Recording in Progress

This class is being recorded

Please turn off your video and/or video if you do not wish to be recorded
CMSC436: Programming handheld systems
Data Management
Today’s Topics

SharedPreferences
Internal Storage
External Storage
SQLite databases
Shared Preferences

Use when you want to store small amounts of primitive data

A persistent map that holds key-value pairs of simple data types

Automatically persisted across application sessions
SharedPreferences

This mechanism is expensive. Should be used for infrequently changing, loss-intolerant data.

Typically used for long-term storage of customizable application data, such as:

- Account name
- Favorite WiFi networks
- User settings
Activity SharedPreferences

Get a SharedPreferences Object associated with a given Activity

Activity.getPreferences (int mode)

    MODE_PRIVATE is default mode
Named SharedPreferences

Get named SharedPreferences file

Single SharedPreferences object for a given name

Context.getSharedPreferences

- String name, int mode

name – name of SharedPreferences file

mode – e.g., MODE_PRIVATE (other modes deprecated)
Writing SharedPreferences

Call SharedPreferences.edit()

Returns a SharedPreferences.Editor instance
Writing SharedPreferences

Use SharedPreferences.Editor instance to add values to SharedPreferences
putInt(String key, int value)
putString(String key, String value)
remove(String key)
Writing SharedPreferences

Commit edited values with Editor.commit() or Editor.apply()

  commit() writes updates synchronously and returns a success result

  apply() writes updates asynchronously and has no return value
Reading SharedPreferences

Use SharedPreferences methods to read values
getAll()
getBoolean(String key, ...)
getString(String key, ...)
DataManagementSharedPreferences

When the user presses the play button, the application displays a random number.
The application keeps track of the highest number seen so far.
DataManagement
SharedPreferences

HighScore:
461

Play
Reset
PreferenceFragment

A class that supports displaying & modifying user preferences
DataManagementPreferenceFragment

This application displays a PreferenceFragment, which allows the user to enter and change a persistent username
DataManagementPreferenceFragment

User Name
Adam

Change User Name
Enter Your User Name
Fred

CANCEL SUBMIT
Internal Storage

Use when you want to store small to medium amounts of private data
External Storage

Use when you want to store larger amounts of possibly non-private data
File

Class that 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.
DataManagementFileInternalMemory

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
Line 1: This is a test of the File Writing API
Line 2: This is a test of the File Writing API
Line 3: This is a test of the File Writing API
Using External Memory Files

Removable media may appear/disappear without warning
Using External Memory Files

String Environment.getExternalStorageState()

Returns

MEDIA_MOUNTED - present & mounted with read/write access

MEDIA_MOUNTED_READ_ONLY - present & mounted with read-only access

MEDIA_REMOVED - not present
If not done already, application reads an image file from its /res/raw directory.

Copies that file to external storage.

Application then reads image data from the file in external storage and then displays the image.

DataManagementFileExternalMemory
DataManagement
FileExternalMemory
Cache Files

Temporary files that may be deleted by the system when storage is low

These files are removed when application is 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
Databases

Use when you want to store small to large amounts of private, structured data
SQLite

SQLite provides in-memory database
Designed to operate within a very small footprint (e.g., <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

Create a subclass of 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.
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>Frank Sinatra</td>
</tr>
<tr>
<td>26</td>
<td>Lady Gaga</td>
</tr>
<tr>
<td>27</td>
<td>Johnny Cash</td>
</tr>
<tr>
<td>28</td>
<td>Ludwig van Beethoven</td>
</tr>
</tbody>
</table>
Examining the Database Remotely

Databases stored in

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

Can view databases using Android Studio Database Inspector

View -> Tool Windows -> App Inspection
Next Time

The Fragment Class
Sample code

DataManagementSharedPreferences
DataManagementPreferenceFragment
DataManagementFileInternalMemory
DataManagementFileExternalMemory
DataManagementSQL