



Πανεπιστήμιο Πειραιώς Τμήμα Ψηφιακών Συστημάτων

Διαδικτυακά και Φορητά Πληροφοριακά Συστήματα (Δικτυακές Υπηρεσίες)

9^η Διάλεξη

Δημοσθένης Κυριαζής

Δευτέρα 7 Μαΐου 2018



Σημερινή διάλεξη

- Σύνοψη προηγούμενης διάλεξης
- Intents
 - Διαδικασία: Implicit intent
- Ρυθμίσεις / προτιμήσεις
 - Εισαγωγή
 - Ιδιότητες
 - Διαδικασία
- Βάση δεδομένων
 - Εισαγωγή
 - Διαχείριση πινάκων
 - Διαχείριση βάσης δεδομένων



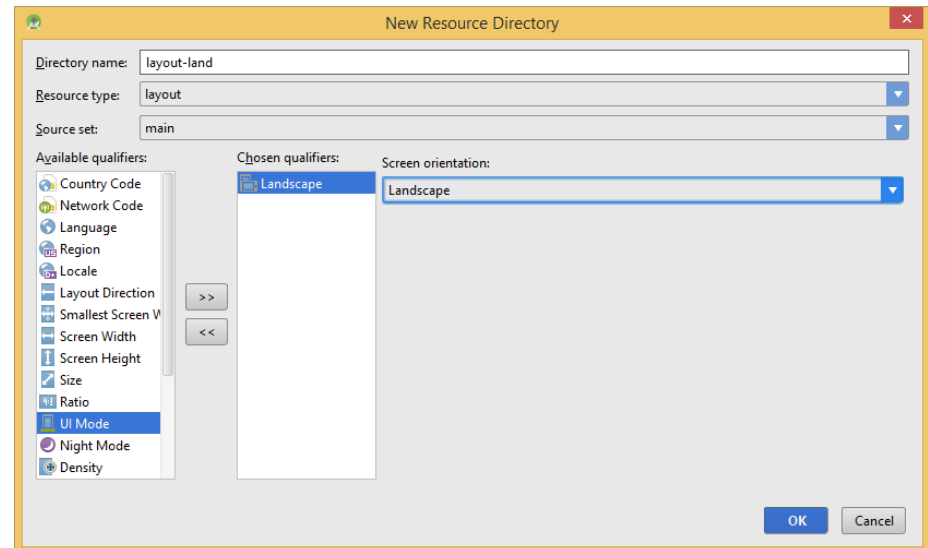
SeekBar

- Προβολή «μπάρας» για επιλογή από χρήστη
- Listener
 - OnSeekBarChange
 - `public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser)`
 - Σε κάθε αλλαγή από το χρήστη
 - `public void onStartTrackingTouch(SeekBar seekBar)`
 - Στην έναρξη των αλλαγών από το χρήστη
 - `public void onStopTrackingTouch(SeekBar seekBar)`
 - Μετά την ολοκλήρωση των αλλαγών από το χρήστη

Υποστήριξη διαφορετικών οθονών (1/2)



- Κάθε Activity διαχειρίζεται αλλαγές στον προσανατολισμό της οθόνης
- Με αλλαγή στο AndroidManifest.xml μπορεί να οριστεί «συγκεκριμένος» προσανατολισμός
 - Π.χ. `android:screenOrientation="portrait"`
- Για διαφορετικά layout ανάλογα με τον προσανατολισμό
 - Δημιουργία ενός καταλόγου `res/layout-land`
 - Δεξί κλικ στο res -> New -> Android Resource Directory
 - Resource type: layout
 - Orientation: landscape



Υποστήριξη διαφορετικών οθονών (2/2)



□ Χρήση “qualifiers”

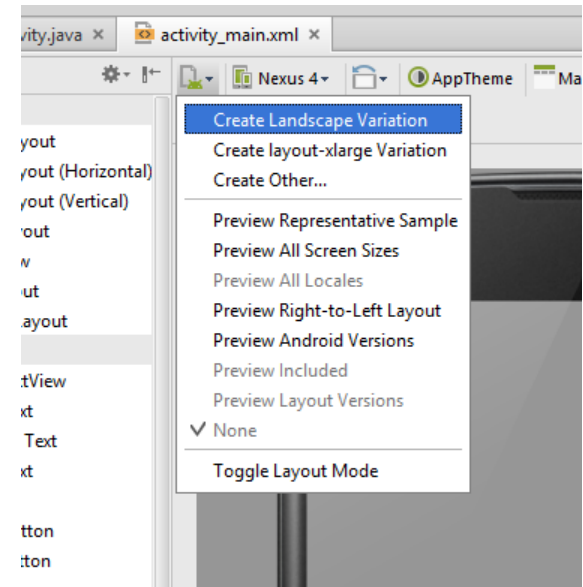
■ Landscape / portrait: land και port

□ 2 layout

- activity_main.xml
- activity_main.xml (land)

□ Αλλαγή orientation στον emulator

- Ctrl+F11: Win, Linux
- Mac: Ctrl+FN+F11





Μενού (1/3)

- title: το κείμενο της επιλογής
- icon: το εικονίδιο της επιλογής
 - Συγκεκριμένα εικονίδια στο φάκελο drawable-xhdpi
- showAsAction: καθορίζει εάν η επιλογή θα εμφανίζεται στο action bar. Λαμβάνει τιμές always, never, ifRoom
 - `app:showAsAction="never"`
- orderInCategory: καθορίζει έναν ακέραιο αριθμό για τη σειρά των επιλογών (ο μικρότερος αριθμός εμφανίζεται πρώτος)
 - `android:orderInCategory="100"`



Μενού (2/3)

- Υπάρχει το αρχικό μενού στο φάκελο res/menu
- Αναφέρεται σε ένα XML αρχείο
 - menu_main.xml
 - Περιλαμβάνει τις επιλογές του μενού
- Επιπλέον XML αρχεία μπορούν να προστεθούν στο συγκεκριμένο φάκελο για άλλες activities που επίσης έχουν μενού



Μενού (3/3)

□ menu_main.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools" tools:context=".MainActivity">
  <item android:id="@+id/action_settings" android:title="@string/action_settings"
        android:orderInCategory="100" app:showAsAction="never" />
  <item
    android:id="@+id/item1"
    android:title="@string/firstItem_mitem"
    app:showAsAction="always"></item>
  <item
    android:id="@+id/item2"
    android:title="@string/secondItem_mitem"
    app:showAsAction="always"></item>
</menu>
```




Διαχείριση events

- Δημιουργούνται όταν ο χρήστης επιλέγει κάποιο αντικείμενο του μενού
- Χρήση της μεθόδου `onOptionsItemSelected()`

- Π.χ. `switch (id) {`

- `case R.id.item1:`

- `Toast.makeText(this, "Item1 selected",
Toast.LENGTH_SHORT).show();`

- `return true;`

- `case R.id.item2:`

- `Toast.makeText(this, "Item2 selected",
Toast.LENGTH_SHORT).show();`

- `return true;`

- `default:`

- `return super.onOptionsItemSelected(item);`

- `}`



Προσθήκη Activity

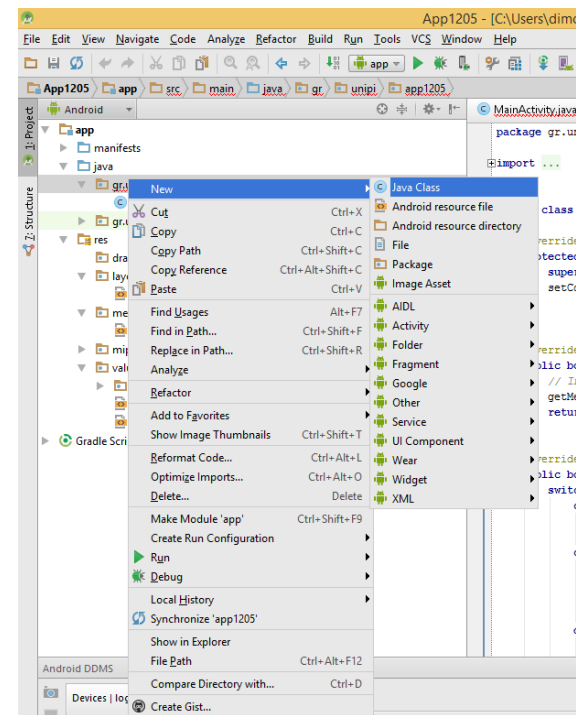
□ Δεξί κλικ στο όνομα του πακέτου -> New

■ -> Java Class

- ... extends Activity
- Προσθήκη layout
- Προσθήκη onCreate()
- Προσθήκη στο AndroidManifest!

■ -> Activity -> Empty Activity

- Προσθέτει το Android Studio το σχετικό κώδικα και το layout
- Π.χ. SecondActivity





Intents (1/2)

- Περιγράφει τη λειτουργία που θα εκτελεστεί
 - Explicit intent: όταν το intent καθορίζει το αντικείμενο στο οποίο θα εκτελεστεί η λειτουργία. Χρησιμοποιείται για τη «μεταφορά» δεδομένων μεταξύ διαφορετικών activities
 - Π.χ. `Intent intent = new Intent(this, OneActivity.class)`
 - Implicit intent: όταν το intent καθορίζει την ενέργεια που θα εκτελεστεί
 - Π.χ. `Intent viewIntent = new Intent (Intent.ACTION_VIEW, uri)`



Intents (2/2)

□ Explicit Intent

■ Constructor

- `Intent(context, class)`: Δημιουργία intent για τη συγκεκριμένη κλάση

■ Μέθοδοι

- `putExtra(name, value)`: Αποθήκευση συγκεκριμένης τιμής – Π.χ. `Intent.putExtra("title", "android lecture")`;
- `getStringExtra(name)`: Ανάκτηση τιμής
- `getIntExtra(name, default)`, `getBooleanExtra ..`



Σημερινή διάλεξη

- Σύνοψη προηγούμενης διάλεξης
- Intents
 - Διαδικασία: Implicit intent
- Ρυθμίσεις / προτιμήσεις
 - Εισαγωγή
 - Ιδιότητες
 - Διαδικασία
- Βάση δεδομένων
 - Εισαγωγή
 - Διαχείριση πινάκων
 - Διαχείριση βάσης δεδομένων



Implicit intent

□ Implicit Intent

■ Constructor

- Intent (action, uri): Δημιουργία intent για τη συγκεκριμένη ενέργεια (σταθερά) και τα συγκεκριμένα δεδομένα (uri)

■ Μέθοδοι: Αναφέρονται στην ενέργεια κατά τη δημιουργία του intent

- ACTION_VIEW: Προβολή στο browser
- ACTION_CALL: Κλήση αριθμού
- ACTION_DIAL
- ...

□ Για κάθε intent απαιτείται η εκτέλεση της σχετικής activity με χρήση της startActivity()

- Π.χ. startActivity(intent);



Διαδικασία (1/3)

- Προσθήκη στην onCreate()

 - `tvLink.setOnClickListener(this);`

 - `tvTel.setOnClickListener(this);`

- Προσθήκη στην onClick ()

 - Δημιουργία Intent και λήψη τιμών

 - `Intent consIntent = getIntent();`

 - `String myLink = consIntent.getStringExtra("link");`

 - `Uri myUri = Uri.parse(myLink);`

 - `String myTel = "tel:"+consIntent.getStringExtra("tel");`

 - `Uri callUri = Uri.parse(myTel);`



Διαδικασία (2/3)

□ Προσθήκη στην onClick () - συνέχεια

■ Έναρξη νέας Activity

```
switch (v.getId()) {  
    case R.id.textViewLink:  
        Intent viewIntent = new Intent(Intent.ACTION_VIEW,myUri);  
        startActivity(viewIntent);  
        break;  
    case R.id.textViewTel:  
        Intent callIntent = new Intent(Intent.ACTION_DIAL,callUri);  
        startActivity(callIntent);  
        break;  
}
```




Διαδικασία (3/3)

- Αλλαγή στην ItemActivity.class

`Intent callIntent = new Intent(Intent.ACTION_CALL,callUri);`

- Δεν επιτρέπεται η κλήση αριθμού

- Αλλαγή του AndroidManifest

`<uses-permission`

`android:name="android.permission.CALL_PHONE"/>`



Σημερινή διάλεξη

- Σύνοψη προηγούμενης διάλεξης
- Intents
 - Διαδικασία: Implicit intent
- Ρυθμίσεις / προτιμήσεις
 - Εισαγωγή
 - Ιδιότητες
 - Διαδικασία
- Βάση δεδομένων
 - Εισαγωγή
 - Διαχείριση πινάκων
 - Διαχείριση βάσης δεδομένων



Εισαγωγή

- Οι προτιμήσεις (preferences / settings) επιτρέπουν στους χρήστες να διαχειριστούν τη λειτουργικότητα της εφαρμογής
 - Το Android παρέχει Preference APIs για τη δημιουργία της διεπαφής των ρυθμίσεων
- Ενώ για τις περισσότερες activities χρησιμοποιούνται υπο-κλάσεις της View class, για τις ρυθμίσεις χρησιμοποιούνται υπο-κλάσεις της Preference class
 - PreferenceGroup, RingtonePreference, CheckBoxPreference, EditTextPreference, PreferenceCategory, ListPreference, κλπ



Ιδιότητες

- Όλα τα αντικείμενα (Preference items) έχουν συγκεκριμένες ιδιότητες
 - key: Το ID που χρησιμοποιείται για τη συγκεκριμένη προτίμηση
 - title: Ο τίτλος της προτίμησης
 - summary: Περιγραφικό κείμενο της προτίμησης
 - defaultValue: Αρχική τιμή
- Κάποια αντικείμενα έχουν επιπλέον ιδιότητες – π.χ. ListPreference έχει entries (ονόματα) και entryValues (τιμές)



Σημερινή διάλεξη

- Σύνοψη προηγούμενης διάλεξης
- Intents
 - Διαδικασία: Implicit intent
- Ρυθμίσεις / προτιμήσεις
 - Εισαγωγή
 - Ιδιότητες
 - Διαδικασία
- Βάση δεδομένων
 - Εισαγωγή
 - Διαχείριση πινάκων
 - Διαχείριση βάσης δεδομένων



Πριν το Android 3.0 (API 11)

- Χρήση της κλάσης PreferenceActivity
 - ... extends PreferenceActivity
- Δημιουργία preferences
- Προσθήκη activity στο AndroidManifest.xml
- Προσθήκη επιλογών στην onCreate()
 - addPreferencesFromResource (R.xml.preferences);



Διαδικασία Android 3.0 (API 11)

- Χρήση της κλάσης PreferenceFragment
- Προσθήκη String Resources
 - `<string name="pref_green_title">Green Background</string>`
 - `<string name="pref_green_summary">Change background color to green</string>`
 - `<string name="pref_usern_title">Username</string>`
 - `<string name="pref_usern_summary">Please provide your username</string>`
- Δημιουργία αρχείου με προτιμήσεις
 - Δεξί κλικ στο res -> New -> Android Resource Directory -> Επιλογή XML
 - Δεξί κλικ στο φάκελο XML, New XML resource File -> Όνομα preferences

Προσθήκη widgets στο preference fragment



□ Προσθήκη αντικειμένων στο preferences.xml

<EditTextPreference

android:title="@string/pref_usern_title"

android:summary="@string/pref_usern_summary"

android:key="prefUsername"/>

<CheckBoxPreference

android:key="pref_green"

android:title="@string/pref_green_title"

android:summary="@string/pref_green_summary"

android:defaultValue="false" />



Δημιουργία preference fragment

- Προσθήκη νέου fragment
 - Δεξί κλικ στο res -> New -> Fragment
 - (PrefFragment)
- Αλλαγή σε Preference Fragment
 - ... extends PreferenceFragment
- Διαγραφή όλων των μεθόδων εκτός της onCreate()
- Προσθήκη Preference Screen στην onCreate()
 - `addPreferencesFromResource(R.xml.preferences);`

Εισαγωγή fragment στη main activity



- Προσθήκη Layout (π.χ. με ID linearLayout1)
- Εκκίνηση στην onCreate()
 - getFragmentManager()
 - .beginTransaction()
 - .add(R.id.linearLayout1, new PrefFragment())
 - .commit();



Επιπλέον στοιχεία

- Ομαδοποίηση επιλογών στο preferences XML

```
<PreferenceCategory android:title="User Preferences">
```

- Υλοποίηση «εξαρτήσεων» στο preferences XML

```
<CheckBoxPreference
```

```
    android:key="pref_green"
```

```
    android:title="@string/pref_green_title"
```

```
    android:summary="@string/pref_green_summary"
```

```
    android:defaultValue="false" />
```

```
<CheckBoxPreference
```

```
    android:key="pref_red"
```

```
    android:dependency="pref_green"
```

```
    android:title="@string/pref_red_title"
```

```
    android:summary="@string/pref_green_summary"
```

```
    android:defaultValue="false" />
```



Χρήση προτιμήσεων χρήστη (1/2)

□ Χρήση αντικειμένου SharedPreferences στην MainActivity

■ Κλάση

- `private SharedPreferences prefs;`
- `public String testStr = "";`

■ Στην onCreate()

- `PreferenceManager.setDefaultValues(this, R.xml.preferences, false);`
- `prefs = PreferenceManager.getDefaultSharedPreferences(this);`

■ Προσθήκη onResume()

- `@Override`
- `public void onResume(){`
- `super.onResume();`
- `testStr = prefs.getString("prefUsername", "");`
- `TextView txt = (TextView) findViewById(R.id.textView1);`
- `txt.setText(testStr);`
- `}`



Χρήση προτιμήσεων χρήστη (2/2)

□ Επεξήγηση

- SharedPreferences: Αντικείμενο για την αποθήκευση των προτιμήσεων του χρήστη
- PreferenceManager: Κλάση διαχείρισης προτιμήσεων
- setDefaultValues(): Αποθήκευση τιμών προτιμήσεων (από το XML στο αντικείμενο SharedPreferences
 - Όρισμα false: Reset τιμές βάσει XML (false: reset μόνο την πρώτη φορά που εκτελείται η activity σε αυτή τη συσκευή)
- getDefaultSharedPreferences(): Ανάκτηση προτιμήσεων για τη συγκεκριμένη activity

Χρήση αλλαγής επιλογών κατά την εκτέλεση της εφαρμογής



- Προσθήκη Listener στη MainActivity
 - ... implements OnSharedPreferencesChangeListener
- Υλοποίηση Listener
 - `public void onSharedPreferencesChanged(SharedPreferences sharedPreferences, String key) {`
 - `testStr = prefs.getString("prefUsername", "");`
 - `TextView txt = (TextView) findViewById(R.id.textView1);`
 - `txt.setText(testStr);`
- Αλλαγή `onResume()`
 - `super.onResume();`
 - `prefs.registerOnSharedPreferencesChangeListener(this);`



Σημερινή διάλεξη

- Σύνοψη προηγούμενης διάλεξης
- Intents
 - Διαδικασία: Implicit intent
- Ρυθμίσεις / προτιμήσεις
 - Εισαγωγή
 - Ιδιότητες
 - Διαδικασία
- Βάση δεδομένων
 - Εισαγωγή
 - Διαχείριση πινάκων
 - Διαχείριση βάσης δεδομένων



Εισαγωγή

- Η βάση δεδομένων είναι ενσωματωμένη
- Όταν δημιουργείται η βάση δεδομένων SQLite, αποθηκεύεται σε ένα αρχείο στη συσκευή
 - Στο αρχείο έχει πρόσβαση η εφαρμογή που το δημιούργησε
- Υποστηρίζει 3 τύπους δεδομένων
 - TEXT
 - INTEGER
 - REAL
- Άλλοι τύποι δεδομένων θα πρέπει να μετατραπούν πριν την αποθήκευση



Διαχείριση πινάκων

- Με χρήση SQL Statements (CREATE TABLE, DROP TABLE, κλπ)
- Βασικές ιδιότητες στηλών
 - PRIMARY KEY: Καθορίζει μια στήλη primary key
 - AUTO INCREMENT: Η τιμή αυξάνεται με την εισαγωγή κάθε γραμμής
 - NOT NULL: Καθορίζει ότι οι τιμές της στήλης δεν μπορεί να είναι NULL
 - UNIQUE: Καθορίζει ότι οι τιμές πρέπει να είναι μοναδικές
 - Παράδειγμα:
 - CREATE TABLE test (
 - _id INTEGER PRIMARY KEY AUTOINCREMENT,
 - test INTEGER NOT NULL,
 - description TEXT
 -)

Διαχείριση βάσης δεδομένων (1/3)



- Με χρήση (κληρονομικότητα) της κλάσης SQLiteOpenHelper
 - onCreate(): Εκτελείται εάν δεν υπάρχει η συγκεκριμένη βάση
 - onUpgrade(): Εκτελείται εάν υπάρχει η βάση δεδομένων και νεότερη έκδοση της βάσης είναι διαθέσιμη
 - query(table, columns, where, whereArgs, groupBy, having, orderBy): Αναζήτηση στο συγκεκριμένο πίνακα και επιστροφή cursor στο αποτέλεσμα
 - getReadableDatabase() / getWritableDatabase(): Άνοιγμα βάσης δεδομένων μόνο για ανάγνωση / ανάγνωση και εγγραφή
 - ▣ π.χ. SQLiteDatabase db = this.getWritableDatabase();
- Βασική μέθοδος
 - execSQL(sqlString): Εκτέλεση συγκεκριμένου SQL Statement

Διαχείριση βάσης δεδομένων (2/3)



□ Cursor

- Χρησιμοποιείται για τη διαχείριση των αποτελεσμάτων
 - ▣ π.χ. `Cursor cursor = db.query(TABLE_PHONES,COLUMNS," id = ?", new String[] { String.valueOf(id) }, null, null, null, null);`
- `moveToFirst()`: Μετακίνηση στην πρώτη γραμμή στον cursor
- `moveToNext()`: Μετακίνηση στην τελευταία γραμμή στον cursor
- `close()`: Κλείσιμο

□ Σταθερές

- Name: Όνομα βάσης δεδομένων (συνηθίζεται `name.db`)
- Version: Έκδοση (αρχική 1)

Διαχείριση βάσης δεδομένων (3/3)



□ ContentValues

- Χρησιμοποιείται για την αποθήκευση δεδομένων στις αντίστοιχες στήλες
 - ▣ π.χ. `ContentValues values = new ContentValues();`
 - ▣ `values.put("name", phones.getName());`

□ Μέθοδοι διαχείρισης δεδομένων

- `insert (table, columns, values)`: Εισαγωγή γραμμής και επιστροφή id για τη νέα γραμμή
- `update (table, columns, values)`: Ενημέρωση γραμμών και επιστροφή IDs για τις γραμμές που ενημερώθηκαν με επιτυχία
 - ▣ π.χ. `db.insert(TABLE_PHONES,null,values);`