**ПРИМЕР КОПИРОВАНИЯ ФАЙЛОВ**

**activity\_main.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/activity\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context="tva.knastu.com.copyfiles.MainActivity"  
 android:orientation="vertical"  
 android:weightSum="1"**>  
  
 <**TextView  
 android:text="Источник"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/textView3"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 android:id="@+id/textView"  
 android:layout\_below="@+id/textView3"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"** />  
  
 <**TextView  
 android:text="Приемник:"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/textView4"  
 android:layout\_marginBottom="18dp"  
 android:layout\_marginTop="100dp"** />  
  
 <**TextView  
 android:text="TextView"  
 android:layout\_width="match\_parent"  
 android:layout\_marginTop="0dp"  
 android:id="@+id/textView2"  
 android:layout\_height="wrap\_content"** />  
</**LinearLayout**>

**MainActivity.java**

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* Программа при запуске копирует все файлы из каталога приложения  
 \* с именем "/files" в системный каталог карты SD  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/***package** tva.knastu.com.copyfiles;  
  
**import** android.content.pm.PackageInfo;  
**import** android.content.pm.PackageManager;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.widget.TextView;  
  
**import** java.io.File;  
**import** java.io.FileInputStream;  
**import** java.io.FileOutputStream;  
**import** java.io.IOException;  
**import** java.nio.channels.FileChannel;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **protected** TextView **tv**, **tv2**;  
 File **source**, **dect**, **SD**, **f1**;  
 PackageInfo **p**;  
 String **root**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
*//получить указатели на компоненты "надпись"* **tv** = (TextView)findViewById(R.id.***textView***);  
 **tv2** = (TextView)findViewById(R.id.***textView2***);  
  
*//Получить указатель на менеджер программных пакетов* PackageManager m = getPackageManager();  
  
*//Получить название текущего пакета* String s = getPackageName();  
  
*//Получение пути к папке приложения* **try** {  
 *//получить указатель на информационную структуру программного пакета* **p** = m.getPackageInfo(s, 0);  
 *//получить строку с путем к текущему приложению (пакету)* **root** = **p**.**applicationInfo**.**dataDir**;  
 } **catch** (PackageManager.NameNotFoundException e) {  
 *//Обработка ошибки* Log.*w*(**"yourtag"**, **"Пакета с именем"** + s + **"НЕ существует!"**, e);  
 }  
*//Получение пути к внешней папке приложения на карте SD* **SD** = getExternalFilesDir(**null**);  
  
*//Формирование пути к источнику (к основному пути пакета добавим папку files* **source** = **new** File(**root**, **"//files//"**);  
*//выведем полученный путь в первом текстовом поле экрана* **tv**.setText(**source**.getPath());  
  
*//получим список объектов в каталоге по указанному пути* String list[] = **new** File(**source**.getPath()).list();

*//Цикл по всем элементам списка* **for**(**int** i = 0; i < list.**length**; i++) {  
 *// составляем объект файла из базового пути и имени файла из списка* **f1** = **new** File(**source**, list[i]);  
 *//если составленный объект является реальным файлом (НЕ каталогом), то* **if**(**f1**.isFile()) {  
 *//составим объект файла с тем же именем для карты SD* **dect** = **new** File(**SD**, list[i]);  
 *//Копируем файл из папки приложения на карту SD* **try** {  
 *copyFile*(**f1**, **dect**);  
 *//Выводим полный путь и имя файла в текстовое поле 2* **tv2**.setText(**f1**.getPath());  
 *//Выводим отметку в Лог об успешном копировании* Log.*d*(**"fileLog"**, **"копирование-"**+i+**" выполнено"**);  
 }  
 **catch** (IOException e) {  
 *//Если ошибка копирования, делаем отметку в Логе* e.printStackTrace();  
 Log.*d*(**"fileLog"**, **"копирование-"**+i+**" НЕ выполнено"**);  
 }  
 } *//конец if(f1.isFile())* } *//Конец цикла по всем элементам списка* } *//Конец метода onCreate  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 Процедура для копирования файлов из указанного каталога в указанный каталог  
Входные параметры:  
 File sourceFile - файл источник (что копируется)  
 File destFile - файл приемник (куда копируется)  
Выходные параметры:  
 Нет выходных параметров  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* **public static void** copyFile(File sourceFile, File destFile) **throws** IOException {  
 *//если путь к целевому файлу НЕ существует, то* **if** (!destFile.getParentFile().exists())  
 *//создать назначенный путь со всеми промежуточными папками* destFile.getParentFile().mkdirs();  
 *//если НЕ существует файл с указанным именем, то* **if** (!destFile.exists()) {  
 *//создать пустой файл с указанным именем* destFile.createNewFile();  
 }  
 *//Определить каналы чтения/записи* FileChannel source = **null**;  
 FileChannel destination = **null**;  
  
 **try** {  
 *//Создать канал, связанный с файлом источником* source = **new** FileInputStream(sourceFile).getChannel();  
 *//Создать канал, связанный с файлом приемником* destination = **new** FileOutputStream(destFile).getChannel();  
 *//Начать передачу данных по каналу приемника из канала источника  
 //и передать данные в количестве полного размера канала источника* destination.transferFrom(source, 0, source.size());  
 } **finally** {  
 *//После окончания передачи - закрыть каналы, если они сами не закрылись* **if** (source != **null**) {  
 source.close();  
 }  
 **if** (destination != **null**) {  
 destination.close();  
 }  
 }*//конец finally* }*//конец процедуры copyFile*}*//конец класса MainActivity*

