

## **Разработка файлового браузера на ОС Android**

**Автор:** Александров Евгений, 5 курс.

**Руководитель:** Ухов Владимир Иосифович, кандидат физико-математических наук.

**Образовательное учреждение:** Государственное бюджетное образовательное учреждение высшего профессионального образования Московской области Международный университет природы, общества и человека «Дубна» филиал «Протвино».

### **DEVELOPMENT OF THE FILE BROWSER ON THE ANDROID OS Alexandrov E.**

Так как мировое Android сообщество расширяется то и программных продуктов становится с каждым днем все больше и больше. Ежедневно количество пользователей Android растет на 4,4 %. И если в 2010 году показатель активаций составлял 160 тыс. ежедневно, то в июле 2011 года, по заявлению Эрика Шмидта, он уже достиг 550 тыс. новых устройств, активируемых каждый день, что говорит о том, что ОС стремительно развивается в нише продуктов для мобильных устройств. Так же по данным на февраль 2011 года устройства с ОС Android уверенно лидируют в списке самых популярных смартфонов Великобритании, оставив позади iPhone 4 с Apple iOS. По прогнозам экспертов это первый шаг ОС Android к глобальному лидерству на рынке мобильных операционных систем. Разрабатываемый продукт вносит свою маленькую песчинку в огромное количество приложений для Android. В данном курсовом проекте необходимо разработать файловый браузер так как в ОС android по умолчанию он отсутствует.

#### **Основные понятия Android приложений**

Каждая программа представляет собой пакет с расширением \*.apk. Этот пакет формируется утилитой aapt и включает в себя весь скомпилированный Java код и все ресурсы приложения. В Android каждому приложению присваивается отдельный процесс. Процесс создается только тогда когда это необходимо, а когда необходимые действия были выполнены - Android автоматически завершает процесс освобождая память для других приложений. Помимо этого, каждое приложение выполняется в отдельной виртуальной машине, а также имеет уникальный Linux User ID. Соответственно у каждого приложения есть права присвоенные его uID. По умолчанию приложение видит лишь свои ресурсы, однако существуют пути их экспорта другим приложениям. Также существует возможность двум приложениям иметь один uID, в таком случае ресурсы оказываются общими для обоих приложений. Одним из самых важных понятий в Android является то, что приложения могут вызывать части других приложений, если имеют на это право.

Для реализации такого подхода пришлось отказаться от классической схемы приложения с одной точкой входа. Говоря другими словами - Android приложение строится таким образом, чтобы каждый отдельный его компонент мог вызываться отдельно при необходимости.

Приложения в Android используют четыре основных компонента:

- **Activities** - это визуальный компонент приложения, который отвечает за UI. Например - список контактов. Этот список является Activity. Приложение может содержать как несколько Activity, так и вовсе не содержать;
- **Services** - это то что выполняется пока приложение не находится в фокусе. Пример - проигрыватель музыки. Сервис может запускаться вместе с системой и работать в фоновом режиме выполняя некоторые действия. Взаимодействие с Service осуществляется средствами интерфейса;
- **Broadcast receivers** - один из важнейших компонентов приложения. Broadcast receiver также как и сервис не имеет видимого интерфейса. Его задача отслеживать определенные действия или системные сообщения и реагировать на это. Но так как сам по себе Broadcast receiver сделать ничего не может - он только передает сигнал другим компонентам;
- **Content providers** - предоставляет определенные данные другим приложениям. Эти данные могут храниться в файловой системе, SQLite базе данных и т.д..

Так же в состав всех приложений входит файл `AndroidManifest.xml`. В этом файле описывается вся структура приложения, будь то Activity, Service или Broadcast receive.

В главном классе было создано несколько функций которые взаимодействуют с интерфейсом наполняют его и получают вызовы. Рассмотрим их далее:

- Функция скрытия верхнего бара и как следствие освобождение места на экране:

```
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
    WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

- Функция навигации по директории и получения списка файлов

```
private void browseTo(final File aDirectory) {  
    if (aDirectory.isDirectory()) {  
        this.currentDirectory = aDirectory;  
        fill(aDirectory.listFiles());  
    }  
}
```

- Функция формирования сообщения (названия папок и файлов)

```
TextView titleManager = (TextView) findViewById(R.id.titleManager);
```

- При открытии файла создаем окно выбора, в зависимости от ответа переходим на функции вызова либо продолжаем работу.

```

        titleManager.setText(aDirectory.getAbsolutePath());
new AlertDialog.Builder(this)

        .setTitle("Подтверждение")
        .setMessage("Хотите открыть файл "+ aDirectory.getName() + ".?")
        .setPositiveButton("Да", okButtonListener)
        .setNegativeButton("Нет", cancelButtonListener)

• Описываем поднятие по директории на интерфейсе
if(selectedFileString.equals("../")){

    this.upOneLevel();

• Создадим функцию “поднятия”
private void upOneLevel(){
    if(this.currentDirectory.getParent() != null) {

        this.browseTo(this.currentDirectory.getParentFile());

• Функция “обновления” списка файлов
private void fill(File[] files) {

    this.directoryEntries.clear();

    if (this.currentDirectory.getParent() != null)

        this.directoryEntries.add("../");

for (File file : files) {

    this.directoryEntries.add(file.getAbsolutePath());

}    .show

```

### Список используемой литературы

1. Сатия Коматинени, Дэйв Маклин, Саид Хашими «Pro Android 3»,., ISBN 5-85582-109-9, формат 70x100/16 (167x236 мм), мягкий переплет, 2011; Вильямс
2. Лорен Дэрсси, Шейн Кондер «Android: Application Development In 24 Hours»,., ISBN 978-5-4252-0318-2, формат 70x100/16 (167x236 мм), мягкий переплет, 2011; Вильямс
3. <http://ru.wikipedia.org/>
4. <http://www.realcoding.net/>
5. <http://developer.android.com/>