Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων

Διαδικτυακά και Φορητά Πληροφοριακά Συστήματα

(Δικτυακές Υπηρεσίες)

10η Διάλεξη

Δημοσθένης Κυριαζής

Δευτέρα 21 Μαΐου 2018

Σημερινή διάλεξη

- Σύνοψη προηγούμενης διάλεξης
- Βάση δεδομένων
  - Εισαγωγή
  - Διαχείριση πινάκων
  - Διαχείριση βάσης δεδομένων
  - Διαδικασία / παράδειγμα
- Threads, Αρχεία
  - Εισαγωγή
  - AsyncTask
  - Αρχεία
  - Διαδικασία / παράδειγμα

Preferences

- Οι προτιμήσεις (preferences / settings) επιτρέπουν στους χρήστες να διαχειριστούν τη λειτουργικότητας 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)

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

- 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" />
```

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();
```

Χρήση προτιμήσεων χρήστη

- 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 αλλαγής επιλογών κατά την εκτέλεση της εφαρμογής

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

Σημερινή διάλεξη

- Σύνοψη προηγούμενης διάλεξης
- Βάση δεδομένων
  - Εισαγωγή
  - Διαχείριση πινάκων
  - Διαχείριση βάσης δεδομένων
  - Διαδικασία / παράδειγμα
- Threads, Αρχεία
  - Εισαγωγή
  - AsyncTask
  - Αρχεία
  - Διαδικασία / παράδειγμα

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 SQLLiteOpenHelper
  - 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)

- ContentValue
  - 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);

Σημερινή διάλεξη

- Σύνοψη προηγούμενης διάλεξης
- Βάση δεδομένων
  - Εισαγωγή
  - Διαχείριση πινάκων
  - Διαχείριση βάσης δεδομένων
  - Διαδικασία / παράδειγμα
- Threads, Αρχεία
  - Εισαγωγή
  - AsyncTask
  - Αρχεία
  - Διαδικασία / παράδειγμα

Διαδικασία (1/10)

- Δημιουργία διεπαφής με τα ακόλουθα widgets
  - 1 Text View (θα αντιστοιχεί στο ID)
  - 2 Edit Text (θα αντιστοιχούν στα Name, Phone)
  - 2 Small Button (θα αντιστοιχούν στα Add, Find)

Διαδικασία (2/10)

- Δημιουργία κλάσης για το μοντέλο δεδομένων
  - Δεξί κλικ -> New -> Java Class (επιλογή main/java) -> Phone (όνομα)
- public class Phone {

```
private int _id;
private String _fullName;
private int _phoneNum;
public Phone(){
}
public Phone(int id, String fullName, int phoneNum) {
   this._id = id;
   this._fullName = fullName;
   this._phoneNum = phoneNum;
}
public Phone (String fullName, int phoneNum) {
   this._fullName = fullName;
   this._phoneNum = phoneNum;
}
public void setID(int id) {
      this._id = id;
   }
public int getID() {
      return this._id;
   }
public void setFullName(String
fullName) {
      this._fullName = fullName;
   }
public String getFullName() {
      return this._fullName;
   }
public void setPhoneNum(int
phoneNum) {
      this._phoneNum = phoneNum;
   }
public int getPhoneNum() {
      return this._phoneNum;    }
```

Διαδικασία (3/10)

- Δημιουργία κλάσης για τη βάση δεδομένων
  - Όνομα: PhoneDBHandler
- public class PhoneDBHandler extends SQLiteOpenHelper {

```
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class PhoneDBHandler extends SQLiteOpenHelper {
    public PhoneDBHandler(Context context, String name, SQLiteDatabase.CursorFactory
        factory, int version) {
      super(context, name, factory, version);
     }
    @Override
    public void onCreate(SQLiteDatabase db) {
     }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
     }
}
```

Διαδικασία (4/10)

- Προσθήκη «στοιχείων» βάσης δεδομένων

```
private static final int DATABASE_VERSION = 1;
private static final String DATABASE_NAME = "phoneBookDB.db";
private static final String TABLE_PHONES = "phones";

public static final String COLUMN_ID = "_id";
public static final String COLUMN_FULLNAME = "fullname";
public static final String COLUMN_PHONENUM = "phone";

public PhoneDBHandler(Context context, String name,
SQLiteDatabase.CursorFactory factory, int version) {
      super(context, DATABASE_NAME, factory, DATABASE_VERSION);
    }
```

Διαδικασία (5/10)

- Υλοποίηση μεθόδων για τη διαχείριση της βάσης δεδομένων

```
@Override
public void onCreate(SQLiteDatabase db) {
        String CREATE_PHONES_TABLE = "CREATE TABLE " + TABLE_PHONES + "(" +
            COLUMN_ID + " INTEGER PRIMARY KEY," +
            COLUMN_FULLNAME + " TEXT," +
            COLUMN_PHONENUM + " INTEGER" + ")";
    db.execSQL(CREATE_PHONES_TABLE);
}
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion,
                                          int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_PHONES);
        onCreate(db);
}
```

Διαδικασία (6/10)

- Υλοποίηση μεθόδων για τη διαχείριση των δεδομένων
  - Όρισμα το instance του μοντέλου δεδομένων

```
public void addPhoneNum (Phone phone) {
    ContentValues values = new ContentValues();
    values.put(COLUMN_FULLNAME, phone.getFullName());
    values.put(COLUMN_PHONENUM, phone.getPhoneNum());

    SQLiteDatabase db = this.getWritableDatabase();

    db.insert(TABLE_PHONES, null, values);
    db.close();
}
```

Διαδικασία (7/10)

```
public Phone findPhoneNum(String fullName) {
        String query = "Select * FROM " + TABLE_PHONES + " WHERE " +
COLUMN_FULLNAME + " =  \' " + fullName + "\' ";
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(query, null);
        Phone phone = new Phone();
        if (cursor.moveToFirst()) {
                cursor.moveToFirst();
                phone.setID(Integer.parseInt(cursor.getString(0)));
                phone.setFullName(cursor.getString(1));
                phone.setPhoneNum(Integer.parseInt(cursor.getString(2)));
                cursor.close();
        } else {
                phone = null;
        }
        db.close();
        return phone;
}
```

Διαδικασία (8/10)

- Προσθήκη / ανάγνωση δεδομένων από τη MainActivity

```
TextView textViewID;
EditText editTextName;
EditText editTextPhone;

 public void newPhone (View view) {
      PhoneDBHandler dbHandler = new PhoneDBHandler(this, null, null, 1);

      int phoneNum = Integer.parseInt(editTextPhone.getText().toString());

      Phone phone = new Phone(editTextName.getText().toString(), phoneNum);

      dbHandler.addPhoneNum(phone);
      editTextName.setText("");
      editTextPhone.setText("");
  }
```

Διαδικασία (9/10)

```
public void findPhone (View view) {
    PhoneDBHandler dbHandler = new PhoneDBHandler(this, null, null, 1);

    Phone phone = dbHandler.findPhoneNum(editTextName.getText().toString());

    if (phone != null) {
       textViewID.setText(String.valueOf(phone.getID()));

       editTextPhone.setText(String.valueOf(phone.getPhoneNum()));
    } else {
       textViewID.setText("Name Not Found");
    }
  }
```

Διαδικασία (10/10)

- Σύνδεση μεθόδων με τα κουμπιά
  - android:onClick="newPhone"
  - android:onClick="findPhone"

Σημερινή διάλεξη

- Σύνοψη προηγούμενης διάλεξης
- Βάση δεδομένων
  - Εισαγωγή
  - Διαχείριση πινάκων
  - Διαχείριση βάσης δεδομένων
  - Διαδικασία / παράδειγμα
- Threads, Αρχεία
  - Εισαγωγή
  - AsyncTask
  - Αρχεία
  - Διαδικασία / παράδειγμα

Εισαγωγή

- Το thread είναι μια ροή εκτέλεσης σε μια εφαρμογή
- Κάθε εφαρμογή Android χρησιμοποιεί ένα μοναδικό thread, που ονομάζεται UI thread και εμφανίζει τη διεπαφή
- Οι ασύγχρονες διεργασίες εκτελούνται σε διαφορετικά threads στο background και δε χρειάζεται να συγχρονίζονται με άλλα threads
  - Για την υλοποίηση τους το Android περιλαμβάνει την κλάση AsyncTask

AsyncTask

- Χρησιμοποιεί generics για τη λειτουργία της κλάσης σε διαφορετικούς τύπους objects
  - 3 βασικοί τύποι: είσοδος, πρόοδος, αποτέλεσμα
    - Π.χ. AsyncTask<String, Integer,String>
- Μέθοδοι
  - onPreExecute(): Εκτελείται στο UI thread πριν την εκτέλεση της διεργασίας
  - doInBackground(Params…): Εκτελείται στο background thread αμέσως μετά την onPreExecute. Επιστρέφει αποτέλεσμα στη μέθοδο onPostExecute
  - onProgressUpdate(Progress…): Εκτελείται στο UI thread με την publishProgress
  - onPostExecute(Result): Εκτελείται στο UI thread μετά την doInBackground

Αρχεία

- Ανάγνωση και εγγραφή μέσω μεθόδων για δημιουργία αντικειμένων InputStream και FileOutputStream
  - openFileInput (filename)
  - openFileOutput (filename, mode)
    - Το mode καθορίζει την πρόσβαση στο αρχείο και τη διαχείριση του (π.χ. MODE_PRIVATE, MODE_APPEND)
    - Δεν ορίζεται path καθώς το Android χρησιμοποιεί το default path της εφαρμογής (βάσει του package name)

Διαδικασία (1/4)

- Στόχος: λήψη rss από http://rss.in.gr/feed/news/greece/
- Μεταβλητές στη MainActicity
  - private TextView textView;
  - private static String URL_STRING="http://rss.in.gr/feed/news/greece/";
  - final String FILENAME ="test.xml";

Διαδικασία (2/4)

- Δημιουργία inner class στη MainActivity

```
class GetFeed extends AsyncTask<String, Void, String>    {
        @Override
         protected String doInBackground(String... arg0) {
           try {
              String urlStr = arg0[0];
              URL url = new URL(urlStr);
              InputStream in = url.openStream();
              FileOutputStream out = openFileOutput(FILENAME,Context.MODE_PRIVATE);
              byte[] buffer = new byte[1024];
              int bytesRead = in.read(buffer);
              while (bytesRead!=-1) {
                                        out.write(buffer, 0, bytesRead);
                                        bytesRead = in.read(buffer);              }
              out.close();
              in.close();    }
            catch (IOException e)        {
              }
             return "finished";          }
```

Διαδικασία (3/4)

```
@Override
protected void onPostExecute(String result)
{
    Context context = MainActivity.this;
    try {
        InputStream is = openFileInput (FILENAME);
        InputStreamReader in = new InputStreamReader(is);
        BufferedReader bf = new BufferedReader(in);
        textView.setText(bf.readLine());
    }
    catch (IOException e) {
        //
    }
}
```

Διαδικασία (4/4)

- Προσθήκη στην onCreate() της MainActivity

```
textView = (TextView) findViewById(R.id.textView1);
new GetFeed().execute(URL_STRING);
```

- Προσθήκη στο AndroidManifest

```
<uses-permission
        android:name="android.permission.INTERNET"/>
```