Android Development

Persisting Data

SoftUni Team

Teodor Kostadinov

Software University

SoftUni Foundation

Software University

http://softuni.bg

Questions

Software University

sli.do

#Android

Shared Preferences

Software University

- It's an encrypted file that is only readable by your app
- You can have many files with different names
- It's a key-value pair format of the data
- Only simple data can be put there
- Used for persisting simple user settings and keys

```
SharedPreferences sharedPref =
        context.getSharedPreferences(FILE_NAME, Context.MODE_PRIVATE);
SharedPreferences.Editor editor = sharedPref.edit();
editor.putInt(KEY_H_SCORE, newHighScore);
editor.commit();
```

Source: https://developer.android.com/training/data-storage/shared-preferences#java

Shared Preferences

Software University

- Use commit only when you are off the main thread
- Use apply to change the SP file asynchronously

```
SharedPreferences sharedPref =
        context.getSharedPreferences(FILE_NAME, Context.MODE_PRIVATE);
int highScore = sharedPref.getInt(KEY_H_SCORE, defaultValue);
```

Source: https://developer.android.com/training/data-storage/shared-preferences#java

Demo

SP

File Storage

Software University

- You can read and write files to the device
- There are two types of storage - Internal and External
- Writing a file to Internal storage makes it private to your app only
  - You have to do special magic to make it available to other apps
  - Files in Internal storage add to the file size of your app when you open Settings
- Writing a file to the external storage makes this file visible to everyone

Source: https://developer.android.com/training/data-storage/files

Internal Storage

Software University

- You can get the directory with getFilesDir()
- You can get a cache directory with getCacheDir()
  - Cache directory is for temporary files that can be deleted without hurting your app
  - The Android OS occasionally or when storage is low, may delete these directories
- No permissions are required for these operations

```
try {
    outputStream = openFileOutput(filename, Context.MODE_PRIVATE);
    outputStream.write(fileContents.getBytes());
    outputStream.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

External Storage

- You can save two types of files on the External storage
  - Both types are publicly accessible
  - Private files get deleted with your app
- Permission is required: WRITE_EXTERNAL_STORAGE
- You should verify if this storage exists

```
/* Checks if external storage is available for read and write */
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
```

External Storage

- When saving public files, choose the correct directory
- The Environment class has constants for all the public folders
- Your files should be placed in the correct folder to allow the system to index them

```
// Get the directory for the user's public pictures directory.
    File file = new File(Environment.getExternalStoragePublicDirectory(
            Environment.DIRECTORY_PICTURES), albumName);
    if (!file.mkdirs()) {
        Log.e(LOG_TAG, "Directory not created");
    }
    return file;
```

Demo

Storage

Databases

Software University

- When needing a database in Android, some options exist:
  - Native SQLite database, low level
  - External wrapper library over SQLite (ORM)
  - Library for NoSql
  - Room

SQLite

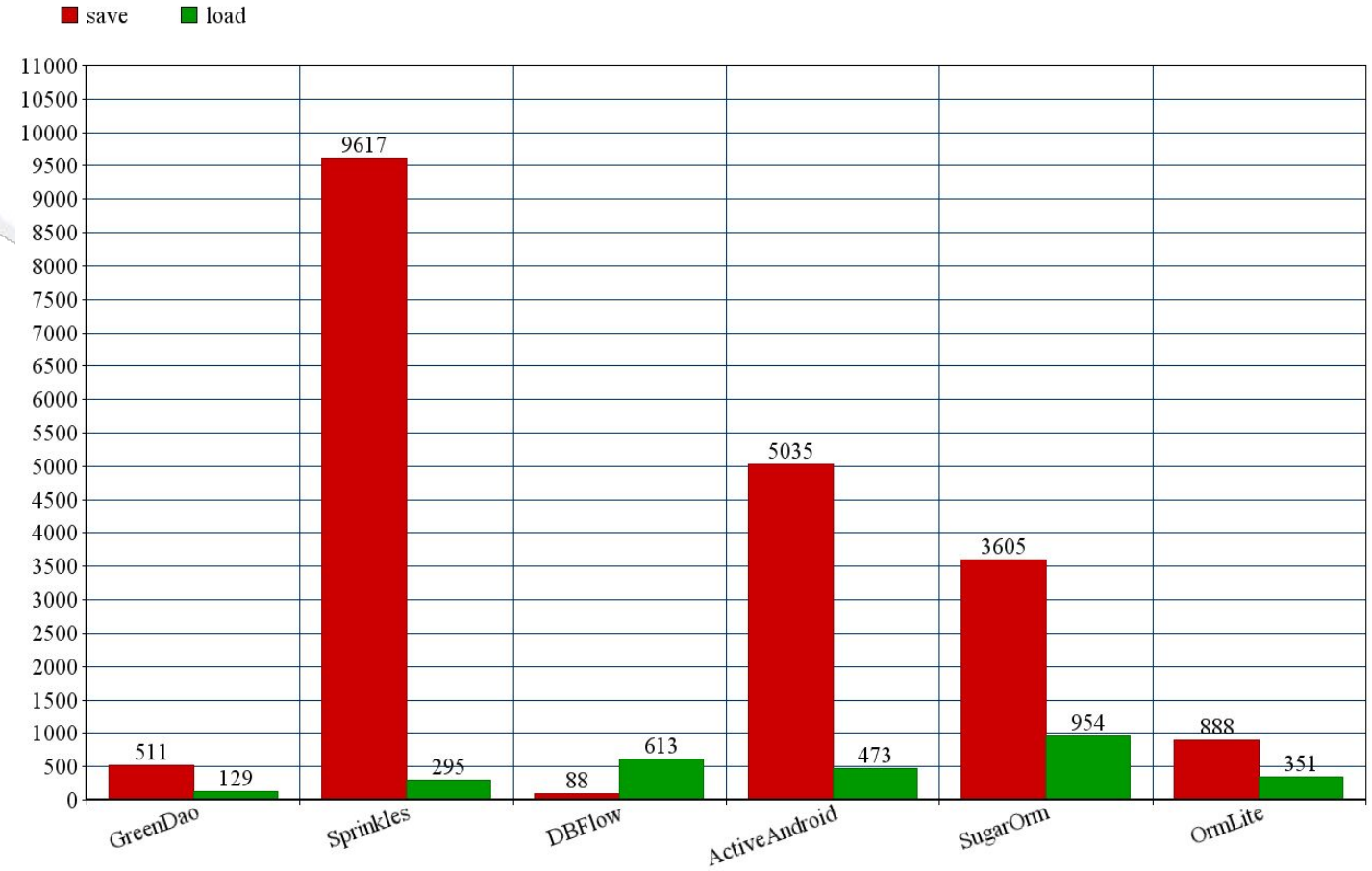
Software University

- Базата данни по подразбиране в Андроид
- Използва се чрез API на сравнително ниско ниво
- Това я прави опасна
- Ако използвате директно SQLite е препоръчително да използвате абстракция, че да се минимизират ръчните грешки
- Това е най-бързия вариант за база данни на устройството
- Трябва да се наследи класа SQLiteOpenHelper
- Той позволява достъп до базата за извършване на CRUD

Source: https://developer.android.com/training/data-storage/sqlite

ORMs

- Object Relational Mapping
- Библиотеки, които, използвайки абстракция, позволяват записване/извличане на Java обекти в базата данни
- ORM Lite
- SugarOrm
- ActiveAndroid
- DBFlow



Room

- Подобно на ORM е абстракция върху SQLite
- Позволява лесно поддържане на актуална онлайн база
- В бъдеще ще позволи заменянето на SQLite с по-добра база данни
- Query-тата се проверяват по време на компилация!!
- Има три главни части
- Database - основен дб контролер
- Entity - данни за една таблица в базата
- Dao - функции на една таблица в базата

Source: https://developer.android.com/training/data-storage/room

Room - Entity

Software University

```
@Entity
public class User {
    @PrimaryKey
    private int uid;

    @ColumnInfo(name = "first_name")
    private String firstName;

    @ColumnInfo(name = "last_name")
    private String lastName;

    // Getters and setters are ignored for brevity,
    // but they're required for Room to work.
}
```

Room - Dao

Software University

```
@Dao
public interface UserDao {
    @Query("SELECT * FROM user") List<User> getAll();

    @Query("SELECT * FROM user WHERE uid IN (:userIds)")
    List<User> loadAllByIds(int[] userIds);

    @Query("SELECT * FROM user WHERE first_name LIKE :first AND "
           + "last_name LIKE :last LIMIT 1")
    User findByName(String first, String last);

    @Insert void insertAll(User... users);

    @Delete void delete(User user);
}
```

Room - Database

```
@Database(entities = {User.class}, version = 1)
public abstract class AppDatabase extends RoomDatabase {
    public abstract UserDao userDao();
}
```

Използване:

```
AppDatabase db = Room.databaseBuilder(getApplicationContext(),
                AppDatabase.class, "database-name").build();
db.userDao().insertAll(new User());
```

Demo

Room

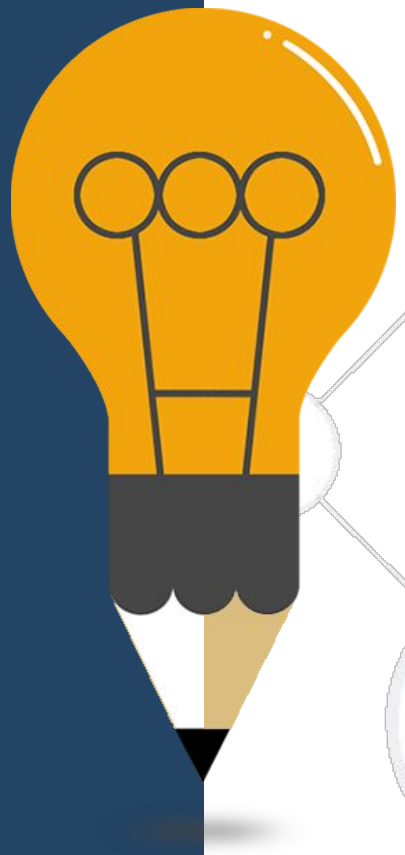
Summary and Resources

Software University

When in need of a database in Android - use Room.

Resources:

Homework (1)



Направете приложение, което има едно поле за въвеждане на код на баркод и един бутон. При натискане на бутона се прави заявка до https://world.openfoodfacts.org/api/v0/product/737628064502.json На потребителя се показва името на продукта и какво съдържа. Информацията се записва в локалната база.
При повторно търсене на същия баркод, информацията се търси първо в локалната база, и чак ако я няма се правя заявка до апи-то.

Този модел работи с данните от АПИ-то:

```
Product {
        String code; //the code may have a leading zero
        String product_name;
        String ingredients_text;
}
```

Може да прочетете повече за АПИ-то на https://world.openfoodfacts.org/data
Може да надградите приложението, като вместо да се въвежда, баркода се сканира с камерата на телефона. Библиотека за това е https://github.com/zxing/zxing

Questions?

SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

СофтУни диамантени партньори

Software University

INDEAVR
Serving the high achievers

INFRAGISTICS®

SoftwareGroup
doing it right

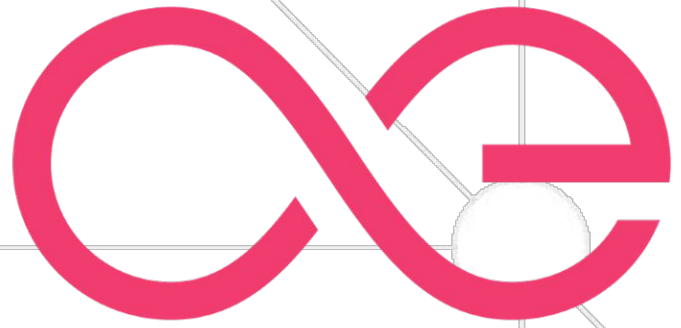
xssoftware

NETPEAK

SUPER HOSTING .BG®

СофтУни диамантени партньори

Software University



Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg