

**ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΕΡΕΥΝΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ**

**ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ**

**Βραχνός Ε., Κουρέτας Ι., Μακρυγιάννης Π., Παραδείση Α.**

**ΕΙΔΙΚΑ ΘΕΜΑΤΑ ΣΤΟΝ  
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΥΠΟΛΟΓΙΣΤΩΝ**

**Γ΄ ΤΑΞΗ ΕΠΑΛ  
ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΟΔΗΓΟΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ**

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ

Πρόεδρος: **Γκλαβάς Σωτήριος**

ΓΡΑΦΕΙΟ ΕΡΕΥΝΑΣ, ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΕΦΑΡΜΟΓΩΝ Β΄

Προϊστάμενος: **Μάραντος Παύλος**

ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Επιστημονικά Υπεύθυνος: **Δρ. Τσαπέλας Θεοδόσιος**, Σύμβουλος Β΄ Πληροφορικής ΙΕΠ

ΣΥΓΓΡΑΦΙΚΗ ΟΜΑΔΑ:

**Βραχνός Ευριπίδης**, καθηγητής πληροφορικής

**Κουρέτας Ιωάννης**, καθηγητής πληροφορικής

**Μακρυγιάννης Παναγιώτης**, καθηγητής πληροφορικής

**Παραδείση Άρτεμις**, καθηγήτρια πληροφορικής

ΕΠΙΜΕΛΕΙΑ ΣΥΝΤΟΝΙΣΜΟΣ ΟΜΑΔΑΣ:

**Μακρυγιάννης Παναγιώτης**, καθηγητής πληροφορικής

**Παραδείση Άρτεμις**, καθηγήτρια πληροφορικής

## Περιεχόμενα

ΕΝΟΤΗΤΑ 1_ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.....	4
Εισαγωγή.....	4
1. Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό.....	6
2. Βασικά Στοιχεία της Γλώσσας.....	8
3. Αντικείμενα και Μέθοδοι.....	9
3.1 Συμβολοσειρές.....	9
3.2 Δημιουργία αντικειμένων.....	10
3.3 Μέθοδοι.....	10
3.4 Κληρονομικότητα.....	11
3.5 Καθοδήγηση από γεγονότα.....	11
4. Αντικειμενοστρεφής προγραμματισμός.....	12
4.1 Κληρονομικότητα.....	12
4.2 Κατασκευαστές.....	13
4.3 Αναφορές και μεταβίβαση παραμέτρων.....	13
4.4 Πολυμορφισμός.....	14
5. Προγραμματισμός οδηγούμενος από γεγονότα.....	15
6. Βάσεις Δεδομένων.....	16
7. Δικτυακός Προγραμματισμός.....	16
8. Ανάπτυξη Ολοκληρωμένης Εφαρμογής.....	18
ΕΝΟΤΗΤΑ 2_ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ (APPS) ΓΙΑ ANDROID.....	19

---

# ΕΝΟΤΗΤΑ 1

## ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

---

### Εισαγωγή

Το μάθημα αποτελεί μια εισαγωγή των μαθητών στον αντικειμενοστρεφή προγραμματισμό με όχημα τη γλώσσα προγραμματισμού Java. Η Java είναι μια γλώσσα προγραμματισμού η οποία μπορεί να υποστηρίξει την ανάπτυξη, οποιασδήποτε εφαρμογής, σε οποιαδήποτε πλατφόρμα, από τους προσωπικούς υπολογιστές και τα κινητά τηλέφωνα με Android μέχρι την ψηφιακή τηλεόραση. Ο σύντομος χρόνος ανάπτυξης μιας εφαρμογής, ειδικά από μη έμπειρους προγραμματιστές είναι στα πλεονεκτήματα αυτής της γλώσσας. Αξιοποιώντας αυτό το πλεονέκτημα της Java οι μαθητές, στο τέλος του μαθήματος, θα πρέπει να μπορούν να σχεδιάσουν και να υλοποιήσουν ολοκληρωμένες εφαρμογές μικρής έκτασης με την αντικειμενοστρεφή φιλοσοφία.

Τα εργαλεία που θα χρησιμοποιήσουμε στην Ενότητα 1 του μαθήματος, είναι ουσιαστικά δυο: Το περιβάλλον προγραμματισμού GreenFoot και το Eclipse. Αρχικά αξιοποιούμε τα εκπαιδευτικά πλεονεκτήματα του GreenFoot (Κεφάλαια 1-3), για μια πρώτη εισαγωγή στα αντικειμενοστρεφή χαρακτηριστικά της γλώσσας και στον γεγονοστρεφή προγραμματισμό. Η παρουσίαση των αλγοριθμικών δομών επιλογής και επανάληψης είναι σύντομη δεδομένου ότι, οι μαθητές τις έχουν ξανασυναντήσει και σε προηγούμενη τάξη, είτε σε ψευδογλώσσα, είτε σε πραγματική γλώσσα προγραμματισμού (Pascal/Python).

Προτείνεται το βασικό προγραμματιστικό περιβάλλον του μαθήματος να είναι το Eclipse το οποίο έχει ενσωματωμένα όλα τα εργαλεία ανάπτυξης εφαρμογών π.χ. javadoc, Ant, JUnit. Ωστόσο αν θεωρηθεί πολύ βαρύ για τους υπολογιστές του εργαστηρίου πληροφορικής μπορούμε να χρησιμοποιήσουμε έναν πολύ ελαφρύ συντάκτη πηγαίου κώδικα όπως το Notepad++. Η μεταγλώττιση και η εκτέλεση των εφαρμογών μας θα πρέπει όμως να γίνεται από τη γραμμή εντολών κάτι που δεν είναι ιδιαίτερα ελκυστικό για τους μαθητές. Ο εκπαιδευτικός μπορεί να χρησιμοποιήσει οποιοδήποτε περιβάλλον προγραμματισμού της Java του επιτρέπει την επίτευξη των διδακτικών στόχων του Αναλυτικού Προγράμματος Σπουδών (ΑΠΣ). Προτείνεται επίσης να χρησιμοποιηθεί η βιβλιοθήκη της Java Task Force (JTF) της ACM, η οποία στην αρχή θα μας βοηθήσει να αποφύγουμε τις δυσκολίες που ανακύπτουν κατά τον προγραμματισμό λειτουργιών εισόδου/εξόδου. Η χρήση της βιβλιοθήκης θα μας βοηθήσει να αποκρύψουμε αρκετές επουσιώδεις λεπτομέρειες της γλώσσας, που στα πρώτα μαθήματα μπορεί να δυσκολέψουν αρκετά τους μαθητές.

Θα πρέπει να σημειωθεί ότι ο γενικότερος σκοπός είναι να έρθουν οι μαθητές σε επαφή με τη φιλοσοφία ανάπτυξης μιας ολοκληρωμένης εφαρμογής σε Java και όχι να μάθουν τις λεπτομέρειες κάποιου πακέτου ή περιβάλλοντος προγραμματισμού, που αύριο μπορεί να θεωρηθεί ξεπερασμένο. Άρα, καθώς θα αναπτύσσουν την εφαρμογή τους, είναι καλή πρακτική να αναζητούν στο διαδίκτυο βέλτιστες λύσεις σε προβλήματα που θα συναντήσουν, μέσω δικτυακών τόπων όπως το *stackoverflow* και να μπορούν να χρησιμοποιούν την τεκμηρίωση των βιβλιοθηκών της Java, ώστε να βρίσκουν γρήγορα και εύκολα αυτό που ψάχνουν.

Προτείνεται η εκπόνηση των έργων να γίνει από ομάδες 3-4 μαθητών. Ο ρόλος του εκπαιδευτικού είναι καταλυτικός σε αυτό το σημείο. Θα πρέπει να καθοδηγήσει τους μαθητές στον καταμερισμό των καθηκόντων τους, π.χ. ένας μαθητής μπορεί να αναλάβει τον σχεδιασμό της γραφικής διεπαφής, ένας άλλος την επικοινωνία με τη βάση δεδομένων κ.ο.κ.

Σε κάθε κεφάλαιο υπάρχουν αρκετά παραδείγματα κώδικα τα οποία επιβάλλεται να εκτελέσουν οι μαθητές στο εργαστήριο και να διερευνήσουν τα αποτελέσματά τους. Μια καλή πρακτική θα ήταν να δώσετε στους μαθητές τα απαραίτητα αρχεία κώδικα ώστε να χτίσουν την εφαρμογή και να πειραματιστούν με αυτή. Σκοπός είναι να ανακαλύψουν, με την καθοδήγηση του

εκπαιδευτικού, τα βασικά χαρακτηριστικά της γλώσσας. Τα προγράμματα που παρουσιάζονται σε κάθε κεφάλαιο θα μπορούσαν να τροποποιηθούν ή να επεκταθούν ώστε να συμπεριλάβουν και πιο γενικές περιπτώσεις.

# 1 Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό

Στο κεφάλαιο αυτό γίνεται μια πρώτη παρουσίαση του εκπαιδευτικού περιβάλλοντος Greenfoot. Το περιβάλλον είναι ιδιαίτερα εύχρηστο, ειδικά σχεδιασμένο για να μεταφέρει τις έννοιες και αρχές του αντικειμενοστρεφούς προγραμματισμού, με ένα καθαρό και εύκολο τρόπο. Το Greenfoot μπορείτε να το κατεβάσετε από την ομώνυμη διεύθυνση [www.greenfoot.org](http://www.greenfoot.org).

Για να κερδίσουμε το ενδιαφέρον των μαθητών στον προγραμματισμό, προτείνεται να δουν ένα μικρό βίντεο <https://www.youtube.com/watch?v=Fkd9TWUtFm0>, όπου ένα δωδεκάχρονο αγόρι δημιουργεί παιχνίδια και εφαρμογές για κινητές συσκευές.

Ακολουθείται η λεγόμενη “πρώτα τα αντικείμενα” (objects first) προσέγγιση, όπου ξεκινάμε με τις βασικές έννοιες του αντικειμενοστρεφούς προγραμματισμού και όχι με τις βασικές αλγοριθμικές δομές της ακολουθίας, επιλογής και επανάληψης. Γίνεται εισαγωγή στις έννοιες της κλάσης και των αντικειμένων που την υλοποιούν. Προτείνεται να δοθούν παραδείγματα από την καθημερινή ζωή για να αποσαφηνίσουν οι μαθητές τη διαφορά μεταξύ κλάσης και αντικειμένου. Για την αποφυγή της ταύτισης μεταξύ της έννοιας της κλάσης (class) και της έννοιας του αντικειμένου (object) πρέπει να χρησιμοποιηθούν παραδείγματα, όπου από μια κλάση να δημιουργούνται πολλά αντικείμενα.

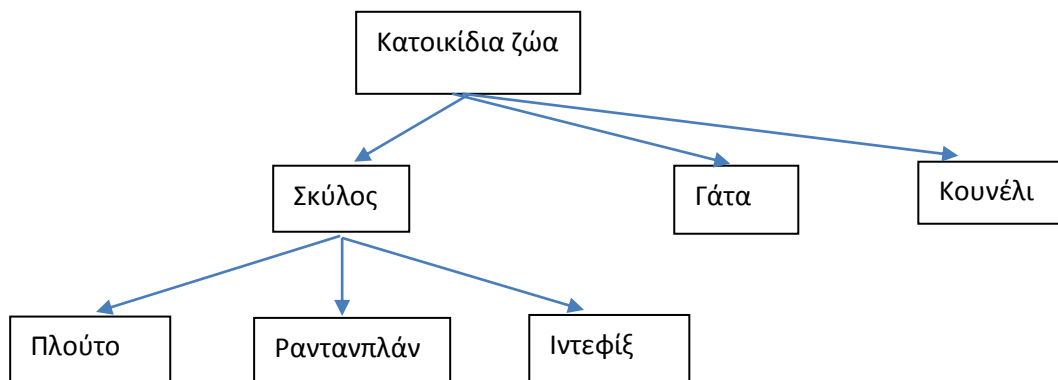
Ο εκπαιδευτικός μπορεί να εντοπίσει τη συγκεκριμένη παρανόηση, απευθύνοντας σχετικές ερωτήσεις ανοικτού ή κλειστού τύπου στους μαθητές ή δίνοντας τους ένα απλό πρόβλημα και ζητώντας τους να ορίσουν τις απαιτούμενες κλάσεις και αντικείμενα.

Ένα ενδεικτικό παράδειγμα είναι η κλάση Σκύλος από την οποία δημιουργούνται τα αντικείμενα: Πλούτο, Ραντανπλάν, Ιντεφίξ.

Μετά την οικοδόμηση στέρεας γνώσης για τις πρωταρχικές έννοιες κλάσης-αντικειμένου, γίνεται η παρουσίαση της ιεραρχίας των κλάσεων (υπερκλάση-υποκλάση) και η διάκριση των σχέσεων κλάσης-αντικειμένου από τις σχέσεις ιεραρχίας.

Κατόπιν, πρέπει να εξηγηθεί γιατί οι προγραμματιστές επιλέγουν να επεκτείνουν μια κλάση δημιουργώντας μια υποκλάση της αντί να δημιουργήσουν μια νέα κλάση από την αρχή (Έμφαση στην επαναχρησιμοποίηση κώδικα).

Σε συνέχεια του παραπάνω παραδείγματος ο εκπαιδευτικός μπορεί να σχεδιάσει το ακόλουθο διάγραμμα:



Για την κατανόηση των χαρακτηριστικών των αντικειμένων, ο εκπαιδευτικός μπορεί να επεκτείνει το παραπάνω παράδειγμα, δίνοντας ως χαρακτηριστικά της κλάσης Κατοικίδια ζώα τις εξής: ράτσα, φύλο, όνομα, ηλικία. Οι κλάσεις Σκύλος, Γάτα και Κουνέλι κληρονομούν όλες τα χαρακτηριστικά της κλάσης Κατοικίδια ζώα. Τα αντικείμενα Πλούτο, Ραντανπλάν και Ιντεφίξ εξειδικεύουν την κλάση Σκύλος με διαφορετικές τιμές στις ιδιότητες τους. Π.χ. η ράτσα του Πλούτο είναι Λαγωνικό, η ράτσα του Ραντανπλάν είναι Κυνηγόσκυλο και η ράτσα του Ιντεφίξ είναι Φοξ τεριέ.

Στις εργαστηριακές ασκήσεις του κεφαλαίου, οι μαθητές θα δημιουργήσουν κλάσεις, θα διαχειριστούν αντικείμενα των κλάσεων και θα διερευνήσουν τις ιδιότητες τους.

Ο εκπαιδευτικός για τον εμπλουτισμό του εκπαιδευτικού υλικού μπορεί να ανατρέξει στις ακόλουθες διευθύνσεις:

Greenfoot <http://www.greenfoot.org/door>

Greenfoot room for teachers <http://greenroom.greenfoot.org/door>

Greenfoot videos <http://www.greenfoot.org/doc/videos.html>

Greenfoot book <http://www.greenfoot.org/book>

Σημειώσεις Greenfoot: <http://users.sch.gr/dimnikolos/greenfoot/1preface.html>

<http://www.tutorialspoint.com/java/index.htm>

<https://sites.google.com/site/ywicnm/lesson-plans/greenfoot/high-school-curriculum>

Σειρά από βίντεο για την εισαγωγή στη java με τη χρήση του Greenfoot:

<http://www.greenfoot.org/doc/joy-of-code>

Greenfoot programmer's manual:

[https://www.google.gr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0CCcQFjABahUKewjMkMGJm93IAhXHjiwKHUgaCEg&url=http%3A%2F%2Fmisssimpson.com%2Fspecialist-ict%2Fgreenfoot-programmers27-manual.pdf&usg=AFQjCNEAF5T5QR3-OWOBguHDaH\\_i62FSGg&sig2=24VBE4eleBNpMhgNBazF4w](https://www.google.gr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0CCcQFjABahUKewjMkMGJm93IAhXHjiwKHUgaCEg&url=http%3A%2F%2Fmisssimpson.com%2Fspecialist-ict%2Fgreenfoot-programmers27-manual.pdf&usg=AFQjCNEAF5T5QR3-OWOBguHDaH_i62FSGg&sig2=24VBE4eleBNpMhgNBazF4w)

Oracle java tutorial: <http://docs.oracle.com/javase/tutorial/java/>

### Στόχοι

Οι μαθητές να μπορούν να:

- Περιγράφουν τα συστατικά στοιχεία του αλληλεπιδραστικού περιβάλλοντος του Greenfoot
- Περιγράφουν όλα τα στάδια ανάπτυξης ενός προγράμματος
- Αναγνωρίζουν τα συγκριτικά πλεονεκτήματα του αντικειμενοστρεφούς προγραμματισμού
- Περιγράφουν τις κλάσεις και τις υποκλάσεις.
- Δημιουργούν και να διαγράφουν κλάσεις και αντικείμενα.
- Αναγνωρίζουν ότι από μία κλάση μπορούν να δημιουργηθούν πολλά αντικείμενα τα οποία θα έχουν τα χαρακτηριστικά και τις μεθόδους της κλάσης αυτής.

## 2 Βασικά Στοιχεία της Γλώσσας

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τους βασικούς τύπους των δεδομένων της Java, τους αντίστοιχους τελεστές (αριθμητικοί, συγκριτικοί, λογικοί), τη δομή επιλογής και τη δομή επανάληψης.

Η παρουσίαση των αλγοριθμικών δομών της ακολουθίας, της επιλογής και της επανάληψης προτείνεται να γίνει με αναφορές στην Python (ή την Pascal) και την ψευδογλώσσα με τις οποίες έχουν έρθει σε επαφή οι μαθητές στην Β' τάξη του Λυκείου. Να μην δοθεί βάρος στην επίλυση δύσκολων αλγοριθμικών προβλημάτων αλλά στην αξιοποίηση των ιδιαίτερων χαρακτηριστικών της γλώσσας για την επίλυση προβλημάτων. Για παράδειγμα θα μπορούσαν οι μαθητές να εξοικειωθούν και με μη δομημένα χαρακτηριστικά της γλώσσας όπως η break.

Θα πρέπει οι μαθητές να εξοικειωθούν με τη χρήση των βιβλιοθηκών προγραμματισμού της γλώσσας, αναζητώντας την κατάλληλη μέθοδο και εφαρμόζοντας την στη συνέχεια για την επίλυση προβλημάτων. Το μεγάλο στοίχημα είναι να μάθουν οι μαθητές πώς να μαθαίνουν μόνοι τους. Δεν χρειάζεται να έχουν διδαχθεί μια κλάση ή μια μέθοδο για να τις χρησιμοποιήσουν, αλλά να πειραματιστούν με αυτές αρκετά ώστε να ανακαλύψουν τα βασικά χαρακτηριστικά της συμπεριφοράς τους για να αξιοποιηθούν κατάλληλα. Η τεκμηρίωση που παρέχεται για τις βιβλιοθήκες της Java είναι πολύ καλά σχεδιασμένη ώστε να μπορεί να βρει εύκολα κάποιος αυτό που θέλει. Οι μαθητές θα πρέπει να εξασκηθούν σε αυτό.

Σε αυτό το κεφάλαιο θα δείξουμε πως μπορούμε να χρησιμοποιήσουμε μια βιβλιοθήκη για την επικοινωνία της εφαρμογής (JOptionPane) με τον χρήστη μέσω παραθύρων διαλόγου.

Η λογική της Java συνοψίζεται στην εξής ιδέα της ελάχιστονος προσπάθειας:

Δεν επανεφεύρουμε τον τροχό. Όταν θέλουμε να αναπτύξουμε μια εφαρμογή ψάχνουμε πρώτα στις βιβλιοθήκες της γλώσσας, μήπως αυτή η εφαρμογή ή τμήματά της έχουν ήδη υλοποιηθεί από άλλον προγραμματιστή ή με άλλα λόγια: «Δεν υπάρχει λόγος να κάνεις κάτι αν το έχει κάνει κάποιος άλλος, πριν από σένα».

Αν από την άλλη πλευρά αναπτύξουμε μια βιβλιοθήκη με τόσο εξειδικευμένες λειτουργίες που δεν έχουν αναπτυχθεί ακόμα, καλό θα ήταν να συνεισφέρουμε και εμείς στην κοινότητα των προγραμματιστών, θέτοντάς την στην διαθεσιμότητα και άλλων προγραμματιστών.

Ιδιαίτερη προσοχή θα πρέπει να δοθεί στην ενότητα 2.4 όπου γίνεται μια πρώτη παρουσίαση των μεθόδων. Οι μαθητές πρέπει να κατανοήσουν ότι ένα αντικείμενο μπορεί να εκτελεί μόνο τις μεθόδους που ορίζονται στην κλάση από την οποία προέρχεται.

Μπορεί να χρησιμοποιηθεί το παράδειγμα του προηγούμενου κεφαλαίου με την κλάση Κατοικίδια ζώα για να δείξει ο εκπαιδευτικός ότι η κλάση Σκύλος ορίζει συγκεκριμένες ενέργειες που μπορούν να κάνουν τα αντικείμενα που προέρχονται από αυτήν, όπως να κινούνται, να τρώνε, να κοιμούνται κτλ.

### Στόχοι

Οι μαθητές να μπορούν να:

- Χρησιμοποιούν τις δομές ελέγχου για την επίλυση προβλημάτων
- Χρησιμοποιούν διάφορα είδη παραθυρικών διαγνωστικών μηνυμάτων
- Βρίσκουν και να χρησιμοποιούν τις έτοιμες συναρτήσεις των προκαθορισμένων βιβλιοθηκών



### 3 Αντικείμενα και Μέθοδοι

Στο πρώτο κεφάλαιο είδαμε πως μπορούμε να δημιουργήσουμε κλάσεις και αντικείμενα χωρίς να γράψουμε κώδικα στο περιβάλλον Greenfoot. Με αυτόν τον τρόπο όμως πρέπει να ξαναδημιουργούμε τα αντικείμενα ένα – ένα κάθε φορά που εκτελείται από την αρχή το πρόγραμμα.

Για να το αποφύγουμε αυτό πρέπει να δημιουργήσουμε μια αυτοματοποιημένη διαδικασία δημιουργίας των αντικειμένων η οποία να εκτελείται κάθε φορά που κατασκευάζεται ο κόσμος. Επίσης μπορούμε να καθορίσουμε τη συμπεριφορά των αντικειμένων μας, υλοποιώντας τις δικές μας μεθόδους.

#### 3.1 Συμβολοσειρές

Αρχικά οι μαθητές θα υλοποιήσουν μικρά προγράμματα για επεξεργασία συμβολοσειρών, με χρήση των μεθόδων της κλάσης String. Εδώ μπορεί να γίνει μια πρώτη εισαγωγή στον τελεστή new για τη δημιουργία αντικειμένων String. Να σχολιαστεί η υπερφόρτωση του τελεστή '+' για τις περιπτώσεις αριθμών και συμβολοσειρών μέσα από παραδείγματα, όπως το παρακάτω και να γίνει συζήτηση για τη συμπεριφορά του τελεστή '+' ειδικά στις δυο τελευταίες εντολές της main.

```
public class Test extends Program {  
  
    public static void main( String args[] ) {  
  
        String word = "Java";  
        String phrase = new String("Houston we have a problem");  
        word = "Java";  
        word = word + " " + phrase;  
        print(word);  
        print(word + " " + 4);  
        print( 4 + " " + word );  
        print(1 + 2 + " " + word);  
        print(word + " " + 1 + 2);  
    }  
  
    public static void print(String message) {  
        System.out.println( message );  
    }  
}
```

Ο εκπαιδευτικός μπορεί να δώσει αντίστοιχα παραδείγματα με μορφή φύλλων εργασίας και να ζητήσει από τους μαθητές να εξηγήσουν τα αποτελέσματα των παραπάνω εντολών. Ο ορισμός μιας απλής εντολής print, όπως φαίνεται παραπάνω είναι ένα παράδειγμα για το πώς μπορούμε να ορίσουμε μια δική μας διασύνδεση για την είσοδο-έξοδο δεδομένων ώστε να αποφύγουμε τον διδακτικό θόρυβο που προκύπτει από την χρήση των μεθόδων εισόδου-εξόδου της Java, κάτι που αποτελεί και ένα από τα μεγάλα μειονεκτήματά της κατά την εισαγωγή της σε αρχάριους προγραμματιστές. Για αυτό τον λόγο προτείνεται η βιβλιοθήκη JTF της ACM που σχεδιάστηκε για αυτόν ακριβώς τον σκοπό, τη διευκόλυνση των μαθητών όσον αφορά τις λειτουργίες εισόδου-εξόδου της Java.

Επίσης η βιβλιοθήκη αυτή αντικαθιστά τη μέθοδο main με τη run η οποία είναι πολύ πιο φιλική στον αρχάριο προγραμματιστή. Το παραπάνω πρόγραμμα με χρήση της βιβλιοθήκης της ACM παίρνει την παρακάτω μορφή:

```

import acm.program.*;

public class Test extends Program {

    public void run( ) {
        println("Το πρώτο μου .... απλό πρόγραμμα");

        String word = "Java";
        String phrase = new String("Houston we have a problem");
        word = "Java";
        word = word + " " + phrase;
        print(word);
        print(word + " " + 4);
        print( 4 + " " + word );
        print(1 + 2 + " " + word);
        print(word + " " + 1 + 2);
    }
}

```

Η μέθοδος `print` είναι ορισμένη στη βιβλιοθήκη `JTF`. Δεν χρειάζεται να την ορίσουμε εμείς ώστε να μειώσουμε τον συντακτικό θόρυβο για τους μαθητές.

Οι συμβολοσειρές (`String`) μπορούν να αποτελέσουν και μια πρώτης τάξεως ευκαιρία για την εισαγωγή των μαθητών στους πίνακες, όχι τόσο λειτουργικά αλλά περισσότερο εννοιολογικά, αφού πρόκειται ουσιαστικά για πίνακες από χαρακτήρες. Μπορεί να μην επιτρέπεται ο συμβολισμός `[]` για την προσπέλαση σε στοιχείο του πίνακα, ωστόσο υπάρχει η μέθοδος `charAt`, την οποία μπορούμε να χρησιμοποιήσουμε για να διατρέξουμε όλους τους χαρακτήρες ενός `String`.

### 3.2 Δημιουργία αντικειμένων

Ο εκπαιδευτικός πρέπει να εξηγήσει στους μαθητές την ανάγκη της δημιουργίας αντικειμένων από μια κλάση κι ότι δεν αρκεί απλά η δήλωση μιας κλάσης.

Ένα παράδειγμα για να κατανοήσουν την ανάγκη δημιουργίας αντικειμένων είναι η παρομοίωση μιας κλάσης με μια συνταγή για κέικ. Για να φτιάξουμε ένα κέικ χρειαζόμαστε μια συνταγή. Η διαφορά μεταξύ της συνταγής (κλάση) και του κέικ (στιγμιότυπο) είναι προφανής. Το κέικ μπορούμε να το φάμε όταν ψηθεί, αλλά δεν μπορούμε να φάμε τη συνταγή (εκτός κι αν μας αρέσει να μασάμε χαρτί). Όπως στην περίπτωση του κέικ, έτσι και στον αντικειμενοστρεφή προγραμματισμό, τα αντικείμενα δημιουργούνται σύμφωνα με αυτά που ορίζονται στην κλάση τους. Αν θέλουμε να ψήσουμε το κέικ χρειαζόμαστε τα υλικά και τις οδηγίες πώς να το παρασκευάσουμε. Αντίστοιχα, αν θέλουμε να δημιουργήσουμε αντικείμενα από μια κλάση, χρειάζεται να ορίσουμε τις ιδιότητες και τις μεθόδους.

### 3.3 Μέθοδοι

Η επόμενη σημαντική έννοια στην οποία θα εισαχθούν οι μαθητές είναι η έννοια της μεθόδου. Οι μαθητές ήδη γνωρίζουν την έννοια της συνάρτησης από το μάθημα του προγραμματισμού της Β Λυκείου, οπότε έννοιες όπως τύπος επιστροφής και παράμετροι τους είναι οικείες. Στις μεθόδους θα πρέπει να επικεντρωθούμε σε δυο σημεία. Το πρώτο αφορά το γεγονός ότι μια μέθοδος ανήκει σε ένα αντικείμενο και καθορίζει τη συμπεριφορά του. Εδώ φαίνεται ότι ένα αντικείμενο δεν περιλαμβάνει μόνο δεδομένα αλλά και κάποια λειτουργικότητα η οποία ορίζεται από τις μεθόδους του. Το δεύτερο σημείο που πρέπει να δώσουμε προσοχή είναι ο μηχανισμός μεταβίβασης παραμέτρων ο οποίος για τους βασικούς τύπους (`int`, `float`, etc) είναι κατά τιμή (`call by value`) αλλά για τα αντικείμενα είναι κατ'αναφορά (`call by reference`), αφού αυτό που αντιγράφεται δεν είναι ολόκληρο το αντικείμενο αλλά η αναφορά στην περιοχή μνήμης του αντικειμένου. Αυτό που πρέπει να συγκρατήσουν οι μαθητές

είναι ότι, στην πρώτη περίπτωση μεταβιβάζεται ένα αντίγραφο της μεταβλητής, ενώ στη δεύτερη το ίδιο αντικείμενο.

Αν ο/η εκπαιδευτικός κρίνει ότι είναι ακόμα νωρίς για αυτή τη συζήτηση, μπορεί να την αναβάλλει για την επόμενη ενότητα αφού πρώτα οι μαθητές εξοικειωθούν με τη δημιουργία νέων κλάσεων και των κατασκευαστών τους.

### 3.4 Κληρονομικότητα

Η κληρονομικότητα πρέπει να προσεγγιστεί εννοιολογικά και όχι προγραμματιστικά. Στο στάδιο αυτό δεν πρέπει να σταθούμε σε τεχνικές λεπτομέρειες, αλλά στην ουσία της αφαίρεσης που συντελείται μέσω του ορισμού μιας κλάσης. Προτείνεται, να μην παρουσιαστεί στον μαθητή η δημιουργία των αντικειμένων ως επακόλουθο του ορισμού της κλάσης, αλλά το αντίστροφο, δηλαδή ο ορισμός μιας κλάσης ως το σύνολο των κοινών χαρακτηριστικών μιας ομάδας αντικειμένων. Αυτό μπορεί να γίνει μέσω παραδειγμάτων από τον πραγματικό κόσμο, π.χ. αυτοκίνητα, ζώα κλπ. Επίσης θα πρέπει να αναφερθεί ότι η κληρονομικότητα μεταξύ δυο κλάσεων είναι μια *is-a* σχέση, αφού κάθε αντικείμενο της υποκλάσης είναι και αντικείμενο της γονικής κλάσης, π.χ. η χελώνα και η αρκούδα είναι και τα δυο ζώα.

### 3.5 Καθοδήγηση από γεγονότα

Μέχρι στιγμής οι μαθητές έχουν στο μυαλό τους το παραδοσιακό μοντέλο του προστακτικού προγραμματισμού, δηλαδή ένα μοντέλο, όπου σε ένα κύριο πρόγραμμα εκτελούνται εντολές η μια μετά την άλλη. Μέσω του Greenfoot και δραστηριοτήτων, όπως η 16<sup>η</sup> εργαστηριακή άσκηση, οι μαθητές θα έρθουν σε επαφή με τον γεγονοστρεφή προγραμματισμό και θα δουν ότι πλέον γράφουν κώδικα ο οποίος συνδέεται με την εκτέλεση κάποιου συμβάντος, όπως το πάτημα ενός πλήκτρου ή την κίνηση του ποντικιού. Προτείνεται οι μαθητές αρχικά να εκτελέσουν την άσκηση όπως δίνεται και στη συνέχεια να την επεκτείνουν βάσει των οδηγιών.

#### Στόχοι

Οι μαθητές να μπορούν να:

- Αναγνωρίζουν και να περιγράφουν τον τρόπο δημιουργίας αντικειμένων
- Αναζητούν και να χρησιμοποιούν τις μεθόδους που χρειάζονται αν αυτές υπάρχουν
- Χρησιμοποιούν αντικείμενα έτοιμων τύπων για την επίλυση προβλημάτων
- Εξηγούν τον μηχανισμό μεταβίβασης παραμέτρων
- Αναγνωρίζουν τον τύπο επιστροφής μιας μεθόδου
- Υλοποιούν μεθόδους για τον χειρισμό γεγονότων του ποντικιού ή του πληκτρολογίου

## 4 Αντικειμενοστρεφής προγραμματισμός

Στο κεφάλαιο αυτό παρουσιάζονται τα βασικά χαρακτηριστικά του αντικειμενοστρεφούς προγραμματισμού που είναι:

- Ενθυλάκωση (Encapsulation)
- Κληρονομικότητα (Inheritance)
- Πολυμορφισμός (Polymorphism)

Το κύριο προγραμματιστικό περιβάλλον που θα χρησιμοποιήσουν οι μαθητές είναι το Eclipse. Το Eclipse αποτελεί ελεύθερο λογισμικό το οποίο μπορείτε να το κατεβάσετε από την ομώνυμη διεύθυνση <http://www.eclipse.org>.

Επίσης εισάγουμε και την βιβλιοθήκη JTF (Java Task Force) της ACM, η οποία έχει αναπτυχθεί για εκπαιδευτική χρήση και έχει στόχο να διευκολύνει την ανάπτυξη εφαρμογών στη γλώσσα Java. Η χρήση της βιβλιοθήκης μας δίνει τη δυνατότητα να χρησιμοποιήσουμε έτοιμα αντικείμενα και μεθόδους για είσοδο/έξοδο δεδομένων, σχεδιασμό γραφικής διεπαφής ή χειρισμό γεγονότων, αποφεύγοντας επουσιώδεις τεχνικές λεπτομέρειες. Μπορείτε να κατεβάσετε τη βιβλιοθήκη από τον δικτυακό τόπο <http://jtf.acm.org>. Στον ίδιο δικτυακό τόπο υπάρχουν και οι απαραίτητοι οδηγοί για τη χρήση της βιβλιοθήκης.

Παρακάτω φαίνεται ένα απλό πρόγραμμα με χρήση της βιβλιοθήκης της ACM.

```
import acm.program.*;

public class Test extends Program {

    public void run( ) {
        println("Το πρώτο μου .... απλό πρόγραμμα");
    }
}
```

Με την εκτέλεση του προγράμματος εκτελείται η μέθοδος run. Η κλάση Test κληρονομεί από την Program της ACM διάφορα πακέτα που περιλαμβάνουν μεθόδους εισόδου/εξόδου όπως είναι η println, για να κάνουμε τη ζωή μας πιο εύκολη.

Η προσθήκη της JTF στο Eclipse γίνεται πολύ εύκολα. Αφού δημιουργήσετε ένα νέο project πρέπει να προσθέσετε το acm.jar στις εξωτερικές βιβλιοθήκες (external libraries/jars). Στη συνέχεια με δεξί κλικ πάνω στο project στο αριστερό παράθυρο (Package Explorer) του Eclipse επιλέγουμε *Properties* → *Java Build Path* → *Add Jars* εφόσον έχουμε προσθέσει το αρχείο στο φάκελο του Project ή αν βρίσκεται σε άλλο φάκελο *Properties* → *Java Build Path* → *Add External Jars*.

Αν δεν χρησιμοποιείτε το Eclipse αλλά προτιμάτε να δουλεύετε από τη γραμμή εντολών, τότε θα πρέπει να συμπεριλάβετε τη βιβλιοθήκη και κατά τη μεταγλώττιση και κατά την εκτέλεση.

Προτείνουμε να μελετήσετε τους παρακάτω οδηγούς της βιβλιοθήκης JTF (αρχεία με όνομα Tutorial.pdf, Rational.pdf) που βρίσκονται αναρτημένα στον δικτυακό τόπο της βιβλιοθήκης.

### 4.1 Κληρονομικότητα

Στην προηγούμενη ενότητα η κληρονομικότητα παρουσιάστηκε στους μαθητές σε ένα πιο αφηρημένο επίπεδο με παραδείγματα από τον πραγματικό κόσμο, ώστε να γίνουν οι απαραίτητες συνδέσεις. Εδώ σκοπός είναι να αναγνωρίσουν οι μαθητές τα προγραμματιστικά οφέλη της κληρονομικότητας. Ένα πολύ καλό παράδειγμα για να ξεκινήσουμε είναι αυτό που αναφέρεται στις σημειώσεις, όπου το ίδιο ακριβώς πρόγραμμα εκτελείται δεύτερη φορά με μια μικρή αλλαγή. Αντί να κληρονομεί από την κλάση ConsoleProgram κληρονομεί από την DialogProgram. Η αλλαγή αυτή αλλάζει τα πάντα όσον αφορά τις λειτουργίες εισόδου-εξόδου και εκεί που το πρόγραμμα χρησιμοποιούσε τη γραμμή εντολών τώρα εμφανίζει παράθυρα διαλόγου με τον χρήστη.

```

import acm.program.*;

public class Test extends ConsoleProgram {

    public void run( ) {
        println("*****");
        println("***** Πρόσθεση Αριθμών *****");
        println("*****");
        int a = readInt("Δώσε τον α = " );
        int b = readInt("Δώσε τον β = " );
        int sum = a + b
        println("Το συνολικό άθροισμα είναι : " + sum);
    }
}

```

Αρκεί να αντικαταστήσουμε το ConsoleProgram με το DialogProgram και το πρόγραμμά μας, από μια εφαρμογή που εκτελείται στην γραμμή εντολών, θα γίνει παραθυρική εφαρμογή που χρησιμοποιεί πλαίσια διαλόγου για την επικοινωνία με τον χρήστη. Εδώ πρέπει να τονίσουμε στους μαθητές ότι αυτή η δυνατότητα είναι εφικτή επειδή οι κλάσεις ConsoleProgram και DialogProgram προέρχονται από την ίδια γονική κλάση Program, οπότε έχουν μια κοινή διασύνδεση (interface) την οποία χρησιμοποιούμε.

## 4.2 Κατασκευαστές

Οι μαθητές θα πρέπει να μάθουν να υλοποιούν πάντα για την κλάση τους έναν τουλάχιστον κατασκευαστή. Να γίνει ειδική μνεία στον προκαθορισμένο κατασκευαστή (default constructor) και αν το κρίνεται απαραίτητο και στον κατασκευαστή αντιγράφων (copy constructor) με μερικά παραδείγματα.

## 4.3 Αναφορές και μεταβίβαση παραμέτρων

Για να κατανοήσουν οι μαθητές τη διαφορά των δυο μηχανισμών μεταβίβασης παραμέτρων, πρέπει υποχρεωτικά να αναφερθούμε στην έννοια της αναφοράς σε αντικείμενο, ως διεύθυνση του αντικειμένου στη μνήμη. Αυτή η έννοια είναι πολύ σημαντική στην Java, αφού όλες οι μεταβλητές-αντικείμενα της Java δεν περιέχουν το ίδιο το αντικείμενο, αλλά μια αναφορά στην περιοχή της μνήμης, όπου αυτό βρίσκεται αποθηκευμένο.

Ένα καλό παράδειγμα για να δείξουμε τι εννοούμε στους μαθητές είναι να τους επιτρέψουμε να διερευνήσουν τη λειτουργία προγραμμάτων όπως αυτό:

```

import javax.swing.JOptionPane;

public class MainProgram {

    public static void main(String args[]) {
        Student daidalos = new Student(18, "Δαίδαλος"),
            ikaros = new Student(16, "Ίκαρος");
        print("Το όνομα του Δαίδαλου είναι " + daidalos.name);
        daidalos = ikaros;
        daidalos.name = "Μινώταυρος";
        print("Το όνομα του Ίκαρου είναι " + ikaros.name);
    }

    public static void print(String message) {
        JOptionPane.showMessageDialog(null, message,
            "Μεταβίβαση με αναφορά", JOptionPane.PLAIN_MESSAGE);
    }
}

```

Είναι σημαντικό να τονίσουμε στους μαθητές ότι το αποτέλεσμα της εντολής

```
daidalos = ikaros;
```

είναι οι δυο μεταβλητές να δείχνουν στο ίδιο αντικείμενο `ikaros`, άρα οποιαδήποτε αλλαγή γίνει την “βλέπουν” και οι δυο. Μια πολύ καλή μεταφορά σχετικά με τη σχέση της έννοιας της αναφοράς με αυτή του αντικειμένου περιγράφεται στο βιβλίο του *Bruce Eckel*, «*Thinking in Java*» (4<sup>η</sup> έκδοση), όπου η αναφορά παραλληλίζεται με ένα τηλεκοντρόλ και το αντικείμενο με την τηλεόραση. Δεν χρειάζεται να μεταφέρουμε μαζί μας την τηλεόραση για να αλλάξουμε κανάλι παρά μόνο το τηλεκοντρόλ και μέσω αυτού χειριζόμαστε την τηλεόραση. Άρα η αναφορά ενός αντικειμένου δεν είναι παρά ένα τηλεχειριστήριο για τον χειρισμό της συμπεριφοράς του αντικειμένου. Τα κουμπιά του τηλεχειριστηρίου αποτελούν τον τρόπο με τον οποίο στέλνουμε μηνύματα στο αντικείμενο καλώντας τις μεθόδους του.

Μπορούμε επίσης να σχεδιάσουμε και ένα σχήμα στον πίνακα, στο οποίο να φαίνεται, που δείχνουν οι αναφορές στην μνήμη, πριν και μετά την εντολή εκχώρησης. Εδώ μπορεί να γίνει και μια συζήτηση σχετικά με την τύχη του αντικειμένου στο οποίο πλέον δεν δείχνει καμία αναφορά, ώστε να εισάγουμε την αναγκαιότητα του συλλέκτη απορριμμάτων για αυτές τις περιπτώσεις.

Στο προηγούμενο παράδειγμα η μέθοδος `mygrint` έχει οριστεί ως μια βοηθητική συνάρτηση, η οποία εμφανίζει το μήνυμα που θέλουμε σε ένα πλαίσιο διαλόγου. Όπως φαίνεται έχουμε παραβιάσει την αρχή της ενθυλάκωσης, αφού έχουμε ορίσει τα δυο μέλη της `Student` ως δημόσια, για χάριν της απλότητας, ώστε να επικεντρωθούν οι μαθητές στις εντολές εκχώρησης μεταξύ των μεταβλητών και όχι σε τεχνικά χαρακτηριστικά που μπορούμε να εξηγήσουμε αργότερα. Στη συνέχεια ακολουθεί ο λιτός ορισμός της κλάσης `Student`.

```
public class Student {
    public int grade;
    public String name;

    public Student(int vathmos, String onoma) {
        grade = vathmos;
        name = onoma;
    }
}
```

#### 4.4 Πολυμορφισμός

Οι μαθητές μπορούν πειραματιστούν με το παράδειγμα των σημειώσεων της ενότητας 4.6, όπου έχει δημιουργηθεί μια λίστα με αντικείμενα της κλάσης `Person`. Όταν προσπαθούμε να εξηγήσουμε τον πολυμορφισμό στους μαθητές θα πρέπει να επικεντρώσουμε στα εξής βασικά σημεία:

- Πρέπει όλα τα είδη των αντικειμένων να προέρχονται από μια γονική κλάση.
- Η πολυμορφική συμπεριφορά έχει νόημα μόνο για τις μεθόδους που ανήκουν στην διασύνδεση της γονικής κλάσης και μόνο για αυτές.
- Οι μέθοδοι της γονικής κλάσης πρέπει να έχουν οριστεί ξεχωριστά για κάθε υποκλάση για να έχει νόημα ο πολυμορφισμός.

Η έννοια του πολυμορφισμού είναι διδακτικά αρκετά δύσκολο να εξηγηθεί στους μαθητές, για αυτό καλό θα ήταν να ακολουθηθεί μια καθαρά διερευνητική προσέγγιση μέσα από ένα παράδειγμα, όπως αυτό της ενότητας 4.6, το οποίο στη συνέχεια θα πρέπει να επεκτείνουν οι μαθητές προσθέτοντας και αντικείμενα νέων κλάσεων. Εδώ θα πρέπει να επισημανθεί ότι ο κώδικας, που χρησιμοποιεί αντικείμενα της γονικής κλάσης, εκτελείται κανονικά και για αντικείμενα κλάσεων που έχουν δημιουργηθεί πρόσφατα, χωρίς να χρειαστεί καμία απολύτως αλλαγή. Αυτό συμβαίνει γιατί δεν έχει αλλάξει καθόλου η διασύνδεση (interface) της κλάσης.

#### Στόχοι

Οι μαθητές να μπορούν να:

- Ορίζουν τις δικές τους κλάσεις
- Επεκτείνουν ήδη υπάρχουσες κλάσεις

- Αναγνωρίζουν την επαναχρησιμοποίηση κώδικα μέσα από την κληρονομικότητα
- Εξηγούν τη διαφορά μεταξύ υπερφόρτωσης και πολυμορφισμού
- Δημιουργούν τα δικά τους πακέτα ως βιβλιοθήκες λογισμικού
- Αναγνωρίζουν τα πλεονεκτήματα του αντικειμενοστρεφούς προγραμματισμού

## 5 Προγραμματισμός οδηγούμενος από γεγονότα

Η Java παρέχει την βιβλιοθήκη AWT (Abstract Windowing Toolkit) το οποίο είναι ένα σύνολο κλάσεων για τη σχεδίαση παραθυρικών εφαρμογών και την υποστήριξη της διαδραστικότητάς τους. Οι εφαρμογές που βασίζονται στην βιβλιοθήκη AWT έχουν την ίδια συμπεριφορά και εμφάνιση της πλατφόρμας στην οποία τρέχουν. Δηλαδή, η ίδια ακριβώς εφαρμογή έχει διαφορετική εμφάνιση, ανάλογα με την πλατφόρμα στην οποία εκτελείται. Αυτό συμβαίνει γιατί, για κάθε μια τέτοια πλατφόρμα, υπάρχει μια εικονική μηχανή (JVM) για εκτέλεση στην συγκεκριμένη πλατφόρμα.

Ωστόσο ενώ η βιβλιοθήκη AWT επιτυγχάνει πολύ καλά την ανεξαρτησία από την πλατφόρμα, μειονεκτεί στην ευελιξία και στην ανάπτυξη απαιτητικών GUI. Για αυτό το λόγο έχει πλέον αντικατασταθεί από μια άλλη βιβλιοθήκη η οποία είναι πολύ πιο ευέλικτη, πιο αποδοτική και επιτρέπει την ανάπτυξη πιο σύνθετων εφαρμογών, την βιβλιοθήκη swing. Επίσης τα αντικείμενα της βιβλιοθήκης swing είναι πολύ πιο ελαφριά (lightweight), όσον αφορά τη κατανάλωση υπολογιστικών πόρων.

Στο μάθημα αυτό δεν θα ασχοληθούμε με τις τεχνικές λεπτομέρειες της βιβλιοθήκης αλλά θα χρησιμοποιήσουμε την αντίστοιχη βιβλιοθήκη της ACM `acm.gui` και `acm.graphics` οι οποίες κρύβουν αρκετές από τις πολύπλοκες λεπτομέρειες της AWT και βοηθούν στο να γίνει η χρήση τους εύκολη και επεξηγηματική.

Ωστόσο δεν θα πρέπει οι μαθητές να αποπροσανατολιστούν με τις λεπτομέρειες σχεδιασμού μιας γραφικής διεπαφής αλλά να εξοικειωθούν με την ανάπτυξη εφαρμογών για τον χειρισμό γεγονότων των μονάδων εισόδου, μέσω της υλοποίησης χειριστών γεγονότων (event handlers).

Τα παραδείγματα που δίνονται στην ενότητα αυτή είναι χαρακτηριστικά και επαρκούν για να πειραματιστούν οι μαθητές και να επεκτείνουν τη λειτουργικότητά τους. Μια ενδιαφέρουσα εφαρμογή με την οποία μπορούν να ασχοληθούν οι μαθητές είναι η ανάπτυξη ενός προγράμματος για το παιχνίδι της τρίλιζας. Η εφαρμογή αυτή χωρίζεται σε τέσσερις φάσεις:

1. Η γραφική διεπαφή
2. Ο χειρισμός των γεγονότων κυρίως του ποντικιού. Όταν ο χρήστης επιλέγει με το ποντίκι ένα τετράγωνο, πρέπει να επιστρέφονται οι συντεταγμένες του τετραγώνου στο πλέγμα 3x3 της τρίλιζας
3. Τέλος, θα πρέπει να υλοποιηθεί η λογική του παιχνιδιού, δηλαδή μια μέθοδος, που κάθε φορά που θα παίζει ένας παίκτης, θα ελέγχει αν κέρδισε ή ήρθε ισοπαλία.
4. Προαιρετικά, μπορούμε να καθοδηγήσουμε τους μαθητές να υλοποιήσουν έναν αλγόριθμο, ώστε να παίζει τρίλιζα ο χρήστης με τον υπολογιστή. Αυτό προϋποθέτει ότι οι μαθητές θα έχουν παίξει αρκετά παιχνίδια μεταξύ τους με την αρχική έκδοση του λογισμικού, έτσι ώστε να εξάγουν κάποιους κανόνες, οι οποίοι θα αποτελέσουν και τη λογική του κεντρικού αλγορίθμου.

Μια αντίστοιχη εφαρμογή σε Java διατίθεται στον δικτυακό τόπο της acm μαζί με άλλες υποδειγματικές εφαρμογές που έχουν αναπτυχθεί με τη χρήση αυτής της βιβλιοθήκης.

### Στόχοι

Οι μαθητές να μπορούν να:

- Σχεδιάζουν απλές γραφικές διεπαφές
- Χρησιμοποιούν πλαίσια διαλόγου για επικοινωνία με τον χρήστη
- Υλοποιούν μεθόδους για τον χειρισμό γεγονότων του ποντικιού ή του πληκτρολογίου
- Αναλύουν την εφαρμογή τους σαν ένα σύνολο μεθόδων κάθε μια από τις οποίες ενεργοποιείται από ένα γεγονός

## 6 Βάσεις Δεδομένων

Μια βάση δεδομένων είναι μια οργανωμένη συλλογή από δεδομένα, έτσι ώστε οι απαραίτητες λειτουργίες (εισαγωγή, διαγραφή, αναζήτηση, ενημέρωση) να γίνονται με απλό, αποδοτικό και συστηματικό τρόπο. Τα συστήματα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ) παρέχουν αποδοτικούς και αποτελεσματικούς τρόπους διαχείρισης και προσπέλασης σε βάσεις δεδομένων. Σήμερα το επικρατέστερο μοντέλο βάσεων δεδομένων είναι το σχεσιακό μοντέλο (relational model) στο οποίο τα δεδομένα είναι οργανωμένα σε πίνακες και η διαχείρισή τους γίνεται με τη γλώσσα SQL (Structured Query Language).

Η Java παρέχει απλούς μηχανισμούς επικοινωνίας με ένα ΣΔΒΔ μέσα από τη βιβλιοθήκη της JDBC (Java Database Connectivity). Το δυνατό σημείο του JDBC API είναι, ότι πρόκειται για ένα καθολικό API, το οποίο ισχύει για όλα τα είδη βάσεων δεδομένων. Αν δηλαδή αλλάξουμε ξαφνικά Σύστημα Διαχείρισης Βάσεων Δεδομένων οι αλλαγές που θα πρέπει να κάνουμε στον κώδικα είναι μηδαμινές.

Προτείνεται να γίνει μια σύντομη υπενθύμιση των εντολών της γλώσσας SQL η οποία θεωρείται γνωστή από το μάθημα “Βάσεις Δεδομένων”. Δεν χρειάζεται να σχεδιάσουμε βάσεις δεδομένων με πολλούς πίνακες και πολύπλοκες εξαρτήσεις. Ο σκοπός δεν είναι να εξοικειωθούν οι μαθητές με το σχεσιακό μοντέλο αλλά με τους μηχανισμούς επικοινωνίας μιας εφαρμογής με μια βάση δεδομένων.

Αρχικά μπορούμε να δώσουμε στους μαθητές τις προδιαγραφές μιας σχετικά απλής βάσης δεδομένων, όπως είναι για παράδειγμα μια δανειστική βιβλιοθήκη, που περιέχει τις οντότητες ΒΙΒΛΙΟ, ΣΥΓΓΡΑΦΕΑΣ, ΧΡΗΣΤΗΣ. Αφού σχεδιάσουμε τους πίνακες της βάσης, δημιουργούμε τη βάση μας κατά προτίμηση στην MySQL. Το επόμενο βήμα είναι να γράψουμε τον κώδικα που θα εκτελεί τη σύνδεση της εφαρμογής με τη βάση μας και στη συνέχεια να υλοποιήσουμε τις απαραίτητες μεθόδους για αναζήτηση, ενημέρωση και εισαγωγή δεδομένων στη βάση. Η επικοινωνία με τον χρήστη μπορεί να γίνει με κατάλληλη γραφική διεπαφή που θα σχεδιαστεί για κάθε είδος συναλλαγής με τη βάση, π.χ. αναζήτηση βιβλίου, δανεισμός, επιστροφή κλπ.

### Στόχοι

Οι μαθητές να μπορούν να:

- Υλοποιούν τη σύνδεση της εφαρμογής τους με τη βάση δεδομένων
- Σχεδιάζουν τις κλάσεις της εφαρμογής με βάση το σχεσιακό μοντέλο της βάσης δεδομένων
- Υλοποιούν μεθόδους για την αναζήτηση, ενημέρωση και εισαγωγή δεδομένων στη βάση

## 7 Δικτυακός Προγραμματισμός

### Εισαγωγή

Ο προγραμματισμός σε επίπεδο δικτύου ή ιστού είναι ένα από τα πιο δυνατά σημεία της Java. Για την Java η διαχείριση των δεδομένων, είτε αυτά προέρχονται από κάποιο αρχείο, είτε από μια ιστοσελίδα στο δίκτυο, γίνεται με τον ίδιο ακριβώς τρόπο, με τη χρήση ρευμάτων (streams). Αυτό διευκολύνει σε μεγάλο βαθμό την ανάπτυξη δικτυακών εφαρμογών. Το πακέτο java.net προσφέρει επικοινωνία μέσω των πρωτοκόλλων TCP/IP και UDP παρέχοντας για το σκοπό αυτό κλάσεις όπως URL, URLConnection, Socket, και ServerSocket για το πρωτόκολλο TCP, και DatagramPacket, DatagramSocket, και MulticastSocket για το πρωτόκολλο UDP.

Προτείνεται να υλοποιηθούν από τους μαθητές τα παραδείγματα με τις κλάσεις URLConnection, Socket και ServerSocket στο εργαστήριο. Η υλοποίηση μιας εφαρμογής ανταλλαγής άμεσων μηνυμάτων σε πραγματικό χρόνο (chat) με γραφική διεπαφή και ο πειραματισμός με αυτή στο εργαστήριο θα μπορούσε να είναι ο τελικός στόχος αυτής της ενότητας.

### Στόχοι

Οι μαθητές να μπορούν να:

- Χρησιμοποιούν αντικείμενα URLs και Sockets για την ανάπτυξη δικτυακών εφαρμογών



- Διακρίνουν τη διαφορά μεταξύ πελάτη και εξυπηρετητή σε μια δικτυακή εφαρμογή
- Να σχεδιάζουν εφαρμογές βασισμένες στην αρχιτεκτονική πελάτη εξυπηρετητή

## 8 Ανάπτυξη Ολοκληρωμένης Εφαρμογής

Σε αυτό το κεφάλαιο θα παρουσιάσουμε συνοπτικά τα στάδια ανάπτυξης μιας ολοκληρωμένης εφαρμογής με βάση τις αρχές του αντικειμενοστρεφούς προγραμματισμού. Δεν θα επικεντρωθούμε στα αρχικά στάδια της ανάλυσης, διότι αυτά αναλύονται πολύ καλά στο μάθημα “Σχεδιασμός και Ανάπτυξη Δικτυακών Εφαρμογών”. Ο σκοπός της ενότητας είναι να μελετήσουμε κάποια εργαλεία όπως το Ant, για την ολοκλήρωση και τη συντήρηση εφαρμογών σε Java και τα JUnit / TestNG, για τον συστηματικό έλεγχο της εφαρμογής. Θα πούμε επίσης λίγα λόγια για το μοντέλο σχεδιασμού Model – View – Controller (MVC), για τον διαχωρισμό της γραφικής διεπαφής με τη λογική και τον έλεγχο της εφαρμογής. Τέλος, θα δείξουμε πως μπορούμε να παράγουμε αυτόματα την τεκμηρίωση για το έργο μας μέσω του εργαλείου javadoc.

Σειρά από βίντεο για την ανάπτυξη εφαρμογής java στο περιβάλλον eclipse:

<http://eclipsetutorial.sourceforge.net/totalbeginner.html>

Μια πλήρης ανάπτυξη ολοκληρωμένης εφαρμογής, με οδηγίες, για την προσομοίωση ενός ATM:

<http://www.math-cs.gordon.edu/courses/cs211/ATMExample/>

### Στόχοι

Οι μαθητές να μπορούν να:

- Διαχωρίζουν τον κώδικα της γραφικής διεπαφής από τον υπόλοιπο κώδικα μέσω τεχνικών αντικειμενοστρεφούς σχεδίασης
- Υλοποιούν μια εφαρμογή αν δίνεται το διάγραμμα κλάσεων και οι περιγραφές των μεθόδων
- Παράγουν την τεκμηρίωση της εφαρμογής μέσω της υπηρεσίας javadoc
- Κάνουν δοκιμές του τελικού λογισμικού και να διορθώνουν πιθανά λάθη και παραλείψεις

*Τα Κεφάλαια 6, 7 και 8 αποτελούν μία πρώτη εισαγωγή στα αντικείμενα που πραγματεύονται, διδάσκονται με μορφή επίδειξης από τον διδάσκοντα και με έμφαση στις εξηγήσεις των βασικών εννοιών προς τους μαθητές. Οι μαθητές καθοδηγούνται από τον διδάσκοντα στην υλοποίηση των Δραστηριοτήτων, που παρατίθενται στις σημειώσεις, στο πλαίσιο του διαθέσιμου χρόνου και ενθαρρύνονται σε μελλοντική ενασχόληση με αυτές. Δεν αποτελεί σκοπό η εμβάθυνση των μαθητών στις έννοιες που διαπραγματεύονται σε αυτά τα Κεφάλαια, αλλά η γνωριμία με αυτές, ώστε να αποκτήσουν μία πληρέστερη εικόνα για τις δυνατότητες της γλώσσας και ιδιαίτερα στο Δικτυακό Προγραμματισμό. Σε κάθε περίπτωση, χρειάζεται προσήλωση στους στόχους που τίθενται σε κάθε Κεφάλαιο.*

---

## ΕΝΟΤΗΤΑ 2

# ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ (APPS) ΓΙΑ ANDROID

---

### Εισαγωγή

Το δεύτερο μέρος στις παρούσες σημειώσεις έχει σκοπό να εξοικειώσει τους μαθητές με την ανάπτυξη ολοκληρωμένων εφαρμογών για την πλατφόρμα Android. Η πλατφόρμα αυτή αφορά κυρίως στην περίπτωση των κινητών συσκευών και ταμπλετών όπου ο ρόλος της στην αγορά είναι σημαντικός. Παρουσιάζει δε σημαντικές ιδιοτυπίες καθώς όχι απλά επιτρέπει αλλά διευκολύνει ιδιαίτερα την επικοινωνία μεταξύ εφαρμογών, την αξιοποίηση τμημάτων της μίας από την άλλη και την από κοινού χρήση πόρων.

Οι επί μέρους ενότητες 2A και 2B του δεύτερου μέρους αποτελούν ένα αρθρωτό σύνολο. Προβλέπονται για τη διδασκαλία τους δύο εναλλακτικά σενάρια. Στο πρώτο η 2A καλύπτει 12Θ+34Ε ώρες και η 2B 1Θ + 13Ε. Η ενότητα 2A αφορά στην ανάπτυξη εφαρμογών, χρησιμοποιώντας πακέτα ανάπτυξης όπως το Eclipse ή το Android Studio και αξιοποίηση java και XML. Η ενότητα 2B αφορά στην ανάπτυξη εφαρμογών στο περιβάλλον ανάπτυξης AppInventor, που χρησιμοποιεί προγραμματισμό ψηφίδων και εστιάζει στην ανάπτυξη εφαρμογής από υπάρχοντα κώδικα από τους μαθητές. Στο δεύτερο σενάριο (**το οποίο εφαρμόζεται μόνο στην περίπτωση που οι συνθήκες του εργαστηρίου δεν επιτρέπουν εγκατάσταση πακέτου ανάπτυξης εφαρμογών για Android με Java (όπως συνδυασμό των SDK και ADT με το Eclipse ή Android Studio) και μετά από συζήτηση με τον αρμόδιο σχολικό Σύμβουλο**), παρουσιάζεται με μορφή επίδειξης από τον καθηγητή το μέρος 2A (11Θ) και οι διαθέσιμες ώρες διδασκαλίας αφιερώνονται στην ενότητα 2B (2Θ +47Ε).

Η ενότητα 2A αφορά, στην εκτεταμένη της μορφή, στην ανάπτυξη μιας εφαρμογής στο πλαίσιο ενός σχολικού εργαστηρίου. Έτσι έχει τη λογική ενός υποστηρικτικού υλικού, που θα χρησιμοποιηθεί μέσα σε ένα σχολικό εργαστήριο, με την παρουσία του/των εκπαιδευτικών, στο πλαίσιο μιας υποστηριζόμενης εργαστηριακής πράξης. Δεν έχει τη λογική αυτόνομου υλικού για ατομική μελέτη ή ενός βιβλίου που εξαντλεί το ούτως ή άλλως εφαρμοσμένο αντικείμενο. Μια επανάληψη κάποιων τμημάτων έχει τη λογική μιας σπείρας Bruner, όπου η δεύτερη εμφάνιση συνδυάζεται με μεγαλύτερη εμβάθυνση. [Στην πρώτη επίδειξη, στη δεύτερη υλοποίηση.]

Η ενότητα 2B έχει αποκλειστικό σκοπό την ανάπτυξη μιας αρθρωτής εφαρμογής στο περιβάλλον προγραμματισμού με ψηφίδες AppInventor. Συνδυάζεται έτσι η ευκολία ανάπτυξης του συγκεκριμένου περιβάλλοντος με τις πλούσιες δυνατότητες της πλατφόρμας Android, για να διευκολυνθούν οι μαθητές στην κατανόηση της λειτουργίας μιας σύνθετης εφαρμογής. Η ενότητα αυτή είναι αναπόσπαστο τμήμα του δεύτερου μέρους των σημειώσεων που αφορά στο Android. Στην εκτεταμένη της μορφή συνδυάζεται με τμήματα θεωρίας της ενότητας 2A, με σκοπό να αποτελέσει το πλαίσιο για την εισαγωγή των σχετικών εννοιών.

Σκοπός του δεύτερου μέρους είναι η εφαρμοσμένη προσέγγιση στην ανάπτυξη εφαρμογών Android. Αν οι συνθήκες επιτρέπουν την αξιοποίηση των γνώσεων που αποκτήθηκαν στο 1ο μέρος και εδώ, αυτό θα ήταν για τους συγγραφείς το ιδανικό.

Σε κάθε περίπτωση συνιστάται στον εκπαιδευτικό η αξιοποίηση της λογικής της ανακαλυπτικής μάθησης, τόσο για την ανακάλυψη λεπτομερειών της εφαρμογής που έχουν εσκεμμένα παραληφθεί, όσο και για την επέκτασή της ξεκινώντας από πιθανές προτάσεις των μαθητών. Η ανάπτυξη μιας παράλληλης εφαρμογής της επιλογής τους θα μπορούσε να επιτρέψει σημαντική εμβάθυνση στη γνώση που κατακτούν οι μαθητές μέσα από την υλοποίηση των παραδειγματικών εφαρμογών των δύο ενότητων.

## Πηγές

- <http://www.eclipse.org>
- <http://developer.android.com>
- <https://www.android.com/>
- <http://android-developers.blogspot.com>
- <http://android-developers.blogspot.gr/>
- <http://ai2.appinventor.mit.edu>
- <http://www.appinventor.org/appInventor2>

## Υποενότητα 2A

### 1 Γνωριμία με προγραμματιστικά περιβάλλοντα για το Android

Στην πρώτη διαδρομή, όπου το κεφάλαιο διδάσκεται με 1Θ+2Ε, δίνεται έμφαση στην εισαγωγή της έννοιας του έργου και του ρόλου του στη διαχείριση και ανάπτυξη μεγάλων εφαρμογών στο πλαίσιο της θεωρίας και στην επίδειξη βασικών στοιχείων του περιβάλλοντος και τη συνακόλουθη εξοικείωση των μαθητών με αυτά στο εργαστήριο. Στην περίπτωση που η επιλογή περιβάλλοντος ανάπτυξης είναι αυτή του Eclipse με SDK και ADT, το περιβάλλον λίγο απέχει από αυτό που αντιμετωπίστηκε στην προηγούμενη ενότητα οπότε, με την προϋπόθεση ότι έχει δοθεί επαρκής έμφαση σε αυτό κατά το πρώτο μέρος, οι εργαστηριακές ώρες μπορούν, κατά την κρίση του εκπαιδευτικού, να διατεθούν σε άλλα κεφάλαια όπου είναι πιο απαραίτητες.

Στη δεύτερη διαδρομή, όπου το κεφάλαιο διδάσκεται με 1Θ, έμφαση δίνεται στη συζήτηση για την επιλογή περιβάλλοντος με βάση τα πλεονεκτήματα και τα μειονεκτήματα του, γενικά και διασταύρωση των απαιτήσεων προδιαγραφών του με τις υπάρχουσες δυνατότητες.

-Επιλογή ή αιτιολόγηση της επιλογής περιβάλλοντος

Συμπληρωματικό υλικό μπορεί να βρεθεί στις θέσεις:

<https://eclipse.org/articles/viewArticle/ViewArticle2.html> και

<https://eclipse.org/articles/using-perspectives/PerspectiveArticle.html>

#### Στόχοι

Οι μαθητές μπορούν να:

- επιλέγουν περιβάλλον ανάπτυξης εφαρμογής με βάση τις ανάγκες και τη διαθέσιμη υποδομή
- αναγνωρίζουν τα βασικά στοιχεία του περιβάλλοντος ανάπτυξης που θα χρησιμοποιήσουν
- μπορούν να πλοηγηθούν σε αυτό
- δημιουργήσουν ένα έργο στο περιβάλλον αυτό

### 2 Εισαγωγή στις Android Εφαρμογές (Apps)

Στην πρώτη διαδρομή, όπου το κεφάλαιο διδάσκεται με 1Θ + 4Ε, έμφαση στη θεωρία δίνεται στην ανάγνωση και αναγνώριση των πληροφοριών σε ένα δηλωτικό έργου, ενώ το εργαστήριο οργανώνεται κλιμακωτά: από την επίδειξη έτοιμου κώδικα, από εφαρμογή κάποιας πολυπλοκότητας, στην

αναζήτηση από τους μαθητές σε αυτή των τμημάτων που διαχειρίζονται είσοδο και έξοδο δεδομένων και τελικά στην τροποποίηση κώδικα Java με σκοπό τη βελτίωση λειτουργικότητας της έτοιμης εφαρμογής.

Στην δεύτερη διαδρομή, όπου το κεφάλαιο διδάσκεται με 1Θ, έμφαση δίνεται στην επίδειξη έτοιμου κώδικα από εφαρμογή κάποιας πολυπλοκότητας και ανάλυσή του.

### Στόχοι

Οι μαθητές να μπορούν να:

- αναγνωρίζουν με ποιο τρόπο περιγράφεται μια εφαρμογή και το φάκελό του
- αναγνωρίζουν την αναγκαιότητα για ποικιλία πόρων ώστε να δημιουργηθεί μια θελκτική εφαρμογή
- αναφέρουν τις βασικές κατηγορίες πόρων για Android εφαρμογή
- αναγνωρίζουν τη σκοπιμότητα χρήσης διαφορετικών γλωσσών (με έμφαση στην Java και την XML) σε τμήματα του έργου.
- εντοπίζουν μέσα σε ένα έργο τον πηγαίο κώδικα
- διαβάζουν το δηλωτικό έργο
- αναγνωρίζουν την υποκείμενη γλώσσα προγραμματισμού σε διάφορα τμήματα κώδικα

## 3 Εισαγωγή στη Σχεδίαση διεπαφής Χρήστη και αλληλεπίδρασή της με το χρήστη

Το κεφάλαιο αυτό εισάγει στη σχεδίαση διεπαφών χρήστη εξοικειώνοντας τους μαθητές με τα δομικά τους στοιχεία (γραφικά και σχεδιαστικά) και τον τρόπο που αλληλεπιδρούν με το χρήστη και την εφαρμογή. Έμφαση δίνεται στην εξοικείωση με τη γλώσσα XML και τα στοιχεία της.

Στην πρώτη διαδρομή, όπου το κεφάλαιο καταλ διδάσκεται με 2Θ+8Ε, στη θεωρία διαπραγματεύονται οι βασικές εντολές XML και ο τρόπος με τον οποίο μια Android εφαρμογή ανταποκρίνεται σε γεγονότα. Στο εργαστήριο οι μαθητές τροποποιούν αρχικά με καθοδήγηση από τον εκπαιδευτικό και αργότερα αυτόνομα κώδικα XML.

Στην δεύτερη διαδρομή, όπου το κεφάλαιο διδάσκεται με 2Θ, η θεωρία παραμένει όπως στην πρώτη διαδρομή και παραλείπεται το εργαστηριακό μέρος.

Περισσότερα στοιχεία σχετικά με τους listeners που εντοπίζουν και διαχειρίζονται τα γεγονότα μπορούν να βρεθούν στη θέση :

<http://developer.android.com/guide/topics/ui/ui-events.html#EventListeners> .

### Στόχοι

Οι μαθητές να μπορούν να:

- αναγνωρίζουν τα βήματα σχεδίασης μιας διεπαφής χρήσης
- αναγνωρίζουν τα στοιχεία της διεπαφής χρήστη
- δημιουργούν και εισάγουν οπτικά στοιχεία στο σχεδιασμό της εφαρμογής τους
- επιλέγουν, δημιουργούν και εισάγουν γραφικά στοιχεία στο σχεδιασμό της εφαρμογής τους
- αναγνωρίζουν τα βασικά στοιχεία της γλώσσας XML
- τροποποιούν κώδικα XML για τις ανάγκες της εφαρμογής τους
- αναγνωρίζουν τις ενέργειες του χρήστη – και αν είναι εφικτό τις εισόδους από αισθητήρες – ως γεγονότα και ενεργοποιητές μεθόδων

## 4 Προγραμματίζοντας σε Java για το Android

Ανάλογα από το πώς έχει εξελιχθεί το πρώτο μέρος (Java) μπορεί είτε να παραληφθεί είτε να γίνουν περισσότερες εργαστηριακές ώρες. Η προτεινόμενη δομή (2Θ+3Ε) στην πρώτη διαδρομή έχει τη λογική της περιληπτικής παρουσίασης του αντικειμενοστραφούς και οδηγούμενου από τα γεγονότα προγραμματισμού με Java.

Στην δεύτερη διαδρομή, το κεφάλαιο παραλείπεται.

### Στόχοι

Οι μαθητές να μπορούν να:

- οργανώνουν την εφαρμογή τους σε κλάσεις και αντικείμενα που επηρεάζονται από γεγονότα
- επεκτείνουν ήδη υπάρχουσες κλάσεις
- αναγνωρίζουν και απαριθμούν τα κύρια γεγονότα που σχετίζονται με την εφαρμογή τους
- αναλύουν την εφαρμογή τους σαν ένα σύνολο μεθόδων κάθε μια από τις οποίες ενεργοποιείται από ένα γεγονός
- σχεδιάζουν κατάλληλες μεθόδους για την διαχείριση αυτών των γεγονότων
- εφαρμόζουν στην εφαρμογή τους την επαναχρησιμοποίηση κώδικα μέσα από την κληρονομικότητα
- δημιουργούν τα δικά τους πακέτα ως βιβλιοθήκες λογισμικού

## 5 Πόροι εφαρμογής (App)

Στην πρώτη διαδρομή, όπου το κεφάλαιο διδάσκεται με 1Θ + 3Ε, στη θεωρία εισάγεται η δόμηση μιας διεπαφής από πόρους, ενώ στο εργαστήριο ξεκινάμε από την εισαγωγή έτοιμων πόρων στην εφαρμογή των μαθητών, προσαρμόζοντας όσα ανέπτυξαν στην προηγούμενη ενότητα ή αναπτύσσοντάς τα εξ αρχής, αν αυτή έχει παραληφθεί. Προτείνεται ιδιαίτερα η αξιοποίηση του σχετικού Παραρτήματος Β (Υποστηριζόμενη προσδιορισμοί) για περαιτέρω διερεύνηση.

Στην δεύτερη διαδρομή, το κεφάλαιο παραλείπεται.

### Στόχοι

Οι μαθητές να μπορούν να:

- αναγνωρίζουν βασικές κατηγορίες πόρων μιας εφαρμογής
- σχεδιάζουν και βελτιστοποιούν μια διεπαφή χρήστη με χρήση αντικειμένων
- σχεδιάζουν την ανταπόκριση σε γεγονότα με μεθόδους των κατάλληλων αντικειμένων
- υλοποιούν κατάλληλες μεθόδους για την διαχείριση αυτών των γεγονότων
- διαμορφώνουν κατάλληλα πλαίσια διαλόγου για επικοινωνία με τον χρήστη
- υλοποιούν μεθόδους για τον χειρισμό γεγονότων (του ποντικιού και του πληκτρολογίου τουλάχιστον)

## 6 Δηλωτικό Έργου

Στην πρώτη διαδρομή, όπου το κεφάλαιο διδάσκεται με 1Θ +2Ε η θεωρία εστιάζει στην παρουσίαση του δηλωτικού έτοιμου έργου και επεξήγηση των εννοιών του. Το εργαστήριο συνίσταται στην ανάπτυξη του δηλωτικού της εφαρμογής τους με τροποποίηση έτοιμου κώδικα και την εισαγωγή προς χρήση στοιχείων διαχείρισης του λειτουργικού, τμημάτων της εφαρμογής, προσβάσεων κλπ.

Στη δεύτερη διαδρομή, όπου το κεφάλαιο περιορίζεται σε (1Θ), έμφαση δίνεται στην παρουσίαση του δηλωτικού έτοιμου έργου και επεξήγηση των εννοιών του.

### Στόχοι

Οι μαθητές να μπορούν να:

- αναγνωρίζουν την ανάγκη επιλογών χειρισμού και την σύνδεσή τους με τις προδιαγραφές της εφαρμογής τους
- αναγνωρίζουν την αναγκαιότητα ενσωμάτωσης των επιλογών αυτών σε μια αυτόνομη εφαρμογή και το ρόλο του δηλωτικού στη διαδικασία αυτή στην περίπτωση Android εφαρμογής
- περιγράφουν την διαχείριση της πολυμορφικότητας σε σχέση με την εκάστοτε χρήση
- αναγνωρίζουν συνήθη στοιχεία ενός δηλωτικού
- προσαρμόζουν έτοιμες εγγραφές στις ανάγκες των έργων τους

## 7 Δεδομένα εφαρμογής

Το θεωρητικό μέρος συνίσταται σε παρουσίαση των διαθέσιμων τύπων δεδομένων και παρουσίαση του τρόπου αξιοποίησης μιας βάσης δεδομένων για διατήρηση δεδομένων μιας εφαρμογής.

Στην πρώτη διαδρομή, όπου το κεφάλαιο εκτείνεται σε (1Θ + 5Ε) οι εργαστηριακές ώρες αφορούν σε σχεδιασμό και υλοποίηση διαχείρισης ασφάλειας δεδομένων μέσω επιλογών τύπου, σχεδιασμό και δημιουργία απλής βάσης (με SQLite), διαχείριση βάσης δεδομένων από την εφαρμογή – population βάσης, υλοποίηση εισαγωγής διαδικτυακών δεδομένων στην εφαρμογή τους, με τροποποίηση υπάρχοντος κώδικα και υλοποίηση προσβάσεων (Permissions).

Η λογική εδώ είναι ανακαλυπτική. Για παράδειγμα, κατά την διαπραγμάτευση των διαθέσιμων τύπων δεδομένων αναμένεται να προκύψουν ερωτήματα σχετικά με τη διαχείρισή τους και τη σύνδεσή τους με τη διεπαφή. Η σχετική διερεύνηση από τους μαθητές θα οδηγήσει εύκολα στον εντοπισμό της σχετικής λειτουργικότητας η οποία μπορεί να αξιοποιηθεί αργότερα όταν δεδομένα θα διαχειρίζονται μέσω βάσης.

Περισσότερα σχετικά με την αξιοποίηση βάσεων δεδομένων μπορούν να βρεθούν στις θέσεις:

<http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html>

<http://developer.android.com/reference/android/database/Cursor.html>

### Στόχοι

Οι μαθητές να μπορούν να:

- περιγράφουν τον τρόπο διαχείρισης των συνηθών τύπων δεδομένων
- διακρίνουν τη διαφορά μεταξύ Private Internal files, Public External Files
- σχεδιάζουν τη διαχείριση βάσης δεδομένων μέσω εφαρμογής
- χρησιμοποιούν τους τύπους δεδομένων για να εξασφαλίζουν την ασφάλειά τους

## 8 Εκτέλεση και εκσφαλμάτωση

Το θεωρητικό μέρος συνίσταται σε παρουσίαση των τύπων συσκευών, παρουσίαση του σχήματος εκσφαλμάτωσης αλλά και της δομής ενός test project. Στη δεύτερη διαδρομή, όπου το κεφάλαιο περιορίζεται σε (1Θ), αυτά αποτελούν το συνολικό αντικείμενο του κεφαλαίου.

Στην πρώτη διαδρομή, όπου το κεφάλαιο εκτείνεται σε (1Θ + 2Ε), οι εργαστηριακές ώρες προτείνεται να εστιάσουν στην επίδειξη λειτουργίας έτοιμου Test project, επίδειξη αξιοποίησης κυρίως εικονικών αλλά και φυσικών συσκευών κατά την εκσφαλμάτωση και εφόσον είναι δυνατό,

εισαγωγή/τροποποίηση από τους μαθητές κώδικα για υλοποίηση κατάλληλου χειρισμού συσκευής στην εφαρμογή τους.

Συμπληρωματικά στοιχεία για την ενότητα μπορούν να βρεθούν στις θέσεις:

<http://developer.android.com/tools/debugging/index.htm>

<http://developer.android.com/training/testing.html>

[http://developer.android.com/tools/testing/testing\\_android.html](http://developer.android.com/tools/testing/testing_android.html)

<http://developer.android.com/tools/help/emulator.html>

### **Στόχοι**

Οι μαθητές να μπορούν να:

- αναγνωρίζουν τη σπουδαιότητα των εικονικών συσκευών για Unix-type λειτουργικά
- περιγράφουν τον τρόπο που το Android αντιλαμβάνεται τις συσκευές
- αναγνωρίζουν τα πλεονεκτήματα της προσομοίωσης στην εκσφαλμάτωση
- αναγνωρίζουν την αναγκαιότητα δομών ελέγχου
- κατασκευάζουν και χρησιμοποιούν ένα test project
- περιγράφουν τον τρόπο που οι συσκευές χρησιμοποιούνται στην εκσφαλμάτωση
- χρησιμοποιούν εργαλεία ελέγχου
- εκτελούν τα βήματα εκσφαλμάτωσης
- εκτελούν την εφαρμογή τους

## **9 Κύκλος ζωής δραστηριότητας (Activity)**

Όμοιο και στις δύο διαδρομές το κεφάλαιο αυτό, που αντιστοιχεί σε (1Θ), αφορά στην εξοικείωση με τις μεθόδους ανάκλησης μιας εφαρμογής και τον συνδυασμό τους με τον κύκλο ζωής της. Προτείνεται η παρατήρηση της συμπεριφοράς έτοιμης εφαρμογής σε διάφορες καταστάσεις και αν αυτό είναι εφικτό, προσπάθεια πρόβλεψης από τους μαθητές της αντίστοιχης συμπεριφοράς.

### **Στόχοι**

Οι μαθητές να μπορούν να:

- περιγράφουν τις διαφορετικές καταστάσεις στις οποίες μπορεί να βρίσκεται μια δραστηριότητα
- αναγνωρίζουν την επίδραση αυτής της κατάστασης στη λειτουργία μια εφαρμογής
- χρησιμοποιούν μεθόδους ανάκλησης
- αναγνωρίζουν τον τρόπο συμπεριφοράς μιας εφαρμογής σε σχέση με την κατάσταση στην οποία βρίσκεται

## **10 Συνήθεις συνιστώσες (components) Android**

Στην πρώτη διαδρομή, όπου το κεφάλαιο εκτείνεται σε 3 αμιγώς εργαστηριακές ώρες (0Θ + 3Ε), παρουσιάζονται σε αυτές στην πράξη, οι υπηρεσίες και ο τρόπος λειτουργίας τους, οι πάροχοι περιεχομένου, ο τρόπος λειτουργίας των δεκτών εκπομπών αλλά δευτερευόντως και των λογικών τμημάτων και του Action Bar.

Στη δεύτερη διαδρομή, το κεφάλαιο περιορίζεται σε(1Θ), η οποία αφορά σε παρουσίαση των ανωτέρω, με έμφαση στη χρήση δεκτών εκπομπών.

Συμπληρωματικό υλικό για την ενότητα βρίσκεται στις θέσεις:

<http://developer.android.com/guide/topics/providers/content-provider-basics.html>

<http://developer.android.com/guide/topics/providers/content-provider-creating.html>



<http://developer.android.com/reference/android/content/BroadcastReceiver.html>

<http://developer.android.com/guide/components/fragments.html> και

<http://developer.android.com/guide/topics/ui/actionbar.html> .

### Στόχοι

Οι μαθητές να μπορούν να:

- χρησιμοποιούν εξωτερικές υπηρεσίες και περιεχόμενο
- διαχειρίζονται τα μηνύματα συστήματος και εφαρμογών
- χρησιμοποιούν έτοιμα λογικά τμήματα
- οργανώνουν την εφαρμογή τους σε επαναχρησιμοποιήσιμα λογικά τμήματα

## 11 Χρησιμοποιώντας δείγματα (samples)

<http://www.java2s.com/Code/Android/CatalogAndroid.htm> και

<https://code.google.com/p/apps-for-android/source/browse/#git> .

### Στόχοι

Οι μαθητές να μπορούν να:

- γνωρίσουν οι μαθητές την ύπαρξη των samples
- αναγνωρίζουν οι μαθητές τη χρησιμότητά τους για την κατανόηση επιμέρους θεμάτων προγραμματισμού για Android
- χρησιμοποιούν τα samples για την επέκταση της κατανόησής τους

## 12 Δημοσίευση και κυκλοφορία έργου

Στην πρώτη διαδρομή, (0Θ + 1Ε), το κεφάλαιο αντιμετωπίζεται εργαστηριακά, ως μια ευκαιρία δημοσίευσης της εφαρμογής τους.

Στη δεύτερη, (1Θ), γίνεται συζήτηση στην τάξη για την αναγκαιότητα πολλαπλών εκδόσεων αλλά και για θέματα πνευματικής ιδιοκτησίας.

Συμπληρωματικό υλικό στις θέσεις:

<https://developer.android.com/tools/publishing/preparing.html>

<http://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html>

<http://developer.android.com/distribute/tools/launch-checklist.html>

Επίσης για το ζήτημα των δικαιωμάτων χρήσης και του ανοιχτού λογισμικού στις παρακάτω θέσεις:

[https://foss.ntua.gr/wiki/index.php/Άδειες\\_Ελεύθερου\\_Λογισμικού\\_και\\_Πατέντες](https://foss.ntua.gr/wiki/index.php/Άδειες_Ελεύθερου_Λογισμικού_και_Πατέντες)

<http://www.eexi.gr/?q=node/16> και <http://www.eexi.gr/?q=node/25> στα ελληνικά.

<http://www.gnu.org/licenses> στα αγγλικά.

### Στόχοι

Οι μαθητές να μπορούν να:

- διαχωρίζουν μεταξύ μιας έκδοσης ανάπτυξης και μιας έκδοσης διανομής
- περιγράφουν την προετοιμασία της δημοσίευσης

- αναγνωρίζουν την ανάγκη πολλαπλών εκδόσεων
- περιγράφουν και υλοποιούν τα βήματα δημοσίευσης μιας εφαρμογής

## ΥΠΟΕΝΟΤΗΤΑ 2B

Η υποενότητα 2B εστιάζει και στις δύο διαδρομές στην ανάπτυξη μιας αρθρωτής εφαρμογής για Android χρησιμοποιώντας το περιβάλλον προγραμματισμού ψηφίδων AppInventor 2.

Στην πρώτη διαδρομή η ενότητα αναπτύσσεται σε (1Θ +13Ε) όπου η ώρα της θεωρίας αφορά στην παρουσίαση των βασικών στοιχείων μιας βάσης αλλά και του σχεδιασμού απλών βάσεων (κεφάλαιο 19). Οι εργαστηριακές ώρες αφορούν στην υλοποίηση μιας εφαρμογής από τον υπάρχοντα στις σημειώσεις κώδικα. Λόγω των περιορισμών χρόνου θα είναι ίσως σκόπιμη η προηγούμενη αποθήκευση τμημάτων του κώδικα στους υπολογιστές τους ή και η διάσχιση ενοτήτων με απλή επίδειξη κατά την κρίση του εκπαιδευτικού.

Στη δεύτερη διαδρομή (2Θ + 47Ε) προστίθεται στη θεωρία, εκτός του κεφαλαίου 19, μία ώρα με θέμα τη διαχείριση των αισθητήρων ως παραγωγών γεγονότων (κεφάλαιο 17). Σε αυτή την περίπτωση δεν υπάρχει λόγος βιασύνης ενώ μπορούν και πρέπει να διερευνηθούν επεκτάσεις της υπάρχουσας εφαρμογής.

Υλικό για το AppInventor 2 μπορεί κανείς να βρει στα ελληνικά στις πηγές για την εβδομάδα κώδικα: <http://codeweek.eu/resources/greece/>

Μια σχετική δουλειά έχει γίνει από το σύλλογο εκπαιδευτικών Πληροφορικής Χίου και διατίθεται στη θέση <http://sepchiou.gr/index.php/yliko/86-appinvcodeclub>, όπου μπορεί να βρει κανείς και σημειώσεις: [http://sepchiou.gr/docs/appinventor\\_code\\_club/AppInventorProgramming.pdf](http://sepchiou.gr/docs/appinventor_code_club/AppInventorProgramming.pdf)

Αντίστοιχα, ανάμεσα στα άλλα σεμινάρια που αναπτύχθηκαν για το β' επίπεδο της επιμόρφωσης εκπαιδευτικών για την εφαρμογή στην τάξη αναπτύχθηκαν κάποια και για AppInventor 2:

[http://epimorfwsh-.wikispaces.com/31\)+%CE%A3%CE%B5%CE%BD%CE%AC%CF%81%CE%B9%CE%B1+-+AppInventor](http://epimorfwsh-.wikispaces.com/31)+%CE%A3%CE%B5%CE%BD%CE%AC%CF%81%CE%B9%CE%B1+-+AppInventor)

Πλούσιο υλικό, περιλαμβανομένων και τμημάτων της εφαρμογής των σημειώσεων του μαθήματος μπορείτε να βρείτε και στην Κοινότητα Εκπαιδευτικής Ανάπτυξης με AppInventor:

<https://plus.google.com/u/0/communities/113682665764045653468> όπου όμως χρειάζεται εγγραφή, χωρίς βέβαια επιβάρυνση.

Φυσικά υλικό θα αναρτηθεί και στο wiki του μαθήματος.

Η στοχοθεσία και η προβλεπόμενη διάρκεια σε ώρες των κεφαλαίων της ενότητας 2B παρατίθεται στη συνέχεια για λόγους πληρότητας

### **13 Το προγραμματιστικό περιβάλλον AppInventor2**

Στην πρώτη διαδρομή καλύπτει (1Ε) και στη δεύτερη (4Ε) ώρες.

Για όσους δεν παρακολούθησαν το μάθημα «Εφαρμογές Η/Υ» στην Α΄ ΕΠΑΛ μαθητές Γ΄τάξης (2015-16 και 2016-17) είναι σκόπιμο να δοθεί μεγαλύτερο βάρος στην ενότητα.

#### **Στόχοι**

Να θυμηθεί ο μαθητής όσα έμαθε στην Α΄ τάξη σχετικά με το προγραμματιστικό περιβάλλον AppInventor2

### **14 Σχεδίαση Διεπαφής χρήστη και γεγονότα**

Στην πρώτη διαδρομή καλύπτει Κείμενο (1Ε) και στη δεύτερη (6Ε) ώρες.

#### **Στόχοι**

Οι μαθητές να μπορούν να:

- αναγνωρίζουν τα βήματα σχεδίασης μιας διεπαφής χρήσης
- αναγνωρίζουν τα στοιχεία της διεπαφής χρήστη
- δημιουργούν και εισάγουν οπτικά στοιχεία στο σχεδιασμό της εφαρμογής τους
- επιλέγουν, δημιουργούν και εισάγουν γραφικά στοιχεία στο σχεδιασμό της εφαρμογής τους
- σχεδιάζουν και υλοποιούν τη διεπαφή της εφαρμογής τους με βάση υπάρχουσα (έτοιμη) σχεδίαση
- χρησιμοποιούν menu
- χρησιμοποιούν τα sound και Player για την εισαγωγή ηχητικής διάστασης στις εφαρμογές τους

### **15 Αντικείμενα στο περιβάλλον AppInventor2**

Στην πρώτη διαδρομή καλύπτει (4Ε) και στη δεύτερη (8Ε) ώρες.

Για όσους δεν παρακολούθησαν το μάθημα «Εφαρμογές Η/Υ» στην Α΄ ΕΠΑΛ μαθητές Γ΄τάξης (2015-16 και 2016-17) είναι σκόπιμο να δοθεί μεγαλύτερο βάρος και σε αυτήν την ενότητα.

#### **Στόχοι**

Οι μαθητές να μπορούν να:

- Να διαχωρίζουν τις βασικές έννοιες του αντικειμενοστρεφούς όπως υλοποιούνται στο περιβάλλον από τις αντίστοιχες της θεωρίας
- χρησιμοποιούν τις ψηφίδες για να υλοποιούν μεθόδους σε αντικείμενα
- αναλύουν την εφαρμογή τους σαν ένα σύνολο μεθόδων κάθε μια από τις οποίες ενεργοποιείται από ένα γεγονός
- οργανώνουν και περιγράφουν την εφαρμογή τους με αντικείμενα και γεγονότα
- σχεδιάζουν κατάλληλες μεθόδους για την διαχείριση αυτών των γεγονότων
- εφαρμόζουν στην εφαρμογή τους την επαναχρησιμοποίηση κώδικα μέσα από την κληρονομικότητα

### **16 Αξιοποίηση υπάρχοντος κώδικα**

Στην πρώτη διαδρομή καλύπτει (1Ε) και στη δεύτερη (5Ε) ώρες.

#### **Στόχοι**

Οι μαθητές να μπορούν να:

- εξοικειωθούν με τη λογική του ανοιχτού λογισμικού
- αναγνωρίζουν την πληθώρα και τη χρησιμότητα του υπάρχοντος κώδικα
- εξοικειωθούν με την ιδέα της επαναχρησιμοποίησης κώδικα
- γνωρίζουν πηγές ειδικά για το περιβάλλον που χρησιμοποιούν

## 17 Διαχείριση αισθητήρων και άλλων στοιχείων κινητού

Στην πρώτη διαδρομή καλύπτει (1Ε) και στη δεύτερη (1Θ+5Ε) ώρες.

### Στόχοι

Οι μαθητές να μπορούν να:

- αναγνωρίζουν συσκευές ως πιθανά αντικείμενα της εφαρμογής τους
- αναγνωρίζουν τους διαθέσιμους αισθητήρες, συσκευές και υπηρεσίες του κινητού τους
- υλοποιούν τη διαχείριση αισθητήρων, συσκευών και υπηρεσιών
- σχεδιάζουν κατάλληλες μεθόδους για την αξιοποίηση αισθητήρων και συσκευών

## 18 Εξωτερική Επικοινωνία

Στην πρώτη διαδρομή καλύπτει (1Ε) και στη δεύτερη (6Ε) ώρες.

### Στόχοι

Οι μαθητές να μπορούν να:

- αντιλαμβάνονται το διαδίκτυο ως πηγή δεδομένων και μέσο επικοινωνίας
- σχεδιάζουν ανοιχτές εφαρμογές με δυνατότητα επικοινωνίας
- συνδυάζουν υπάρχουσες υπηρεσίες με την εφαρμογή τους
- ενσωματώνουν υπηρεσίες στην εφαρμογή τους
- περιγράφουν το μοντέλο πελάτη εξυπηρετητή
- χρησιμοποιούν το μοντέλο πελάτη εξυπηρετητή στις υλοποιήσεις τους.

## 19 Οργάνωση και διαχείριση δεδομένων

Στην πρώτη διαδρομή καλύπτει (1Θ+3Ε) και στη δεύτερη (1Θ+7Ε) ώρες.

### Στόχοι

Οι μαθητές να μπορούν να:

- διαχειρίζονται και οργανώνουν εσωτερικά δεδομένα των εφαρμογών τους.
- σχεδιάζουν τη διαχείριση βάσης δεδομένων μέσω εφαρμογής
- ολοκληρώνουν τις εφαρμογές τους προσθέτοντας το στοιχείο της ιστορικότητας με τη διατήρηση δεδομένων

## 20 Δημοσίευση μιας εφαρμογής

Στην πρώτη διαδρομή καλύπτει (1Ε) και στη δεύτερη (6Ε) ώρες.

### Στόχοι

Οι μαθητές να μπορούν να:

- διαχωρίζουν μεταξύ πρωτότυπων και μη εφαρμογών
- εξοικειωθούν με τις προϋποθέσεις για δημοσίευση εφαρμογής με διατήρηση δικαιωμάτων
- αναγνωρίζουν τις διαφορές μεταξύ ερασιτεχνικής και επαγγελματικής ανάπτυξης
- αναγνωρίζουν τις διαφορετικές απαιτήσεις μεταξύ εκπαιδευτικών και επαγγελματικών εφαρμογών