

Gerenciamento de Dados

- Gerenciamento de dados em dispositivos Android
- Classes de suporte ao armazenamento de dados no Android

Referências

- DEITEL, Harvey, DEITEL, Paul, DEITEL, Abbey, MORGANO, Michael. Android para programadores: uma abordagem baseada em aplicativos. Porto Alegre: Bookman. 2013.
- Programming Handheld Systems – Prof. Adam Porter of University of Maryland (<https://www.coursera.org/course/android>)
- <http://developer.android.com>
- <https://developers.google.com/university/courses/mobile>
- <http://oreilly.com/training/androidapps>

SharedPreferences

- Conjuntos pequenos de dados primitivos
- Um mapa persistente com pares de chave-valor de um tipo simples de dado
- Os dados são automaticamente mantidos através das sessões das aplicações

SharedPreferences

- Frequentemente utilizados para o armazenamento a longo prazo de dados configuráveis de uma aplicação:
 - Dados da conta de um usuário
 - Redes WiFi do usuário
 - Configurações do usuário

Activity SharedPreferences

- Para obter um objeto `SharedPreferences` associado a uma `Activity`:

```
Activity.getSharedPreferences (int mode)  
MODE_PRIVATE
```

Named SharedPreferences

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

name – **nome do arquivo** SharedPreferences
mode – MODE_PRIVATE

Registrando *SharedPreferences*

- Chamar `SharedPreferences.edit()` para retornar uma instância do objeto

`SharedPreferences.Editor`

Registrando *SharedPreferences*

- A escrita de valores ao `SharedPreferences` é feita com o `SharedPreferences.Editor`:

```
putInt(String key, int value)  
putString(String key, String value)  
remove(String key)
```

- O `commit` dos valores editados é feito com `SharedPreferences.Editor.commit()`

Recuperando *SharedPreferences*

- A leitura do `SharedPreferences` é feita com os métodos:

`getAll()`

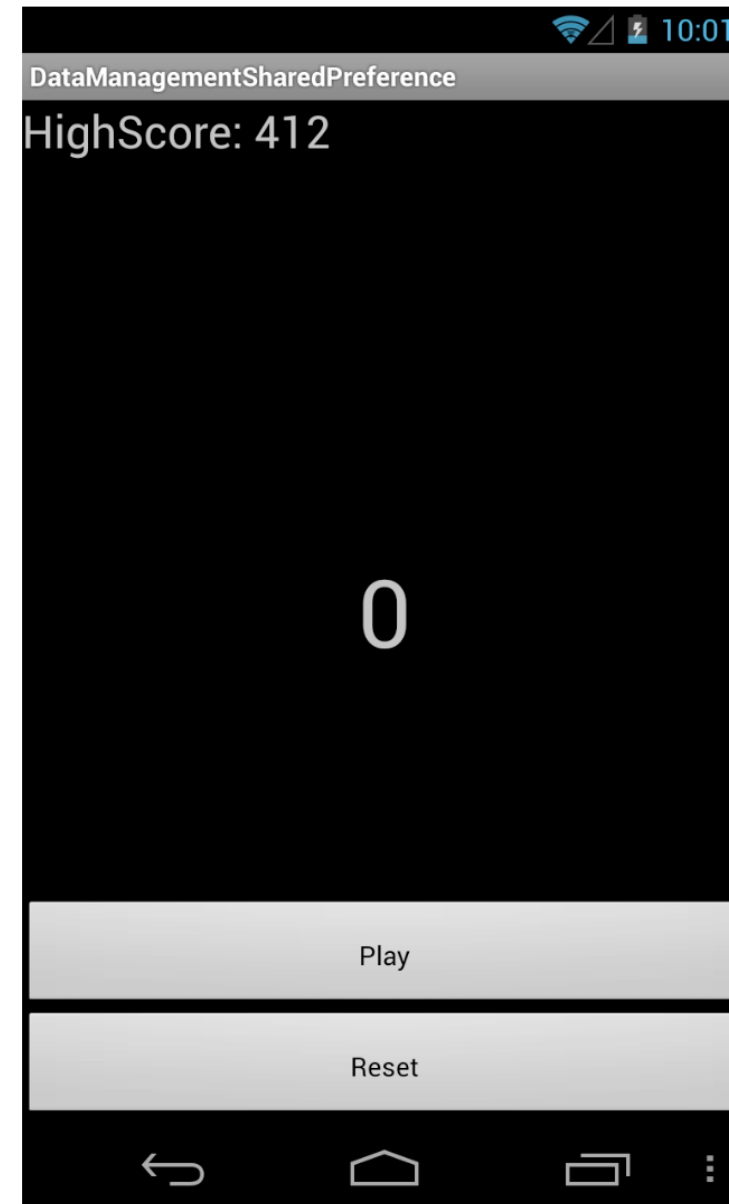
`getBoolean(String key, ...)`

`getString(String key, ...)`

Exemplo

- Quando o usuário pressiona o `Play`, a app apresenta um número randômico
- A aplicação mantém o maior número

Veja no SGA o Código-Fonte da App *DataManagementSharedPreferences*



Arquivo (`File`)

- Classe que representa uma entidade arquivo de sistema identificado pelo seu `pathname`

Arquivo (File)

- As áreas de armazenamento de arquivos são classificadas em `internal` e `external`
- `Internal memory` é normalmente utilizada para armazenar um conjunto pequeno de dados privados
- `External memory` é normalmente utilizada para armazenar conjuntos de dados não-privados, geralmente maiores

File API

- Para abrir um arquivo para escrita e cria-lo se não existir use:

```
FileOutputStream  
    openFileOutput (  
        String name, int mode)
```

File API

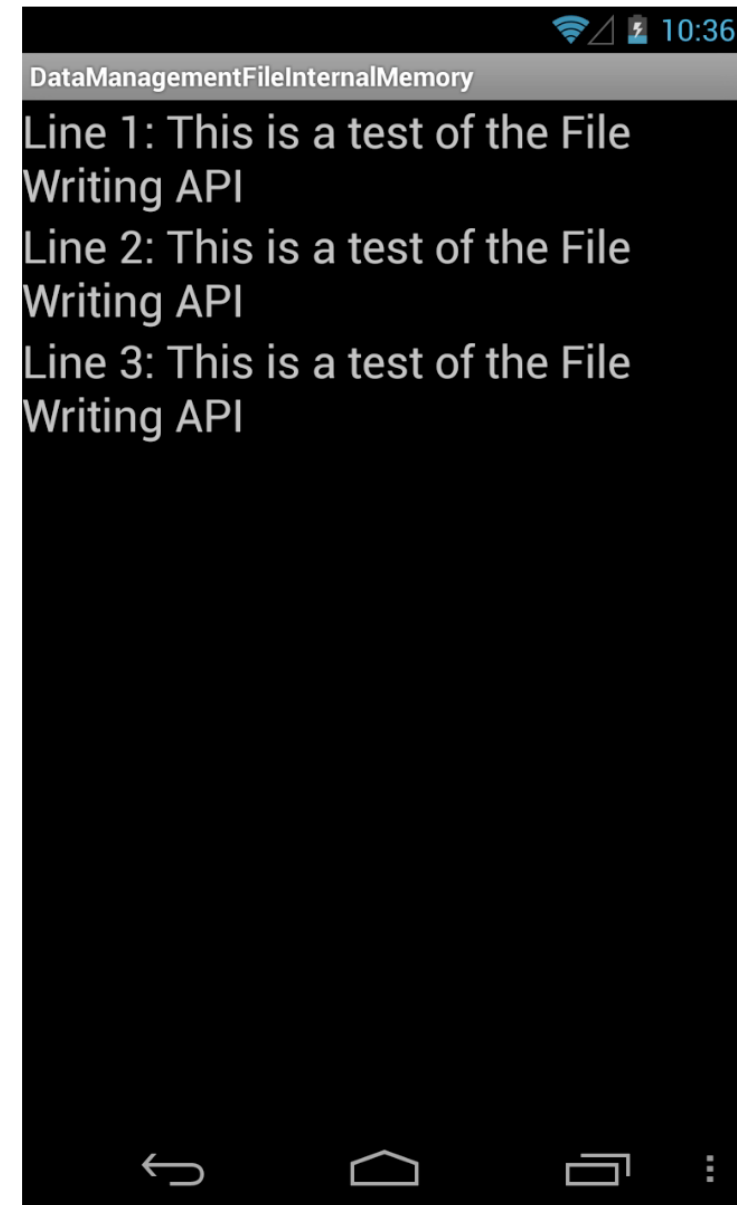
- Para abrir um arquivo privado para leitura use:

```
FileInputStream
```

```
    openFileInput (String name)
```

Exemplo

Veja no SGA o Código-Fonte da App
DataManagementFileInternalMemory



Utilizando Arquivos em Memória Externa

`StringEnvironment.`

`getExternalStorageState()`

- `MEDIA_MOUNTED` – presente e montado para acesso de escrita e leitura
- `MEDIA_MOUNTED_READ_ONLY` – presente e montado para acesso de somente leitura
- `MEDIA_REMOVED` – não presente

Utilizando Arquivos em Memória Externa

- Mídias removíveis podem aparecer ou desaparecer sem aviso
- Normalmente são utilizadas para armazenar um conjunto de dados não privados

Utilizando Arquivos em Memória Externa

- Permissão para escrita em arquivos externos

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

Exemplo

- Lê uma imagem de um arquivo
- Copia o arquivo para uma memória externa
- Lê a imagem do arquivo da memória externa
- Exibe a imagem desse arquivo

Veja no SGA o Código-Fonte da App *DataManagementFileExternalMemory*

Cache Files

- São arquivos temporários que podem ser excluídos pelo sistema quando se precisar de espaço de armazenamento
- São arquivos removidos quando a aplicação é removida do dispositivo

Cache Files

`File Context.getCacheDir()`

- Retorna o caminho absoluto para um diretório específico da aplicação que pode ser utilizado por arquivos temporários

Cache Files

`Context.getExternalCacheDir()`

- Retorna o caminho para um diretório específico da aplicação em uma memória externa que pode ser utilizado por arquivos temporários

SQLite

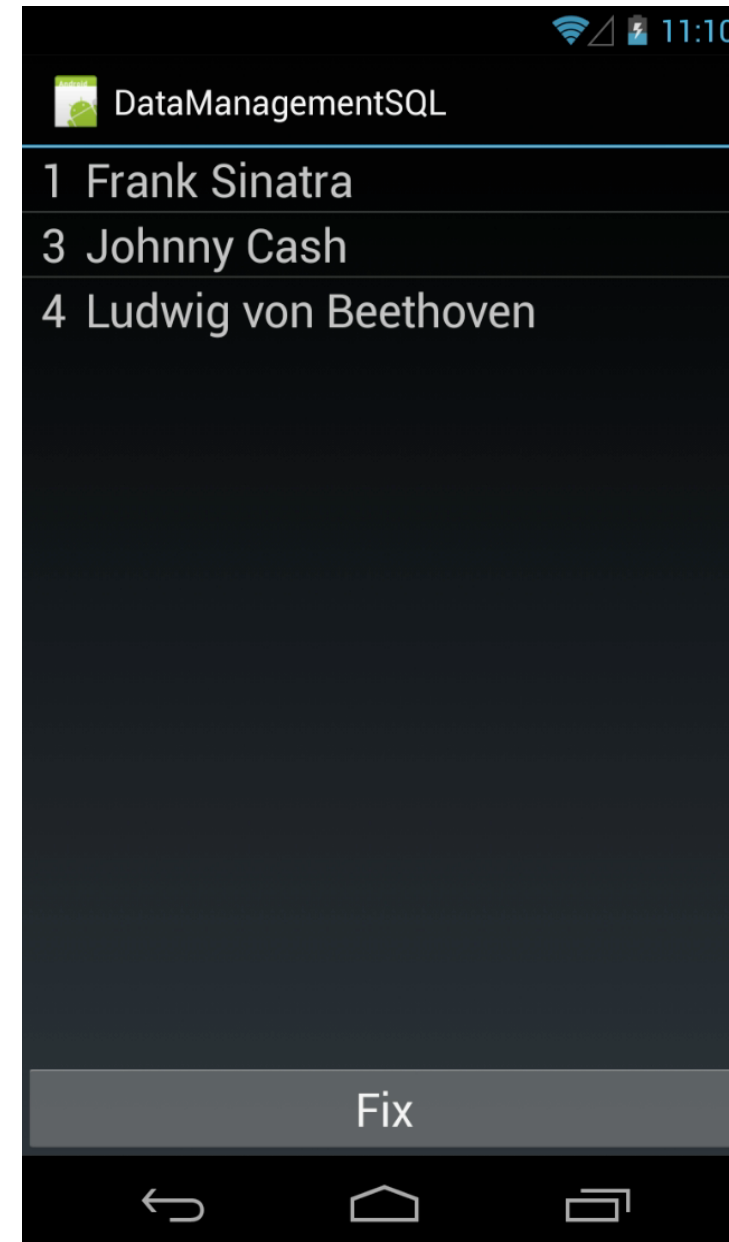
- O SQLite provê um banco de dados *in-memory*
- Ele é projetado para operar com bases de dados de até 300KB
- Implementa a maioria do padrão SQL92

Veja mais detalhes em <http://www.sqlite.org/>

Exemplo

- Cria um banco de dados SQLite e insere registros, alguns com erros
- Quando o usuário pressiona o botão `Fix`, a aplicação exclui, atualiza e reexibe os registros corretos

Veja no SGA o Código-Fonte da App
DataManagementSQL



Examinando o banco de dados

- Os banco de dados são armazenados em

`/data/<package name>/databases`

- Os banco de dados podem ser examinados o `sqlite3` no `adb`

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

```
# sqlite3 /data/data/
```

```
course.examples.DataManagement.Data  
BaseExample/databases/artist_dbd
```

TP2 (Parte 2/x)

1. Acrescentar a funcionalidade de se anexar conteúdo aos itens de menu criados; este conteúdo pode ser arquivos, vídeos, textos e links.
2. O código da app deve estar bem documentado
3. A app deve usar as melhores práticas de interface com o usuário