# Lesson 18 – Complex Data Types, Files & Views

* Type declaration
* Arrays of Type
* List of Type
* Map of Type
* KVS Files
* CustomListView
* B4XComboBox

What students should know

**4**

Συχνά ένας προγραμματιστής θέλει να ομαδοποιήσει μεταβλητές που σχετίζονται μεταξύ τους. Για παράδειγμα ένας μαθητής έχει Όνομα, Επώνυμο, Διεύθυνση, Τηλέφωνο κλπ. Μπορεί φυσικά να δημιουργήσει ξεχωριστές μεταβλητές για κάθε διαφορετικό στοιχείο του μαθητή αλλά είναι πιο βολικό να δημιουργήσει ένα νέο τύπο δεδομένων που περιέχει όλες αυτές τις τιμές μαζί.

## Η δήλωση type

Ο τύπος δεδομένων περιεγράφηκε ονομάζεται **type** και δηλώνεται όπως παρακάτω:

Type Student(LastName As String, FirstName As String, Address As String, PhoneNumber As String)

Παρατηρήστε ότι πρώτα γράφεται η λέξη κλειδί type στη συνέχεια ένα όνομα για τον νέο τύπο που δημιουργείται και τέλος μέσα σε παρένθεση όλες οι μεταβλητές που συμπεριλαμβάνονται στον νέο τύπο. Η δήλωση type τοποθετείται πάντα μέσα στο Class\_Global και επίσης είναι πάντα Global τύπος.

Στο παράδειγμα του μαθητή η δήλωση type γίνεται:

**Sub** Class\_Globals

**Type** Student(LastName **As String**, FirstName **As String**, \_

Address **As String**, PhoneNumber **As String**)

**End Sub**

Πλέον έχει δημιουργηθεί ένας νέος τύπος δεδομένων που ονομάζεται Student και μπορείτε να δηλώσετε νέες μεταβλητές βασισμένες στον τύπο αυτό.

**Sub** Class\_Globals

**Type** Student(LastName **As String**, FirstName **As String**, \_

Address **As String**, PhoneNumber **As String**)

**Public** Student1 **As Student**

**End Sub**

Η μεταβλητή Student1 είναι πλέον μία μεταβλητή τύπου Student και για να έχετε πρόσβαση στα δεδομένα της χρησιμοποιείτε το όνομα της μεταβλητής με μία τελεία και στη συνέχεια το όνομα των στοιχείων που περιλαμβάνονται στη δήλωση type πχ.

Student1.LastName = “Ioannidis”

Student1.FirstName = “Alkinoos”

Student1.Address = “Athens, Greece”

Student1.PhoneNumber = "+303465854234”

Μπορείτε να δημιουργήσετε όσες μεταβλητές του τύπου Student θέλετε και να αναθέσετε την μία στην άλλη. Για παράδειγμα

Public Student2 as Student

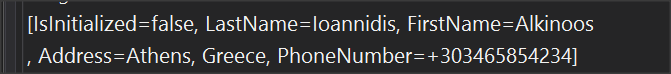
Student2 = Student1

Εδώ η μεταβλητή Student2 περιέχει πλέον τα ίδια ακριβώς δεδομένα με τη μεταβλητή Student1

### Εμφάνιση των στοιχείων

Μπορείτε να χρησιμοποιήσετε την εντολή Log για να εμφανίσετε τα περιεχόμενα του Student1 αλλά το αποτέλεσμα θα είναι κάτι σαν το παρακάτω:

Log(Student1)



Για να εμφανίσετε ή να χρησιμοποιήσετε όλα τα στοιχεία ενός type χρησιμοποιήστε ολόκληρο το όνομα μαζί με το εκάστοτε στοιχείο.

Log(Student1.LastName & Student1.FirstName)

Συνήθως εξυπηρετεί η δημιουργία μίας ρουτίνας η οποία να δέχεται μία μεταβλητή τύπου type και στη συνέχεια να εμφανίζει ή να επεξεργάζεται τα στοιχεία του type.

**Private** **Sub** LogStudent(st **As** Student)

**Log**(st.FirstName)

**Log**(st.LastName)

**Log**(st.Address)

**Log**(st.PhoneNumber)

**End Sub**

## Πίνακας ενός type

Μπορείτε να δημιουργήσετε και πίνακες τύπου type όπου θα περιλαμβάνει περισσότερους μαθητές. Για παράδειγμα

**Sub** Class\_Globals

**Type** Student(LastName **As String**, FirstName **As String**, \_

Address **As String**, PhoneNumber **As String**)

**Public** Students(10) **As Student**

**End Sub**

Η δήλωση Student(10) δημιουργεί ένα πίνακα από 10 Students. Και η χρήση κάθε στοιχείου μπορεί να γίνει με ένα for loop ή με ξεχωριστή αναφορά στο καθένα με τον δείκτη του.

Students(0).LastName = “Paul”

Students(0).FirstName = “Belmond”

## Λίστα από Type

Μία λίστα μπορεί να περιέχει μεταβλητή τύπου Type.

**Sub** Class\_Globals

**Type** Student(LastName **As String**, FirstName **As String**, \_

Address **As String**, PhoneNumber **As String**)

**Public** listStudents **As List**

**End Sub**

**Private Sub B4XPage\_Created (Root1 As B4XView)**

**…**

**…**

**listStudents.Initialize**

**listStudents.Add(Student1)**

**‘Get list items and log**

**For i = 0 To** listStudents**.Size-1**

**Private** st **As Student**

st **=** listStudents**.Get(i)**

**LogStudent(**st**)**

**Next**

**End Sub**

**Private Sub** LogStudent**(**st **As Student)**

**Log**(st.FirstName)

**Log**(st.LastName)

**Log**(st.Address)

**Log**(st.PhoneNumber)

**End Sub**

## Maps από Type

Η χρήση map προκειμένου να αποθηκεύσει δεδομένα τύπου type είναι επίσης εφικτή αρκεί να έχετε προνοήσει για κάποιο μοναδικό κλειδί για κάθε μεταβλητή τύπου type που εισάγετε.

Στην περίπτωση των Students θα μπορούσε να είναι για παράδειγμα ένας αριθμός ταυτότητας, ένας αριθμός μητρώου, ένας λογαριασμός email κλπ. Πρέπει επίσης, να βελτιωθεί το παράδειγμα των students ώστε να περιλαμβάνει και μία μοναδική πληροφορία για κάθε μαθητή.

**Sub** Class\_Globals

**Type** Student(id **As String**, LastName **As String**, \_

FirstName **As String**, Address **As String**, PhoneNumber **As String**)

**Public** mapStudents **As Map**

**End Sub**

**Private Sub B4XPage\_Created (Root1 As B4XView)**

Student1.ID = "FXA47345S3"

Student1.LastName = "Ioannidis"

Student1.FirstName = "Alkinoos"

Student1.Address = "Athens, Greece"

Student1.PhoneNumber = "+303465854234"

mapStudents**.Initialize**

mapStudents**.Put(**Student1**.**ID**,** Student1**)**

**End Sub**

Με το παραπάνω τρόπο είναι εύκολη πλέον και η εύρεση ενός μαθητή με κλειδί τον αριθμό ταυτότητας καλώντας τη μέθοδο Get.

## Αποθήκευση σε αρχεία KVS

Τα αρχεία που συζητήθηκαν στο κεφάλαιο 17 αφορούν αρχεία κειμένου τα οποία συχνά δεν εξυπηρετούν για αποθήκευση περίπλοκων δομών όπως η δηλώσεις τύπου type, οι λίστες και τα Maps. Ένα άλλο είδος αρχειοθέτησης που υποστηρίζει η B4J είναι τα Key Value Store (KVS) αρχεία.

Ο τρόπος λειτουργίας τους είναι παρόμοιος με αυτό των maps. Αρκεί ένα κλειδί και η δομή που θέλετε να αποθηκεύσετε. Τα αρχεία KVS ουσιαστικά κρύβουν μία βάση δεδομένων αλλά το σημαντικό είναι ότι ο προγραμματιστής δεν χρειάζεται να γνωρίζει τίποτα από αυτές αρκεί να χρησιμοποιήσει τις κατάλληλες μεθόδους.

Οι σημαντικές εντολές στα αρχεία KVS είναι οι παρακάτω:

### Δήλωση ενός αρχείου τύπου KVS

Για να δηλώσετε ένα αρχείο KVS πρέπει να χρησιμοποιήσετε την βιβλιοθήκη KeyValueStore στην οποία και θα κάνετε κλικ στην καρτέλα των βιβλιοθηκών.

Στη συνέχεια δημιουργείστε μία μεταβλητή τύπου **KeyValueStore.**

**Private** kvsFile **As** **KeyValueStore**

### Αρχικοποίηση του αρχείου KVS

Η μέθοδος Initialize ορίζει τον φάκελο στον οποίο θα αποθηκευτεί το αρχείο και το όνομα που αυτό θα έχει. Αν ο φάκελος δεν υπάρχει μπορείτε να τον δημιουργήσετε με την εντολή File.MakeDir.

File.**MakeDir**(File.**DirTemp**, "lesson18")

kvsFile.**Initialize**(File.DirTemp & "lesson18", "kvsData.dat")

**Log**(File.**DirTemp** & "lesson18")

Οι παραπάνω εντολές δημιουργούν ένα φάκελο με όνομα lesson18 μέσα στον φάκελο temp, δημιουργούν ένα αρχείο με όνομα «kvsData.dat» και τέλος εμφανίζει τη διαδρομή του αρχείου στην οθόνη Log.

Το αρχείο που θα δημιουργηθεί δεν μπορεί να ανοιχτεί με άλλο πρόγραμμα προβολής (για παράδειγμα το notepad++) αλλά πρέπει πάντα να χρησιμοποιείται από το πρόγραμμα που κατασκευάσατε.

### Εισαγωγή στοιχείων σε αρχείο KVS

Η μέθοδος Put χρησιμοποιείται για να εισάγει τα δεδομένα σε ένα KVS file. Κάθε εισαγωγή που γίνεται ονομάζεται **Εγγραφή**. Απαραίτητη προϋπόθεση είναι να έχετε αποφασίσει για ένα μοναδικό κλειδί προκειμένου να μπορείτε να ανατρέχετε αργότερα στην εγγραφή. Στο παρακάτω παράδειγμα γίνεται εγγραφή της μεταβλητής Student1 τύπου student με κλειδί τον αριθμό ταυτότητας του μαθητή.

kvsFile.**Put**(Student1.ID, Student1)

Με τον τρόπο αυτό μπορείτε να γράψετε οποιοδήποτε τύπο όπως Lists, Maps, Strings, απλές μεταβλητές (αριθμούς), types και πίνακες (μόνο πίνακες από bytes ή αντικείμενα), όπως επίσης και συνδυασμούς (μία λίστα από maps για παράδειγμα ).

Οι δηλώσεις type θα πρέπει να δηλώνονται στην B4XMainPage. Τέλος, υποστηρίζονται και εικόνες με την μέθοδο **PutBitmap** / GetBitmap.

Τα maps μπορούν επίσης να αποθηκεύονται με την μέθοδο PutMapAsync. Είναι και ο καλύτερος τρόπος αποθήκευσης map αφού εισάγει ως κλειδί της εγγραφής το κλειδί του map.

kvsFile.**PutMapAsync**(mapStudents)

### Ανάκτηση στοιχείων από αρχείο KVS

Με τη μέθοδο Get γίνεται ανάκτηση ενός στοιχείου από ένα αρχείο KVS. Η επιστρεφόμενη τιμή είναι ένα αντικείμενο τύπου object. Με άλλα λόγια θα πρέπει να προσέχετε ώστε να εκχωρήσετε την αποδιδόμενη τιμή σε μεταβλητή του ίδιου τύπου.

Στο παρακάτω παράδειγμα διαβάζεται από ένα αρχείο KVS η εγγραφή ενός μαθητή type Student.

**Student3 = kvsFile.Get("FS23534X21")**

LogStudent(Student3)

Η τιμή "FS23534X21" είναι το κλειδί της εγγραφής και η ρουτίνα LogStudent έχει περιγραφεί προηγούμενα όπου δέχεται ένα type Student και εμφανίζει με την εντολή Log τα περιεχόμενα του.

**Remember**

Αν δεν υπάρχει το κλειδί τότε δημιουργείται πρόβλημα στη λειτουργία του προγράμματος. Πρέπει να ελέγχεται η ύπαρξη της εγγραφής πριν την αναθέσετε σε μία μεταβλητή. Ο έλεγχος γίνεται με την μέθοδο **ContainsKey** που περιγράφεται παρακάτω.



Μπορείτε να διαβάσετε ένα ολόκληρο map με την μέθοδο **GetMapAsync.** Η συγκεκριμένη εντολή δέχεται ως παράμετρο μια λίστα από κλειδιά και επιστρέφει ένα map με τα κλειδιά και τις αντίστοιχες αξίες τους. Μπορείτε να έχετε πρόσβαση στα κλειδιά με την μέθοδο GetKeys.

**Log**("Show keys")

**Private** keys **As** **List** = kvsFile.ListKeys

**For** i = 0 **To** keys.**Size**-1

**Log**(keys.**Get**(i))

**Next**

Τέλος στη συνέχεια του παραπάνω κώδικα η GetMapAsync γίνεται:

**Private** mapSt **Αs** **Map**

mapSt = kvsFile.**GetMapAsync**(keys)

### Έλεγχος ύπαρξης εγγραφής

Για να γίνει έλεγχος ύπαρξης ενός κλειδιού σε αρχείο KVS πρέπει να χρησιμοποιήσετε την μέθοδο ContainsKey.

**If** kvsFile.**ContainsKey**("FS23534X21") **Then**

Student3 = kvsFile.**Get**("FS23534X21")

**Else**

**Log**("Wrong id key")

**End If**

Η ContainsKey επιστρέφει True αφού εντοπίσει το κλειδί στο αρχείο.

### Διαγραφή περιεχομένου αρχείο KVS

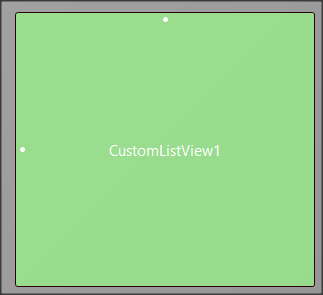
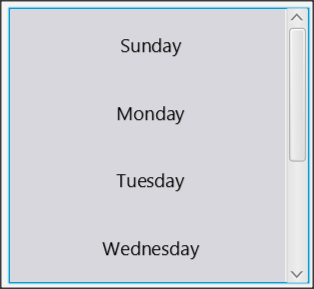
Η μέθοδος remove διαγράφει ένα κλειδί μαζί με την αξία του από ένα KVS file και η μέθοδος DeleteAll διαγράφει όλα τα δεδομένα.

## Περισσότερα Views

Δύο σημαντικά views που θα εξυπηρετήσουν προκειμένου να εμφανίσετε ή να επιλέξετε στοιχεία από σύνθετες δομές types, lists και maps είναι η CustomListView και το B4XComboBox.

### CustomListView

Δημιουργεί μία λίστα με στοιχεία τα οποία μπορείτε να επιλέξετε με κλικ του ποντικού.



Η επιστρεφόμενη τιμή είναι μία αξία την οποία έχετε καθορίσει τη στιγμή της δημιουργίας της ListView.

Sub **Class**\_**Globals**

Private Root As **B4XView**

Private xui As **XUI**

Private lstItems As **List**

1

Private CustomListView1 **As** **CustomListView**

Private lblDate As **Label**

End Sub

**Private** Sub B4XPage\_Created (Root1 As **B4XView**)

Root = Root1

Root.**LoadLayout**("MainPage")

lstItems.**Initialize**

2

lstItems.**AddAll**(**Array** **As** **String**("Sunday", "Monday", \_

"Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))

**For** i = 0 **To** lstItems.Size-1

3

CustomListView1.AddTextItem(lstItems.**Get**(i), i)

**Next**

**End Sub**

4

**Private** Sub CustomListView1\_ItemClick(Index As Int, Value As Object)

lblDate.Text = Value

**End Sub**

1. Δήλωση μιας μεταβλητής τύπου CustomListView
2. Δημιουργία μίας δομής λίστας που θα περιέχει τα στοιχεία που θα εμφανιστούν στο CustomListView
3. Σάρωση όλης της λίστας και κάθε στοιχείο της τοποθετείται στην CustomListView1 με την μέθοδο **AddTextItem**.

Η **AddTextItem** δέχεται μία τιμή που θα εμφανίσει και ένα δείκτη που αντιστοιχεί στην αξία που θέλετε να εμφανίσετε. Στο παραπάνω παράδειγμα αντιστοιχούν οι τιμές από 0 έως 6 για τις μέρες από Κυριακή έως Σάββατο.

1. Όταν γίνει κλικ σε κάποιο στοιχείο της λίστας ενεργοποιείται το γεγονός **\_ItemClick.** Στο παράδειγμα εμφανίζεται ο δείκτης του στοιχείου που έγινε κλικ στην ετικέτα lblDate.

Επιπλέον μέθοδοι της CustomListView είναι και οι παρακάτω:

* **Clear** As String

Διαγράφει όλα τα στοιχεία του CustomListView

* **GetValue** (Index As Int) As Object

Επιστρέφει την τιμή για τη τιμή δείκτη που ορίζεται στην παρένθεση.

* **Size** As Int [read only]

Επιστρέφει τον αριθμό των στοιχείων.

#### Μαρκάρισμα στοιχείου λίστας

Όπως αναφέρθηκε προηγουμένως όταν γίνεται κλικ σε στοιχείο μίας λίστας ενεργοποιείται το γεγονός \_ItemClick. Προκειμένου να παραμείνει μαρκαρισμένο το στοιχείο αυτό θα πρέπει ο προγραμματιστής είτε να χρησιμοποιήσει κάποια από τα έτοιμα εργαλεία που του παρέχει η γλώσσα (π.χ. η βιβλιοθήκη CLVSelections) ή να γράψει το δικό του κώδικα. Ο παρακάτω αλγόριθμος υποθέτει ότι έχετε ορίσει μία μεταβλητή στο Class\_Global με όνομα selectedItem τύπου Int. Ο κώδικας ενημερώνει την μεταβλητή selectedItem με την τιμή του στοιχείου στο οποίο έγινε κλικ στο CLVList

**Private** **Sub** CustomListView1\_ItemClick (Index **As** **Int**, Value **As** **Object**)

**If** selectedItem = -1 **Then**

**1**

**Private** p **As B4XView** = CustomListView1.**GetPanel**(Index)

p.GetView(0).**Color** = xui.**Color\_Blue**

selectedItem = Index

**Else**

**2**

**Private** p **As B4XView** = CustomListView1.**GetPanel**(selectedItem)

p.GetView(0).Color = xui.**Color\_White**

**If** selectedItem = Index **Then**

selectedItem = -1

**Else**

**3**

**Private** p **As B4XView** = CustomListView1.**GetPanel**(Index)

p.GetView(0).Color = xui.**Color\_Blue**

selectedItem = Index

**End If**

**End If**

**End Sub**

Η αρχική τιμή του selectedItem ορίστηκε -1 όπου σημαίνει ότι δεν έχει επιλεγεί τίποτα. Ο κώδικας χρησιμοποιεί δύο εντολές:

**Private** p **As B4XView** = CustomListView1.**GetPanel**(Index)

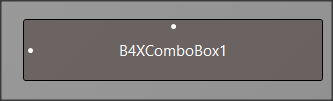
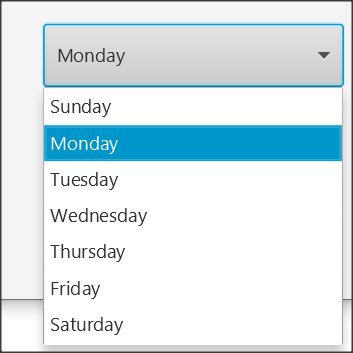
p.GetView(0).**Color** = xui.**Color\_Blue**

Κάθε στοιχείο μιας λίστας δημιουργείται από την γλώσσα μέσα σε ένα πλαίσιο που ονομάζεται panel. Μπορείτε να ορίσετε το χρώμα αποκτώντας πρόσβαση στο πλαίσιο με την μέθοδο GetPanel(Index) όπου Index η τρέχουσα τιμή της γραμμής στην οποία έγινε το κλικ. Στη συνέχεια η εντολή p.GetView(0).Color ορίζει το χρώμα που επιθυμεί ο προγραμματιστής να ορίσει. Στη συνεχεια:

1. Αν γίνει κλικ σε ένα στοιχείο της λίστα η ρουτίνα ελέγχει την τιμή του selectedItem και αν αυτό είναι -1 τότε ορίζει χρώμα φόντου Blue στη γραμμή που έγινε κλικ και ορίζει ως τιμή selectedItem το τη τιμή του Index που δίνει το γεγονός \_ItemClick
2. Αν το selectedItem έχει ήδη μία τιμή ορίζεται χρώμα λευκό ως φόντο στο πλαίσιο και μετά υπάρχουν δύο περιπτώσεις
   1. Να έχει γίνει κλικ στο ήδη επιλεγμένο στοιχείο, οπότε σταματά να είναι επιλεγμένο το στοιχείο και το selectedItem γίνεται -1
   2. Να έχει γίνει κλικ σε άλλο στοιχείο οπότε το selectedItem παίρνει την αξία του Index.

### B4XComboBox

Το B4XComboBox εμφανίζει μία αναδιπλούμενη λίστα με στοιχεία. Ο χρήστης μπορεί να κάνει κλικ σε ένα από αυτά και να το επιλέξει. Σε αντίθεση με τη CustomListView η επιστρεφόμενη τιμή είναι πάντα ένας αριθμός που αφορά την σειρά τοποθέτησης του στοιχείου στο CοmboBox επομένως πρέπει ο προγραμματιστής να μπορεί στη συνέχεια να αντιστοιχήσει την τιμή με κάποια δομή.



Sub Class\_Globals

Private Root As **B4XView**

Private xui As **XUI**

Private lstItems As **List**

1

Private B4XComboBox1 As **B4XComboBox**

Private lblCmbDate As **Label**

End Sub

Private Sub B4XPage\_Created (Root1 As **B4XView**)

Root = Root1

Root.**LoadLayout**("MainPage")

lstItems.**Initialize**

2

lstItems.**AddAll**(Array As String("Sunday", "Monday", \_

"Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))

3

B4XComboBox1.**SetItems**(lstItems)

End Sub

4

Private Sub B4XComboBox1**\_SelectedIndexChanged** (Index As **Int**)

lblCmbDate.**Text** = Index

End Sub

1. Δήλωση μιας μεταβλητής τύπου **B4XComboBox**
2. Δημιουργία μίας δομής λίστας που θα περιέχει τα στοιχεία που θα εμφανιστούν στοB4XComboBox
3. Τοποθέτηση των στοιχείων της λίστας στο B4XComboBox. Προσοχή δεν απαιτείται επαναληπτική δομή όπως στις CustomListView
4. Αν επιλεγεί κάποιο στοιχείο τότε ενεργοποιείται το γεγονός **\_SelectedIndexChanged** και επιστρέφει τον δείκτη του στοιχείου που επιλέχθηκε.

## Ασκήσεις

1. Να δημιουργεί ένα type με όνομα Customer και μεταβλητές

ID, FirstName, LasName, Phone

1. Να δημιουργεί μία λίστα από Customer με τα παρακάτω στοιχεία

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **FirstName** | **LasName** | **Phone** |
| A3473 | John | Smith | 4563454 |
| B1753 | Selim | Al Huarizmi | 6532578 |
| C6544 | Mateo | Sandor | 7345346 |
| C6323 | Lucía | Graham | 1231345 |
| F1462 | Noam | Bell | 6978323 |

1. Να αποθηκευτεί η λίστα σε αρχείο KVS
2. Να κατασκευάσετε πλήκτρο που όταν γίνεται κλικ να γεμίζει μία CustomListView με τα στοιχεία των πελατών.
3. Να κατασκευάσετε πλήκτρο που όταν γίνει κλικ να εμφανίζει κατάλληλο DialogBox που να ζητάει τα στοιχεία νέου πελάτη όταν εισαχθούν να τοποθετούνται τελευταία στη CustomListView.
4. Να αποθηκεύει το νέο πελάτη και στο KVS αρχείο.