ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΠΡΟΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΑΝΑΠΤΥΞΗΣ (IDE) ΓΙΑ ΤΗΝ ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ANDROID

Διπλωματική Εργασία

του

Σπύρου Μουράτη

Βόλος, Οκτώβριος 2015

ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΠΕΡΙΒΑΛΛΟΝΤΩΝ ΑΝΑΠΤΥΞΗΣ (IDE) ΓΙΑ ΤΗΝ ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ANDROID

Σπύρος Μουράτης

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΠΡΟΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΟ ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Επιβλέπων Καθηγητής
Αλκιβιάδης Ακρίτας

Εγκρίθηκε από την εξεταστική επιτροπή την …../…../2015

Αλκιβιάδης Ακρίτας	Δασκαλοπούλου Ασπασία
.....	.....

Σπύρος Μουράτης

.....

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι να παρουσιάσει μία συγκριτική μελέτη ανάμεσα σε δύο Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (Integrated Development Environment - IDE) εφαρμογών για Android συσκευές. Τα περιβάλλοντα αυτά αφορούν την ανάπτυξη και την σύγκριση μιας εφαρμογής σε δύο διαφορετικές προγραμματιστικές γλώσσες την C# και την Java.

Για την ανάπτυξη μιας εφαρμογής σε C# η μόνη επιλογή IDE είναι αυτή της Microsoft με το Visual Studio. Στην συγκεκριμένη περίπτωση έγινε χρήση του Microsoft Visual Studio 2010 Ultimate Edition με χρήση του πρόσθετου Xamarin. Το Xamarin είναι μια εφαρμογή τύπου add-on για τον Visual Studio το οποίο επιτρέπει την υλοποίηση εφαρμογών Android με χρήση της γλώσσας προγραμματισμού C#. Εναλλακτικά προσφέρει και δικό του περιβάλλον ανάπτυξης το Xamarin Studio, με περιορισμένες δυνατότητες για ελεύθερη χρήση. Στην υλοποίηση με χρήση της γλώσσας προγραμματισμού Java οι επιλογές των IDE ήταν περισσότερες αν και στην πραγματικότητα χρησιμοποιούν την ίδια βάση. Πιο συγκεκριμένα, δύο από τα πιο γνωστά IDE για ανάπτυξη εφαρμογής σε Java για Android, είναι το Eclipse με προσθήκη των ανάλογων πακέτων ή το Android Studio το οποίο και χρησιμοποιήθηκε Στην πραγματικότητα το Android Studio είναι μια υλοποίηση του Eclipse από την Google με τα ανάλογα απαιτούμενα για Android πακέτα.

Τα περιβάλλοντα ανάπτυξης επιλέχθηκαν με βασικό κριτήριο το να είναι ελεύθερα προς χρήση. Μόνη εξαίρεση το Visual Studio το οποίο όμως χρησιμοποιήθηκε με MSDNA Licence δηλαδή την Ακαδημαϊκή έκδοση η

οποία παρέχεται δωρεάν στους φοιτητές των Πανεπιστημίων που έχουν την ανάλογη σύμβαση με την Microsoft.

Στα πλαίσια της διπλωματικής δημιουργήθηκε μία εφαρμογή αριθμομηχανής (Calculator) στα δύο παραπάνω IDE που επιλέχθηκαν. Η μία εφαρμογή υλοποιήθηκε σε γλώσσα προγραμματισμού C# και η άλλη σε Java. Στη συνέχεια έγιναν συγκρίσεις σχετικά με την ευκολία προγραμματισμού στα δύο IDE καθώς και στα εργαλεία που προσφέρουν, στην ανάπτυξη κώδικα μεταξύ των δύο γλωσσών προγραμματισμού κλπ και εξήχθησαν σχετικά συμπεράσματα

Λέξεις Κλειδιά: C#, Java, Android, Android Studio, MS Visual Studio, Xamarin, IDE

Abstract

The aim of this thesis is to present a comparative study between two Integrated Development Environments (Integrated Development Environment - IDE) application for Android devices. Environments concern the development and comparison of an application in two different programming languages of C # and Java.

For the development of an application in C # IDE the only option is the Microsoft Visual Studio. In this case it was made use of Microsoft Visual Studio 2010 Ultimate Edition using the additional Xamarin. The Xamarin is an add-on type application for Visual Studio that allows the creation of Android application using the programming language C#. Alternative offers its own development environment Xamarin Studio, with limited possibilities for free use. In the implementation of the programm in which is used the Java programming language options of IDE are far more numerous although they use the same base. Specifically, two of the most popular IDE for application development on Java for Android, is the Eclipse by adding the similar packages or Android Studio which was used. In fact, Android Studio is an implementation of Eclipse by Google analogues required for Android packages.

The development environments were selected with main criterion being free for use. The only exception in Visual Sudio but which used to MSDNA Licence namely the Academic version which is free for students of universities that have the similar contract with Microsoft.

In this thesis an application calculator (Calculator) was created on the two above IDE selected. One application was implemented in C#

programming language and the other in Java. Then comparisons were made on the ease of programming on both IDE and tools offered in code development between the two programming languages, etc. and exported conclusions

Keywords: C#, Java, Android, Android Studio, MS Visual Studio, Xamarin, IDE

Ευχαριστίες

Με την περάτωση της παρούσας εργασίας, θα ήθελα να ευχαριστήσω θερμά τους επιβλέποντες της Διπλωματικής εργασίας Αλκιβιάδη Ακρίτα και Δασκαλοπούλου Ασπασία για την εμπιστοσύνη που επέδειξαν στο πρόσωπό μου.

Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου για την συμπαράσταση και υποστήριξη της καθ' όλη την διάρκεια των φοιτητικών μου χρόνων.

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

1. Εισαγωγή

Σε αυτή την διπλωματική γίνεται σύγκριση σε δύο γλώσσες προγραμματισμού, Java και C#, και σε δύο περιβάλλοντα ανάπτυξης κώδικα, Visual Studio και Android Studio, πάνω στην δημιουργία και εκτέλεση μιας εφαρμογής αριθμομηχανής για Android. Η ίδια εφαρμογή γράφηκε και στις δύο γλώσσες προγραμματισμού. Η εφαρμογή της αριθμομηχανής θα περιλαμβάνει τις βασικές πράξης της πρόσθεσης, αφαίρεσης, πολλαπλασιασμού και διαίρεσης Θα περιλαμβάνει επίσης την δυνατότητα προσθήκης δεκαδικού ψηφίου, της διαγραφής και διόρθωσης του αριθμού που εισάγεται και μηδενισμού του αποτελέσματος.

1.1 Η γλώσσα προγραμματισμού Java

Η γλώσσα προγραμματισμού της Java αρχικά αναπτύχθηκε από την Sun Microsystems υπό την αιγίδα των James Gosling και Bill Joy, σαν μέρος ενός ερευνητικού έργου ανάπτυξης λογισμικού για ηλεκτρονικές συσκευές καταναλωτικού επιπέδου πχ video και τηλεόραση. Ο τρόπος αυτός χρήσης της, την μετέτρεψε σε μία ιδανική γλώσσα για την δημιουργία εκτελέσιμων προγραμμάτων μέσω του Διαδικτύου - Internet καθώς επίσης και σε μια γενικού σκοπού γλώσσα προγραμματισμού για την ανάπτυξη εφαρμογών τα οποία θα είναι εύκολα στην χρήση και θα μπορούν να μεταφέρονται σε διαφορετικά Λειτουργικά Συστήματα με διαφορετικό Υλικό - Hardware. Χρησιμοποιήθηκε από την Sun σε πολλές εφαρμογές με σκοπό την δημιουργία προϊόντων για την ηλεκτρονική αγορά. Ενδιαφέρον προκάλεσε στο κοινό κυρίως όταν συνδυάστηκε με το πρόγραμμα

ανάγνωσης ιστοσελίδων - browser. Αυτό είχε ως αποτέλεσμα, η γλώσσα αυτή να συνδεθεί στενά με την ανάπτυξη μικρό-εφαρμογών στο Διαδίκτυο. Η πραγματική όμως αύξηση της δημοτικότητας της Java ξεκίνησε όταν η εταιρεία Netscape ενσωμάτωσε την δυνατότητα να τρέχει μικρό-εφαρμογές μέσα στο δικό της πρόγραμμα ανάγνωσης ιστοσελίδων.

Κατά την διάρκεια της δημιουργίας της πλατφόρμας .NET της Microsfot οι κλάσεις και οι βιβλιοθήκες γράφτηκαν χρησιμοποιώντας ένα Μεταγλωττιστή Compiler με το όνομα Simple Managed C - SMC. Τον Ιανουάριο του 1999 ο Anders Hejlsberg συγκρότησε μια ομάδα με σκοπό να φτιάξει μια καινούρια γλώσσα με όνομα Cool - C-like Object Oriented Language. Παρόλο που η Microsoft σκεφτόταν να κρατήσει το όνομα Cool σαν το τελικό όνομα της γλώσσας αυτό δεν έγινε ποτέ για λόγους πνευματικών δικαιωματών. Μέχρι τον Ιούλιο του 2000 όπου ανακοινώθηκε η πλατφόρμα .NET η γλώσσα είχε είδη μετονομαστεί σε C# στην οποία αργότερα εισήχθησαν οι βιβλιοθήκες της ASP.NET.

Ο James Gosling, σχεδιαστής της Java, το 1994 μαζί με τον Bill Joy ,συνιδρυτής της Sun Microsystems, αποκάλεσαν την C# μια απομίμηση της Java. Ο Gosling επίσης συμπλήρωσε ότι η C# είναι ίδια με την Java χωρίς αξιοπιστία, παραγωγικότητα και ασφάλεια. Οι συγγραφείς ενός βιβλίου της C# ισχυρίστηκαν ότι η Java και η C# είναι πανομοιότυπες επαναληπτικές και χωρίς καινοτομίες. Τον Ιούνιο του 2000 ο Anders Hejlsberg υποστήριξε ότι η C# δεν είναι κλώνος της Java αλλά ότι είναι πολύ πιο κοντά στην C++. Από τότε που κυκλοφόρησε η δεύτερη έκδοση της C# τον Νοέμβριο του 2005, η C# και η Java άρχισαν να απομακρύνονται η μια από την άλλη μοιάζοντας όλο και λιγότερο.

1.2 Λειτουργικό Σύστημα Android

Το Android είναι ένα Λειτουργικό Σύστημα για κινητές συσκευές όπως έξυπνα τηλέφωνα - smartphones, tablets κλπ το οποίο είναι βασισμένο πάνω στο Λειτουργικό Σύστημα Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους προγραμματιστές να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι στο μεγαλύτερο μέρος του σχεδιασμένο για συσκευές με οθόνες αφής. Παρόλα αυτά το λειτουργικό Android έχει εισέλθει και σε άλλες συσκευές όπως τηλεοράσεις, ρολόγια, κονσόλες παιχνιδιών, φωτογραφικές μηχανές.

Συγκριτικά με τα υπόλοιπα Λειτουργικά Συστήματα, είναι το πιο διαδεδομένο κατέχοντας μεγαλύτερο ποσοστό χρήσης απ ότι τα Windows, το iOS και το OSX μαζί.

Εικόνα-1. Το βασικό λογότυπο της εταιρίας Android

1.3 Εκδόσεις Android

Όλες οι εκδόσεις του λειτουργικού Android ακολουθούν μία αλφαβητική σειρά έχοντας όλες τους όνομα προερχόμενο από κάποιο γλυκό.

1.3.1 Android Beta

Πρώτη έκδοση του Android. Είχαν επικεντρωθεί στην πραγματοποίηση δοκιμών προκείμενου να προσδώσουν χρηστικότητα στην συσκευή. Η πρώτη αυτή έκδοση είχε πολλά προβλήματα στην ταχύτητα και στην απόδοση.

1.3.2 Android Astro

Πρώτη πλήρης έκδοση του λειτουργικού Android, που παρουσιάστηκε τον Σεπτέμβριο του 2008. Είχε υποστήριξη wifi και bluetooth, δεν είχε λειτουργικότητα copy-paste στον web browser και γενικά ήταν αρκετά αργό ως προς την λειτουργία του.

1.3.3 Android Cupcake 1.5

Παρουσιάστηκε στις 30 Απριλίου 2009. Προστέθηκε η αυτόματη περιστροφή η λειτουργικότητα copy-paste και υπήρχε αισθητή βελτίωση στη ταχύτητα όχι όμως στο επιθυμητό επίπεδο.

1.3.4 Android Donut 1.6

Παρουσιάστηκε στις 15 Σεπτεμβρίου 2009. Προστέθηκε η φωνητική αναζήτηση καθώς και το search box. Ήταν αρκετά πιο γρήγορο από τους προκατόχους του.

1.3.5 Android Eclair 2.0/2.1

Παρουσιάστηκε στις 26 Οκτωβρίου 2009. Υποστήριζε Bluetooth 2.1, βελτιωμένη ταχύτητα πληκτρολόγησης σε εικονικό πληκτρολόγιο με καλύτερο λεξικό. Όπως και στις προηγούμενες εκδόσεις δεν υπάρχει υποστήριξη για Adobe Flash.

1.3.6 Android Foryo 2.2

Παρουσιάστηκε 20 Μαΐου 2010. Υποστήριζε Adobe Flash 10.1.

1.3.7 Android Gingerbread 2.3

Κυκλοφόρησε στις 6 Δεκεμβρίου 2010. Βελτιώθηκε το περιβάλλον εργασίας χρήστη με υψηλή απόδοση και ταχύτητα. Έχουμε κλήσεις μέσω Διαδικτύου. Επιλογή λέξης one touch και copy / paste, νέο πληκτρολόγιο για πιο γρήγορη εισαγωγή κειμένου. Ήταν η πιο επιτυχημένη έκδοση του Android από τις προηγούμενες εκδόσεις. Δεν υποστηρίζει όμως multi-core επεξεργαστές.

1.3.8 Android Honeycomb 3.0

Κυκλοφόρησε στις 22 Φεβρουαρίου 2011. Υποστήριζε multi-core επεξεργαστές και είχε την δυνατότητα για την κρυπτογράφηση όλων των δεδομένων του χρήστη. Αυτή η έκδοση του Android ήταν διαθέσιμη μόνο για τα tablets.

1.3.9 Android IceCreamSandwich(ICS) 4.0

Κυκλοφόρησε στις 14 Νοεμβρίου του 2011. Διέθετε εικονικό κουμπί στο User Interface - UI, μια νέα οικογένεια γραμματοσειράς για το UI. Επίσης είχε την δυνατότητα να καλεί τις εφαρμογές που χρησιμοποιούν δεδομένα στο παρασκήνιο.

1.3.10 Android JellyBean 4.1

Παρουσιάστηκε στις 9 Ιουλίου του 2012. Αυτή την στιγμή κατέχει περίπου το 50% των Android συσκευών. Προσέθεσε το Google Now, τον ψηφιακός βοηθό της Google.

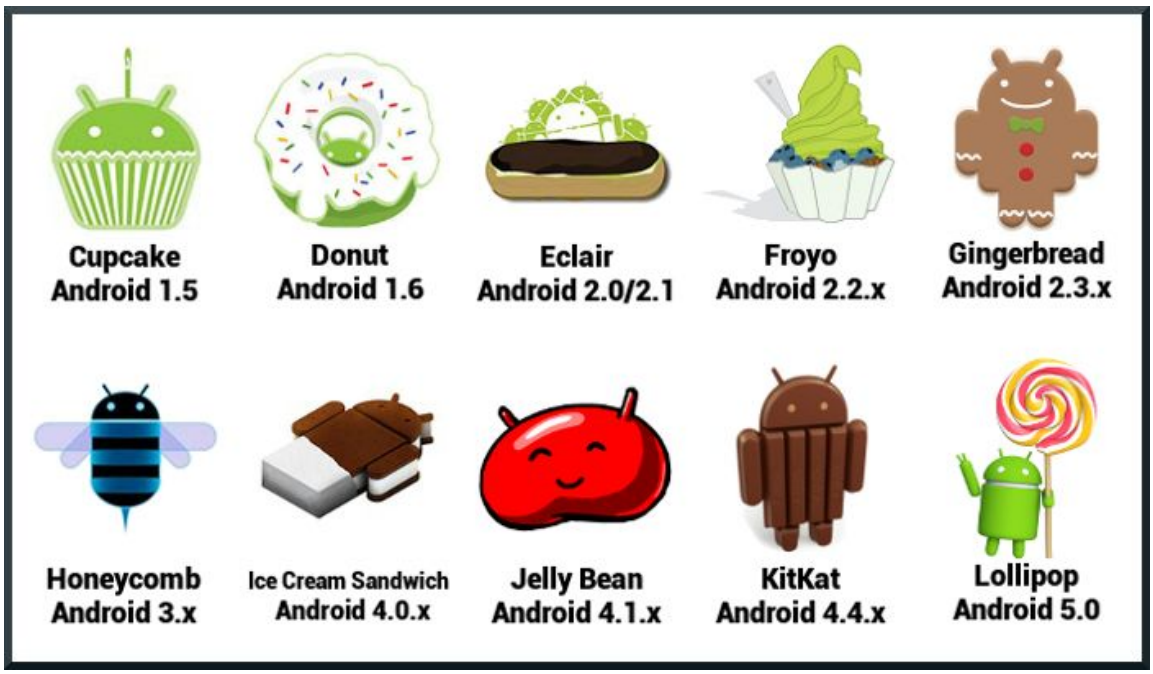
1.3.11 Android KitKat 4.4

Παρουσιάστηκε στις 31 Οκτωβρίου του 2013. Ενίσχυσε το UI. Χάρη στις πολλές βελτιώσεις των επιδόσεων που έγιναν υποστήριξε και συσκευές

με μόλις 512MB RAM. Αυτό ήταν ένα μεγάλο βήμα καθώς μέχρι τότε οι εταιρίες ήταν αδιάφορες ως προς την υποστήριξη των φτηνών συσκευών.

1.3.12 Android Lollilop 5.0

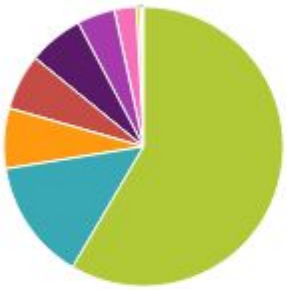
Η τελευταία έκδοση του Android η οποία παρουσιάστηκε στις 12 Νοεμβρίου 2014. Διαθέτει καλύτερη διαχείριση ενέργειας της μπαταρίας για μεγαλύτερη διάρκεια, αυτόματη κρυπτογράφηση δεδομένων για μεγαλύτερη ασφάλεια και πολλαπλά profile για περισσότερους χρήστες σε μία συσκευή.



Εικόνα-2. Τα λογότυπα των διαφόρων εκδόσεων Android

Πίνακας-1. Στατιστικά χρήσης εκδόσεων Android

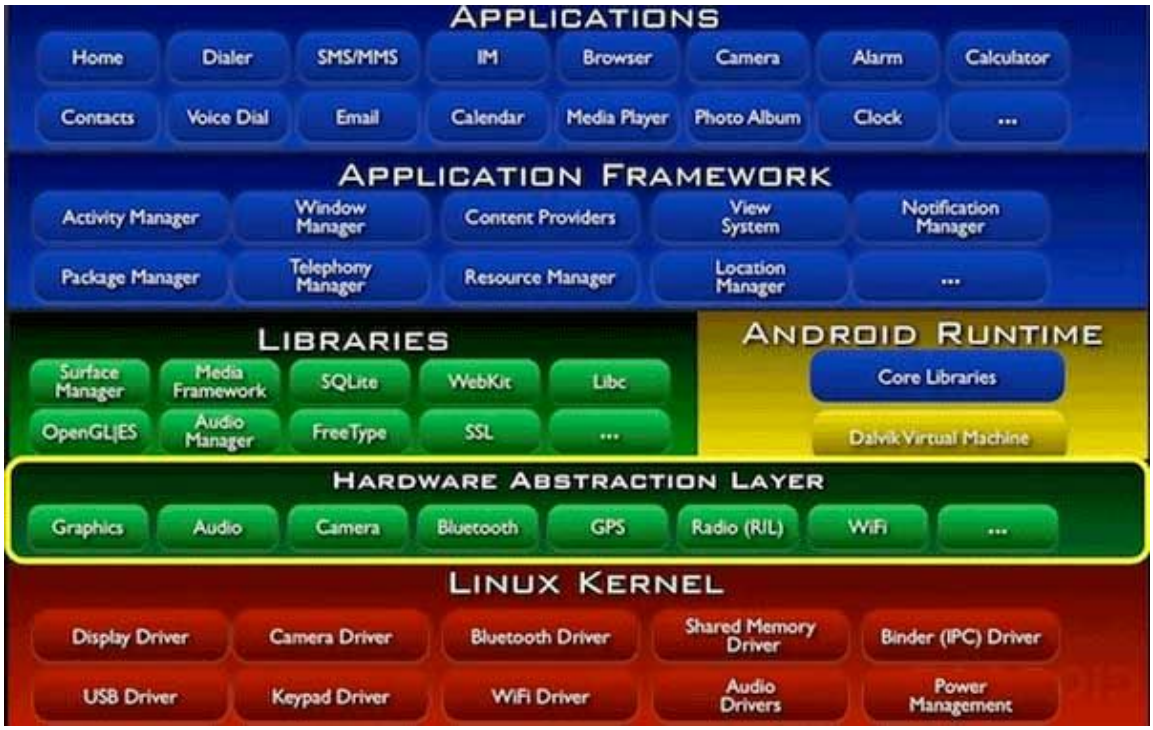
CURRENT INSTALLS BY DEVICE ON JAN 29, 2015



	YOUR APP	
Android 4.4	1,176	58.48%
Android 4.1	281	13.97%
Android 2.3.3 - 2.3.7	140	6.96%
Android 4.2	130	6.46%
Android 4.0.3 - 4.0.4	128	6.36%
Android 4.3	88	4.38%
Android 5.0	51	2.54%
Android 2.2	9	0.45%
Android 3.2	5	0.25%
Android 3.1	2	0.10%
Others	1	0.05%

1.4 Αρχιτεκτονική του λειτουργικού Android

Το Λειτουργικό Σύστημα Android είναι μια συλογή στοιχείων λογισμικού που χωρίζονται σε πέντε ενότητες και τέσσερα κύρια στρώματα, όπως φαίνεται στην Εικόνα-3.



Εικόνα-3. Αρχιτεκτονική λειτουργικού Android

Ο Πυρήνας - Linux Kernel είναι το χαμηλότερο επίπεδο και είναι υπεύθυνο για την αλληλεπίδραση με το Hardware. Επίσης προσφέρει την βασική διεργασία στο σύστημα όπως επεξεργασία δεδομένων, διαχείριση της μνήμης και διαχείριση των συσκευών – περιφερειακών όπως την κάμερα, το πληκτρολόγιο, την οθόνη κλπ. Τέλος ασχολείται με την διαχείριση του δικτύου και προσφέρει όλους τους απαραίτητους Οδηγούς - Drivers που απαιτούνται για την λειτουργία των περιφεριακών.

Οι Βιβλιοθήκες - Libraries του συστήματος βρίσκονται πάνω από τον πυρήνα. Εκεί περιλαμβάνονται οι libc, το WebKit και το SQLite Database που χρησιμοποιείται για την αποθήκευση και διαμοιρασμό δεδομένων της εφαρμογής. Συνήθως είναι προγραμματισμένες σε γλώσσα C/C++ και η πρόσβαση σε αυτές δεν είναι δυνατή καθώς χρησιμοποιούνται έμμεσα μέσω αυτών του ανώτερου επιπέδου.

Στην συνέχεια συναντάμε το Android Runtime που περιλαμβάνει το Dalvik Virtual Machine που είναι ένα είδος Εικονικής Μηχανής Java ειδικά σχεδιασμένης και βελτιστοποιημένης για Android. Αυτό που το κάνει ξεχωριστό είναι ο σχεδιασμός του, που δίνει την δυνατότητα σε κάθε εφαρμογή να τρέχει σε δικό της εικονικό μηχάνημα.

Το Πλαίσιο Εφαρμογών - Application Framework περιέχει ένα πλαίσιο ανάπτυξης που τυποποιεί και διευκολύνει τον προγραμματισμό των Android εφαρμογών. Αυτές οι βιβλιοθήκες είναι γραμμένες σε Java και μπορούν να χρησιμοποιηθούν από τους Developers.

Οι Εφαρμογές - Applications είναι το βασικό στοιχείο του Android. Διαθέτει μια γκάμα προεγκατεστημένων εφαρμογών όπως το Ημερολόγιο, τις Επαφές, τα Μηνύματα SMS κλπ. Μέσα από το Google Play Store και άλλα Markets που κυκλοφορούν ο χρήστης μπορεί να εγκαταστήσει όποια εφαρμογή επιθυμεί.

2 Microsoft Visual Studio 2010

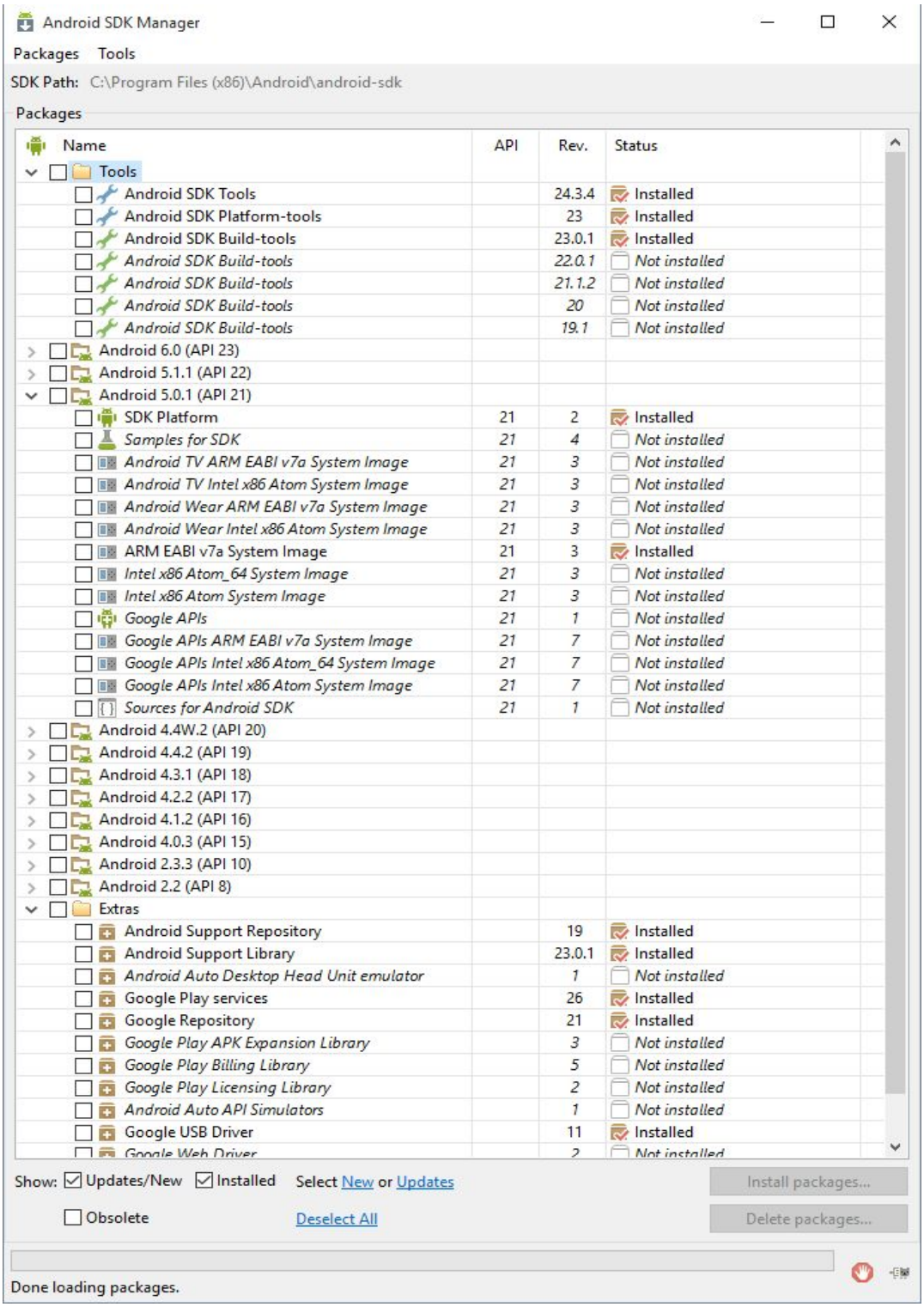
Αρχικά εγκαταστάθηκε το MS Visual Studio 2010 Ultimate Edition. Επιλέχθηκε η έκδοση 2010 και όχι κάποια από τις εκδόσεις 2013 ή 2015 που υπάρχουν. Ο λόγος που δεν επιλέχθηκε κάποια από τις άλλες δύο εκδόσεις είναι ότι υπάρχει μικρός αριθμός στοιχείων σε βιβλιογραφία και σε κείμενα εκμάθησης για τον προγραμματισμό εφαρμογών για Android συσκευές. Επίσης υπάρχουν πολλές ανομοιομορφίες μεταξύ των πιο πάνω εκδόσεων κυρίως λόγω των αλλαγών που επιτέλεσε η Microsoft, πράγμα που καθιστά την χρήση της έκδοσης 2010 περισσότερο εύκολη στην χρήση. Εδώ επισημαίνεται ότι τα αρχεία apk εμπεριέχουν μόνο java κώδικα. Επιλέχθηκε η έκδοση 32bit ανεξαρτήτως Λειτουργικού Συστήματος γιατί η δωρεάν έκδοση του MSDNA για το Visual Studio είναι επίσης 32bit. Κατεβάζουμε το MS Visual Studio 2010 Ultimate Edition και προχωράμε στην εγκατάσταση του. Κατά την εγκατάσταση επιλέγουμε την γλώσσα C# που μας ενδιαφέρει.

2.1 Εγκατάσταση Android SDK

Πριν γίνει η εγκατάσταση του Android SDK χρειάζεται να γίνει λήψη και εγκατάσταση της έκδοσης 8 JDK (Java Development Kit) της Oracle για 32bit λειτουργικό σύστημα Windows [1].

Στην συνέχεια εγκαταστάθηκε το Android SDK [2] και Android NDK [3]. Το Android SDK (Android Software Development Kit) είναι το εργαλείο της Google για αυτούς που θέλουν να δημιουργήσουν εφαρμογές σε Android. Το Android SDK περιλαμβάνει μια μεγάλη λίστα με εργαλεία

ανάπτυξης όπως Εργαλεία Debugging για τις εφαρμογές, βιβλιοθήκες, εξομοιωτής συσκευών (Android Virtual Machines - AVM), κείμενα βοήθειας κλπ. Μέσω του SDK μπορούμε να χρησιμοποιήσουμε εργαλεία όπως το ADB, για να μεταφέρουμε αρχεία σε χώρους που κανονικά δεν επιτρέπεται και το fastboot για να εγκαθιστούμε custom recovery εικόνες και να ξεκλειδώνουμε τον bootloader της συσκευής μας. Τέλος περιέχει συγκεκριμένα αρχεία για την μεταγλώττιση του πηγαίου κώδικα ώστε να τρέχει στην εικονική μηχανή Dalvik 27.

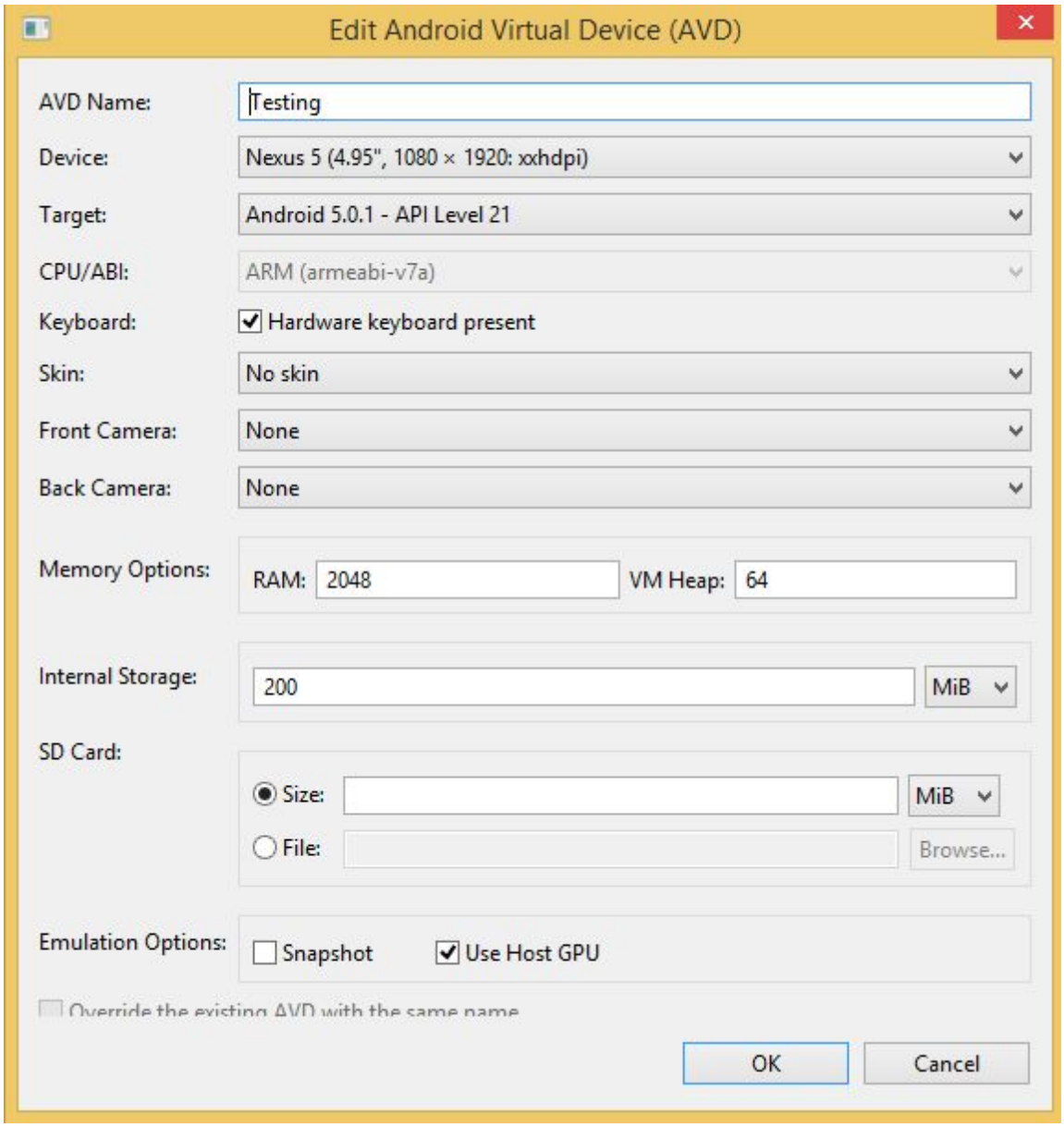


Εικόνα-4. Απαραίτητα πακέτα SDK για εγκατάσταση

2.2 Δημιουργία AVD

Το Android SDK περιλαμβάνει μια κινητή συσκευή προσομοίωσης emulator που μπορεί και εκτελείται στον υπολογιστή. Ο εξομοιωτής επιτρέπει την ανάπτυξη και δοκιμή Android εφαρμογών χωρίς τη χρήση μιας φυσικής συσκευής. Ο εξομοιωτής του Android χρησιμοποιεί τις Android Virtual Device (AVD). Οι AVD επιτρέπουν τον ορισμό διαφόρων χαρακτηριστικών του υλικού που θέλουμε να προσομοιάσουμε. Η εφαρμογή που εκτελείται στον εξομοιωτή, μπορεί να χρησιμοποιήσει τις υπηρεσίες της πλατφόρμας Android για να επικαλεστεί άλλες εφαρμογές, να έχει πρόσβαση στο δίκτυο, να αναπαράγει ήχο και βίντεο, να αποθηκεύει και να ανακτά δεδομένα.

Το Android Virtual Device (AVD) επιτρέπει την δημιουργία μίας εικονικής συσκευής Android για την οποία προορίζεται το πρόγραμμα το οποίο γράφεται. Μπορούμε με σχετικά μεγάλη ευκολία να εκτελέσουμε οποιαδήποτε εφαρμογή δημιουργήσουμε. Το πλεονέκτημα της AVD είναι ότι μπορούμε να δούμε πώς θα προβάλλεται το λογισμικό που δημιουργήσαμε σε διαφορετικές συσκευές με Android. Για την δημιουργία μιας τέτοιας εικονικής συσκευής εκτελούμε την εφαρμογή AVD manager που βρίσκεται C:\Program Files (x86)\Android\android-sdk. (Περιγραφή παραμετροποιήσεων) .

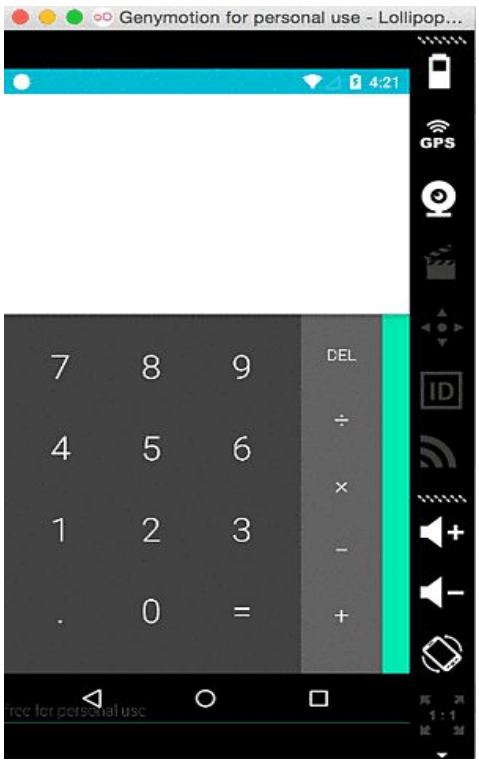


Εικόνα-5. Χαρακτηριστικά για την δημιουργία εικονικής μηχανής

2.3 Genymotion

Αντί της παραπάνω επιλογής για να δημιουργήσουμε μια εικονική συσκευή Android και να τρέξουμε το πρόγραμμα που έχουμε δημιουργήσει μπορούμε να χρησιμοποιήσουμε το Genymotion. Είναι ένας εξομοιωτής

του Android ο οποίος περιλαμβάνει ένα πλήρες σύνολο των αισθητήρων και τα χαρακτηριστικών μιας Android συσκευής ώστε να μπορεί ο χρήστης να αλληλεπιδράσει πλήρως με μία εικονική συσκευή Android. Με το Genymotion, δύναται να δοκιμαστούν οι Android εφαρμογές σε ένα ευρύ φάσμα των εικονικών συσκευών για αναπτυξιακούς σκοπούς, δοκιμή και επίδειξη. Σε σχέση με το AVD που περιγράψαμε παραπάνω υπερτερεί εξαιτίας του γεγονότος ότι μπορούμε να επιλέξουμε η εικονική μας μηχανή να έχει παραπάνω από έναν πυρήνες οπότε να είναι αισθητά πιο γρήγορη.



Εικόνα-6. Στιγμιότυπο εικονικής μηχανής Genymotion

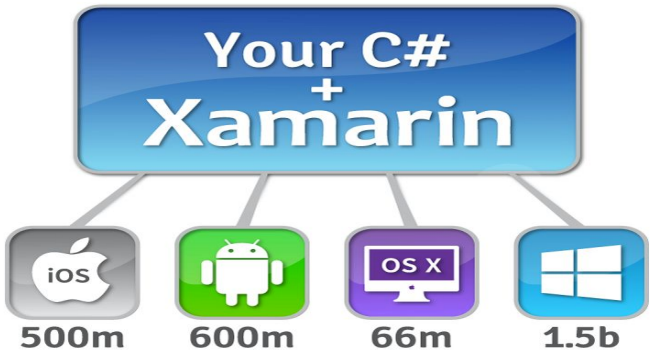
2.4 Εγκατάσταση Android NDK

Μετά από το SDK εγκαταστάθηκε το NDK (Native Development Kit). Το NDK είναι ένα εργαλείο που επιτρέπει τη μεταγλώττιση κάποιων κομματιών μίας android εφαρμογής χρησιμοποιώντας γλώσσες προγραμματισμού όπως είναι η C και η C++. Για ορισμένους τύπους εφαρμογών, αυτό μπορεί να είναι χρήσιμο, καθώς με αυτόν τον τρόπο μπορούν να χρησιμοποιηθούν ξανά ορισμένες υπάρχουσες βιβλιοθήκες που ενδέχεται να έχουν αυξημένη απόδοση.

Αξίζει να σημειωθεί ότι, η χρήση τέτοιου έτοιμου κώδικα στις εφαρμογές Android, τις περισσότερες φορές δεν οδηγεί σε αισθητή βελτίωση των επιδόσεων, αλλά αυξάνει την πολυπλοκότητα της εφαρμογής. Σε γενικές γραμμές, θα πρέπει η χρήση του NDK να γίνεται μόνο όταν ταιριάζει στη φύση της εφαρμογής που αναπτύσσεται και όχι απλά επειδή ο προγραμματιστής της εν λόγω εφαρμογής, προτιμά γλώσσες σαν τη C ή τη C++.

Τυπικά, εφαρμογές που αναπτύσσονται καλύτερα με χρήση NDK, είναι εφαρμογές που είναι αυτόνομες και που δε χρησιμοποιούν πολλή μνήμη. Κατά την εξέταση του κατά πόσο θα πρέπει ή όχι να αναπτυχθεί η εφαρμογή με την βοήθεια τέτοιου είδους κώδικα, πρέπει να ληφθούν σοβαρά υπόψη οι απαιτήσεις της καθώς επίσης και το αν τα Android APIs, που θα χρησιμοποιηθούν, παρέχουν τη λειτουργικότητα που χρειάζεται.

2.5 Xamarin



Εικόνα-7. Λογότυπο του λογισμικού Xamarin

Η πλατφόρμα του Xamarin είναι ένα εργαλείο ανάπτυξης κώδικα (Integrated Development Environment - IDE) που επιτρέπει την συγγραφή κώδικα σε γλώσσα C# για εφαρμογές iOS και Android.

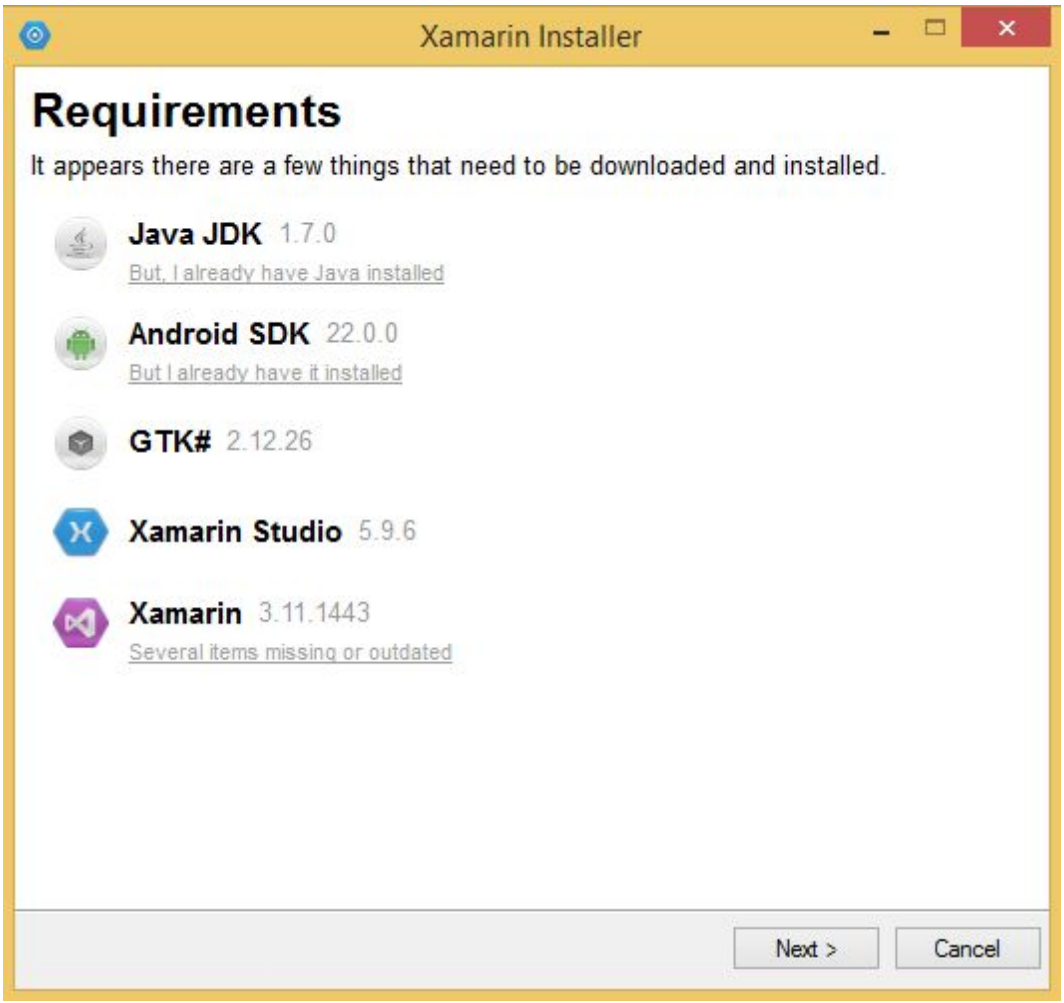
Παρά το γεγονός ότι ο κώδικας ο οποίος γράφεται στο Xamarin μπορεί να χρησιμοποιηθεί σε διάφορες πλατφόρμες η διαδικασία μετατροπής του κώδικα για το κάθε σύστημα είναι αρκετά διαφορετική.

Η ανάπτυξη κώδικα στο Xamarin μπορεί να γίνει είτε μέσα από το Xamarin Studio είτε μέσα από το Visual Studio όπου το Xamarin έχοντας εγκατασταθεί ως add-on προσφέρει μία πλήρη εργαλειοθήκη.

Η βασική λειτουργία του Xamarin είναι να μετατρέπει κάθε γραμμή κώδικα C# σε γλώσσα μηχανής και στην συνέχεια να δημιουργεί ένα .apk αρχείο προς εκτέλεση.

Για την εγκατάσταση του απαιτούνται τα JDK και Android SDK τα οποία περιγράψαμε παραπάνω. Η εγκατάσταση του Xamarin μπορεί να γίνει με δύο τρόπους. Ο ένας τρόπος είναι να κατεβάσουμε και να εκτελέσουμε

τον web-installer (https://xamarin.com/download). Εδώ όλα τα απαραίτητα εργαλεία εγκαθίστανται αυτόματα.



Εικόνα-8. Xamarin installer

Ο δεύτερος τρόπος είναι να εγκαταστήσει κανείς μόνος του τα επιμέρους πακέτα.

Πίνακας-2. Επιμέρους κομμάτια που πρέπει να εγκατασταθούν

SDK (JDK)

Android SDK

GTK# [4]

Xamarin Studio για Windows [5]

Xamarin Visual Studio [6]

3 Android Studio

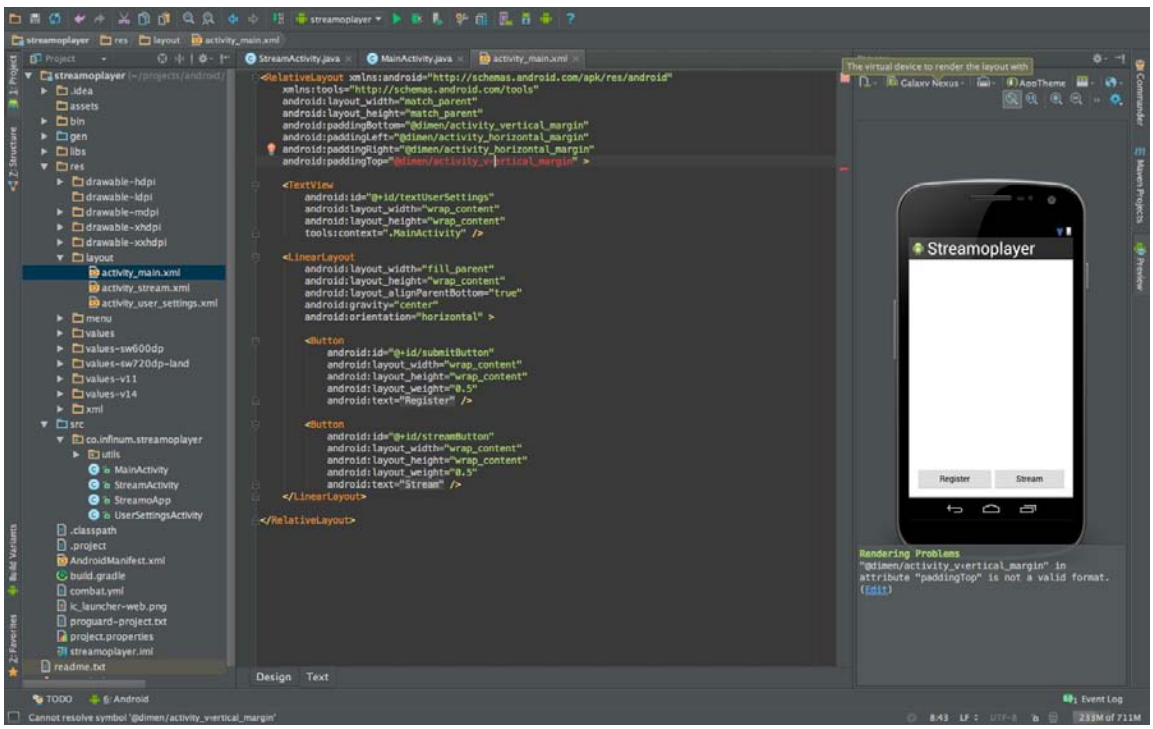
Ένα άλλο Ολοκληρωμένο Περιβάλλον Ανάπτυξης είναι το Android Studio το οποίο είναι δημιουργία της Google. Το Android Studio ήταν σε αρχικό στάδιο από την έκδοση 0.1 Μάιο του 2013, στη συνέχεια πέρασε στις δοκιμαστικές εκδόσεις beta ξεκινώντας από την έκδοση 0.8 που κυκλοφόρησε τον Ιούνιο του 2014, ενώ η πρώτη σταθερή έκδοση 1.0 κυκλοφόρησε το Δεκέμβριο του 2014.

Σε αντίθεση με άλλα IDE τα οποία είναι σχεδιασμένα για δημιουργία κώδικα για πολλές πλατφόρμες το Android Studio με βάση το IntelliJ λογισμικό IDEA έχει σχεδιαστεί ειδικά για την ανάπτυξη κώδικα μόνο για Android συσκευές.

Εικόνα-9. Λογότυπο Android Studio

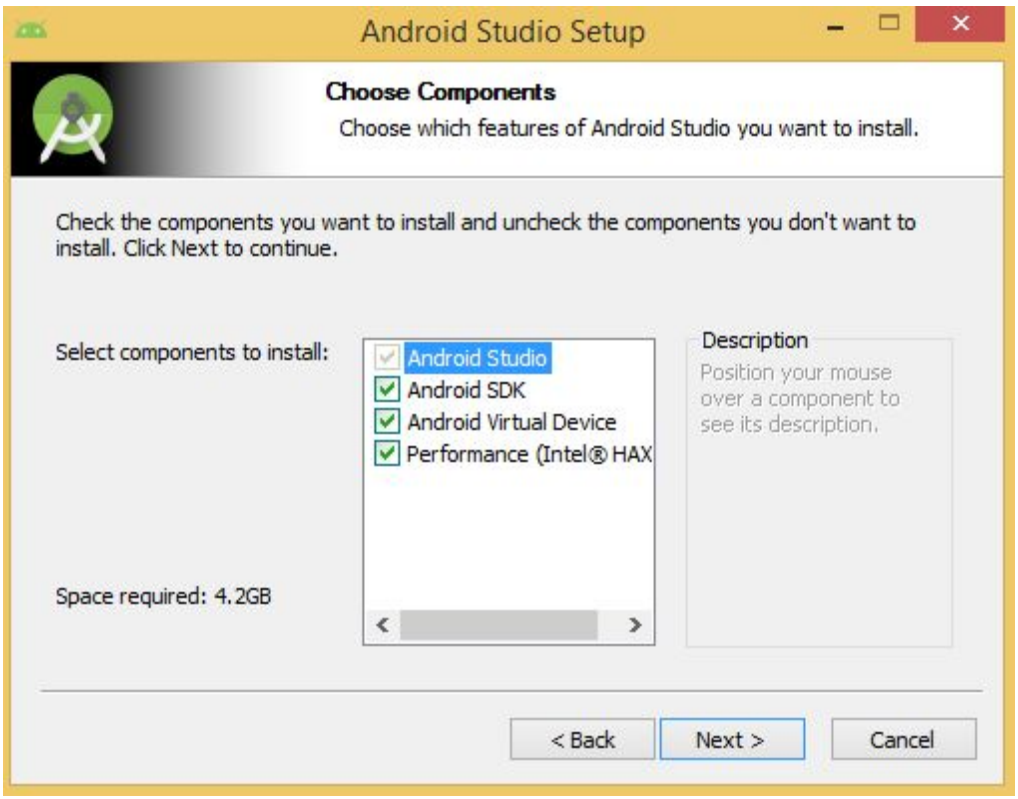
Μερικά από τα βασικά χαρακτηριστικά του Android Studio

- Δυνατότητα ταυτόχρονης προβολής σε πολλαπλές οθόνες
- Ευέλικτο σύστημα compiling/building
- Δημιουργία πολλαπλών εκδόσεων αρχείων apk
- Πρότυπα κώδικα
- Πλούσιο επεξεργαστή διάταξης με υποστήριξη drag and drop
- Εργαλεία lin για την παρακολούθηση απόδοσης, τη χρηστικότητας, τη συμβατότητας έκδοσης, και άλλων προβλημάτων
- Σύστημα app-signing
- Ενσωματωμένη υποστήριξη για την cloud πλατφόρμα της Google



Εικόνα-10. Στιγμιότυπο απο την κονσόλα του Android Studio

Για την εγκατάσταση του είναι απαραίτητο να υπάρχει η έκδοση JDK της Java απο την Oracle. Στην συγκεκριμένη περίπτωση βέβαια η έκδοση θα πρέπει να είναι η 8η της Oracle για 64bit λειτουργικό σύστημα Windows σε αντίθεση με την 32bit Windows του Visual Studio. Τα SDK και NDK είναι και εδώ προαπαιτούμενα για την λειτουργία του Android Studio και η εγκατάσταση τους γίνεται χειροκίνητα όπως έχει περιγραφεί παραπάνω. Κατεβάζοντας και εκτελώντας τον installer του Android Studio [2]. Το πακέτο SDK ασχέτως αν έχει εγκατασταθεί πιο πριν εγκαθίστανται ξανά αυτόματα.

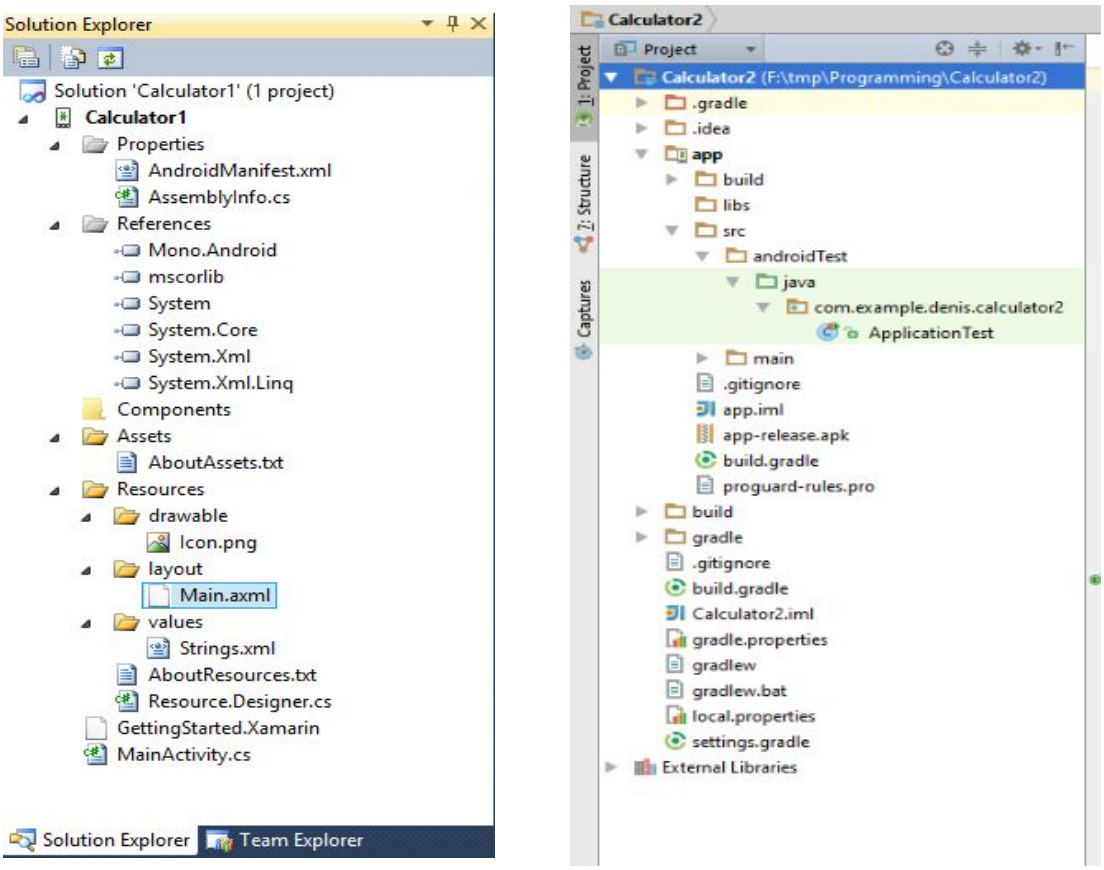


Εικόνα-11. Στιγμιότυπο απο την εγκατάσταση του Android Studio

4 Ανάπτυξη Εφαρμογών

4.1 Περιγραφή Προγράμματος

Οι εφαρμογές για Android συσκευές αποτελούνται από διαφορετικά ξεχωριστά αρχεία τα οποία επικοινωνούν μεταξύ τους. Αποτελούνται από Activities και τα αντίστοιχα xml αρχεία, το manifest αρχείο, εικόνες διάφορων μεγεθών, αρχεία ήχου και ότι άλλο απαιτεί η υλοποίηση της κάθε εφαρμογής.



Εικόνα-12. Οι φάκελοι αρχείων των δύο εφαρμογών(αριστερά Visual Studio, δεξιά Android Studio)

Ο πηγαίος κώδικας των εφαρμογών βρίσκεται στα αρχεία .java και .cs τα οποία είναι γραμμένα σε γλώσσα Java και C# αντίστοιχα και είναι υπεύθυνα για το λειτουργικό κομμάτι της εφαρμογής. Το σχεδιαστικό μέρος της εφαρμογής βρίσκεται στα αρχεία .xml και .axml τα οποία είναι γραμμένα σε γλώσσα XML (Extensible Markup Language).

Αυτά τα αρχεία είναι υπεύθυνα για αυτό που βλέπει ο χρήστης. Χαρακτηριστικό είναι ότι κάθε διαφορετική οθόνη που παρουσιάζεται στον χρήστη όταν αυτός επιλέγει μια λειτουργία, είναι ένα διαφορετικό αρχείο .xml. Άλλα αρχεία που συνδυάζονται μέσω του Android Studio είναι τα γνωστά .jpg και .png τα οποία περιέχουν τις σχετικές εικόνες που χρειάζονται για να μπορεί ο χρήστης να βλέπει το επιθυμητό αποτέλεσμα. Τέλος μπορεί να υπάρχει ένα αρχείο .html το οποίο είναι γραμμένο σε γλώσσα HTML (HyperText Markup Language) στο οποίο περιέχονται πληροφορίες για την εφαρμογή αναλόγως το είδος της. Χρησιμοποιήθηκε ο ίδιος πηγαίος κώδικας και στις δύο περιπτώσεις ώστε να μπορέσει να γίνει σύγκριση ανάμεσα στα δύο προγράμματα.

Αρχικά καλείται η κατάλληλη μέθοδος για την δημιουργία του layout η μέθοδος αυτή παρέχεται αυτόματα και καλείται για τις δύο γλώσσες. Στην συνέχεια δημιουργούνται τα ανάλογα αντικείμενα. Η δημιουργία των αντικειμένων είναι μια γέφυρα μεταξύ της γλώσσας προγραμματισμού που χρησιμοποιήται και του αρχείου που περιέχει το layout.

```
Button button0 = FindViewById<Button>(Resource.Id.button0);
```

Κώδικας C# για την δημιουργία button

```
Button button0 = (Button)findViewById(R.id.button0);
```

Κώδικας Java για την δημιουργία button

Στην συνέχεια δημιουργούνται τα κατάλληλα event πχ τι θα κάνει το πρόγραμμα στην περίπτωση που πατηθεί κάποιο button.

Στην περίπτωση της C# χρησιμοποιήθηκαν handlers με τα οποία το πρόγραμμα αυτόματα διακλαδώνεται στην ανάλογη μέθοδο που εμπεριέχεται ως όρισμα του hadler.

```
button0.Click += new EventHandler(button0_Click);
```

Ορισμός handler στην C# για την κλήση του button0_Click()

Η Java δεν χρησιμοποιεί handler τουλάχιστον με την έννοια με την οποία ορίζονται στην C#. Αντίστοιχα χρησιμοποιεί listener. Το listener είναι μια αόριστη έκφραση η οποία ορίζεται ανάλογα με το event. Στην συνέχεια ορίζεται ξεχωριστή μέθοδος για το event του listener μέσα στο οποίο ελέγχουμε απο ποίο αντικείμενο έγινε η κλίση.

```
{
button0.setOnClickListener(this);
}
public void onClick(View view) {
if (view.getId()==R.id.button0)
            TypeNum("0");
}
```

Ορισμός listener στην Java για την κλήση του button0_Click() στην περίπτωση που πατηθεί το button0

Κάθε φορά που πατιέται κάποιο button απο 0-9 καλείται η μέθοδος TypeNum. Η μέθοδος αυτή προσθέτει στα δεξιά του TextNum το ψηφίο που

πατήθηκε. πχ αν το TextNum έχει τιμή 123 και πατηθεί το button4 τότε το TextNum θα πάρει την τιμή 1234. Στην περίπτωση που το TextNum είναι κενό ή μηδέν (αρχική κατάσταση ή κατάσταση μετά απο καθαρισμό ψηφίων) εμφανίζεται μόνο ο αριθμός που πατήθηκε. πχ Αν το TextNum έχει την τιμή 0 και πατηθεί το button4 το TextNum θα πάρει την τιμή 4.

Αν πληκτρολογήσουμε το buttonC τότε καλείται η μέθοδος ClearResNum με την οποία καθαρίζονται τα περιεχόμενα των TextRes, TextNum, buttonAri.

Αν πατηθεί το buttonCE τότε καλείται η μέθοδος ClearNum με την οποία καθαρίζονται το περιεχόμενο του TextNum.

Αν πατηθεί το buttonDel τότε εάν το μήκος του TextNum είναι μεγαλύτερο του ένα δηλαδή αν το TextNum περιέχει έναν διψήφιο αριθμό τότε καλείται η μέθοδος DelNum διαφορετικά καλείται η μέθοδος ClearNum. Η μέθοδος DelNum αφαιρεί το τελευταίο ψηφίο απο τα δεξιά του αριθμού. πχ Αν το TextNum έχει την τιμή 123 και πατηθεί το buttonDel τότε το TextNum παίρνει την τιμή 12. Αν το textnum έχει την τιμή 3 και πατηθεί το buttonDel τότε το TextNum παίρνει την τιμή 0.

Για κάθε μία απο τις αριθμητικές πράξεις καλείται η μέθοδος TypeAri. Στην μέθοδο αυτή γίνεται έλεγχος αν το buttonAri δεν περιέχει τον χαρακτήρα "=" αν ισχύει η συνθήκη τότε πρώτα καλείται η μέθοδος Calculate παίρνοντας σαν όρισμα την πράξη που θέλουμε να εκτελεστεί και στην συνέχεια εμφανίζουμε στο buttonAri την αριθμητική αυτή πράξη. Διαφορετικά οι δύο παραπάνω μέθοδοι εκτελούνται αντίστροφα.

Στη μέθοδο Calculate ελέγχουμε αν το TextRes και το TextNum δεν είναι κενά. Αυτό γιατί η αριθμητική πράξη θα πρέπει να εκτελεστεί μεταξύ των δύο αυτών αντικειμένων Αν η συνθήκη είναι αληθής τότε εάν η λογική

πράξη είναι η πρόσθεση τότε προσθέτουμε τις τιμές των δύο αντικείμενων TextRes και TextNum. Τα δύο αυτά αντικείμενα είναι αλφαριθμητικά.

Για να μπορέσει να εκτελεστεί οποιαδήποτε πράξη θα πρέπει πρώτα να γίνει η μετατροπή τους σε αριθμητικό. Επειδή όμως το αποτέλεσμα γράφεται στο TextRes το οποίο είναι αλφαριθμητικό το αποτέλεσμα της αριθμητικής πράξης θα πρέπει να ξαναγίνει μετατροπή σε αλφαριθμητικό. Πρακτικά κάτι τέτοιο δεν χρειάζονταν αλλά έγινε ώστε να μπορούμε να δούμε πόσο εύκολο είναι να γίνουν τέτοιες ενέργειες μεταξύ των δύο γλωσσών.

```
editTextRes.Text=Convert.ToString(Convert.ToDecimal(editTextR
es.Text) + Convert.ToDecimal(editTextNum.Text));
```

Κωδικας C# για την πρόσθεση μεταξύ δύο αλφαριθμητικών

```
editTextRes.setText(new
BigDecimal(Double.parseDouble(editTextRes.getText().toString(
)) +
Double.parseDouble(editTextNum.getText().toString())).toPlain
String());
```

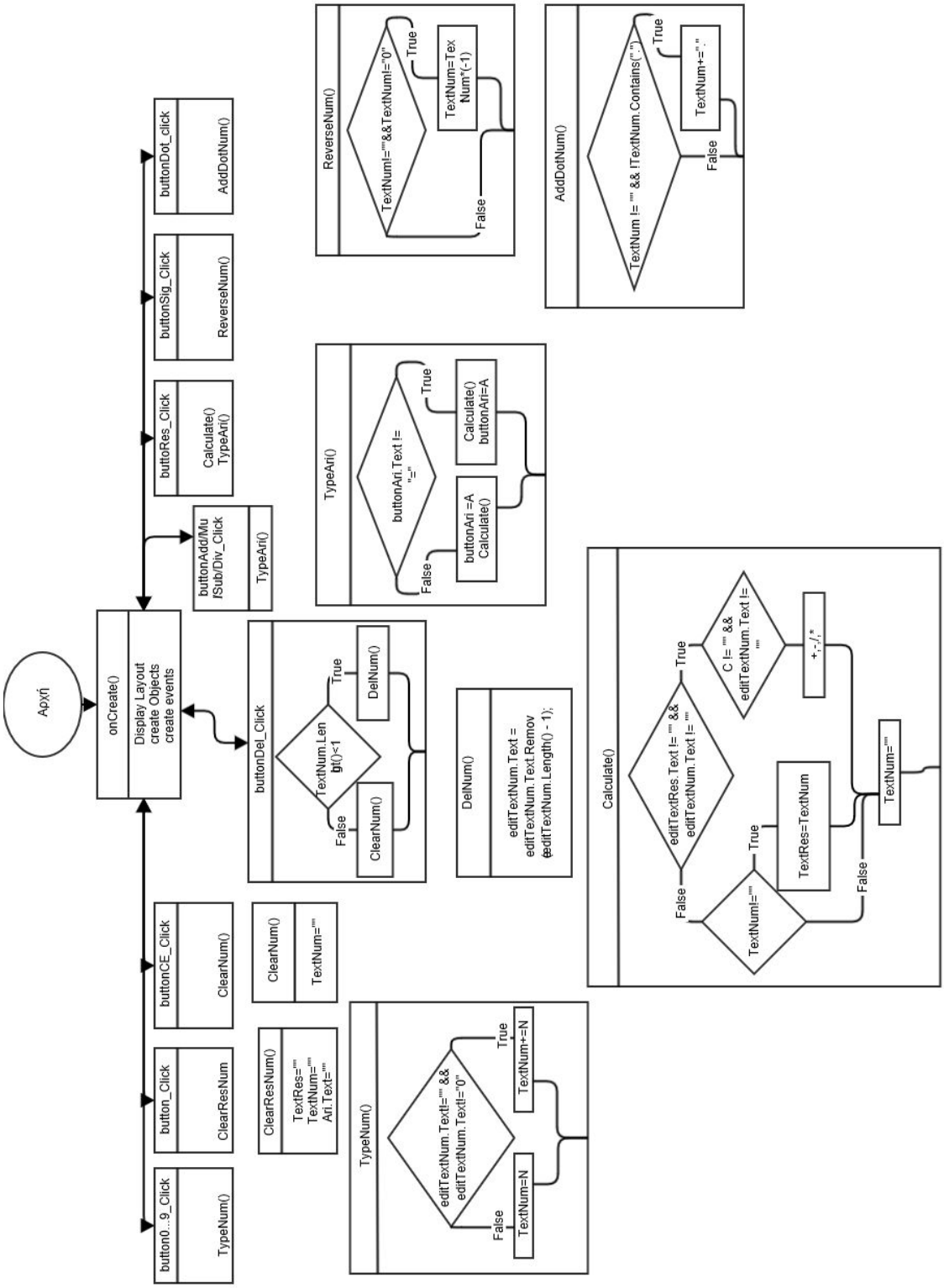
Κώδικας Java για την πρόσθεση μεταξύ δύο αλφαριθμητικών

Στην περίπτωση της διαίρεσης λήφθηκε μέριμνα για την αποτροπή της διαίρεσης με το μηδέν. Σε μία τέτοια περίπτωση εμφανίζεται αναδυόμενο μήνυμα στο κάτω μέρος της εφαρμογής.

Στη μέθοδος ReverseNum ελέγχουμε αν το TextNum δεν είναι κενό και δεν είναι μηδέν. Αν ισχύει η συνθήκη πολλαπλασιάζουμε το TextNum με -1, δηλαδή αντιστρέφεται η τιμή που έχει. πχ Αν το TextNum περιέχει

τον αριθμό 123 μετά την εκτέλεση της ReverseNum το TextNum θα περιέχει την τιμή -123.

Η μέθοδος AddDotNum προσθέτει τον χαρακτήρα "." στα δεξιά του TextNum. Αν το TextNum δεν είναι κενό και δεν περιέχει ήδη "." τότε το TextNum ισούται με την τιμή του συν τον χαρακτήρα ".".

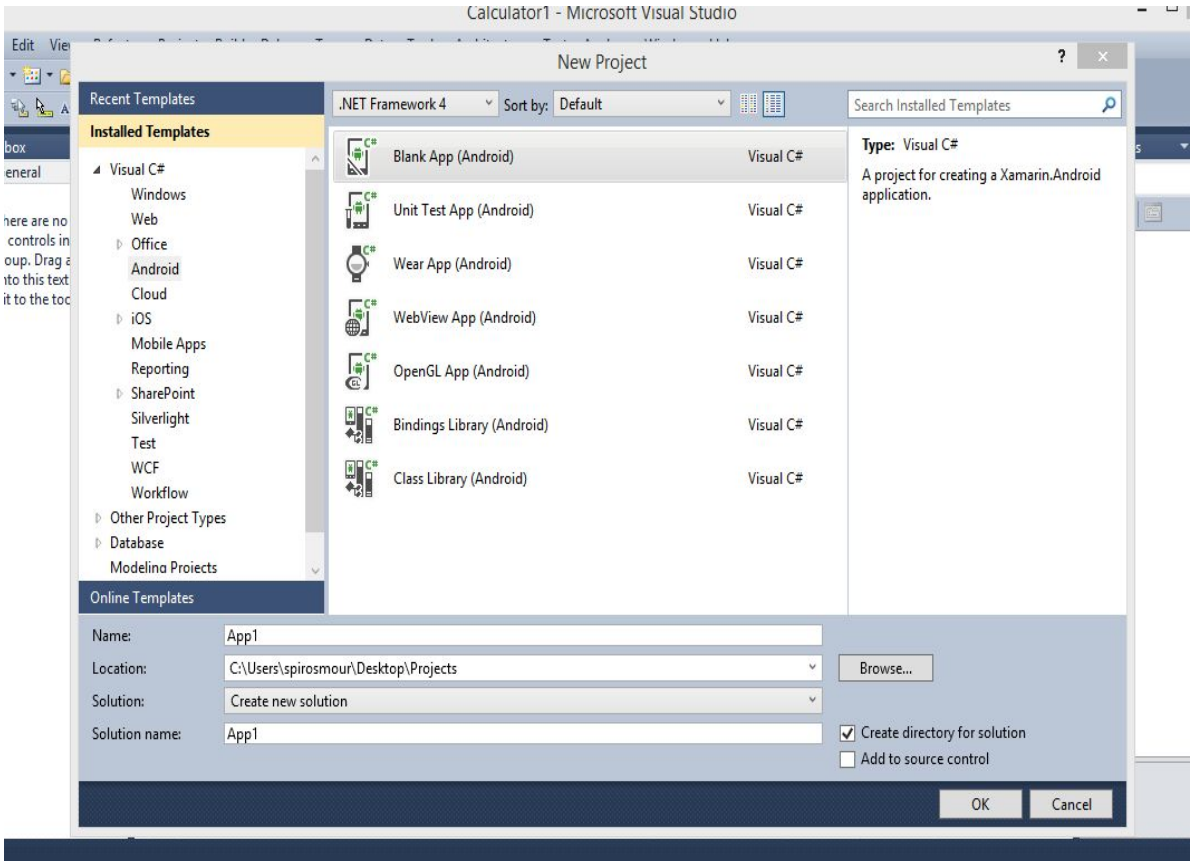


Εικόνα-13. Το διάγραμμα ροής της εφαρμογής

4.2 Ανάπτυξη εφαρμογής C# με χρήση Visual Studio 2010

Για να δημιουργήσουμε ένα καινούργιο Project επιλέγουμε File →New→ Project.

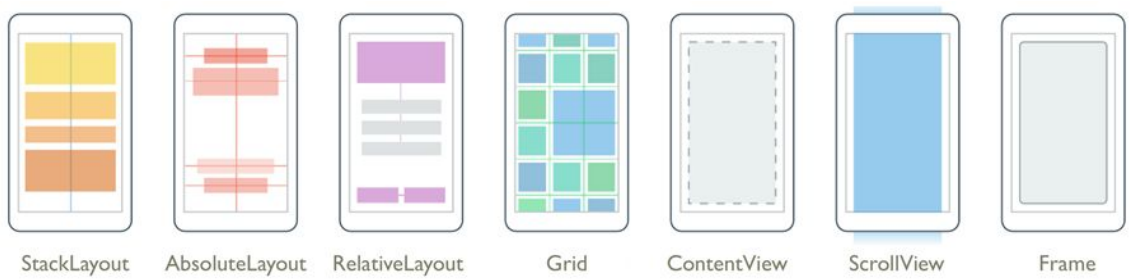
Στο παράθυρο το οποίο εμφανίζεται στην Εικόνα-14 διαλέγουμε από την αριστερή στήλη την επιλογή Visual C# και στην συνέχεια την κατηγορία Android. Έπειτα στο κεντρικό menu που θα εμφανιστεί θα επιλέξουμε την κατηγορία Blank App (Android), θα εισάγουμε το επιθυμητό όνομα του project στο κάτω μέρος του παραθύρου και θα πατήσουμε το κουμπί OK για την δημιουργία του.



Εικόνα-14. Στιγμιότυπο δημιουργίας εφαρμογής Visual Studio 2010

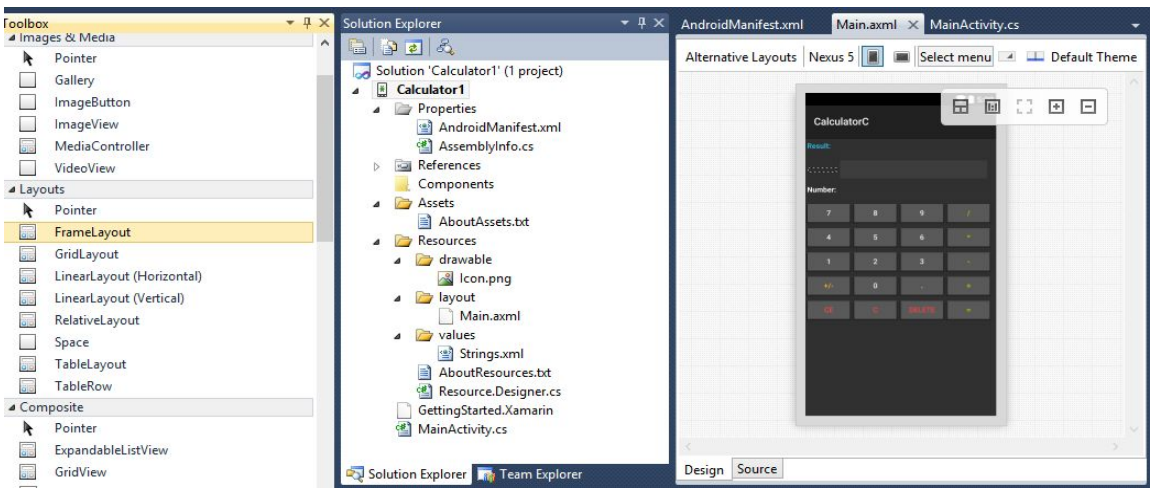
Στο φάκελο resources όπως φαίνεται και στην Εικόνα-12 ορίζεται το layout των activities, οι διάφορες εικόνες και τα strings που

χρησιμοποιούνται στα activities. Το layout αρχείο είναι το αρχείο Main.axml. Το αρχείο αυτό διαμορφώνεται από τον χρήστη και του επιτρέπει να διατάξει με κάποιον από τους παρακάτω τρόπους τα στοιχεία της εφαρμογής του στο πλαίσιο της οθόνης της ανάλογης συσκευής Adnroid.



Εικόνα-15. Τα διάφορα layouts του Visual Studio

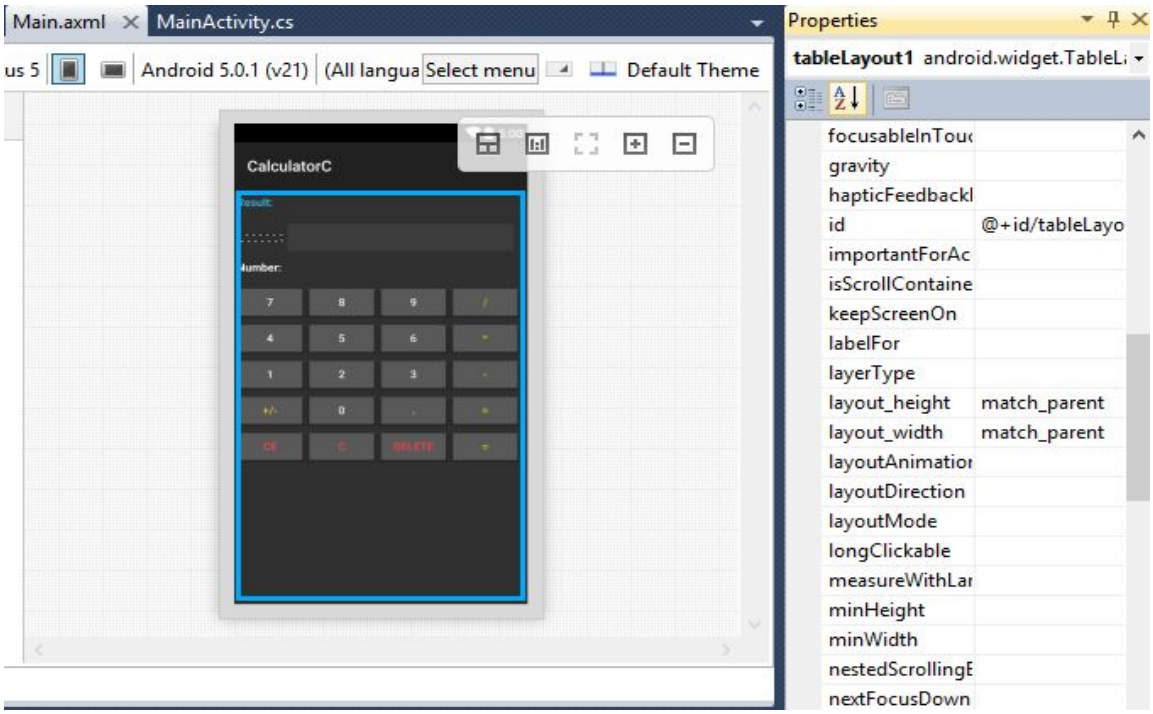
Για την δημιουργία ενός layout επιλέγουμε από την εργαλειοθήκη (toolbox) το ανάλογο layout και το "σέρνουμε" μέσα στο πλαίσιο του desging του Main.axml (διαδικασία drag&drop) όπως φαίνεται στην Εικόνα-16. Στη συγκεκριμένη περίπτωση δουλέψαμε με table layout.



Εικόνα-16. Διαδικασία εισαγωγής layout Visual Studio

Μετά την εισαγωγή όλων των αντικειμένων πάνω στο layout παραμετροποιούμε κάποια από αυτά. Η παραμετροποίηση κάθε

αντικειμένου γίνεται από τα αντίστοιχα πεδία του παραθύρου properties όπως φαίνονται στην Εικόνα-17. Μέσα από τις ιδιότητες κάθε αντικειμένου μπορούμε να καθορίσουμε το όνομα του, τη τιμή του, το χρώμα των χαρακτήρων, το χρώμα πλαισίου, το μέγεθος των χαρακτήρων, τη μορφή (πλάγια, υπογραμμισμένη, έντονη), το πλήθος των δεκαδικών ψηφίων κα. Ο πλήρης κώδικας του αρχείου Main.axml βρίσκεται στο Παράρτημα-Α.

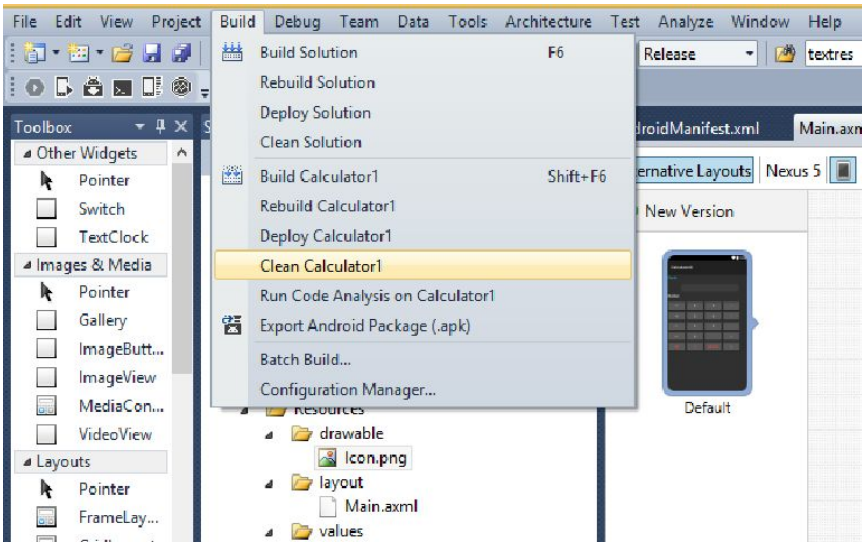


Εικόνα-17. Παράθυρο properties Visual Studio

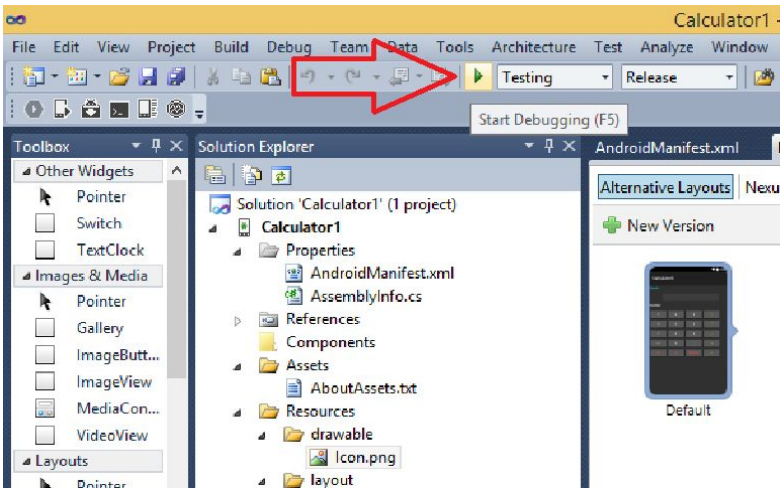
Μετά από την δημιουργία του layout ξεκινάμε τη συγγραφή κώδικα η συγγραφή κώδικα γίνεται στο αρχείο MainActivity.cs. Ο κώδικας στο σύνολο του εμφανίζεται στο Παράρτημα-Β. Για να παραχθεί το τελικό APK αρχείο υπάρχουν δύο τρόποι. Ο ένας όπως φαίνεται και στην Εικόνα-18 είναι να γίνει χειροκίνητα η διαδικασία clean project στην συνέχεια build project και αφού έχουμε εκκινήσει εμείς την εφαρμογή του AVD manager

για την δημιουργία της εικονικής μηχανής να εμφανιστεί η εφαρμογή στη εικονική συσκευή μας.

Ο δεύτερος τρόπος που είναι και πιο απλός και φαίνεται στην Εικόνα-19 είναι να πατήσουμε το κουμπί run και η όλη παραπάνω διαδικασία που περιγράψαμε θα εκτελεστεί αυτόματα και θα εμφανιστεί η εικονική συσκευή μέσα στην οποία θα τρέχει η εφαρμογή.



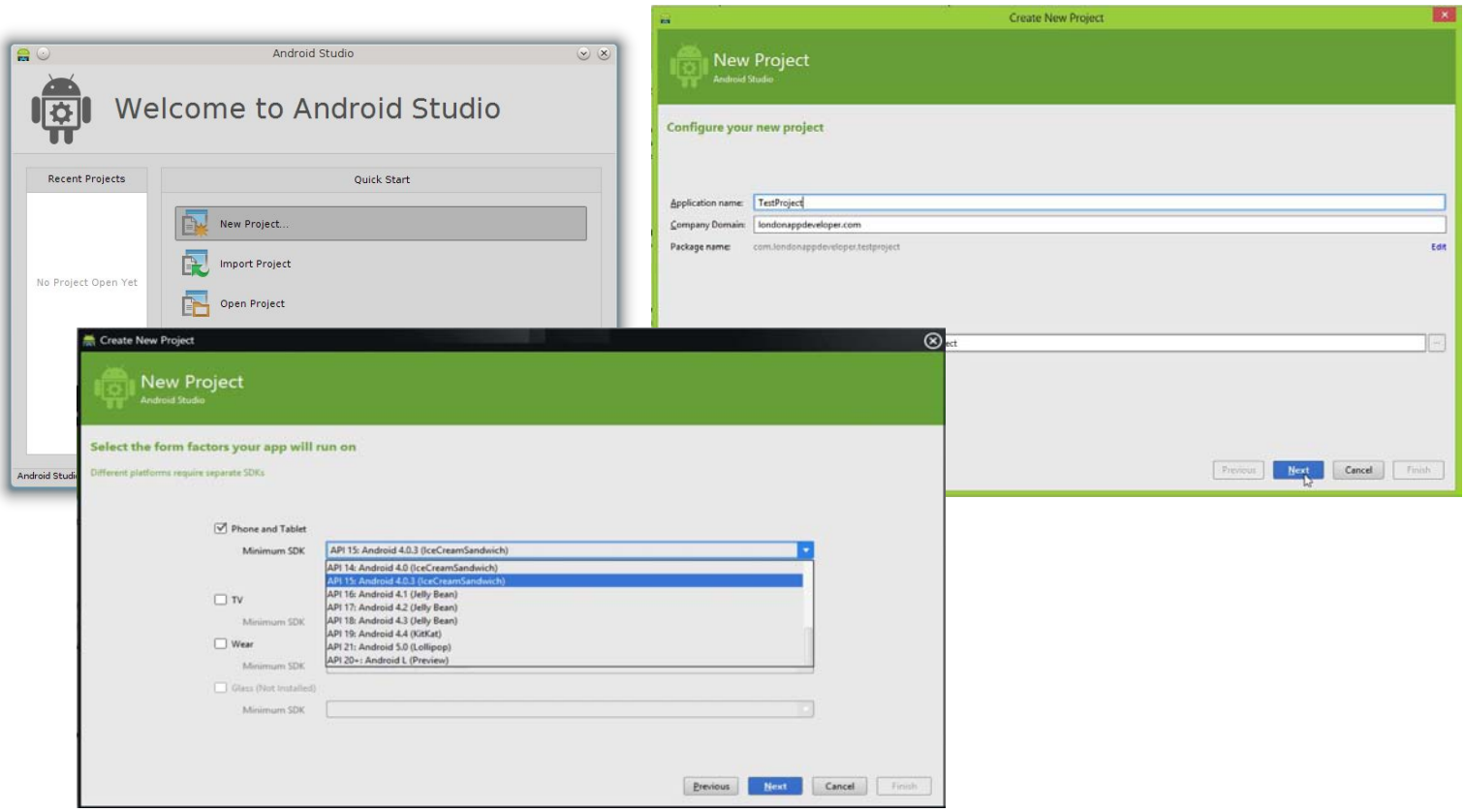
Εικόνα-18. Τρόπος παραγωγής APK αρχείου Visual Studio



Εικόνα-19. Τρόπος παραγωγής APK αρχείου Visual Studio

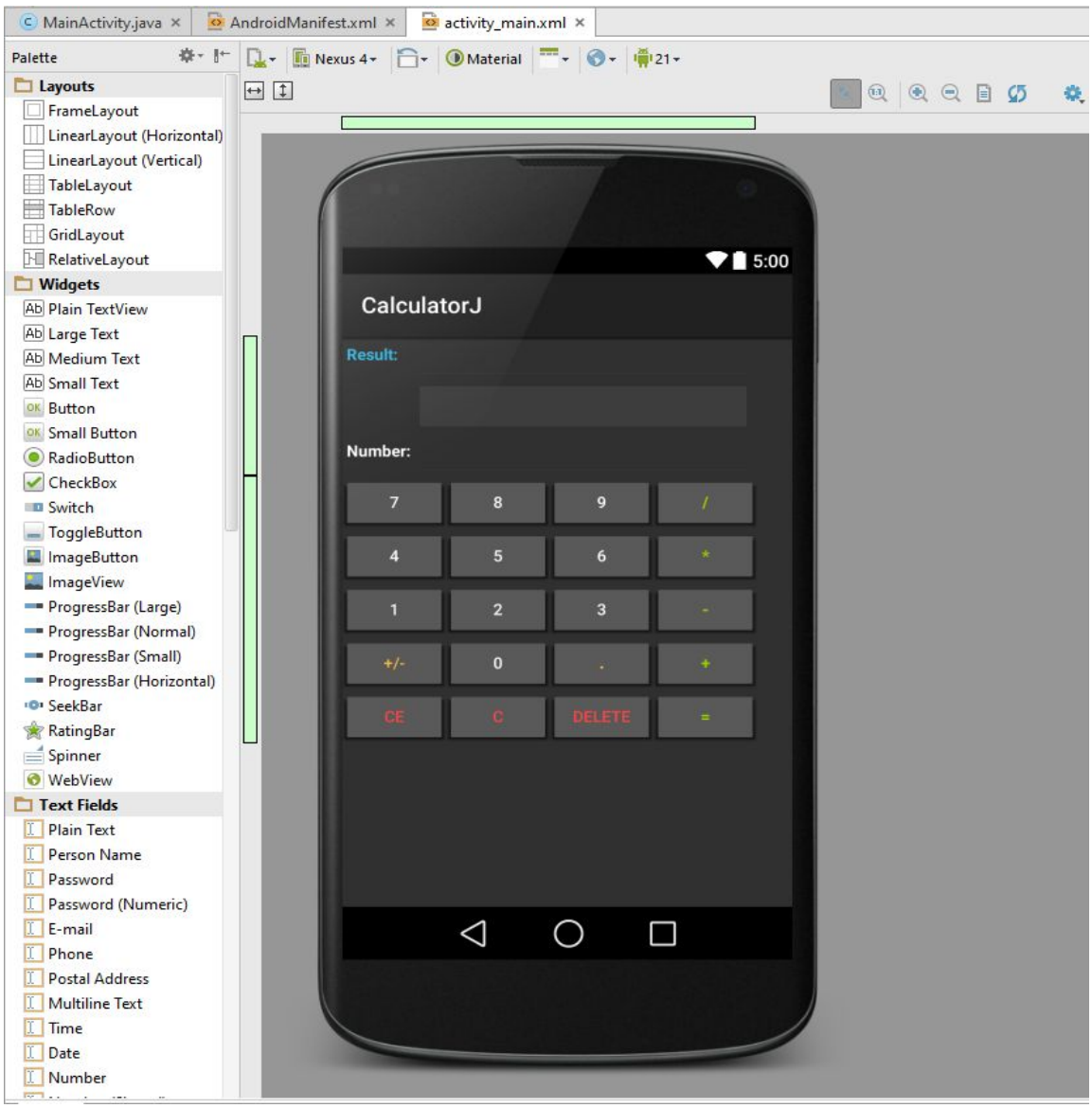
4.3 Ανάπτυξη εφαρμογής Java με χρήση Android Studio

Στο πρώτο παράθυρο που θα εμφανιστεί επιλέγουμε New Project για την δημιουργία μίας καινούργιας εφαρμογής. Στο επόμενο παράθυρο που εμφανίζεται πρέπει να συμπληρώσουμε κάποια στοιχεία για την εφαρμογή όπως Application Name που θα είναι το όνομα της εφαρμογής που θα βλέπει ο χρήστης, το όνομα του Package καθώς και το Company Domain το όνομα του οποίου ενσωματώνεται στο Package Name. Επιλέγουμε τοποθεσία αποθήκευσης του Project και πατάμε το κουμπί Next. Στο επόμενο και τελευταίο παράθυρο που εμφανίζεται πρέπει να επιλέξουμε την ελάχιστη και μέγιστη έκδοση Android στην οποία η εφαρμογή μας θα είναι λειτουργική. Δεν επιλέγουμε τίποτα απο τα TV, Wear και Glass.



Εικόνα-20. Στιγμιότυπο δημιουργίας εφαρμογής Android Studio

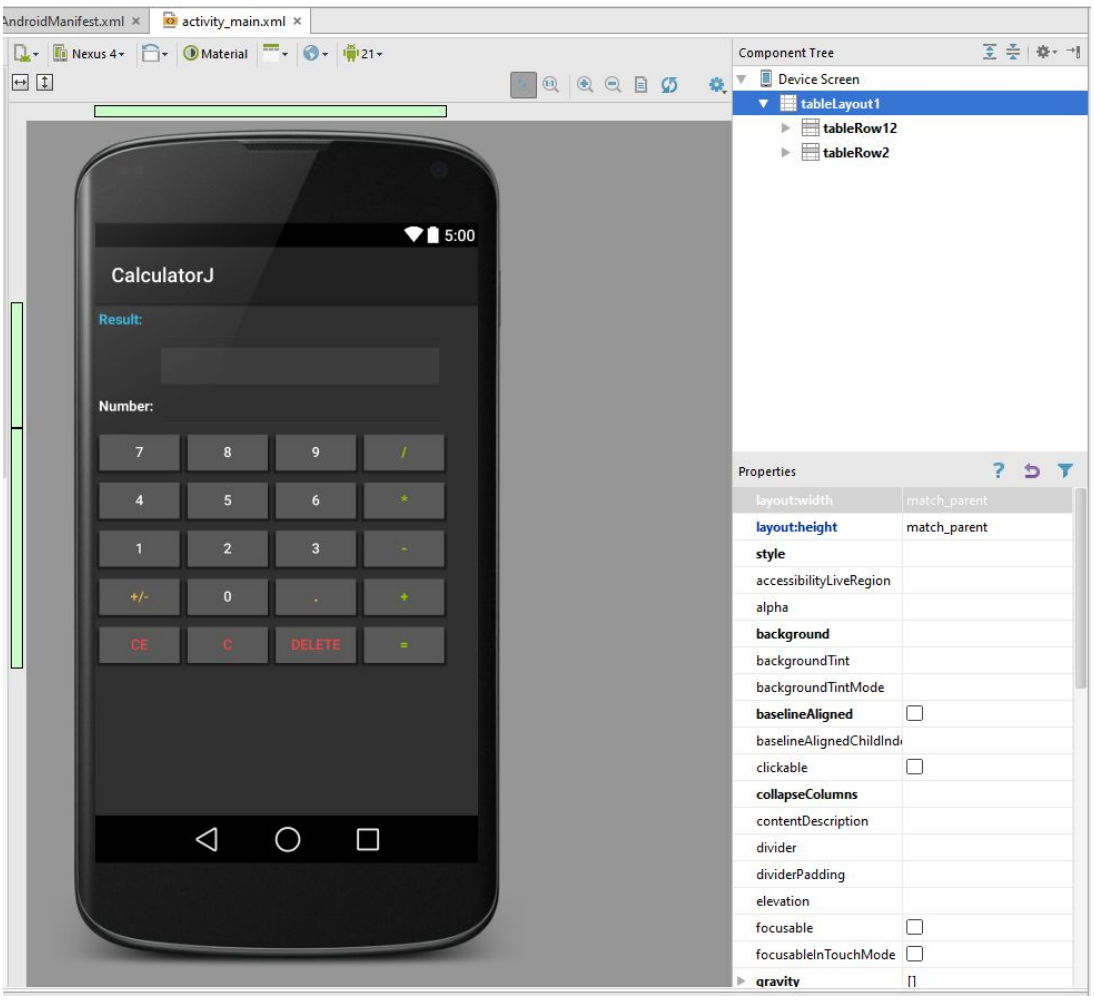
Για την δημιουργία ενός layout επιλέγουμε από την εργαλειοθήκη (palette) το ανάλογο layout και το "σέρνουμε" μέσα στο πλαίσιο του desging του Activity_Main.xml (διαδικασία drag&drop) όπως φαίνεται στην Εικόνα-21. Στη συγκεκριμένη περίπτωση δουλέψαμε με table layout.



Εικόνα-21. Διαδικασία εισαγωγής layout Android Studio

Μετά την εισαγωγή όλων των αντικειμένων πάνω στο layout παραμετροποιούμε κάποια από αυτά. Η παραμετροποίηση κάθε

αντικειμένου γίνεται από τα αντίστοιχα πεδία του παραθύρου properties όπως φαίνονται στην Εικόνα-22. Μέσα από τις ιδιότητες κάθε αντικειμένου μπορούμε να καθορίσουμε το όνομα του, τη τιμή του, το χρώμα των χαρακτήρων, το χρώμα πλαισίου, το μέγεθος των χαρακτήρων, τη μορφή (πλάγια, υπογραμμισμένη, έντονη), το πλήθος των δεκαδικών ψηφίων κα. Ο πλήρης κώδικας του αρχείου Activity_Main.xml βρίσκεται στο Παράρτημα-Α.

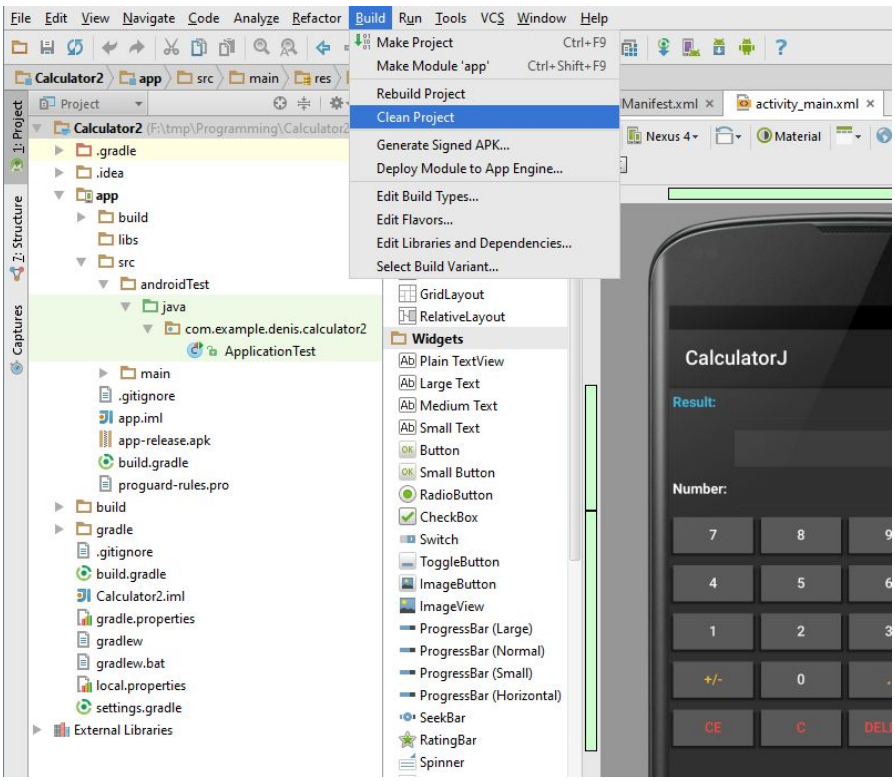


Εικόνα-22. Παράθυρο properties Android Studio

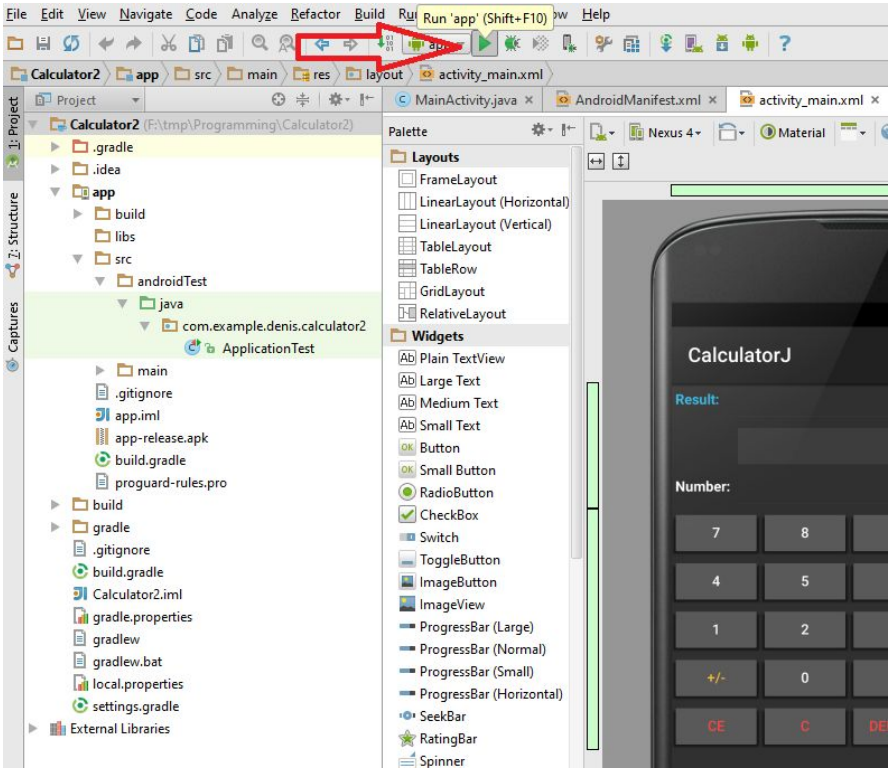
Μετά από την δημιουργία του layout ξεκινάμε τη συγγραφή κώδικα η συγγραφή κώδικα γίνεται στο αρχείο MainActivity.java. Ο κώδικας στο

σύνολο του εμφανίζεται στο Παράρτημα-Γ. Για να παραχθεί το τελικό APK αρχείο υπάρχουν δύο τρόποι. Ο ένας όπως φαίνεται και στην Εικόνα-23 είναι να γίνει χειροκίνητα η διαδικασία clean project, make project και generate signed APK αφού έχουμε εκκινήσει εμείς την εφαρμογή του AVD manager για την δημιουργία της εικονικής μηχανής να εμφανιστεί η εφαρμογή στη εικονική συσκευή μας. Για την παραγωγή πιστοποιημένου APK αρχείου το Android Studio ζητάει την δημιουργία πιστοποιητικού το οποίο περιέχει πληροφορίες του συγγραφέα καθώς και κάποιο κρυφό κωδικό για την προστασία του πηγαίου κώδικα.

Ο δεύτερος τρόπος που είναι και πιο απλός και φαίνεται στην Εικόνα-24 είναι να πατήσουμε το κουμπί run app και η όλη παραπάνω διαδικασία που περιγράψαμε θα εκτελεστεί αυτόματα και θα εμφανιστεί η εικονική συσκευή μέσα στην οποία θα τρέχει η εφαρμογή.



Εικόνα-23. Τρόπος παραγωγής APK αρχείου Android Studio



Εικόνα-24. Τρόπος παραγωγής APK αρχείου Android Studio

5 Συμπεράσματα - Βελτιώσεις

Αρχικά έγινε προσπάθεια της χρήσης της νέας έκδοσης του Visual Studio Community 2015. Χρονικά η κυκλοφορία αυτής της έκδοσης συνέπεσε με το ξεκίνημα της ανάπτυξης του κώδικα της εν λόγω εργασίας. Η έκδοση αυτή περιλαμβάνει όλα τα απαραίτητα εργαλεία Android εφαρμογών σε C#. Πρακτικά ενσωματώνει το Xamarin add-on. Δυστυχώς δεν μπορέσαμε να χρησιμοποιήσουμε την έκδοση αυτήν αν και δίνεται δωρεάν σε όλους απο την Microsoft. Ο κυριότερος λόγος ήταν ότι υπήρχε πολύ μικρός αριθμός τεκμηρίωσης. Η νέα αυτή έκδοση εισάγει πολλές νέες καινοτομίες που έχουν σχέση και με την μορφή του Project αλλά και ολόκληρου του σχεδιαστικού μέρους του συγκεκριμένου IDE.

Προτιμήθηκε λοιπόν μια παλαιότερη έκδοση του Visual Studio η οποία κατά την άποψη μας ήταν περισσότερο "λογική". Σε αυτήν την περίπτωση όλο το Xamarin θα έπρεπε να εγκατασταθεί χειροκίνητα. Όπως ειπώθηκε σε προηγούμενα κεφάλαια η εγκατάσταση του Xamarin μπορεί να γίνει είτε με ένα tool όπου όλα τα απαραίτητα στοιχεία του εγκαθίστανται αυτόματα είτε μπορεί να εγκατασταθούν ένα-ένα χειροκίνητα από τον χρήστη. Αρχικά χρησιμοποιήθηκε η αυτοματοποιημένη εγκατάσταση. το συγκεκριμένο tool του Xamarin συνδέεται σε έναν απομακρυσμένο server της εταιρίας από τον οποίο κατεβάζει ένα-ένα τα απαραίτητα πακέτα που χρειάζεται. Την χρονική στιγμή που επιχειρήθηκε η εγκατάσταση ο συγκεκριμένος server αντιμετώπιζε προβλήματα με αποτέλεσμα η εγκατάσταση μετά από ένα ορισμένο χρονικό διάστημα να διακόπτεται λόγω προβλήματος σε ένα συγκεκριμένο πακέτο. Έτσι επιλέχθηκε η χειροκίνητη εγκατάσταση. Αυτό προσδίδει στην όλη διαδικασία έναν επιπλέον φόρτο αφού ο χρήστης θα πρέπει να ψάξει μόνος του στο

διαδίκτυο ώστε να βρει τα απαραίτητα πακέτα που χρειάζονται και να τα εγκαταστήσει με συγκεκριμένη σειρά. Τα στοιχεία που απαιτούνται για χρήση του add-on του Xamarin καθώς και η σειρά με την οποία πρέπει να γίνουν εγκατάσταση δίνεται από την ίδια εταιρία.

Αφού εγκαταστάθηκε το add-on Xamarin έπρεπε να γίνει η διαδικασία της ενεργοποίησης του. Η διαδικασία αυτή αν και περιγράφεται λεπτομερώς στην σελίδα της εταιρίας δεν ήταν επιτυχής λόγω του ότι ο τρόπος ενεργοποίησης έχει αλλάξει και δεν έχουν ενημερωθεί τα ανάλογα αρχεία. Η ενεργοποίηση τελικά έγινε μέσω ανάλογης επιλογής του Xamarin Studio το οποίο αν και δεν χρησιμοποιήθηκε θελήσαμε να το ερευνήσουμε για εκπαιδευτικούς λόγους.

Οι developer Android εφαρμογών κατά κύριο λόγο χρησιμοποιούν Java και μόλις πριν από ένα χρόνο άρχισαν να χρησιμοποιούν μέσω του Xamarin και C#. Ως εκ τούτου ο αριθμός των sample code όπως και οποιοδήποτε άλλων βοηθημάτων ηλεκτρονικών ή έντυπων θα μπορούσε να χαρακτηριστεί ελάχιστος. Μέσα από ανάλογες εφαρμογές μπορέσαμε να μαζέψουμε τα απαραίτητα στοιχεία για την συγγραφή κώδικα σε C#. Το περιβάλλον ανάπτυξης μπορούμε να το θεωρήσουμε πλήρες σε μέγιστο βαθμό. Και στη σχεδίαση του layout αλλά και στην συγγραφή κώδικα το Visual Studio κρίθηκε σε όλα πολύ καλό. Η δημιουργία του layout μπορεί να γίνει πολύ εύκολα μέσα από κατανοητά μενού (toolbox) και να τροποποιήσει οποιαδήποτε ιδιότητα οποιουδήποτε στοιχείου πάρα πολύ εύκολα. Η συγγραφή του κώδικα έγινε πολύ εύκολα καθώς το Visual Studio διαθέτει την δυνατότητα IntelliSense. Μέσω αυτής δίνεται η δυνατότητα αυτόματης συμπλήρωσης (autocomplete) κατά την πληκτρολόγηση μεθόδων και αντικειμένων.

Η εξομοίωση του προγράμματος έγινε αρχικά μέσω του AVD. Η δημιουργία μιας εικονικής συσκευής παρόμοια με του χρήστη δεν είναι πάντοτε εφικτή. Συγκεκριμένα η Google έχει προκαθορισμένες συσκευές στο AVD αυτές της Nexus (δηλαδή προϊόντων που ανήκουν σε αυτή). Η δημιουργία μιας συσκευής που να είναι πιο κοντά με αυτήν του χρήστη (στην συγκεκριμένη περίπτωση με ένα Samsung Galaxy Note 2) απαιτεί πολλές δοκιμές και μεγάλη παραμετροποίηση. Η χρήση του AVD είναι πολύ χρονοβόρα. Συγκεκριμένα η εικονική συσκευή εκκινεί μετά το πέρας αρκετών λεπτών. Η χρήση της είναι πολύ χρονοβόρα μιας και η εικονική συσκευή αντιδρά στις αλληλεπιδράσεις του χρήστη μετά από καθυστέρηση αρκετών δευτερολέπτων. Εδώ θα πρέπει να επισημάνουμε και την αστάθεια την οποία χαρακτηρίζει το συγκεκριμένο πρόγραμμα. Συγκεκριμένα τις μισές φορές που χρειάστηκε να ανοίξει, είτε κατέρρεε είτε έπρεπε να τερματιστεί από τον χρήστη λόγω κολλήματος. Η συγκεκριμένη συμπεριφορά έχει να κάνει με την μορφή της εφαρμογής AVD. Η εφαρμογή κάνει προσομοίωση ενός ολόκληρου λειτουργικού συστήματος τελείως διαφορετικού με οποιοδήποτε άλλο λειτουργικό σύστημα πχ Windows, MacOS κτλ. Επίσης το hardware που εξομοιώνει είναι τελείως διαφορετικό με αυτό των home υπολογιστών στους οποίους αναπτύσσεται ο κώδικας. Η προσομοίωση του AVD γίνεται με χρήση μόνο ενός πυρήνα της CPU, πράγμα που σημαίνει ότι σε όλα τα νέα πολυπύρηνα συστήματα μένει αναξιοποίητη πολύτιμη επεξεργαστική ισχύς.

Ένας τρόπος για να αποφευχθούν τα παρακάτω να εκτελεστεί μία φορά το AVD και στην συνέχεια να παραμένει ανοιχτό καθ όλη την διάρκεια ανάπτυξης και δοκιμής της εφαρμογής. Κάτι τέτοιο περιόριζε σημαντικά τα προβλήματα κολλήματος και κρεμάσματος της εφαρμογής

αλλά όχι της ταχύτητας της. Για να ξεπεραστεί το πρόβλημα της ταχύτητας δοκιμάστηκε μια άλλη εφαρμογή προσομοίωσης της εφαρμογής, αυτή της Genymotion. Η συγκεκριμένη εφαρμογή προσομοιώνει το Android λειτουργικό μέσω της εφαρμογης προσομοίωσης Virtual Box. Το Genymotion όπως και το Virtual Box είναι ελεύθερες διανομές. Το μεγάλο πλεονέκτημα του Genymotion είναι ότι μέσω του Virtual Box μπορεί και χρησιμοποιεί περισσότερους από έναν πυρήνες της CPU πράγμα που το καθιστά εξαιρετικά γρήγορο. Αντιμετωπίζει αρκετές ασυμβατότητες με λειτουργικά συστήματα τελευταίας γενιάς πχ Windows 10 και γι αυτόν τον λόγο δεν χρησημοποιήθηκε. Συγκεκριμένα μετά από έναν ορισμένο αριθμό εκτελέσεων ο συγκεκριμένος προσομοιωτής χρησιμοποιεί και αυτός έναν μόνο πυρήνα.

Μια εναλλακτική περίπτωση εκτέλεσης του κώδικα είναι αυτή της απευθείας μεταφοράς και εκτέλεσης στην συσκευή μας. Η αναγνώριση της συσκευής γίνεται μέσω του AVD και μέσω εγκατάστασης των ανάλογων USB Drivers κατά την διαδικασία εγκατάστασης του Android SDK. Η διαδικασία αυτή μπόρεσε να λειτουργήσει μόνο στην περίπτωση της ανάπτυξης κώδικα με το Android Studio. Φυσικά η εκτέλεση κώδικα απευθείας στην συσκευή επιφυλάσσει πολλούς κίνδυνους όπως για παράδειγμα η εφαρμογή να παρουσιάσει δυσλειτουργία στο λειτουργικό του τηλεφώνου, στο hardware αυτού η στα δεδομένα του, ανάλογα με την φύση της εφαρμογής.

Κατά την διάρκεια ανάπτυξης της εφαρμογής πολλές φορές για άγνωστους λόγους η συσκευή προσομοίωσης του AVD δεν εμφανιζόταν στο add-on του Xamarin στο Visual Studio. Αυτό έχει ως συνέπεια όταν πατούσαμε το κουμπί του start debugging ή start without debugging το

Visual Studio να αρνείται να την εκτελέσει λόγω έλλειψης target (AVD συσκευής). Το πρόβλημα αυτό αντιμετωπίστηκε εκτελώντας χειροκίνητα την συσκευή προσομοίωσης του AVD καθώς και την δημιουργία apk αρχείου (clean-make-build project)

Το Android Studio είναι ένα περιβάλλον το οποίο μοιάζει εξαιρετικά με το Eclipse περιέχει όπως και το Visual Studio μια αρκετά καλά οργανωμένη εργαλειοθήκη (toolbox) και παραμετροποίησης των αντικειμένων μέσα από το properties window. Περιέχει πολλούς τρόπους εμφάνισης της δομής του project που πολλές φορές το κάνει περισσότερο δύσχρηστο παρά φιλικό. Η συγγραφή του κώδικα γίνεται εξίσου εύκολα με αυτή του Visual Studio. Συγκεκριμένα να αναφέρουμε ότι κατά την εισαγωγή μίας νέας μεθόδου το περιβάλλον μας προέτρεπε για την ενσωμάτωση της στην κεφαλίδα του προγράμματος. Το περιβάλλον προτρέπει τον χρήστη καθ όλη την διάρκεια συγγραφής του κώδικα για errors και warnings όπως επίσης και για τρόπους βελτιστοποίησης του. Η τελευταία αυτή δυνατότητα δεν έχει πάντα τα αναμενόμενα αποτελέσματα. Συγκεκριμένα σε πολλά σημεία του προγράμματος το περιβάλλον αρχικά προέτρεπε στην δημιουργία και χρήση ξεχωριστής μεθόδου και στην συνέχεια την απέτρεπε. Η εκτέλεση του κώδικα με την χρήση της εικονικής συσκευής του AVD στην περίπτωση του Android Studio δεν αντιμετώπισε κανένα απολύτως πρόβλημα. Το ίδιο απροβλημάτιστη ήταν και η εκτέλεση της εφαρμογής απευθείας στην συσκευή μέσω USB σύνδεσης του AVD.

Παρατηρήθηκε ότι αν και χρησιμοποιήθηκαν και στις δύο περιπτώσεις τα ίδια αντικείμενα για την δημιουργία του layout τα δύο IDE παρήγαγαν ελαφρός διαφοροποιημένο xml αρχείο. Για ομοιομορφία και για καλύτερη σύγκριση της παραγόμενης εφαρμογής προτιμήθηκε ο κώδικας

που παρήγαγε το Android Studio ο οποίος ακολουθεί πιστότερα το xml στάνταρ. Ο κώδικας του xml του Android Studio έγινε copy-paste στο Visual Studio χωρίς κανένα απολύτως πρόβλημα. Επίσης κάποιες επιπλέον λειτουργίες που θα μπορούσαν να φανούν χρήσιμες στον developer όπως για παράδειγμα η εκ των υστέρων μετονομασία του project ή/και του ονόματος της εφαρμογής και του apk αποδείχθηκαν πολύ ευκολότερες στην περίπτωση του Visual Studio. Η παραγωγή του apk έγινε γρηγορότερα στο Android Studio απ ότι στο Visual Studio. Κάτι τέτοιο είναι απολύτως φυσιολογικό αφού, αν και τα δύο προϋποθέτουν την χρήση του Android SDK, στην περίπτωση του Visual Studio το Xamarin θα έπρεπε να μετατρέψει τον C# κώδικα.

Η ανάπτυξη του κώδικα σε C# φάνηκε περισσότερο απλούστερη. Αυτό έχει να κάνει κυρίως με την δομή της γλώσσας η οποία παραλείπει πολλά απο τα στοιχεία της Java όπως φάνηκε σε τμήματα κώδικα προηγούμενων κεφαλαίων. Προτιμήθηκε η χρήση δημιουργίας μεθόδων. Αν και η κλήση των μεθόδων είναι μία χρονοβόρα διαδικασία κατά την εκτέλεση του προγράμματος, θέλαμε να διαπιστώσουμε πόσο εύκολα γίνεται κάτι τέτοιο ώστε το πρόγραμμα να εμφανίζεται περισσότερο δομημένο.

Η μόνη διαφορά του κώδικα C# με αυτόν της Java ήταν αντίστοιχα η χρήση handlers έναντι listeners. Αν και θεωρητικά οι δύο αυτοί μέθοδοι είναι πανομοιότυποι στην πράξη στην περίπτωση των listeners θα έπρεπε να εισαχθεί ή μια αργή εντολή ελέγχου τύπου case είτε μια λιγότερο χρονοβόρα τύπου if. Στην περίπτωση των handler κάτι τέτοιο αποφεύγεται. Μία ακόμα διαφοροποίηση ήταν η περίπτωση της σωστής εμφάνισης των δεκαδικών αποτελεσμάτων. Χωρίς καμία παραμετροποίηση οι δεκαδικοί

αριθμοί στην C# εμφανίζονται εξ ορισμού στην στάνταρ μορφή. Αντίθετα στην Java οι δεκαδικοί αριθμοί εμφανίζονται εξ ορισμού στην scientific μορφή. Η αλλαγή σε στάνταρ μορφή κάθε άλλο παρά εύκολη υπόθεση αποδείχθηκε. Αυτό έχει να κάνει με το αιώνιο πρόβλημα της Java της εισαγωγής και εμφάνισης στοιχείων του χρήστη.

Για την παραγωγή πιστοποιημένου APK αρχείου το Android Studio ζητάει την δημιουργία πιστοποιητικού το οποίο περιέχει πληροφορίες του συγγραφέα καθώς και κάποιο κρυφό κωδικό για την προστασία του πηγαίου κώδικα. Αντίθετα το Visual Studio έχει αυτοματοποιήσει αυτήν την διαδικασία και δεν ζητάει καμία απολύτως πληροφορία απο τον χρήστη.

Κατά την παραγωγή των APK αρχείων παρατηρήθηκε μεγάλη διαφοροποίηση στο μέγεθος των παραγόμενων αρχείων. Συγκεκριμένα το APK που παράχθηκε με C# είχε μέγεθος 3.5MB. Αντίθετα αυτό του Android Studio είχε μέγεθος 30KB. Η μεγάλη αυτή διαφορά έγκειται στο ότι το Xamarin μεταγλωττίζει τον C# κώδικα σε Java κώδικα, που είναι στην ουσία ο κώδικας των APK αρχείων.

Κατά την διάρκεια εκτέλεσης των δύο εφαρμογών στην συσκευή μας παρατηρήθηκε ελαφρώς ταχύτερη εκτέλεση της εφαρμογής που ήτανε γραμμένη σε Java. Κάτι τέτοιο θεωρείται φυσιολογικό μιας και παρόλο της εισαγωγής των χρονικών βεβαρημένων if στην Java, ο εκτελέσιμος κώδικας ήταν κατά πολύ μικρότερος. Αυτό προσδίδει στην εφαρμογή της Java πού ταχύτερους χρόνους κατά το άνοιγμα της εφαρμογής αλλά και κατά την διάρκεια εκτέλεσης της. Αν και η εφαρμογή δεν επιτελούσε κάποια βεβαρημένη εργασία η διαφορά αυτή μπορούσε να γίνει αντιληπτή.

Η εφαρμογή σε όποια μορφή και αν επιλεχθεί μπορεί να βελτιωθεί ως προς το layout, το πλήθος των αριθμητικών πράξεων,των εμπλουτισμό σε μηνύματα βοήθειας κτλ.

6 Αναφορές - Βιβλιογραφία

[1] Oracle, *Java SE Development Kit 8 Downloads*,

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html, [last access 05/10/2015]

[2] Android, *Android Studio and SDK Tools,*

https://developer.android.com/sdk/index.html, [last access 05/10/2015]

[3] Android, *Android NDK,* https://developer.android.com/tools/sdk/ndk/, [last access 05/10/2015]

[4] MonoProject, *GtkSharp*,

http://www.mono-project.com/download/, [last access 05/10/2015]

[5] Xamarin, *Xamarin Studio*,

https://forums.xamarin.com/discussion/47935/stable-release-xamarin-studio-5-9-5-cycle-5-service-release-3, [last access 05/10/2015]

[6] Xamarin, *Xamarin Visual Studio,*

https://forums.xamarin.com/discussion/47936/stable-release-xamarin-android-5-1-5-cycle-5-service-release-3, [last access 05/10/2015]

Παράρτημα-Α (Main.axml, Activity_Main.xml)

```
      <?xml version="1.0" encoding="utf-8"?>
      <TableLayout
xmlns:p1="http://schemas.android.com/apk/res/android"
          p1:layout_width="match_parent"
          p1:layout_height="match_parent"
          p1:id="@+id/tableLayout1">
          <TableRow
              p1:id="@+id/tableRow12"
              p1:layout_height="match_parent"
              p1:layout_width="match_parent">
              <TableLayout
                  p1:layout_column="0"
                  p1:id="@+id/tableLayout3">
                  <TableRow
                      p1:id="@+id/tableRow13"
                      p1:layout_width="match_parent"
                      p1:layout_height="wrap_content">
                      <TextView
                          p1:text=" Result:"
                          p1:layout_column="0"
                          p1:id="@+id/textView9"
                          p1:enabled="false"

p1:textColor="@android:color/holo_blue_light"
                          p1:textStyle="bold" />
                      <EditText
                          p1:inputType="number"
                          p1:layout_column="1"
                          p1:id="@+id/editTextRes"

p1:textColor="@android:color/holo_blue_light"
                          p1:cursorVisible="false"
```

```
                        p1:enabled="false"
                        p1:textStyle="bold"
                        p1:maxLength="24" />
                </TableRow>
                <TableRow
                    p1:id="@+id/tableRow14"
                    p1:layout_height="wrap_content"
                    p1:layout_width="wrap_content">
                    <Button
                        p1:layout_column="1"
                        p1:id="@+id/buttonAri"
                        p1:enabled="false"

p1:textColor="@android:color/holo_green_light"
                        p1:textStyle="bold" />
                </TableRow>
                <TableRow
                    p1:id="@+id/tableRow15"
                    p1:layout_width="match_parent"
                    p1:layout_height="wrap_content">
                    <TextView
                        p1:text=" Number: "
                        p1:layout_column="0"
                        p1:id="@+id/textView10"
                        p1:enabled="false"

p1:textColor="@android:color/background_light"
                        p1:textStyle="bold" />
                    <EditText
                        p1:inputType="number"
                        p1:layout_column="1"
                        p1:id="@+id/editTextNum"
                        p1:editable="false"
                        p1:enabled="false"
```

```
p1:textColor="@android:color/background_light"
                        p1:layout_width="285.5dp"
                        p1:maxLength="12" />
                </TableRow>
            </TableLayout>
        </TableRow>
        <TableRow
            p1:id="@+id/tableRow2"
            p1:layout_width="match_parent"
            p1:layout_height="match_parent">
            <TableLayout
                p1:layout_column="0"
                p1:id="@+id/tableLayout2"
                p1:layout_width="match_parent"
                p1:layout_height="wrap_content">
                <TableRow
                    p1:id="@+id/tableRow4"
                    p1:layout_width="match_parent"
                    p1:layout_height="wrap_content">
                    <Button
                        p1:text="7"
                        p1:layout_column="0"
                        p1:id="@+id/button7"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="8"
                        p1:layout_column="1"
                        p1:id="@+id/button8"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="9"
```

```
                        p1:layout_column="2"
                        p1:id="@+id/button9"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="/"
                        p1:layout_column="3"
                        p1:id="@+id/buttonDiv"

p1:textColor="@android:color/holo_green_light"
                        p1:textStyle="bold"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                </TableRow>
                <TableRow
                    p1:id="@+id/tableRow5"
                    p1:layout_width="match_parent"
                    p1:layout_height="wrap_content">
                    <Button
                        p1:text="4"
                        p1:layout_column="0"
                        p1:id="@+id/button4"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="5"
                        p1:layout_column="1"
                        p1:id="@+id/button5"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="6"
                        p1:layout_column="2"
                        p1:id="@+id/button6"
```

```
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="*"
                        p1:layout_column="3"
                        p1:id="@+id/buttonMul"

p1:textColor="@android:color/holo_green_light"
                        p1:textStyle="bold"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                </TableRow>
                <TableRow
                    p1:id="@+id/tableRow6"
                    p1:layout_width="match_parent"
                    p1:layout_height="wrap_content">
                    <Button
                        p1:text="1"
                        p1:layout_column="0"
                        p1:id="@+id/button1"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="2"
                        p1:layout_column="1"
                        p1:id="@+id/button2"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="3"
                        p1:layout_column="2"
                        p1:id="@+id/button3"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
```

```
                    <Button
                        p1:text="-"
                        p1:layout_column="3"
                        p1:id="@+id/buttonSub"

p1:textColor="@android:color/holo_green_light"
                        p1:textStyle="bold"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                </TableRow>
                <TableRow
                    p1:id="@+id/tableRow7"
                    p1:layout_width="match_parent"
                    p1:layout_height="wrap_content">
                    <Button
                        p1:text="+/-"
                        p1:layout_column="0"
                        p1:id="@+id/buttonSig"

p1:textColor="@android:color/holo_orange_light"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="0"
                        p1:layout_column="1"
                        p1:id="@+id/button0"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="."
                        p1:layout_column="2"
                        p1:id="@+id/buttonDot"

p1:textColor="@android:color/holo_orange_light"
```

```
                        p1:textStyle="bold"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="+"
                        p1:layout_column="3"
                        p1:id="@+id/buttonAdd"

p1:textColor="@android:color/holo_green_light"
                        p1:textStyle="bold"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                </TableRow>
                <TableRow
                    p1:id="@+id/tableRow10"
                    p1:layout_width="match_parent"
                    p1:layout_height="wrap_content">
                    <Button
                        p1:text="CE"
                        p1:layout_column="0"
                        p1:id="@+id/buttonCE"

p1:textColor="@android:color/holo_red_light"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="C"
                        p1:layout_column="1"
                        p1:id="@+id/buttonC"

p1:textColor="@android:color/holo_red_light"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
```

```
                        p1:text="Delete"
                        p1:layout_column="2"
                        p1:id="@+id/buttonDel"

p1:textColor="@android:color/holo_red_light"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                    <Button
                        p1:text="="
                        p1:layout_column="3"
                        p1:id="@+id/buttonRes"

p1:textColor="@android:color/holo_green_light"
                        p1:textStyle="bold"
                        p1:layout_width="match_parent"
                        p1:layout_height="wrap_content" />
                </TableRow>
            </TableLayout>
        </TableRow>
    </TableLayout>
```

Παράρτημα-Β (MainActivity.cs)

```
using System;
using Android.App;
using Android.Content;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using Android.OS;

namespace Calculator1
{
    [Activity(Label = "CalculatorC", MainLauncher = true,
Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        EditText editTextRes;
        EditText editTextNum;
        Button buttonAdd;
        Button buttonMul;
        Button buttonSub;
        Button buttonDiv;
        Button buttonAri;

        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);

            // Display Layout
            SetContentView(Resource.Layout.Main);

            //Create objects by Main.axml Id
            editTextRes =
FindViewById<EditText>(Resource.Id.editTextRes);
            editTextNum =
FindViewById<EditText>(Resource.Id.editTextNum);
            Button button0 =
FindViewById<Button>(Resource.Id.button0);
            Button button1 =
FindViewById<Button>(Resource.Id.button1);
            Button button2 =
FindViewById<Button>(Resource.Id.button2);
            Button button3 =
FindViewById<Button>(Resource.Id.button3);
            Button button4 =
FindViewById<Button>(Resource.Id.button4);
```

```
            Button button5 =
FindViewById<Button>(Resource.Id.button5);
            Button button6 =
FindViewById<Button>(Resource.Id.button6);
            Button button7 =
FindViewById<Button>(Resource.Id.button7);
            Button button8 =
FindViewById<Button>(Resource.Id.button8);
            Button button9 =
FindViewById<Button>(Resource.Id.button9);
            Button buttonSig =
FindViewById<Button>(Resource.Id.buttonSig);
            Button buttonDot =
FindViewById<Button>(Resource.Id.buttonDot);
            Button buttonCE =
FindViewById<Button>(Resource.Id.buttonCE);
            Button buttonC =
FindViewById<Button>(Resource.Id.buttonC);
            Button buttonDel =
FindViewById<Button>(Resource.Id.buttonDel);
            buttonAdd =
FindViewById<Button>(Resource.Id.buttonAdd);
            buttonMul =
FindViewById<Button>(Resource.Id.buttonMul);
            buttonSub =
FindViewById<Button>(Resource.Id.buttonSub);
            buttonDiv =
FindViewById<Button>(Resource.Id.buttonDiv);
            buttonAri =
FindViewById<Button>(Resource.Id.buttonAri);
            Button buttonRes =
FindViewById<Button>(Resource.Id.buttonRes);

            //Create events on click
            button0.Click += new EventHandler(button0_Click);
            button1.Click += new EventHandler(button1_Click);
            button2.Click += new EventHandler(button2_Click);
            button3.Click += new EventHandler(button3_Click);
            button4.Click += new EventHandler(button4_Click);
            button5.Click += new EventHandler(button5_Click);
            button6.Click += new EventHandler(button6_Click);
            button7.Click += new EventHandler(button7_Click);
            button8.Click += new EventHandler(button8_Click);
            button9.Click += new EventHandler(button9_Click);
            buttonC.Click += new EventHandler(buttonC_Click);
            buttonCE.Click += new
EventHandler(buttonCE_Click);
            buttonDel.Click += new
EventHandler(buttonDel_Click);
```

```
            buttonAdd.Click += new
EventHandler(buttonAdd_Click);
            buttonSub.Click += new
EventHandler(buttonSub_Click);
            buttonMul.Click += new
EventHandler(buttonMul_Click);
            buttonDiv.Click += new
EventHandler(buttonDiv_Click);
            buttonRes.Click += new
EventHandler(buttonRes_Click);
            buttonSig.Click += new
EventHandler(buttonSig_Click);
            buttonDot.Click += new
EventHandler(buttonDot_Click);
        }

        void button0_Click(object sender, EventArgs e)
        {
            TypeNum("0");
        }

        void button1_Click(object sender, EventArgs e)
        {
            TypeNum("1");
        }

        void button2_Click(object sender, EventArgs e)
        {
            TypeNum("2");
        }

        void button3_Click(object sender, EventArgs e)
        {
            TypeNum("3");
        }

        void button4_Click(object sender, EventArgs e)
        {
            TypeNum("4");
        }

        void button5_Click(object sender, EventArgs e)
        {
            TypeNum("5");
        }

        void button6_Click(object sender, EventArgs e)
        {
            TypeNum("6");
```

```
}

void button7_Click(object sender, EventArgs e)
{
    TypeNum("7");
}

void button8_Click(object sender, EventArgs e)
{
    TypeNum("8");
}

void button9_Click(object sender, EventArgs e)
{
    TypeNum("9");
}

void buttonC_Click(object sender, EventArgs e)
{
    ClearResNum();
}

void buttonCE_Click(object sender, EventArgs e)
{
    ClearNum();
}

void buttonDel_Click(object sender, EventArgs e)
{
    if (editTextNum.Length() > 1)
        DelNum();
    else
        ClearNum();
}

void buttonAdd_Click(object sender, EventArgs e)
{
    TypeAri("+");
}

void buttonSub_Click(object sender, EventArgs e)
{
    TypeAri("-");
}

void buttonMul_Click(object sender, EventArgs e)
{
    TypeAri("*");
}
```

```
        void buttonDiv_Click(object sender, EventArgs e)
        {
            TypeAri("/");
        }

        void buttonRes_Click(object sender, EventArgs e)
        {
            Calculate(buttonAri.Text);
            TypeAri("=");
        }

        void buttonSig_Click(object sender, EventArgs e)
        {
            ReverseNum();
        }

        void buttonDot_Click(object sender, EventArgs e)
        {
            AddDotNum();
        }

        private void TypeNum(string N)
        {
            if (editTextNum.Text!="" &&
editTextNum.Text!="0")
                editTextNum.Text += N;
            else
                editTextNum.Text = N;
        }

        private void ClearResNum()
        {
            editTextRes.Text = "";
            editTextNum.Text = "";
            buttonAri.Text = "";
        }

        private void ClearNum()
        {
            editTextNum.Text = "";
        }

        private void DelNum()
        {
            editTextNum.Text =
editTextNum.Text.Remove(editTextNum.Length() - 1);
        }
```

```
        private void TypeAri(string A)
        {
            if (buttonAri.Text != "=")
            {
                Calculate(buttonAri.Text);
                buttonAri.Text = A;
            }
            else
            {
                buttonAri.Text = A;
                Calculate(buttonAri.Text);
            }
        }

        private void Calculate(string C)
        {
            if (editTextRes.Text != "" && editTextNum.Text !=
"")
            {
                if (C != "" && editTextNum.Text != "")
                    if (C == "+")
                        editTextRes.Text =
Convert.ToString(Convert.ToDecimal(editTextRes.Text) +
Convert.ToDecimal(editTextNum.Text));
                    else if (C == "-")
                        editTextRes.Text =
Convert.ToString(Convert.ToDecimal(editTextRes.Text) -
Convert.ToDecimal(editTextNum.Text));
                    else if (C == "*")
                        editTextRes.Text =
Convert.ToString(Convert.ToDecimal(editTextRes.Text) *
Convert.ToDecimal(editTextNum.Text));
                    else if (C == "/" && editTextNum.Text !=
"0")
                        editTextRes.Text =
Convert.ToString(Convert.ToDecimal(editTextRes.Text) /
Convert.ToDecimal(editTextNum.Text));
                    else if (C == "/" && editTextNum.Text ==
"0")
                    {
                        ClearResNum();
                        Toast.MakeText(this, "Err: Divide by
zero", ToastLength.Short).Show();
                    }
            }
            else if (editTextNum.Text != "")
                editTextRes.Text = editTextNum.Text;
            editTextNum.Text = "";
        }
```

```
        private void ReverseNum()
        {
            if (editTextNum.Text!="" &&
editTextNum.Text!="0")
                editTextNum.Text =
Convert.ToString(Convert.ToDecimal(editTextNum.Text) * -1);
        }

        private void AddDotNum()
        {
            if (editTextNum.Text != "" &&
!editTextNum.Text.Contains("."))
                editTextNum.Text += ".";
        }

    }
}
```

Παράρτημα-Γ (MainActivity.java)

```
package com.example.denis.calculator2;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.math.BigDecimal;

public class MainActivity extends Activity implements
View.OnClickListener {

      EditText editTextRes;
      EditText editTextNum;
      Button buttonAdd;
      Button buttonMul;
      Button buttonSub;
      Button buttonDiv;
      Button buttonAri;

      @Override
      protected void onCreate(Bundle savedInstanceState) {
      // Display Layout
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      //Create objects by activity_main.xml Id
      editTextRes = (EditText)findViewById(R.id.editTextRes);
      editTextNum = (EditText)findViewById(R.id.editTextNum);
      Button button0 = (Button)findViewById(R.id.button0);
      Button button1 = (Button)findViewById(R.id.button1);
      Button button2 = (Button)findViewById(R.id.button2);
      Button button3 = (Button)findViewById(R.id.button3);
      Button button4 = (Button)findViewById(R.id.button4);
      Button button5 = (Button)findViewById(R.id.button5);
      Button button6 = (Button)findViewById(R.id.button6);
      Button button7 = (Button)findViewById(R.id.button7);
      Button button8 = (Button)findViewById(R.id.button8);
      Button button9 = (Button)findViewById(R.id.button9);
      Button buttonSig = (Button)findViewById(R.id.buttonSig);
      Button buttonDot = (Button)findViewById(R.id.buttonDot);
      Button buttonCE = (Button)findViewById(R.id.buttonCE);
      Button buttonC = (Button)findViewById(R.id.buttonC);
      Button buttonDel = (Button)findViewById(R.id.buttonDel);
```

```
buttonAdd = (Button)findViewById(R.id.buttonAdd);
buttonMul = (Button)findViewById(R.id.buttonMul);
buttonSub = (Button)findViewById(R.id.buttonSub);
buttonDiv = (Button)findViewById(R.id.buttonDiv);
buttonAri = (Button)findViewById(R.id.buttonAri);
Button buttonRes = (Button)findViewById(R.id.buttonRes);

//Create events on click
button0.setOnClickListener(this);
button1.setOnClickListener(this);
button2.setOnClickListener(this);
button3.setOnClickListener(this);
button4.setOnClickListener(this);
button5.setOnClickListener(this);
button6.setOnClickListener(this);
button7.setOnClickListener(this);
button8.setOnClickListener(this);
button9.setOnClickListener(this);
buttonC.setOnClickListener(this);
buttonCE.setOnClickListener(this);
buttonDel.setOnClickListener(this);
buttonAdd.setOnClickListener(this);
buttonSub.setOnClickListener(this);
buttonMul.setOnClickListener(this);
buttonDiv.setOnClickListener(this);
buttonRes.setOnClickListener(this);
buttonSig.setOnClickListener(this);
buttonDot.setOnClickListener(this);
}

@Override
public void onClick(View view) {
if (view.getId()==R.id.button0)
     TypeNum("0");
else if (view.getId()==R.id.button1)
     TypeNum("1");
else if (view.getId()==R.id.button2)
     TypeNum("2");
else if (view.getId()==R.id.button3)
     TypeNum("3");
else if (view.getId()==R.id.button4)
     TypeNum("4");
else if (view.getId()==R.id.button5)
     TypeNum("5");
else if (view.getId()==R.id.button6)
     TypeNum("6");
else if (view.getId()==R.id.button7)
     TypeNum("7");
else if (view.getId()==R.id.button8)
```

```
            TypeNum("8");
      else if (view.getId()==R.id.button9)
            TypeNum("9");
      else if (view.getId()==R.id.buttonC)
            ClearResNum();
      else if (view.getId()==R.id.buttonCE)
            ClearNum();
      else if (view.getId()==R.id.buttonDel)
      {
            if (editTextNum.length() > 1)
                  DelNum();
            else
                  ClearNum();
      }
      else if (view.getId()==R.id.buttonAdd)
            TypeAri("+");
      else if (view.getId()==R.id.buttonSub)
            TypeAri("-");
      else if (view.getId()==R.id.buttonMul)
            TypeAri("*");
      else if (view.getId()==R.id.buttonDiv)
            TypeAri("/");
      else if (view.getId()==R.id.buttonRes)
      {
            Calculate(buttonAri.getText().toString());
            TypeAri("=");
      }
      else if (view.getId()==R.id.buttonSig)
            ReverseNum();
      else if (view.getId()==R.id.buttonDot)
            AddDotNum();
      }

      private void TypeNum(String N) {
      if (!editTextNum.getText().toString().equals("") &&
!editTextNum.getText().toString().equals("0"))

editTextNum.setText(editTextNum.getText().toString()+N);
      else
            editTextNum.setText(N);
      }

      private void ClearResNum() {
      editTextRes.setText("");
      editTextNum.setText("");
      buttonAri.setText("");
      }

      private void ClearNum() {
```

```
      editTextNum.setText("");
      }

      private void DelNum() {

editTextNum.setText(editTextNum.getText().toString().substrin
g(0, editTextNum.length() - 1));
      }

      private void TypeAri(String A) {
      if (!buttonAri.getText().toString().equals("="))
      {
            Calculate(buttonAri.getText().toString());
            buttonAri.setText(A);
      }
      else
      {
            buttonAri.setText(A);
            Calculate(buttonAri.getText().toString());
      }
      }

      private void Calculate(String C) {
      if (!editTextRes.getText().toString().equals("") &&
!editTextNum.getText().toString().equals(""))
      {
            if (!C.equals("") &&
!editTextNum.getText().toString().equals(""))
                  if (C.equals("+"))
                        editTextRes.setText(new
BigDecimal(Double.parseDouble(editTextRes.getText().toString(
)) +
Double.parseDouble(editTextNum.getText().toString())).toPlain
String());
                  else if (C.equals("-"))
                        editTextRes.setText(new
BigDecimal(Double.parseDouble(editTextRes.getText().toString(
)) -
Double.parseDouble(editTextNum.getText().toString())).toPlain
String());
                  else if (C.equals("*"))
                        editTextRes.setText(new
BigDecimal(Double.parseDouble(editTextRes.getText().toString(
)) *
Double.parseDouble(editTextNum.getText().toString())).toPlain
String());
                  else if (C.equals("/") &&
!editTextNum.getText().toString().equals("0"))
                        editTextRes.setText(new
```

```
BigDecimal(Double.parseDouble(editTextRes.getText().toString(
)) /
Double.parseDouble(editTextNum.getText().toString())).toPlain
String());
                else if (C.equals("/") &&
editTextNum.getText().toString().equals("0"))
                {
                      ClearResNum();
                      Toast.makeText(this, "Err: Divide by
zero", Toast.LENGTH_SHORT).show();
                }
     }
     else if (!editTextNum.getText().toString().equals(""))

editTextRes.setText(editTextNum.getText().toString());
     editTextNum.setText("");
     }

     private void ReverseNum() {
     if (!editTextNum.getText().toString().equals("") &&
!editTextNum.getText().toString().equals("0"))
          editTextNum.setText(new
BigDecimal(Double.valueOf(editTextNum.getText().toString()) *
-1).toPlainString());
     }

     private void AddDotNum() {
     if (!editTextNum.getText().toString().equals("") &&
!editTextNum.getText().toString().contains("."))

editTextNum.setText(editTextNum.getText().toString()+".");
     }
}
```