

Android Internal Storage: пример работы с внутренней памятью

Внутренняя память Android. Теория

Любое устройство на Android позволяет хранение личных данных во внутренней памяти. Если одно приложение сохраняет или загружает файл на внутреннюю память, то другие приложения не смогут получить доступ к этим файлам. **Этими файлами может распоряжаться только это приложение.** Когда пользователь удаляет его, то все внутренние файлы этого приложения также удаляются.

Тем не менее, обратите внимание, что некоторые пользователи могут получить root-доступ на свои Android устройства. В этом случае, пользователи смогут получать доступ **к любым файлам**.

Создание и чтение файлов с внутренней памяти Android

Android использует методы `openFileInput()` и `openFileOutput()` классов Java IO для работы с локальными файлами.

Обратите внимание, что эти методы **не принимают** на вход путь к файлу (например, **path/to/fileName.txt**), а просто получают имена файлов «**fileName.txt**».

Метод `openFileOutput()` используется для создания и сохранения файла. Пример использования приведен ниже:

Java

```
FileOutputStream fileOutputStream =  
    openFileOutput("fileName.txt", Context.MODE_PRIVATE);
```

Метод `openFileOutput()` возвращает экземпляр `FileOutputStream`. После этого мы можем записать данные в файл. Пример:

Java

```
String string = "какой-то текст, который мы будем записывать в файл";  
fileOutputStream.write(str.getBytes());  
fileOutputStream.close();
```

Метод `openFileInput()` используется для чтения данных из файла. Он возвращает экземпляр `FileInputStream`. Пример использования:

Java

```
1 FileInputStream fileInputStream = openFileInput(file);
```

Далее мы посимвольно считываем данные из файла в строку. Например, так:

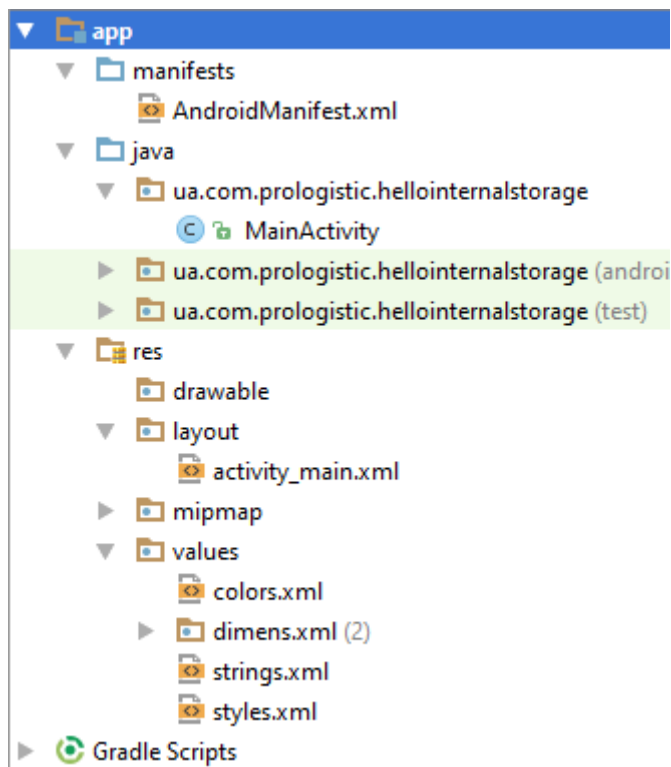
Java

```
int ch;  
  
String resultString = "";  
while((ch = fileInputStream.read()) != -1){  
    resultString = resultString + Character.toString((char)ch);  
}
```

```
fileInputStream.close();
```

Пример работы с внутренней памятью (Android Internal Storage). Практика

Ниже представлена структура нашего простого Android проекта:



Код макета `activity_main.xml` содержит кнопки `writeBtn` и `readBtn` для записи данных в файл и чтения. Обратите внимание, что мы определили метод `onClick` в макете, а не в коде (можете сами назначить слушатель в коде — это ничего не меняет. Просто мы учим Android, поэтому пробуем различные способы управлять поведением виджетов):

Java

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="ua.com.prologistic.hellointernalstorage.MainActivity">

    <TextView
        android:id="@+id/textView_title"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:padding="3dp"
android:text="Чтение/запись в Android Internal Storage"
android:textSize="24sp"
android:textStyle="bold" />
```

```
<EditText
    android:id="@+id/edit_field"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/textView_title"
    android:layout_margin="3dp"
    android:layout_marginTop="20dp"
    android:hint="Введите текст и нажмите 'Запись'"
    android:inputType="text"
    android:minLines="3">
    <!--фокус ввода данных будет в этом виджете-->
    <requestFocus />
</EditText>
```

```
<Button
    android:id="@+id/writeBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignEnd="@+id/edit_field"
    android:layout_alignRight="@+id/edit_field"
    android:layout_alignTop="@+id/readBtn"
    android:onClick="writeToFile"
    android:text="Запись в файл" />
```

```
<Button
    android:id="@+id/readBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/edit_field"
    android:layout_alignStart="@+id/edit_field"
    android:layout_centerVertical="true"
    android:onClick="readFromFile"
    android:text="Чтение из файла" />
```

```
</RelativeLayout>
```

Чтобы назначать метод `onClick()` в макете, мы должны определить метод в коде с параметром `view`, иначе Android выдаст ошибку, что метод имеет неправильную сигнатуру (в параметрах метода нет `view`).

Класс `MainActivity` содержит реализацию методов чтения и записи файлов:

```

package ua.com.prologistic.hellointernalstorage;

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

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;

public class MainActivity extends AppCompatActivity {

    private EditText editText;
    // определяем размер буфера при считывании с файла
    private static final int READ_BLOCK_SIZE = 100;

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

        editText = (EditText) findViewById(R.id.edit_field);
    }

    // метод для записи в файл
    public void writeToFile(View v) {
        // открываем файл по названию
        try {
            FileOutputStream fileOutputStream = openFileOutput("simplefile.txt",
                                                                MODE_PRIVATE);
            OutputStreamWriter outputStreamWriter = new
                OutputStreamWriter(fileOutputStream);
            outputStreamWriter.write(editText.getText().toString());
            outputStreamWriter.close();

            // создаем всплывающее окно с результатом выполнения записи в файл
            Toast.makeText(getApplicationContext(), "Запись в файл успешно проведена!",
                Toast.LENGTH_LONG).show();

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

        // метод для чтения из файла
        public void readFromFile(View v) {
            try {
                FileInputStream fileInputStream = openFileInput("simplefile.txt");
                InputStreamReader reader = new InputStreamReader(fileInputStream);

                char[] inputBuffer = new char[READ_BLOCK_SIZE];
                String s = "";
                int charRead;

                // цикл читает данные из файла,
                while ((charRead = reader.read(inputBuffer)) != -1) {
                    // конвертируем char в строку
                    String rString = String.copyValueOf(inputBuffer, 0, charRead);
                    s += rString;
                }
                reader.close();
                editText.setText(s);
                // создаем всплывающее окно с результатом выполнения чтения из файла
                Toast.makeText(getBaseContext(), "Чтение из файла успешно проведено!",
                               Toast.LENGTH_LONG).show();

            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

```

Результат работы с внутренней памятью (Android Internal Storage):



Мы ввели текст в поле, нажали кнопку 'Запись в файл' и получили сообщение об успешной записи. Теперь удалите текст из поля ввода и нажмите 'Чтение из файла' — вы получите сообщение об успешном чтении из файла.

Любознательным: Где хранится файл `simplefile.txt`?

Файл `simplefile.txt` хранится в вашем смартфоне по адресу

`data --> data --> [имя пакета] --> files --> simplefile.txt.`

В нашем случае

`[имя пакета] = ua.com.prologistic.hellointernalstorage.`

Доступ к этому файлу можно через File Explorer.

Следите за обновлениями раздела [Разработка под Android](#) и подписывайтесь на новые статьи!