Programming
Handheld Systems
Data Management
Today’s Topics

SharedPreferences
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
SharedPreferences

A persistent map

Holds key-value pairs of simple data types

Automatically persisted across application sessions
SharedPreferences

Often used for long-term storage of customizable application data

Account name

Favorite WiFi networks

User customizations
Activity SharedPreferences

To get a SharedPreferences object associated with a given Activity
Activity.getSharedPreferences (int mode)
    MODE_PRIVATE
Named SharedPreferences

Context.getSharedPreferences (String name, int mode)

NAME – NAME OF SharedPreferences FILE
MODE – MODE_PRIVATE
Writing SharedPreferences

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 SharedPreferences

Commit edited values with
SharedPreferences.Editor.commit()
Reading SharedPreferences

Use SharedPreferences methods to read values

getAll()

getBoolean(String key, ...)

getString(String key, ...)
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.
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 redisplayes 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