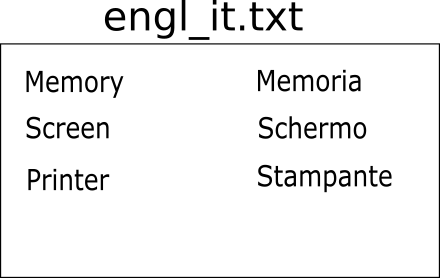
# Lesson 17 – Files

* What is a file?
* File location in B4J
* File Methods

What students should know

**2h**

Αρχείο ονομάζουμε μία συλλογή δεδομένων με ομοειδές περιεχόμενο που αποθηκεύεται στην μόνιμο χώρο συνήθως στο σκληρό δίσκο του υπολογιστή. Είναι από τα πιο σημαντικά χαρακτηριστικά των γλωσσών προγραμματισμού δεδομένου ότι ενώ όλη η εργασία προσωρινής αποθήκευσης γίνεται με την κεντρική μνήμη, αφού ολοκληρωθεί μια εφαρμογή θα πρέπει ήδη τα δεδομένα της να είναι αποθηκευμένα μόνιμα.



Picture Αρχείο engl\_it.txt

Γενικά τα αρχεία διακρίνονται σε βάσεις δεδομένων και απλά αρχεία τα οποία και θα εξετάσουμε.

## Φάκελοι Αποθήκευσης

Κάθε λειτουργικό σύστημα διαθέτει πολλούς και διαφορετικούς φακέλους για να αποθηκεύει τις εφαρμογές τα δεδομένα ή άλλα αρχεία. Προκειμένου η B4J να δουλέψει με αρχεία ανεξάρτητα από το λειτουργικό σύστημα χρησιμοποιεί λέξεις κλειδιά που αναφέρονται σε συγκεκριμένου είδους φακέλων.

### File.DirAssets

Περιλαμβάνει τα αρχεία που περιέχονται στο φάκελο files της εφαρμογής και έχουν προστεθεί από τη διαχείριση αρχείων του B4J κατά τη φάση της ανάπτυξης μιας εφαρμογής. Αυτά τα αρχεία είναι μόνο για ανάγνωση και δεν μπορεί να γίνει κάποια προσθήκη αρχείου κατά τη διάρκεια εκτέλεσης της εφαρμογής. Συνήθως γράφονται αρχεία από τον προγραμματιστή προκειμένου μετά την εγκατάσταση της εφαρμογής να αντιγραφούν σε άλλους φακέλους για να χρησιμοποιηθούν.

### xui.DefaultFolder

Επιστρέφει ότι και η εντολή File.DirData και είναι κατάλληλη για να αρχεία που πρόκειται να γράφουμε δεδομένα. Απαιτείται πρώτα να καλέσετε την εντολή SetDataFolder μία φορά πριν την χρησιμοποιήσετε.

xui.SetDataFolder(AppName As String)

### File.DirData

Επιστρέφει τον φάκελο που αποθηκεύονται τα δεδομένα της εφαρμογής και είναι κατάλληλος για δημιουργία αρχείων και αποθηκεύσεις δεδομένων.

Σε περιβάλλον Windows επιστρέφει τον φάκελο «user data folder» που συνήθως βρίσκεται στην διαδρομή

*C:\Users\[user name]\AppData\Roaming\[AppName]*

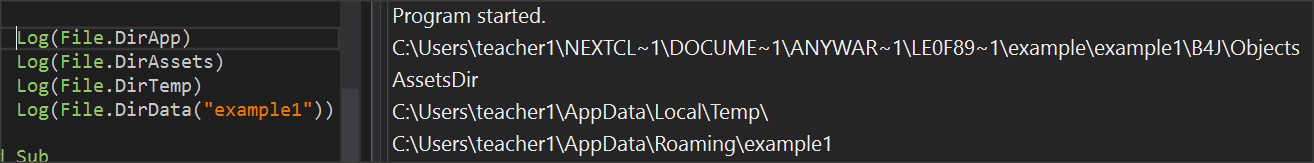
Σε λειτουργικά συστήματα εκτός Windows επιστρέφει ότι και η File.DirApp

### File.DirApp

Επιστρέφει τον φάκελο που είναι εγκαταστημένη η εφαρμογή. Στα Windows αυτός ο φάκελος συνήθως είναι ο “Program Files” και είναι μόνο για διάβασμα.

### File.DirTemp

Επιστρέφει τον φάκελο προσωρινών αρχείων. Όπως φανερώνει και το όνομα του τον χρησιμοποιείτε όταν θέλετε μόνο για προσωρινή αποθήκευση.



Μπορείτε να χρησιμοποιήσετε συνδυαστικά τις παραπάνω εντολές για να δηλώσετε ένα νέο φάκελο μέσα στους προηγούμενους. Για παράδειγμα

**Private** strFolder **As** **String** = File.DirTemp & “lesson17\”

Log(strFolder)

Εμφανίζει C:\Users\teacher1\AppData\Local\Temp\lesson17\

## Δημιουργία φακέλων

Μπορείτε να δημιουργήσετε ένα νέο φάκελο με την εντολή

File.MakeDir (Parent As String, Dir)

File.MakeDir(File.DirTemp, “lesson17")

Δημιουργεί ένα φάκελο με όνομα lesson17 μέσα στο C:\Users\teacher1\AppData\Local\Temp\ του προηγούμενου παραδείγματος

## Έλεγχος ύπαρξης αρχείου

Πριν χρησιμοποιήσετε ένα αρχείο συνήθως η πρώτη εργασία που πρέπει να γίνει είναι ο έλεγχος ύπαρξης. Η εντολή είναι

**File.Exists (Dir As String, FileName As String)**

Και επιστρέφει αλήθεια ή ψέμα που μπορείτε να το ελέγξετε με μια εντολή if.

**Private** fn **as** **String** = “mydata.txt”

**If** File.Exists(File.DirTemp, fn) **Then**

**Log**("File " & fn & " Exists")

**Else**

**Log**("There is no File: " & fn)

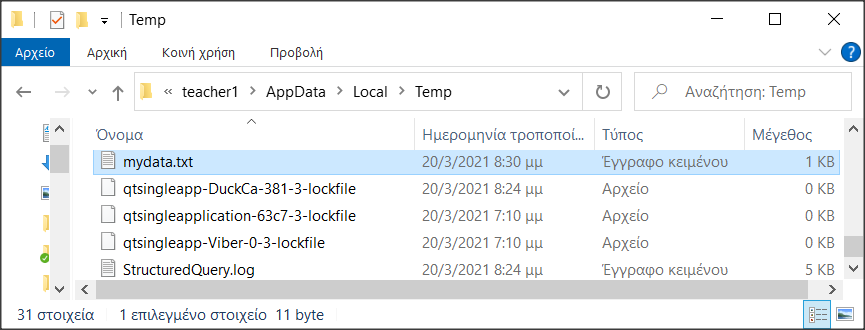
**End If**

## Δημιουργία και εγγραφή σε αρχείο

Η εντολή δημιουργίας ενός αρχείο είναι η:

**File.OpenOutput** (Dir As String, FileName As String, Append As Boolean)

**Private** fn **as** **String** = “mydata.txt”

File.OpenOutput(File.DirTemp, fn, True)

Αν το αρχείο δεν υπάρχει τότε δημιουργείται.

Η λογική τιμή True ή False στην τρίτη παράμετρο αφορά αν το αρχείο ανοίξει για εγγραφή σβήνοντας τα παλαιότερα δεδομένα ή θα προσθέσει στις ήδη υπάρχουσες εγγραφές.

## Γράφοντας και διαβάζοντας δεδομένα

Μπορείτε να γράψετε ή να διαβάσετε δεδομένα από μία λέξη έως πολυπλοκότερες δομές όπως lists και maps.

### Μεταβλητές τύπου string

File.WriteString (Dir As String, FileName As String, Text As String)

**Private** fn **as** **String** = “mydata.txt”

Private msg As String = "Hello World"

File.WriteString(File.DirTemp, fn, msg)

Γράφει μια μεταβλητή τύπου string στο αρχείο. Προσοχή το αρχείο δημιουργείται από την αρχή ακόμα και αν υπήρχε ήδη. Οπότε τα δεδομένα του διαγράφονται.

Για ένα αρχείο που υπάρχει ήδη μπορείτε να διαβάσετε τα δεδομένα του και να τα μεταφέρετε σε μία μεταβλητή τύπου string με την εντολή

**File.ReadString** (Dir As String, FileName As String) As String

Private fn as String = “mydata.txt”

Private strFileContent As String

strFileContent = File.ReadString(strFolder, fn)

### Λίστες

Για να γράψετε μία λίστα σε ένα αρχείο χρησιμοποιείστε την εντολή:

**File.WriteList(Dir As String, FileName As String, List As List)**

File.WriteList(strFolder, "mydata.txt", List1)

Με την παραπάνω εντολή κάθε στοιχείο της λίστας μετατρέπεται σε string και δημιουργείται μία νέα γραμμή στο αρχείο.

Για να διαβάσετε μία λίστα από ένα αρχείο χρησιμοποιείστε την εντολή:

**File.ReadList (Dir As String, FileName As String)**

List1 = File.ReadList(File.DirRootExternal, " mydata.txt")

Η εντολή δημιουργεί ένα νέο στοιχείο στη λίστα ανά γραμμή του αρχείου.

### Maps

Η εγγραφή ενός ολόκληρου map σε αρχείο γίνεται με την εντολή

**File.WriteMap(Dir As String, FileName As String, Map As Map)**

File.WriteMap(File.DirInternal, “file.txt", map1)

Η πιο συνηθισμένη λειτουργία είναι η δημιουργία ενός αρχείου ρυθμίσεων της εφαρμογής.

Ένα map διαβάζεται με την εντολή

**ReadMap(Dir As String, FileName As String)**

map1 = File.ReadMap(File.DirInternal, "file.txt")

Η σειρά των στοιχείων δεν είναι απαραίτητα ίδια με τη σειρά του αρχείου αλλά αυτό δεν έχει σημασία σε ένα map.

## Ασκήσεις

1. Δίνεται το πρόγραμμα Lesson17-ex1 το οποίο δημιουργεί μία λίστα με ποδοσφαιρικές ομάδες και ένα σύντομο ιστορικό τους.
   1. Να δημιουργήσετε map με κλειδί το όνομα της ομάδας και περιεχόμενο το ιστορικό της.
   2. Να δημιουργήσετε αρχείο που να αποθηκεύει το παραπάνω map με όνομα teams.txt
   3. Να δημιουργήσετε φόρμα που ζητάει ένα όνομα ομάδας και να εμφανίζει το ιστορικό της χρησιμοποιώντας το map που δημιουργήσατε.