτων κατά τη διάρκεια της κλήσης τους λέγονται παράμετροι.



Το υποπρόγραμμα πλέον λειτουργεί με δεδομένα που περιέχονται μέσα στις παραμέτρους a, b και όχι τις μεταβλητές intA και IntB.

## Η μνήμη του υποπρογράμματος στην B4X

Κάθε υποπρόγραμμα διαθέτει δικό του χώρο στη μνήμη προκειμένου να αποθηκεύει οι μεταβλητές τους. Εξαίρεση στον κανόνα αυτό είναι το υποπρόγραμμα Class\_Globals του οποίου τα δεδομένα είναι γνωστά σε όλα τα υποπρογράμματα του B4XMainPage και μπορούν να αναφέρονται σε αυτά μόνο με το όνομά τους.



Από τον κώδικα της εικόνας μπορείτε να παρατηρήσετε ότι η μεταβλητή intSum είναι γνωστή και στα δύο υποπρογράμματα και χρησιμοποιείται απλά γράφοντας το όνομά της. Στην B4X οι μεταβλητές αυτές εμφανίζονται με διαφορετικό χρώμα για να μπορεί ο προγραμματιστής να τις διακρίνει εύκολα. Αντίθετα οι μεταβλητές που δηλώνονται μέσα στις υπόλοιπες μεταβλητές ζουν μόνο μέσα σε αυτές και δεν μπορεί ένα άλλο υποπρόγραμμα να τις χρησιμοποιήσει με το όνομα τους.

**Remember**

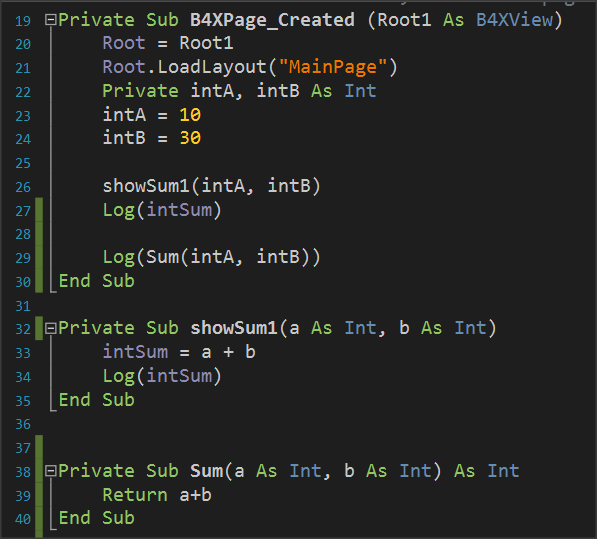
Οι μεταβλητές που δηλώνονται μέσα στο Class\_Globals είναι γνωστές σε όλα τα υποπρογράμματα και ονομάζονται καθολικές (**global**)

Οι μεταβλητές που δηλώνονται μέσα στα υποπρογράμματα ονομάζονται τοπικές (**local**) και δεν είναι γνωστές στα υπόλοιπα.



## Επιστροφή τιμής από υποπρόγραμμα.

Ένα υποπρόγραμμα μπορεί να επιστρέψει μία τιμή στον κώδικα που το καλεί. Αυτό γίνεται μέσα από το ίδιο το όνομα του υποπρογράμματος ως εξής:



Το πρόγραμμα πρέπει να δηλωθεί ως ένας τύπος μεταβλητής. Αυτός μπορεί να είναι οποιοσδήποτε απλός ή και σύνθετος τύπος που θα δούμε αργότερα.

Επιπλέον μέσα στο υποπρόγραμμα πρέπει να χρησιμοποιηθεί η εντολή return προκειμένου να επιστρέψει την τιμή που υπολογίστηκε.

Τέλος ο κώδικας που κάλεσε το υποπρόγραμμα δέχεται πίσω την αξία που υπολογίστηκε και μπορεί να τη χρησιμοποιήσει όπως κάθε μεταβλητή.

**Remember**

Συχνά οι ρουτίνες που επιστρέφουν τιμές στο κώδικα που τις καλεί ονομάζονται και **Συναρτήσεις** (Functions).



### Παράδειγμα 2

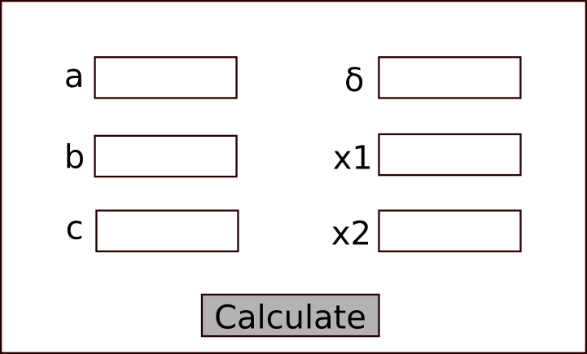
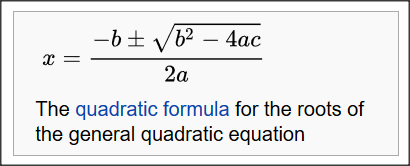
Να γίνει υποπρόγραμμα που να δέχεται 3 ακέραιες μεταβλητές και να επιστρέφει την μεγαλύτερη τιμή.

1. Εικόνα που περιέχει κείμενο

   Περιγραφή που δημιουργήθηκε αυτόματαΑρχικά δηλώνονται τρεις ακέραιες μεταβλητές (IntA, intB, intC) μέσα στο υποπρόγραμμα B4XPage\_Created.
2. Εκχωρούνται τιμές στις τρεις μεταβλητές.
3. Καλείται το υποπρόγραμμα sMax με παραμέτρους τις τρεις μεταβλητές.
4. Το υποπρόγραμμα εφαρμόζει τον αλγόριθμο του ΜΑΧ για τις τρεις μεταβλητές a, b, c και επιστρέφει την τιμή intM που υπολογίστηκε.
5. Τέλος εμφανίζεται στην οθόνη Log η μεγαλύτερη τιμή.

## Ασκήσεις

1. Να γίνει πρόγραμμα που να υπολογίζει το εμβαδό ενός κύκλου. Ο χρήστης πρέπει να εισάγει σε κατάλληλο textField την ακτίνα ενός κύκλου και στη συνέχεια με χρήση υποπρογράμματος να υπολογίζεται και να επιστρέφεται το εμβαδό. Τέλος, το πρόγραμμα εμφανίζει σε κατάλληλο textField το εμβαδό που υπολογίστηκε.
2. Να γίνει πρόγραμμα που να υπολογίζει την λύση της δευτεροβάθμιας εξίσωσης ax2 + bx + c = 0.
   1. Ο χρήστης της εφαρμογής πρέπει να εισάγει τους συντελεστές της εξίσωσης σε κατάλληλα TextFields.
   2. Το αποτέλεσμα να εμφανίζεται σε ένα, ή δύο textFields ανάλογα με την τιμή της διακρίνουσας.
   3. Ο υπολογισμός της διακρίνουσας πρέπει να γίνεται με υποπρόγραμμα που να επιστρέφει την αξία της.
   4. Δεν θα επιτρέπεται υπολογισμός της διακρίνουσας αν δεν συμπληρωθούν όλα τα πεδία των συντελεστών a, b, c. Σε αντίθετη περίπτωση να εμφανίζεται το μήνυμα: «Πρέπει να συμπληρωθούν όλα τα πεδία».

Σημείωση για την εμφάνιση του μηνύματος να χρησιμοποιηθεί η εντολή **xui.MsgboxAsync("Message","Title")**

1. Να γίνει πρόγραμμα που να χρησιμοποιεί τη χελώνα και να σχεδιάζει τετράγωνα με πλευρά που δίνεται από τον χρήστη. Για την σχεδίαση της χελώνας να δημιουργηθεί υποπρόγραμμα που να δέχεται την πλευρά και στη συνέχεια να σχεδιάζει το τετράγωνο ξεκινώντας από το σημείο που βρίσκεται ήδη η χελώνα και προχωρώντας δεξιόστροφα.
2. Ένας θέατρο διαθέτει τρεις κατηγορίες εισιτηρίων, Εξώστης Α, Εξώστης Β, Πλατεία. Κάθε εισιτήριο κοστίζει 20, 30, 40€. Να γίνει πρόγραμμα που:
   1. Να δηλώνει ο χρήστης τον κωδικό 1, 2, 3 που αντιπροσωπεύει αντίστοιχα τις κατηγορίες Εξώστης Α, Εξώστης Β και Πλατεία
   2. Το πλήθος των θέσεων που θέλει
   3. Να υπολογίζει την αξία των εισιτηρίων. Ο υπολογισμός της αξίας να γίνεται με υποπρόγραμμα που επιστρέφει την αξία και η οποία θα εμφανίζεται σε κατάλληλο textField.