



**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.**

**«Εκμάθηση Μαθηματικών Σε παιδιά Δημοτικού Με  
Εφαρμογή Android»**

**Πτυχιακή Εργασία των**

**Ειρήνη Φυρού Α.Μ: 15015**

**Γεώργιος Βιδάλης Α.Μ: 14233**

**Επιβλέπουσα Καθηγήτρια: Λαμπρινή Σερεμέτη**

**ΑΘΗΝΑ, ΟΚΤΩΒΡΙΟΣ 2018**

## ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία αναλύονται και περιγράφονται τεχνικές ανάπτυξης εφαρμογών για “smart phones” καθώς επίσης υλοποιείται μια εφαρμογή σε λειτουργικό Android.

Στην εισαγωγή γίνεται αναφορά στον όλο ένα αυξανόμενο ρυθμό δημιουργίας νέων εφαρμογών σε διάφορα λειτουργικά συστήματα που αφορούν έξυπνες συσκευές καθώς επίσης και τον λόγο που προβήκαμε στην επιλογή ανάπτυξης μιας εφαρμογής σε λειτουργικό Android.

Στο δεύτερο κεφάλαιο γίνεται μια παρουσίαση του λειτουργικού συστήματος Android στην οποία περιλαμβάνονται πληροφορίες σχετικά με τη φύση του και την εξέλιξη του από την στιγμή που εμφανίστηκε στο πεδίο της τεχνολογίας έως και σήμερα. Στο ίδιο κεφάλαιο γίνεται μια σύγκριση αυτού με άλλα λειτουργικά συστήματα της αγοράς καθώς επίσης παρουσιάζονται και τα επίπεδα της αρχιτεκτονικής του.

Στο επόμενο κεφάλαιο γίνεται μια περιγραφή των κινητών εφαρμογών (mobile applications) και στην συνέχεια μια περιγραφή για το πώς αναπτύσσουμε μια εφαρμογή σε Android.

Αμέσως μετά και συγκεκριμένα στο κεφάλαιο πέντε δίνονται πληροφορίες σχετικά με την διδακτική των μαθηματικών στις τάξεις του δημοτικού όπου αναλύονται οι παιδαγωγικές αρχές της διδασκαλίας, κάποιες σύγχρονες αρχές και ενέργειες καθώς και τα συστατικά που μπορεί να χρησιμοποιήσει ένας δάσκαλος προκειμένου να βελτιστοποιήσει την διδασκαλία του μαθήματος.

Στην συνέχεια, στο κεφάλαιο έξι παρουσιάζονται τα βασικά στοιχεία που περιλαμβάνει μια εφαρμογή Android τα οποία χρησιμοποιούνται στην ανάπτυξη της εφαρμογής που υλοποιήθηκε η οποία περιγράφεται στο κεφάλαιο επτά μαζί με τις βασικές τεχνικές που αποτέλεσαν τη βάση της ανάπτυξης της. Η κεντρική ιδέα της εφαρμογής είναι η εκμάθηση μαθηματικών για μαθητές δημοτικού μέσα από ένα παιχνίδι που θα τους δίνει την δυνατότητα να μάθουν με ένα διασκεδαστικό τρόπο.

## **ABSTRACT**

In this diploma thesis we analyze and describe application development techniques for smart phones as well as an application on Android operating system.

In the introduction we refer to the increasing pace of new applications creation in a variety of operating systems for smart devices as well as the reason that we have chosen Android as the operation system of our application

The second chapter introduces a presentation of the Android operating system that includes information about its nature and its evolution from the moment it appeared in the field of technology up to date of. In the same chapter we compare Android system with other operation systems as well as the levels of its architecture.

In the following chapter we give a description of mobile applications and then a description of how to develop an Android application.

In Chapter five we give information about the teaching of mathematics in the Primary school where the pedagogical principles of teaching are analyzed. Also we give information about modern principles and actions as well as the means that a teacher can use in order to optimize the teaching of mathematics.

Then, chapter six shows the basics components of an Android application that is used to develop the application that has been crafted. Our application is described in chapter seven along with the basic techniques that we based on for its development. The main idea of the application is to learn mathematics for elementary students through a game that will enable them to learn in a fun way.

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα θέλαμε να ευχαριστήσουμε θερμά την καθηγήτρια μας Λαμπρινή Σερεμέτη που μας εμπιστεύτηκε το θέμα της πτυχιακής μας εργασίας δίνοντας μας τη δυνατότητα να ασχοληθούμε με ένα άκρως ενδιαφέρον και μοντέρνο θέμα, η ενασχόληση με το οποίο μας παρείχε ένα ακόμη λιθαράκι γνώσης καθώς φτάνουμε στο τέλος της φοιτητικής μας διαδρομής. Επιπρόσθετα, οφείλουμε να αφιερώσουμε την πτυχιακή μας εργασία στους γονείς μας που μας συμπαρυστάθηκαν όλα τα χρόνια της φοίτησης μας στο Τ.Ε.Ι Δυτικής Ελλάδος.

## ΠΕΡΙΕΧΟΜΕΝΑ

1.	ΕΙΣΑΓΩΓΗ .....	10
2.	ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ANDROID.....	11
2.1	Τι είναι Android; .....	11
2.2	Εξέλιξη του Android .....	11
2.3	Σύγκριση με άλλα λογισμικά .....	14
2.4	Επίπεδα Αρχιτεκτονικής.....	17
3.	ΤΙ ΕΙΝΑΙ ΤΑ MOBILE APPLICATIONS .....	20
➤	"Mobile - First" Ανάπτυξη .....	21
4.	ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ANDROID.....	22
5.	Η ΔΙΔΑΚΤΙΚΗ ΤΩΝ ΜΑΘΗΜΑΤΙΚΩΝ.....	24
5.1	Παιδαγωγικές αρχές για την διδασκαλία των μαθηματικών .....	24
5.2	Σύγχρονες αρχές της διδασκαλίας των μαθηματικών .....	25
5.3	Δασκαλικές ενέργειες για την υποβοήθηση της ανακάλυψης στα μαθηματικά....	25
5.4	Τα Συστατικά μιας επιτυχημένης διδασκαλίας στα μαθηματικά.....	27
6.	ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΜΙΑ ΕΦΑΡΜΟΓΗΣ ANDROID .....	29
6.1	Δραστηριότητες .....	29
6.2	Υπηρεσίες.....	29
6.3	Δέκτες εκπομπής .....	30
6.4	Παροχές Περιεχομένου .....	31
7.	ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ.....	33
7.1	Αρχές και τεχνικές αντικειμενοστραφή προγραμματισμού .....	33
7.2	Σχεδιασμός του μοντέλου MVC.....	34
7.3	Βασικά εργαλεία ανάπτυξης .....	35
7.4	Η Γλώσσα Προγραμματισμού Java .....	38

7.5 Παρουσίαση Εφαρμογής «Λύσε Το» .....	42
7.6 Ανάλυση Τμημάτων Πηγαίου Κώδικα .....	50
7.7 Μελλοντική Ανάπτυξη .....	70

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 2.1: Οι εκδόσεις του Android.....	15
Πίνακας 7.6.1: Κώδικας Μοντέλου (Μέρος Α) .....	50
Πίνακας 7.6.2: Κώδικας Μοντέλου (Μέρος Β) .....	54
Πίνακας 7.6.3: Κώδικας Μοντέλου (Μέρος Γ).....	56
Πίνακας 7.6.4: Κώδικας του Layout της κύριας οθόνης (μέρος Α) .....	57
Πίνακας 7.6.5: Κώδικας του Layout της κύριας οθόνης (μέρος Β) .....	58
Πίνακας 7.6.6: Κώδικας του Layout της κύριας οθόνης (μέρος Γ) .....	59
Πίνακας 7.6.7: Κώδικας του Layout της κύριας οθόνης (μέρος Δ) .....	59
Πίνακας 7.6.8: Κώδικας του Layout της κύριας οθόνης (μέρος Ε) .....	61
Πίνακας 7.6.9: Κώδικας του Layout της κύριας οθόνης (μέρος ΣΤ) .....	64

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1: Το σήμα κατατεθέν του Android.....	13
Εικόνα 2.2: Το λογότυπο του Android .....	14
Εικόνα 2.3: Το διάγραμμα της αρχιτεκτονικής του Android .....	18
Εικόνα 7.1.1: Το μοντέλο MVC.....	33
Εικόνα 7.2.1: Το UML διάγραμμα του μοντέλου της εφαρμογής .....	34
Εικόνα 7.3.1 Το λογότυπο του Android Studio .....	35
Εικόνα 7.3.2 Το περιβάλλον του Android Studio .....	36
Εικόνα 7.5.1: Το λογότυπο .....	41
Εικόνα 7.5.2: Κεντρικό Μενού .....	42
Εικόνα 7.5.3: Ρυθμίσεις.....	43
Εικόνα 7.5.4: Αρχική κατάσταση του παιχνιδιού .....	44
Εικόνα 7.5.5: Επιλογή απάντησης .....	45
Εικόνα 7.5.6 Η εμφάνιση της εφαρμογής μετά από σωστή απάντηση .....	46
Εικόνα 7.5.7 Η εμφάνιση της εφαρμογής μετά από λάθος απάντηση .....	46
Εικόνα 7.5.8: Οθόνη με μήνυμα επιβράβευσης προς τον χρήστη .....	47
Εικόνα 7.5.9 Οθόνη με μήνυμα παρακίνησης προς τον χρήστη .....	48
Εικόνα 7.5.10: Το παιχνίδι στο δύσκολο επίπεδο .....	49



## **ΠΡΟΛΟΓΟΣ**

Η παρούσα πτυχιακή εργασία εκπονήθηκε στο πλαίσιο της ολοκλήρωσης των σπουδών μας στο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Ελλάδας της σχολής Διοίκησης και Οικονομίας του τμήματος Εφαρμογών Πληροφορικής στην Διοίκηση και την Οικονομία. Αποτελεί μια εργασία με υλοποίηση μιας εφαρμογής λειτουργικού συστήματος Android η οποία συμπληρώνεται με ένα βασικό θεωρητικό μέρος γύρω από το συγκεκριμένο λειτουργικό σύστημα καθώς και γενικότερα τις εφαρμογές.

## 1. ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια έχει παρατηρηθεί μια ευρεία ανάπτυξη στο κλάδο των έξυπνων συσκευών και ιδιαίτερα σε εκείνον της κινητής τηλεφωνίας. Ολο ένα και μεγαλύτερος ρυθμός αύξησης στην έκδοση νέων εφαρμογών που αφορούν τα κινητά τηλέφωνα αποτελεί ένα ενδιαφέρον αντικείμενο για τους λάτρεις της τεχνολογίας. Η ποικιλία τέτοιων εφαρμογών είναι πλούσια δίνοντας την δυνατότητα στους χρήστες να επιλέγουν σύμφωνα με τις ανάγκες τους. Θέλοντας και εμείς να ακολουθήσουμε αυτή την εξέλιξη επιλέξαμε να δημιουργήσουμε μια τέτοια διαδραστική εφαρμογή ώστε να κερδίσουμε όσο το δυνατόν περισσότερη γνώση γύρω από αυτό το τεχνολογικό πεδίο. Ανάμεσα στα διάφορα λειτουργικά συστήματα που υπάρχουν στην αγορά επιλέξαμε εκείνο του Android μιας και θεωρούμε ότι είναι ιδιαίτερα διαδεδομένο καθώς επίσης και το λειτουργικό που και εμείς οι ίδιοι χρησιμοποιούμε. Η φύση της εφαρμογής μας επιλέξαμε να είναι εκπαιδευτικού χαρακτήρα (εκμάθησης μαθηματικών) διότι θεωρούμε ότι η μεταλαμπάδευση γνώσης είναι ένας πολύ σημαντικός παράγοντας ώστε ο άνθρωπος να χρησιμοποιεί την ανάπτυξη της τεχνολογίας προς όφελος του. Έτσι δημιουργήθηκε η ιδέα της ανάπτυξης μια διαδραστικής εφαρμογής σε λειτουργικό σύστημα Android.

## **2. ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ANDROID**

### **2.1 Τι είναι Android;**

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να δημιουργούν τα δικά τους προγράμματα και εφαρμογές με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα (smart phones) και τα tablet. Πέρα αυτών τα τελευταία χρόνια έχει χρησιμοποιηθεί και για άλλες συσκευές με διαφορετικό περιβάλλον χρήσης όπως τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ (π.χ. το HP Slate 21) και σε άλλες ηλεκτρονικές συσκευές. Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της ApacheLicense, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου και σχεδιάστηκε από τη γραφίστρια Ιρίνα Μπλόκ. Η τελευταία έκδοση καλείται Android 8.1 Oreo και φέρνει σημαντικές αλλαγές.

### **2.2 Εξέλιξη του Android**

Το λειτουργικό σύστημα Android ιδρύθηκε στο Palo Alto της Καλιφόρνια τον Οκτώβριο του 2003 από τους Andy Rubin, Rich Miner, Nick Sears και Chris White. Οι πρώτες προθέσεις της εταιρείας ήταν να αναπτύξει ένα προηγμένο λειτουργικό σύστημα για ψηφιακές φωτογραφικές μηχανές και αυτή ήταν η βάση του πεδίου για τους επενδυτές τον Απρίλιο του 2004. Η εταιρεία αποφάσισε τότε ότι η αγορά των φωτογραφικών μηχανών δεν ήταν αρκετά μεγάλη για τους στόχους της, πέντε μήνες αργότερα είχε εκτρέψει τις προσπάθειές της και ανέβαζε το Android ως λειτουργικό

σύστημα χειρός που θα ανταγωνιζόταν τα Symbian και τα Microsoft Windows Mobile. Ο Ρούμπιν είχε δυσκολία να προσελκύσει επενδυτές από νωρίς και το Android αντιμετώπιζε έξωση από το χώρο γραφείων του. Τον Ιούλιο του 2005, η Google απέκτησε το Android Inc. για τουλάχιστον 50 εκατομμύρια δολάρια. Οι βασικοί υπάλληλοί της, όπως οι Rubin, Miner και White, εντάχθηκαν στο Google ως μέρος της εξαγοράς. Δεν γνώριζαν πολλά για το μυστικό Android την εποχή εκείνη, καθώς η εταιρεία παρείχε λίγες λεπτομέρειες εκτός από το ότι έκανε λογισμικό για κινητά τηλέφωνα. Στη Google, η ομάδα με επικεφαλής τον Rubin ανέπτυξε μια πλατφόρμα κινητών συσκευών που τροφοδοτείται από τον πυρήνα του Linux. Η Google προώθησε την πλατφόρμα στους κατασκευαστές και τους μεταφορείς με την υπόσχεση παροχής ενός ευέλικτου, αναβαθμίσιμου συστήματος. Μέχρι το 2008, τόσο η Nokia όσο και το BlackBerry ανακοίνωσαν έξυπνα τηλέφωνα με βάση το touch για να ανταγωνιστεί το iPhone 3G και η εστίαση του Android τελικά μετατράπηκε σε απλές οθόνες αφής. Το πρώτο εμπορικά διαθέσιμο smartphone που τρέχει το Android ήταν το HTC Dream, επίσης γνωστό ως T-Mobile G1, που ανακοινώθηκε στις 23 Σεπτεμβρίου 2008.

Στις 5 Νοεμβρίου 2007, η Open Handset Alliance, μια κοινοπραξία εταιρειών τεχνολογίας όπως η Google, κατασκευαστές συσκευών όπως HTC, Motorola και Samsung, ασύρματες εταιρείες όπως η Sprint και η T-Mobile και κατασκευαστές chipset όπως η Qualcomm και η Texas Instruments με στόχο την ανάπτυξη της πρώτης πραγματικά ανοικτής και ολοκληρωμένης πλατφόρμας για κινητές συσκευές. Μέσα σε ένα χρόνο η Open Handset Alliance αντιμετώπισε δύο άλλους ανταγωνιστές ανοικτής πηγής, το Symbian Foundation και το LiMo Foundation, το τελευταίο αναπτύσσοντας επίσης ένα κινητό λειτουργικό σύστημα όπως το Google. Τον Σεπτέμβριο του 2007, το Information Week κάλυψε μια μελέτη Evaluate serve που ανέφερε ότι η Google είχε καταθέσει αρκετές αιτήσεις ευρεσιτεχνίας στον τομέα της κινητής τηλεφωνίας.



**Εικόνα 2.10: Το σήμα κατατεθέν του Android**

Από το 2008, το Android έχει δει πολλές αναβαθμίσεις, οι οποίες έχουν βελτιώσει σταδιακά το λειτουργικό σύστημα, προσθέτοντας νέα χαρακτηριστικά και διορθώνοντας σφάλματα σε προηγούμενες κυκλοφορίες. Κάθε μεγάλη κυκλοφορία ονομάζεται με αλφαβητική σειρά μετά από επιδόρπιο ή ζαχαρούχο σκεύασμα, με τις πρώτες εκδόσεις του Android να ονομάζονται "Cupcake", "Donut", "Eclair" και "Froyo", με αυτή τη σειρά. Το 2010, η Google παρουσίασε τη σειρά συσκευών Nexus, μια σειρά στα οποία η Google συνεργάστηκε με διαφορετικούς κατασκευαστές συσκευών για την παραγωγή νέων συσκευών και την εισαγωγή νέων εκδόσεων Android. Η σειρά περιγράφηκε ότι έχει διαδραματίσει βασικό ρόλο στο ιστορικό του Android εισάγοντας νέες επαναλήψεις λογισμικού και πρότυπα υλικού σε όλο το πλάτος και έγινε γνωστός για το λογισμικό "free-bloat" με "έγκαιρες ενημερώσεις". Στη διάσκεψη προγραμματιστών τον Μάιο του 2013, η Google ανακοίνωσε μια ειδική έκδοση του Samsung Galaxy S4, όπου, αντί να χρησιμοποιήσει την προσαρμογή του Android, το τηλέφωνο έτρεξε "stock Android" και υποσχέθηκε να λάβει γρήγορα νέες ενημερώσεις συστήματος. Η συσκευή θα γίνει η αρχή του προγράμματος Google Play και θα ακολουθήσει άλλες συσκευές, όπως η έκδοση HTC One Google Play και η έκδοση Moto G Google Play. Από το 2008 έως το 2013, ο Hugo Barra υπηρέτησε ως εκπρόσωπος των προϊόντων, εκπροσωπώντας το Android σε συνεντεύξεις Τύπου και το Google I / O, το ετήσιο συνέδριο της Google για τους προγραμματιστές. Έφυγε από την Google τον Αύγουστο του 2013 για να ενώσει την κινεζική εταιρεία κατασκευής τηλεφώνων Xiaomi. Λιγότερο από έξι μήνες νωρίτερα, ο τότε διευθύνων σύμβουλος της Google, Larry Page ανακοίνωσε σε blog, ότι ο Andy Rubin είχε μετακομίσει από το τμήμα Android για να αναλάβει νέα έργα στο Google και ότι η Sundar Pichai θα γίνει το νέο Android. Τον Ιούνιο του 2014, η Google ανακοίνωσε το Android One, ένα σύνολο μοντέλων αναφοράς υλικού που θα επιτρέψουν στους κατασκευαστές συσκευών να δημιουργήσουν εύκολα τηλέφωνα υψηλής ποιότητας με χαμηλό κόστος, σχεδιασμένα για καταναλωτές στις αναπτυσσόμενες χώρες. Τον Σεπτέμβριο, η Google ανακοίνωσε το πρώτο σετ Android Onephones για κυκλοφορία στην Ινδία. Μια αναφορά από τις πληροφορίες τον Ιανουάριο του 2017 δήλωσε ότι η Google επεκτείνει το πρόγραμμα AndroidOne με χαμηλό κόστος στις Ηνωμένες Πολιτείες, αν και η The Verge σημειώνει ότι η εταιρεία πιθανότατα δεν θα παράγει τις ίδιες τις συσκευές.

# android

Εικόνα 2.11: Το λογότυπο του Android

Η Google εισήγαγε τα smartphone Pixel και Pixel XL τον Οκτώβριο του 2016, τα οποία διατέθηκαν στο εμπόριο ως τα πρώτα τηλέφωνα που έκανε η Google και παρουσίαζαν αποκλειστικά ορισμένες λειτουργίες λογισμικού, όπως ο Βοηθός Google (GoogleAssistant), πριν από την ευρύτερη ανάπτυξη. Τα τηλέφωνα Pixel αντικατέστησαν τη σειρά Nexus, με μια νέα γενιά τηλεφώνων Pixel που ξεκίνησε τον Οκτώβριο του 2017.

Πίνακας 2.1: Οι εκδόσεις του Android

Έκδοση	Κωδική Ονομασία	Ημερομηνία
8.1	Oreo	21-Αυγ-17
8.0		
7.1	Nougat	4-Οκτ-16
7.0		22-Αυγ-16
6.0	Marshmallow	5-Οκτ-15
5.1	Lollipop	9-Μαρ-15
5.0		3-Νοε-14
4.4	KitKat	31-Οκτ-13
4.3	Jelly Bean	24-Ιουλ-13
4.2		13-Νοε-12
4.1		9-Ιουλ-12
4.0	Ice Cream Sandwich	16-Δεκ-11
3.2	Honeycomb	15-Ιουλ-11
2.3	Gingerbread	9-Φεβ-11
2.2	Froyo	20-Μαϊ-10
2.0	Eclair	26-Οκτ-09
1.6	Donut	15-Σεπ-09

## 2.3 Σύγκριση με άλλα λογισμικά

Ο ανταγωνισμός μεταξύ των κατασκευαστών έξυπνων συσκευών, ειδικά έξυπνων τηλεφώνων (smartphones), είναι τεράστιος και συνεχώς αυξάνεται. Ανταγωνίζονται ο ένας τον άλλον πάνω σε διαφορετικά χαρακτηριστικά των συσκευών όπως χωρητικότητα αποθήκευσης, την ταχύτητα του επεξεργαστή, την ανάλυση της φωτογραφικής μηχανής κλπ. Όλα αυτά σίγουρα είναι βασικά χαρακτηρίστηκα για την επιλογή μια συσκευής

αλλά όχι όσο είναι το λειτουργικό σύστημα το οποίο κάνει τη μεγάλη διαφορά. Όταν μιλάμε για λειτουργικό σύστημα με εννοούμε μόνο το interface αλλά για το σύνολο των χαρακτηριστικών που προσφέρει. Για αυτό το λόγο επιχειρούμε να κάνουμε μια σύντομη σύγκριση του Android με κάποια άλλα λειτουργικά. Οι επιλογές λειτουργικών που υπάρχουν διαθέσιμα στην αγορά είναι αρκετά. Εμείς επιλέξαμε τα πιο διαδεδομένα όπως το iOS της Apple και των Windows. Επίσης, επιλέξαμε πέντε βασικές κατηγορίες γύρω από τις οποίες θα πραγματοποιήσουμε την σύγκριση μας οι οποίες είναι η προσιτότητα, οι εφαρμογές, η ενημέρωση, η προσαρμοστικότητα (Customizability) και το περιβάλλον (interface).

### ➤ Προσιτότητα

Το πρώτο χαρακτηριστικό που θα συγκρίνουμε είναι ίσως και το πιο βασικό είναι η τιμή. Είναι γεγονός ότι η τιμή αφορά ολόκληρη τη συσκευή και όχι μεμονωμένα το λειτουργικό αλλά επειδή αυτά τα δύο είναι άρρηκτα συνδεδεμένα, η τιμή της συσκευής αντικατοπτρίζει σε μεγάλο βαθμό και εκείνη του λειτουργικού. Η Apple όπως ξέρουμε διαθέτει smartphones στην αγορά τα οποία έχουν αρκετά τσιμημένη τιμή, που δεν την αντέχουν πολλοί. Το Android απ' την άλλη κατέχει μεγαλύτερη γκάμα κινητών με ποικίλες τιμές. Παρομοίως και τα Windows έχουν αρκετά προσιτές-χαμηλές τιμές. Σε αυτό που υπερτερεί το Android είναι η τεράστια ποικιλία συσκευών, μιας και αρκετοί κατασκευαστές το έχουν επιλέξει σαν λειτουργικό των συσκευών τους.

### ➤ Εφαρμογές

Εδώ υπάρχει μια μεγάλη διαμάχη όχι τόσο σχετικά με τον αριθμό αλλά με την ποιότητα των εφαρμογών. Παραθέτοντας ενδεικτικά κάποια νούμερα το Android κατέχει περίπου 3 εκατομμύρια εφαρμογές, το iOS 2.2 εκατομμύρια και τα Windows περίπου 500.000. Παρόλα αυτά παραδοσιακά το iOS υπήρξε μια πιο κερδοφόρα πλατφόρμα για προγραμματιστές οπότε υπήρξε μια τάση για νέες εφαρμογές να εμφανίζονται εκεί πρώτα, κάτι που αρχίζει να αλλάζει καθώς το μερίδιο αγοράς του Android συνεχίζει να αυξάνεται. Επίσης υπάρχει η αντίληψη ότι η Apple έχει περισσότερες «ποιοτικές» εφαρμογές σε σχέση με το Android. Αυτό αλλάζει αλλά εν μέρει ισχύει καθώς εφαρμογές για μουσική, φωτογραφίες, βίντεο και άλλα υπάρχουν σε μεγάλη γκάμα και καλύτερης ποιότητας στο Apple Store από ότι στο Google Play. Δεν μπορούμε να πούμε το ίδιο για τα Windows καθώς βρίσκονται αρκετά πίσω όσο αναφορά και αριθμό και ποιότητα, συνεπώς είναι εκτός ανταγωνισμού.

### ➤ **Ενημέρωση Λογισμικού**

Όσον αφορά την ενημέρωση λογισμικού να αναφέρουμε ότι και οι 3 ενημερώνονται τακτικά. Οι εταιρίες φροντίζουν απ' την μία να βγάζουν μεγάλες αναβαθμίσεις με επανασχεδιασμό και νέες ενέργειες και απ' την άλλη μικρότερες με διορθώσεις σφαλμάτων. Ίσως η Apple να έχει ένα μικρό προβάδισμα καθώς κάθε χρόνο αφήνει πίσω της μια συσκευή όσον αφορά τις αναβαθμίσεις, αλλά τα καταφέρνει καλύτερα απ' το Android που είναι στο έλεος των εκάστοτε εταιριών, και έχει τους πελάτες να περιμένουν για τις αναβαθμίσεις.

### ➤ **Προσαρμοστικότητα (Customization)**

Σε αυτή την κατηγορία το Android φαίνεται να υπερτερεί σημαντικά. Διαθέτει ποικίλες εφαρμογές για να αλλάξεις το καθετί. Μπορείς να εγκαταστήσεις διαφορετικούς launcher που θα αλλάξουν το στυλ του UI (Διεπαφή Χρήστη). Επίσης μπορείς να τοποθετήσεις κλείδωμα οθόνης, background, widgets ενώ στο συγκεκριμένο κομμάτι τα άλλα 2 λογισμικά έχουν περιορισμένες δυνατότητες. Το iOS από τη μεριά του παρέχει κάποιες αλλαγές αλλά περιορισμένες. Πλέον υποστηρίζει ξεχωριστά πληκτρολόγια αλλά δεν φτάνει το Android. Σε μια παρόμοια κατάσταση βρίσκονται και τα Windows.

Υπάρχουν πολλές κατηγορίες για να συγκρίνουμε τα λογισμικά όπως, συνδεσιμότητα, ασφάλεια, χάρτες, φωνητικοί βοηθοί κτλ αλλά πιστεύουμε περιπλέκουν την κατάσταση και δημιουργούν διχόνοια. Ίσως βέβαια να τα αναφέρουμε σε ένα επόμενο άρθρο ακόμα πιο αναλυτικά.

Καταλαβαίνουμε βέβαια ότι η επιλογή κινητού είναι μία καθαρά προσωπική άποψη, ανάλογα με τις ανάγκες του καθενός, και δεν υπάρχει υποβαθμισμένο κινητό, αναφερόμενοι βέβαια στις 2 μεγάλες εταιρίες καθώς το WP έχει ακόμα πολλά να δώσει. Κάθε λογισμικό έχει τα υπέρ και τα κατά που μας οδηγούν στην απόκτηση του, και αν πρόκειται να συγκρίνουμε 2 διαφορετικά software αυτό δεν είναι τα τεχνικά χαρακτηριστικά.

Εσείς είστε ευχαριστημένοι από τις αναβαθμίσεις της εταιρίας σας; Θα αλλάζατε στην επόμενη αγορά κινητού για να πάτε σε άλλο λογισμικό;

### ➤ **Περιβάλλον και Σχεδιασμός**

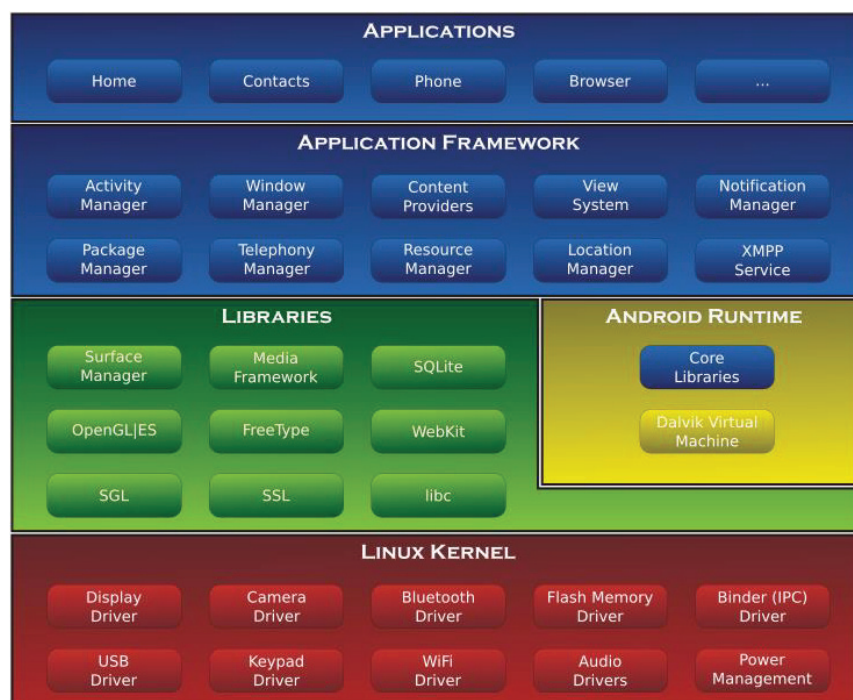
Σε ότι αφορά το εσωτερικό του λογισμικού και συγκεκριμένα στο UI (User Interface) καταλαβαίνουμε καλύτερα την μεγάλη διαφορά μεταξύ των λογισμικών. Το Android



απ'τη μεριά του διαφέρει έστω και λίγο από κινητό σε κινητό καθώς οι εταιρίες αφήνουν το δικό τους στίγμα στο UI. Η βασική αρχή όμως παραμένει η ίδια, στην αρχική οθόνη υπάρχουν συντομεύσεις εφαρμογών, με την μπάρα ειδοποιήσεων στο πάνω μέρος, και το συρτάρι εφαρμογών όπως ονομάζεται στην συνέχεια. Το iOS έκανε μια μεγάλη αλλαγή στις τελευταίες ενημερώσεις. Οι εφαρμογές βρίσκονται όλες στην αρχική, πλέον διαθέτει και μία μπάρα ειδοποιήσεων από την οποία είναι δυνατή η πρόσβαση σε ένα control center για βασικές ρυθμίσεις και τις ειδοποιήσεις. Τα Windows μοιάζουν αρκετά με εκείνα του υπολογιστή, καθώς αποτελούνται απ' εικονίδια τα οποία αλλάζουν μέγεθος και θέση για να ταιριάζουν στον χρήστη. Επίσης μπορείτε να τα διαμορφώσετε όπως εσείς προτιμάτε.

## 2.4 Επίπεδα Αρχιτεκτονικής

Η αρχιτεκτονική του android στηρίζεται πάνω σε μια στοίβα επίπεδων. Όπως βλέπουμε στην εικόνα 2.3 τα επίπεδα αυτά είναι τέσσερα (4) και παρουσιάζονται στη συνέχεια.



Εικόνα 2.3: Το διάγραμμα της αρχιτεκτονικής του Android

### ➤ Linux Kernel

Στο χαμηλότερο επίπεδο δηλαδή στη βάση της αρχιτεκτονικής βρίσκεται ο πυρήνας των linux. Πρόκειται για ένα πυρήνα λειτουργικού συστήματος που χρησιμοποιείτε από την οικογένεια των λειτουργικών συστημάτων Linux, το οποίο βασίζεται στο Unix (δεν χρησιμοποιεί κώδικα από αυτό αλλά θεωρείται παρεμφερές σύστημα – ο ακριβής όρος

είναι Unix –οειδές ). Ο ρόλος του είναι να παρέχει τα απαιτούμενα προγράμματα οδήγησης (drivers). Πιο συγκεκριμένα , κάποια από τα βασικά στοιχεία που παρέχει είναι:

- υποστήριξη πολύ επεξεργαστικών συστημάτων (SMP)
- πραγματική πολυδιεργασία
- εικονική μνήμη
- διαμοιραζόμενες βιβλιοθήκες
- σωστή διαχείριση μνήμης
- δικτύωση μέσω TCP/IP

#### ➤ **Libraries**

Πάνω από τον πυρήνα του Linux υπάρχουν οι φυσικές βιβλιοθήκες που περιέχουν των κώδικα του λειτουργικού και βοηθάνε ώστε να παρέχονται τα απαιτούμενα χαρακτηριστικά στον χρήστη. Για παράδειγμα η βιβλιοθήκη WebKit είναι υπεύθυνη για την υποστήριξη του προγράμματος περιήγησης (webbrowser), η SQLite παρέχει υποστήριξη αποθήκευσης στην βάση δεδομένων , ενώ η FreeType για την υποστήριξη της γραμματοσειράς.

#### ➤ **Android Runtime**

Στο ίδιο επίπεδο συναντάμε το Android Runtime, το οποίο περιέχει τις βιβλιοθήκες πυρήνα (core libraries) και την εικονική μηχανή Dalvik (Dalvik virtual machine). Οι βιβλιοθήκες πυρήνα είναι ένα σύνολο βασικών βιβλιοθηκών που επιτρέπουν στους προγραμματιστές να γράψουν εφαρμογές χρησιμοποιώντας JAVA. Η εικονική μηχανή Dalvik επιτρέπει κάθε εφαρμογή να τρέξει την δικιά της εργασία. Πρόκειται για μια εξειδικευμένη εικονική μηχανή, ειδικά διαμορφωμένη για κινητές συσκευές που έχουν περιορισμένη μνήμη και ισχύ με στόχο την γρηγορότερη απόδοση.

#### ➤ **Application Framework**

Στο επόμενο επίπεδο βρίσκεται το Android framework το οποίο περιλαμβάνει τα API(Application Programming Interface) του Android, όπως UI (User Interface), τηλεφωνία, πόρους, τοποθεσίες, παροχές περιεχομένου (δεδομένα) και διαχειριστές πακέτων. Πέραν αυτών παρέχει πολλές κατηγορίες και διεπαφές για ανάπτυξη εφαρμογών Android από τους προγραμματιστές.

### ➤ Applications

Στην κορυφή της αρχιτεκτονικής βρίσκεται το επίπεδο εφαρμογών. Σε αυτό περιλαμβάνονται τόσο οι εφαρμογές που είναι προ-εγκατεστημένες και αφορούν τη λειτουργία της συσκευή του χρήστη (τηλέφωνο, επαφές, μουσική κ.α.), όσο και εκείνες που εγκαθιστά από μόνος του ο χρήστης σύμφωνα με τις ανάγκες του. Όλες αυτές χρησιμοποιούν το Android framework , το οποίο με τη σειρά του χρησιμοποιεί το android runtime και τις βιβλιοθήκες του Android και αυτά με τη σειρά τους χρησιμοποιούν το Linux Kernel.

### **3. ΤΙ ΕΙΝΑΙ ΤΑ MOBILE APPLICATIONS**

Τα mobile application (ή mobile apps όπως τις συναντάμε κατά κόρον) είναι προγράμματα λογισμικού που έχουν αναπτυχθεί για κινητές συσκευές όπως smartphones και tablet. Μετατρέπουν τις κινητές συσκευές σε μικροσκοπικές μονάδες λειτουργίας και διασκέδασης. Ορισμένες συσκευές κυκλοφορούν με ορισμένες προ-φορτωμένες εφαρμογές, προερχόμενες από τους κατασκευαστές τους ή από τους παροχείς υπηρεσιών κινητής τηλεφωνίας με τους οποίους είναι συνδεδεμένοι. Πέραν αυτών, πολλές εφαρμογές είναι διαθέσιμες μέσω των appstores που αφορούν την εκάστοτε συσκευή.

#### **➤ Λειτουργίες**

Οι στόχοι αυτών των εφαρμογών είναι να τρέξουν το φάσμα των δυνατοτήτων των συσκευών από τη χρησιμότητα, την παραγωγικότητα και την πλοήγηση μέχρι την ψυχαγωγία, τον αθλητισμό, τη γυμναστική και οτιδήποτε άλλο μπορεί να φανταστεί κανείς. Τα κοινωνικά μέσα ενημέρωσης είναι ένας από τους πιο δημοφιλείς τομείς ανάπτυξης αυτών των εφαρμογών. Για παράδειγμα το Facebook το οποίο υπήρξε η πιο ευρέως χρησιμοποιούμενη εφαρμογή για το 2017 σε όλες τις πλατφόρμες. Πολλές διαδικτυακές οντότητες διαθέτουν και ιστότοπους για κινητά και mobileapps. Σε γενικές γραμμές, η διαφορά έγκειται στο σκοπό: Μια εφαρμογή είναι συνήθως μικρότερη σε σχέση με έναν ιστότοπο για κινητά, προσφέρει περισσότερη διαδραστικότητα και παρουσιάζει πιο συγκεκριμένες πληροφορίες σε μορφή που είναι εύκολη και διαισθητική για χρήση σε μια κινητή συσκευή.

#### **➤ Συμβατότητα λειτουργικού συστήματος**

Ένας προγραμματιστής εφαρμογών για κινητές συσκευές δημιουργεί μια εφαρμογή ειδικά για το λειτουργικό σύστημα στο οποίο θα τρέξει. Για παράδειγμα, οι κινητές εφαρμογές για το iPad υποστηρίζονται από το iOS της Apple, αλλά όχι από το Android της Google. Μια εφαρμογή της Apple δεν μπορεί να εκτελεστεί σε τηλέφωνο Android και αντίστροφα. Συχνά, οι προγραμματιστές δημιουργούν μια έκδοση για το καθένα.

### ➤ Διαφορές από τις "κανονικές" εφαρμογές

Πολλές εφαρμογές για κινητές συσκευές έχουν αντίστοιχα προγράμματα που προορίζονται να λειτουργούν σε επιτραπέζιους υπολογιστές. Ωστόσο, οι εφαρμογές για κινητά πρέπει να λειτουργούν με διαφορετικούς περιορισμούς από τα ισοδύναμα τους. Οι κινητές συσκευές έχουν μεγάλη ποικιλία μεγεθών οθόνης, χωρητικότητας μνήμης, δυνατότητες επεξεργαστών, γραφικές διεπαφές, κουμπιά και λειτουργίες αφής και οι προγραμματιστές πρέπει να τα λάβουν όλα υπόψη τους. Για παράδειγμα, οι χρήστες κινητών εφαρμογών (όπως οι επισκέπτες του ιστότοπου) δεν θέλουν να 'σκρολάρουν' προς τα πλάγια για να δουν το κείμενο, τις εικόνες ή τα διαδραστικά σημεία επαφής, ούτε θέλουν να δυσκολευτούν με την ανάγνωση μικροσκοπικού κειμένου. Μια πρόσθετη μέριμνα για τους προγραμματιστές κινητών εφαρμογών είναι η δυνατότητα αφής που είναι κοινή στις κινητές συσκευές.

### ➤ "Mobile - First" Ανάπτυξη

Πριν από τη γενικευμένη υιοθέτηση κινητών συσκευών, το λογισμικό αναπτύχθηκε για πρώτη φορά σε επιτραπέζιους υπολογιστές και φορητούς υπολογιστές, με την κινητή έκδοση να έρχεται μετά. Η χρήση tablet και smartphone ξεπερνά εκείνη των επιτραπέζιων υπολογιστών και των φορητών υπολογιστών, αντανακλώντας τις τάσεις των πωλήσεων εφαρμογών. Στην πραγματικότητα, προβλέφθηκε ότι μέσα στο 2018 οι χρήστες θα «κατεβάσουν» 197 δισεκατομμύρια εφαρμογές. Ως εκ τούτου, πολλοί προγραμματιστές έχουν στραφεί σε μια προσέγγιση "mobile - first", αντικατοπτρίζοντας μια παρόμοια τάση στον σχεδιασμό ιστοσελίδων.

#### 4. ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ANDROID

Η ανάπτυξη εφαρμογών για κινητές συσκευές απαιτεί την εξέταση των περιορισμών και των δυνατοτήτων αυτών των συσκευών. Οι κινητές συσκευές λειτουργούν με μπαταρία και διαθέτουν λιγότερο ισχυρούς επεξεργαστές από τους προσωπικούς υπολογιστές και έχουν επίσης περισσότερες δυνατότητες, όπως ανίχνευση θέσης και κάμερες. Οι προγραμματιστές πρέπει επίσης να εξετάσουν ένα ευρύ φάσμα μεγεθών οθόνης, προδιαγραφών υλικού και διαμορφώσεων λόγω του έντονου ανταγωνισμού στο λογισμικό κινητών συσκευών και των αλλαγών σε κάθε πλατφόρμα (αν και αυτά τα θέματα μπορούν να ξεπεραστούν με την ανίχνευση κινητών συσκευών).

Η ανάπτυξη εφαρμογών για κινητά απαιτεί τη χρήση εξειδικευμένων ολοκληρωμένων αναπτυξιακών περιβαλλόντων (integrated development environments-IDE). Οι εφαρμογές για κινητά δοκιμάζονται πρώτα μέσα στο περιβάλλον ανάπτυξης χρησιμοποιώντας εξομοιωτές και αργότερα υποβάλλονται σε δοκιμές πεδίου. Οι εξομοιωτές παρέχουν έναν φθηνό τρόπο με τον οποίο οι εφαρμογές μπορούν να δοκιμαστούν σε κινητά τηλέφωνα στα οποία οι προγραμματιστές ενδέχεται να μην έχουν φυσική πρόσβαση.

Ο σχεδιασμός διεπαφής χρήστη για κινητά (UserInterface - UI) είναι επίσης απαραίτητος. Το UI εξετάζει τους περιορισμούς και τα περιβάλλοντα, την οθόνη, την είσοδο και την κινητικότητα ως περιγράμματα για το σχεδιασμό. Ο χρήστης είναι συχνά το επίκεντρο αλληλεπίδρασης με τη συσκευή του και η διασύνδεση περιλαμβάνει στοιχεία υλικού και λογισμικού. Η είσοδος χρήστη επιτρέπει στους χρήστες να χειρίζονται ένα σύστημα και η έξοδος της συσκευής επιτρέπει στο σύστημα να υποδείξει τα αποτελέσματα του χειρισμού των χρηστών. Οι περιορισμοί σχεδιασμού του UI περιλαμβάνουν λίγους παράγοντες προσοχής και μορφής, όπως το μέγεθος της οθόνης μιας κινητής συσκευής για το χέρι του χρήστη. Τα γενικό πλαίσιο του UI υποδεικνύεται από τη δραστηριότητα των χρηστών, όπως η τοποθεσία και ο προγραμματισμός που μπορούν να εμφανιστούν από τις αλληλεπιδράσεις των χρηστών μέσα σε μια εφαρμογή για κινητά. Γενικά, ο στόχος του σχεδιασμού του UI είναι κατά κύριο λόγο ένα κατανοητό, φιλικό περιβάλλον προς το χρήστη.

Το UI (πολλές φορές ονομάζεται και front-end) βασίζεται στην υποστήριξη που παρέχει το επίπεδο πρόσβασης δεδομένων (data access layer ή πιο γνωστό ως back-end) προκειμένου να αποκτήσει πρόσβαση στο σύστημα. Το back-end διευκολύνει τη

δρομολόγηση δεδομένων, την ασφάλεια, τον έλεγχο ταυτότητας, την εξουσιοδότηση, την εκτός σύνδεσης λειτουργία (off-line) και την ενορχήστρωση των υπηρεσιών. Αυτή η λειτουργικότητα υποστηρίζεται από ένα μείγμα μέσω του μεσαίου λογισμικού, συμπεριλαμβανομένων των διακομιστών κινητής εφαρμογής (mobile app servers), του MBaaS (Mobile Back end as a service) και της υποδομής SOA (service-oriented architecture).

## 5. Η ΔΙΔΑΚΤΙΚΗ ΤΩΝ ΜΑΘΗΜΑΤΙΚΩΝ

Τα μαθηματικά είναι η γλώσσα των αριθμών. Είναι ένας κόσμος με τύπους, ορισμούς, θεωρίες, δυσκολίες και ανακαλύψεις, ο οποίος έχει καταφέρει να αποκτήσει φανατικούς οπαδούς και ορκισμένους εχθρούς. Το βέβαιο είναι ότι τα μαθηματικά επηρέασαν και επηρεάζουν την εξέλιξη του κόσμου και του ανθρώπου συμβάλλοντας σημαντικά - ιδιαίτερα στις μέρες μας- στην έκρηξη της τεχνολογίας και των επιτευγμάτων του ανθρώπινου νου.

### 5.1 Παιδαγωγικές αρχές για την διδασκαλία των μαθηματικών

Όσον αφορά την παιδαγωγική των μαθηματικών επηρεάζεται από δυο κυρίως αρχές: του απολυτισμού και του ημιεμπειρικισμού.

- Σύμφωνα με τις αρχές του απολυτισμού:
  - Τα μαθηματικά είναι ένα σύνολο από αιώνιες αλήθειες, ανεξάρτητες από κοινωνικές και ιδεολογικές επιδράσεις. Είναι δηλαδή ουδέτερα και όλοι οι μαθητές μπορούν να τα καταλάβουν, αρκεί να διαθέτουν το κατάλληλο μυαλό.
  - Τα μαθηματικά είναι ένα σπουδαίο εργαλείο για την πειθαρχία του νου και την πνευματική συγκρότηση.
  - Τα σχολικά βιβλία περιέχουν αυστηρά δομημένη και ουδέτερη γνώση, η οποία έχει διάφορα επίπεδα που μπορούν να κατακτήσουν οι μαθητές ανάλογα με τις πνευματικές ικανότητές τους.
  - Τα λάθη στα μαθηματικά είναι δηλωτικά της αποτυχίας του μαθητή να φτάσει στη γνώση και της αδυναμίας του να εκπληρώσει τις υποχρεώσεις του στο σχολείο. Το λάθος επομένως δεν έχει καμιά παιδαγωγική αξία.
- Σύμφωνα με τις αρχές του ημιεμπειρικισμού:
  - Οι μαθηματικές αλήθειες δεν είναι απόλυτες και είναι προϊόντα της ανθρώπινης δραστηριότητας. Περιέχουν επομένως το στοιχείο της αβεβαιότητας. Η μαθηματική γνώση αποχτιέται από το μαθητή με την αλληλεπίδρασή του μέσα στο κοινωνικό σύνολο, δηλαδή χτίζεται σιγά σιγά.
  - Τα μαθηματικά διδάσκονται για να αναπτύξουν οι μαθητές την κριτική σκέψη. Με την επίλυση προβλημάτων μέσα από τη ζωή αποχτούν οι μαθητές θετική στάση απέναντι στα μαθηματικά. Ο μαθητής πρέπει να συμμετέχει σε δραστηριότητες που περιέχουν προβληματικές καταστάσεις τις οποίες προσπαθεί να λύσει.
  - Τα σχολικά μαθηματικά δεν πρέπει να θεωρούνται σαν επιβαλλομένη γνώση, αλλά να ελκύουν το μαθητή σε μια περιπέτεια με δοκιμές, πειραματισμούς,



υποθέσεις, τροποποιήσεις. Η μαθηματική γνώση δεν προσφέρεται έτοιμη αλλά καταχτιέται.

- Το λάθος στα μαθηματικά θεωρείται ένα σπουδαίο παιδαγωγικό εργαλείο για το ψάξιμο των παραγόντων που εμποδίζουν την κατανόηση του περιεχομένου της μάθησης.

## **5.2 Σύγχρονες αρχές της διδασκαλίας των μαθηματικών**

Μέχρι πριν λίγα χρόνια οι δάσκαλοι θεωρούσαν ότι οι μαθητές ήταν άδεια δοχεία τα οποία έπρεπε να γεμίσουν με μαθηματικές γνώσεις. Αρκούσε να ήταν επιμελείς και προσεκτικοί, αρκούσε να παρακολουθούν προσεκτικά την παράδοση του δασκάλου και θα έκαναν κτήμα τη μαθηματική γνώση χωρίς πολλή προσπάθεια. Σήμερα η παιδαγωγική άποψη για τη διδασκαλία των μαθηματικών διαφέρει αρκετά από την προηγούμενη θεώρηση. Η κοινωνική σημασία της μάθησης οδηγεί στο συμπέρασμα ότι μαθητής για να κάνει κτήμα του τη μαθηματική γνώση θα πρέπει να εμπλακεί σε διαδικασίες ανακάλυψής της, να κινηθεί για να την αποκτήσει. Ο δάσκαλος θα πρέπει να σχεδιάζει δραστηριότητες με τις οποίες θα ασχοληθεί ο μαθητής, να παρακινεί το μαθητή, να τον ενισχύει, να τον καθοδηγεί ώστε να ανακαλύψει τη γνώση και να εκμεταλλευτεί την παιδαγωγική σημασία του λάθους. Το λάθος στα μαθηματικά δεν είναι σημάδι αδυναμίας ή άγνοιας αλλά είναι στοιχείο για την ανακάλυψη των παραγόντων που δεν επιτρέπουν στο άτομο να αναπτύξει τη μαθηματική σκέψη στον ίδιο βαθμό με τα υπόλοιπα άτομα. Το κλίμα της τάξης επίσης θα πρέπει να είναι ευχάριστο και πρόσφορο για δημιουργικές δραστηριότητες που εξασφαλίζουν την ισότιμη συμμετοχή όλων των παιδιών.

## **5.3 Δασκαλικές ενέργειες για την υποβοήθηση της ανακάλυψης στα μαθηματικά**

Έχουν ορισθεί κάποιες ενέργειες με τις οποίες ο δάσκαλος επιτυγχάνει την ομαλότερη και αποτελεσματικότερη εκμάθηση και κατανόηση των μαθηματικών στους μαθητές. Αυτές οι ενέργειες περιλαμβάνουν τα ακόλουθα:

- Ο δάσκαλος πρέπει να έχει ορίσει απ' την αρχή τους στόχους της μαθηματικής ενότητας και τις διδακτικές στρατηγικές που θα εφαρμόσει κατά τη διδασκαλία.
- Πρέπει να προσφέρει την κατάλληλη καθοδήγηση που χρειάζεται το παιδί στη συγκεκριμένη στιγμή. Η καθοδήγηση θα πρέπει να είναι μεγαλύτερη στην αρχή της προσπάθειας και να ελαττώνεται όσο εξελίσσεται η προσπάθεια του μαθητή για την εύρεση της λύσης.

- Θα πρέπει ο δάσκαλος να θέτει ερωτήματα ή να κάνει υποθέσεις και να δείχνει πολλά παραδείγματα στα παιδιά. Θα πρέπει επίσης να δίνει την ευχέρεια στα παιδιά να κάνουν τα ίδια υποθέσεις και να συζητούν για την ορθότητα των υποθέσεών τους.
- Θα πρέπει να κάνει συχνά ανακεφαλαιώσεις κατά τη διαδικασία της ανακαλυπτικής διαδικασίας. Με τις ανακεφαλαιώσεις τα παιδιά οργανώνουν τα δεδομένα που υπάρχουν ως τη στιγμή εκείνη σε οργανωμένα σύνολα και προχωρούν στα επόμενα βήματα έχοντας υπόψη τις στρατηγικές που ακολούθησαν ως εδώ.
- Εκείνη που «μετράει» στα μαθηματικά είναι η απόδειξη, η οποία εξασφαλίζει την τελική αλήθεια των συλλογισμών. Ο δάσκαλος δεν πρέπει να επιμένει στην προφορική έκφραση των συμπερασμάτων από όλα τα παιδιά ιδίως στις μικρές τάξεις, γιατί συμβαίνει πολλές φορές να έχει κάποιο παιδί σχηματίσει την ιδέα της λύσης στο μυαλό του αλλά να μη μπορεί να την εκφράσει. Ο δάσκαλος με κατάλληλα παραδείγματα οφείλει να καταλάβει αν η σκέψη του παιδιού βρίσκεται στο σωστό δρόμο όσο αφορά τη λύση μαθηματικών προβλημάτων και να το βοηθήσει να την εκφράσει προφορικά.

Πέραν των παραπάνω καταλυτικό ρόλο παίζει και η εργαστηριακή μάθηση(learning by doing). Η μάθηση αυτή μπορεί να εφαρμοστεί στις πρώτες τάξεις του δημοτικού σχολείου, αλλά σε ορισμένες περιπτώσεις θεωρείται χρήσιμη και απαραίτητη ακόμα και στις μεγαλύτερες τάξεις. Οι μαθητές ασκούνται αρχίζοντας σε πολύ μικρή ηλικία με κατάλληλο υλικό(ψαλίδι, χαρτί, σχήματα, γεωπίνακας, καθημερινά αντικείμενα) σε απλές μαθηματικές έννοιες, ώστε αργότερα να είναι σε θέση να κάνουν πιο πολύπλοκους υπολογισμούς. Η εργαστηριακή μάθηση όμως προϋποθέτει σοβαρό και υπεύθυνο προγραμματισμό από το δάσκαλο. Πρέπει επίσης να έχουν ασκηθεί οι μαθητές προηγουμένως στη χρήση του εποπτικού υλικού και να γνωρίζουν τι θα κάνουν κάθε φορά. Απαιτείται δηλαδή καλή οργάνωση της διδασκαλίας.

Στην αρχή ο δάσκαλος ασκεί μεγαλύτερη καθοδήγηση η οποία στη συνέχεια γίνεται πιο χαλαρή. Οι μαθητές μπορούν να χωριστούν σε ομάδες και να επωφελούνται από τη συνεργασία μεταξύ τους. Ο δάσκαλος κατά την εργαστηριακή μάθηση πρέπει να βοηθάει τις ομάδες όταν φτάνουν σε αδιέξοδο, να παρατηρεί συνέχεια την εργασία των μαθητών, να κάνει ερωτήσεις για να ξεκαθαρίσει διάφορες ιδέες, να ενθαρρύνει τους μαθητές στη συνεργασία, να τους αφήνει να φτάνουν στη λύση με το δικό του τρόπο ο καθένας, να έχει μεγάλα αποθέματα υπομονής, να βραβεύει την πρωτοτυπία, την φαντασία και την

συνεργασία, να κάνει συχνές ανακεφαλαιώσεις, για να δώσει έμφαση σε ορισμένες τεχνικές.

#### **5.4 Τα Συστατικά μιας επιτυχημένης διδασκαλίας στα μαθηματικά**

Η διδασκαλία των μαθηματικών έχει στόχο να κεντρίσει το ενδιαφέρον του μαθητή μέσα από μια προβληματική κατάσταση και να προκαλέσει την ενεργοποίηση των διανοητικών του ικανοτήτων με δραστηριότητες στις οποίες θα εμπλακεί ως την επίλυσή της. Για να θεωρηθεί επιτυχημένη μια διδασκαλία των μαθηματικών πρέπει να αρχίσει με μια οργανωμένη παρουσίαση από το δάσκαλο. Η παρουσίαση πρέπει να έχει μικρή διάρκεια, να διαθέτει ζωντάνια, να εκφράζεται με λόγο πρόσχαρο, απλό, λιτό, να εκφράζει θετική στάση ως προς το αντικείμενο της μάθησης και να στρέψει το μαθητή προς την κατανόηση του προβλήματος. Ο δάσκαλος κατά την παρουσίαση λειτουργεί σαν ζωντανό μοντέλο σκέψης και με τη στάση του επηρεάζει τη στάση του μαθητή απέναντι στα μαθηματικά είτε θετικά είτε αρνητικά. Η παρουσίαση του δασκάλου λοιπόν πρέπει να γεννά θετικές προσδοκίες και να ωθεί σε δημιουργικές δραστηριότητες τους μαθητές.

Ξέρουμε σήμερα ότι η γνώση δε μεταφέρεται παθητικά από το δάσκαλο στο μαθητή, αλλά κατασκευάζεται στον εσωτερικό κόσμο του μαθητή με την ελεύθερη συμμετοχή του σε δραστηριότητες που έχουν στόχο την επίλυση προβληματικών καταστάσεων με συνεργασία, συζήτηση και ελεύθερη έκφραση. Ο δάσκαλος λοιπόν πρέπει να εξασφαλίζει συνθήκες κατάλληλες που να ευνοούν τη συνεργασία, τον διάλογο και την ελεύθερη έκφραση των απόψεων των μαθητών. Πρέπει επίσης να εξασφαλίζει την πρακτική εξάσκηση του μαθητή σε διάφορες προβληματικές καταστάσεις, στις οποίες καλείται να εφαρμόσει τις θεωρητικές γνώσεις που απέκτησε και είναι σχετικές με τις καταστάσεις αυτές. Πρέπει ο δάσκαλος να προσφέρει κίνητρα να ασχοληθούν οι μαθητές με τεχνικές και με στρατηγικές επίλυσης προβλημάτων με την παρουσίαση προβλημάτων που έχουν σχέση με την καθημερινή ζωή.

Πρέπει να κινητοποιηθεί ο μαθητής να ερευνήσει, να αναζητήσει, να μαντέψει, να δοκιμάσει, να τροποποιήσει τη σκέψη του, να ξαναδοκιμάσει μέχρι να φτάσει στο επιθυμητό αποτέλεσμα. Πρέπει το πρόβλημα που θα του δοθεί να είναι πρωτότυπο και να κεντρίζει την περιέργειά του, δεν πρέπει να είναι πολύ δύσκολο, γιατί εύκολα αποθαρρύνεται, ούτε πολύ εύκολο, γιατί γίνεται βαρετό και ατονεί το ενδιαφέρον του. Πρέπει επίσης το πρόβλημα να κρύβει κάποιο μυστήριο ώστε να διεγείρει την φαντασία του μαθητή. Ο δάσκαλος πρέπει να τονώνει συνέχεια το ενδιαφέρον του μαθητή

δείχνοντας ενθουσιασμό για το αντικείμενο που διδάσκει, γιατί ο δάσκαλος που πλήττει και δεν ενδιαφέρεται για το αντικείμενο που διδάσκει είναι αποτυχημένος και αυτό το βλέπει ο μαθητής.

## 6. ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΜΙΑ ΕΦΑΡΜΟΓΗΣ ANDROID

Τα στοιχεία της εφαρμογής (application components) είναι τα βασικά δομικά στοιχεία μιας εφαρμογής Android. Κάθε στοιχείο είναι ένα σημείο εισόδου μέσω του οποίου το σύστημα ή ένας χρήστης μπορεί να εισάγει την εφαρμογή σας. Υπάρχουν τέσσερις διαφορετικοί τύποι στοιχείων της εφαρμογής:

- Δραστηριότητες (Activities)
- Υπηρεσίες (Services)
- Δέκτες εκπομπής (Broadcast Receivers)
- Παροχείς περιεχομένου (Content Providers)

Κάθε τύπος εξυπηρετεί έναν ξεχωριστό σκοπό και έχει έναν ξεχωριστό κύκλο ζωής ο οποίος καθορίζει τον τρόπο με τον οποίο το στοιχείο δημιουργείται και καταστρέφεται. Στη συνέχεια ακολουθεί συνοπτική περιγραφή για κάθε έναν τύπο στοιχείου από τους παραπάνω.

### 6.1 Δραστηριότητες

Μια δραστηριότητα είναι το σημείο εισόδου για την αλληλεπίδραση με τον χρήστη. Αντιπροσωπεύει μια ενιαία οθόνη με μια διεπαφή χρήστη (User Interface - UI). Για παράδειγμα, μια εφαρμογή ηλεκτρονικού ταχυδρομείου μπορεί να έχει μια δραστηριότητα που εμφανίζει μια λίστα με νέα μηνύματα ηλεκτρονικού ταχυδρομείου, μια άλλη δραστηριότητα για τη σύνταξη ενός μηνύματος ηλεκτρονικού ταχυδρομείου και μια άλλη δραστηριότητα για την ανάγνωση μηνυμάτων ηλεκτρονικού ταχυδρομείου. Αν και οι δραστηριότητες συνεργάζονται για να σχηματίσουν μια συνολική εμπειρία χρήστη στην εφαρμογή email, κάθε μία είναι ανεξάρτητη από τις άλλες. Ως εκ τούτου, μια διαφορετική εφαρμογή μπορεί να ξεκινήσει οποιαδήποτε από αυτές τις δραστηριότητες αν το επιτρέπει η εφαρμογή ηλεκτρονικού ταχυδρομείου. Για παράδειγμα, μια εφαρμογή κάμερας μπορεί να ξεκινήσει τη δραστηριότητα στην εφαρμογή ηλεκτρονικού ταχυδρομείου που συνθέτει νέα μηνύματα ηλεκτρονικού ταχυδρομείου για να επιτρέψει στο χρήστη να μοιραστεί μια φωτογραφία.

### 6.2 Υπηρεσίες

Μια υπηρεσία είναι ένα σημείο εισόδου γενικού σκοπού για τη διατήρηση μιας εφαρμογής που εκτελείται στο παρασκήνιο για κάθε είδους λόγο. Πρόκειται για ένα στοιχείο που εκτελείται στο παρασκήνιο για εκτέλεση εκτεταμένων εργασιών ή για εκτέλεση εργασίας για απομακρυσμένες διεργασίες. Μια υπηρεσία δεν παρέχει διεπαφή χρήστη. Για παράδειγμα, μια υπηρεσία μπορεί να αναπαράγει μουσική στο παρασκήνιο ενώ ο χρήστης βρίσκεται σε διαφορετική εφαρμογή ή μπορεί να ανακτήσει δεδομένα

μέσω του δικτύου χωρίς να εμποδίζει την αλληλεπίδραση των χρηστών με μια δραστηριότητα. Ένα άλλο στοιχείο, όπως μια δραστηριότητα, μπορεί να ξεκινήσει την υπηρεσία και να την αφήσει να τρέξει ή να συνδεθεί σε αυτήν για να αλληλεπιδράσει με αυτήν. Υπάρχουν δύο πολύ ξεχωριστές υπηρεσίες σημασιολογίας που λένε στο σύστημα πώς να διαχειριστεί μια εφαρμογή: Οι υπηρεσίες που ξεκίνησαν λένε στο σύστημα να τις κρατήσει σε λειτουργία μέχρι να ολοκληρωθεί η εργασία τους. Αυτό θα μπορούσε να είναι ο συγχρονισμός ορισμένων δεδομένων στο παρασκήνιο ή η αναπαραγωγή μουσικής ακόμα και μετά την έξοδο από την εφαρμογή από τον χρήστη. Ο συγχρονισμός δεδομένων στο παρασκήνιο ή η αναπαραγωγή μουσικής αντιπροσωπεύουν επίσης δύο διαφορετικούς τύπους υπηρεσιών που ξεκινούν και οι οποίες τροποποιούν τον τρόπο χειρισμού του συστήματος.

### **6.3 Δέκτες εκπομπής**

Ένας δέκτης εκπομπής είναι ένα στοιχείο το οποίο επιτρέπει στο σύστημα να εκπέμπει συμβάντα στην εφαρμογή εκτός μιας κανονικής ροής του χρήστη, επιτρέποντας στην εφαρμογή να ανταποκρίνεται σε ανακοινώσεις εκπομπής σε όλο το σύστημα. Επειδή οι δέκτες εκπομπής είναι μια άλλη καλά καθορισμένη είσοδος στην εφαρμογή, το σύστημα μπορεί να παραδώσει εκπομπές ακόμα και σε εφαρμογές που δεν εκτελούνται αυτήν τη στιγμή. Έτσι, για παράδειγμα, μια εφαρμογή μπορεί να προγραμματίσει ένα συναγερμό για να δημοσιεύσει μια ειδοποίηση για να πει στο χρήστη για ένα επερχόμενο συμβάν και με την παράδοση αυτού του συναγερμού σε έναν Broadcast Receiver της εφαρμογής, δεν υπάρχει ανάγκη η εφαρμογή να παραμείνει σε λειτουργία μέχρι ο συναγερμός να σβήσει. Πολλές εκπομπές προέρχονται από το σύστημα - για παράδειγμα, μια εκπομπή που αναγγέλλει ότι η οθόνη έχει απενεργοποιηθεί, η μπαταρία είναι χαμηλή ή έχει τραβηχτεί μια φωτογραφία. Οι εφαρμογές μπορούν επίσης να εκκινήσουν εκπομπές - για παράδειγμα, για να επιτρέψουν σε άλλες εφαρμογές να γνωρίζουν ότι ορισμένα δεδομένα έχουν ληφθεί στη συσκευή και είναι διαθέσιμα για χρήση από αυτούς. Αν και οι δέκτες εκπομπής δεν εμφανίζουν διεπαφή χρήστη, ενδέχεται να δημιουργήσουν μια ειδοποίηση γραμμής κατάστασης για την ειδοποίηση του χρήστη όταν συμβεί κάποιο γεγονός εκπομπής. Συνηθέστερα, όμως, ένας δέκτης εκπομπής είναι απλώς μια πύλη προς άλλα εξαρτήματα και προορίζεται να κάνει πολύ ελάχιστη ποσότητα εργασίας.

## 6.4 Παροχές Περιεχομένου

Ένας πάροχος περιεχομένου διαχειρίζεται ένα κοινό σύνολο δεδομένων της εφαρμογής που μπορείτε να αποθηκεύσετε στο σύστημα αρχείων, σε μια βάση δεδομένων SQLite, στον ιστό ή σε οποιαδήποτε άλλη μόνιμη τοποθεσία αποθήκευσης στην οποία μπορεί να έχει πρόσβαση η εφαρμογή σας. Μέσω του παρόχου περιεχομένου, άλλες εφαρμογές μπορούν να ζητήσουν ή να τροποποιήσουν τα δεδομένα, αν το επιτρέπει ο πάροχος περιεχομένου. Για παράδειγμα, το σύστημα Android παρέχει έναν παροχέα περιεχομένου που διαχειρίζεται τα στοιχεία επικοινωνίας του χρήστη. Ως εκ τούτου, οποιαδήποτε εφαρμογή με τα κατάλληλα δικαιώματα μπορεί να ζητήσει από τον πάροχο περιεχομένου, όπως `Contacts Contract. Data`, να διαβάσει και να γράψει πληροφορίες σχετικά με ένα συγκεκριμένο άτομο. Είναι δελεαστικό να σκεφτόμαστε έναν πάροχο περιεχομένου ως μια αφαίρεση σε μια βάση δεδομένων, επειδή υπάρχουν πολλά API και υποστήριξη που ενσωματώνονται σε αυτά για αυτή την κοινή υπόθεση. Ωστόσο, έχουν διαφορετικό βασικό σκοπό από την άποψη του σχεδιασμού του συστήματος. Στο σύστημα, ένας πάροχος περιεχομένου είναι ένα σημείο εισόδου σε μια εφαρμογή για τη δημοσίευση ονομάτων δεδομένων που προσδιορίζονται από ένα σχήμα URI. Έτσι, μια εφαρμογή μπορεί να αποφασίσει πώς θέλει να αντιστοιχίσει τα δεδομένα που περιέχει σε ένα χώρο ονομάτων URI, παραδίδοντας τα URI σε άλλες οντότητες, οι οποίες με τη σειρά τους μπορούν να τις χρησιμοποιήσουν για να έχουν πρόσβαση στα δεδομένα.

Οι παροχές περιεχομένου είναι επίσης χρήσιμοι για την ανάγνωση και τη σύνταξη δεδομένων που είναι ιδιωτικά στην εφαρμογή σας και δεν μοιράζονται. Για παράδειγμα, η εφαρμογή δειγμάτων `Note Pad` χρησιμοποιεί έναν παροχέα περιεχομένου για την αποθήκευση σημειώσεων. Ένας πάροχος περιεχομένου υλοποιείται ως υποκατηγορία του `Content Provider` και πρέπει να εφαρμόσει ένα πρότυπο σύνολο API που επιτρέπουν σε άλλες εφαρμογές να εκτελούν συναλλαγές. Για περισσότερες πληροφορίες, ανατρέξτε στον οδηγό προγραμματιστών `Providers` περιεχομένου .

Μια μοναδική πτυχή του σχεδιασμού συστήματος Android είναι ότι κάθε εφαρμογή μπορεί να ξεκινήσει μια συνιστώσα μιας άλλης εφαρμογής. Για παράδειγμα, εάν θέλετε ο χρήστης να τραβήξει μια φωτογραφία με τη φωτογραφική μηχανή της συσκευής, ίσως υπάρχει κάποια άλλη εφαρμογή που το κάνει και η εφαρμογή σας μπορεί να το χρησιμοποιήσει αντί να αναπτύξει μια δραστηριότητα για να τραβήξετε μόνοι σας μια φωτογραφία. Δεν χρειάζεται να ενσωματώσετε ή ακόμα και να συνδέσετε τον κώδικα από την εφαρμογή φωτογραφικής μηχανής. Αντι'αυτού, μπορείτε απλά να ξεκινήσετε τη

δραστηριότητα στην εφαρμογή φωτογραφικής μηχανής που συλλαμβάνει μια φωτογραφία. Όταν ολοκληρωθεί, η φωτογραφία επιστρέφεται στην εφαρμογή σας, ώστε να μπορείτε να την χρησιμοποιήσετε. Για τον χρήστη, φαίνεται ότι η κάμερα είναι στην πραγματικότητα μέρος της εφαρμογής σας.

Όταν το σύστημα ξεκινά ένα στοιχείο, ξεκινά τη διαδικασία για αυτήν την εφαρμογή αν δεν εκτελείται ήδη και παράγει τις κλάσεις που απαιτούνται για το στοιχείο. Για παράδειγμα, αν η εφαρμογή σας ξεκινήσει τη δραστηριότητα στην εφαρμογή φωτογραφικής μηχανής που συλλαμβάνει μια φωτογραφία, αυτή η δραστηριότητα εκτελείται στη διαδικασία που ανήκει στην εφαρμογή φωτογραφικής μηχανής και όχι στη διαδικασία της εφαρμογής σας. Επομένως, σε αντίθεση με τις εφαρμογές στα περισσότερα άλλα συστήματα, οι εφαρμογές Android δεν έχουν ένα μόνο σημείο εισόδου (δεν υπάρχει λειτουργία `main()` ).

Επειδή το σύστημα εκτελεί κάθε εφαρμογή σε ξεχωριστή διαδικασία με δικαιώματα αρχείων που περιορίζουν την πρόσβαση σε άλλες εφαρμογές, η εφαρμογή σας δεν μπορεί να ενεργοποιήσει άμεσα ένα στοιχείο από άλλη εφαρμογή. Ωστόσο, το σύστημα Android μπορεί. Για να ενεργοποιήσετε ένα στοιχείο σε άλλη εφαρμογή, παραδώστε ένα μήνυμα στο σύστημα που καθορίζει την πρόθεσή σας να ξεκινήσετε ένα συγκεκριμένο στοιχείο. Το σύστημα ενεργοποιεί τότε το στοιχείο για εσάς.

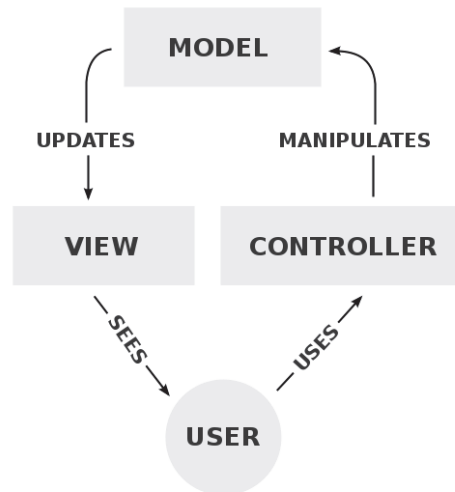


## 7. ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ

Στο κεφάλαιο αυτό θα παρουσιάσουμε την εφαρμογή που αναπτύξαμε και θα εξηγήσουμε τα βασικά της μέρη. Προτού όμως προχωρήσουμε στην παρουσίαση θα δούμε κάποιες βασικές στοιχεία πάνω στα οποία στηρίξαμε την ανάπτυξη μας.

### 7.1 Αρχές και τεχνικές αντικειμενοστραφή προγραμματισμού

Προκειμένου να πραγματοποιήσουμε την υλοποίηση της εφαρμογής μας χρησιμοποιήσαμε την ευρέως διαδεδομένη γλώσσα προγραμματισμού Java. Πρόκειται για μια αντικειμενοστραφή γλώσσα προγραμματισμού που χρησιμοποιείται κατά κόρον τόσο για την ανάπτυξη εφαρμογών σε λογισμικό Android όσο και για άλλες εφαρμογές. Η συγκεκριμένη γλώσσα υπακούει πιστά στις αρχές και τις ιδέες του αντικειμενοστραφή προγραμματισμού (Object Oriented Programming – OOP). Έτσι και εμείς κατά την ανάπτυξη του κώδικα μας χρησιμοποιήσαμε μερικές από τις πιο βασικές αυτές ιδέες. Αρχικά χρησιμοποιήσαμε την αφαίρεση των δεδομένων (data abstraction). Πρόκειται για την ιδιότητα των κλάσεων να αναπαριστούν αφαιρετικά πολύπλοκες οντότητες στο προγραμματιστικό περιβάλλον. Ουσιαστικά μια κλάση αποτελεί ένα αφαιρετικό μοντέλο κάποιας κατηγορίας αντικειμένων. Μια άλλη αρχή που ακολουθούμε είναι η ενθυλάκωση δεδομένων (data encapsulation). Αυτή είναι η ιδιότητα που προσφέρουν οι κλάσεις να «κρύβουν» τα ιδιωτικά δεδομένα τους από το υπόλοιπο πρόγραμμα και να εξασφαλίζουν πως μόνο μέσω των δημόσιων μεθόδων τους θα μπορούν αυτά να προσπελαστούν. Πέρα από τις παραπάνω αρχές, μέσα στην υλοποίηση μας χρησιμοποιούμε και ένα μοντέλο αρχιτεκτονικής λογισμικού που ονομάζεται Model-View –Controller (MVC).



Εικόνα 7.1.1: Το μοντέλο MVC

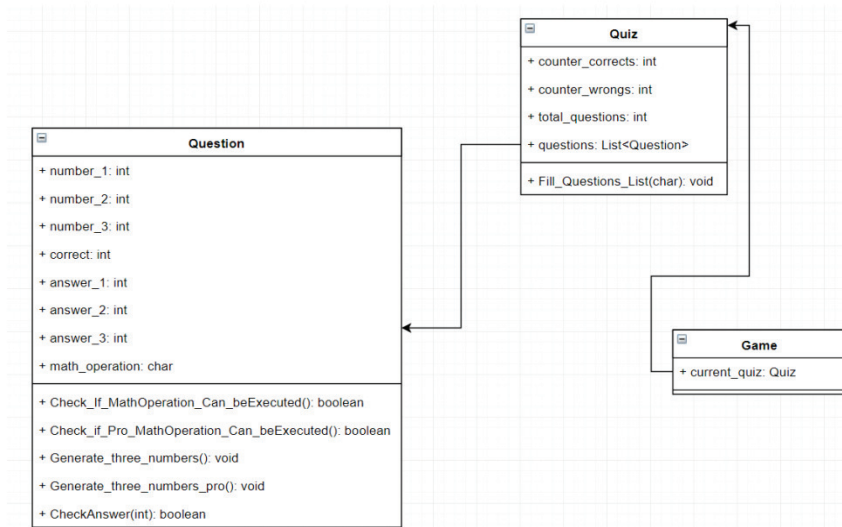
Στο μοντέλο αυτό (βλ. εικόνα 7.1.1) η εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στον χρήστη από την μορφή που έχει αποθηκευτεί στο σύστημα. Το κύριο μέρος του μοντέλου είναι το αντικείμενο Model το οποίο διαχειρίζεται την ανάκτηση/αποθήκευση των δεδομένων στο σύστημα. Το αντικείμενο View χρησιμοποιείται μόνο για να παρουσιάζεται η πληροφορία στον χρήστη (στην περίπτωση μας είναι το user interface). Το τρίτο μέρος είναι ο Controller ο οποίος δέχεται την είσοδο και στέλνει εντολές στο αντικείμενο Model και στο View.

## 7.2 Σχεδιασμός του μοντέλου MVC

Στην προηγούμενη υποενότητα έγινε αναφορά στο MVC. Το πρώτο πράγμα που απαιτείται προκειμένου να υλοποιηθεί αυτό το αρχιτεκτονικό μοντέλο είναι η δημιουργία του Model. Παρακάτω παραθέτουμε την μορφή του δικού μας Model:

Αποτελείται από τρεις βασικές κλάσεις, την κλάση Question, την κλάση Quiz και την κλάση Game. Η Question περιέχει τις μεταβλητές number\_1, number\_2, correct, answer\_1, answer\_2, answer\_3, math\_operation και τις μεθόδους Generate\_three\_numbers, CheckAnswer, Check\_If\_MathOperation\_Can\_beExecuted με τις οποίες επιτυγχάνονται η δημιουργία τριών τυχαίων αριθμών που θα αποτελούν τις τρεις από τις τέσσερις επιλογές που θα δοθούν στον χρήστη (η τέταρτη θα είναι η σωστή απάντηση), ο έλεγχος του αποτελέσματος και ο έλεγχος αν η πράξη που δημιουργούμε μπορεί να εκτελεστεί (π.χ. μπορεί οι δυο τυχαίοι αριθμοί να μην μπορούν να διαιρεθούν αν ο ένας είναι το μηδέν) αντίστοιχα. Επειδή η εφαρμογή υποστηρίζει δύσκολο επίπεδο στο οποίο

η μαθηματική πράξη περιλαμβάνει και τρίτο αριθμό χρησιμοποιούμε και τη μεταβλητή `number_3` καθώς και τη μέθοδο `Check_If_Pro_MathOperation_Can_beExecuted`. Η κλάση `Quiz` περιλαμβάνει τις μεταβλητές `counter_corrects`, `counter_wongs`, `total_questions` και `questions` που είναι μια λίστα τύπου `Question` όπου περιέχει αντικείμενα της κλάσης `Question` καθώς και την μέθοδο `Fill_Questions_List` με την οποία παράγουμε τις ερωτήσεις του `Quiz`. Τέλος η κλάση `Game` περιέχει την μεταβλητή `current_quiz` που είναι μια λίστα με αντικείμενα της κλάσης `Quiz`. Το μοντέλο αυτό παρουσιάζεται στην εικόνα 7.2.1



Εικόνα 7.2.1: ΤοUML διάγραμμα του μοντέλου της εφαρμογής

### 7.3 Βασικά εργαλεία ανάπτυξης

Για να αναπτύξουμε την εφαρμογή μας χρειάστηκε να χρησιμοποιήσουμε κάποιο προγραμματιστικό περιβάλλον. Τα τελευταία χρόνια έχει υπερισχύσει από τους προγραμματιστές των Android εφαρμογών η χρήση του Android Studio το οποίο παρέχει ένα πλήρες περιβάλλον ανάπτυξης το οποίο περιλαμβάνει όλα τα απαραίτητα στοιχεία που χρειάζονται. Μέχρι την εκτόξευση του Android Studio το περιβάλλον που χρησιμοποιούταν ήταν το Eclipse που έδινε την δυνατότητα ανάπτυξης κώδικα και άλλων γλωσσών προγραμματισμού όπως C, C ++, C #, COBOL, Fortran , JavaScript, Perl, PHP, Prolog, Python, R, Ruby.

## Android Studio

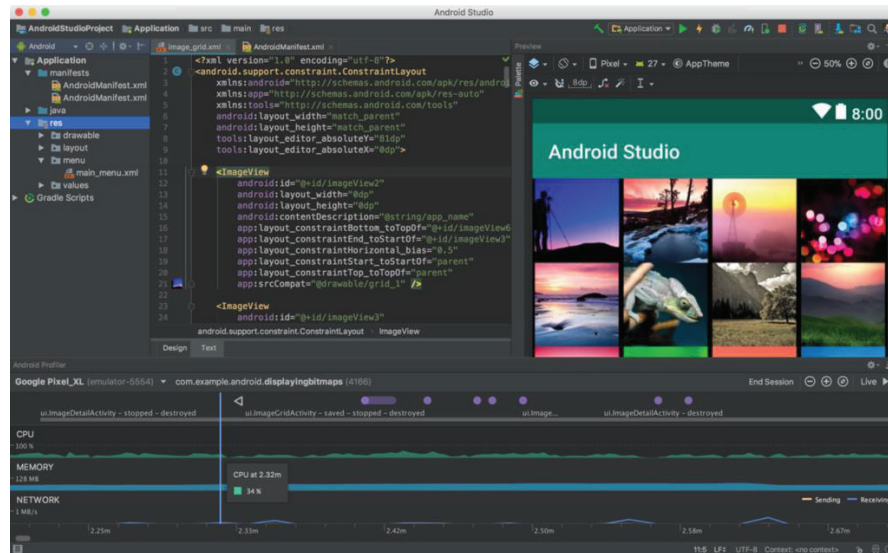
Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE) για το λειτουργικό σύστημα Android, το οποίο βασίζεται στο λογισμικό IntelliJIDEA της Jet Brains και έχει σχεδιαστεί ειδικά για την ανάπτυξη Android εφαρμογών. Είναι διαθέσιμο για λήψη σε λειτουργικά συστήματα που βασίζονται σε Windows , mac OS και Linux. Είναι αντικαταστάτης του Eclipse Android Development Tools (ADT) ως πρωτεύον IDE για την ανάπτυξη εφαρμογών Android. Το Android Studio ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο I/O Google . Ήταν σε στάδιο πρώιμης προεπισκόπησης πρόσβασης ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013 και στη συνέχεια εισήλθε σε beta έκδοση ξεκινώντας από την έκδοση 0.8 που κυκλοφόρησε τον Ιούνιο του 2014. Η πρώτη σταθερή έκδοση κυκλοφόρησε τον Δεκέμβριο του 2014, ξεκινώντας από την έκδοση 1.0. Η τρέχουσα σταθερή έκδοση είναι 3.1.4, η οποία κυκλοφόρησε τον Αύγουστο του 2018.



Εικόνα 7.3.1 Το λογότυπο του AndroidStudio

- Χαρακτηριστικά
  - Υποστήριξη δημιουργίας βάσει Gradle
  - Επανασχεδιασμός ειδικά για Android και γρήγορες επιδιορθώσεις
  - Εργαλεία για να πιάσει την απόδοση, τη χρηστικότητα, τη συμβατότητα έκδοσης και άλλα προβλήματα
  - Οδηγός προτύπων για τη δημιουργία κοινών σχεδίων και εξαρτημάτων Android

- Ένας πλούσιος επεξεργαστής διάταξης που επιτρέπει στους χρήστες να σχεδιάζουν και να παραθέτουν στοιχεία User Interface(UI) και δυνατότητα προεπισκόπησης των σχημάτων σε διάφορες διαμορφώσεις οθόνης
- Εικονική συσκευή Android (Emulator) για την εκτέλεση και τον εντοπισμό σφαλμάτων σε εφαρμογές Android.



Εικόνα 7.3.2 Το περιβάλλον του AndroidStudio

### Επιπρόσθετα Στοιχεία

Στο AndroidStudio περιλαμβάνονται δυο βασικά στοιχεία τα οποία απαιτούνται για την ανάπτυξη μιας εφαρμογής. Πρόκειται για το AndroidSDK και το AVDManager. Τα δύο αυτά στοιχεία περιλαμβάνονται στο AndroidStudio και εγκαθίστανται κατά την εγκατάσταση του περιβάλλοντος σε αντίθεση με το Eclipse στο οποίο ο χρήστης θα πρέπει να τα εγκαταστήσει ξεχωριστά.

#### ➤ **AndroidSDK**

Πρόκειται για τον εξοπλισμό ανάπτυξης λογισμικού για Android (AndroidSoftwareDevelopmentKit). Με αυτό εκτελείται η διαδικασία με την οποία δημιουργούνται νέες εφαρμογές για συσκευές που εκτελούνται στο λειτουργικό του Android. Σύμφωνα με την Google οι εφαρμογές Android μπορούν να αναπτυχθούν με τις γλώσσες Kotlin, Java και C++. Το AndroidSDK περιλαμβάνει ένα ολοκληρωμένο σύνολο εργαλείων ανάπτυξης τα οποία περιέχουν μεταξύ άλλων ένα πρόγραμμα εντοπισμού σφαλμάτων, βιβλιοθήκες, εγχειρίδια και εκπαιδευτικό υλικό.

## ➤ AVDManager

Παρέχει ένα γραφικό περιβάλλον εργασίας στο οποίο ο προγραμματιστής μπορεί να δημιουργήσει και να διαχειριστεί τις εικονικές συσκευές Android (Android Virtual Devices - AVDs), τα οποία απαιτούνται από το Emulator Android.

## 7.4 Η Γλώσσα Προγραμματισμού Java

Η Java είναι μια γλώσσα προγραμματισμού γενικής χρήσης που είναι ταυτόχρονα βασισμένη σε αντικείμενα αλλά και σχεδιασμένη ώστε να έχει όσο το δυνατόν λιγότερες εξαρτήσεις εφαρμογής.

Σκοπός είναι να επιτρέψει στους προγραμματιστές εφαρμογών να «γράψουν μία φορά και να τρέξουν οπουδήποτε» (Write Once, Run Anywhere - WORA), δηλαδή ο μεταγλωττισμένος κώδικας Java να μπορεί να τρέξει σε όλες τις πλατφόρμες που υποστηρίζουν την Java χωρίς την ανάγκη αντιγραφής . Οι εφαρμογές Java συνήθως μεταγλωττίζονται σε byte code που μπορεί να τρέξει σε οποιαδήποτε εικονική μηχανή Java (Java VirtualMachine - JVM) ανεξάρτητα από την αρχιτεκτονική του υπολογιστή.

Από το 2016, η Java είναι μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού που χρησιμοποιούνται , ιδιαίτερα για web εφαρμογές πελάτη-διακομιστή (client – server web applications). Η Java αναπτύχθηκε αρχικά από τον JamesGosling στην SunMicrosystems (η οποία από τότε έχει αποκτηθεί από την OracleCorporation ) και κυκλοφόρησε το 1995 ως βασικό στοιχείο της πλατφόρμας Java της Sun Microsystems. Η γλώσσα αποδίδει μεγάλο μέρος της σύνταξης της στη C και τη C ++ , αλλά έχει λιγότερες εγκαταστάσεις χαμηλού επιπέδου από ότι αυτές οι δύο. Η τελευταία έκδοση είναι η Java 10 , η οποία κυκλοφόρησε στις 20 Μαρτίου 2018, η οποία ακολούθησε την Java 9 μετά από μόλις έξι μήνες. Η Java 8 εξακολουθεί να υποστηρίζεται, αλλά δεν θα υπάρχουν περισσότερες ενημερώσεις ασφαλείας. Οι εκδόσεις που προηγούνται της Java 8 υποστηρίζονται από εταιρείες σε εμπορική βάση. π.χ. από την Oracle.

## ➤ Αρχές

Υπήρχαν πέντε πρωταρχικοί στόχοι στη δημιουργία της γλώσσας, οι οποίοι δήλωναν ότι πρέπει:

- να είναι απλή, αντικειμενοστραφής και οικεία
- να είναι ανθεκτική και ασφαλής
- να είναι ουδέτερη στην αρχιτεκτονική και φορητή.
- να εκτελείται με υψηλή απόδοση
- να διερμηνεύεται, να νηματοποιείται και να είναι δυναμική

## ➤ Σύστημα εκτέλεσης

- Java JVM και Bytecode

Ένας σχεδιαστικός στόχος της Java είναι η φορητότητα, πράγμα που σημαίνει ότι τα προγράμματα που έχουν γραφτεί για την πλατφόρμα αυτή πρέπει να λειτουργούν ομοίως σε οποιοδήποτε συνδυασμό υλικού και λειτουργικού συστήματος με επαρκή υποστήριξη χρόνου εκτέλεσης. Αυτό επιτυγχάνεται με την κατάρτιση του κώδικα γλώσσας Java σε μια ενδιάμεση αναπαράσταση που ονομάζεται Javabytecode, αντί για τον κώδικα μηχανής που αφορά συγκεκριμένα την αρχιτεκτονική. Οι οδηγίες Javabytecode είναι ανάλογες με τον κώδικα του μηχανήματος, αλλά προορίζονται να εκτελούνται από μια εικονική μηχανή (VirtualMachine - VM) γραμμένη ειδικά για το υλικό του κεντρικού υπολογιστή. Οι τελικοί χρήστες χρησιμοποιούν συνήθως ένα περιβάλλον εκτέλεσης Java (JavaRuntimeEnvironment - JRE) εγκατεστημένο στο δικό τους μηχάνημα για αυτόνομες εφαρμογές Java ή σε ένα πρόγραμμα περιήγησης ιστού για εφαρμογές Java. Οι τυπικές βιβλιοθήκες παρέχουν έναν γενικό τρόπο για την πρόσβαση σε λειτουργίες συγκεκριμένου κεντρικού υπολογιστή, όπως γραφικά, νήματα και δικτύωση. Η χρήση του καθολικού bytecode καθιστά εύκολη τη μεταφορά. Ωστόσο, η επιβάρυνση της διερμηνείας του bytecode σε οδηγίες μηχανής έκαναν τα διερμηνευμένα προγράμματα σχεδόν πάντα να τρέχουν πιο αργά από τα εγγενή εκτελέσιμα. Οι μεταγλωττιστές Just-in-Time (JIT) που μεταγλωττίζουν τα bytecodes σε κώδικα μηχανής κατά τη διάρκεια του χρόνου εκτέλεσης εισήχθησαν σε μεταγενέστερο στάδιο. Η ίδια η Java είναι ανεξάρτητη από πλατφόρμα και είναι προσαρμοσμένη στη συγκεκριμένη πλατφόρμα που πρόκειται να εκτελέσει μια εικονική μηχανή Java γι' αυτήν, η οποία μεταφράζει τον Javabytecode στη γλώσσα της μηχανής της πλατφόρμας.

- **Απόδοση**

Τα προγράμματα γραμμένα σε Java έχουν τη φήμη ότι είναι πιο αργά και απαιτούν περισσότερη μνήμη από αυτά που γράφονται στην C ++. Ωστόσο, η ταχύτητα εκτέλεσης των προγραμμάτων Java βελτιώθηκε σημαντικά με την εισαγωγή των Just-in-Time το 1997/1998 για την Java 1.1. , την προσθήκη χαρακτηριστικών γλώσσας που υποστηρίζουν καλύτερη ανάλυση κώδικα ( όπως την κλάση StringBuilder, τις εσωτερικές κλάσεις, τις προαιρετικούς ισχυρισμούς κ.λπ.) καθώς και με την βελτιστοποίηση στην εικονική μηχανή Java. Με την Java 1.5, η απόδοση βελτιώθηκε με την προσθήκη του πακέτου java.util.concurrent συμπεριλαμβανομένης των υλοποιήσιμων χωρίς κλειδαριές(lockfree) των ConcurrentMaps και άλλων συλλογών πολλαπλών πυρήνων, και βελτιώθηκε περαιτέρω με την Java 1.6.

- **Αυτόματη διαχείριση μνήμης**

Η Java χρησιμοποιεί έναν αυτόματο συλλέκτη σκουπιδιών(garbagecollector) για τη διαχείριση μνήμης στον κύκλο ζωής του προγράμματος. Ο προγραμματιστής καθορίζει πότε δημιουργούνται τα αντικείμενα και ο χρόνος εκτέλεσης Java είναι υπεύθυνος για την ανάκτηση της μνήμης μόλις τα αντικείμενα δεν χρησιμοποιούνται πλέον. Μόλις δεν παραμένουν αναφορές σε ένα αντικείμενο, η μνήμη που κατανέμεται δυναμικά (unreachablenmemory) καθίσταται επιλέξιμη για να ελευθερωθεί αυτόματα από τον συλλέκτη απορριμμάτων. Καταλαβαίνουμε την σημαντικότητα του συλλέκτη σκουπιδιών αν αναλογιστούμε ότι κάτι παρόμοιο με διαρροή μνήμης μπορεί να συμβεί αν ο κώδικας ενός προγραμματιστή κρατά μια αναφορά σε ένα αντικείμενο που δεν χρειάζεται πλέον, συνήθως όταν αντικείμενα που δεν χρειάζονται πλέον αποθηκεύονται σε θέσεις που είναι ακόμα σε χρήση. Εάν κληθούν μέθοδοι για ανύπαρκτο αντικείμενο, προκύπτει μια 'εξαιρέση μηδενικού δείκτη' («nullpointerexception»).

Μια από τις ιδέες πίσω από το μοντέλο αυτόματης διαχείρισης μνήμης της Java είναι ότι οι προγραμματιστές μπορούν να απαλλαγούν από την επιβάρυνση που απαιτείται για τη χειρωνακτική διαχείριση της μνήμης. Σε ορισμένες γλώσσες, η μνήμη για τη δημιουργία αντικειμένων κατανέμεται στη στοίβα ή εκχωρείται και καταργείται από τον σωρό . Στην τελευταία περίπτωση, η ευθύνη της διαχείρισης της μνήμης βρίσκεται στον προγραμματιστή. Εάν το πρόγραμμα δεν απομακρύνει ένα αντικείμενο, εμφανίζεται διαρροή μνήμης . Εάν το πρόγραμμα προσπαθεί να αποκτήσει πρόσβαση ή να απομακρύνει μνήμη που έχει ήδη αφαιρεθεί, το αποτέλεσμα είναι απροσδιόριστο και



δύσκολο να προβλεφθεί και το πρόγραμμα είναι πιθανό να γίνει ασταθές ή να καταρρεύσει. Αυτό μπορεί να αντιμετωπιστεί εν μέρει με τη χρήση έξυπνων δεικτών, οι οποίοι όμως προσθέτουν επιβάρυνση και πολυπλοκότητα. Σημειώστε ότι η συλλογή απορριμμάτων δεν εμποδίζει τη διαρροή "λογικής" μνήμης, δηλαδή εκείνες όπου η μνήμη εξακολουθεί να αναφέρεται, αλλά δεν χρησιμοποιείται ποτέ.

Η συλλογή σκουπιδιών μπορεί να συμβεί ανά πάσα στιγμή. Στην ιδανική περίπτωση, θα συμβεί όταν ένα πρόγραμμα είναι αδρανές. Είναι εγγυημένη η ενεργοποίησή του εάν δεν υπάρχει αρκετή ελεύθερη μνήμη στον σωρό για να φιλοξενήσει ένα νέο αντικείμενο. Αυτό μπορεί να προκαλέσει προσωρινή διακοπή ενός προγράμματος. Δεν είναι δυνατή η διαχείριση ρητής μνήμης στην Java.

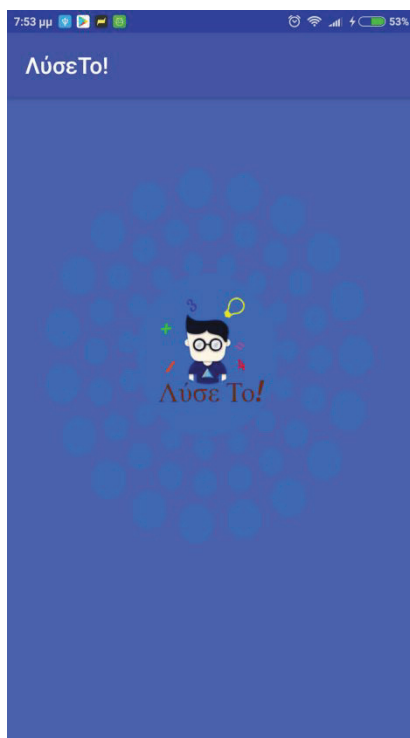
Η Java δεν υποστηρίζει την αριθμητική δείκτη όπως συμβαίνει στη C / C ++, όπου οι διευθύνσεις αντικειμένων και οι μη υπογεγραμμένοι ακέραιοι (συνήθως μεγάλοι ακέραιοι) μπορούν να χρησιμοποιηθούν εναλλακτικά. Αυτό επιτρέπει στον συλλέκτη απορριμμάτων να μεταφέρει τα αναφερόμενα αντικείμενα και εξασφαλίζει ασφάλεια και ασφάλεια τύπου (type safety).

Η Java περιέχει πολλούς τύπους συλλεκτών σκουπιδιών. Από προεπιλογή, χρησιμοποιείται ο παράλληλος συλλέκτης απορριμμάτων. Ωστόσο, υπάρχουν επίσης αρκετοί άλλοι συλλέκτες απορριμμάτων που μπορούν να χρησιμοποιηθούν για τη διαχείριση του σωρού.

## 7.5 Παρουσίαση Εφαρμογής «Λύσε Το»

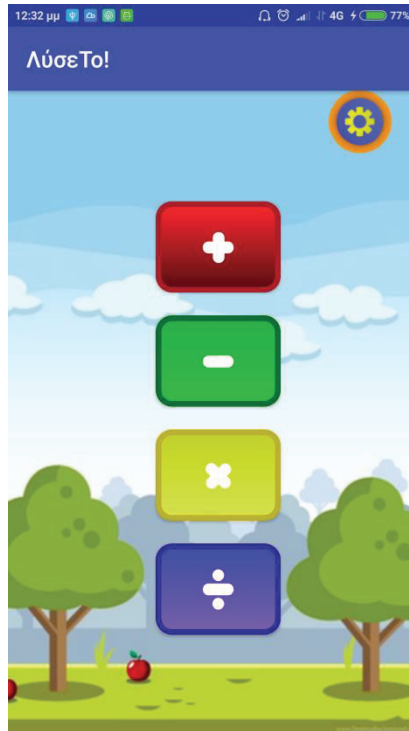
Η εφαρμογή που επιλέξαμε να αναπτύξουμε σχετίζεται με την εκμάθηση μαθηματικών για μαθητές δημοτικού. Την ονομάσαμε «Λύσε Το». Πρόκειται για μια διαδραστική εφαρμογή η οποία αποτελεί ουσιαστικά ένα παιχνίδι κατηγορίας quiz , με το οποίο ένα παιδί μπορεί να εξασκήσει τις δυνατότητες του στα μαθηματικά με έναν πιο ξέγνοιαστο και διασκεδαστικό τρόπο.

Έχοντας λοιπόν εξηγήσει τις βασικές τεχνικές ανάπτυξης της εφαρμογής μας μπορούμε να συνεχίσουμε στην παρουσίαση της. Αρχικά , κατά το φόρτωμα της εφαρμογής εμφανίζεται στην οθόνη του χρήστη το λογότυπο της.



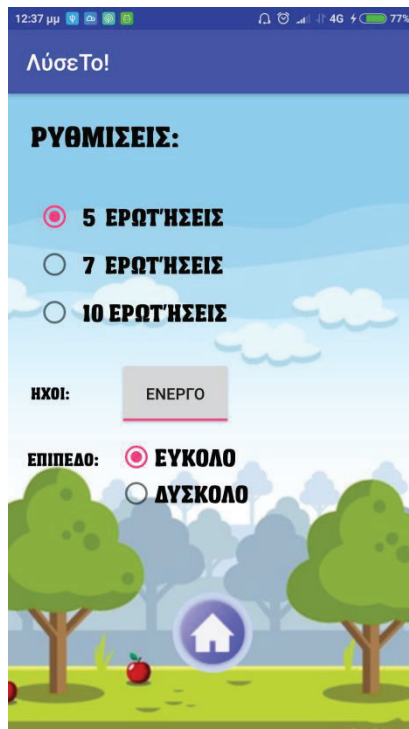
Εικόνα 7.5.1: Το λογότυπο

Μετά από λίγο έρχεται στην οθόνη το κεντρικό μενού από το οποίο ο χρήστης μπορεί να επιλέξει με ποια μαθηματική πράξη θέλει να παίξει. Η εφαρμογή στην τωρινή της έκδοση περιλαμβάνει τις τέσσερις βασικές πράξεις (πρόσθεση, αφαίρεση, πολλαπλασιασμό, διαίρεση). Στην οθόνη αυτή (βλ. Εικόνα 7.5.2) υπάρχουν τέσσερα κουμπιά (ένα για κάθε πράξη) τα οποία είναι τοποθετημένα σε κάθετη στοίχιση στο κέντρο της οθόνης. Αυτά εμφανίζονται με ένα εφέ κίνησης (animation) ώστε να προσδώσει ένα πιο ελκυστικό χαρακτήρα στην εφαρμογή μας.



Εικόνα 7.5.2: Κεντρικό Μενού

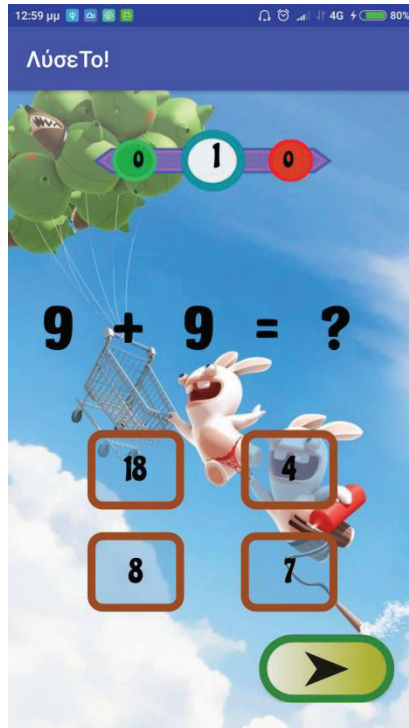
Εκτός από τα παραπάνω ο χρήστης μπορεί να κάνει και κάποιες ρυθμίσεις σχετικά με το παιχνίδι. Πιο συγκεκριμένα, επιλέγοντας ένα στρογγυλό κουμπί που βρίσκεται στο πάνω δεξιό τμήμα της οθόνης πλοηγείται σε μία νέα οθόνη όπου μπορεί να επιλέξει πόσες ερωτήσεις θα περιλαμβάνει κάθε παιχνίδι(βλ. εικόνα 7.5.3).



Εικόνα 7.5.12: Ρυθμίσεις

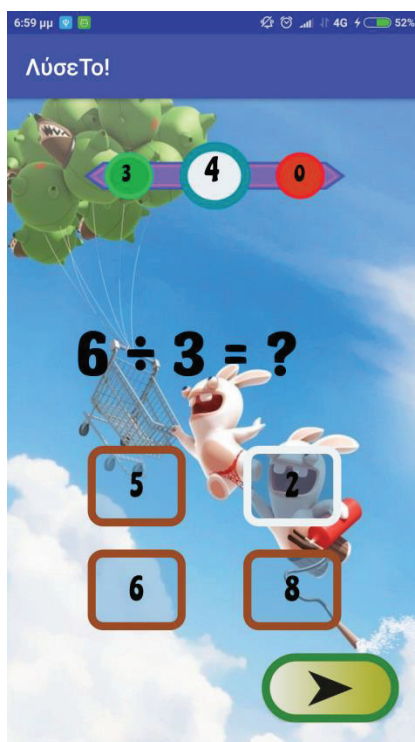
Όταν ο χρήστης επιλέξει μια από τις προεπιλογές που διαθέτει η εφαρμογή για τον αριθμό των ερωτήσεων (πέντε, επτά ή δέκα) που θα παίζει θα πρέπει να πατήσει το κουμπί που υπάρχει στην οθόνη (το ίδιο το συναντάμε και σε οθόνη που εμφανίζεται σε παρακάτω στάδιο της εφαρμογής) προκειμένου να επιστρέψει στο κεντρικό μενού από όπου στην συνέχεια θα αποφασίσει με ποια μαθηματική πράξη θα συνεχίσει. Αξίζει να σημειωθεί ότι αν ο χρήστης δεν επιλέξει τον αριθμό των ερωτήσεων, το παιχνίδι θα παιχτεί με πέντε ερωτήσεις. Επίσης, στην οθόνη των ρυθμίσεων ο χρήστης έχει την δυνατότητα να επιλέξει αν οι ήχοι της εφαρμογής καθώς και το επίπεδο του παιχνιδιού (εύκολο ή δύσκολο). Προκειμένου οι επιλογές του χρήστη να εφαρμοστούν πρέπει να πατήσει το κουμπί που βρίσκεται στο κάτω μέρος της οθόνης με το οποίο επιστρέφει ξανά στο κεντρικό μενού.

Αφού πραγματοποιηθεί η επιλογή μιας μαθηματικής πράξης το παιχνίδι ξεκινάει. Στην νέα οθόνη(βλ. Εικόνα 7.5.4)που εμφανίζεται υπάρχουν 3 βασικά μέρη. Αρχικά στο πάνω μέρος εμφανίζεται μια μπάρα στην οποία ο χρήστης βλέπει πληροφορίες σχετικά με την κατάσταση του παιχνιδιού καθώς και με την πορεία της επίδοσης του σε αυτό. Πιο συγκεκριμένα, στο κέντρο της μπάρας και σε άσπρο κύκλο εμφανίζεται ο τρέχον αριθμός της ερώτησης. Δεξιά και σε κόκκινο κύκλο αναγράφεται το πλήθος των λανθασμένων απαντήσεων και αριστερά σε πράσινο κύκλο ο αριθμός των σωστών απαντήσεων. Λίγο πιο χαμηλά, στο μέσο της οθόνης εμφανίζεται η πράξη που του ζητείτε να συμπληρώσει το αποτέλεσμα της. Στο κάτω μέρος της οθόνης εμφανίζονται τέσσερα κουμπιά σε τετράγωνη παράθεση στα οποία περιέχονται τέσσερις πιθανές απαντήσεις (μία στο καθένα) από τις οποίες μπορεί να επιλέξει την απάντηση του. Πέραν αυτών, στο κάτω μέρος υπάρχει και ένα επιπλέον κουμπί το οποίο μεταφέρει τον χρήστη στην επομένη ερώτηση.



Εικόνα 7.5.13: Αρχική κατάσταση του παιχνιδιού

Όταν ο χρήστης επιλέξει μια απάντηση, το περίγραμμα του κουμπιού στο οποίο περιέχεται εκείνη αλλάζει χρώμα (γίνεται λευκό). Σε περίπτωση που αλλάξει την επιλογή του τότε το περίγραμμα του νέου κουμπιού που επιλέχθηκε γίνεται εκείνο λευκό και το προηγούμενο επανέρχεται στην προηγούμενη (κανονική) κατάσταση (βλ. Εικόνα 7.5.5).



Εικόνα 7.5.14: Επιλογή απάντησης

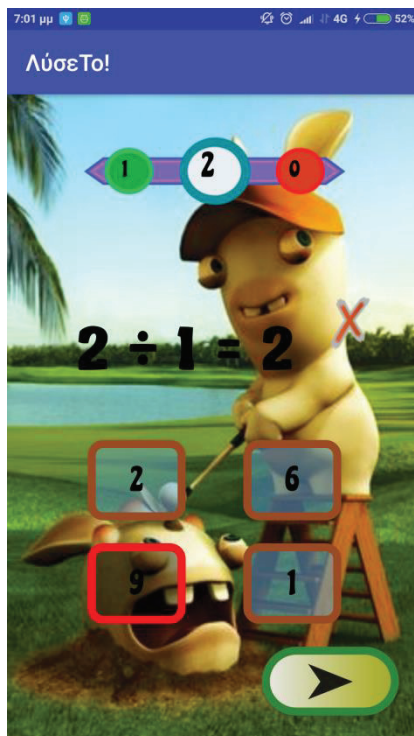
Στην συνέχεια εάν πατήσει το κουμπί «επόμενο» (είναι το κουμπί που βρίσκεται στο κάτω δεξιά τμήμα της οθόνης και έχει ένα βέλος με δεξιά κατεύθυνση) εμφανίζονται στην οθόνη τα αποτελέσματα της απάντησης του.

Μόλις ο χρήστης πατήσει το κουμπί «επόμενο» όλα τα κουμπιά εκτός εκείνου που επέλεξε απενεργοποιούνται. Μετά, το αποτέλεσμα της πράξης το οποίο μέχρι εκείνη τη στιγμή υπάρχει με λατινικό ερωτηματικό αφού εκτελέσει ένα εφέ κίνησης εμφανίζει τον αριθμό της απάντησης. Αναλόγως αν η απάντηση είναι σωστή ή λάθος εμφανίζονται τα ανάλογα αποτελέσματα. Εάν η απάντηση είναι σωστή τότε το περίγραμμα του κουμπιού που επιλέχθηκε γίνεται πράσινο και δίπλα στο αποτέλεσμα εμφανίζεται ένα checkmark (βλ. Εικόνα 7.5.6).



Εικόνα 7.5.15 Η εμφάνιση της εφαρμογής μετά από σωστή απάντηση

Εάν η απάντηση είναι λάθος τότε το περίγραμμα του επιλεγμένου κουμπιού γίνεται κόκκινο και δίπλα στην απάντηση εμφανίζεται ένα X (βλ. Εικόνα 7.5.7).



Εικόνα 7.5.16 Η εμφάνιση της εφαρμογής μετά από λάθος απάντηση

Τα παραπάνω αποτελέσματα εμφανίζονται για περιορισμένο χρόνο. Στην συνέχεια εφόσον εκκρεμούν και άλλες ερωτήσεις στο παιχνίδι (δηλαδή δεν βρισκόμασταν στην τελευταία) η ερώτηση ανανεώνεται. Όταν συμβεί αυτό πέραν των νέων αριθμών που εμφανίζονται στη ζητούμενη πράξη, αυξάνεται τόσο ο αριθμός της τρέχουσα ερώτησης , όσο και ο αριθμός των σωστών ή λάθος απαντήσεων του χρήστη (αναλόγως την έκβαση της προηγούμενης ερώτησης ).

Σε διαφορετική περίπτωση εμφανίζεται μια νέα οθόνη στην οποία ο χρήστης ενημερώνεται για τις επιδόσεις του. Αναλόγως το πλήθος των σωστών απαντήσεων που έχει δώσει στην οθόνη εμφανίζεται ένα διαφορετικό μήνυμα καθώς επίσης και μία εικόνα. Αν ο χρήστης έχει απαντήσει σωστά το 70% των ερωτήσεων λαμβάνει συγχαρητήριο μήνυμα (βλ. Εικόνα7.5.8).



Εικόνα 7.5.17: Οθόνη με μήνυμα επιβράβευσης προς τον χρήστη

Σε διαφορετική περίπτωση λαμβάνει μήνυμα παρακίνησης (βλ. Εικόνα7.5.9) Σε αυτή την οθόνη εμφανίζονται επίσης και δυο επιλογές. Η μια είναι το κουμπί «Μενού» (το είδαμε και προηγουμένως στην οθόνη των ρυθμίσεων) όπου ο χρήστης επιστρέφει στο αρχικό μενού όπου μπορεί να επιλέξει μια νέα πράξη για να παίξει. Η άλλη είναι το «παίξεξανά» (στρογγυλό βέλος που γυρνάει προς τα πίσω) όπου το παιχνίδι ξεκάνει πάλι με την ίδια μαθηματική πράξη που είχε και το προηγούμενο.

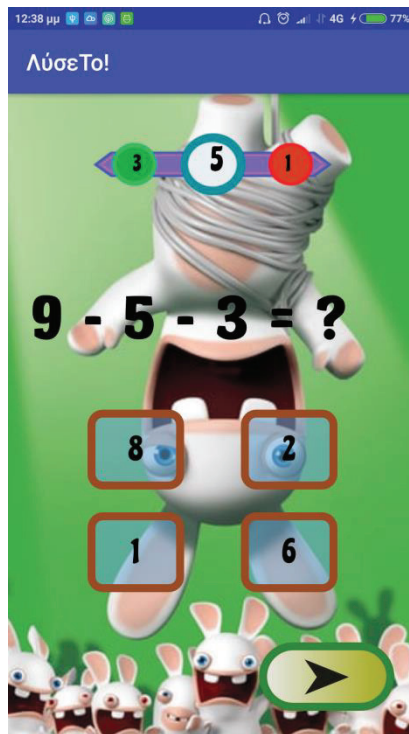




Εικόνα 7.5.18 Οθόνη με μήνυμα παρακίνησης προς τον χρήστη

Αξίζει να σημειωθεί ότι μέσα στην κεντρική οθόνη του παιχνιδιού και ενώ εκείνο βρίσκεται σε εξέλιξη, όταν η ερώτηση ανανεώνεται η εικόνα που βρίσκεται στο background της οθόνης μεταβάλλεται έτσι ώστε το παιχνίδι να γίνεται πιο ζωντανό και να αυξήσει το ενδιαφέρον του χρήστη.

Όπως αναφέραμε στην οθόνη των ρυθμίσεων η εφαρμογή υποστηρίζει δύσκολο επίπεδο. Σε αυτό το επίπεδο η μαθηματική πράξη που καλείται να απαντήσει ο χρήστης αποτελείται από τρεις αριθμούς (βλ. Εικόνα 7.5.10). Η συγκεκριμένη επιλογή επιπέδου αφορά μόνο τις πράξεις της πρόσθεσης και της αφαίρεσης.



Εικόνα 7.5.10: Το παιχνίδι στο δύσκολο επίπεδο

## 7.6 Ανάλυση Τμημάτων Πηγαίου Κώδικα

Στο προηγούμενο κεφάλαιο περιγράψαμε την λειτουργία της εφαρμογής και τα επιμέρους μέρη της. Στο τρέχον κεφάλαιο επιλέξαμε να περιγράψουμε και να αναλύσουμε κάποια σημαντικά μέρη της εφαρμογής μας, περιλαμβάνοντας και τον πηγαίο κώδικα τους. Αρχικά θα παρουσιάσουμε το κώδικα των κλάσεων που συνθέτουν το μοντέλο της εφαρμογής μας (περιγράφεται στο κεφάλαιο 7.2). Στη συνέχεια θα παρουσιάσουμε ορισμένα Activities τα οποία αποτελούν και τις οθόνες της εφαρμογής μας μαζί με τα Layouts αυτών τα οποία είναι υπεύθυνα για την παράθεση των στοιχείων (ImageView, TextView, Button κλπ) που περιέχει κάθε Activity.

## ➤ ΚΛΑΣΕΙΣ ΜΟΝΤΕΛΟΥ

Όπως περιγράφηκε και στο κεφάλαιο 7.2 το μοντέλο μας αποτελείται από τρεις κλάσεις την κλάση Question , την κλάση Quiz και την κλάση Game. Παρακάτω ακολουθεί ο κώδικάς τους:

Πίνακας 7.6.1: Κώδικας Μοντέλου (Μέρος Α)

```
publicclass Question
{
int number_1;
int number_2;
    int number_3;
charmath_operation;
int correct;
int answer_1;
int answer_2;
int answer_3;
inta_random_number;

    Random rand =new Random();
    List<Integer>numb_list;

public Question(charselected_math_operation)
{
numb_list=newArrayList<Integer>();
math_operation=selected_math_operation;

do
{
    number_1 =rand.nextInt(9)+1;
    number_2 =rand.nextInt(9)+1;
```

```
}while(!Check_if_MathOperation_Can_beExecuted());
```

```
switch(selected_math_operation)
```

```
{
```

```
case'+':
```

```
    correct = number_1 + number_2;
```

```
break;
```

```
case'-':
```

```
    correct = number_1 - number_2;
```

```
break;
```

```
case'/':
```

```
    correct = number_1 / number_2;
```

```
break;
```

```
case'*':
```

```
    correct = number_1 * number_2;
```

```
break;
```

```
default:
```

```
break;
```

```
}
```

```
Generate_three_numbers();
```

```
    answer_1 = numb_list.get(0);
```

```
    answer_2 = numb_list.get(1);
```

```
    answer_3 = numb_list.get(2);
```

```

}

public Question(charselected_math_operation, Boolean is_difficult)

{
numb_list=newArrayList<Integer>(0);
math_operation=selected_math_operation;

do
{
    number_1 =rand.nextInt(3)+7;
    number_2 =rand.nextInt(3)+4;
    number_3 =rand.nextInt(3)+1;
}while(!Check_if_Pro_MathOperation_Can_beExecuted());

switch(selected_math_operation)
{
case'+':
System.out.println("Addition");
    correct = number_1 + number_2 + number_3;
break;

case'-':
System.out.println("Subtraction");
    correct = number_1 - number_2 - number_3;
break;

default:
System.out.println("Default case");
break;
}

Generate three numbers pro();

    answer_1 = numb_list.get(0);
    answer_2 = numb_list.get(1);
    answer_3 = numb_list.get(2);
}

privatebooleanCheck_if_MathOperation_Can_beExecuted()

{

if(math_operation=='/')

{

if(number_1>=number_2 && number_1%number_2 ==0)

{

returntrue;

}

else

```

```
{  
returnfalse;  
}  
}  
  
if(math_operation=='-')  
{  
if(number_1>=number_2)  
{  
returntrue;  
}  
else  
{  
returnfalse;  
}  
}  
  
if(math_operation=='+')  
{  
returntrue;  
}  
  
if(math_operation=='*')  
{  
returntrue;  
}  
  
//Not necessary  
returntrue;
```

```
}

private void Generate_three_numbers()
{
    for(int i=0; i<3; i++)
    {
        do
        {
            a_random_number=rand.nextInt(9)+1;

        } while(num_list.contains(a_random_number)||a_random_number== correct);

        num_list.add(a_random_number);
    }
}

public boolean CheckAnswer(int answer)
{
    if(answer == this.correct)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

```

}

private boolean Check_if_Pro_MathOperation_Can_beExecuted()
{
    int aux_res=0;
    if(math_operation=='-')
    {
        aux_res= number_1 - number_2;
        if(aux_res- number_3 >=0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    else
    {
        return true;
    }
}

private void Generate_three_numbers_pro()
{
    for(int i=0;i<3;i++)
    {
        do
        {
            if(math_operation=='-')
            {
                a_random_number=rand.nextInt(9)+1;
            }
            else
            {
                //If math operation is addition the max potential result will be 27
                a_random_number=rand.nextInt(27)+1;
            }
        }while(num_list.contains(a_random_number)||a_random_number== correct);

        num_list.add(a_random_number);
    }
}
}

```

Στο κώδικα της κλάσης Question αρχικά βλέπουμε τις μεταβλητές της. Στην συνέχεια υπάρχει η μέθοδος δημιουργός της στον οποίο αρχικοποιούμε τις μεταβλητές math\_operation η οποία περιλαμβάνει το σύμβολο της πράξης που θα έχει η πράξη που επέλεξε ο χρήστης να παίξει καθώς επίσης και μία λίστα με τρεις τυχαίους ακεραίους οι οποίοι θα είναι οι πιθανές απαντήσεις που θα φαίνονται στο χρήστη (η τέταρτη θα είναι η



σωστή απάντηση). Παρακάτω βρίσκεται ένας βρόγχος επανάληψης στον οποίο επιλέγουμε δύο τυχαίους αριθμούς που θα είναι οι βασικοί αριθμοί της πράξης. Η επανάληψη χρησιμοποιείται ώστε να εκτελούμε αυτή την επιλογή μέχρι αυτοί οι δύο αριθμοί να μπορούν να εκτελέσουν την μαθηματική πράξη (π.χ. αν ο πρώτος είναι μικρότερος από τον δεύτερο και πρέπει να αφαιρεθούν τότε το αποτέλεσμα θα είναι αρνητικό, επομένως θα πρέπει να ξανά επιλέξουμε αριθμούς). Η μέθοδος που κάνει τον έλεγχο εγκυρότητας της πράξης είναι η `Check_if_MathOperation_Can_beExecuted`. Αμέσως μετά αναλόγως την μαθηματική πράξη που επιλέχθηκε εκχωρούμε το αποτέλεσμα της στην μεταβλητή `correct` και μετά καλούμε την μέθοδο `Generate_three_numbers` η οποία θα παράξει τους τρεις τυχαίους αριθμούς που αναφέραμε παραπάνω και θα τους εισάγει στη λίστα `numb_list` από την οποία θα τους εισάγουμε στις μεταβλητές `answer_1`, `answer_2`, `answer_3`. Ο κώδικας της κλάσης περιλαμβάνει μία ακόμη μέθοδο, την `CheckAnswer` με την οποία ελέγχουμε το αποτέλεσμα της επιλογής που θα κάνει ο χρήστης. Στο τέλος του κώδικα μας περιλαμβάνονται οι μέθοδοι όπου χρησιμοποιούνται για να εκτελέσουν τις ίδιες ενέργειες με εκείνες των συνονόματων τους αλλά στην περίπτωση που ο χρήστης επιλέξει να παίξει το δύσκολο παιχνίδι (`Check_if_Pro_MathOperation_Can_beExecuted`, `Generate_three_numbers_Pro`).

**Πίνακας 7.6.2: Κώδικας Μοντέλου (Μέρος Β)**

```
publicclass Quiz
{
    intcounter_corrects;
    intcounter_wongs;
    inttotal_questions;

    List<Question>questions_list;

    public Quiz(intnumber_of_questions,charselcted_math_operation)
    {
        counter_corrects=0;
```

```

counter_wongs=0;

this.total_questions=number_of_questions;
questions_list=newArrayList<Question>();

Fill_Questions_List(selected_math_operation);
}

privatevoidFill_Questions_List(charselected_math_operation)
{
for(inti=0;i<total_questions;i++)
{
    Question quest =new Question(selected_math_operation);
questions_list.add(quest);
}
}
}
}

```

Στον κώδικα της κλάσης Quiz βλέπουμε και εδώ τις σχετικές μεταβλητές τις οποίες χρειαζόμαστε προκειμένου να δημιουργήσουμε ένα σετ από ερωτήσεις. Στη μέθοδο δημιουργό της κλάσης αρχικοποιούμε τους μετρητές counter\_corrects και counter\_wongs οι οποίοι διατηρούν το πλήθος των σωστών και λάθος απαντήσεων του χρήστη αντίστοιχα. Στην συνέχεια αρχικοποιούμε και γεμίζουμε τη λίστα questions\_list η οποία περιλαμβάνει αντικείμενα τύπου Question. Αυτό επιτυγχάνεται με την κλήση της μεθόδου Fill\_Questions\_List η οποία δημιουργεί τόσα αντικείμενα Question όσο το περιεχόμενο της μεταβλητής total\_questions η οποία έχει το πλήθος των ερωτήσεων που έχει επιλέξει ο χρήστης να παίζει.

Πίνακας 7.6.3: Κώδικας Μοντέλου (Μέρος Γ)

```
publicclass Game
{
    Quiz current_quiz;

    public Game(intnumber_of_questions,charselected_math_operation)
    {
        current_quiz=new Quiz(number_of_questions,selected_math_operation);
    }
}
```

Η τελευταία κλάση του μοντέλου είναι η Game. Αυτή είναι πιο συνοπτική και περιέχει μόνο την μεταβλητή current\_quiz που είναι τύπου Quiz. Μέσα στη μέθοδο δημιουργό της αρχικοποιούμε την μεταβλητή αυτή ώστε να δημιουργήσουμε ένα νέο quiz το οποίο θα περιέχει τις ερωτήσεις του παιχνιδιού.

#### ➤ ΑΡΧΕΙΟ LAYOYT ΤΗΣ ΚΥΡΙΑΣ ΟΘΟΝΗΣ

Στην προηγούμενη ενότητα επιλέξαμε να παραθέσουμε και να εξηγήσουμε τον πηγαίο κώδικα του μοντέλου στο οποίο στηρίξαμε την εφαρμογή μας. Σε αυτή την ενότητα θα παρουσιάσουμε τα βασικά μέρη της κύριας οθόνης της εφαρμογής. Πρόκειται για την MainActivity η οποία περιέχει την μαθηματική πράξη που βλέπει ο χρήστης, τις επιλογές που έχει προκειμένου να την απαντήσει καθώς και πληροφορίες σχετικά με την επίδοση του χρήστη (βλ ενότητα 7.5.4).

Κάθε activity συνδέεται με ένα αρχείο του φακέλου layout(που περιέχει κώδικα xml) το οποίο περιλαμβάνει την δομή και την παράθεση των στοιχείων του activity(buttons, textviews, imageviewsκλπ). Το αρχείο αυτό ουσιαστικά είναι υπεύθυνο για την εμφάνιση που έχει κάθε στοιχείο καθώς και τη θέση που λαμβάνει στην οθόνη. Κάθε στοιχείο ή ομάδα στοιχείων της οθόνης πρέπει να περιέχεται μέσα σε ένα layout το οποίο καθορίζει τη διάταξη του στην οθόνη. Υπάρχουν διάφορα είδη layout όπως το LinearLayout το οποίο παραθέτει τα στοιχεία σε σειρά, είτε κάθετα είτε οριζόντια, το FrameLayout το

οποίο τοποθετεί τα στοιχεία σε διαφορετικά επίπεδα (το ένα πάνω από το άλλο) , το RelativeLayout το οποίο συνδέει τα στοιχεία με μία πιο συγκεκριμένη συσχέτιση μεταξύ τους. Αρχικά, το αρχείο μας έχει σαν βασικό layout το FrameLayout μέσα στο οποίο περιέχονται όλα τα στοιχεία.

Πίνακας 7.6.4: Κώδικας του Layout της κύριας οθόνης (μέρος Α)

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".NewMainActivity">
```

Στο παραπάνω κώδικα δηλώνεται η επιλογή του βασικού layout καθώς και το μέγεθος του σε σχέση με την οθόνη. Με την τιμή "match\_parent" δηλώνουμε ότι το ύψος και το πλάτος του layout θα γίνει ίσο με το περιβάλλον στο οποίο περιέχεται( εδώ επειδή πρόκειται για το βασικό layout , το περιβάλλον στο οποίο περιέχεται είναι η οθόνη). Στην συνέχεια τοποθετούμε τα στοιχεία που έχουμε αποφασίσει να εμφανίζονται στην οθόνη. Το πρώτο που εισάγουμε είναι ένα imageView το οποίο θα παίζει το ρόλο του background της οθόνης.

Πίνακας 7.6.5: Κώδικας του Layout της κύριας οθόνης (μέρος Β)

```
<ImageView
android:id="@+id/imageViewBackGround"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:srcCompat="@drawable/rabbits_2"/>
```

Όπως φαίνεται στον παραπάνω κώδικα, κάθε tag ενός στοιχείου περιέχει διάφορες πληροφορίες για το ίδιο. Αρχικά τον τύπο του (εδώ έχουμε ένα imageView). Στην συνέχεια δηλώνουμε ένα μοναδικό αναγνωριστικό κωδικό (id) με το οποίο μπορούμε να

αναφερόμαστε στο συγκεκριμένο στοιχείο. Αμέσως μετά δηλώνουμε τις διαστάσεις του καθώς επίσης και το path από το οποίο το imageView θα πάρει την εικόνα του. Στην συνέχεια του αρχείου χρησιμοποιούμε ένα νέο layout (συγκεκριμένα ένα LinearLayout) το οποίο θα περιέχει όλα τα στοιχεία που θα χρησιμοποιήσουμε. Για να καταλάβουμε πιο εύκολα την βασική ιδέα που χρησιμοποιεί το αρχείο αυτό αρκεί να φανταστούμε τη γνωστή ξύλινη κούκλα Ματριόσκα (πιο γνωστή ως μπαμπούσκα). Το χαρακτηριστικό αυτής είναι ότι κάθε μια κούκλα περιέχει μέσα της μια μικρότερη. Έτσι και εδώ ένα layout περιέχει ένα άλλο layout με διάφορα στοιχεία ή σκέτα στοιχεία. Το εξωτερικό layout είναι αυτό που καθορίζει κάθε φορά την δομή των στοιχείων που περιέχει. Έτσι λοιπόν χρησιμοποιούμε ένα LinearLayout με κάθετο προσανατολισμό των περιεχομένων του , τα οποία είναι πέντε ακόμα layout τα οποία με την σειρά τους περιλαμβάνουν στοιχεία.

**Πίνακας 7.6.6: Κώδικας του Layout της κύριας οθόνης (μέρος Γ)**

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">
```

Όπως φαίνεται στον κώδικα μας και εδώ καθορίζουμε τις διαστάσεις του καθώς και τον τύπο του προσανατολισμού(κάθετος). Το πρώτο layout που περιέχει είναι ένα FrameLayout το οποίο περιέχει ένα imageView και ένα LinearLayout με τρία textview. Το imageView περιέχει την εικόνα της μπάρας που περιέχει πληροφορίες των τριών textview για το πόσα σωστά και λάθη έχει ο χρήστης καθώς και τον αριθμό της τρέχουσας ερώτησης.

**Πίνακας 7.6.7: Κώδικας του Layout της κύριας οθόνης (μέρος Δ)**

```
<FrameLayout  
    android:layout_width="match_parent"  
    android:layout_height="152dp">  
  
<ImageView
```

```
android:id="@+id/imageView7"  
android:layout_width="wrap_content"  
android:layout_height="124dp"  
app:srcCompat="@drawable/current_status_bar"/>
```

```
<LinearLayout
```

```
android:layout_width="match_parent"  
android:layout_height="139dp"  
android:orientation="horizontal"  
android:padding="40dp">
```

```
<TextView
```

```
android:id="@+id/textViewCorrects"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
  
android:layout_marginLeft="70dp"  
android:fontFamily="@font/chicle_regular"  
android:text="1"  
android:textColor="#000000"  
android:textSize="20dp"/>
```

```
<TextView
```

```
android:id="@+id/textViewQuestNumb"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
  
android:layout_marginLeft="61dp"  
android:fontFamily="@font/chicle_regular"
```

```

android:text="5"

android:textColor="#000000"

android:textSize="30dp"/>

<TextView

android:id="@+id/textViewWrongs"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_marginLeft="55dp"

android:fontFamily="@font/chicle_regular"

android:text="2"

android:textColor="#000000"

android:textSize="20dp"/>

</LinearLayout>

</FrameLayout>

```

Στο παραπάνω κώδικα φαίνεται το layout που περιγράψαμε. Σχετικά με το imageView ισχύουν οι ίδιες ιδιότητες(μέγεθος, πηγή εικόνας) που είδαμε και προηγουμένως. Σε ότι αφορά τα textView και εδώ χρησιμοποιούμε διάφορες ιδιότητες για να τα καθορίσουμε. Πέραν των βασικών (μοναδικό id, μέγεθος) χρησιμοποιούμε και κάποιες επιπλέον ιδιότητες. Προκειμένου να καθορίσουμε την γραμματοσειρά που θα έχει χρησιμοποιούμε το fontFamily στο οποίο αναθέτουμε ένα αρχείο ttf(TrueTypeFonts - αρχεία γραμματοσειράς) το οποίο έχουμε εισάγει στο project μας. Με τα text, textColor και textSize καθορίζουμε το κείμενο που θα εμφανίζεται, το χρώμα του και το μέγεθος του αντίστοιχα. Επίσης χρησιμοποιούμε και την layout\_marginLeft η οποία καθορίζει το κενό που θα έχει το textView σε σχέση με ότι υπάρχει αριστερά του. Το δεύτερο layout που περιέχει το LinearLayout είναι ένα άλλο LinearLayout με οριζόντιο προσανατολισμό το οποίο περιέχει πέντε textView και ένα imageView. Τα textView αντιστοιχούν στην μαθηματική πράξη που βλέπει ο χρήστης. Πιο συγκεκριμένα τα textView αντιστοιχούν σε τρεις αριθμούς (οι δυο που αποτελούν την κύρια πράξη και ο ένας που αποτελεί το

αποτέλεσμα) και σε δύο σύμβολα, ένα της πράξης και ένα της ισότητας. Το imageView περιέχει την εικόνα με το σύμβολο του tik ή το x που προβάλλεται δίπλα στην πράξη αναλόγως αν η απάντηση είναι σωστή ή λάθος.

Πίνακας 7.6.8: Κώδικας του Layout της κύριας οθόνης (μέρος E)

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="147dp"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/textViewN1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="30dp"
        android:layout_marginTop="20dp"
        android:layout_weight="1"
        android:fontFamily="@font/garishing_worse"
        android:text="5"
        android:textColor="#000000"
        android:textSize="50dp"/>

    <TextView
        android:id="@+id/textViewSymbol"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="30dp"
        android:layout_marginTop="20dp"
        android:layout_weight="1"
        android:fontFamily="@font/garishing_worse"
```



```

android:text="+
android:textColor="#000000"
android:textSize="50dp"/>

<TextView
android:id="@+id/textViewN2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="30dp"
android:layout_marginTop="20dp"
android:layout_weight="1"
android:fontFamily="@font/garishing_worse"
android:text="1"
android:textColor="#000000"
android:textSize="50dp"/>

<TextView
android:id="@+id/textViewEquality"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="30dp"
android:layout_marginTop="20dp"
android:layout_weight="1"
android:fontFamily="@font/garishing_worse"
android:text=""
android:textColor="#000000"
android:textSize="50dp"/>

<TextView

```

```

android:id="@+id/textViewAnswer"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="30dp"
android:layout_marginTop="20dp"
android:layout_weight="1"
android:fontFamily="@font/garishing_worse"
android:text="6"
android:textColor="#000000"
android:textSize="50dp"/>

<ImageView
android:id="@+id/imageViewMark"
android:layout_width="50dp"
android:layout_height="70dp"
android:layout_weight="1"
android:scaleX="2"
android:scaleY="2"
app:srcCompat="@drawable/check_mark_3"/>

</LinearLayout>

```

Όπως και προηγουμένως έτσι και εδώ χρησιμοποιούμε διάφορες ιδιότητες. Σε σχέση με τις υπόλοιπες που είδαμε εδώ χρησιμοποιούμε επιπλέον την `layout_marginTop` η οποία ρυθμίζει το κενό που θα έχει ένα στοιχείο με ότι βρίσκεται από πάνω του και τις `scaleX`, `scaleY` οι οποίες κλιμακώνουν το μέγεθος της ανάλυσης της εικόνας ως προς τους άξονες  $x$  και  $y$  αντίστοιχα. Τα υπόλοιπα τρία layouts του αρχικού `LinearLayout` είναι τρία ακόμα `LinearLayout`. Τα δυο από αυτά περιέχουν από δύο κουμπιά το καθένα σε οριζόντια παράθεση και το τρίτο ένα κουμπί.

Πίνακας 7.6.9: Κώδικας του Layout της κύριας οθόνης (μέρος ΣΤ)

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="50dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/buttonTopLeft"
        android:layout_width="100dp"
        android:layout_height="70dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="50dp"
        android:background="@drawable/button_customization"
        android:layout_weight="1"
        android:fontFamily="@font/chicle_regular"
        android:textSize="30dp"
        android:textColor="#000000"
        android:text="7"/>

    <Button
        android:id="@+id/buttonTopRight"
        android:layout_width="100dp"
        android:layout_height="70dp"
        android:layout_marginRight="20dp"
        android:background="@drawable/button_customization"
        android:layout_weight="1"
        android:fontFamily="@font/chicle_regular"
        android:textSize="30dp"
```

```

android:textColor="#000000"

android:text="6"/>

</LinearLayout>

<LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:layout_marginHorizontal="50dp"

android:layout_marginVertical="20dp"

android:orientation="horizontal">

<Button

android:id="@+id/buttonBottomLeft"

android:layout_width="100dp"

android:layout_height="70dp"

android:layout_marginLeft="20dp"

android:layout_marginRight="50dp"

android:background="@drawable/button_customization"

android:layout_weight="1"

android:fontFamily="@font/chicle_regular"

android:textSize="30dp"

android:textColor="#000000"

android:text="3"/>

<Button

android:id="@+id/buttonBottomRight"

android:layout_width="100dp"

android:layout_height="70dp"

android:layout_marginRight="20dp"

```

```

android:background="@drawable/button_customization"

android:layout_weight="1"

android:fontFamily="@font/chicle_regular"

android:textSize="30dp"

android:textColor="#000000"

android:text="9"/>

</LinearLayout>

<LinearLayout

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_gravity="end"

android:layout_marginRight="20dp"

android:orientation="vertical">

<Button

android:id="@+id/buttonNext"

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:background="@drawable/button_next_shape"

android:textSize="43dp"

android:fontFamily="@font/chicle_regular"

android:text="▶"/>

</LinearLayout>

```

Εδώ παρόλο που έχουν buttons αντί για textviews χρησιμοποιούμε παρόμοιες ιδιότητες. Εκτός από αυτές χρησιμοποιούμε μια επιπλέον που είναι η background η οποία καθορίζει την μορφή που θα έχει το κουμπί. Με τον ίδιο τρόπο με το αρχείο layout που

περιγράψαμε κινούνται και τα υπόλοιπα σχετικά αρχεία που καθορίζουν αναλόγως κάθε ένα activity που περιέχονται στη εφαρμογή μας.

## **7.7 Μελλοντική Ανάπτυξη**

Στην δημιουργία της εφαρμογής μπαίνει μια άνω τελεία. Για τους σκοπούς της εργασίας θεωρήσαμε ότι ήταν αρκετό να χτίσουμε μια καλή βάση στην οποία περιελάβαμε κάποιες από τις γνώσεις που διδαχτήκαμε σαν φοιτητές. Για το μέλλον έχουμε σχεδιάσει κάποια επιπλέον κομμάτια τα οποία επιθυμούμε να προσθέσουμε. Ένα από αυτά είναι ο εμπλουτισμός του βασικού μενού. Πιο συγκεκριμένα θέλουμε να προσθέσουμε κάποια επιπλέον μαθηματικά προβλήματα που θα δυσκολέψουν το παιχνίδι περισσότερο έτσι ώστε να απευθύνονται και σε παιδιά μεγαλύτερης ηλικίας. Επίσης, έχουμε σκεφθεί να προσθέσουμε επιπλέον επιλογές στις ρυθμίσεις. Μια τέτοια είναι να ζητάμε από τον χρήστη το όνομα του με σκοπό να τον προσφωνούμε σε κάποια σημεία του παιχνιδιού. Στόχος μας είναι να εξελίξουμε την εφαρμογή έτσι ώστε να αποκτήσουμε περεταίρω γνώση στο αντικείμενο ,στηριγμένοι στις βάσεις που μας έδωσε η εκπόνηση της παρούσας εργασίας.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1][https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))(Προσπελάστηκε 6/2018)
- [2]<https://developer.android.com/guide/components/fundamentals>(Προσπελάστηκε 6/2018)
- [3]<https://www.quora.com/What-are-the-five-components-of-Android-architecture-made-up>(Προσπελάστηκε 6/2018)
- [4][https://en.wikipedia.org/wiki/Android\\_software\\_development](https://en.wikipedia.org/wiki/Android_software_development)(Προσπελάστηκε 6/2018)
- [5][https://en.wikipedia.org/wiki/Mobile\\_app](https://en.wikipedia.org/wiki/Mobile_app)(Προσπελάστηκε 6/2018)
- [6]<https://www.lifewire.com/what-is-a-mobile-application-2373354>(Προσπελάστηκε 6/2018)
- [7]<https://www.eduportal.gr/mathimatika-mokias/>(Προσπελάστηκε 6/2018)
- [8]<https://www.digitaltrends.com/mobile/android-vs-ios/>(Προσπελάστηκε 6/2018)
- [9]<https://techblog.gr/software/android-vs-apple-vs-windows-phone/>(Προσπελάστηκε 6/2018)
- [10]<https://bl.ocks.org/shifty/0e2516f91a044acfb396>(Προσπελάστηκε 6/2018)
- [11]<https://developer.android.com/studio/intro/>(Προσπελάστηκε 6/2018)
- [12][https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio)(Προσπελάστηκε 6/2018)
- [13]<http://android4beginners.com/2013/06/lesson-0-8-additional-information-about-project-structure-in-android-studio-java-and-resources-folders/>(Προσπελάστηκε 6/2018)
- [14]<https://howtodoinjava.com/android/android-tutorial-android-project-structure-files-and-resources/>(Προσπελάστηκε 6/2018)
- [15][https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))(Προσπελάστηκε 6/2018)
- [16]<https://stackoverflow.com>(Προσπελάστηκε 9/2018)
- [17]<https://www.1001fonts.com>(Προσπελάστηκε 8/2018)
- [18]<https://designapp.io/app>(Προσπελάστηκε 8/2018)
- [19]<https://www.journaldev.com/9481/android-animation-example>(Προσπελάστηκε 8/2018)