



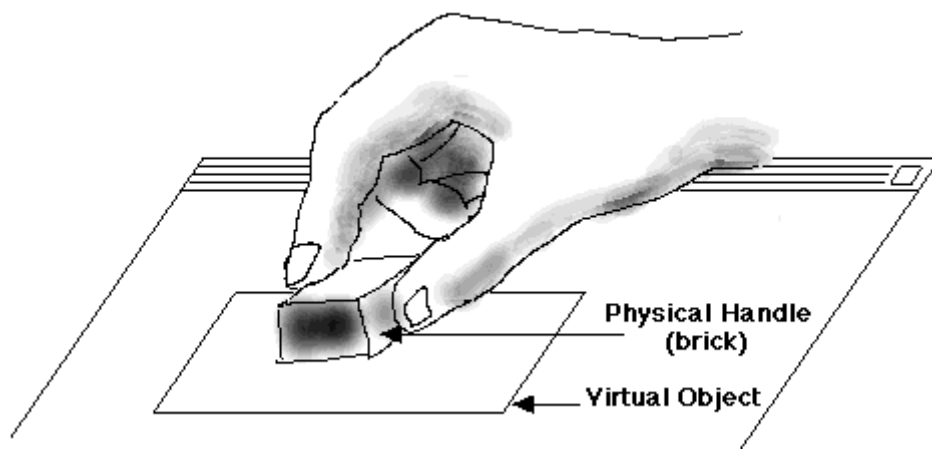
**ΤΕΙ ΠΑΤΡΩΝ**

**Σχολή Διοίκησης και Οικονομίας**

**Τμήμα Επιχειρηματικού Σχεδιασμού και Πληροφοριακών Συστημάτων**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Υλοποίηση Εφαρμογών σε Java με Χρήση Απτών Διεπαφών  
Χρήστη**



**ΦΟΙΤΗΤΗΣ: Βασίλειος Τσάπαλος**

**ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ: Δρ. Χρήστος Γκουμόπουλος**

**Πάτρα – 15/9/2009**

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΠΡΟΛΟΓΟΣ</b> .....	<b>3</b>
<b>ΕΙΣΑΓΩΓΗ</b> .....	<b>5</b>
<b>ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVA</b> .....	<b>7</b>
ΙΣΤΟΡΙΑ:.....	7
ΦΙΛΟΣΟΦΙΑ:.....	8
ΑΥΤΟΜΑΤΗ ΔΙΑΧΕΙΡΙΣΗ ΜΝΗΜΗΣ: .....	10
ΒΙΒΛΙΟΘΗΚΕΣ: .....	11
<b>ΑΠΤΕΣ ΔΙΕΠΑΦΕΣ ΧΡΗΣΤΗ</b> .....	<b>14</b>
ΠΑΡΑΔΕΙΓΜΑΤΑ:.....	14
<b>ΥΛΙΚΟ ΤΗΣ ΠΤΥΧΙΑΚΗΣ</b> .....	<b>15</b>
RFID-READER και RFID TAGS.....	15
INTERFACE KIT 8/8/8 .....	16
IR DISTANCE SENSOR: .....	17
MINI JOYSTICK SENSOR:.....	18
ROTATION SENSOR:.....	19
SLIDER SENSOR: .....	19
TOUCH SENSOR: .....	19
VOLTAGE DIVIDER:.....	20
<b>ΠΕΡΙΗΓΗΣΗ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ (NETBEANS)</b> .....	<b>21</b>
<b>ΕΦΑΡΜΟΓΗ 1 - ΧΡΗΣΗ RFID ΓΙΑ ΑΝΑΓΝΩΡΙΣΗ ΤΟΥ ΧΡΗΣΤΗ ΚΑΙ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ</b> .....	<b>24</b>
ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ.....	24
ΠΕΡΙΗΓΗΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.....	26
ΠΕΡΙΗΓΗΣΗ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΣΤΟ ΥΛΙΚΟ .....	27
<b>ΕΦΑΡΜΟΓΗ 2 - ΕΝΑΣ ΜΙΚΡΟΣ MEDIA PLAYER ΜΕ ΤΗ ΧΡΗΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ RHIDGETS</b> .....	<b>37</b>
ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ ΚΑΙ ΤΟ ΥΛΙΚΟ.....	37
ΠΕΡΙΗΓΗΣΗ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	41
<b>ΕΦΑΡΜΟΓΗ 3 - ΑΝΙΧΝΕΥΣΗ ΕΙΣΒΟΛΕΑ ΚΑΙ ΕΠΟΠΤΕΙΑ ΧΩΡΟΥ</b> .....	<b>51</b>
ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ ΚΑΙ ΤΟ ΥΛΙΚΟ.....	51
ΠΕΡΙΗΓΗΣΗ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	54
<b>ΕΦΑΡΜΟΓΗ 4 - ΑΠΟΣΤΟΛΗ LIVE ΒΙΝΤΕΟ ΜΕΣΩ INTERNET ΣΕ ΕΠΙΛΕΓΜΕΝΟΥΣ ΧΡΗΣΤΕΣ</b> .....	<b>72</b>
ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ ΚΑΙ ΤΟ ΥΛΙΚΟ.....	72
ΠΕΡΙΗΓΗΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.....	74
ΠΕΡΙΗΓΗΣΗ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	75
<b>ΕΦΑΡΜΟΓΗ 5 - ΑΣΦΑΛΕΙΑ ΛΕΙΤΟΥΡΓΙΑΣ Η/Υ ΑΠΟ ΕΝΑ ΑΤΟΜΟ ΧΩΡΙΣ ΔΙΚΑΙΩΜΑ ΠΡΟΣΒΑΣΗΣ ΚΑΙ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΕΙΣΒΟΛΕΑ</b> .....	<b>90</b>
ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ ΚΑΙ ΤΟ ΥΛΙΚΟ.....	90
ΠΕΡΙΗΓΗΣΗ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	94
<b>ΣΥΜΠΕΡΑΣΜΑΤΑ</b> .....	<b>99</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b> .....	<b>101</b>

## ΠΡΟΛΟΓΟΣ

Ο προγραμματισμός είναι μέχρι σήμερα η κυριότερη μέθοδος, με την οποία ο άνθρωπος καθοδηγεί τον υπολογιστή στην επίλυση προβλημάτων. Πρόκειται για μια από τις συνθετότερες νοητικές διαδικασίες, η οποία εξελίσσεται διαρκώς μέσα από τις διάφορες προσπάθειες συστηματοποίησής της, που γίνονται συνεχώς από τους ερευνητές.

Στα πλαίσια αυτής της πτυχιακής εργασίας παρατηρούμε πως ο προγραμματισμός μπορεί να εφαρμοσθεί και να μας παρέχει λύση σε διάφορα καθημερινά προβλήματα πέρα από την απλή χρήση του υπολογιστή. Πλέον μπορεί ο υπολογιστής να χρησιμοποιείται σε συνδυασμό με διάφορες συσκευές κατανεμημένες στο χώρο και ανάλογα με τα δεδομένα που του στέλνουν να μπορεί με τη χρήση κάποιας λογικής να επεξεργαστεί τα δεδομένα και να παράγει πληροφορία την οποία θα δώσει είτε στο χρήστη είτε στις συσκευές που παρέχουν έξοδο στο περιβάλλον. Για να γίνει πιο κατανοητό θα μπορούμε να έχουμε μια συσκευή η οποία για παράδειγμα μετρά τις συνθήκες του περιβάλλοντος (π.χ., θερμοκρασία, υγρασία, φωτεινότητα) του χώρου στον οποίο βρίσκεται (π.χ. σπίτι, γραφείο, δημόσιος χώρος), στη συνέχεια να στέλνει τις τιμές αυτές στον υπολογιστή και αυτός με τη σειρά του να υπολογίζει αν είναι ανεκτή η θερμοκρασία για τους ανθρώπους που κινούνται στο χώρο και στη συνέχεια αποφασίζει αν θα ανοίξει το κλιματιστικό στον ζεστό ή στον ψυχρό αέρα και σε ποια θερμοκρασία. Ακόμη μπορεί να εποπτεύει τον χώρο για ανθρώπινη παρουσία και σε περίπτωση που κανείς δεν βρίσκεται στο χώρο να κλείνει το κλιματιστικό για εξοικονόμηση ενέργειας.

Στις εφαρμογές που ακολουθούν κατανοούμε ακόμα περισσότερο τις δυνατότητες του προγραμματισμού και την αμεσότητα που μπορούν να έχουν με την ζωή των ανθρώπων είτε ως προς τη διασκέδαση είτε ως προς τη διευκόλυνση-αυτοματοποίηση κάποιων λειτουργιών ή ακόμα και τη χρήση τους για επαγγελματικούς λόγους.

Μέσα από την αναλυτική περιγραφή των εφαρμογών που αναπτύχθηκαν μπορούμε να κατανοήσουμε ακόμα περισσότερο τη χρήση και την έννοια του προγραμματισμού σε συνδυασμό τη χρήση ειδικού υλικού ως είσοδο και έξοδο.

Τέλος, θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Χρήστο Γκουμόπουλο για την ανάθεση της εργασίας, την επίβλεψη και την καθοδήγηση κατά τη διάρκεια της

εκπόνησης της εργασίας. Επίσης και όλα τα μέλη της ερευνητικής ομάδας DAISy (<http://daisy.cti.gr/>) του EAITY για τον εξοπλισμό και τη βοήθεια που μου παρείχαν.

## ΕΙΣΑΓΩΓΗ

Μια απτή διεπαφή χρήστη (Tangible User Interface) είναι μια διεπαφή χρήστη στην οποία κάποιος αλληλεπιδρά με τον υπολογιστή ή οποιοδήποτε άλλο μηχάνημα μέσω του φυσικού περιβάλλοντος. Ένα από τα πρώτα παραδείγματα απτών UI που αναπτύχθηκαν είναι ο μαρμάρινος αυτόματος τηλεφωνητής του Durrell Bishop (1992). Ένας γυάλινος βόλος αντιπροσωπεύει ένα ενιαίο μήνυμα που αφήνεται στον αυτόματο τηλεφωνητή. Η ρίψη ενός βόλου σε ένα πιάτο αναπαράγει το σχετικό μήνυμα ή ξανακαλεί τον επισκέπτη.

Στις εφαρμογές που αναπτύχθηκαν παρουσιάζονται διάφορες χρήσεις των απτών διεπαφών χρήστη. Οι εφαρμογές αυτές έχουν απεριόριστες δυνατότητες χρήσης και μπορεί να χρησιμοποιηθούν από εργαλεία διασκέδασης, μέχρι ως εργαλεία ασφάλειας ή εκπαίδευσης.

Στην αρχή του κειμένου δίνονται βασικά χαρακτηριστικά των εργαλείων λογισμικού που χρησιμοποιήθηκαν για την ανάπτυξη των εφαρμογών της παρούσας πτυχιακής. Τα εργαλεία περιλαμβάνουν τη γλώσσα προγραμματισμού Java, το περιβάλλον NetBeans για την ολοκληρωμένη ανάπτυξη εφαρμογών (Integrated Development Environment) καθώς και της Microsoft Access για την υλοποίηση των σχετικών Βάσεων Δεδομένων. Επίσης υπάρχει μια μικρή ανάλυση που δείχνει τον τρόπο λειτουργίας του υλικού που χρησιμοποιείται για κάθε εφαρμογή για να έχει ο χρήστης μια πιο καλή προσέγγιση στον κώδικα των εφαρμογών.

Η πρώτη εφαρμογή στοχεύει στο να συμπεραίνει ο υπολογιστής, όταν χρησιμοποιείται από πολλούς χρήστες, ποιος χρήστης συνδέεται και στη συνέχεια να εφαρμόζει τις ρυθμίσεις που έχει ζητήσει ο συγκεκριμένος χρήστης χωρίς να χρειαστεί να δώσει κάποια περαιτέρω εντολή. Για το σκοπό αυτό ο κάθε χρήστης είναι εφοδιασμένος με μια RFID κάρτα και υλοποιείται ο σχετικός αναγνώστης καρτών.

Η δεύτερη εφαρμογή στοχεύει πιο πολύ στη διασκέδαση και έτσι έχουμε δημιουργήσει έναν μικρό Media Player ο οποίος παρέχει τις βασικές του δυνατότητες (παύση, ρύθμιση φωνής, σταμάτημα, αλλαγή τραγουδιού) μέσω απτών διεπαφών χρήστη.

Η τρίτη εφαρμογή αναφέρεται στην ασφάλεια κάποιου χώρου και έτσι έχουμε δημιουργήσει μια εφαρμογή η οποία μόλις ανιχνεύσει κίνηση στο χώρο που ελέγχει αρχίζει να καταγράφει τον εισβολέα και στέλνει το βίντεο εκεί που έχει ρυθμίσει ο

χρήστης ή το αποθηκεύει στο σκληρό δίσκο με σκοπό να το δει ο χρήστης μόλις επιστρέψει στο χώρο του.

Η τέταρτη εφαρμογή στοχεύει στη διασκέδαση του χρήστη και την επικοινωνία του με τους φίλους του και έτσι έχουμε δημιουργήσει μια εφαρμογή στην οποία μέσω των απτών διεπαφών που βρίσκονται στο χώρο ο χρήστης δείχνει τη διαθεσιμότητά του στους φίλους του και αν ο ίδιος το επιθυμεί μέσω μιας κάμερας αρχίζει να στέλνει τον «εαυτό» του στους άλλους. Όπως στη δεύτερη εφαρμογή έτσι και εδώ οι βασικές λειτουργίες της κάμερας μπορούν να ελεγχθούν μέσω των απτών διεπαφών.

Η πέμπτη και τελευταία εφαρμογή αναφέρεται στην ασφάλεια. Στην προκειμένη περίπτωση όμως όχι ενός χώρου αλλά του ίδιου του υπολογιστή. Σε περίπτωση που κάποιο άλλο άτομο, που μπορεί να έχει πρόσβαση στον υπολογιστή, έχει υποκλέψει τον κωδικό του εξουσιοδοτημένου χρήστη πλέον θα έχει την δυνατότητα να εισέλθει στον υπολογιστή. Σε περίπτωση όμως που τρέχει και η συγκεκριμένη εφαρμογή τότε ο χρήστης θα πρέπει να περάσει και μια συγκεκριμένη κάρτα για να συνεχίσει, αλλιώς θα καταγραφεί από την κάμερα και στη συνέχεια ο υπολογιστής θα κλείσει.

## ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVA

### ΙΣΤΟΡΙΑ:

Ο James Gosling (γνωστός ως «Ο πατέρας της Java», τον προσέλαβε η Sun Microsystems για να υλοποιήσει την ιδέα του για την Java) ξεκίνησε τη μελέτη της γλώσσας Java τον Ιούνιο του 1991. Η γλώσσα αρχικά ονομαζόταν Oak. Το όνομα αυτό προήρθε από μια βελανιδιά που βρισκόταν έξω από το γραφείο του Gosling. Επίσης πήρε το όνομα Green και αργότερα κατέληξε να μετονομαστεί σε Java μέσα από μια λίστα τυχαίων λέξεων. Ο Gosling σκόπευε να πραγματοποιήσει μια εικονική μηχανή και μια γλώσσα που θα είχε μια παρόμοια σύνταξη με τις C, C++.

Η γλώσσα προγραμματισμού Java τελικά αναπτύχθηκε από τη Sun Microsystems και εκδόθηκε το 1995. Η γλώσσα προέρχεται από το συντακτικό της C και της C++ αλλά έχει ένα απλούστερο αντικειμενοστραφές μοντέλο και λιγότερες λειτουργίες χαμηλού επιπέδου. Οι εφαρμογές της Java μεταγλωττίζονται σε bytecode που μπορεί να «τρέχει» σε κάθε Java Virtual Machine (JVM) αδιαφορώντας για την αρχιτεκτονική του εκάστοτε υπολογιστή. Το Μάιο του 2007 σύμφωνα με τις προδιαγραφές από τη Java Community Process η Sun έκανε διαθέσιμες τις περισσότερες βιβλιοθήκες από τη Java ως δωρεάν λογισμικό κάτω από τη GNU General Public License. Άλλοι επίσης ανέπτυξαν διαφορετικές εφαρμογές σαν αυτές της Sun για τη Java, όπως ο GNU Compiler για Java.

Η Java υποσχέθηκε το: «Γράφεις μια φορά, τρέχει παντού», παρέχοντας μικρό κόστος κατά τη διάρκεια της εκτέλεσης στις διάσημες πλατφόρμες. Παρέχει πλήρη ασφάλεια η οποία όμως είναι πλήρως ρυθμίσιμη, επιτρέπει αποκλεισμό σε δικτυακή πρόσβαση καθώς και σε πρόσβαση στα αρχεία. Η πλειοψηφία των προγραμμάτων περιήγησης του ιστού ενσωμάτωσαν τη δυνατότητα να τρέχουν ασφαλείς μικροεφαρμογές Java (applets) μέσα στις ιστοσελίδες και η Java σύντομα έγινε διάσημη. Με την έλευση της Java 2 (παρουσιάστηκε αρχικά ως J2SE 1.2 το Δεκέμβριο του 1998), νέες εκδόσεις είχαν πολλαπλές διαμορφώσεις φτιαγμένες για διαφορετικούς τύπους πλατφορμών. Για παράδειγμα η J2EE στόχευε σε επιχειρηματικές εφαρμογές, ενώ η απογυμνωμένη έκδοση J2ME σε εφαρμογές κινητών. Η J2SE καθορίστηκε ως η πρότυπη έκδοση. Το 2006, για λόγους μάρκετινγκ η Sun μετονόμασε τις νέες J2 εκδόσεις Java EE, Java ME και Java SE αντίστοιχα.

Το 1997 η Sun Microsystems πλησίασε το ISO/TEC και αργότερα η Ecma International προσπάθησε να τυποποιήσει την Java αλλά σύντομα παραιτήθηκε από την όλη διαδικασία. Η Java ελέγχεται μέσω της Java Community Process. Κάποια στιγμή η Sun διένειμε τις περισσότερες από τις εγκαταστάσεις της Java χωρίς χρέωση παρόλο που είχε τα δικαιώματα ιδιοκτησίας του λογισμικού. Η Sun έχει έσοδα από την Java μέσω των πωλήσεων αδειών για ειδικά προϊόντα όπως το Java Enterprise System. Η Sun παρέχει το Standard Development Kit (SDK) που αναφέρεται στους προγραμματιστές και το Java Runtime Environment (JRE) το οποίο είναι υποσύνολο του SDK, του λείπουν ο μεταγλωττιστής, κάποια ωφέλιμα προγράμματα και άλλα αναγκαία αρχεία.

#### ΦΙΛΟΣΟΦΙΑ:

Κατά τη δημιουργία της γλώσσας υπήρχαν 5 πρωτεύοντες στόχοι:

1. Θα έπρεπε να ακολουθεί τη μεθοδολογία του αντικειμενοστραφούς προγραμματισμού.
2. Θα έπρεπε το ίδιο πρόγραμμα να εκτελείται σε μηχανήματα με διαφορετικά λειτουργικά συστήματα.
3. Θα έπρεπε να περιέχει ενσωματωμένη υποστήριξη για χρήση υπολογιστικών δικτύων.
4. Θα έπρεπε να σχεδιαστεί να εκτελεί κώδικα με ασφάλεια από απομακρυσμένες πηγές.
5. Θα έπρεπε να είναι εύκολη στη χρήση επιλέγοντας τα κομμάτια άλλων αντικειμενοστραφών γλωσσών προγραμματισμού που θεωρούνται καλύτερα.

Ένα χαρακτηριστικό της Java είναι ότι τα προγράμματα που είναι γραμμένα στη γλώσσα αυτή πρέπει να τρέχουν το ίδιο σε οποιαδήποτε πλατφόρμα υλικού-λειτουργικού συστήματος που την υποστηρίζει. Ο οποιοσδήποτε πρέπει να είναι σε θέση να γράψει ένα πρόγραμμα μια φορά, να το μεταγλωττίσει μια φορά, και να το τρέξει οπουδήποτε.

Αυτό επιτυγχάνεται μεταγλωττίζοντας τον κώδικα της Java όχι σε κώδικα μηχανής αλλά σε Java bytecode (μια ενδιάμεση μορφή κώδικα) ο οποίος όμως πρόκειται να μεταφραστεί σε εκτελέσιμη μορφή μέσω της virtual machine (VM) η οποία είναι γραμμένη ανάλογα με το υλικό του υπολογιστή. Οι τελικοί χρήστες κυρίως



χρησιμοποιούν ένα JRE που είναι εγκατεστημένο στο μηχάνημα τους ή στο πρόγραμμα περιήγησης του ιστού.

Τυποποιημένες βιβλιοθήκες παρέχουν ένα γενικευμένο τρόπο πρόσβασης σε συγκεκριμένα χαρακτηριστικά όπως γραφικά, νήματα και δίκτυο. Σε μερικές εκδόσεις της JVM ο bytecode μπορεί να μεταγλωττιστεί σε κώδικα μηχανής πριν ή κατά τη διάρκεια εκτέλεσης του προγράμματος καταλήγοντας σε γρηγορότερη εκτέλεση.

Το μεγαλύτερο πλεονέκτημα της χρήσης bytecode είναι η φορητότητα. Εντούτοις, η επιλογή της ερμηνείας συνεπάγεται ότι τα μεταγλωττισμένα σε bytecode προγράμματα σχεδόν πάντα τρέχουν πιο αργά από τα προγράμματα που συντάσσονται άμεσα σε γλώσσα μηχανής, έτσι η Java απέκτησε μια φήμη για κακή απόδοση. Αυτό το κενό έχει μικρύνει κατά πολύ στις τελευταίες εκδόσεις της JVM.

Μια τέτοια τεχνική, γνωστή όπως (Just In Time) JIT, μεταφράζει τον Java bytecode σε κώδικα μηχανής την πρώτη φορά που ο κώδικας εκτελείται, έπειτα το αποθηκεύει στη μνήμη cache. Πολυπλοκότερα VMs χρησιμοποιούν επίσης τη δυναμική επαναμεταγλώττιση, στην οποία η VM αναλύει τη συμπεριφορά του τρέχοντος προγράμματος και επαναμεταγλωττίζει επιλεκτικά και βελτιστοποιεί μέρη του προγράμματος. Η δυναμική επαναμεταγλώττιση μπορεί να επιτύχει βελτιστοποιήσεις ανώτερες απ' ό,τι η στατική μεταγλώττιση επειδή ο δυναμικός μεταγλωττιστής μπορεί να βασίσει τις βελτιστοποιήσεις στη γνώση για το περιβάλλον χρόνου εκτέλεσης και το σύνολο των φορτωμένων κλάσεων, και μπορεί να προσδιορίσει τα κρίσιμα σημεία - μέρη του προγράμματος, συχνά εσωτερικοί βρόγχοι, οι οποίοι παίρνουν τον περισσότερο χρόνο εκτέλεσης. Η μεταγλώττιση JIT και η δυναμική επαναμεταγλώττιση επιτρέπουν στα προγράμματα της Java να πλησιάσουν την ταχύτητα του κώδικα μηχανής χωρίς απώλεια φορητότητας.

Μια άλλη τεχνική, συνήθως γνωστή ως στατική μεταγλώττιση, ή ahead-of-time (AOT) μεταγλώττιση, είναι να μεταγλωττιστεί άμεσα στον εγγενή κώδικα όπως σε κάποιον παραδοσιακό μεταγλωττιστή. Οι στατικοί μεταγλωττιστές της Java μεταφράζουν τον πηγαίο κώδικα της Java ή το bytecode σε εγγενή αντικειμενοστραφές κώδικα. Αυτό επιτυγχάνει την καλή εκτέλεση έναντι της ερμηνείας, εις βάρος της φορητότητας το πρόγραμμα αυτών των μεταγλωττιστών μπορεί μόνο να εκτελεστεί σε μια μόνο αρχιτεκτονική.

Αξίζει να σημειωθεί πως η απόδοση της Java έχει βελτιωθεί ουσιαστικά από τις πρώτες εκδόσεις, και η απόδοση των μεταγλωττιστών JIT αναφορικά με τους εγγενείς μεταγλωττιστές σε μερικές δοκιμές έχουν αποδειχθεί να είναι αρκετά

παρόμοιοι. Η απόδοση των μεταγλωττιστών δεν δείχνει απαραίτητως την απόδοση του συνταγμένου κώδικα, δηλαδή αν ο κώδικας δεν έχει γραφτεί σωστά με στόχο την ταχύτητα δεν ευθύνεται ο μεταγλωττιστής της Java.

Επιπλέον, σε περιβάλλοντα μηχανών χρόνου εκτέλεσης όπως η Java υπάρχουν εργαλεία που συνδέονται με τη μηχανή χρόνου εκτέλεσης και κάθε φορά που μια εξαίρεση που μας ενδιαφέρει εμφανίζεται, καταγράφουν τις πληροφορίες διόρθωσης που υπήρξαν στη μνήμη τη στιγμή που η εξαίρεση εμφανίστηκε. Αυτά τα αυτοματοποιημένα εργαλεία χειρισμού εξαιρέσεων παρέχουν πληροφορίες για τις εξαιρέσεις στα προγράμματα της Java που τρέχουν σε περιβάλλοντα παραγωγής, δοκιμής ή ανάπτυξης.

#### ΑΥΤΟΜΑΤΗ ΔΙΑΧΕΙΡΙΣΗ ΜΝΗΜΗΣ:

Η Java χρησιμοποιεί έναν αυτόματο συλλέκτη απορριμμάτων για να διαχειριστεί τη μνήμη στον κύκλο της ζωής ενός αντικείμενου. Ο προγραμματιστής καθορίζει πότε τα αντικείμενα δημιουργούνται, και ο πυρήνας της Java κατά τον χρόνο εκτέλεσης είναι αρμόδιος για την ανάκτηση της μνήμης μόλις τα αντικείμενα δεν είναι πλέον σε χρήση. Όταν δεν υπάρχει καμία αναφορά σε ένα αντικείμενο, το αδέσποτο αντικείμενο επιλέγεται για να ελευθερωθεί αυτόματα από το συλλέκτη απορριμμάτων. Οι διαρροές μνήμης μπορούν ακόμα να εμφανιστούν εάν ο κώδικας ενός προγραμματιστή κρατά μια αναφορά σε ένα αντικείμενο που δεν χρειάζεται πια.

Μια από τις ιδέες πίσω από το μοντέλο αυτόματης διαχείρισης μνήμης της Java είναι ότι οι προγραμματιστές δεν χρειάζεται να την διαχειρίζονται χειροκίνητα. Σε μερικές γλώσσες όπως η C και η C++ η μνήμη για τη δημιουργία των αντικειμένων διατίθεται σιωπηρά στο σωρό(stack), ή επίσης διατίθεται και απελευθερώνεται από το σωρό. Καθένας από τους προηγούμενους τρόπους διαχείρισης μνήμης δεν απαλλάσσει τον προγραμματιστή από την ευθύνη διαχείρισής της. Εάν το πρόγραμμα δεν απελευθερώνει ένα αντικείμενο, μια διαρροή μνήμης εμφανίζεται. Εάν το πρόγραμμα προσπαθεί να έχει πρόσβαση ή να απελευθερώσει στη μνήμη ένα αντικείμενο που έχει ήδη απελευθερωθεί, το αποτέλεσμα είναι απροσδιόριστο και το πρόγραμμα μπορεί να γίνει ασταθές ή/και μπορεί να τερματιστεί απρόσμενα. Αυτό μπορεί να θεραπευθεί μερικώς με την χρήση των έξυπνων δεικτών, αλλά το πρόγραμμα γίνεται αρκετά πολύπλοκο.

Η συλλογή απορριμμάτων επιτρέπεται να συμβεί οποιαδήποτε στιγμή. Ιδανικά, θα εμφανιστεί όταν ένα πρόγραμμα είναι σε κατάσταση αδράνειας. Σίγουρα όμως θα

εμφανιστεί εάν υπάρχει ανεπαρκής ελεύθερη μνήμη στο σωρό για να δημιουργήσει ένα νέο αντικείμενο, το οποίο μπορεί να προκαλέσει μια μικρή καθυστέρηση στο πρόγραμμα.

Η Java δεν υποστηρίζει την αριθμητική δεικτών τύπου C/C ++, όπου οι διευθύνσεις των αντικειμένων και οι ακέραιοι αριθμοί (συνήθως τύπου long) μπορούν να χρησιμοποιηθούν εναλλακτικά. Αυτό επιτρέπει στο συλλέκτη απορριμμάτων να επανεντοπίσει τα παραπεμφθέντα αντικείμενα (αντικείμενα που τους έχει δώσει ο προγραμματιστής την τιμή null και μόλις τρέξει η JVM τον garbage collector θα διαγραφούν), και να εξασφαλίσει την τυπική ασφάλεια και την προστασία.

**BIBΛΙΟΘΗΚΕΣ:** Η Java συνοδεύεται από τις βιβλιοθήκες-πακέτα που παρέχονται από τη Sun Microsystems ή από τρίτους. Όπως είναι φυσιολογικό οι βιβλιοθήκες που παρέχονται από τρίτους είναι πάρα πολλές και ποικίλουν ανάλογα με το τι ήθελε να καταφέρει ο δημιουργός τους. Κάποια χαρακτηριστικά παραδείγματα βιβλιοθηκών της Java είναι η ύπαρξη βιβλιοθηκών που χρησιμοποιούνται για κλήσεις SOAP μέσα από κινητό, η δυνατότητα δημιουργίας προγράμματος Messenger χρησιμοποιώντας το e-mail που έχουμε φτιάξει στην Microsoft, η δυνατότητα να χειριστούμε και να προγραμματίσουμε το hardware που παρέχεται από διάφορες εταιρίες (Phidgets, Sun, CrossBow) ακόμα και η δυνατότητα να υποκλέψουμε τα e-mails που έχει κάποιος αποθηκευμένα στο Outlook.

Βλέποντας τις βιβλιοθήκες που παρέχονται με την εγκατάσταση της γλώσσας παρατηρούμε πως εξυπηρετούν τους βασικούς σκοπούς που χρειάζεται μια γλώσσα με στόχο τη δημιουργία απλών εφαρμογών. Κάποιες από τις βασικές βιβλιοθήκες οι οποίες χρησιμοποιήθηκαν και στα παρακάτω προγράμματα είναι οι εξής:

- «[java.applet](#)»: Παρέχει τις απαραίτητες κλάσεις για τη δημιουργία ενός applet.
- «[java.awt](#)»: Παρέχει όλες τις κλάσεις για τη δημιουργία διεπαφών χρήστη αλλά και για τη δημιουργία απλών γραφικών και εικόνων.
- «[java.awt.color](#)»: Παρέχει κλάσεις για τη δημιουργία χρωμάτων.
- «[java.awt.event](#)»: Παρέχει τις απαραίτητες κλάσεις για τη διαχείριση διαφόρων τύπων γεγονότων που προκαλούνται από τα AWT αντικείμενα.
- «[java.io](#)»: Παρέχει κλάσεις για τη χρήση εισόδου-εξόδου δεδομένων από το σύστημα και από αρχεία αλλά και δυνατότητα σειριοποίησης αντικειμένων.
- «[java.lang](#)»: Εμπεριέχει τις θεμελιώδεις κλάσεις της Java.

- «[java.lang.reflect](#)»: Παρέχει κλάσεις οι οποίες στοχεύουν στο να παίρνουν πληροφορίες από άλλες κλάσεις.
- «[java.math](#)»: Παρέχει κλάσεις που χρησιμοποιούνται για μαθηματικές πράξεις μεγάλης ακρίβειας.
- «[java.net](#)»: Παρέχει κλάσεις που χρησιμοποιούνται για την υλοποίηση δικτυακών συνδέσεων σε μια εφαρμογή.
- «[java.nio](#)»: Παρέχει βελτιωμένες κλάσεις για τη χρήση εισόδου-εξόδου δεδομένων από το σύστημα και από αρχεία
- «[java.rmi](#)»: Παρέχει κλάσεις για τη δημιουργία και την κλήση απομακρυσμένων (σε άλλον υπολογιστή) μεθόδων.
- «[java.security](#)»: Παρέχει κλάσεις για ασφάλεια κατά την επικοινωνία μέσα από το δίκτυο.
- «[java.sql](#)»: Παρέχει κλάσεις για την αποθήκευση και διάβασμα σε βάση δεδομένων.
- «[java.util](#)»: Παρέχει κλάσεις χρήσιμες για βελτίωση του προγραμματισμού όπως κλάσεις τύπου πίνακα αλλά με περισσότερες δυνατότητες, κλάσεις με δυνατότητα να παίρνουν την τρέχουσα ημέρα, ώρα ακόμα και κλάσεις για τη δημιουργία ακολουθίας ψευδοτυχαίων αριθμών.

Εκτός των παραπάνω βιβλιοθηκών κατά καιρούς έχουν εκδοθεί από τη Sun βιβλιοθήκες οι οποίες δεν ανήκουν στον κορμό της Java αλλά η ονομασία πακέτου αρχίζει ως javax(τα επιπρόσθετα πακέτα που δημιουργούνται από τη Sun αρχίζουν ως javax).Τέτοιου τύπου βιβλιοθήκες είναι η βιβλιοθήκη που χρησιμοποιείται για την χρήση κάμερας και αναπαραγωγή βίντεο από τη Java.

- «[javax.media](#)»: Είναι το πακέτο που χρησιμοποιείται για να υλοποιήσουμε τις βασικές λειτουργίες που προσφέρονται από τα υπόλοιπα πακέτα.
- «[javax.media.control](#)»: Είναι το πακέτο που χρησιμοποιείται για να ελέγξουμε τον τρόπο αναπαραγωγής αρχείων πολυμέσων.
- «[javax.media.datasink](#)»: Είναι το πακέτο που χρησιμοποιείται για την αποθήκευση αρχείων πολυμέσων που έρχονται από το δίκτυο ή αναπαράγονται τοπικά στον υπολογιστή.

- [«javax.media.format»](#): Είναι το πακέτο που χρησιμοποιείται για να καθορίσουμε τη μορφή αρχείου πολυμέσων που θα αποθηκευτεί ή αναπαραχθεί ή σταλεί στο δίκτυο.
- «javax.media.protocol»: Είναι το πακέτο που χρησιμοποιείται για να ελέγξουμε τον τρόπο αποστολής αρχείων πολυμέσων στο δίκτυο.
- «javax.media.renderor»: Είναι το πακέτο που χρησιμοποιείται για να δημιουργήσουμε μια διεπαφή χρήστη κατά τη διάρκεια αναπαραγωγής βίντεο.
- «javax.media.rtp»: Είναι το πακέτο που χρησιμοποιείται και το οποίο συνδυάζεται με το «javax.media.protocol» πακέτο για αποστολή στο δίκτυο με σκοπό την ταυτόχρονη αναπαραγωγή από τον υπολογιστή που λαμβάνει.
- «javax.media.rtp.event»: Είναι το πακέτο που χρησιμοποιείται για τη δημιουργία γεγονότων για να δείξει ότι όλα πάνε καλά ή το αντίθετο καθώς και να περιγράψουν το πρόβλημα.

Όπως προαναφέραμε υπάρχουν βιβλιοθήκες οι οποίες δίνονται από διάφορες εταιρείες με σκοπό να χρησιμοποιηθεί το υλικό τους για την ανάπτυξη διάφορων εφαρμογών. Οι εφαρμογές που ακολουθούν χρησιμοποιούν υλικό της εταιρείας Phidgets, συνεπώς χρησιμοποιούν και τα πακέτα που παρέχει η εταιρεία αυτή. Τα πακέτα αυτά είναι:

- «com.phidgets»: Είναι το πακέτο που χρησιμοποιείται για προετοιμάσουμε το υλικό για χρήση.
- «com.phidgets.event»: Είναι το πακέτο που χρησιμοποιείται για να μπορέσουμε να χρησιμοποιήσουμε τις εισόδους που δίνει το υλικό μέσω γεγονότων όπως συνηθίζεται στη γλώσσα Java.

## ΑΠΤΕΣ ΔΙΕΠΑΦΕΣ ΧΡΗΣΤΗ

Μια απτή διεπαφή χρήστη (Tangible User Interface) είναι μια διεπαφή χρήστη στην οποία κάποιος αλληλεπιδρά με τον υπολογιστή ή οποιοδήποτε άλλο μηχάνημα μέσω του φυσικού περιβάλλοντος. Το αρχικό όνομα ήταν Graspable User Interface, το οποίο δεν χρησιμοποιείται πλέον. Ένας από τους πρωτοπόρους στις απτές διεπαφές χρήστη είναι ο Hiroshi Ishii, ένας καθηγητής στο MIT Media Laboratory που διευθύνει την ερευνητική ομάδα Media Group. Το ιδιαίτερο όραμά του για απτά UIs, αποκαλούμενο Tangible Bits, επιχειρεί να δώσει φυσική μορφή στις ψηφιακές πληροφορίες, που καθιστούν τα bits εύκολα στο χειρισμό και άμεσα αντιληπτά.

### ΠΑΡΑΔΕΙΓΜΑΤΑ:

Κάποια παραδείγματα μπορούν να βοηθήσουν στην εξήγηση για το τι είναι οι απτές διεπαφές χρήστη αλλά και πως χρησιμοποιούνται.

Ένα παράδειγμα ενός απτού UI είναι ο μαρμάρινος αυτόματος τηλεφωνητής του Durrell Bishop (1992). Ένας γυάλινος βόλος αντιπροσωπεύει ένα ενιαίο μήνυμα που αφήνεται στον αυτόματο τηλεφωνητή. Η ρίψη ενός βόλου σε ένα πιάτο αναπαράγει το σχετικό μήνυμα ή ξανακαλεί τον επισκέπτη.

Ένα άλλο παράδειγμα είναι το σύστημα Torobo. Οι δεσμοί στο Torobo είναι όπως οι δεσμοί LEGO που μπορούν να σπάσουν απότομα όλοι μαζί, αλλά μπορούν επίσης να κινηθούν μόνοι τους χρησιμοποιώντας τα μηχανικά μέρη τους. Ο οποιοσδήποτε μπορεί να σπρώξει, να τραβήξει, και να στρίψει αυτούς τους δεσμούς, και οι δεσμοί μπορούν να απομνημονεύσουν αυτές τις μετακινήσεις και να τις επαναλάβουν.

Μια άλλη εφαρμογή επιτρέπει στο χρήστη να σχεδιάσει μια εικόνα σε μια επιτραπέζια επιφάνεια με ένα πραγματικό απτό μολύβι. Χρησιμοποιώντας τις κινήσεις του χεριού, ο χρήστης μπορεί να κλωνοποιήσει την εικόνα και να την τεντώσει στους άξονες X και Y ακριβώς όπως κάποιος σε ένα πρόγραμμα χρωμάτων. Ακολουθεί μια εκτενέστερη ανάλυση πάνω στις συσκευές που χρησιμοποιήθηκαν για την εργασία.

## ΥΛΙΚΟ ΤΗΣ ΠΤΥΧΙΑΚΗΣ

### RFID-READER και RFID TAGS



Εικόνα 1

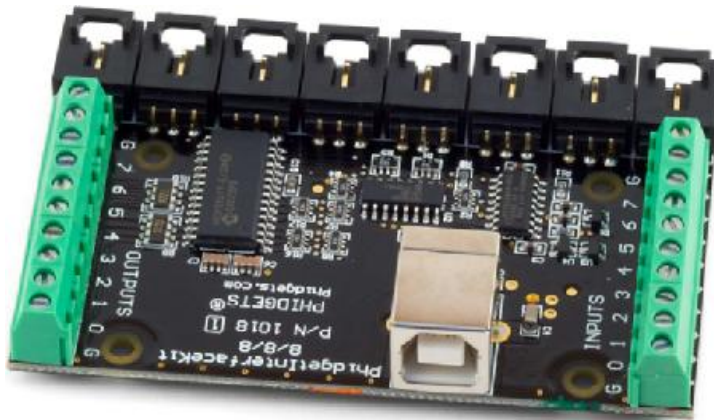
Τα RFID συστήματα χρησιμοποιούν συμβολοσειρές που είναι αποθηκευμένες εσωτερικά σε ετικέτες RFID (ή αναμεταδότες) για να αναγνωρίζουν μεμονωμένους ανθρώπους ή αντικείμενα όταν ανιχνεύονται από έναν αναγνώστη RFID. Αυτοί οι τύποι συστημάτων βρίσκονται μέσα σε πολλές εφαρμογές όπως το διαβατήριό , προσδιορισμός ζώων, συστήματα ελέγχου καταλόγων, και ασφαλή συστήματα ελέγχου πρόσβασης.

Ένας αναγνώστης RFID για να επικοινωνεί με μια ετικέτα RFID, πρέπει μοιραστούν ένα κοινό πρωτόκολλο. Αυτό το πρωτόκολλο ενεργεί όπως ένα σύνολο κανόνων για τον τρόπο που τα δεδομένα μεταφέρονται ασύρματα μεταξύ του αναγνώστη και της ετικέτας. Οι ετικέτες PhidgetRFID (καθώς επίσης και RFID που πωλούνται απ' την Phidgets) χρησιμοποιεί το πρωτόκολλο EM4102. Οποιοσδήποτε άλλες ετικέτες που χρησιμοποιούν επίσης το πρωτόκολλο EM4102 μπορούν να χρησιμοποιηθούν με το PhidgetRFID.

Οι ετικέτες RFID έρχονται σε δύο κύριες μορφές: παθητικές και ενεργητικές. Οι ενεργητικές ετικέτες έχουν την δική τους παροχή ηλεκτρικού ρεύματος και τη χρησιμοποιούν για να τροφοδοτήσουν μια κεραία για να επικοινωνούν και να μεταδίδουν δεδομένα. Οι παθητικές ετικέτες αντλούν τη δύναμη που απαιτείται για να λειτουργήσουν άμεσα από την RF έξοδο του αναγνώστη RFID, και καμία άλλη παροχή ηλεκτρικού ρεύματος δεν είναι απαραίτητη. Αυτό καθιστά τις παθητικές ετικέτες φτηνότερες στην παραγωγή και ευκολότερες στην χρήση.

Επειδή οι παθητικές ετικέτες απαιτούν έναν ισχυρό τομέα RF για να λειτουργήσουν, η αποτελεσματικότητά τους περιορίζεται σε μια περιοχή γεωγραφικά πολύ κοντά στον αναγνώστη RFID. Στην περίπτωση του PhidgetRFID, ετικέτες που παρουσιάζονται στον αναγνώστη RFID μέσα σε περίπου 5 εκατοστά μπορούν να διαβαστούν. Η απόσταση στην οποία η ετικέτα RFID μπορεί να χρησιμοποιηθεί επηρεάζεται από διάφορα πράγματα, όπως μορφή των ετικετών και μέγεθος τους, τα διάφορα υλικά που χρησιμοποιούνται στην περιοχή κοντά στον αναγνώστη, καθώς και ο προσανατολισμός του αναγνώστη σε συνδυασμό με τον προσανατολισμό της ετικέτας. Όσο μικρότερη είναι μια ετικέτα, τόσο πιο κοντά πρέπει να είναι στον αναγνώστη για να λειτουργήσει.

### INTERFACE KIT 8/8/8



Εικόνα 2

Το Interface Kit που χρησιμοποιείται στις παρακάτω εφαρμογές έχει 8 αναλογικές εισόδους, 8 ψηφιακές εισόδους και 8 ψηφιακές εξόδους.

Οι αναλογικές εισοδοί χρησιμοποιούνται για να συνδεθούν διάφοροι τύποι αισθητήρων. Κάθε αναλογική είσοδος παρέχει ρεύμα (κόκκινο καλώδιο) της τάξης των +5VDC, γείωση (μαύρο καλώδιο), και επιστρέφει την τάση-τιμή του αισθητήρα (άσπρο καλώδιο). Το PhidgetInterfaceKit μετρά συνεχώς την τάση(τιμή) που επιστρέφεται και ενημερώνει την εφαρμογή. Στις αναλογικές εισόδους του συνδέουμε διάφορους αισθητήρες όπως αισθητήρας θερμοκρασίας, υγρασίας, πίεσης κτλ. Επίσης από ηλεκτρολογικής πλευράς αξίζει να σημειωθεί ότι μέγιστο συνολικό ρεύμα που καταναλώνεται από όλες τις αναλογικές εισόδους πρέπει να περιοριστεί στα 400mA. Η αναλογική μέτρηση αντιπροσωπεύεται στο λογισμικό μέσω της μεταβλητής SensorValue ως τιμή μεταξύ 0 και 1000. Μια μονάδα στην τιμή των αισθητήρων αντιπροσωπεύει μια τάση περίπου 5 millivolts.



Στις ψηφιακές του εισόδους συνδέουμε οποιαδήποτε συσκευή μπορεί να μας δώσει τιμές Boolean. Για να συνδέσουμε μια συσκευή στην ψηφιακή είσοδο αρκεί να συνδέσουμε τη μια άκρη της σε μια ψηφιακή είσοδο και την άλλη στη γείωση. Η κατάσταση των ψηφιακών εισόδων αναφέρεται στο υπολογιστή περιοδικά. Κατά τη διάρκεια αυτής της περιόδου, εάν μια ψηφιακή είσοδος ήταν σε κατάσταση true για διάστημα μεγαλύτερο από 4.0ms, η ψηφιακή είσοδος εγγυάται ότι θα επιστρέψει true στο λογισμικό. Αυτό καθιστά την ψηφιακή είσοδο πιο ευαίσθητη στο να αναφέρει την κατάσταση true. Οποιαδήποτε true γεγονότα της ψηφιακής εισόδου παρατηρούνται για λιγότερο από 1.5ms δεν αναφέρονται ποτέ.

Οι ψηφιακές του έξοδοι μπορούν να ελέγξουν οποιαδήποτε συσκευή δέχεται Boolean τιμές και χρειάζεται 5 volt για να λειτουργήσει. Οι συσκευές συνδέονται στις ψηφιακές εξόδους όπως και στις εισόδους. Οι συνδέσεις για τη γείωση στο InterfaceKit μοιράζονται από κοινού τη γείωση του USB του υπολογιστή με τον οποίο συνδέονται. Εάν μια αντίστροφη τάση ή μια επικίνδυνα υψηλή τάση εφαρμοστεί στην ψηφιακή είσοδο ή έξοδο μπορεί να προκληθεί ζημιά στο υλικό που χρησιμοποιείται ή ακόμα και στον υπολογιστή.

Όπως προαναφέραμε, στο υλικό που χρησιμοποιείται έχουμε τη δυνατότητα να συνδέσουμε διάφορους αισθητήρες. Παρακάτω ακολουθεί μια ανάλυση στους εξής αισθητήρες: IR distance sensor, mini joystick, rotation sensor, slider, touch sensor και ένα voltage divider σε συνδυασμό με έναν αισθητήρα μεταβλητής αντίστασης.

### IR DISTANCE SENSOR:



Εικόνα 3

Ο αισθητήρας της Sharp GP2D12 μετρά τις αποστάσεις από 10 cm έως 80 cm. Οι τιμές των αισθητήρων κυμαίνονται από 80 έως 500. Οι τιμές αυτές είναι περίπου

αντιστρόφως ανάλογες με την απόσταση. Ο τύπος που μετράει τις αποστάσεις που επιστρέφονται από τον αισθητήρα είναι ο εξής:

$$\text{Απόσταση(cm)}=4800/(\text{τιμή\_αισθητήρα}-20).$$

Ο τύπος αυτός είναι εσφαλμένος για τιμές μικρότερες του 80 καθώς και μεγαλύτερες του 500. Πέραν της χρήσης του για να μετράει αποστάσεις μπορεί να χρησιμοποιηθεί για να παρατηρεί αν κάτι πέρασε από μπροστά του(έτσι χρησιμοποιείται σε όλα τα προγράμματα που ακολουθούν).

Η λειτουργία του IR Distance Adapter Board (χρησιμοποιείται για να συνδεθεί ο αισθητήρας με το Interface Kit) είναι να ρυθμίζει το απαιτούμενο ρεύμα που χρειάζεται ο αισθητήρας GP2D12. Αν και η καθορισμένη κατανάλωση ρεύματος του GP2D12 είναι μόλις 30mA, ο αισθητήρας χρειάζεται έως και 300mA κατά τις μικρές χρονικές περιόδους που μετράει-επιστρέφει τις τιμές του. Εάν δύο ή περισσότεροι αισθητήρες συνδέονται άμεσα με τις αναλογικές εισαγωγές ενός PhidgetInterfaceKit (το οποίο μπορεί να παρέχει ένα μέγιστο ρεύμα έως 500mA) το InterfaceKit, μπορεί ενδεχομένως να προκληθεί υπερφόρτωση. Μέχρι οκτώ αισθητήρες GP2D12 της Sharp μπορούν ακίνδυνα να συνδεθούν ταυτόχρονα με το PhidgetInterfaceKit 8/8/8 χρησιμοποιώντας IR Distance Adapter Boards, οι οποίοι αποτρέπουν την πιθανότητα υπερφόρτωσης.

#### MINI JOYSTICK SENSOR:



Εικόνα 4

Αυτό το μικροσκοπικό joystick έχει δύο άξονες, τον οριζόντιο και τον κάθετο, και ένα κουμπί. Έχει ένα ποτενσιόμετρο με μια κεντρική τιμή περίπου 500. Όταν κινείται το joystick από την κεντρική του θέση της η τιμή θα αυξηθεί ή θα μειωθεί ανάλογα με την κατεύθυνση.

Πατώντας προς τα κάτω το joystick δημιουργείται μια στιγμιαία επαφή. Η τιμή που επιστρέφεται από την επαφή αυτή μπορεί να εισαχθεί στο PhidgetInterfaceKit συνδέοντας το joystick σε μια από τις ψηφιακές εισόδους του.

### ROTATION SENSOR:



Εικόνα 5

Για τον αισθητήρα αυτόν δεν υπάρχουν κάποια ιδιαίτερα χαρακτηριστικά. Είναι ένας αισθητήρας που παίρνει τιμές από 0 έως 1000. Έχει δυνατότητα περιστροφής με τη φορά του ρολογιού αλλά και αντίθετη έτσι ώστε να επανέλθει στην αρχική του θέση. Η μέγιστη αντίσταση του ποντεσιόμετρου είναι 10 Kohm.

### SLIDER SENSOR:



Εικόνα 6

Ο αισθητήρας αυτός είναι παρόμοιος με τον προηγούμενο. Οι τιμές του κυμαίνονται από 0 έως 1000 και για την επιστροφή των τιμών αυτών χρησιμοποιείται μια παρόμοια αντίσταση με τον προηγούμενο της τάξης των 10 Kohm.

### TOUCH SENSOR:



Εικόνα 7

Ο αισθητήρας αφής αλλάζει την τιμή του από 1000 σε 0 όταν αγγίζεται. Πιο συγκεκριμένα, ο αισθητήρας αυτός είναι πραγματικά ένας χωρητικός αισθητήρας αλλαγής. Όταν η χωρητικότητα αλλάζει ο αισθητήρας επιστρέφει μηδέν. Θα λειτουργήσει ανάμεσα 1/8 ίντσας ενός γυαλιού, ενός πλαστικού, ή ενός χαρτιού.

VOLTAGE DIVIDER:



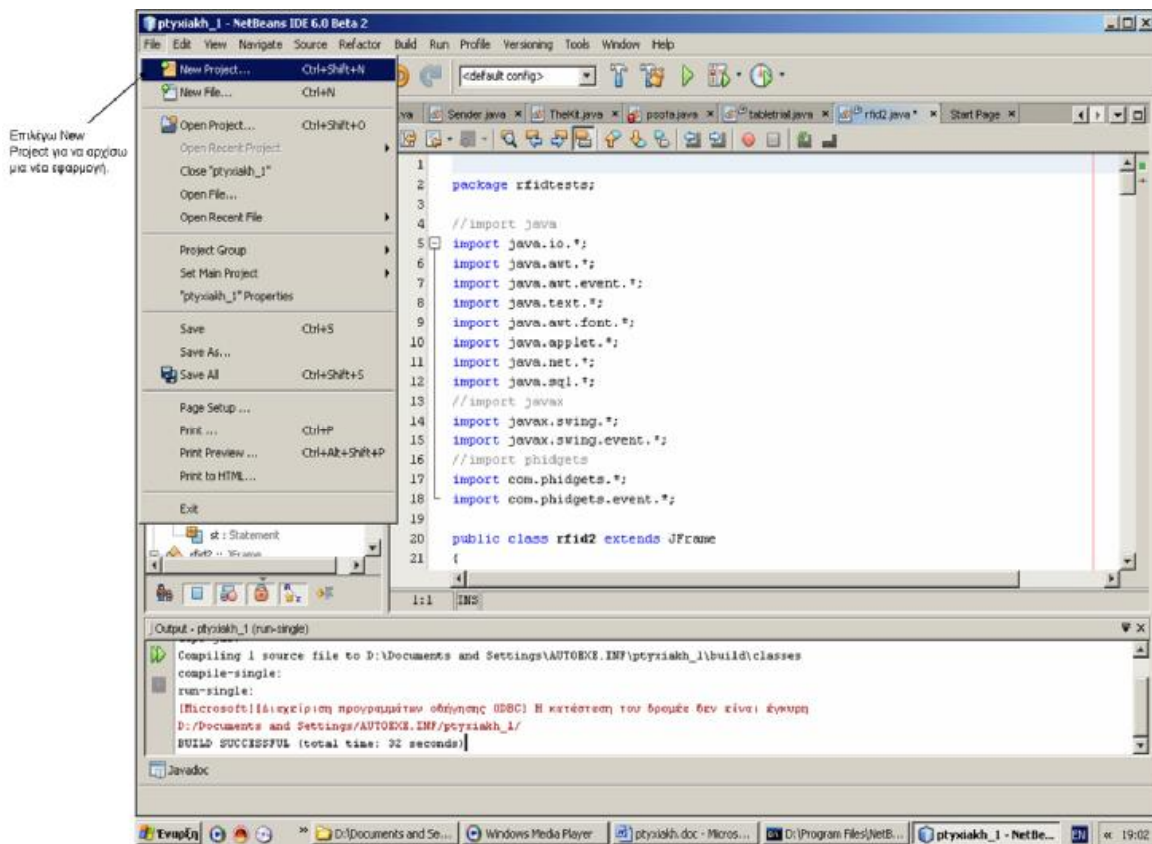
Εικόνα 8

Η αναλογική είσοδος μπορεί να μετρήσει μια τάση μεταξύ 0V και 5V. Η αναλογική αυτή μέτρηση αντιπροσωπεύεται στο λογισμικό ως τιμή μεταξύ 0 και 1000, έτσι η τιμή 1 μονάδας στο λογισμικό αντιπροσωπεύει μια τάση περίπου 5 millivolts. Κατά τη χρήση των μεταβλητών συσκευών αντίστασης ο διαιρέτης τάσης (voltage divider) χρησιμοποιείται συνήθως για να μετρήσει τη μεταβαλλόμενη αντίσταση. Οι δύο αντιστάτες μέσα στον διαιρέτη διαμορφώνει μια τάση, βασισμένη στην αναλογία μεταξύ τους. Αυτή η τάση μετρείται έπειτα από την αναλογική είσοδο.

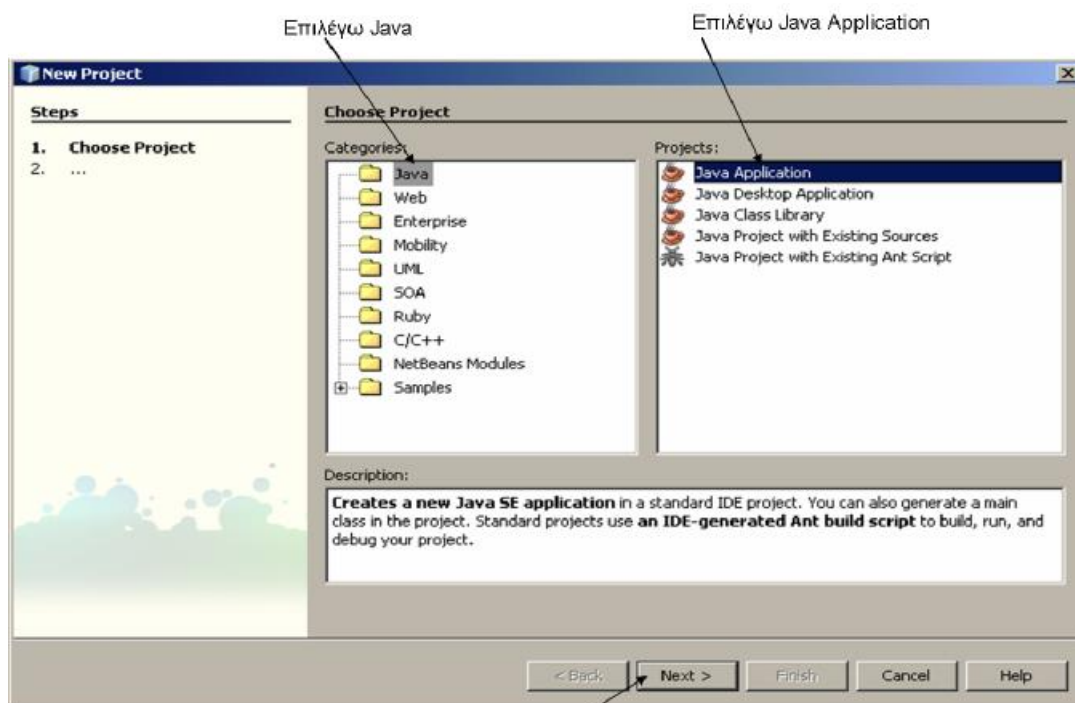
## ΠΕΡΙΗΓΗΣΗ ΣΤΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ (NETBEANS)

Ξεκινώντας αξίζει να αναφέρουμε ότι όλες οι εφαρμογές δημιουργήθηκαν με το περιβάλλον NetBeans IDE 6.0. Οπότε κάποια βήματα που θα αναφερθούν σε όλες τις εφαρμογές έχουν γίνει σύμφωνα με το πρόγραμμα αυτό(π.χ. η εισαγωγή βιβλιοθηκών στην classpath του συστήματος).

Για να δημιουργήσουμε μια εφαρμογή επιλέγουμε File → New Project ,όπως Εικόνα 9 και στη συνέχεια εμφανίζεται ένα νέο παράθυρο όπως Εικόνα 10. Επιλέγοντας το κουμπί Next εμφανίζεται το τελευταίο παράθυρο για να καθορίσω τις τελικές ρυθμίσεις για το project μου(Εικόνα 11). Έτσι αρχίζουμε οποιοδήποτε καινούριο project. Όμως κάποιες εφαρμογές χρειάζονται εξωτερικές βιβλιοθήκες. Πάνω αριστερά στο NetBeans υπάρχει ένα δέντρο που περιέχει τις εφαρμογές. Μόλις ανοίξουμε αυτή που θέλουμε εμφανίζονται τρεις υποφακέλοι. Θα εισάγουμε τη βιβλιοθήκη στο φάκελο Libraries, όπως φαίνεται στις Εικόνες 12 και 13.



Εικόνα 9

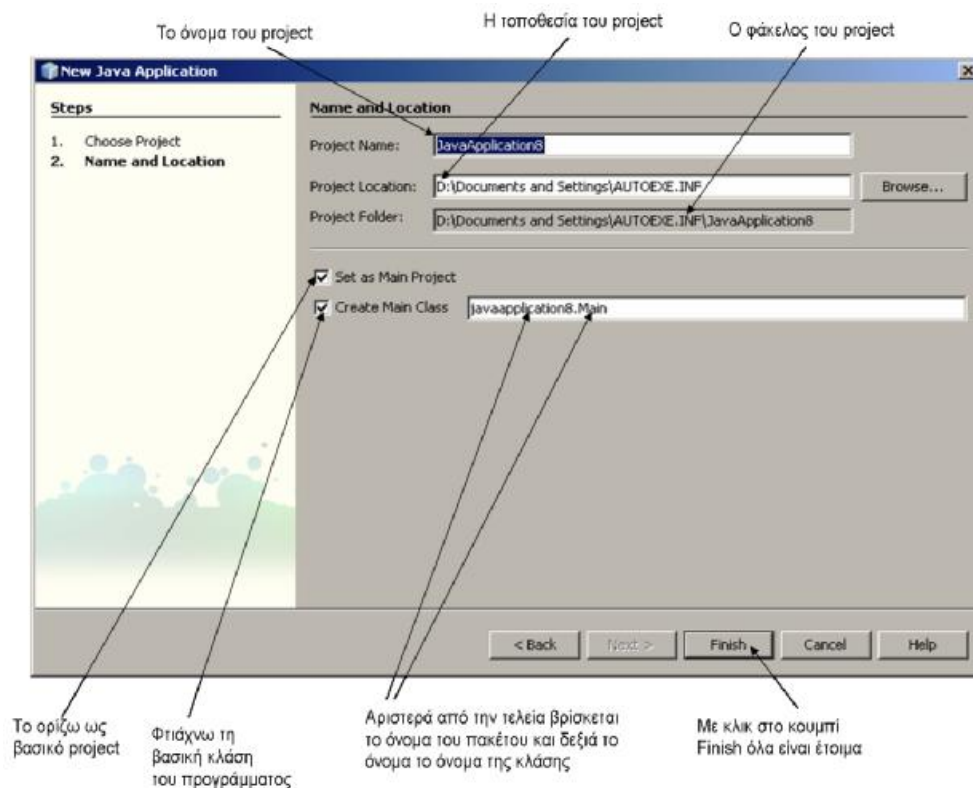


Επιλέγω Java

Επιλέγω Java Application

Τέλος κάνω κλικ στο κουμπί Next

Εικόνα 10



Το όνομα του project

Η τοποθεσία του project

Ο φάκελος του project

Το ορίζω ως βασικό project

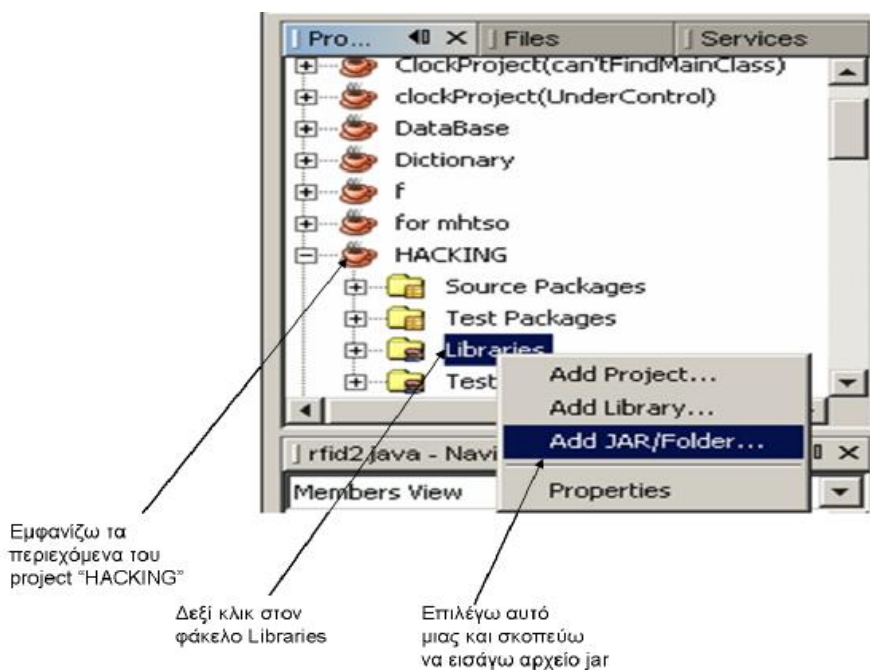
Φτιάχνω τη βασική κλάση του προγράμματος

Αριστερά από την τελεία βρίσκεται το όνομα του πακέτου και δεξιά το όνομα της κλάσης

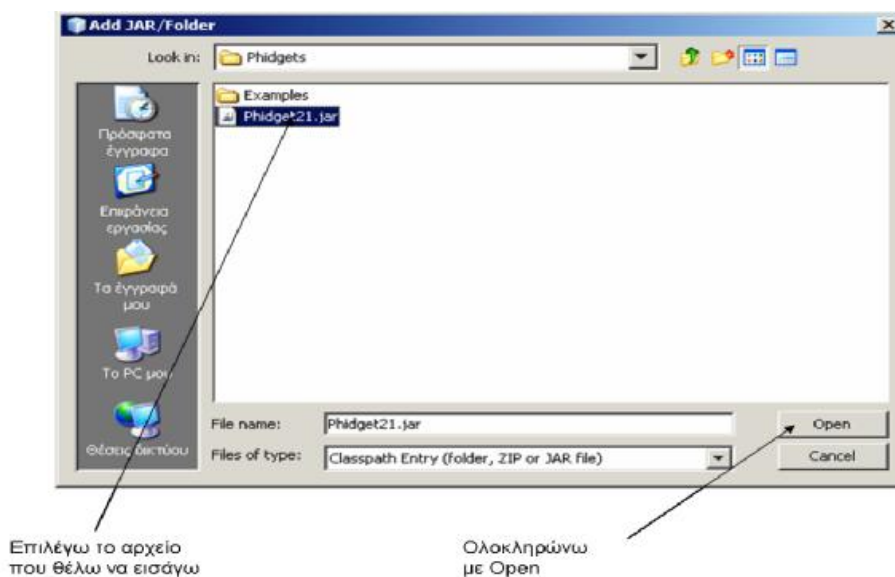
Με κλικ στο κουμπί Finish όλα είναι έτοιμα

Εικόνα 11





Εικόνα 12



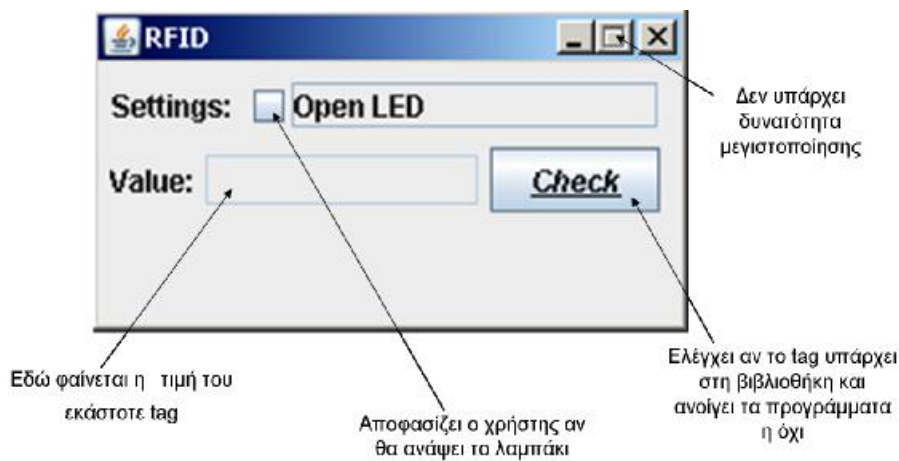
Εικόνα 13

## ΕΦΑΡΜΟΓΗ 1 - ΧΡΗΣΗ RFID ΓΙΑ ΑΝΑΓΝΩΡΙΣΗ ΤΟΥ ΧΡΗΣΤΗ ΚΑΙ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ

### ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ

Στόχος της εφαρμογής αυτής είναι μέσω της χρήσης ενός RFID reader και πολλαπλών tag (ένα για κάθε χρήστη) να ανοίγουν τα αγαπημένα τους προγράμματα χωρίς να χρειαστεί να τα εκτελούν χειροκίνητα κάθε φορά που ανοίγει ο υπολογιστής.

Αυτό το επιτυγχάνουμε μέσω μιας εφαρμογής που επικοινωνεί με το χρήστη (Εικόνα 14) και η οποία συνδέεται με μια βάση δεδομένων που εμπεριέχει διάφορα στοιχεία και ρυθμίσεις για κάθε χρήστη.



Εικόνα 14

Παρατηρούμε πως το παράθυρο της εφαρμογής μας δεν έχει δυνατότητα μεγιστοποίησης γιατί θα έχουμε ένα μεγάλο παράθυρο με ελάχιστο περιεχόμενο, κάτι το οποίο είναι άσχημο αισθητικά.

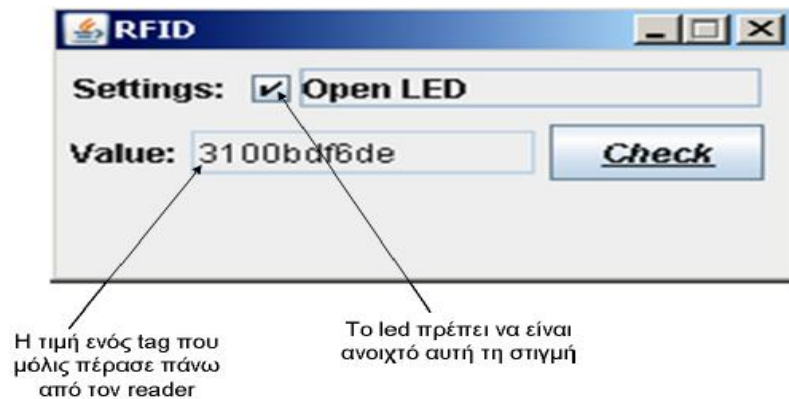
Επιπλέον, μια ακόμα ρύθμιση είναι το άνοιγμα ενός μικρού led που βρίσκεται πάνω στον RFID reader. Μπορεί να χρησιμοποιηθεί από τον χρήστη είτε για αισθητικούς λόγους είτε για να επιβεβαιωθεί πως ο reader δουλεύει σωστά και είναι έτοιμος για χρήση.

Αξίζει να σημειωθεί ότι στο πεδίο κειμένου που ακολουθεί το JLabel που αναγράφει «Value:» δυνατότητα εγγραφής έχει μόνο το rfid reader και όχι κάποιο άλλο υλικό όπως το πληκτρολόγιο για παράδειγμα.

Μόλις κάνουμε κλικ στο κουμπί Check το πρόγραμμα κοιτάει στη βάση δεδομένων που του έχουμε ορίσει να δει αν υπάρχει κάποιος χρήστης που να έχει ως κλειδί το tag (την τιμή που υπάρχει στο πεδίο κειμένου δίπλα στο Value, Εικόνα

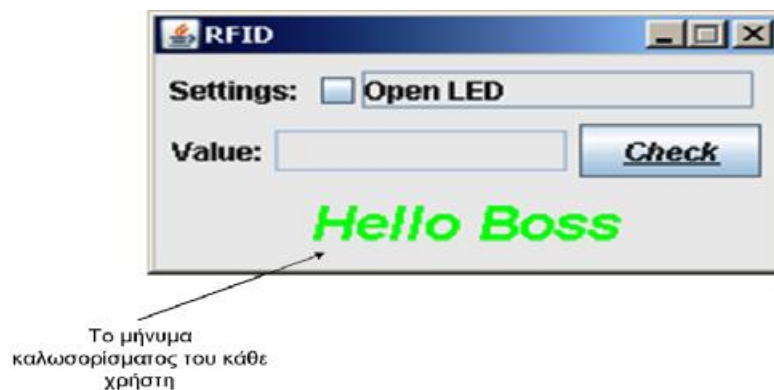


15 ).Στη βάση δεδομένων έχουμε ορίσει ως πρωτεύον κλειδί την τιμή των tags μιας και καθένα έχει ξεχωριστή τιμή από τα υπόλοιπα.



Εικόνα 15

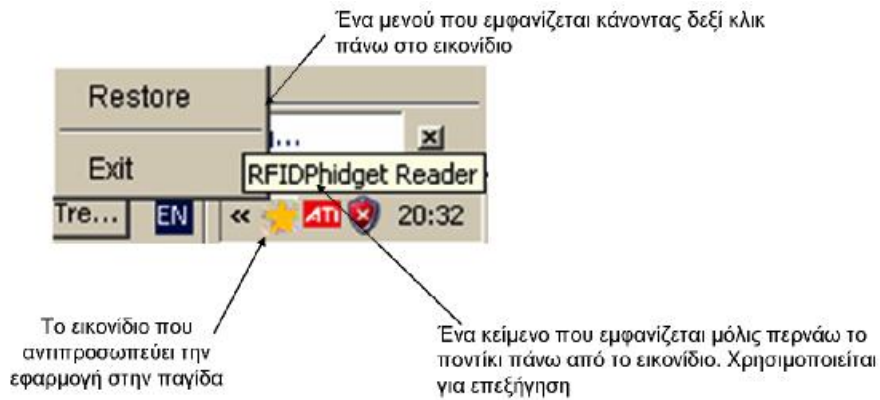
Μόλις ολοκληρωθούν οι διαδικασίες που εκτελούνται με το πάτημα του κουμπιού εμφανίζεται ένα μήνυμα που καλωσορίζει το χρήστη(Εικόνα 16) αλλιώς εμφανίζει ένα μήνυμα λάθους(Εικόνα 17). Το μήνυμα λάθους εμφανίζεται αν δεν βρεθεί μια εγγραφή στη βάση με το ζητούμενο tag. Όταν όμως το tag είναι σωστό η εφαρμογή αρχίζει τα προγράμματα του χρήστη και πηγαίνει στα εικονίδια δίπλα στο ρολόι(Εικόνα 18).



Εικόνα 16



Εικόνα 17



Εικόνα 18

Αξίζει να σημειωθεί πως το κείμενο επεξήγησης στις Ηνωμένες Πολιτείες Αμερικής είναι υποχρεωτικό να χρησιμοποιείται. Για περισσότερες λεπτομέριες μπορούμε να ανατρέξουμε στο δικτυακό τόπο της Sun Microsystems (<http://www.sun.com/access/background/laws.html>).

## **ΠΕΡΙΗΓΗΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ**

Όπως προαναφέραμε η εφαρμογή ψάχνει την τιμή-κλειδί που ταιριάζει στο tag που πέρασε από τον reader μέσα σε μια βάση δεδομένων. Η εφαρμογή αυτή συνδέεται με μια βάση δεδομένων φτιαγμένη με το πρόγραμμα Access της σουίτας γραφείου Microsoft Office. Το γεγονός ότι τα δεδομένα βρίσκονται κάπου εκτός της εφαρμογής, αποθηκευμένα σε γνωστή μορφή αρχείου, συνεπάγεται ανασφάλεια αλλά και ελαστικότητα του προγράμματος.

Από την πλευρά της ανασφάλειας, το πρόβλημα μπορεί να λυθεί με κλείδωμα του αρχείου της βάσης. Έτσι θα έχει τη δυνατότητα μόνο ένας εξουσιοδοτημένος χρήστης να αλλάξει τα δεδομένα.

Από την άλλη πλευρά όμως μπορούν να προστεθούν ή να αναιρεθούν από τη βάση όσοι χρήστες θέλει ο διαχειριστής της χωρίς τη μεταβολή του κώδικα της εφαρμογής.

Με μια ματιά στη βάση δεδομένων που χρησιμοποιούμε, παρατηρούμε ότι περιέχει έναν πίνακα ο οποίος αποτελείται από πέντε στήλες με περιεχόμενα ορισμένα από το χρήστη από τις οποίες μόνο η πρώτη στήλη έχει υποχρεωτική ανάθεση τιμής στο κελί(Εικόνα 19). Σε περίπτωση κενού πεδίου η εφαρμογή χρησιμοποιεί ως προεπιλεγμένη τιμή την κενή τιμή.

Το πρωτεύον κλειδί της βάσης είναι η τιμή του tag  
 Το όνομα του χρήστη. Όπως επιθυμεί ο ίδιος να λέγεται  
 Το μήνυμα καλωσορισμού που επιθυμεί ο κάθε χρήστης να έχει  
 Η διαδρομή του αρχείου ήχου που θα αναπαράγεται με την είσοδο του κάθε χρήστη  
 Η διαδρομή του προγράμματος που θα εκτελείται με την είσοδο του κάθε χρήστη

tagValue	userName	writableMessage	playableMessage	openableProgram
01065e8486	user1	Hi user1	D:\rfidProperties\audio\applause.wav	
0106708856	user7		D:\rfidProperties\audio\gunshot3.wav	D:\Program Files\Adobe\Acrob
0f00ad62bb	user4			
3000b4e006	user8	Πάλι εσύ;		D:\Documents and Settings\A\
31006d3a4d	user9	Καλώς ήρθες	D:\rfidProperties\audio\gunshot3.wav	
3100ac2f27	user6	Τι τρέχει γιατρέ;		D:\WINDOWS\notepad.exe
3100b3bb80	user5	Βρε, βρε...	D:\rfidProperties\audio\zap.wav	
3100bd6fde	Bill	Hello Boss	D:\rfidProperties\audio\tada.wav	D:\Program Files\Windows Me
3100be20d2	user3	Καλώς τον μου	D:\rfidProperties\audio\explosion.wav	D:\Program Files\Internet Expli
3100c3efe1	user2	Welcome user2		
*				

Εικόνα 19

Πηγαίνοντας όλο και σε πιο βαθιά ανάλυση της εφαρμογής έφτασε η στιγμή να δούμε αυτό που δεν βλέπουν οι χρήστες στις εφαρμογές που χρησιμοποιούν, τον πηγαίο κώδικα. Ας κάνουμε μια περιήγηση στον κώδικα της εφαρμογής για να δούμε πως φαίνεται από την πλευρά του προγραμματιστή μια εφαρμογή.

### ΠΕΡΙΗΓΗΣΗ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΣΤΟ ΥΛΙΚΟ

Η εφαρμογή αποτελείται από τρεις κλάσεις. Η πρώτη η οποία φροντίζει να ξεκινήσει το πρόγραμμα είναι η RFID\_Main. Παρακάτω ακολουθεί η Database και η ManageRFIDReader, οι οποίες αναλαμβάνουν την επικοινωνία με τη βάση δεδομένων και την ετοιμασία του RFID reader για είσοδο δεδομένων αντίστοιχα. Πριν από τις κλάσεις πρέπει να εισάγουμε τα πακέτα τα οποία μας βοηθούν να πάρουμε κάποια έτοιμα πράγματα ώστε να υλοποιήσουμε το πρόγραμμα. Εισάγουμε τα βασικά πακέτα της java(java packages),τα επεκτεταμένα πακέτα(javax packages) καθώς και τα πακέτα της εταιρείας που έφτιαξε το υλικό που δουλεύουμε (rhidgets packages).

Αρχίζοντας την ανάλυση του κώδικα ξεκινάμε με την κλάση RFID\_Main. Αυτή είναι η βασική κλάση για το λόγο ότι αυτή εμπεριέχει τη μέθοδο main(...)(γραμμές 232-235). Με το που αρχίζουμε να αναλύουμε την κλάση βλέπουμε πως είναι δηλωμένα κάποια αντικείμενα ως private(γραμμές 13-24) όπως ορίζει και η φιλοσοφία της γλώσσας. Ένα αντικείμενο(γραμμή 15) είναι ορισμένο και ως static για

να το χρησιμοποιούν άλλες κλάσεις χωρίς να χρειάζεται να δημιουργήσουν ένα αντικείμενο της ίδιας της κλάσης. Δυο από τα αντικείμενα δεν έχουν οριστεί ως public μιας και οι κλάσεις τους δεν είναι public.

Ακολουθεί ο constructor της κλάσης(γραμμή 26). Προσθέτω έναν ακροατή συμβάντων παραθύρου(γραμμές 33-40) για να παρακολουθεί τότε επιθυμεί ο χρήστης να κλείσει το παράθυρο ώστε να σταματήσει τον RFID reader να δέχεται είσοδο, να κλείσει το led,να αποσυνδεθεί από τη βάση δεδομένων και τέλος να κλείσει την εφαρμογή. Μέσα στον constructor τοποθετώ τα διάφορα αντικείμενα στο παράθυρο και τους προσθέτω ακροατές συμβάντων. Έτσι έχω έναν ακροατή συμβάντων για το JCheckBox ο οποίος παρακολουθεί αν είναι επιλεγμένο το κουτάκι και ανοιγοκλείνει το led του reader(γραμμές 47-65). Υπάρχει ένας ακόμα ακροατής για το κουμπί "Check"(γραμμές 75-212) ο οποίος έχει αναλάβει να πάρει την τιμή του tag,να «ρωτήσει» τη βάση δεδομένων αν υπάρχει η συγκεκριμένη τιμή και να πάρει τα δεδομένα(γραμμή 86), να εμφανίσει τα κατάλληλα μηνύματα(ηχητικά και μη) για τον εκάστοτε χρήστη(γραμμές 95-96,107-113), να ανοίξει το επιλεγμένο πρόγραμμα(γραμμή 128) και να στείλει την εφαρμογή στην παγίδα δίπλα από το ρολόι(γραμμές 135-184).

Αξίζει να σημειωθεί πως τα μηνύματα που εμφανίζονται καθώς και το κείμενο του κουμπιού(γραμμή 70) έχει δημιουργηθεί μέσω html κώδικα, χωρίς τη χρήση αντικειμένων φόντου που προσφέρει η java για λόγους ταχύτητας στη λειτουργία του προγράμματος και ευκολίας στον προγραμματισμό(δε θα άξιζε τον κόπο να δημιουργώ επιπλέον αντικείμενα για μια τόσο απλή χρήση).

Η μέθοδος repaintWindow(), η οποία με τη σειρά της καλεί τη μέθοδο validate() του αντικειμένου JFrame, χρησιμοποιείται για την ανανέωση των γραφικών του παραθύρου έτσι ώστε να εμφανίζονται να νέα μηνύματα που πρέπει να εμφανιστούν ή/και να σβηστούν κάποια προηγούμενα(γραμμές 221-226).

Με τη χρήση της μεθόδου setTextToField (γραμμές 228-230) μπορούμε να έχουμε πρόσβαση στο αντικείμενο που επιθυμούμε χωρίς να έρθουμε σε αντιπαράθεση με τη φιλοσοφία της γλώσσας.

Παρατηρώντας την υλοποίηση του παραθύρου βλέπουμε πως πριν το παράθυρο γίνει ορατό για τον χρήστη καλώ τις μεθόδους-δημιουργούς των δυο βοηθητικών κλάσεων(γραμμές 216-218) που θα χρειαστώ. Αυτό το κάνω για να μη μπορεί ο χρήστης να εισάγει δεδομένα χωρίς να έχουν υλοποιηθεί όλες οι κλάσεις προς αποφυγή εξαιρέσεων. Ακολουθεί ο κώδικας:

```

1
2 package TUI_App_1;
3
4 import java.io.*;
5 import java.awt.*;
6 import java.awt.event.*;
7 import java.applet.*;
8 import java.net.*;
9 import javax.swing.*;
10
11 public class RFID_Main extends JFrame
12 {
13     private JPanel pane,welcome;
14     private JCheckBox box;
15     private static JTextField field;
16     private JButton check;
17     private JLabel message;
18     //set the path for the sounds
19     private String ErrorNoise;
20     private SystemTray tray;
21     private TrayIcon icon;
22     private PopupMenu ppm;
23     private ManagerRFIDReader rf;
24     private Database db;
25
26     public RFID_Main()
27     {
28         super("RFID");
29         setResizable(false);
30         addWindowListener(new WindowAdapter()
31         {
32             @Override
33             public void windowClosing(WindowEvent we)
34             {
35                 rf.ledPlay(false);
36                 rf.close();
37                 db.closeAll();
38                 System.exit(0);
39             }
40         });
41         setBounds(250,250,250,130);
42         pane=new JPanel();
43         pane.add(new JLabel("Settings:"));
44         box=new JCheckBox("Open LED");
45         pane.add(box);
46         //set the listener
47         box.addItemListener(new ItemListener()
48         {
49             public void itemStateChanged(ItemEvent ie)
50             {
51                 int state=ie.getStateChange();
52                 if(state==ItemEvent.SELECTED)
53                 {
54                     //open the led
55                     rf.ledPlay(true);
56                 }
57                 else if(state==ItemEvent.DESELECTED)
58                 {
59                     //close the led
60                     rf.ledPlay(false);
61                 }

```

```

62         else
63             System.err.println("SOMETHING IS WRONG");
64     }
65 });
66 pane.add(new JLabel("Value:"));
67 field=new JTextField(10);
68 field.setEditable(false);
69 pane.add(field);
70 String htmlButton="<html><center><p><u><i>Check</i></u></p>";
71 check=new JButton(htmlButton);
72 pane.add(check);
73 message=new JLabel("");
74 pane.add(message);
75 check.addActionListener(new ActionListener()
76 {
77     public void actionPerformed(ActionEvent ae)
78     {
79         if(ae.getSource() instanceof JButton)
80         {
81             //set the welcome or error panel
82             //edw tha kanei erwthsh sth bash
83
84             try
85             {
86                 String[] data=db.getData(field.getText());
87                 field.setText("");
88                 if(!data[0].equals(null))
89                 {
90                     pane.remove(message);
91                     try
92                     {
93                         if(!data[1].equals(null))
94                         {
95                             String
htmlr="<html><center><p><i><b><font size=\"6\" " +
96 "color=\"#00FF00\">"+data[1]+"</font></b></i></p>";
97                             message.setText(htmlr);
98                         }
99                             pane.add(message);
100                     }
101                     catch(NullPointerException npe){}
102                     //paizei hxo
103                     try
104                     {
105                         if(!data[2].equals(null))
106                         {
107                             String
rightNoise=data[2].replace("\\", "/");
108                             rightNoise="file:///"+rightNoise;
109                             try
110                             {
111                                 AudioClip
ac=Applet.newAudioClip
112                                     (new URL(rightNoise));
113                                 ac.play();
114                             }
115                             catch(IOException ioe){}
116                         }
117                         field.setText("");
118                         repaintWindow();

```

```

119         }
120         catch(NullPointerException npe){}
121         //as ανοικσει to programma
122         try
123         {
124             if(!data[3].equals(null))
125             {
126                 try
127                 {
128
Runtime.getRuntime().exec(data[3]);
129                 }
130                 catch(Exception
e){System.err.println(e.getMessage());}
131             }
132         }
133         catch(NullPointerException npe){}
134         //send to tray
135         if(SystemTray.isSupported())
136         {
137             try
138             {
139                 tray=SystemTray.getSystemTray();
140                 String str=new
File("R").getCanonicalPath();
141                 str=str.substring(0,str.length()-
1);
142                 ImageIcon imic=new
ImageIcon(str+"\\star.GIF");
143                 Image im=imic.getImage();
144                 ppm=new PopupMenu();
145                 MenuItem mil=new
MenuItem("Restore");
146                 mil.addActionListener(new
ActionListener()
147                 {
148                     public void
actionPerformed(ActionEvent evt)
149                     {
150                         setVisible(true);
151                         tray.remove(icon);
152                     }
153                 });
154                 ppm.add(mil);
155                 ppm.addSeparator();
156                 MenuItem mi2=new MenuItem("Exit");
157                 mi2.addActionListener(new
ActionListener()
158                 {
159                     public void
actionPerformed(ActionEvent evt)
160                     {
161                         System.exit(0);
162                     }
163                 });
164                 ppm.add(mi2);
165                 icon=new TrayIcon(im,"RFIDPhidget
Reader",ppm);
166                 setVisible(false);
167                 tray.add(icon);

```

```

168             icon.addMouseListener(new
MouseListener()
169             {
170                 @Override
171                 public void
mouseClicked(MouseEvent me)
172                 {
173                     int b=me.getButton();
174                     if(b==1)
175                     {
176                         setVisible(true);
177                         tray.remove(icon);
178                     }
179                     else if(b==3)
180                     {
181                         icon.setPopupMenu(ppm);
182                     }
183                 }
184             });
185         }
186         catch(Exception
awte){System.err.println(awte.getMessage());}
187     }
188 }
189 }
190 catch(NullPointerException npe)
191 {
192     pane.remove(message);
193     String html="<html><center><p><b><i><font
size="6" " +
194 "color="FF0000">ERROR</font></i></b></p>";
195     message.setText(html);
196     pane.add(message);
197     try
198     {
199         ErrorNoise=new
File("r").getCanonicalPath();
200     ErrorNoise=ErrorNoise.substring(0,ErrorNoise.length()-1);
201     ErrorNoise=ErrorNoise.replace("\\","/");
202     System.err.println(ErrorNoise);
203     AudioClip ac=Applet.newAudioClip
204         (new
URL("file:///"+ErrorNoise+"errorLogIn.wav"));
205     ac.play();
206     }
207     catch(IOException
ioe){System.err.println(ioe.getMessage());}
208     field.setText("");
209     repaintWindow();
210 }
211 }
212 }
213 });
214 setContentPane(pane);
215 //construct the rfid class
216 rf=new ManagerRFIDReader();
217 //edw tha anoigei th syndesh me th bash
218 db=new Database();
219 setVisible(true);

```



```

220     }
221     public void repaintWindow()
222     {
223         //setContentPane(pane);
224         //setVisible(true);
225         validate();
226     }
227
228     public static void setTextToField(String text){
229         field.setText(text);
230     }
231
232     public static void main(String[] args)
233     {
234         new RFID_Main();
235     }
236 }

```

Η κλάση Database αναλαμβάνει να επικοινωνήσει με τη βάση δεδομένων. Όλα τα προγράμματα που χρησιμοποιούν μια προέλευση δεδομένων ODBC θα χρησιμοποιούν το πρόγραμμα οδήγησης γέφυρας JDBC-ODBC sun.jdbc.odbc.JdbcOdbcDriver. Έτσι η μέθοδος-δομητής μέσα σε μπλοκ try-catch φορτώνει την προηγούμενη κλάση και στη συνέχεια καθορίζει μια σύνδεση προς την προέλευση δεδομένων(γραμμές 21-29).

Ακολουθεί μια μέθοδος η οποία έχει ως είσοδο την τιμή του tag και σύμφωνα με αυτή εκτελεί SQL ερώτηση στη βάση δεδομένων για να πάρει τις ρυθμίσεις του χρήστη. Επιστρέφει τα στοιχεία που χρειάζεται σε μορφή πίνακα(γραμμές 30-56). Υπάρχει μια ακόμα μέθοδος που αναλαμβάνει να κλείσει οποιαδήποτε επικοινωνία με τη βάση δεδομένων(γραμμές 57-66). Ακολουθεί ο κώδικας:

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package TUI_App_1;
7
8  /**
9   *
10  * @author AUTOEXE.INF
11  */
12
13  import java.sql.*;
14
15  public class Database
16  {
17      private Connection con;
18      private Statement st;
19      private ResultSet rs;
20
21      public Database()
22      {

```

```

23     try
24     {
25         Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
26         con=DriverManager.getConnection("jdbc:odbc:ptyxiakh","","");
27     }
28     catch(Exception e){System.err.println(e.getMessage());}
29 }
30 public String[] getData(String tagValue)//01065e8486
31 {
32     String[] arr=new String[] {null,null,null,null};
33     try
34     {
35         String query="SELECT GlobalSettings.* "+
36                     "FROM GlobalSettings "+
37                     "WHERE
38 ((GlobalSettings.tagValue)='"+tagValue+"');";
39
40         st=con.createStatement();
41         String[] sta=new String[8];
42         rs=st.executeQuery(query);
43         //System.err.println("here");
44         rs.next();
45         for(int i=0;i<5;i++)
46         {
47             sta[i]=rs.getString(i+1);
48             //System.err.println(sta[i]);
49         }
50         arr[0]=sta[0];
51         arr[1]=sta[2];
52         arr[2]=sta[3];
53         arr[3]=sta[4];
54     }
55     catch(Exception e){System.err.println(e.getMessage());}
56     return arr;
57 }
58 public void closeAll()
59 {
60     try
61     {
62         rs.close();
63         st.close();
64         con.close();
65     }
66     catch(Exception e){}
67 }

```

Εν τέλει έχουμε την κλάση ManageRFIDReader η οποία όπως προαναφέραμε κάνει την εφαρμογή μας να επικοινωνεί με τον RFID reader και να δέχεται είσοδο. Μέσα στη μέθοδο-δημιουργό(γραμμές 22-40) ετοιμάζω το υλικό να δέχεται είσοδο και μόλις δεχτεί εκτελεί τον κώδικα που βρίσκεται στην interface(γραμμές 30-37). Συνεχίζοντας την μελέτη της κλάσης παρατηρούμε πως υπάρχουν δυο μέθοδοι. Η πρώτη αναλαμβάνει να ανοιγοκλείνει το led του reader(γραμμές 41-48) και η δεύτερη σταματάει τη συσκευή να δέχεται είσοδο(γραμμές 49-56). Ακολουθεί ο κώδικας:

1 /\*

```

2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package TUI_App_1;
7
8  /**
9   *
10  * @author AUTOEXE.INF
11  */
12
13  import com.phidgets.*;
14  import com.phidgets.event.*;
15
16  public class ManagerRFIDReader
17  {
18      private RFIDPhidget ph;
19      private String value;
20      private boolean open=true;
21
22      public ManagerRFIDReader()
23      {
24          try
25          {
26              ph=new RFIDPhidget();
27              ph.openAny();
28              ph.waitForAttachment();
29              ph.setAntennaOn(true);
30              ph.addTagGainListener(new TagGainListener()
31              {
32                  public void tagGained(TagGainEvent tge)
33                  {
34                      value=tge.getValue();
35                      RFID_Main.setTextToField(value);
36                  }
37              });
38          }
39          catch(PhidgetException pe){System.err.println(pe.getMessage());}
40      }
41      public void ledPlay(boolean bool)
42      {
43          try
44          {
45              ph.setLEDOOn(bool);
46          }
47          catch(PhidgetException pe){System.err.println(pe.getMessage());}
48      }
49      public void close()
50      {
51          try
52          {
53              ph.close();
54          }
55          catch(PhidgetException pe){System.err.println(pe.getMessage());}
56      }
57  }

```

Αξίζει να σημειωθεί πως οι μέθοδοι που κλείνουν οτιδήποτε ανοίγει δεν είναι απαραίτητο να υλοποιηθούν μιας και εκτελούνται μόνο ακριβώς πριν τερματιστεί το

πρόγραμμα και αυτό γιατί η java με την έξοδο της εφαρμογής μας αναλαμβάνει να κλείσει βίαια για λόγους ασφαλείας οτιδήποτε δεν έκλεισε ο προγραμματιστής. Ακολουθούν δυο εικόνες για το πως είναι το υλικό που χρησιμοποιούμε (Εικόνα 20) καθώς και τα είδη των tag υπάρχουν(Εικόνα 21).



**Εικόνα 20**



Διάφορα είδη tag

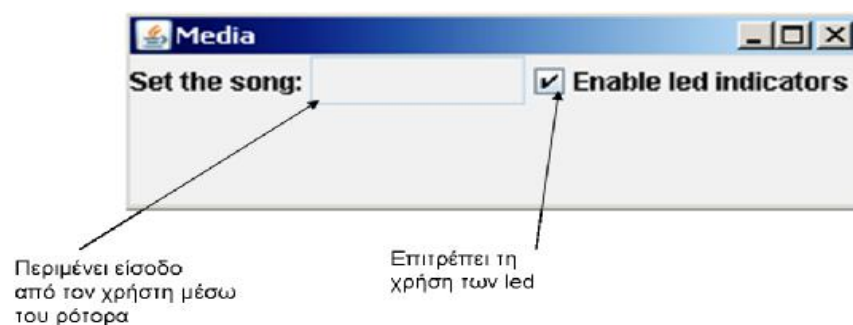
**Εικόνα 21**

## ΕΦΑΡΜΟΓΗ 2 - ΕΝΑΣ ΜΙΚΡΟΣ MEDIA PLAYER ΜΕ ΤΗ ΧΡΗΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ PHIDGETS

### ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ ΚΑΙ ΤΟ ΥΛΙΚΟ

Στόχος της εφαρμογής αυτής είναι να φτιάξουμε έναν media player που θα εκτελεί τις βασικές λειτουργίες ενός τέτοιου προγράμματος. Οι λειτουργίες που έχει αυτός ο media player έχουν απομονωθεί από το πληκτρολόγιο και το ποντίκι. Η εφαρμογή αυτή δουλεύει με συνδυασμό των υλικών της εταιρείας Phidgets. Σε αυτή την εφαρμογή θα χρησιμοποιήσουμε τα εξής υλικά: rotation sensor, voltage divider, slider, touch sensor.

Το rotation sensor χρησιμοποιείται για να επιλέξουμε κάποιο τραγούδι μέσα από μια λίστα έξι τραγουδιών(Εικόνα 22). Με την περιστροφή του rotation sensor η μοναδική περιοχή κειμένου που υπάρχει στο αρχικό παράθυρο παίρνει σαν κείμενο ακέραιες τιμές από 1 έως 6. Το rotation sensor παίρνει τιμές από 0 έως 999 και με μαθηματικούς τύπους αντιστοιχίζουμε τις τιμές του σε 6 μόλις αρχεία. Παρατηρούμε πως η περιοχή κειμένου δεν έχει δυνατότητα εισαγωγής δεδομένων από το πληκτρολόγιο και γενικότερα δυνατότητα επεξεργασίας κάτι το οποίο απαγορεύει στο χρήστη να εισάγει τιμές εκτός των επιτρεπτών ορίων(1-6).



Εικόνα 22

Μέχρι όμως να φορτωθεί το αρχείο καθώς και για την προετοιμασία της αναπαραγωγής περνάει κάποιος χρόνος. Έτσι θα πρέπει ο χρήστης να καταλάβει πως όλα πάνε καλά και πως το πρόγραμμα δεν «κρέμασε». Για το λόγο αυτό εμφανίζεται ένα άλλο παράθυρο που μας ενημερώνει πως όλα είναι σχεδόν έτοιμα(Εικόνα 23).

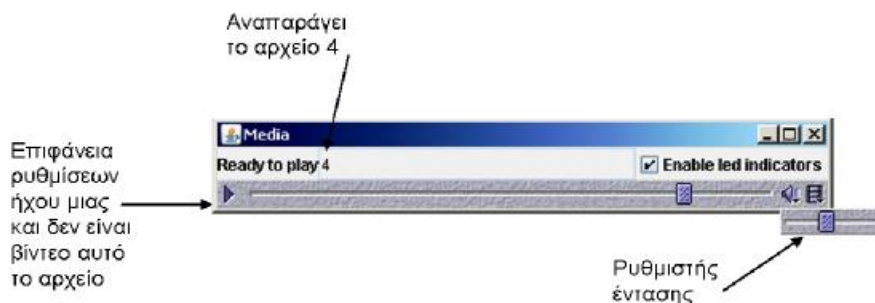
Εμφανίζεται μέχρι να ετοιμαστεί το πρόγραμμα και να φορτώσει το αρχείο



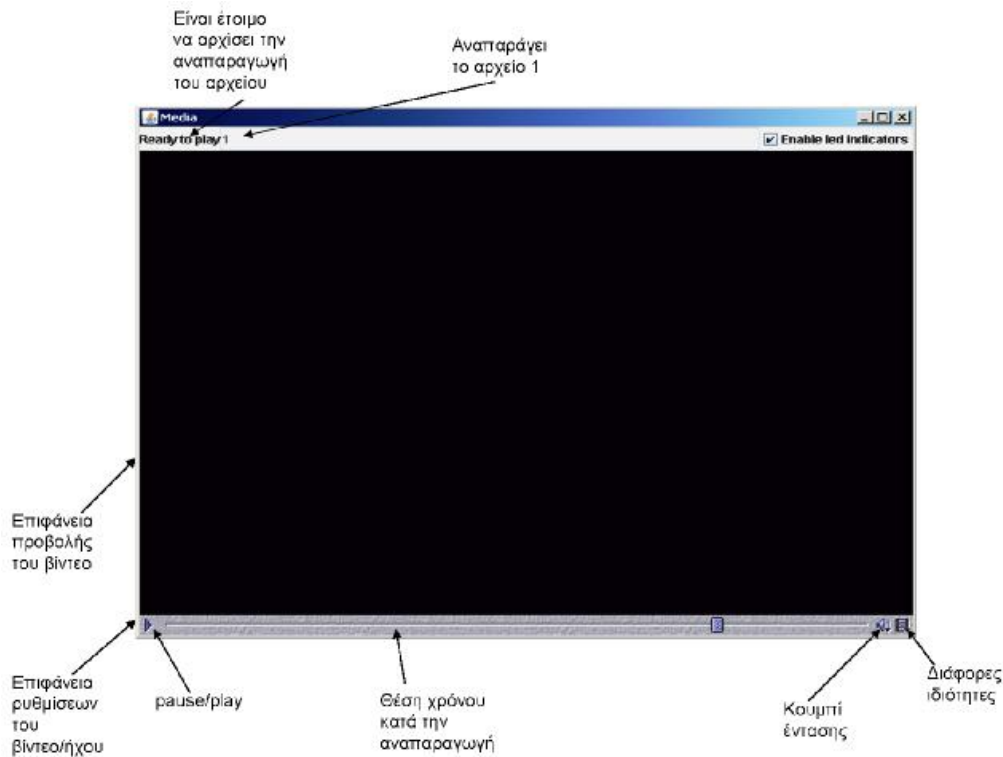
Εικόνα 23

Αφού όλα ετοιμαστούν το πρόγραμμά μας είναι έτοιμο να αναπαράγει το εκάστοτε αρχείο που του έχει ζητηθεί. Η εφαρμογή είναι σε θέση να αναπαράγει αρχεία ήχου(Εικόνα 24) και βίντεο(Εικόνα 25). Παρατηρούμε ότι η μοναδική διαφορά μεταξύ του παραθύρου αναπαραγωγής βίντεο και του παραθύρου αναπαραγωγής ήχου είναι η επιφάνεια προβολής βίντεο μιας και τα αρχεία ήχου επιστρέφουν ένα null αντικείμενο για την επιφάνεια βίντεο. Στο κάτω μέρος του παραθύρου παρατηρούμε την επιφάνεια ελέγχου. Υπάρχει ένα κουμπί για ξεκίνημα/παύση του βίντεο, υπάρχει ένας slider για να δείχνει το χρονικό σημείο του αρχείου που αναπαράγεται. Ακολουθούν άλλα δυο κουμπιά από τα οποία το ένα αναλαμβάνει να ρυθμίζει την ένταση της φωνής και το άλλο εμφανίζει κάποιες ιδιότητες του αρχείου. Με τη χρήση της τεχνολογίας αυτής μεταφέρουμε σε εξωτερικό υλικό τις εξής λειτουργίες: ξεκίνημα/παύση, σταμάτημα, ρύθμιση έντασης καθώς επίσης και τη λειτουργία του κλεισίματος του παραθύρου.

Επιπρόσθετα αξίζει να σημειωθεί πως με χρήση των ψηφιακών εισόδων και ψηφιακών εξόδων επιτυγχάνουμε μια πιο όμορφη εμφάνιση του υλικού γι' αυτό το πρόγραμμα. Αν το JCheckBox που υπάρχει στο παράθυρο της εφαρμογής μας είναι επιλεγμένο τότε ανάβουν κάποιες led ενδείξεις ανάλογα με το πόσο μεταβάλλουμε την ένταση της φωνής.



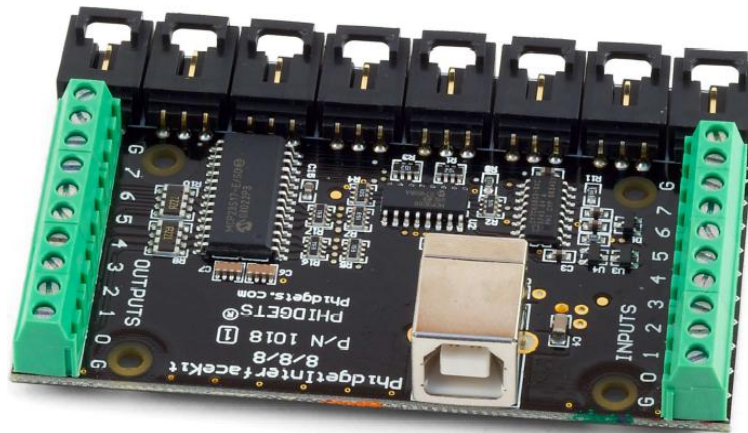
Εικόνα 24



Εικόνα 25

Συνεχίζοντας την ανάλυση του υλικού θα αναφέρουμε τον touch sensor. Πατώντας τον παίρνει μια τιμή 997-999. Μόλις τον αφήσουμε παίρνει την τιμή 0. Έτσι μόλις λάβουμε μια υψηλή τιμή απ' αυτόν καταλαβαίνουμε ότι πατήθηκε και στη συνέχεια εκτελείται η παύση ή η έναρξη της αναπαραγωγής. Ακολουθεί ο slider ο οποίος παίρνει τιμές από 0 έως 999 και αυτός. Μέσω αυτού ρυθμίζουμε την ένταση της φωνής και καθορίζουμε ποιο led θα ανάψει. Επειδή η φωνή ρυθμίζεται με δεκαδικό αριθμό μικρότερο του 1 θα πρέπει με μαθηματικό τύπο να μετατραπεί ο ακέραιος αριθμός που επιστρέφεται από τον slider σε δεκαδικό. Τέλος, ακολουθεί ο voltage divider ο οποίος παίρνει τιμές από 0 όταν δεν ακουμπάει κάτι πάνω του και αυξάνεται μέχρι 991 (μέχρι εκεί έχει παρατηρηθεί η μέγιστη τιμή) ανάλογα με το βάρος που βρίσκεται πάνω του. Έτσι αφού λάβουμε μια υψηλή τιμή απ' αυτόν σταματάει η αναπαραγωγή ολοκληρωτικά. Τέλος αφού ασχοληθήκαμε με το υλικό που επιστρέφει αναλογικές τιμές θα πρέπει να ρίξουμε και μια ματιά στο ψηφιακό υλικό. Το ψηφιακό υλικό που μου δίνει είσοδο είναι ένας απλός διακόπτης ενώ για ψηφιακή έξοδο έχω τα led. Με μεταβλητές boolean ελέγχουμε αν ο διακόπτης είναι ανοιχτός ή κλειστός καθώς επίσης με boolean μεταβλητές ελέγχουμε αν θα ανοίξει ή θα κλείσει κάποιο led. Όλα αυτά συνδέονται πάνω σ' ένα interface kit το οποίο συνδέεται μέσω usb στον Υ/Η. Ακολουθούν κάποιες εικόνες για να δούμε πως είναι το υλικό που χρησιμοποιούμε.





Εικόνα 26 - Interface kit



Εικόνα 27 - Rotation sensor



Εικόνα 28 - Slider



Εικόνα 29 - Touch sensor



Εικόνα 30 - Voltage divider



## **ΠΕΡΙΓΗΓΗΣΗ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ**

Αφού είδαμε και κατανοήσαμε τη λειτουργία της εφαρμογής και των υλικών μπορούμε να κάνουμε μια ανάλυση στον κώδικα για να δούμε πως επικοινωνεί το υλικό με την εφαρμογή και γενικότερα να δούμε τον τρόπο που χτίστηκε η εφαρμογή.

Η εφαρμογή αυτή αποτελείται από δυο κλάσεις (TheKit, Media), όπως και στην προηγούμενη η μια κλάση (TheKit) περιέχει τη μέθοδο main (γραμμές 311-315) η οποία καλείται πρώτη από το λειτουργικό σύστημα με σκοπό την εκκίνηση του προγράμματος και η δεύτερη κλάση (Media) είναι βοηθητική. Επεξηγώντας λίγο περισσότερο θα μπορούσαμε να πούμε ότι η πρώτη και βασική για το πρόγραμμα κλάση φροντίζει την επικοινωνία του χρήστη-υλικού και η δεύτερη αναλαμβάνει την επικοινωνία υλικού-παραθύρου εφαρμογής.

Στην κλάση «TheKit» ξεκινάω με τη δήλωση κάποιων μεταβλητών και σε κάποιες απ'αυτές εκχωρώ ορισμένες τιμές ως προεπιλεγμένες (γραμμές 11-16). Στη συνέχεια ακολουθεί ο δομητής της κλάσης (γραμμές 18-130) που αναλαμβάνει να ετοιμάσει το υλικό για λειτουργία. Εντός του δομητή υπάρχουν δυο διασυνδέσεις η πρώτη εκ των οποίων είναι ακροατής συμβάντων ψηφιακής εισόδου (γραμμές 27-39) και καλεί τη μέθοδο «ledPlay», η οποία θα επεξηγηθεί παρακάτω. Αμέσως μετά συναντούμε μια ακόμα διασύνδεση (γραμμές 40-123) που λειτουργεί ως ακροατής αναλογικών συμβάντων που ελέγχει από ποια συσκευή έρχεται η είσοδος από τον χρήστη και εν συνεχεία καλεί μια μέθοδο.

Με το πέρας του δομητή ακολουθεί η μέθοδος ledPlay(...) (γραμμές 127-310) η οποία όπως δηλώνει και το όνομα της σκοπός της είναι να «παίζει» με τις ενδείξεις των led! Η μέθοδος αυτή μας δείχνει κατά κάποιο τρόπο το πόσο δυνατά είναι ο ήχος. Στην αρχή όλα τα led είναι σβηστά και καθώς αυξάνεται ο ήχος από το μηδέν μέχρι το μέγιστο έχω τις εξής ενδείξεις με αυτή τη σειρά: τρεμοπαίζει το πράσινο led, σταθεροποιείται το πράσινο led, σβήνει το πράσινο και τρεμοπαίζει το κίτρινο led, σταθεροποιείται το κίτρινο led, σβήνει το κίτρινο led και τρεμοπαίζει το κόκκινο led και στο τέλος σταθεροποιείται το κόκκινο led. Στη μέθοδο αυτή υπάρχουν πολλαπλές υλοποιήσεις της διασύνδεσης ActionListener που βοηθούν διαφορετικά αντικείμενα javax.swing.Timer να τρεμοπαίζουν τα led (γραμμές 173-187, 220-234, 267-281).

Σειρά έχει η βασική μέθοδος main(...) που όπως παρατηρούμε φορτώνει πρώτα την κλάση που αντιστοιχεί στο υλικό (TheKit) και μετά την κλάση του

παραθύρου (media) και αυτό γίνεται για να έχει ετοιμαστεί το υλικό να δεχτεί εισόδο πριν η εφαρμογή γίνει ορατή στο χρήστη. Σε αντίθετη περίπτωση θα «πετάγονταν» εξαιρέσεις τύπου PhidgetException από σφάλμα εισόδου του υλικού. Ακολουθεί ο κώδικας για την κλάση «TheKit».

Στο τέλος ακολουθούν μέθοδοι που έχουν πρόσβαση σε διάφορα πεδία της κλάσης. Οι μέθοδοι αυτές είναι η setPauseState (γραμμές 316-318) η οποία ρυθμίζει την κατάσταση που βρίσκεται η εφαρμογή ως προς την αναπαραγωγή και η getAnalogInput (γραμμές 320-322) η οποία διαβάζει την κατάσταση της αναλογικής εισόδου του υλικού.

```

1 package TUI_App_2;
2
3 import java.awt.event.*;
4 import javax.swing.*;
5 import com.phidgets.*;
6 import com.phidgets.event.*;
7
8 //TheKit class is for phidgets
9 public class TheKit
10 {
11     private static InterfaceKitPhidget ikp;
12     private static Timer timerg=null,timero=null,timerr=null;
13     private int value,weightValue,track,previousTrack=-1,i=0;
14     private float volume;
15     private static Media med=null;
16     private static boolean
paused=true,analIn=true,openg=false,openo=false,openr=false;
17
18     public TheKit()
19     {
20         try
21         {
22             ikp=new InterfaceKitPhidget();
23             ikp.openAny();
24             ikp.waitForAttachment();
25             analIn=ikp.getInputState(0);
26             TheKit.ledPlay(analIn,ikp.getSensorValue(1),false);
27             ikp.addInputChangeListener(new InputChangeListener()
28             {
29                 public void inputChanged(InputChangeEvent evt)
30                 {
31                     analIn=evt.getState();
32                     try
33                     {
34
TheKit.ledPlay(analIn,ikp.getSensorValue(1),false);
35                     }
36                     catch(Exception e){}
37                     med.setSelected(analIn);
38                 }
39             });
40             ikp.addSensorChangeListener(new SensorChangeListener()
41             {
42                 public void sensorChanged(SensorChangeEvent evt)
43                 {

```

```

44         int position=evt.getIndex();
45         int valueB=0;
46         int valueW=0;
47         try
48         {
49             valueB=ikp.getSensorValue(2);
50             valueW=ikp.getSensorValue(3);
51             if(valueB<=50&valueW>=50)
52             {
53                 TheKit.ledPlay(false,0,true);
54             }
55         }
56         catch(Exception e){}
57         if(position==0)
58         {
59             //this is the rotor value
60             value=evt.getValue();
61             value=value==0 ? 1 : value;
62             //pick the track
63             track=(int)value/167+1;
64             med.setTextToField(""+track);
65             if(previousTrack!=track)
66             {
67                 try
68                 {
69                     med.setTextToLabel("Please wait...");
70                     med.totalStop();
71                     paused=true;
72                     med.remove(med.getControlComponent());
73                     med.remove(med.getVideoComponent());
74                 }
75                 catch(NullPointerException npe){}
76                 med.instantiateMedia();
77                 med.setTextToLabel("Almost ready... ");
78                 previousTrack=track;
79             }
80         }
81         else if(position==1)
82         {
83             //this is the slider value
84             value=evt.getValue();
85             TheKit.ledPlay(analIn,value,false);
86             volume=value*0.001f;
87             if(med.getReady())
88             {
89                 med.setVolume(volume);
90             }
91         }
92         else if(position==2)
93         {
94             //this is the button value
95             value=evt.getValue();
96             if((med.getReady())&(value<=50)&(paused))
97             {
98                 med.play();
99                 paused=false;
100            }
101            else if((med.getReady())&(value<=50)&(!paused))
102            {
103                med.pause();
104                paused=true;

```

```

105         }
106         else{}
107     }
108     else if(position==3)
109     {
110         //this is the weight sensor value
111         weightValue=evt.getValue();
112         if(weightValue<=50)
113         {
114             med.totalStop();
115             paused=true;
116         }
117     }
118     else
119     {
120         //what is this?
121     }
122 }
123 });
124 }
125 catch(Exception e){}
126 }
127 public static void ledPlay(boolean input,int sliderValue,boolean
close)
128 {
129     if(close)
130     {
131         try
132         {
133             TheKit.ikp.setOutputState(0,false);
134             TheKit.ikp.setOutputState(7,false);
135             TheKit.ikp.setOutputState(6,false);
136         }
137         catch(Exception e){}
138         finally
139         {
140             System.exit(0);
141         }
142     }
143     else
144     {
145         if(input)
146         {
147             if(sliderValue<5)
148             {
149                 try
150                 {
151                     if(timerg!=null)
152                     {
153                         timerg.stop();
154                         timerg=null;
155                     }
156                     ikp.setOutputState(0,false);
157                     ikp.setOutputState(7,false);
158                     ikp.setOutputState(6,false);
159                 }
160                 catch(Exception e){}
161             }
162             else if(sliderValue>=5&sliderValue<100)
163             {
164                 try

```

```

165         {
166             ikp.setOutputState(0,false);
167             ikp.setOutputState(7,false);
168         }
169         catch(Exception e){}
170         //on-off green
171         if(timerg==null)
172         {
173             timerg=new Timer(100,new ActionListener()
174             {
175                 @Override
176                 public void actionPerformed(ActionEvent
177                 evt)
178                 {
179                     try
180                     {
181                         openg=!openg;
182                     }
183                     catch(Exception e){}
184                 }
185             });
186         }
187         timerg.start();
188     }
189     else if(sliderValue>=100&sliderValue<300)
190     {
191         try
192         {
193             if(timerg!=null)
194             {
195                 timerg.stop();
196                 timerg=null;
197             }
198             if(timero!=null)
199             {
200                 timero.stop();
201                 timero=null;
202             }
203             ikp.setOutputState(0,false);
204             ikp.setOutputState(7,false);
205             ikp.setOutputState(6,true);
206         }
207         catch(Exception e){}
208     }
209     else if(sliderValue>=300&sliderValue<400)
210     {
211         try
212         {
213             ikp.setOutputState(6,false);
214             ikp.setOutputState(7,false);
215         }
216         catch(Exception e){}
217         //on-off orange
218         if(timero==null)
219         {
220             timero=new Timer(100,new ActionListener()
221             {
222                 @Override

```

```

223         public void actionPerformed(ActionEvent
224         evt)
225             {
226                 try
227                 {
228                     TheKit.ikp.setOutputState(0,openo);
229                     openo=!openo;
230                 }
231                 catch(Exception e){}
232             }
233         });
234         timero.start();
235     }
236     else if(sliderValue>=400&sliderValue<600)
237     {
238         try
239         {
240             if(timero!=null)
241             {
242                 timero.stop();
243                 timero=null;
244             }
245             if(timerr!=null)
246             {
247                 timerr.stop();
248                 timerr=null;
249             }
250             ikp.setOutputState(7,false);
251             ikp.setOutputState(6,false);
252             ikp.setOutputState(0,true);
253         }
254         catch(Exception e){}
255     }
256     else if(sliderValue>=600&sliderValue<750)
257     {
258         try
259         {
260             ikp.setOutputState(0,false);
261             ikp.setOutputState(6,false);
262         }
263         catch(Exception e){}
264         //on-off red
265         if(timerr==null)
266         {
267             timerr=new Timer(100,new ActionListener()
268             {
269                 @Override
270                 public void actionPerformed(ActionEvent
271                 evt)
272                 {
273                     try
274                     {
275                         TheKit.ikp.setOutputState(7,openr);
276                         openr=!openr;
277                     }
278                     catch(Exception e){}
279                 }
280             });

```

```

280         }
281         timerr.start();
282     }
283     else
284     {
285         try
286         {
287             if(timerr!=null)
288             {
289                 timerr.stop();
290                 timerr=null;
291             }
292             ikp.setOutputState(0,false);
293             ikp.setOutputState(6,false);
294             ikp.setOutputState(7,true);
295         }
296         catch(Exception e){}
297     }
298 }
299 else
300 {
301     try
302     {
303         ikp.setOutputState(0,false);
304         ikp.setOutputState(7,false);
305         ikp.setOutputState(6,false);
306     }
307     catch(Exception e){}
308 }
309 }
310 }
311 public static void main(String[] args)
312 {
313     new TheKit();
314     med=new Media();
315 }
316 public static void setPauseState(boolean state){
317     paused=state;
318 }
319
320 public static boolean getAnalogInputState(){
321     return analIn;
322 }
323 }

```

Η δεύτερη και τελευταία κλάση που ονομάζεται Media είναι αυτή που δημιουργεί το παράθυρο επικοινωνίας με το χρήστη. Ξεκινώντας τη μελέτη της κλάσης αυτής παρατηρώ πως είναι υποκλάση της JFrame (γραμμή 18). Ο δομητής της δημιουργεί το παράθυρο που βλέπει-ακούει ο χρήστης τα αρχεία πολυμέσων (γραμμές 30-56). Μέσα του υπάρχει μια ανώνυμη εσωτερική κλάση που λειτουργεί ως ακροατής συμβάντων παραθύρου αλλά μπαίνει σε λειτουργία μόνο σε συμβάν κλεισίματος του παραθύρου (γραμμές 34-41).

Ακολουθεί η μέθοδος instantiateMedia (γραμμές 58-102). Δουλειά της είναι να φορτώνει το αρχείο προς αναπαραγωγή καθώς και να φορτώνει αντικείμενα τύπου

javax.media.Player μέσω της ανώνυμης κλάσης ControllerAdapter και να το ετοιμάζει για την αναπαραγωγή του εκάστοτε αρχείου (γραμμές 67-97).

Υπάρχουν άλλες τέσσερις μέθοδοι οι οποίες στοχεύουν στην καλύτερη λειτουργία-χρήση του προγράμματος από κάποιον τρίτο προγραμματιστή. Η μέθοδος play() (γραμμές 103-106) όπως δηλώνει και το όνομα της αναλαμβάνει να αρχίσει την αναπαραγωγή, η μέθοδος pause() (γραμμές 107-111) κάνει παύση ενώ η totalStop() (γραμμές 112-116) σταματάει την αναπαραγωγή και σε περίπτωση που πατηθεί το κουμπί που αρχίζει την αναπαραγωγή θα αρχίσει να το παίζει απ' την αρχή το αρχείο. Τελευταία μέθοδος της κλάσης είναι η setVolume(...) (γραμμές 117-121) και σκοπός της είναι να ρυθμίζει την ένταση του ήχου. Ακολουθεί ο κώδικας της κλάσης Media.

Τέλος, ακολουθούν κάποιες μέθοδοι που δίνουν πρόσβαση στα πεδία της κλάσης από άλλες κλάσεις. Οι μέθοδοι αυτές είναι η setSelected (γραμμές 123-125), setTextToField (γραμμές 126-128), setTextToLabel (γραμμές 129-131), getControlComponent (γραμμές 132-134), getVideoComponent (γραμμές 135-137), setReady (γραμμές 138-140), getReady (γραμμές 141-143).

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package TUI_App_2;
7
8  /**
9   *
10  * @author AUTOEXE.INF
11  */
12
13  import java.awt.*;
14  import java.awt.event.*;
15  import javax.swing.*;
16  import javax.media.*;
17
18  public class Media extends JFrame
19  {
20      private JPanel pane,pane2;
21      private JLabel label;
22      private JTextField field;
23      private JCheckBox box;
24      private MediaLocator loc;
25      private Player pl;
26      private Component video,control;
27      private Media med;
28      private boolean ready=false;
29
30      public Media()
31      {
32          super("Media");

```



```

33     setBounds(100,100,320,100);
34     this.addWindowListener(new WindowAdapter()
35     {
36         @Override
37         public void windowClosing(WindowEvent evt)
38         {
39             TheKit.ledPlay(false,0,true);
40         }
41     });
42     pane=new JPanel();
43     pane.setLayout(new BorderLayout());
44     pane2=new JPanel();
45     pane2.setLayout(new BorderLayout());
46     label=new JLabel("Set the song: ");
47     pane2.add(label,BorderLayout.WEST);
48     field=new JTextField(20);
49     field.setEditable(false);
50     box=new JCheckBox("Enable led
indicators",TheKit.getAnalogInputState());
51     pane2.add(box,BorderLayout.EAST);
52     pane2.add(field,BorderLayout.CENTER);
53     pane.add(pane2,BorderLayout.NORTH);
54     setContentPane(pane);
55     setVisible(true);
56 }
57
58 public void instatiateMedia()
59 {
60     try
61     {
62         String track=field.getText();
63         String url="file:///D:/ptyxiakhSounds/"+track+".mpg";
64
65         loc=new MediaLocator(url);
66         pl=javax.media.Manager.createPlayer(loc);
67         pl.addControllerListener(new ControllerAdapter()
68         {
69             @Override
70             public void realizeComplete(RealizeCompleteEvent evt)
71             {
72                 pl.prefetch();
73             }
74             @Override
75             public void prefetchComplete(PrefetchCompleteEvent evt)
76             {
77
78                 try
79                 {
80                     video=pl.getVisualComponent();
81                     pane.add(video,BorderLayout.CENTER);
82                 }
83                 catch(NullPointerException npe){}
84                 control=pl.getControlPanelComponent();
85                 pane.add(control,BorderLayout.SOUTH);
86                 validate();
87                 pack();
88                 ready=true;
89                 label.setText("Ready to play");
90             }
91             @Override
92             public void endOfMedia(EndOfMediaEvent evt)

```

```

93         {
94             pl.setMediaTime(new Time(0));
95             TheKit.setPauseState(true);
96             pl.stop();
97         }
98     });
99     pl.realize();
100 }
101 catch(Exception e){}
102 }
103 public void play()
104 {
105     pl.start();
106 }
107 public void pause()
108 {
109
110     pl.stop();
111 }
112 public void totalStop()
113 {
114     pl.stop();
115     pl.setMediaTime(new Time(0));
116 }
117 public void setVolume(float value)
118 {
119     GainControl con=pl.getGainControl();
120     con.setLevel(value);
121 }
122
123 public void setSelected(boolean selected){
124     box.setSelected(selected);
125 }
126 public void setTextToField(String text){
127     field.setText(text);
128 }
129 public void setTextToLabel(String text){
130     label.setText(text);
131 }
132 public Component getControlComponent(){
133     return control;
134 }
135 public Component getVideoComponent(){
136     return video;
137 }
138 public void setReady(boolean state){
139     ready=state;
140 }
141 public boolean getReady(){
142     return ready;
143 }
144 }

```

## **ΕΦΑΡΜΟΓΗ 3 - ΑΝΙΧΝΕΥΣΗ ΕΙΣΒΟΛΕΑ ΚΑΙ ΕΠΟΠΤΕΙΑ ΧΩΡΟΥ**

### **ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ ΚΑΙ ΤΟ ΥΛΙΚΟ**

Το ζητούμενο αυτή τη φορά είναι να φτιάξουμε μια εφαρμογή η οποία θα ανιχνεύει αν υπάρχει κίνηση στο χώρο και μόλις ανιχνεύσει κίνηση να δίνει στον εισβολέα συγκεκριμένο χρόνο κάθε φορά για να επιβεβαιώσει την ταυτότητα του και αν δεν είναι ένας εγκεκριμένος χρήστης θα πρέπει η κάμερα να αρχίσει να καταγράφει το δωμάτιο και να αποθηκεύει στο σκληρό δίσκο το βίντεο ή και να το στέλνει στο δίκτυο, ανάλογα με τις ρυθμίσεις που έχει εισάγει ο χρήστης.

Εδώ όμως έρχεται ένα κρίσιμο ερώτημα ασφαλείας. Τι θα συμβεί σε περίπτωση που ο κακόβουλος επισκέπτης κλείσει το πρόγραμμα είτε πατώντας το κλείσιμο στο παράθυρο σε περίπτωση που του επιτρέπει το πρόγραμμα μου να κλείσει ή, ακόμα και αν του αφαιρέσουμε τη δυνατότητα κλεισίματος, να το κλείσει μέσα από τη διαχείριση εργασιών; Η δομή της εφαρμογής αυτής καταφέρνει να λύσει αυτό το πρόβλημα.

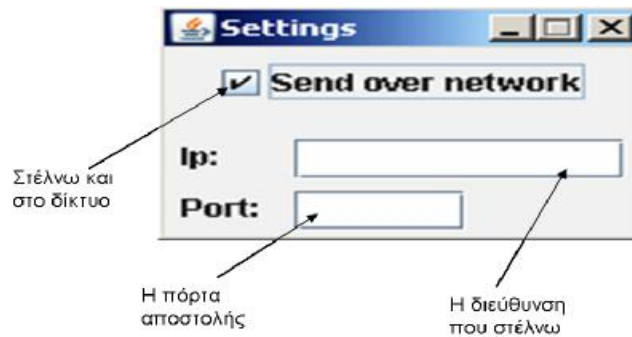
Η εφαρμογή στην ουσία είναι δυο εκτελέσιμα αρχεία. Το πρώτο εκτελέσιμο αρχείο είναι αυτό που εφαρμόζει ο χρήστης τις ρυθμίσεις και αυτό που ανιχνεύει κίνηση και επιβεβαιώνει τον χρήστη. Αν αυτό το πρόγραμμα κλείσει με επιβεβαίωση του σωστού χρήστη δημιουργεί ένα αρχείο. Το δεύτερο εκτελέσιμο είναι αυτό που αναλαμβάνει την εγγραφή βίντεο. Αυτό εκτελείται αμέσως μόλις περάσουμε στο επόμενο παράθυρο απ' αυτό που ρυθμίζουμε την αποστολή βίντεο. Το δεύτερο εκτελέσιμο προσπαθεί να μετονομάσει το πρώτο δίνοντας του το παλιό του όνομα βέβαια. Όσο δε μπορεί να κάνει τη μετονομασία σημαίνει πως το αρχείο χρησιμοποιείται και δεν έχει κλείσει, και γι' αυτό το λειτουργικό σύστημα δε δίνει τη δυνατότητα μετονομασίας. Μόλις μας δοθεί η δυνατότητα αυτή σημαίνει πως το αρχείο έκλεισε και τώρα το μόνο που μένει είναι να δούμε αν έκλεισε σωστά ή βίαια. Αν έκλεισε σωστά υπάρχει ένα αρχείο το οποίο όπως προαναφέραμε δημιουργείται από το άλλο εκτελέσιμο, αλλιώς δε δημιουργείται το αρχείο αυτό. Αν έχει δημιουργηθεί το αρχείο εμφανίζεται ένα μήνυμα καλωσορίσματος αλλιώς αρχίζει να καταγράφει η κάμερα. Ας δούμε όμως πως εμφανίζεται αυτή η εφαρμογή στο χρήστη και πως λειτουργούν οι επιλογές της.

Ξεκινώντας από το πρώτο παράθυρο θα πρέπει να πούμε ότι εδώ είναι το σημείο που ο χρήστης καθορίζει αν η εφαρμογή θα αποθηκεύει το βίντεο στο

σκληρό(Εικόνα 31) ή και θα το στέλνει στο δίκτυο σε κάποια συγκεκριμένη διεύθυνση(Εικόνα 32). Μέσω της ψηφιακής εισόδου μπορούμε είτε να επιλέξουμε το JCheckBox είτε όχι. Αξίζει να σημειωθεί πως έχει χαθεί η δυνατότητα επιλογής του με το ποντίκι(έχουν αφαιρεθεί όλοι οι προεπιλεγμένοι MouseListeners από το JCheckBox) ώστε να υπάρχει συγχρονισμός μεταξύ της τιμής που έχει η ψηφιακή είσοδος του υλικού και σ' αυτό που δείχνει το παράθυρο.



Εικόνα 31



Εικόνα 32

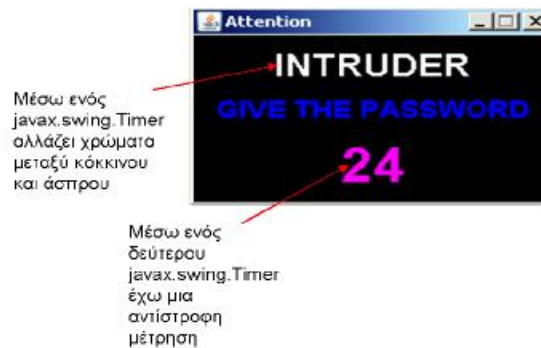
Μόλις τελειώσουμε με το παράθυρο των ρυθμίσεων αποστολής και αφού εκτελέσουμε το άλλο αρχείο, πατάμε τον αισθητήρα αφής για να γίνουν αποδεκτές οι ρυθμίσεις από το πρόγραμμα και να ετοιμαστεί η εφαρμογή να δεχτεί δεδομένα από τον υπέρυθρο αισθητήρα κίνησης αλλά και να ανοίξει τον RFIDReader για να δέχεται είσοδο. Όσο δεν κινείται κάτι μπροστά από τον αισθητήρα η εφαρμογή παραμένει σε αναμονή(Εικόνα 33).



Περιμένει να περάσει κάτι από τον υπέρυθρο αισθητήρα

Εικόνα 33

Μόλις περάσει κάτι από τον υπέρυθρο αισθητήρα η εφαρμογή αλλάζει εμφάνιση για μια ακόμα φορά(Εικόνα 34) και δίνει στον εισβολέα κάποιο χρόνο για να επιβεβαιώσει την ταυτότητα του μέσω ενός tag. Σε περίπτωση που το tag που περάσει δεν είναι το σωστό η αντίστροφη μέτρηση γίνεται πιο γρήγορα. Αν η εφαρμογή κλείσει βίαια, δηλαδή με οποιονδήποτε τρόπο πέραν της χρήσης του σωστού tag, δεν αφήνει κάποιο υπόλειμμα ενώ αν κλείσει με ομαλό τρόπο αφήνει ένα αρχείο ως υπόλειμμα και εμφανίζει ένα παράθυρο καλωσορίσματος του χρήστη(Εικόνα 35).



Μέσω ενός `javax.swing.Timer` αλλάζει χρώματα μεταξύ κόκκινου και άσπρου

Μέσω ενός δεύτερου `javax.swing.Timer` έχω μια αντίστροφη μέτρηση

Εικόνα 34



Εικόνα 35 - Υπέρυθρος αισθητήρας



Εικόνα 36

Όμως υπάρχει και μια ακόμα εφαρμογή που τρέχει κάτω στην παγίδα (δίπλα στο ρολόι) η οποία αναλαμβάνει να καταγράψει τον εισβολέα. Η εφαρμογή αυτή με σκοπό να παραμείνει όσο πιο αόρατη γίνεται δεν έχει ούτε εικονίδιο ούτε κάποιο άλλο αναδυόμενο μενού. Δουλεύει με έναν `javax.swing.Timer` ο οποίος ελέγχει αν η άλλη εφαρμογή χρησιμοποιείται ακόμα και σε περίπτωση που δε χρησιμοποιείται ελέγχει αν υπάρχει το υπόλειμμα. Σε περίπτωση που υπάρχει το αρχείο όλα είναι μια χαρά, αλλιώς αρχίζει η κάμερα να καταγράφει τις κινήσεις του εισβολέα και το βίντεο αποστέλλεται στην διεύθυνση που ορίσαμε ή αποθηκεύεται στο σκληρό δίσκο.

### **ΠΕΡΙΗΓΗΣΗ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ**

Όπως προαναφέραμε η εφαρμογή αυτή χωρίζεται σε δυο εκτελέσιμα αρχεία. Στο εκτελέσιμο αρχείο που ελέγχουμε την κάμερα υπάρχει μόλις μια κλάση, η `CamRecorder`. Στο άλλο εκτελέσιμο υπάρχουν τέσσερις κλάσεις, η `TagChecker`, η `Attention`, η `Welcome`, η `NowFirstFrame`.

Η `CamRecorder` αποτελείται από τον δομητή ο οποίος αναλαμβάνει τη δημιουργία και εκτέλεση όλου του προγράμματος. Μέσα στον δομητή η εφαρμογή πηγαίνει στην παγίδα (γραμμές 29-31, 90) και αρχίζει να εκτελείται ο `javax.swing.Timer` που ελέγχει αν θα πρέπει η κάμερα να αρχίσει να καταγράφει (γραμμές 52-88). Αναλυτικότερα, στο κενό εικονίδιο που εμφανίζεται στην παγίδα μέσω ενός ακροατή συμβάντων ποντικιού μπορώ να κλείσω την εφαρμογή κάνοντας δεξί κλικ (γραμμές 36-48). Ο `Timer` που χρησιμοποιείται ελέγχει μέσω τριών συνθηκών `if`, ανά 100 milliseconds, αν θα πρέπει να αρχίσει η εγγραφή η όχι. Σε περίπτωση που πρέπει να αρχίσει υπάρχουν δυο διαφορετικές μέθοδοι. Η μια απ' αυτές μόνο αποθηκεύει το βίντεο στο σκληρό ενώ η άλλη το στέλνει και στο δίκτυο. Η `captureAndSaveAndTransmit(...)` (γραμμές 95-168) διαβάζει από δυο αρχεία, από το ένα διαβάζει τη διεύθυνση γραμμές 99-110) και από το άλλο τη θύρα γραμμές 111-122) που θα το στέλνει. Επίσης η μέθοδος αυτή κάνει διπλή αποστολή, δηλαδή

στέλνει σε δυο διευθύνσεις (η μια αποστολή γίνεται στη διεύθυνση που έχει ορίσει ο χρήστης (γραμμές 137-139) και η άλλη στέλνεται τοπικά (γραμμές 141-142) με σκοπό να πάρει το πρόγραμμα αυτό το βίντεο ως είσοδο για να το αποθηκεύσει στο σκληρό (γραμμές 183-206) ). Ενώ η `captureAndSave()` είναι η μέθοδος η οποία μόνο αποθηκεύει στο σκληρό.

Πρέπει να σημειωθεί πως για την καταγραφή από την κάμερα, στη χρήση της `CaptureUtil.getCaptureDS(...)` (γραμμές 125,187) , έχουν χρησιμοποιηθεί εξωτερικές κλάσεις. Αυτές είναι οι εξής: `CaptureUtil`, `MonitorCDS`, `MonitorStream` καθώς και κάποιες από τις μεθόδους που εμπεριέχονται στις κλάσεις αυτές. Οι κλάσεις αυτές προέρχονται από τη Sun Microsystems και εμπεριέχονται στο πακέτο με όνομα `jmfsample`.

Στη συνέχεια ακολουθεί η μέθοδος `finish()` (γραμμές 170-182) η οποία αναλαμβάνει να κλείσει οτιδήποτε χρησιμοποιείται για την εγγραφή και την αποστολή βίντεο καθώς και να αποθηκεύσει.

Τέλος, ακολουθεί η μέθοδος `main(...)`(γραμμές 207-210) η οποία αλληλεπιδρά με το λειτουργικό σύστημα για να ξεκινήσει η εφαρμογή να εκτελείται. Αυτή καλεί τον δομητή της `startMonitor`.

```
1 package TUI_App_3;
2
3
4
5 //import java
6 import java.io.*;
7 import java.awt.*;
8 import java.awt.event.*;
9 import java.net.*;
10 //import javax
11 import javax.swing.*;
12 import javax.media.*;
13 import javax.media.control.*;
14 import javax.media.datasink.*;
15 import javax.media.format.*;
16 import javax.media.protocol.*;
17 import javax.media.rtp.*;
18
19 public class CamRecorder
20 {
21     private SystemTray st;
22     private TrayIcon icon;
23     private Processor p,pr;
24     private DataSink outFile;
25     private boolean entered=false;
26     private Timer time;
27     public CamRecorder()
28     {
29         if(SystemTray.isSupported())
30         {
```

```

31     try
32     {
33         st=SystemTray.getSystemTray();
34         ImageIcon ii=new ImageIcon("");
35         icon=new TrayIcon(ii.getImage());
36         icon.addMouseListener(new MouseAdapter()
37         {
38             @Override
39             public void mouseClicked(MouseEvent evt)
40             {
41                 int b=evt.getButton();
42                 if(b==3)
43                 {
44                     st.remove(icon);
45                     //stop the monitoring to save the captured
data
46                     finish();
47                     //exit
48                     System.exit(0);
49                 }
50             }
51         });
52         time=new Timer(100,new ActionListener()
53         {
54             public void actionPerformed(ActionEvent evt)
55             {
56                 try
57                 {
58                     File f=new File("");
59                     File jar=new
File(f.getCanonicalPath()+"\\first.jar");
60
61                     //System.out.println(jar.getCanonicalPath());
62                     File helper=new
File(f.getCanonicalPath()+"\\help");
63                     File ipf=new
File(f.getCanonicalPath()+"\\ip.txt");
64                     File portf=new
File(f.getCanonicalPath()+"\\port.txt");
65                     if(jar.exists())
66                     {
67                         if(jar.renameTo(jar))
68                         {
69                             if((!helper.exists())&(!entered)&(!ipf.exists())&(!portf.exists()))
70                             {
71                                 //start monitoring and save
72                                 entered=true;
73                                 time.stop();
74                                 captureAndSave();
75                             }
76                             else
77                             if((!helper.exists())&(!entered)&(ipf.exists())&(portf.exists()))
78                             {
79                                 //start monitoring and save and
transmit
80                                 entered=true;
81                                 time.stop();
captureAndSaveAndTransmit(ipf,portf);

```



```

82         }
83     }
84 }
85 }
86         catch(Exception e){System.err.println("i:
"+e.getMessage());}
87     }
88 });
89     time.start();
90     st.add(icon);
91 }
92     catch(Exception e){System.err.println(e.getMessage());}
93 }
94 }
95 public void captureAndSaveAndTransmit(File ipf,File portf)
96 {
97     try
98     {
99         //ip text
100        FileInputStream fis=new FileInputStream(ipf);
101        int i=-1;char c=' ';StringBuffer sb=new StringBuffer();
102        do
103        {
104            i=fis.read();
105            c=(char)i;
106            sb.append(c);
107        }while(i>-1);
108        String ip=sb.toString();
109        ip=ip.substring(0,ip.length()-1);
110        System.out.println(ip);
111        //port text
112        FileInputStream fis1=new FileInputStream(portf);
113        int i1=-1;char c1=' ';StringBuffer sb1=new StringBuffer();
114        do
115        {
116            i1=fis1.read();
117            c1=(char)i1;
118            sb1.append(c1);
119        }while(i1>-1);
120        String port=sb1.toString();
121        port=port.substring(0,port.length()-1);
122        System.out.println(port);
123        //video work to do
124        VideoFormat video=new VideoFormat(VideoFormat.JPEG_RTP);
125        DataSource source=CaptureUtil.getCaptureDS(video,null);
126        Format[] f=new Format[]{video};
127        FileTypeDescriptor ftd=new
FileTypeDescriptor(FileTypeDescriptor.RAW_RTP);
128        ProcessorModel pm=new ProcessorModel(source, f, ftd);
129        p=Manager.createRealizedProcessor(pm);
130        DataSource ds=p.getDataOutput();
131        PushBufferDataSource pbds=(PushBufferDataSource)ds;
132        PushBufferStream[] pbs=pbds.getStreams();
133        javax.media.rtp.RTPManager
rtp=javax.media.rtp.RTPManager.newInstance();
134        SessionAddress local=new
SessionAddress(InetAddress.getLocalHost(),1230);
135        rtp.initialize(local);
136        //transmit over network
137        SessionAddress netDest=new
SessionAddress(InetAddress.getByName(ip),

```

```

138         Integer.parseInt(port));
139         rtp.addTarget(netDest);
140         //transmit to me
141         SessionAddress locDest=new
SessionAddress(InetAddress.getLocalHost(),1232);
142         rtp.addTarget(locDest);
143
144         SendStream ss=rtp.createSendStream(ds,0);
145         ss.start();
146         p.start();
147         //save to hard disk
148         MediaLocator ml=new MediaLocator("rtp://" +
149
InetAddress.getLocalHost().getHostAddress()+
150             ":"+1232+"/video");
151         Format[] f2=new Format[]{new
VideoFormat(VideoFormat.JPEG_RTP)};
152         FileTypeDescriptor ftd2=new
FileTypeDescriptor(FileTypeDescriptor.QUICKTIME);
153         ProcessorModel pm2=new ProcessorModel(ml, f2, ftd2);
154         pr=Manager.createRealizedProcessor(pm2);
155         DataSource ds2=pr.getDataOutput();
156         File f1=new File("");
157         File jar=new File(f1.getCanonicalPath()+"\\capture.flv");
158         String dest=jar.getCanonicalPath();
159         dest=dest.replaceAll("\\\\", "/");
160         dest="file:///"+dest;
161         System.out.println(dest);
162         MediaLocator ml2=new MediaLocator(dest);
163         outFile=Manager.createDataSink(ds2, ml2);
164         outFile.open();
165         outFile.start();
166         pr.start();
167     }
168     catch(Exception e){}
169 }
170 public void finish()
171 {
172     try
173     {
174         p.stop();
175         p.close();
176         outFile.stop();
177         outFile.close();
178         pr.stop();
179         pr.close();
180     }
181     catch(Exception e){}
182 }
183 public void captureAndSave() throws Exception
184 {
185     VideoFormat vf=new VideoFormat(VideoFormat.MJPEG);
186     //AudioFormat af=new
AudioFormat(AudioFormat.LINEAR,44100,16,2);
187     DataSource in=CaptureUtil.getCaptureDS(vf,null);
188     Format[] f=new Format[1];
189     f[0]=vf;
190     FileTypeDescriptor ftd=new
FileTypeDescriptor(FileTypeDescriptor.QUICKTIME);
191     ProcessorModel pm=new ProcessorModel(in, f, ftd);
192     p=Manager.createRealizedProcessor(pm);

```

```

193     DataSource out=p.getDataOutput();
194     File file=new File("");
195     File avi=new File(file.getCanonicalPath()+"\\capture.flv");
196     //convert \\ to /
197     String path=avi.getCanonicalPath();
198     path=path.replaceAll("\\\\", "/");
199     path="file:///"+path;
200     System.out.println(path);
201     MediaLocator ml=new MediaLocator(path);
202     outFile=Manager.createDataSink(out, ml);
203     outFile.open();
204     outFile.start();
205     p.start();
206 }
207 public static void main(String[] args)
208 {
209     new CamRecorder();
210 }
211 }

```

Στη δεύτερη εφαρμογή υπάρχουν περισσότερες κλάσεις. Ξεκινώντας θα αναφερθούμε στην tagChecker.

Η κλάση TagChecker είναι υποκλάση της JFrame (γραμμή 10), επομένως εμφανίζει ένα παράθυρο(το παράθυρο που εμφανίζεται μέχρι να περάσει κάτι από τον υπέρυθρο αισθητήρα). Δουλειές της είναι να κάνει βίαιη έξοδο και να κλείνει το υλικό που περιμένει είσοδο μόλις ο εισβολέας κλείσει το τρέχον παράθυρο (γραμμές 27-38), να διαγράψει το αρχείο-υπόλειμμα που μπορεί να υπάρχει από προηγούμενη εκτέλεση της εφαρμογής κατά την υλοποίηση του παραθύρου(γραμμές 47-53). Αξίζει να σημειωθεί πως χρησιμοποιείται κώδικας html (γραμμή 41) για την μορφοποίηση του κειμένου που φαίνεται στο παράθυρο. Εκτός από τον δομητή που εκτελεί τα παραπάνω η κλάση αυτή έχει και κάποιες μεθόδους. Μια απ'αυτές βέβαια είναι και η main(...)(γραμμές 59-65). Μετά ακολουθεί η μέθοδος openSensor() (γραμμές 66-142) η οποία προετοιμάζει το υλικό για να δεχθεί ψηφιακή (γραμμές 77-80) και αναλογική είσοδο (γραμμές 81-139). Η ψηφιακή είσοδος αλλάζει την κατάσταση που βρίσκεται το JCheckBox (γραμμή 78). Επίσης δέχεται είσοδο από τον υπέρυθρο αισθητήρα (γραμμή 90) που προχωράει την εφαρμογή ένα βήμα (γραμμές 94-104) παρακάτω καθώς και από τον αισθητήρα αφής (γραμμή 91), αλλά μόνο που σε αυτή την περίπτωση πριν προχωρήσει η εφαρμογή στο επόμενο επίπεδο γίνεται κάποιος έλεγχος για την ορθότητα των δεδομένων (γραμμές 105-136).

Λόγω της φιλοσοφίας της γλώσσας έχει δημιουργηθεί μια ακόμα μέθοδος που δίνει πρόσβαση σε ένα πεδίο της κλάσης. Η μέθοδος αυτή είναι η getAttentionObject (γραμμές 143-145).

```

1 package TUI_App_3;
2
3 import java.io.*;
4 import java.awt.*;
5 import java.awt.event.*;
6 import javax.swing.*;
7 import com.phidgets.*;
8 import com.phidgets.event.*;
9
10 public class TagChecker extends JFrame
11 {
12     private JPanel pane;
13     private static InterfaceKitPhidget pik;
14     private static Attention att;
15     private static TagChecker tag;
16     private static NowFirstFrame nff;
17     public TagChecker()
18     {
19         super();
20         setTitle("Waiting");
21         setBounds(100,100,250,80);
22         setResizable(false);
23         //setDefaultCloseOperation(EXIT_ON_CLOSE);
24         addWindowListener(new WindowAdapter()
25         {
26             @Override
27             public void windowClosing(WindowEvent evt)
28             {
29                 try
30                 {
31                     pik.close();
32                 }
33                 catch(Exception e){System.err.println(e.getMessage());}
34                 finally
35                 {
36                     System.exit(0);
37                 }
38             }
39         });
40         pane=new JPanel();
41         JLabel label=new JLabel("<html><center>" +
42             "<b><font size='6'>IS ANYONE HERE?</font></b>");
43         label.setForeground(Color.RED);
44         pane.add(label);
45         setContentPane(pane);
46         //delete the previous help file
47         try
48         {
49             File f=new File("");
50             File helper=new File(f.getCanonicalPath()+"\\help");
51             helper.delete();
52         }
53         catch(Exception e){System.err.println(e.getMessage());}
54         //open the sensor
55         //openSensor();
56         setVisible(true);
57         //pack();
58     }
59     public static void main(String[] args)
60     {
61         nff=new NowFirstFrame();

```

```

62         //tag=new TagChecker();
63         //new attention();
64         //new welcome();
65     }
66     public static void openSensor()
67     {
68         try
69         {
70             pik=new InterfaceKitPhidget();
71             pik.openAny();
72             pik.waitForAttachment();
73             NowFirstFrame.setNetworkState(pik.getInputState());
74             pik.addInputChangeListener(new InputChangeListener()
75             {
76                 public void inputChanged(InputChangeEvent evt)
77                 {
78                     NowFirstFrame.setComboboxSelected(evt.getState());
79                 }
80             });
81             pik.addSensorChangeListener(new SensorChangeListener()
82             {
83                 public void sensorChanged(SensorChangeEvent evt)
84                 {
85                     int value0=0,value1=0;
86                     //sto 0 na mpei o sensor gia kinsh
87                     //sto 1 o touch sensor
88                     try
89                     {
90                         value0=pik.getSensorValue(0);
91                         value1=pik.getSensorValue(1);
92                     }
93                     catch(Exception
e){System.err.println(e.getMessage());}
94                     if(value0>200)
95                     {
96                         //someone passed
97                         try
98                         {
99                             tag.dispose();
100                            att=new Attention();
101                            pik.close();
102                        }
103                        catch(Exception e){System.err.println("phidget:
"+e.getMessage());}
104                    }
105                    if(value1<100)
106                    {
107                        if(TagChecker.tag==null)
108                        {
109                            //an checked elegxos to periexomeno twm
field
110                            boolean perOk=false;
111                            if(NowFirstFrame.getNetworkState())
112                                perOk=NowFirstFrame.checkTheFields();
113                            //an ola ok,checked kai ok periexomena
114                            if(NowFirstFrame.getNetworkState())
115                            {
116                                if(perOk)
117                                {
118                                    // grafw se arxeio
119                                    NowFirstFrame.store();

```

```

120 //hide to 1 kai show to allo mesw
shmeiwshs V
121 TagChecker.nff.dispose();
122 TagChecker.tag=new TagChecker();
123 }
124 else
125 {
126 //beep kai menei to idio para8yro
127 Toolkit.getDefaultToolkit().beep();
128 }
129 }
130 else
131 {
132 //hide to 1 kai show to allo mesw
shmeiwshs V
133 TagChecker.nff.dispose();
134 TagChecker.tag=new TagChecker();
135 }
136 }
137 }
138 }
139 });
140 }
141 catch(Exception e){System.err.println("phidget:
"+e.getMessage());}
142 }
143 public static Attention getAttentionObject(){
144 return att;
145 }
146 }

```

Δεύτερη έρχεται η κλάση Attention (γραμμές 13-156) που και αυτή εμφανίζει ένα παράθυρο (γραμμή 18). Χρησιμοποιεί έναν ακροατή συμβάντων παραθύρου (γραμμές 33-49) με σκοπο να κλείνει το παραθυρο, να σβήνουν τα led και να κλείνει ο RFID reader. Χρησιμοποιεί δυο javax.swing.Timer για κάποιου είδους εφέ (αναβοσβήνει μια λέξη ανά 300 milliseconds (γραμμές 54-66) και εμφανίζει μια αντίστροφη μέτρηση (γραμμές 81-108) για 30 δευτερόλεπτα). Όμως προτού εμφανιστεί το παράθυρο στο χρήστη καλείται η μέθοδος openRFIDReader() (γραμμές 116-160). Η openRFIDReader() ετοιμάζει τον rfid reader να αρχίζει να διαβάζει τα tags που περνούν (γραμμές 123-134). Αν περάσει το σωστό tag σταματάει ο Timer που κάνει την αντίστροφη μέτρηση (γραμμές 139-149), αλλιώς αν περάσει λάθος tag αντί να κάνει το μέτρημα ανά δευτερόλεπτο το κάνει ανά 100 milliseconds σαν ποινή για τον εισβολέα (γραμμή 152).

```

1 /*
2 * To change this template, choose Tools | Templates
3 * and open the template in the editor.
4 */
5
6 package TUI_App_3;
7
8 /**

```

```

9  *
10 * @author AUTOEXE.INF
11 */
12 import java.awt.*;
13 import java.awt.event.*;
14 import javax.swing.*;
15 import com.phidgets.*;
16 import com.phidgets.event.*;
17
18 public class Attention extends JFrame
19 {
20     private JPanel pane2;
21     private JLabel countDown,intruder;
22     private static Welcome wel;
23     private RFIDPhidget rfid;
24     private String tagValue;
25     private javax.swing.Timer time2;
26     public Attention()
27     {
28         super();
29         setTitle("Attention");
30         setBounds(100,100,230,150);
31         setResizable(false);
32         //setDefaultCloseOperation(EXIT_ON_CLOSE);
33         addWindowListener(new WindowAdapter()
34         {
35             @Override
36             public void windowClosing(WindowEvent evt)
37             {
38                 try
39                 {
40                     rfid.setLEDOn(false);
41                     rfid.close();
42                 }
43                 catch(Exception e){System.err.println(e.getMessage());}
44                 finally
45                 {
46                     System.exit(0);
47                 }
48             }
49         });
50         pane2=new JPanel();
51         intruder=new JLabel("<html><center> " +
52             "<b><font size='6'>INTRUDER</font></b>");
53         //start the intruder Timer
54         javax.swing.Timer time=new javax.swing.Timer(300,new
ActionListener()
55         {
56             public void actionPerformed(ActionEvent evt)
57             {
58                 if(intruder.getForeground().equals(Color.RED))
59                 {
60                     intruder.setForeground(Color.WHITE);
61                 }
62                 else
63                 {
64                     intruder.setForeground(Color.RED);
65                 }
66             }
67         });
68         intruder.setForeground(Color.RED);

```

```

69     pane2.setBackground(Color.BLACK);
70     pane2.add(intruder);
71     JLabel pass=new JLabel("<html><center>" +
72         "<b><font size='5'>GIVE THE PASSWORD</font></b>");
73     pass.setForeground(Color.BLUE);
74     pane2.add(pass);
75     countdown=new JLabel("<html><center>" +
76         "<b><font size='7'>30</font></b>");
77     countdown.setName("30");
78     countdown.setForeground(Color.MAGENTA);
79     pane2.add(countDown);
80     //start the countDown Timer
81     time2=new javax.swing.Timer(1000,new ActionListener()
82     {
83         public void actionPerformed(ActionEvent evt)
84         {
85             String number=countDown.getName();
86             int no=Integer.parseInt(number);
87             no=no-1;
88             number=String.valueOf(no);
89             if(no>10)
90             {
91                 countDown.setText("<html><center>" +
92                     "<b><font
size='7'>" +number+"</font></b>");
93                 countDown.setName(number);
94             }
95             else if((no<11)&(no>0))
96             {
97                 countDown.setText("<html><center>" +
98                     "<b><font
size='7'>" +number+"</font></b>");
99                 Toolkit.getDefaultToolkit().beep();
100                countDown.setName(number);
101            }
102            else if(no<1)
103            {
104                countDown.setName(number);
105                System.exit(0);
106            }
107        }
108    });
109    setContentPane(pane2);
110    //anoigw ton rfid reader
111    openRFIDReader();
112    setVisible(true);
113    time.start();
114    time2.start();
115 }
116 public void openRFIDReader()
117 {
118     try
119     {
120         rfid=new RFIDPhidget();
121         rfid.openAny();
122         rfid.waitForAttachment();
123         rfid.addTagGainListener(new TagGainListener()
124         {
125             public void tagGained(TagGainEvent evt)
126             {
127                 try

```



```

128         {
129             rfid.setLEDOn(true);
130             tagValue=evt.getValue();
131         }
132         catch(Exception
133 e){System.err.println(e.getMessage());}
134     });
135     rfid.addTagLossListener(new TagLossListener()
136     {
137         public void tagLost(TagLossEvent evt)
138         {
139             try
140             {
141                 rfid.setLEDOn(false);
142                 if(tagValue.equals("3100bdf6de"))
143                 {
144                     TagChecker.getAttentionObject().dispose();
145                     time2.stop();
146                     time2=null;
147                     wel=new Welcome();
148                     rfid.close();
149                 }
150                 else
151                 {
152                     time2.setDelay(100);
153                 }
154             }
155             catch(Exception
156 e){System.err.println(e.getMessage());}
157         }
158     });
159     catch(Exception e){System.err.println(e.getMessage());}
160 }
161 }

```

Ακολουθεί η κλάση Welcome (γραμμές 18-41). Η κλάση αυτή είναι ένα παράθυρο (γραμμή 18) καλωσορίσματος για τον εξουσιοδοτημένο χρήστη (γραμμές 28-29) μόνο από πλευρά εμφάνισης. Στην ουσία όμως η κλάση αυτή λίγο πριν γίνει ορατό το παράθυρο στο χρήστη δημιουργεί ένα αρχείο (γραμμές 32-37) (αυτό που ονομάσαμε υπόλειμμα σε προηγούμενες αναφορές) για να δείξει ομαλή έξοδο από το πρόγραμμα.

```

1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6 package TUI_App_3;
7
8 /**
9  *
10 * @author AUTOEXE.INF
11 */
12 //import java

```

```

13 import java.io.*;
14 import java.awt.*;
15 import java.awt.event.*;
16 import javax.swing.*;
17
18 public class Welcome extends JFrame
19 {
20     private JPanel pane3;
21     public Welcome()
22     {
23         super("Welcome");
24         setBounds(100,100,150,80);
25         setResizable(false);
26         setDefaultCloseOperation(EXIT_ON_CLOSE);
27         pane3=new JPanel();
28         pane3.add(new JLabel("<html><center>" +
29             "<b><font size='6'>Hi Bill...</font></b>"));
30         setContentPane(pane3);
31         //create the file
32         try
33         {
34             File f=new File("");
35             File helper=new File(f.getCanonicalPath()+"\\help");
36             helper.createNewFile();
37         }
38         catch(Exception e){System.err.println(e.getMessage());}
39         setVisible(true);
40     }
41 }

```

Τέλος υπάρχει και άλλη μια κλάση, η NowFirstFrame (γραμμές 19-230), η οποία όμως χρειάζεται μεγαλύτερη ανάλυση από τις προηγούμενες. Η κλάση αυτή είναι και αυτή υποκλάση της JFrame (γραμμή 19). Το παράθυρο που δημιουργεί αποτελείται από ένα JCheckBox (γραμμή 37) και από ένα ακόμα JPanel που χρησιμοποιεί διάταξη καρτών στα στοιχεία του( γραμμές 62-97). Γι' αυτό και μόλις αλλάζει η κατάσταση στο JCheckBox αλλάζει αυτό που δείχνει το JFrame (γραμμές 98-107). Επίσης λίγο πριν γίνει ορατό το παράθυρο διαγράφει τα προηγούμενα αρχεία που εμπεριέχουν τη διεύθυνση αποστολής (γραμμές 110-116) καθώς και αυτό που έχει τη θύρα αποστολής (γραμμές 117-123). Σειρά έχει η μέθοδος checkTheFields()(γραμμές 126-191). Η μέθοδος αυτή ελέγχει αν τα πεδία έχουν λογικές τιμές (π.χ. αν οι τιμές είναι νούμερα και όχι χαρακτήρες, αν η πόρτα που δίνεται είναι μέσα στο πλαίσιο θυρών από 1 έως 65535 και αν η διεύθυνση είναι σε μορφή διεύθυνσης). Τέλος, η μέθοδος store() (γραμμές 192-220) είναι αυτή που αποθηκεύει σε αρχεία την διεύθυνση (γραμμές 197-206) και την θύρα αποστολής (γραμμές 208-217).

Λόγω της φιλοσοφίας της γλώσσας έχουν δημιουργηθεί κάποιες ακόμα μέθοδοι που δίνουν πρόσβαση σε πεδία της κλάσης. Οι μέθοδοι αυτές είναι η

setNetworkState (γραμμές 221-223), η getNetworkState (γραμμές 224-227), setComboBoxSelected (γραμμές 227-229). Οι δυο πρώτες ρυθμίζουν την κατάσταση αποστολής βίντεο και η τελευταία επιλέγει ή αποεπιλέγει το checkbox.

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package TUI_App_3;
7
8  /**
9   *
10  * @author AUTOEXE.INF
11  */
12  import java.io.*;
13  import java.awt.*;
14  import java.awt.event.*;
15  import java.util.*;
16  import javax.swing.*;
17  import javax.swing.event.*;
18
19  public class NowFirstFrame extends JFrame
20  {
21      private static JCheckBox box;
22      private JTextArea area;
23      private static JTextField ip,port;
24      private JPanel cards,card1,card2;
25      private static boolean state=false;
26      public static final String F="FALSE",T="TRUE";
27      public NowFirstFrame()
28      {
29          super("Settings");
30          setBounds(100,100,170,120);
31          setResizable(false);
32          setDefaultCloseOperation(EXIT_ON_CLOSE);
33          //twra open sensor
34          TagChecker.openSensor();
35          JPanel pane=new JPanel();
36          //edw getstate apo to analog input
37          box=new JCheckBox("Send over network",state);
38          MouseListener[] ml=box.getMouseListeners();
39          for(int i=0;i<ml.length;i++)
40              box.removeMouseListener(ml[i]);
41          box.addItemListener(new ItemListener()
42          {
43              @Override
44              public void itemStateChanged(ItemEvent evt)
45              {
46                  boolean bool=box.isSelected();
47                  if(bool)
48                  {
49                      CardLayout cl=(CardLayout)cards.getLayout();
50                      cl.show(cards,T);
51                  }
52                  else
53                  {
54                      CardLayout cl=(CardLayout)cards.getLayout();

```

```

55         cl.show(cards,F);
56     }
57     validate();
58 }
59 });
60 pane.add(box);
61 //card1
62 card1=new JPanel();
63 area=new JTextArea(5,1);
64 area.setText("The default value is not to \nsend video over the
network.");
65 area.setEditable(false);
66 area.setBackground(card1.getBackground());
67 card1.add(area);
68 //card2
69 card2=new JPanel();
70 GroupLayout layout=new GroupLayout(card2);
71 card2.setLayout(layout);
72 layout.setAutoCreateGaps(true);
73 layout.setAutoCreateContainerGaps(true);
74 JLabel ipl=new JLabel("Ip:");
75 JLabel portl=new JLabel("Port:");
76 ip=new JTextField(10);
77 ip.setMaximumSize(ip.getPreferredSize());
78 port=new JTextField(5);
79 port.setMaximumSize(port.getPreferredSize());
80 layout.setHorizontalGroup(layout.createSequentialGroup())
81
.addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
82         .addComponent(ipl)
83         .addComponent(portl))
84
.addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
85         .addComponent(ip)
86         .addComponent(port));
87     layout.setVerticalGroup(layout.createSequentialGroup())
88
.addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
89         .addComponent(ipl)
90         .addComponent(ip))
91
.addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
92         .addComponent(portl)
93         .addComponent(port));
94     cards=new JPanel();
95     cards.setLayout(new CardLayout());
96     cards.add(card1,F);
97     cards.add(card2,T);
98     if(state)
99     {
100         CardLayout cl=(CardLayout)cards.getLayout();
101         cl.show(cards,T);
102     }
103     else
104     {
105         CardLayout cl=(CardLayout)cards.getLayout();
106         cl.show(cards,F);
107     }
108     pane.add(cards, BorderLayout.CENTER);
109     setContentPane(pane);
110     try

```

```

111     {
112         File f=new File("");
113         File ipf=new File(f.getCanonicalPath()+"\\ip.txt");
114         ipf.delete();
115     }
116     catch(Exception e){System.err.println(e.getMessage());}
117     try
118     {
119         File f=new File("");
120         File portf=new File(f.getCanonicalPath()+"\\port.txt");
121         portf.delete();
122     }
123     catch(Exception e){System.err.println(e.getMessage());}
124     setVisible(true);
125 }
126 public static boolean checkTheFields()
127 {
128     boolean check1=false,check2=false;
129     Vector<Integer> in=new Vector<Integer>();
130     if(state)
131     {
132         String ips=ip.getText();
133         //check1z
134         try
135         {
136             String[] ipsa=ips.split("\\.");
137             for(int i=0;i<ipsa.length;i++)
138             {
139                 System.out.println("Contains: "+ipsa[i]);
140                 in.add(Integer.parseInt(ipsa[i]));
141             }
142             check1=true;
143             System.out.println(ipsa.length);
144             if(ipsa.length>4)
145             {
146                 check1=false;
147             }
148             if(check1)
149             {
150                 for(int i=0;i<in.size();i++)
151                 {
152                     if(in.get(i)>255||in.get(i)<0)
153                     {
154                         check1=false;
155                         break;
156                     }
157                 }
158             }
159         }
160     catch(Exception
161 e){System.err.println("check1");check1=false;}
161     ///////////////////////////////////////////////////
162     String ports=port.getText();
163     //check2
164     try
165     {
166         int p=Integer.parseInt(ports);
167         check2=true;
168         if(p>65535)
169         {
170             check2=false;

```

```

171         }
172         else if(p<0)
173         {
174             check2=false;
175         }
176     }
177     catch(Exception
e){System.err.println(e.getMessage());check2=false;}
178     if(check1&check2)
179     {
180         return true;
181     }
182     else
183     {
184         return false;
185     }
186 }
187 else
188 {
189     return false;
190 }
191 }
192 public static void store()
193 {
194     try
195     {
196         //ip
197         File parenti=new File("");
198         File outputi=new
File(parenti.getCanonicalPath()+"\\ip.txt");
199         FileWriter fosi=new FileWriter(outputi);
200         BufferedWriter bosi=new BufferedWriter(fosi);
201         PrintWriter psi=new PrintWriter(bosi);
202         String st1=ip.getText();
203         psi.print(st1);
204         psi.close();
205         bosi.close();
206         fosi.close();
207         //port
208         File parentp=new File("");
209         File outputp=new
File(parentp.getCanonicalPath()+"\\port.txt");
210         FileWriter fosp=new FileWriter(outputp);
211         BufferedWriter bosp=new BufferedWriter(fosp);
212         PrintWriter psp=new PrintWriter(bosp);
213         String st2=port.getText();
214         psp.print(st2);
215         psp.close();
216         bosp.close();
217         fosp.close();
218     }
219     catch(Exception e){System.err.println(e.getMessage());}
220 }
221 public static void setNetworkState(boolean netState){
222     state=netState;
223 }
224 public static boolean getNetworkState(){
225     return state;
226 }
227 public static void setComboboxSelected(boolean selected){
228     box.setSelected(selected);

```

```
229     }  
230 }
```

## **ΕΦΑΡΜΟΓΗ 4 - ΑΠΟΣΤΟΛΗ LIVE ΒΙΝΤΕΟ ΜΕΣΩ INTERNET ΣΕ ΕΠΙΛΕΓΜΕΝΟΥΣ ΧΡΗΣΤΕΣ**

### **ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ ΚΑΙ ΤΟ ΥΛΙΚΟ**

Η εφαρμογή αυτή στοχεύει στο να μοιράζεται κάποιος επιλεγμένες στιγμές της ζωής του με κάποιους από τους φίλους του. Αυτή είναι η κεντρική ιδέα της εφαρμογής αυτής αλλά ας εντρυφήσουμε στις επιμέρους λειτουργίες της.

Μόλις εκτελέσουμε την εφαρμογή εμφανίζεται το παράθυρο της εικόνας 37 και ενεργοποιείται και ο υπέρυθρος αισθητήρας κίνησης. Μέχρι να ανιχνευθεί κίνηση εξακολουθεί να είναι ορατό το αρχικό παράθυρο. Μόλις όμως περάσει κάτι από τον αισθητήρα εμφανίζεται ένα καινούριο παράθυρο(Εικόνα 38). Την ίδια χρονική στιγμή ενεργοποιείται και η κάμερα και αρχίζει η καταγραφή του χώρου που βρίσκεται. Μέχρις ότου το παράθυρο αυτό πάρει το σωστό του μέγεθος περιμένουμε έτσι ώστε να προετοιμαστεί η κάμερα και οι άλλοι πόροι του προγράμματος να ολοκληρώσουν τις διάφορες εργασίες που πιθανόν να κάνουν ώστε να κυλήσει ομαλά η εκτέλεση της εφαρμογής χωρίς λάθη-εξαιρέσεις.

Πλέον το πρόγραμμα μας είναι έτοιμο να δεχθεί είσοδο περνώντας τα διάφορα tags πάνω από τον rfid reader. Οι τιμές του καθενός απ'αυτά είναι μοναδικές από τον κατασκευαστή τους. Έτσι χρησιμοποιούμε μια βάση δεδομένων(Εικόνα 42 στο κεφάλαιο 38) για να αποθηκεύσουμε διάφορες ρυθμίσεις όπως το όνομα ενός φίλου, την διεύθυνση και την θύρα αποστολής καθώς και τη διαδρομή μιας εικόνας που θέλουμε να φαίνεται για κάθε φίλο που έχουμε επιλέξει. Οι εικόνες 40 και 41 μας δείχνουν πως φαίνεται η καρτέλα για καθέναν από τους φίλους μου. Θα πρέπει να πούμε πως υπάρχει και μια καρτέλα με το όνομα του χρήστη. Η ροζ περιοχή στην εικόνα 42 είναι περιοχή όπου φαίνεται τι καταγράφει εκείνη τη στιγμή η κάμερα.

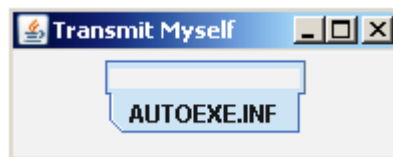
Αφού ολοκληρώσαμε την ανάλυση από μεριά εμφάνισης θα πρέπει να ασχοληθούμε και με τον χειρισμό της εφαρμογής αυτής. Όπως προαναφέραμε η εφαρμογή είναι δομημένη σε καρτέλες και ο μοναδικός τρόπος προσπέλασης των καρτελών είναι μέσω ενός ρότορα. Παρέχεται επίσης η δυνατότητα μετακίνησης του παραθύρου με τη χρήση ενός joystick και σε περίπτωση που χρειαστεί να επαναφέρουμε το παράθυρο στην αρχική του θέση χρησιμοποιούμε έναν αισθητήρα πίεσης. Στο υλικό που χρησιμοποιείται υπάρχει και ένας αισθητήρας αφής που



σταματάει ή αρχίζει πάλι τη μετάδοση στο φίλο που αντιπροσωπεύει η ενεργή καρτέλα. Αξίζει να σημειωθεί πως αν πατηθεί μαζί ο αισθητήρας αφής και ο αισθητήρας πίεσης η εφαρμογή κλείνει. Η εφαρμογή έχει τη δυνατότητα να σταματήσει ή να ξαναρχίσει τη μετάδοση σε όλους τους φίλους ταυτόχρονα μέσω ενός διακόπτη. Σε περίπτωση όμως που έχω πολλούς φίλους πως μπορώ να ξέρω αν στέλνω σε όλους ή όχι; Αυτό το σημείο έρχεται να καλύψει η χρήση τριών led. Έτσι: αν ανάβει το πράσινο στέλνω σε όλους, αν ανάβει το πορτοκαλί στέλνω σε μερικούς, αλλιώς αν ανάβει το κόκκινο δε στέλνω σε κανέναν. Επιπρόσθετα αξίζει να σημειωθεί πως πέρα από τις ενδείξεις led το όνομα της κάθε καρτέλας είναι κόκκινο για κατάσταση παύσης και μαύρο για κατάσταση αποστολής. Ας δούμε πως είναι ορατή η εφαρμογή στον χρήστη.



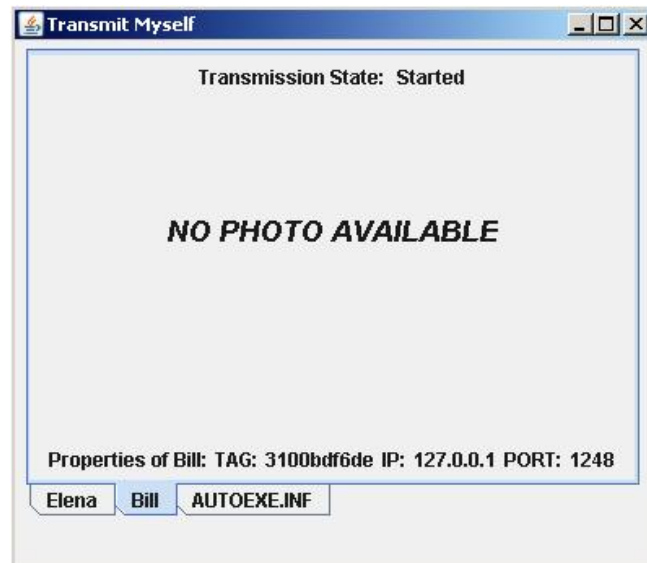
Εικόνα 37



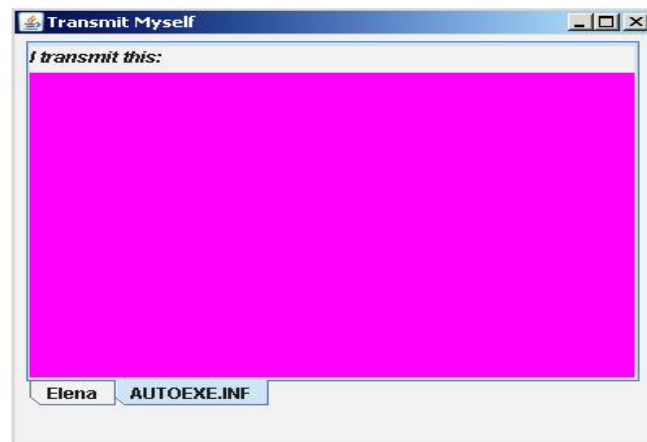
Εικόνα 38



Εικόνα 39



Εικόνα 40



Εικόνα 41

## **ΠΕΡΙΓΗΓΗΣΗ ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ**

Η ΒΔ είναι μια απλή βάση όπως αυτή που συναντήσαμε στην πρώτη μας εφαρμογή. Για τη δημιουργία της ΒΔ έχει χρησιμοποιηθεί η Microsoft Access.

Η ΒΔ περιλαμβάνει έναν μόνο πίνακα ο οποίος αποτελείται από 5 στήλες οι οποίες βοηθούν το χρήστη να κάνει ρυθμίσεις στο πρόγραμμα χωρίς να χρειαστεί να επέμβει στον κώδικα. Η στήλη «TAG» αντιπροσωπεύει τις διάφορες τιμές των RFID tags τα οποία αντιστοιχούν σε δεδομένα φίλων του χρήστη, που είναι αποθηκευμένοι στη βάση, και είναι το πρωτεύον κλειδί της πίνακα. Η στήλη «NAME» έχει τα ονόματα των φίλων που αποστέλλεται το βίντεο και χρησιμοποιείται από το πρόγραμμα για να δείχνει στο χρήστη το όνομα του κάθε ατόμου που επικοινωνεί. Οι δυο στήλες που

ακολουθούν «IP address» και «port» εκφράζουν τη διεύθυνση ip που έχει ο εκάστοτε φίλος μας και την πόρτα που έχει ανοίξει ο φίλος για να δέχεται τα δεδομένα που του στέλνει η εφαρμογή. Στη τελευταία στήλη «Photo» μπορούμε να βάζουμε μια διαδρομή που αντιστοιχεί σε μια φωτογραφία που θα θέλαμε να εμφανίζεται στην καρτέλα που έχει δημιουργήσει η εφαρμογή για τον φίλο αυτόν.

Θα πρέπει να δοθεί ιδιαίτερη σημασία στις ρυθμίσεις για την διεύθυνση και την πόρτα διότι χωρίς αυτά δε θα μπορεί να υλοποιηθεί καθόλου η σύνδεση, αλλά θα πρέπει να προσέξουμε και την τιμή των tag καθώς ένα λάθος μπορεί να προκαλέσει στην εφαρμογή μια ορατή δυσλειτουργία όπως να μην τρέχει η εφαρμογή ή να τρέχει αλλά να στέλνει το βίντεο σε κάποιον άλλο φίλο του χρήστη χωρίς να το γίνει αντιληπτό.

friendsData : Πίνακας					
	TAG	NAME	IP address	port	Photo
	01065e8486	Jim	127.0.0.1	1234	
	0106708856	Chris	127.0.0.1	1235	
	0f00ad62bb	Nick	127.0.0.1	1236	
	3000b4e006	John	127.0.0.1	1237	
	31006d3a4d	Tom	127.0.0.1	1238	
	3100ac2f27	Maria	127.0.0.1	1239	
	3100b3bb80	Anastasia	127.0.0.1	1240	
	3100bdf6de	Bill	127.0.0.1	1248	
	3100be20d2	Katerina	127.0.0.1	1242	
	3100c3efe1	Elena	127.0.0.1	1244	D:\Documents and Settings\AUTOEXE.INF\Ει
*					

Εικόνα 42

## ΠΕΡΙΓΗΓΗΣΗ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή αυτή αποτελείται από τρεις κλάσεις, τη AppHandler η οποία εμπεριέχει τη συνάρτηση main και υλοποιεί το παράθυρο, τη PhidgetHandler η οποία αναλαμβάνει την επικοινωνία του υλικού με την εφαρμογή και την DatabaseHandler που αναλαμβάνει τη σύνδεση με τη βάση δεδομένων. Όπως όλα τα προηγούμενα προγράμματα έτσι και αυτό υλοποιείται με τη χρήση συναρτήσεων για την καλύτερη επεξήγηση του κώδικα αλλά και συντήρηση του.

Ο δομητής της βασικής κλάσης της εφαρμογής μας δημιουργεί το παράθυρο (γραμμές 41,85) υλοποιεί έναν ακροατή κλεισίματος παραθύρου (γραμμές 52-67), προετοιμάζει μέσα σε ένα νήμα κάποιο μέρος από το υλικό που χρησιμοποιείται (γραμμές 74-81). Συνεχίζοντας μέσα στην ίδια κλάση ακολουθεί η συνάρτηση myExit() (γραμμές 86-118) που ελέγχει πως κλείνει το παράθυρο (αν κλείνει από την

οπτική διεπαφή χρήστη (γραμμές 88-96) ή από την απτή διεπαφή χρήστη (γραμμές 97-117) ) και κλείνει με διαφορετικό τρόπο προκειμένου να αποφύγει τις εξαιρέσεις που «πετάγονται». Η `addFriend` μέθοδος έχει ως παράμετρο τα δεδομένα από τη βάση τα οποία αποθηκεύονται σε `Vectors`(γραμμές 126-129). Σύμφωνα με την κατάσταση αποστολής εμφανίζονται τα κατάλληλα μηνύματα (γραμμές 138-147) και στη συνέχεια ρυθμίζουμε τις διαστάσεις της φωτογραφίας(γραμμές 129-172). Έπειτα εμφανίζονται τα χαρακτηριστικά για τον συγκεκριμένο χρήστη στην καρτέλα του (γραμμές 173-188) και ξεκινάει η αποστολή βίντεο(γραμμές 187-194). Η συνάρτηση `setVisibleIndex()`(γραμμές 207-219) παίρνει την τιμή που έχει ο ρότορας και φέρνει στην επιφάνεια διαφορετική καρτέλα καθώς αυτός περιστρέφεται. Η συνάρτηση `checkAndOpenTheLed()`(γραμμές 221-273) ελέγχει τις καταστάσεις αποστολής βίντεο και ανάβει το κατάλληλο λαμπάκι. Η `pauseAll()`(γραμμές 274-318) στοχεύει στο να σταματάει συνολικά την αποστολή και να την ξαναρχίζει μαζικά. Η `startOrPause()`(γραμμές 318-349) σταματάει (γραμμές 322-336) η αρχίζει (γραμμές 338-345) την αποστολή βίντεο στον φίλο που αντιπροσωπεύει η ενεργή καρτέλα και στη συνέχεια ανάβει το κατάλληλο λαμπάκι(γραμμή 346). Η `updateFromSensor()`(γραμμές 350-408) αναλαμβάνει να απενεργοποιήσει όλους τους ακροατές συμβάντων από το ποντίκι για τις καρτέλες(γραμμές 360-363), να ενεργοποιήσει τον RFID reader (γραμμή 367) και να προετοιμάσει την κάμερα για αποστολή (γραμμές 369-376) αλλά και να εμφανίσει την πρώτη καρτέλα με τίτλο το όνομα του χρήστη η οποία προβάλλει το βίντεο που στέλνει αυτή τη χρονική στιγμή στους φίλους του(γραμμές 388-406). Πρέπει επίσης να σημειωθεί πως για την καταγραφή από την κάμερα, στη χρήση της `CaptureUtil.getCaptureDS(...)`(γραμμή 371), έχουν χρησιμοποιηθεί εξωτερικές κλάσεις. Αυτές είναι οι εξής:`CaptureUtil`, `MonitorCDS`, `MonitorStream` καθώς και κάποιες από τις μεθόδους που εμπεριέχονται στις κλάσεις αυτές. Οι κλάσεις αυτές προέρχονται από τη Sun Microsystems και εμπεριέχονται στο πακέτο με όνομα `jmfsample`. Ακολουθούν οι `moveUp()`(γραμμές 422-427), `moveDown()`(γραμμές 428-433), `moveRight()`(γραμμές 416-421), `moveLeft()`(γραμμές 410-415) και `returnToDefault()`(γραμμές 434-437) που αναλαμβάνουν να μετακινούν το παράθυρο πάνω, κάτω, δεξιά, αριστερά και να το επαναφέρουν στην αρχική του θέση αντίστοιχα. Η `closeCamera()`(γραμμές 438-449) κλείνει τα διάφορα αντικείμενα του προγράμματος που χρησιμοποιούν την κάμερα.

Τέλος, υπάρχουν δυο ακόμα μέθοδοι οι οποίες δίνουν πρόσβαση σε πεδία της κλάσης. Οι μέθοδοι αυτές είναι η `getAppHandlerObject` (γραμμές 454-456) η οποία

επιστρέφει ένα αντικείμενο τύπου AppHandler και η getTabObject (γραμμές 457-459) η οποία επιστρέφει ένα αντικείμενο τύπου JTabbedPane.

```
1 package TUI_App_4;
2
3 import java.io.*;
4 import java.awt.*;
5 import java.awt.image.*;
6 import java.awt.event.*;
7 import java.net.*;
8 import java.util.*;
9 import javax.swing.*;
10 import javax.imageio.*;
11 import javax.media.*;
12 import javax.media.rtp.*;
13 import javax.media.format.*;
14 import javax.media.protocol.*;
15
16 public class AppHandler extends JFrame
17 {
18     private JPanel pane,paneTab,vpane;
19     private JLabel inf;
20     private PhidgetHandler phi,phr;
21     private static AppHandler me;
22     private JTabbedPane tab;
23     private Vector<String> verifier,ipVector,portVector,imageVector;
24     private Vector<Boolean> boolVector;
25     private Vector<JLabel> condVector;
26     private Vector dataSink,mediaLocator;
27     private DataSource dsIn,dsOut;
28     private Processor pr;
29     private ProcessorModel pm;
30     private FileTypeDescriptor ftd;
31     private RTPManager rtp;
32     private PushBufferDataSource pbds;
33     private PushBufferStream[] pbs;
34     private SessionAddress local,dest;
35     private SendStream ss;
36     private Player pl;
37     private boolean entered=false;
38     private DataSource out;
39     private javax.swing.Timer timer;
40
41     public AppHandler()
42     {
43         super("Transmit Myself");
44         //setResizable(false);
45         setBounds(100,100,200,80);
46         verifier=new Vector<String>(10);
47         ipVector=new Vector<String>(10);
48         portVector=new Vector<String>(10);
49         imageVector=new Vector<String>(10);
50         boolVector=new Vector<Boolean>(10);
51         condVector=new Vector<JLabel>(10);
52         addWindowListener(new WindowAdapter()
53         {
54             @Override
55             public void windowClosing(WindowEvent evt)
56             {
```

```

57         try
58         {
59             myExit('x');
60         }
61         catch(Exception e){}
62         finally
63         {
64             System.exit(0);
65         }
66     }
67 });
68 pane=new JPanel();
69 inf=new JLabel();
70 inf.setText("<html><center><i><b><font size=\"5\">" +
71             "NOBODY HERE...</font></b></i></html>");
72 pane.add(inf);
73 setContentPane(pane);
74 Thread thr=new Thread(new Runnable()
75 {
76     public void run()
77     {
78         phi=new PhidgetHandler('i');
79     }
80 });
81 thr.start();
82 dataSink=new Vector(10);
83 mediaLocator=new Vector(10);
84 setVisible(true);
85 }
86 public void myExit(char x_Or_p)//x->mouse close , p->phidget close
87 {
88     if(x_Or_p=='x')
89     {
90         //close interface
91         phi.closeInterface();
92         //close RFID
93         phr.closeRfid();
94         //close camera
95         closeCamera();
96     }
97     else if(x_Or_p=='p')
98     {
99         //close the Leds
100        for(int i=0;i<8;i++)
101        {
102            try
103            {
104                phi.getInterfaceKitObject().setOutputState(i,false);
105            }
106            catch(Exception e){}
107        }
108        try
109        {
110            phr.getRFIDReaderObject().setLEDon(false);
111        }
112        catch(Exception e){}
113        //close RFID
114        phr.closeRfid();
115        //close camera
116        closeCamera();

```

```

117     }
118 }
119 public void addFriend(String[] data)
120 {
121     if(!verifier.contains(data[0]))
122     {
123         verifier.insertElementAt(data[0],0);
124         ipVector.insertElementAt(data[2],0);
125         portVector.insertElementAt(data[3],0);
126         imageVector.insertElementAt(data[4],0);
127         boolean bin=false;
128         try
129         {
130             bin=phi.getInterfaceKitObject().getInputState(0);
131         }
132         catch(Exception e){System.err.println("NEW ERROR");}
133         boolVector.insertElementAt(Boolean.valueOf(bin),0);
134         JLabel cond;
135         if(bin)
136         {
137             cond=new JLabel("Started");
138             tab.setForegroundAt(0,Color.BLACK);
139         }
140         else
141         {
142             cond=new JLabel("Paused");
143             tab.setForegroundAt(0,Color.RED);
144         }
145         condVector.insertElementAt(cond,0);
146         vpane=new JPanel();
147         JPanel help=new JPanel();
148         vpane.setLayout(new BorderLayout());
149         JPanel fPane=new JPanel();
150         JLabel trans=new JLabel("Transmission State: ");
151         fPane.add(trans);
152         fPane.add(cond);
153         vpane.add(fPane,BorderLayout.NORTH);
154         JPanel foto=new JPanel();
155         JLabel labelFoto=new JLabel();
156         try
157         {
158             File file=new File(data[4]);
159             BufferedImage bi=ImageIO.read(file);
160             Image
img=bi.getScaledInstance(350,200,BufferedImage.SCALE_DEFAULT);
161             ImageIcon icon=new ImageIcon(img);
162             labelFoto.setIcon(icon);
163             //System.out.println(file.getCanonicalPath());
164         }
165         catch(Exception e)
166         {
167             labelFoto.setText("<html><center><br><br><br><br><i><b><font size='5'>" +
168                 "NO PHOTO AVAILABLE</font></b></i></html>");
169         }
170         foto.add(labelFoto);
171         vpane.add(foto,BorderLayout.CENTER);
172         JLabel title=new JLabel("Properties of "+data[1]+":");
173         JLabel tag=new JLabel("TAG:");
174         JLabel tagValue=new JLabel(data[0]);
175         JLabel ip=new JLabel("IP:");

```

```

176         JLabel ipValue=new JLabel(data[2]);
177         JLabel port=new JLabel("PORT:");
178         JLabel portValue=new JLabel(data[3]);
179         help.add(title);
180         help.add(tag);help.add(tagValue);
181         help.add(ip);help.add(ipValue);
182         help.add(port);help.add(portValue);
183         vpane.add(help, BorderLayout.SOUTH);
184         tab.insertTab(data[1],null, vpane,null,0);
185         tab.setSelectedIndex(0);
186         //my transmission's state
187         try
188         {
189             int index=verifier.indexOf(data[0]);
190             dest=new SessionAddress(InetAddress.getByName(data[2]),
191                                     Integer.parseInt(data[3]));
192             if(bin)
193                 rtp.addTarget(dest);
194         }
195         catch(IOException ioe){}
196         catch(NotRealizedError nre){}
197         catch(SecurityException se){}
198         catch(InvalidSessionAddressException isae){}
199         catch(NumberFormatException nfe){}
200
201         //what i'm receiving from this friend
202
203         //open the appropriate led
204         checkAndOpenTheLed();
205     }
206 }
207 public void setVisibleIndex(int rotorValue)
208 {
209     int allTabs=tab.getTabCount();
210     int rotorIndex=(int)1000/allTabs+1;
211     if(rotorValue<=rotorIndex)
212     {
213         tab.setSelectedIndex(0);
214     }
215     else if(rotorValue>rotorIndex)
216     {
217         tab.setSelectedIndex((int)rotorValue/rotorIndex);
218     }
219 }
220 }
221 public void checkAndOpenTheLed()
222 {
223     int fCount=0;
224     int tCount=0;
225     for(int i=0;i<boolVector.size();i++)
226     {
227         Boolean bool=boolVector.get(i);
228         if(bool.booleanValue())
229         {
230             tCount=tCount+1;
231         }
232         else
233         {
234             fCount=fCount+1;
235         }
236     }

```



```

237         if(tCount<boolVector.size())
238         {
239             try
240             {
241                 for(int i=0;i<8;i++)
242                 {
243
244                     phi.getInterfaceKitObject().setOutputState(i,false);
245                     phi.getInterfaceKitObject().setOutputState(0,true);
246                 }
247             } catch(Exception e){}
248         }
249         if(tCount==boolVector.size())
250         {
251             try
252             {
253                 for(int i=0;i<8;i++)
254                 {
255
256                     phi.getInterfaceKitObject().setOutputState(i,false);
257                     phi.getInterfaceKitObject().setOutputState(6,true);
258                 }
259             } catch(Exception e){}
260         }
261         if(fCount==boolVector.size())
262         {
263             try
264             {
265                 for(int i=0;i<8;i++)
266                 {
267
268                     phi.getInterfaceKitObject().setOutputState(i,false);
269                     phi.getInterfaceKitObject().setOutputState(7,true);
270                 }
271             } catch(Exception e){}
272         }
273     }
274     public void pauseAll(boolean bool)
275     {
276         System.out.println(bool);
277         if(bool)
278         {
279             for(int i=0;i<boolVector.size();i++)
280             {
281                 if(!boolVector.get(i).booleanValue())
282                 {
283                     try
284                     {
285                         rtp.addTarget(new SessionAddress(
286                             InetAddress.getByAddress(ipVector.get(i)),
287                             Integer.parseInt(portVector.get(i))));
288
289                         boolVector.setElementAt(Boolean.valueOf(true),i);
290                         condVector.get(i).setText("Started");
291                         tab.setForegroundAt(i,Color.BLACK);
292                     } catch(Exception e){}
293                 }

```

```

294     }
295     }
296     else
297     {
298         for(int i=0;i<boolVector.size();i++)
299         {
300             if(boolVector.get(i).booleanValue())
301             {
302                 try
303                 {
304                     rtp.removeTarget(new SessionAddress(
305                         InetAddress.getByName(ipVector.get(i)),
306                         Integer.parseInt(portVector.get(i))
307                             , " "));
308
309                     boolVector.setElementAt(Boolean.valueOf(false),i);
310                     condVector.get(i).setText("Paused");
311                     tab.setForegroundAt(i,Color.RED);
312                 }
313                 catch(Exception e){}
314             }
315         }
316         checkAndOpenTheLed();
317     }
318     public void startOrPause()
319     {
320         try
321         {
322             if(boolVector.get(tab.getSelectedIndex()).booleanValue())
323             {
324                 rtp.removeTarget(new SessionAddress(
325
326                     InetAddress.getByName(ipVector.get(tab.getSelectedIndex())),
327                     Integer.parseInt(portVector.get(tab.getSelectedIndex()))
328                         , " "));
329                 boolVector.setElementAt(Boolean.valueOf(false),tab.getSelectedIndex());
330                 condVector.get(tab.getSelectedIndex()).setText("Paused");
331                 tab.setForegroundAt(tab.getSelectedIndex(),Color.RED);
332                 //xwris thn ektypwsh emfanizetai to deytero mysthrio
333                 System.out.println(tab.getSelectedIndex()+" "+
334                     boolVector.get(tab.getSelectedIndex()).booleanValue()+" "+
335                     ipVector.get(tab.getSelectedIndex())+" "+
336                     portVector.get(tab.getSelectedIndex()));
337             }
338             else
339             {
340                 rtp.addTarget(new SessionAddress(
341
342                     InetAddress.getByName(ipVector.get(tab.getSelectedIndex())),
343                     Integer.parseInt(portVector.get(tab.getSelectedIndex()))));
344                 boolVector.setElementAt(Boolean.valueOf(true),tab.getSelectedIndex());
345                 condVector.get(tab.getSelectedIndex()).setText("Started");

```

```

344 tab.setForegroundAt(tab.getSelectedIndex(),Color.BLACK);
345     }
346     checkAndOpenTheLed();
347     }
348     catch(Exception e){}
349 }
350 public void updateFromSensor()
351 {
352     try
353     {
354         pane.remove(inf);
355         //to remove the inf totally
356         setVisible(false);
357         setVisible(true);
358         tab=new JTabbedPane(JTabbedPane.BOTTOM);
359         MouseListener[] mla=tab.getMouseListeners();
360         for(int i=0;i<mla.length;i++)
361         {
362             tab.removeMouseListener(mla[i]);
363         }
364         paneTab=new JPanel();
365
366         tab.addTab(System.getProperty("user.name"),paneTab);
367         pane.addTab();
368         phr=new PhidgetHandler('r');
369         //prepare the camera to send
370         VideoFormat vf=new VideoFormat(VideoFormat.JPEG RTP);
371         Format[] f=new Format[]{vf};
372         dsIn=CaptureUtil.getCaptureDS(vf,null);
373         ftd=new FileTypeDescriptor(FileTypeDescriptor.RAW RTP);
374         pm=new ProcessorModel(dsIn, f, ftd);
375         pr=javax.media.Manager.createRealizedProcessor(pm);
376         dsOut=pr.getDataOutput();
377         pbds=(PushBufferDataSource)dsOut;
378         pbs=pbds.getStreams();
379         rtp=RTPManager.newInstance();
380         local=new SessionAddress(InetAddress.getLocalHost(),1237);
381         rtp.initialize(local);
382         SessionAddress locDest=new
383 SessionAddress(InetAddress.getLocalHost(),
384                 1234);
385         rtp.addTarget(locDest);
386         ss=rtp.createSendStream(dsOut,0);
387         ss.start();
388         pr.start();
389         //what i'm sending to
390         paneTab.setLayout(new BorderLayout(5,5));
391         paneTab.add(new JLabel("<html><center><i><b><font
size=\"3\">\" +
392                 \"I transmit
this:</font></b></i></html>\"),BorderLayout.NORTH);
393         MediaLocator ml=new
394 MediaLocator("rtp://127.0.0.1:1234/video");
395         pl=javax.media.Manager.createRealizedPlayer(ml);
396         pl.addControllerListener(new ControllerAdapter()
397         {
398             @Override
399             public void prefetchComplete(PrefetchCompleteEvent evt)
400             {
401                 Component video=pl.getVisualComponent();

```

```

400         paneTab.add(video, BorderLayout.CENTER);
401         pl.start();
402     }
403     });
404     pl.prefetch();
405     setSize(400,350);
406     validate();
407 }
408 catch(Exception e){System.err.println(e.getMessage());}
409 }
410 public void moveLeft(int pix)
411 {
412     int curX=(int)getLocation().getX();
413     int curY=(int)getLocation().getY();
414     setLocation(curX-pix,curY);
415 }
416 public void moveRight(int pix)
417 {
418     int curX=(int)getLocation().getX();
419     int curY=(int)getLocation().getY();
420     setLocation(curX+pix,curY);
421 }
422 public void moveUp(int pix)
423 {
424     int curX=(int)getLocation().getX();
425     int curY=(int)getLocation().getY();
426     setLocation(curX,curY-pix);
427 }
428 public void moveDown(int pix)
429 {
430     int curX=(int)getLocation().getX();
431     int curY=(int)getLocation().getY();
432     setLocation(curX,curY+pix);
433 }
434 public void returnToDefault()
435 {
436     setLocation(100,100);
437 }
438 public void closeCamera()
439 {
440     try
441     {
442         pl.stop();
443         pl.close();
444         pr.stop();pr.close();
445         ss.stop();
446         ss.close();
447     }
448     catch(IOException ioe){}
449 }
450 public static void main(String[] args)
451 {
452     me=new AppHandler();
453 }
454 public static AppHandler getAppHandlerObject(){
455     return me;
456 }
457 public JTabbedPane getTabObject(){
458     return tab;
459 }
460 }

```

Αμέσως μετά έχουμε την κλάση PhidgetHandler η οποία επικοινωνεί με το υλικό οπότε όπως είναι αναμενόμενο ο δομητής της κλάσης αυτής προετοιμάζει το υλικό για χρήση (γραμμές 33-77,79-163) αλλά και την εφαρμογή ώστε να δεχτεί είσοδο από αυτό και ανάλογα από ποια συσκευή έρχονται τα δεδομένα εκτελεί κάποιες από τις παραπάνω μεθόδους. Εν συνεχεία ακολουθούν δυο συναρτήσεις, closeRfid()(γραμμές 166-175) και closeInterface()(γραμμές 176-187), που αναλαμβάνουν να κλείσουν τη σύνδεση με το υλικό και να τερματίσουν τα αντικείμενα που δέσμευσαν.

Τέλος, υπάρχουν δυο ακόμα μέθοδοι οι οποίες δίνουν πρόσβαση σε πεδία της κλάσης. Οι μέθοδοι αυτές είναι η getRFIDReaderObject (γραμμές 189-191) η οποία επιστρέφει ένα αντικείμενο τύπου RFIDReader και η getInterfaceKitObject (γραμμές 192-194) η οποία επιστρέφει ένα αντικείμενο τύπου InterfaceKit.

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package TUI_App_4;
7
8  import com.phidgets.*;
9  import com.phidgets.event.*;
10
11 /**
12  *
13  * @author AUTOEXE.INF
14  */
15 class PhidgetHandler
16 {
17     private InterfaceKitPhidget ikp;
18     private RFIDPhidget rfid;
19     private boolean firstTime=true;
20     private int SensorValue=0;//0 position
21     private int RotorValue=0;//1 position
22     private int SliderValue=0;//2 position
23     private int TouchValue=999;//3 position
24     private int StickSensorLRValue=450,StickSensorUDValue=450;//4,5
position
25     private int WeightSensorValue=0;//6 position
26     private String Tag=null;
27     private String[] data;//0-tag,1-name,3-ip,4-port
28
29     public PhidgetHandler(char c)//r for rfid,i for interface kit
30     {
31         if(c=='r')
32         {
33             try
34             {
35                 rfid=new RFIDPhidget();

```

```

36         rfid.open(33830);
37         rfid.waitForAttachment();
38         rfid.setAntennaOn(true);
39         rfid.setLEDOn(true);
40         rfid.addTagGainListener(new TagGainListener()
41         {
42             public void tagGained(TagGainEvent evt)
43             {
44                 Tag=evt.getValue();
45                 try
46                 {
47                     rfid.setLEDOn(false);
48                     //check a vector if exists:
49                     //leave it ? store it and check the base
50                     DataBaseHandler dbh=new
DataBaseHandler(Tag);
51                     data=dbh.searchDB();
52                     Thread thr=new Thread(new Runnable()
53                     {
54                         public void run()
55                         {
56
57                             AppHandler.getAppHandlerObject().addFriend(data);
58                             });
59                             thr.start();
60                         }
61                     catch(PhidgetException pe){}
62                 }
63             });
64         rfid.addTagLossListener(new TagLossListener()
65         {
66             public void tagLost(TagLossEvent evt)
67             {
68                 try
69                 {
70                     rfid.setLEDOn(true);
71                 }
72                 catch(PhidgetException pe){}
73             }
74         });
75     }
76     catch(PhidgetException pe){}
77 }
78 else if(c=='i')
79 {
80     try
81     {
82         ikp=new InterfaceKitPhidget();
83         ikp.open(36911);
84         ikp.waitForAttachment();
85         ikp.addInputChangeListener(new InputChangeListener()
86         {
87             public void inputChanged(InputChangeEvent evt)
88             {
89
90                 AppHandler.getAppHandlerObject().pauseAll(evt.getState());
91                 }
92             });
93         ikp.addSensorChangeListener(new SensorChangeListener()
94         {

```

```

94         public void sensorChanged(SensorChangeEvent evt)
95         {
96             if(firstTime)
97             {
98                 try
99                 {
100                     SensorValue=ikp.getSensorValue(0);
101                     if(SensorValue>300)
102                     {
103                         firstTime=false;
104
105                     }
106                 }
107                 catch(PhidgetException pe){}
108             }
109             try
110             {
111                 WeightSensorValue=ikp.getSensorValue(6);
112                 TouchValue=ikp.getSensorValue(3);
113
114                 if((TouchValue<300)&&(WeightSensorValue>200))
115                 {
116                     try
117                     {
118                         AppHandler.getAppHandlerObject().myExit('p');
119                     }
120                     catch(Exception
121                     e){System.err.println(e.getMessage());}
122                     //finally
123                     {
124                         System.exit(0);
125                     }
126                     RotorValue=ikp.getSensorValue(1);
127
128                     if(AppHandler.getAppHandlerObject().getTabObject()!=null)
129                     {
130                         AppHandler.getAppHandlerObject().setVisibleIndex(RotorValue);
131                     }
132                     SliderValue=ikp.getSensorValue(2);
133                     if(TouchValue<300)
134                     {
135                         AppHandler.getAppHandlerObject().startOrPause();
136                     }
137                     StickSensorLRValue=ikp.getSensorValue(4);
138                     if(StickSensorLRValue<400)
139                     {
140                         AppHandler.getAppHandlerObject().moveRight(Math.abs(400-
141                         StickSensorLRValue));
142                     }
143                     else if(StickSensorLRValue>500)
144                     {
145                         AppHandler.getAppHandlerObject().moveLeft(Math.abs(StickSensorLRValue-
146                         500));
147                     }
148                 }
149             }
150         }

```

```

144         StickSensorUDValue=ikp.getSensorValue(5);
145         if(StickSensorUDValue<450)
146         {
147
AppHandler.getAppHandlerObject().moveDown(Math.abs(450-
StickSensorUDValue));
148         }
149         else if(StickSensorUDValue>550)
150         {
151
AppHandler.getAppHandlerObject().moveUp(Math.abs(StickSensorUDValue-550));
152         }
153         if(WeightSensorValue>200)
154         {
155
AppHandler.getAppHandlerObject().returnToDefault();
156         }
157         }
158         catch(PhidgetException pe){}
159     }
160     });
161     }
162     catch(PhidgetException pe){}
163 }
164 }
165
166 public void closeRfid()
167 {
168     try
169     {
170         rfid.setLEDOn(false);
171         rfid.close();
172     }
173     catch(PhidgetException pe){}
174     catch(NullPointerException npe){}
175 }
176 public void closeInterface()
177 {
178     try
179     {
180         for(int i=0;i<8;i++)
181         {
182             ikp.setOutputState(i,false);
183         }
184         ikp.close();
185     }
186     catch(PhidgetException pe){}
187 }
188
189 public RFIDPhidget getRFIDReaderObject(){
190     return rfid;
191 }
192 public InterfaceKitPhidget getInterfaceKitObject(){
193     return ikp;
194 }
195 }

```

Τελευταία κλάση για την εφαρμογή αυτή είναι η DatabaseHandler και όπως προαναφέραμε αυτή συνδέει την εφαρμογή με τη βάση δεδομένων. Αναλυτικότερα η



σύνδεση με τη βάση δεδομένων υλοποιείται στον δομητή της κλάσης αυτής (γραμμές 20-21) και στη συνέχεια η συνάρτηση που ακολουθεί, searchDB()(γραμμές 27-43), όπως δηλώνει και το όνομα της ψάχνει τη βάση δεδομένων να βρει τις τιμές των στηλών που ταιριάζουν στην τιμή του tag που είχε σαν παράμετρο ο δομητής της κλάσης αυτής.

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6 package TUI_App_4;
7
8 import java.sql.*;
9
10 /**
11  *
12  * @author AUTOEXE.INF
13  */
14 class DataBaseHandler
15 {
16     private Connection con;
17     private String TheTag;
18
19     public DataBaseHandler(String tag)
20     {
21         try
22         {
23             TheTag=tag;
24             Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
25             con=DriverManager.getConnection("jdbc:odbc:ptyxiakh3","","");
26         }
27         catch(ClassNotFoundException cnfe){}
28         catch(SQLException sqle){}
29     }
30
31     public String[] searchDB()
32     {
33         try
34         {
35             String query="SELECT friendsData.* "+
36                 "FROM friendsData "+
37                 "WHERE (((friendsData.TAG)='"+TheTag+"')";
38             PreparedStatement ps=con.prepareStatement(query);
39             ResultSet rs=ps.executeQuery();
40             rs.next();
41             String[] arr=new String[]{rs.getString(1),rs.getString(2),
42                 rs.getString(3),rs.getString(4),
43                 rs.getString(5)};
44             return arr;
45         }
46         catch(SQLException sqle){return new
47             String[]{"ERROR","ERROR","ERROR","ERROR"};
48     }
49 }
```

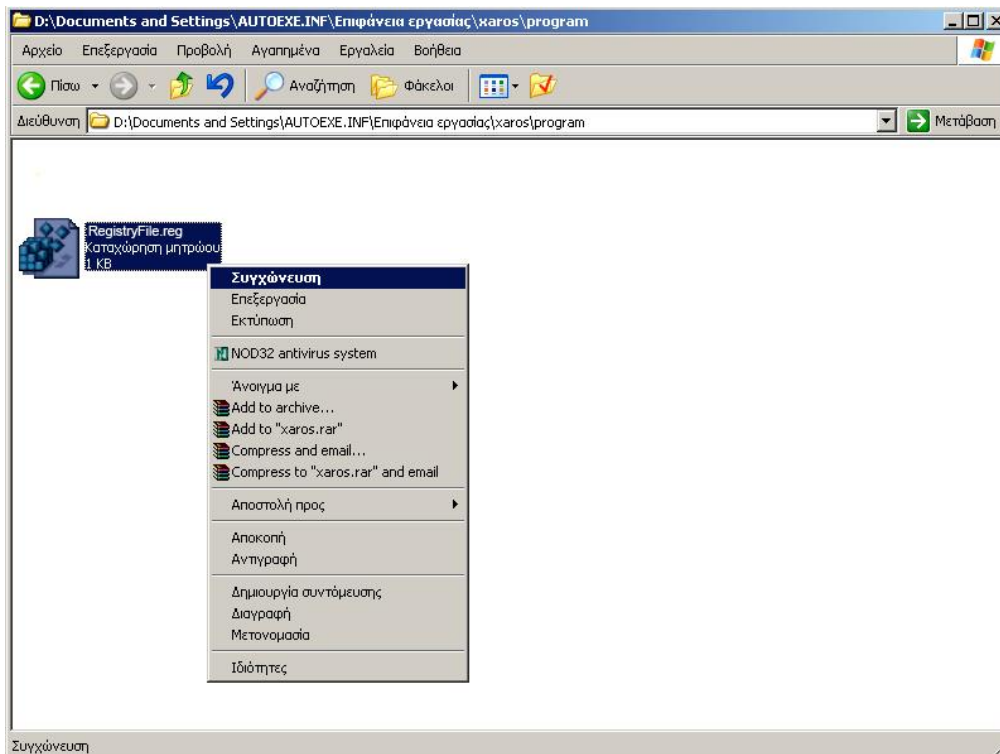
## **ΕΦΑΡΜΟΓΗ 5 - ΑΣΦΑΛΕΙΑ ΛΕΙΤΟΥΡΓΙΑΣ Η/Υ ΑΠΟ ΕΝΑ ΑΤΟΜΟ ΧΩΡΙΣ ΔΙΚΑΙΩΜΑ ΠΡΟΣΒΑΣΗΣ ΚΑΙ ΚΑΤΑΓΡΑΦΗ ΤΟΥ ΕΙΣΒΟΛΕΑ ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΕΦΑΡΜΟΓΗ ΚΑΙ ΤΟ ΥΛΙΚΟ**

Η εφαρμογή αυτή μοιάζει πολύ με την τρίτη εφαρμογή που αναπτύξαμε (δείτε το σχετικό κεφάλαιο). Η ειδοποιός διαφορά τους είναι πως η προηγούμενη προϋποθέτει πως ο υπολογιστής είναι ανοιχτός από πριν και καταγράφει όποιον μπει στο δωμάτιο και δεν είναι ο εξουσιοδοτημένος χρήστης ενώ η τρέχουσα εφαρμογή θέλει τον υπολογιστή κλειστό και στη συνέχεια καταγράφει οποιονδήποτε μη εξουσιοδοτημένο χρήστη προσπαθήσει να ανοίξει τον υπολογιστή. Ας ρίξουμε μια ματιά στην ουσία της εφαρμογής αυτής.

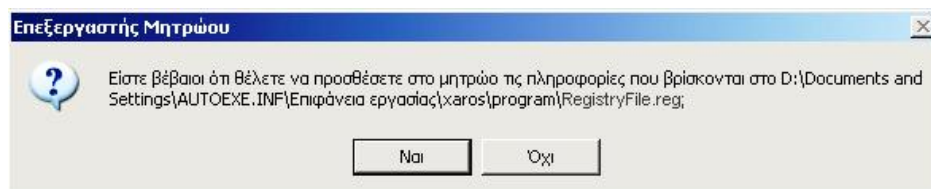
Στόχος της εφαρμογής αυτής είναι να εξασφαλίσει ο χρήστης ένα παραπάνω επίπεδο ασφαλείας πέρα από τον κωδικό πρόσβασης των windows. Αυτό αποσκοπεί στο να απαγορέψει σε έμπειρους χρήστες που έχουν υποκλέψει τον κωδικό να έχουν πρόσβαση στον υπολογιστή. Οι δυνατότητες της εφαρμογής όμως δε σταματούν εδώ. Σε περίπτωση που κάποιος κάνει μια αποτυχημένη απόπειρα η κάμερα τον καταγράφει αποθηκεύει το βίντεο με όνομα αρχείου την ημερομηνία που έγινε η παραβίαση.

Η εφαρμογή αυτή δεν έχει καμία ρύθμιση όπως οι προηγούμενες, αυτό δε σημαίνει όμως ότι δε χρειάζεται τόση ανάλυση όσο οι προηγούμενες εφαρμογές. Ξεκινώντας την ανάλυση θα πρέπει να αναφέρουμε το κλειδί και την τιμή του που αποθηκεύουμε στο μητρώο. Η εφαρμογή αυτή συνοδεύεται από ένα αρχείο τύπου «καταχώρηση μητρώου». Επιλέγοντας να ανοίξουμε το συγκεκριμένο αρχείο με κάποιον επεξεργαστή κειμένου βλέπουμε τέσσερις γραμμές. Η πρώτη και η δεύτερη( εννοούμε την κενή γραμμή) γραμμή είναι σταθερές σε κάθε αρχείο τέτοιου τύπου. Συνεχίζοντας την εξέταση των περιεχομένων στην τρίτη γραμμή βρίσκεται η θέση που θα αποθηκεύσουμε το κλειδί στο μητρώο. Στη συγκεκριμένη θέση θα βρούμε κλειδιά για όλα τα προγράμματα που ξεκινούν μόλις ανοίγουμε τον υπολογιστή. Τέλος ακολουθεί η γραμμή που είναι ο συνδυασμός κλειδί-τιμή. Κλειδί ονομάζεται το αριστερό μέρος από το ίσον(=) και το δεξί είναι η τιμή του, όπου όπως παρατηρούμε η τιμή είναι η διεύθυνση του αρχείου στο σκληρό δίσκο. Εν συνεχεία κλείνουμε τον επεξεργαστή κειμένου και κάνουμε δεξί κλικ πάνω στο αρχείο και επιλέγουμε «Συγχώνευση»(Εικόνα 43). Πατάμε «Ναι» στο παράθυρο επιβεβαίωσης(Εικόνα 44)

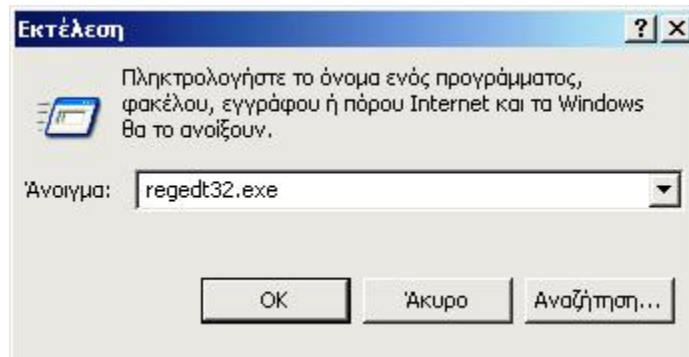
που εμφανίζεται και όλα είναι έτοιμα. Για να επιβεβαιωθούμε πως όλα είναι μια χαρά στο μητρώο πάμε στην «Έναρξη» και επιλέγουμε «Εκτέλεση». Στο πεδίο κειμένου γράφουμε «regedt32.exe»(Εικόνα 45) και ανοίγει το πρόγραμμα επεξεργασίας του μητρώου. Στο δέντρο που υπάρχει αριστερά βρίσκουμε τη διαδρομή που καταχωρήσαμε στην τρίτη γραμμή και επιλέγουμε τον τελικό της κόμβο. Μόλις τον επιλέξουμε στα δεξιά εμφανίζονται τα περιεχόμενά του και μέσα σ'αυτά θα πρέπει να υπάρχει και το κλειδί της δικής μας εφαρμογής(Εικόνα 46). Θα πρέπει να αναφέρουμε πως η εφαρμογή αυτή δεν εγκαθιστά αυτόματα τίποτα στον υπολογιστή και η εκχώρηση στο μητρώο σκόπιμα έχει αφαιρεθεί από τις δυνατότητες της εφαρμογής. Οποιαδήποτε στιγμή θέλει ο χρήστης καταργεί το κλειδί και έτσι η εφαρμογή παύει να εκτελείται.



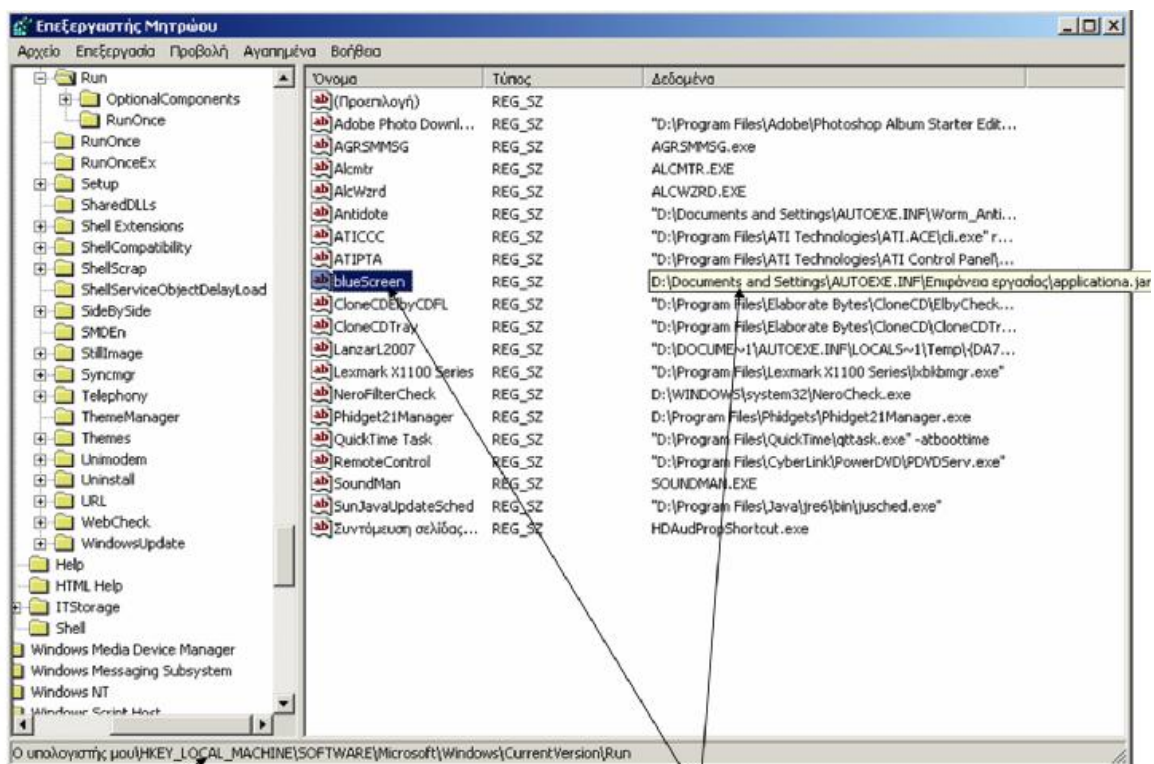
Εικόνα 43



Εικόνα 44



Εικόνα 45



Η θέση που βρίσκεται το κλειδί στο registry

Το όνομα του κλειδιού και η διαδρομή του αρχείου

Εικόνα 46

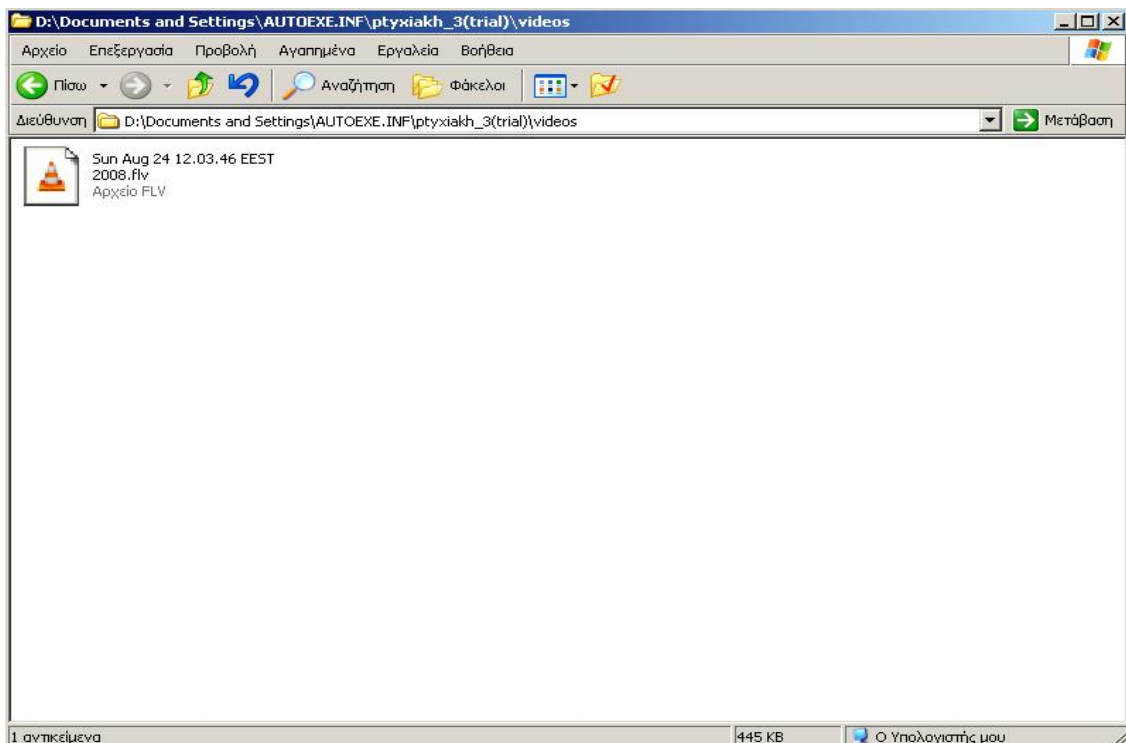
Αφού εξασφαλίσουμε ότι θα «τρέχει» η εφαρμογή μας με το άνοιγμα του υπολογιστή θα πρέπει να απαγορέψει την αλληλεπίδραση του χρήστη με το λειτουργικό σύστημα όσο το πρόγραμμα είναι σε λειτουργία. Έτσι το πρόγραμμα ελέγχει πόσες οθόνες έχει ο υπολογιστής και εμφανίζει παράθυρα(με χρήση της κλάσης Window) με μπλε χρώμα. Κάποιοι χρήστες ίσως εγκαταλείψουν από τώρα την προσπάθεια πιστεύοντας πως το σύστημα «κρέμασε». Θα πρέπει επίσης να εξασφαλίσουμε πως το πρόγραμμα σταματάει μόνο με έναν τρόπο που τον ξέρει αποκλειστικά ο εξουσιοδοτημένος χρήστης της εφαρμογής αυτής. Για τον λόγο αυτόν

έχουν αφαιρεθεί οι δυνατότητες κλεισίματος μέσω της διαχείρισης εργασιών είτε μέσω της συντόμευσης κλεισίματος παραθύρου(Alt+F4). Επιπλέον για να μην έχει κάποιος όλο το χρόνο με το μέρος του η εφαρμογή δίνει συγκεκριμένο χρόνο στο χρήστη για να επιβεβαιώσει την ταυτότητά του πριν το πρόγραμμα εκτελέσει τις απαραίτητες λειτουργίες, σε περίπτωση μη ή λανθασμένης ταυτοποίησης του κωδικού πρόσβασης. Τέλος αν όλα πάνε καλά εμφανίζεται το μήνυμα καλωσορίσματος προς τον χρήστη(Εικόνα 47) αλλιώς αρχίζει να καταγράφει η κάμερα τον εισβολέα για 5 δευτερόλεπτα, αποθηκεύει το βίντεο στο φάκελο videos με όνομα αρχείου την ημερομηνία και την ώρα που έγινε η εγγραφή(Εικόνα 48). Αυτό γίνεται για να δει μετά ο εξουσιοδοτημένος χρήστης τις αποτυχημένες προσπάθειες για την παραβίαση του συστήματος καθώς επίσης και από ποιον έγινε.



Ένα παράθυρο καλωσορίσματος που καλωσορίζει το χρήστη με το όνομα που έχει ο ίδιος ονομαστεί στον υπολογιστή

Εικόνα 47



Εικόνα 48

## **ΠΕΡΙΗΓΗΣΗ ΣΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ**

Η εφαρμογή αυτή αποτελείται αποκλειστικά από μια κλάση, τη PCAccess. Ο δομητής της κλάσης ελέγχει πόσες οθόνες χρησιμοποιεί ο χρήστης(γραμμές 36-37). Στη συνέχεια υλοποιούμε έναν Vector με Window μεγέθους ίδιου με τον αριθμό των οθονών που έχει ο υπολογιστής(γραμμές 39-45). Μέσα στην επανάληψη δίνουμε στα παράθυρα μπλε χρώμα (γραμμή 42)και τα μεγιστοποιούμε(γραμμή 44). Ο javax.swing.Timer (γραμμές 49-59) που ακολουθεί αναλαμβάνει να φέρνει τα παράθυρα στην επιφάνεια ανά 10 milliseconds(ο χρόνος αυτός είναι πολύ λίγος για να καταφέρει ο χρήστης να κλείσει το παράθυρο ή να βάλει άλλο παράθυρο στην επιφάνεια). Μετά ετοιμάζεται ο RFIDReader(γραμμές 65-70) ώστε να μπορεί να δεχτεί είσοδο και στον ακροατή συμβάντων ρυθμίζουμε να σβήνει το led(γραμμές 71-81) όταν περνάει από πάνω κάποιο tag και όταν το tag απομακρύνεται ελέγχουμε αν είναι το σωστό(γραμμές 96-126). Αν ναι, τότε σταματάει ο προηγούμενος Timer(γραμμή 100), κλείνουν τα μπλε παράθυρα(γραμμές 102-105) και εμφανίζεται το μήνυμα καλωσορίσματος(γραμμές 111-113). Αν όμως είναι λάθος τότε εκτελείται η μέθοδος startRecording()(γραμμή 122). Λίγο πριν κλείσουμε με τον δομητή να αναφέρουμε τον τελευταίο Timer ο οποίος χρονομετρά 5 δευτερόλεπτα(γραμμές 131-145). Μόλις περάσει ο χρόνος αυτός εκτελείται πάλι η startRecording()(αυτό γίνεται για να μην έχει ο χρήστης απεριόριστο χρόνο για να παρακάμψει το πρόγραμμα).

Η μέθοδος startRecording()(γραμμές 148-201)χρησιμοποιείται για να καταγράψει η κάμερα τον εισβολέα, στο block try-catch(γραμμές 164-170) δημιουργεί τον φάκελο που χρειάζεται και αποθηκεύει το βίντεο με όνομα αρχείου την τρέχουσα ημέρα και ώρα(γραμμή 171). Πριν το πέρας της μεθόδου αυτής υπάρχει ένας ακόμα Timer(γραμμές 177-198) ίδιος με τους προηγούμενους. Αυτός επαναλαμβάνεται ανά 7 δευτερόλεπτα, στην ουσία όμως τρέχει μόλις μια φορά. Όταν του θέσουμε επανάληψη ανά 7 δευτερόλεπτα καταφέρνουμε να καταγραφεί ο εισβολέας για 5 δευτερόλεπτα.

Πρέπει να σημειωθεί πως για την καταγραφή από την κάμερα, κατά τη χρήση της CaptureUtil.getCaptureDS(...)(γραμμή 154), έχουν χρησιμοποιηθεί εξωτερικές κλάσεις. Αυτές είναι οι εξής:CaptureUtil, MonitorCDS, MonitorStream καθώς και κάποιες από τις μεθόδους που εμπεριέχονται στις κλάσεις αυτές. Οι κλάσεις αυτές προέρχονται από τη Sun Microsystems και εμπεριέχονται στο πακέτο με όνομα jmfsample.

Τέλος, ακολουθεί η απαραίτητη συνάρτηση για κάθε εκτέλεσιμη κλάση, η main()(γραμμές 202-206), η οποία απλά καλεί τον δομητή της κλάσης αυτής.

```

1 package TUI_App_5;
2
3 //java
4 import java.io.*;
5 import java.awt.*;
6 import java.awt.event.*;
7 import java.util.*;
8 //javax
9 import javax.swing.*;
10 import javax.swing.event.*;
11 import javax.media.*;
12 import javax.media.rtp.*;
13 import javax.media.rtp.event.*;
14 import javax.media.rtp.rtcp.*;
15 import javax.media.format.*;
16 import javax.media.protocol.*;
17 //phidgets
18 import com.phidgets.*;
19 import com.phidgets.event.*;
20
21 public class PcAccess {
22     private Vector<Window> wv;
23     private int counter=0,phidgetCounter=0;
24     private RFIDPhidget rfid;
25     private boolean tagPassed=false;
26     private javax.swing.Timer pt,timer;
27     private static DataSource processorOut;
28     private static Processor pr;
29     private static DataSink ds;
30     private GraphicsDevice[] sd;
31
32     public PcAccess()
33     {
34         //blue Screen
35         ////////////////
36         GraphicsEnvironment
ge=GraphicsEnvironment.getLocalGraphicsEnvironment();
37         sd=ge.getScreenDevices();
38         wv=new Vector<Window>(sd.length);
39         for(int i=0;i<sd.length;i++)
40         {
41             Window w=new Window(null);
42             w.setBackground(Color.BLUE);
43             wv.add(w);
44             sd[i].setFullScreenWindow(wv.get(i));
45         }
46         wv.trimToSize();
47         timer=new javax.swing.Timer(10,new ActionListener()
48         {
49             @Override
50             public void actionPerformed(ActionEvent evt)
51             {
52                 for(int i=0;i<wv.size();i++)
53                 {
54                     sd[i].setFullScreenWindow(wv.get(i));
55                     wv.get(i).setVisible(true);
56                 }
57             }
58         }

```

```

58     });
59     timer.start();
60     ///////////////////////////////////
61
62     //phidgets
63     ///////////////////////////////////
64     try
65     {
66         rfid=new RFIDPhidget();
67         //gia panw apo 1 phidget 8elw thn open(serial)
68         rfid.openAny();
69         rfid.waitForAttachment();
70         rfid.setAntennaOn(true);
71         rfid.setLEDOn(true);
72         rfid.addTagGainListener(new TagGainListener()
73         {
74             public void tagGained(TagGainEvent evt)
75             {
76                 try
77                 {
78                     rfid.setLEDOn(false);
79                 }
80                 catch(PhidgetException pe){}
81             }
82         });
83         rfid.addTagLossListener(new TagLossListener()
84         {
85             public void tagLost(TagLossEvent evt)
86             {
87                 try
88                 {
89                     tagPassed=true;
90                     String value=evt.getValue();
91                     rfid.close();
92                     pt.stop();
93                     //System.out.println("Exit from passed");
94                     //System.exit(0);
95
96                     //if(value.equals("0107757c8e")|(value.equals("04165854cf"))
97                         //|(value.equals("01068de218")))//
98                     if(value.equals("3100bdf6de"))
99                     {
100                         //bges apo blue screen
101                         timer.stop();
102                         //einai swsto emfanise para8yro
103                         for(int i=0;i<wv.size();i++)
104                         {
105                             wv.get(i).dispose();
106                         }
107                         JFrame frame=new JFrame("WELCOME");
108                         frame.setBounds(100,100,100,100);
109                         frame.setResizable(false);
110
111                         JPanel pane=new JPanel();
112                         String message="Welcome "
113                             +System.getProperty("user.name");
114                         String html="<html><center><i><b><font
115                             size=|\"5|\">"+message+"</font></b></i>";
116                         pane.add(new JLabel(html));
117                         frame.setContentPane(pane);

```



```

116         frame.setVisible(true);
117         frame.pack();
118     }
119     else
120     {
121         //arxise na grafeis
122         startRecording();
123     }
124 }
125 catch(PhidgetException pe){}
126 }
127 });
128 }
129 catch(PhidgetException pe){}
130 //o timer gia 5 sec wait
131 pt=new javax.swing.Timer(1000,new ActionListener()
132 {
133     @Override
134     public void actionPerformed(ActionEvent evt)
135     {
136         phidgetCounter=phidgetCounter+1;
137         if(phidgetCounter>5)
138         {
139             phidgetCounter=0;
140             pt.stop();
141             startRecording();
142         }
143     }
144 });
145 pt.start();
146 ////////////////
147 }
148 public void startRecording()
149 {
150     try
151     {
152         VideoFormat vf=new VideoFormat(VideoFormat.MJPEG);
153         Format[] fa=new Format[]{vf};
154         DataSource capture=CaptureUtil.getCaptureDS(vf,null);
155         FileTypeDescriptor ftd=new
FileTypeDescriptor(FileTypeDescriptor.QUICKTIME);
156         ProcessorModel pm=new ProcessorModel(capture, fa, ftd);
157         pr=javax.media.Manager.createRealizedProcessor(pm);
158         processorOut=pr.getDataOutput();
159         Date date=new Date();
160         String dt=date.toString();
161         dt=dt.replace(":", ".");
162         File file=new File("");
163         String cpath=null;
164         try
165         {
166             cpath=file.getCanonicalPath();
167             cpath=cpath+"\\videos\\";
168             cpath.replace("\\", "/");
169         }
170         catch(Exception e){}
171         MediaLocator ml=new
MediaLocator("file:///"+cpath+dt+".flv");
172         ds=javax.media.Manager.createDataSink(processorOut, ml);
173         ds.open();
174         ds.start();

```

```

175         pr.start();
176         //recorder timer
177         javax.swing.Timer rt=new javax.swing.Timer(7000,new
ActionListener()
178         {
179             @Override
180             public void actionPerformed(ActionEvent evt)
181             {
182                 try
183                 {
184                     pr.stop();
185                     pr.close();
186                     ds.stop();
187                     ds.close();
188                 }
189                 catch(Exception ioe){}
190                 //try
191                 {
192                     //Runtime.getRuntime().exec("shutdown -s -t
00");
193                     System.exit(0);
194                 }
195                 //catch (IOException
ioe){System.out.println(ioe.getMessage());}
196             }
197         });
198         rt.start();
199     }
200     catch(Exception e){System.err.println(e.getMessage());}
201 }
202 public static void main(String[] args)
203 {
204     new PcAccess();
205 }
206 }
207 }

```

## ΣΥΜΠΕΡΑΣΜΑΤΑ

Τα συμπεράσματα μετά από κάθε εφαρμογή που ολοκληρωνόταν επιτυχώς ήταν ποικίλα. Μετά από κάθε εφαρμογή που ολοκληρωνόταν επιβεβαιωνόταν η αρχή πως ο προγραμματισμός μπορεί να βοηθήσει σε ποικίλους τομείς της καθημερινής ζωής του ανθρώπου. Η διαδικασία της ανάπτυξης λογισμικού για τις διαφορετικές εφαρμογές που υλοποιήθηκαν ακολούθησε μια αυξητική πορεία καθώς γνώσεις που αποκομίζονταν από τις αρχικές εφαρμογές αξιοποιούνταν στις επόμενες εφαρμογές. Αυτό το μοντέλο σταδιακής ανάπτυξης λογισμικού (incremental development model) μαζί με την αντικειμενοστρεφή προσέγγιση (object-oriented approach) που επιβάλλει η επιλογή της γλώσσας προγραμματισμού Java, καθώς και η εις βάθος εξοικείωση με αυτή την πολύ διαδεδομένη σήμερα γλώσσα προγραμματισμού είναι σημαντικά οφέλη που αποκομίσθηκαν κατά την εκπόνηση της πτυχιακής. Παρατηρώντας μεμονωμένα όμως κάθε εφαρμογή θα δούμε πως αποκομίσθηκαν διαφορετικές γνώσεις.

Με το πέρας της πρώτης εφαρμογής οι γνώσεις μας διευρύνθηκαν ως προς την επικοινωνία μιας εφαρμογής με μια βάση δεδομένων για το πως αποθηκεύουμε και διαβάζουμε δεδομένα από έναν πίνακα σε μια ΒΔ. Πέρα όμως από αυτό αποκτήθηκαν γνώσεις για το πως μπορούμε να χρησιμοποιήσουμε σε μια εφαρμογή το hardware και τα APIs που παρέχονται από άλλες εταιρείες.

Με την ολοκλήρωση της δεύτερης εφαρμογής χρησιμοποιήσαμε τις γνώσεις από την προηγούμενη εφαρμογή για την επικοινωνία της εφαρμογής μας με το hardware που τη συνοδεύει, άλλα κάναμε και ένα μικρό βήμα ως προς την κατανόηση του Java Media Framework (JMF) με το να δημιουργήσουμε μια εφαρμογή που αναπαράγει βίντεο και παρέχει διάφορες ρυθμίσεις που εφαρμόζονται πάνω στο βίντεο όπως παύση, σταμάτημα κτλ με τη χρήση απτών διεπαφών .

Στην τρίτη εφαρμογή καταφέραμε και φτιάξαμε μια εφαρμογή που επικοινωνεί με ακόμα περισσότερα κομμάτια hardware συνδυάζοντας γνώσεις από τις προηγούμενες εφαρμογές. Οι γνώσεις που αποκτήσαμε από αυτή την εφαρμογή όμως δε σταματούν εδώ. Μπήκαμε ακόμα πιο βαθιά στις δυνατότητες του JMF καταγράφοντας μέσα από την κάμερα κάποιον συγκεκριμένο χώρο αλλά και αποθηκεύοντας το βίντεο στο σκληρό δίσκο του υπολογιστή.

Χρησιμοποιώντας τεχνογνωσία που έχει προσέλθει από τις προηγούμενες εφαρμογές πάνω στο hardware που χρησιμοποιείται καθ' όλο το εύρος της εργασίας αυτής, στην προτελευταία εφαρμογή επιχειρήσαμε και καταφέραμε να εντρυφήσουμε ακόμα πιο βαθιά στο JMF και πέραν από την καταγραφή βίντεο διδαχθήκαμε το πως στέλνουμε βίντεο μέσα από ένα δίκτυο. Όχι όμως με μαζική αποστολή αλλά κάτι πιο εξειδικευμένο, αποστολή κατά επιλογή, δηλαδή μπορούμε να στείλουμε βίντεο σε επιλεγμένους χρήστες και όχι σε όλους τους χρήστες ενός υποδικτύου.

Η τελευταία εφαρμογή της εργασίας αυτής δεν μας προσέφερε κάτι καινούριο από τεχνογνωσία στο hardware ή από εμπειρία προγραμματισμού. Στόχος της εφαρμογής αυτής είναι να δούμε πως μπορούμε με τις υπάρχουσες γνώσεις που έχουμε ήδη αποκτήσει να “βελτιώσουμε” τις υπηρεσίες που προσφέρονται από τον υπολογιστή μας. Η εφαρμογή αυτή μας έμαθε περισσότερο κάποιες εξειδικευμένες λειτουργίες των Windows, όπως ο Registry Editor και οι δυνατότητες που παρέχονται μέσα από αυτόν.

Οι εφαρμογές που ήδη παρουσιάστηκαν μας έμαθαν πολλά χρήσιμα στοιχεία γύρω από τον προγραμματισμό και την ανάπτυξη εφαρμογών που συνδυάζει υλικό και λογισμικό. Ο προγραμματισμός παρέχει πολύ περισσότερες δυνατότητες και εδώ χρησιμοποιήθηκαν ένα μέρος των δυνατοτήτων αυτών. Μελετώντας τη βιβλιογραφία που ακολουθεί μπορεί κανείς να βρει παραδείγματα ακόμα πιο εξειδικευμένου προγραμματισμού.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

Cadenhead, Rogers, Lemay, Laura, (2003), Πλήρες εγχειρίδιο της Java 2, Γκιούρδας Μ.

Sun Microsystems, The Really Big Index, Online, Διαθέσιμο:  
<http://java.sun.com/docs/books/tutorial/reallybigindex.html>

Sun Microsystems, Java SE api Documentation, Online, Διαθέσιμο:  
<http://java.sun.com/docs/books/tutorial/reallybigindex.html>

Sun Microsystems, Java Media Framework (JMF) api Documentation, Online, Διαθέσιμο:  
<http://java.sun.com/javase/technologies/desktop/media/jmf/2.1.1/apidocs/>

Phidgets, Phidgets Programming Manual, Online, Διαθέσιμο:  
[http://www.phidgets.com/documentation/Programming\\_Manual.pdf](http://www.phidgets.com/documentation/Programming_Manual.pdf)

Phidgets, Getting started with Phidgets in Java, Online, Διαθέσιμο:  
[http://www.phidgets.com/documentation/Tutorials/Getting\\_Started\\_Java.pdf](http://www.phidgets.com/documentation/Tutorials/Getting_Started_Java.pdf)

E. Costanza, S.B. Shelley, J. Robinson, “Introducing audio D-touch: a tangible user interface for music composition and performance” Proc. of the 6th Int. Conference on Digital Audio Effects (DAFX-03), London, UK, September 8-11, 2003

Phidgets, Phidgets Java Documentation, Online, Διαθέσιμο:  
<http://www.phidgets.com/documentation/web/javadoc/index.html>

Albrecht Schmidt, Dominik Schmidt, Paul Holleis, Matthias Kranz, “A Display Cube as a Tangible User Interface”, University of Munich, Online, Διαθέσιμο:  
[http://www.hcilab.org/matthias/PosterForDemo\\_KranzSchmidtHolleisSchmidt\\_ADisplayCubeAsATangibleUserInterface\\_UbiComp2005.pdf](http://www.hcilab.org/matthias/PosterForDemo_KranzSchmidtHolleisSchmidt_ADisplayCubeAsATangibleUserInterface_UbiComp2005.pdf)

Saul Greenberg, “Rapid Prototyping of Physical User Interfaces”, Department of Computer Science, University of Calgary, Online, Διαθέσιμο:  
<http://www.graphicsinterface.org/proceedings/2002/greenbergInvitedGI2002.pdf>

Rafael Ballagas, Meredith Ringel, Maureen Stone<sup>1</sup>, Jan Borchers, “iStuff: A Physical User Interface Toolkit for Ubiquitous Computing Environments”, Online, Διαθέσιμο:  
<http://hci.stanford.edu/publications/2003/iStuff/istuff.pdf>

Matthias Lampe, Christian Flörkemeier, “The Smart Box Application Model”, Online, Διαθέσιμο: [http://www.vs.inf.ethz.ch/res/papers/SmartBoxModel\\_Pervasive2004.pdf](http://www.vs.inf.ethz.ch/res/papers/SmartBoxModel_Pervasive2004.pdf)

Trevor Pering, Intel Research, iCube: a Tangible Interface for Mobile Interaction, Online, Διαθέσιμο:  
<http://docs.google.com/gview?a=v&q=cache:AEIn4BEo96EJ:hci.rwth-aachen.de/tuit/tiki->

[download\\_file.php?fileId=15+iCube:+a+Tangible+Interface+for+Mobile+Interaction.+Trevor+Pering,+Intel+Research&hl=el&gl=gr&pli=1](download_file.php?fileId=15+iCube:+a+Tangible+Interface+for+Mobile+Interaction.+Trevor+Pering,+Intel+Research&hl=el&gl=gr&pli=1)

Maria CRONÉ, Erik ELIASSON, Johan MATTSSON, Paweł WIATR, Magic Bowl: a Tangible User Interface for Configuration of Interactive Environments, Online, Διαθέσιμο:

<ftp://ftp-sop.inria.fr/acacia/COOP2004/soumissions/coop2004-Crone-41.pdf>