

Работа с файловой системой в Android

Чтение и сохранение файлов

Работа с настройками уровня activity и приложения позволяет сохранить небольшие данные отдельных типов (string, int), но для работы с большими массивами данных, такими как графически файлы, файлы мультимедиа и т.д., нам придется обращаться к файловой системе.

ОС Android построена на основе Linux. Этот факт находит свое отражение в работе с файлами. Так, в путях к файлам в качестве разграничителя в Linux использует слеш "/", а не обратный слеш "\" (как в Windows). А все названия файлов и каталогов являются регистрозависимыми, то есть "data" это не то же самое, что и "Data".

Приложение Android сохраняет свои данные в каталоге `/data/data/<название_пакета>/` и, как правило, относительно этого каталога будет идти работа.

Для работы с файлами абстрактный класс `android.content.Context` определяет ряд методов:

- `deleteFile(String name)`: удаляет определенный файл
- `fileList()`: получает все файлы, которые содержатся в подкаталоге */files* в каталоге приложения
- `getCacheDir()`: получает ссылку на подкаталог *cache* в каталоге приложения
- `getDir(String dirName, int mode)`: получает ссылку на подкаталог в каталоге приложения, если такого подкаталога нет, то он создается
- `getExternalCacheDir()`: получает ссылку на папку */cache* внешней файловой системы устройства
- `getExternalFilesDir()`: получает ссылку на каталог */files* внешней файловой системы устройства
- `getFilePath(String filename)`: возвращает абсолютный путь к файлу в файловой системе
- `openFileInput(String filename)`: открывает файл для чтения
- `openFileOutput (String name, int mode)`: открывает файл для записи

- Все файлы, которые создаются и редактируются в приложении, как правило, хранятся в подкаталоге */files* в каталоге приложения.
- Для непосредственного чтения и записи файлов применяются также стандартные классы Java из пакета `java.io`.

Пример

Итак, применим функционал чтения-записи файлов в приложении. Пусть у нас будет следующая примитивная разметка layout:

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <EditText
        android:id="@+id/save_text"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:gravity="start"
        android:layout_weight="4"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:layout_margin="16dp"
        android:layout_gravity="center"
        android:onClick="saveText"
        android:text="Сохранить"/>
    <TextView
        android:layout_marginTop="80dp"
        android:id="@+id/open_text"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:gravity="start"
        android:layout_weight="4"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_margin="16dp"
        android:layout_weight="1"
        android:layout_gravity="center"
        android:onClick="openText"
        android:text="Открыть"/>
</LinearLayout>
```

Поле `EditText` предназначено для ввода текста, а `TextView` - для вывода ранее сохраненного текста. Для сохранения и восстановления текста добавлены две кнопки.

Теперь в коде Activity пропишем
обработчики кнопок с сохранением и
чтением файла


```
package com.example.eugene.filesapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    private final static String FILE_NAME = "content.txt";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```
// сохранение файла
public void saveText(View view){

    FileOutputStream fos = null;
    try {
        EditText textBox = (EditText) findViewById(R.id.save_text);
        String text = textBox.getText().toString();

        fos = openFileOutput(FILE_NAME, MODE_PRIVATE);
        fos.write(text.getBytes());
        Toast.makeText(this, "Файл сохранен", Toast.LENGTH_SHORT).show();
    }
    catch(IOException ex) {

        Toast.makeText(this, ex.getMessage(), Toast.LENGTH_SHORT).show();
    }
    finally{
        try{
            if(fos!=null)
                fos.close();
        }
        catch(IOException ex){

            Toast.makeText(this, ex.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    }
}
```

```
// открытие файла
public void openText(View view){

    FileInputStream fin = null;
    TextView textView = (TextView) findViewById(R.id.open_text);
    try {
        fin = openFileInput(FILE_NAME);
        byte[] bytes = new byte[fin.available()];
        fin.read(bytes);
        String text = new String (bytes);
        textView.setText(text);
    }
    catch(IOException ex) {

        Toast.makeText(this, ex.getMessage(), Toast.LENGTH_SHORT).show();
    }
    finally{

        try{
            if(fin!=null)
                fin.close();
        }
        catch(IOException ex){

            Toast.makeText(this, ex.getMessage(), Toast.LENGTH_SHORT).show();
        }
    }
}
}
```

При нажатии на кнопку сохранения будет создаваться поток вывода `FileOutputStream fos = openFileOutput(FILE_NAME, MODE_PRIVATE)`

В данном случае введенный текст будет сохраняться в файл "content.txt". При этом будет использоваться режим `MODE_PRIVATE`

Система позволяет создавать файлы с двумя разными режимами:

- **MODE_PRIVATE**: файлы могут быть доступны только владельцу приложения (режим по умолчанию)
- **MODE_APPEND**: данные могут быть добавлены в конец файла

Поэтому в данном случае если файл "content.txt" уже существует, то он будет перезаписан. Если же нам надо было дописать файл, тогда надо было бы использовать режим `MODE_APPEND`:

- `FileOutputStream fos = openFileOutput(FILE_NAME, MODE_APPEND);`

Для чтения файла применяется поток ввода **`FileInputStream`**:

- `FileInputStream fin = openFileInput(FILE_NAME);`

В итоге после нажатия кнопки сохранения весь текст будет сохранен в файле `/data/data/название_пакета/files/content.txt`



15:11

FilesApp

Hello world

СОХРАНИТЬ

Hello world

ОТКРЫТЬ

