

PROGRAMMING HANDHELD SYSTEMS

DATA MANAGEMENT

TODAY'S TOPICS

SHARED PREFERENCES

INTERNAL STORAGE

EXTERNAL STORAGE

SQLITE DATABASES

SHARED PREFERENCES

SMALL AMOUNTS OF PRIMITIVE DATA

INTERNAL STORAGE

SMALL TO MEDIUM AMOUNTS OF PRIVATE
DATA

EXTERNAL STORAGE

LARGER AMOUNTS OF NON-PRIVATE DATA

DATABASES

STORE SMALL TO LARGE AMOUNTS OF
PRIVATE, STRUCTURED DATA

SHARED PREFERENCES

A PERSISTENT MAP

HOLDS KEY-VALUE PAIRS OF SIMPLE DATA
TYPES

AUTOMATICALLY PERSISTED ACROSS
APPLICATION SESSIONS

SHARED PREFERENCES

OFTEN USED FOR LONG-TERM STORAGE OF
CUSTOMIZABLE APPLICATION DATA

ACCOUNT NAME

FAVORITE WIFI NETWORKS

USER CUSTOMIZATIONS

ACTIVITY SHARED PREFERENCES

TO GET A SharedPreferences OBJECT
ASSOCIATED WITH A GIVEN ACTIVITY

Activity.getSharedPreferences (int mode)

MODE_PRIVATE

NAMED SHARED PREFERENCES

```
Context.getSharedPreferences (  
    String name, int mode)
```

NAME – NAME OF SharedPreferences FILE

MODE – MODE_PRIVATE

WRITING SHARED PREFERENCES

CALL `SharedPreferences.edit()`

RETURNS A `SharedPreferences.Editor`
INSTANCE

WRITING SHARED PREFERENCES

ADD VALUES TO SharedPreferences USING
SharedPreferences.Editor instance

`putInt(String key, int value)`

`putString(String key, String value)`

`remove(String key)`

WRITING SHARED PREFERENCES

COMMIT EDITED VALUES WITH
`SharedPreferences.Editor.commit()`

READING SHARED PREFERENCES

USE SharedPreferences METHODS TO READ
VALUES

`getAll()`

`getBoolean(String key, ...)`

`getString(String key, ...)`

DATAMANAGEMENTSHAREDREFERENCES

WHEN THE USER PRESSES THE PLAY BUTTON,
THE APPLICATION DISPLAYS A RANDOM
NUMBER

THE APPLICATION KEEPS TRACK OF THE
HIGHEST NUMBER SEEN SO FAR



PREFERENCEFRAGMENT

A CLASS THAT SUPPORTS DISPLAYING &
MODIFYING USER PREFERENCES

DATAMANAGEMENT PREFERENCEFRAGMENT

THIS APPLICATION DISPLAYS A
PREFERENCEFRAGMENT, WHICH ALLOWS THE
USER TO ENTER AND CHANGE A PERSISTENT
USER NAME



FILE

CLASS REPRESENTS A FILE SYSTEM ENTITY
IDENTIFIED BY A PATHNAME

FILE

STORAGE AREAS ARE CLASSIFIED AS
INTERNAL OR EXTERNAL

INTERNAL MEMORY USUALLY USED FOR
SMALLER, APPLICATION PRIVATE DATA SETS

EXTERNAL MEMORY USUALLY USED FOR
LARGER, NON-PRIVATE DATA SETS

FILE API

FILEOUTPUTSTREAM OPENFILEOUTPUT (
STRING NAME, INT MODE)

OPEN PRIVATE FILE FOR WRITING. CREATES THE FILE IF
IT DOESN'T ALREADY EXIST

FILEINPUTSTREAM OPENFILEINPUT (
STRING NAME)

OPEN PRIVATE FILE FOR READING

MANY OTHERS. SEE DOCUMENTATION.

DATAMANAGEMENT

FILEINTERNALMEMORY

IF A TEXT FILE DOES NOT ALREADY EXIST,
APPLICATION WRITES TEXT TO THAT TEXT FILE

APPLICATION THEN READS DATA FROM THE
TEXT FILE AND DISPLAYS IT



USING EXTERNAL MEMORY FILES

REMOVABLE MEDIA MAY APPEAR/DISAPPEAR
WITHOUT WARNING

USING EXTERNAL MEMORY FILES

String Environment.

`getExternalStorageState()`

`MEDIA_MOUNTED` – PRESENT & MOUNTED
WITH READ/WRITE ACCESS

`MEDIA_MOUNTED_READ_ONLY` – PRESENT
& MOUNTED WITH READ-ONLY ACCESS

`MEDIA_REMOVED` – NOT PRESENT

ETC.

USING EXTERNAL MEMORY FILES

PERMISSION TO WRITE EXTERNAL FILES

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

DATAMANAGEMENT

FILEEXTERNALMEMORY

APPLICATION READS AN IMAGE FILE FROM
THE RESOURCES DIRECTORY

COPIES THAT FILE TO EXTERNAL STORAGE

READS IMAGE DATA FROM THE FILE IN
EXTERNAL STORAGE

THEN DISPLAYS THE IMAGE



CACHE FILES

TEMPORARY FILES THAT MAY BE DELETED BY
THE SYSTEM WHEN STORAGE IS LOW

FILES REMOVED WHEN APPLICATION
UNINSTALLED

CACHE FILES

File Context.getCacheDir()

RETURNS ABSOLUTE PATH TO AN
APPLICATION-SPECIFIC DIRECTORY THAT CAN
BE USED FOR TEMPORARY FILES

SAVING CACHE FILES

`Context.getExternalCacheDir()`

RETURNS A FILE REPRESENTING EXTERNAL
STORAGE DIRECTORY FOR CACHE FILES

SQLITE

SQLITE PROVIDES IN-MEMORY DATABASE

DESIGNED TO OPERATE WITHIN A VERY SMALL
FOOTPRINT (<300kB)

IMPLEMENTS MOST OF SQL92

SUPPORTS ACID TRANSACTIONS

ATOMIC, CONSISTENT, ISOLATED & DURABLE

USING A DATABASE

RECOMMENDED METHOD RELIES ON A HELPER
CLASS CALLED SQLiteOpenHelper

USING A DATABASE

SUBCLASS SQLiteOpenHelper

CALL SUPER() FROM SUBCLASS

CONSTRUCTOR TO INITIALIZE UNDERLYING
DATABASE

USING A DATABASE

OVERRIDE onCreate()

OVERRIDE onUpgrade()

EXECUTE CREATE TABLE COMMANDS

USING A DATABASE

USE SQLITEOPENHELPER METHODS TO
OPEN & RETURN UNDERLYING DATABASE

EXECUTE OPERATIONS ON UNDERLYING
DATABASE

DATAMANAGEMENTSQL

APPLICATION CREATES AN SQLITE DATABASE AND INSERTS RECORDS, SOME WITH ERRORS, INTO IT

WHEN USER PRESSES THE FIX BUTTON, THE APPLICATION DELETES, UPDATES AND REDISPLAYS THE CORRECTED DATABASE RECORDS



EXAMINING THE DATABASE REMOTELY

DATABASES STORED IN

/data/data/<package name>/databases/

CAN EXAMINE DATABASE WITH SQLITE3

```
# adb -s emulator-5554 shell
```

```
# sqlite3 /data/data/  
course.examples.DataManagement.Data  
BaseExample/databases/artist_dbd
```

NEXT TIME

CONTENTPROVIDER