

# Разработка мобильных приложений

Новосибирский национальный  
исследовательский  
государственный университет (НГУ)

## Лекция 4

Чиркунов Кирилл

[cyril.chirkunov@computer.org](mailto:cyril.chirkunov@computer.org)  
[vk.com/chirkunov](https://vk.com/chirkunov)

---

# На прошлой лекции

- Views (GroupView, TextView, Spinner, EditView and etc.)
- AbsoluteLayout, FrameLayout, LinearLayout, TableLayout, RelativeLayout
- Оптимизация разметки
- Drawables & Menus

# Сегодня

- Intents & Receivers
- Broadcast messaging
- Data Adapters
- Preferences & Settings & Files

# Намерения (Intents)

Применяются для:

- объявления о запуске Activity или Service, чтобы выполнить какие-либо действия на основе некоторых данных
- передачи уведомлений о том, что произошло событие
- явного запуска определенного Service или Activity

# Код результата закрытия Activity

- Вызов из родительской активности:

***startActivityForResult(Intent i, int call\_Identifier)***

- Выставление результата в дочерней активности:

***setResult(int result, Intent i)***

- Коды возврата:

- ▣ ***Activity.RESULT\_OK***

- ▣ ***Activity.RESULT\_CANCELED*** (по умолчанию)

- Другие, более специализированные:

- ▣ ***RESULT\_FIRST\_USER***

- ▣ ***RESULT\_COMMON\_ERROR***

- ▣ и др.

# Код результата закрытия Activity

@Override

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    switch(requestCode) {  
        case (SHOW_SUB_ACTIVITY_ONE) : {  
            if (resultCode == Activity.RESULT_OK) {  
                Uri horse = data.getData();  
                boolean inputCorrect = data.getBooleanExtra(IS_INPUT_CORRECT, false);  
                String selectedPistol = data.getStringExtra(SELECTED_PISTOL);  
            }  
            break;  
        }  
        case (SHOW_SUB_ACTIVITY_TWO) : {  
            if (resultCode == Activity.RESULT_OK) {  
                // TODO: Handle OK click.  
            }  
            break;  
        }  
    }  
}
```

# Стандартные действия (часть)

ACTION_VIEW	ACTION_SEND	ACTION_MAIN
ACTION_DIAL	ACTION_SENDTO	ACTION_CALL
ACTION_EDIT	ACTION_ANSWER	ACTION_WEB_SEARCH
ACTION_SEARCH	ACTION_INSERT	ACTION_DELETE

# Стандартные категории (не все)

CATEGORY_DEFAULT	CATEGORY_SELECTED_ALTERNATIVE
CATEGORY_BROWSABLE	CATEGORY_LAUNCHER
CATEGORY_GADGET	CATEGORY_HOME
CATEGORY_ALTERNATIVE	CATEGORY_PREFERENCE



# Данные намерения

## Имя хоста

- Google.com
- Microsoft.com
- и т.д.

## Тип данных (MIME type, case sensitive)

- image/jpeg
- audio/mpeg4-generic
- vnd. android.cursor.dir/\*
- и т.д.

## Путь

- Доступные значения для пути (/transport/ports)

## Порт

- Доступные порты для заданного сервера

## Схема данных

- http
- Content
- и др.

## Extra

- ...

# Утверждение намерений

**Система собирает  
список всех доступных  
Intent-Filters из  
установленных пакетов**



**Фильтры, которые не  
соответствуют  
действию или  
категории Intent,  
удаляются**



**Когда есть несколько  
совпадений,  
пользователю  
предлагается выбрать  
Activity из списка**



**Каждая часть пути URI  
из намерения  
сравнивается с  
разделом data  
фильтра Intent-Filter**

# Делегирование намерений

```
Intent intent = getIntent();  
... //some code  
startNextMatchingActivity(intent);
```

# Выбор контакта из списка (демо)

# Расширяемость меню

```
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    Intent intent = new Intent();
    intent.setData(MoonBaseProvider.CONTENT_URI);
    intent.addCategory(Intent.CATEGORY_SELECTED_ALTERNATIVE);
    int menuGroup = 0;
    int menuItemId = 0;
    int menuItemOrder = Menu.NONE;
    ComponentName caller = getComponentName();
    Intent[] specificIntents = null;
    // The menu items created from the previous Intents
    // will populate this array.
    MenuItem[] outSpecificItems = null;
    // Set any optional flags.
    int flags = Menu.FLAG_APPEND_TO_GROUP;
    // Populate the menu
    menu.addIntentOptions(menuGroup, menuItemId, menuItemOrder, caller, specificIntents,
        intent,
        flags,
        outSpecificItems);

    return true;
}
```

## Linkify

Определение через код:

```
TextView textView = (TextView) findViewById (R.id.myTextView);  
Linkify.addLinks (textView, Linkify.WEB_URLS | Linkify.EMAIL_ADDRESSES);
```

Определение через xml-файл:

```
<TextView  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:text="@string/linkify_me"  
    android:autoLink="phone | email"  
/>
```

Определение через регулярные выражения:

```
int flags = Pattern.CASE_INSENSITIVE;  
Pattern p = Pattern.compile("\ \bquake[0-9]*\ \b", flags);  
Linkify.addLinks(myTextView, p,  
    "content://com.paad.earthquake/earthquakes/");
```

## Linkify MatchModifier & TransformFilter

```
class MyMatchFilter implements MatchFilter {  
    public boolean acceptMatch(CharSequence s, int start, int  
    end) {  
        return (start == 0 || s.charAt(start-1) != '!');  
    }  
}
```

```
class MyTransformFilter implements TransformFilter {  
    public String transformUrl(Matcher match, String url) {  
        return url.toLowerCase();  
    }  
}
```

```
Linkify.addLinks(myTextView, pattern, prefixWith,  
    new MyMatchFilter(), new MyTransformFilter());
```

# Регистрация широковещательного приемника

Через manifest.xml:

```
<receiver android:name=".LifeformDetectedBroadcastReceiver">  
  <intent-filter>  
    <action android:name="com.paad.action.NEW_LIFEFORM"/>  
  </intent-filter>  
</receiver>
```

Через код:

```
IntentFilter filter = new IntentFilter(NEW_LIFEFORM_DETECTED);  
LifeformDetectedBroadcastReceiver r = new  
LifeformDetectedBroadcastReceiver();  
registerReceiver(r, filter);  
...  
unregisterReceiver(receiver);
```



```
public class LifeformDetectedBroadcastReceiver extends BroadcastReceiver {

    public static final String BURN = "com.paad.alien.action.BURN_IT_WITH_FIRE";

    @Override
    public void onReceive(Context context, Intent intent) {
        // Get the lifeform details from the intent.
        Uri data = intent.getData();
        String type = intent.getStringExtra("type");
        double lat = intent.getDoubleExtra("latitude", 0);
        double lng = intent.getDoubleExtra("longitude", 0);
        Location loc = new Location("gps");
        loc.setLatitude(lat);
        loc.setLongitude(lng);

        if (type.equals("alien")) {
            Intent startIntent = new Intent(BURN, data);
            startIntent.putExtra("latitude", lat);
            startIntent.putExtra("longitude", lng);

            context.startActivity(startIntent);
        }
    }
}
```

# Продвинутое оповещение

- Упорядоченное оповещение:
  - `sendOrderedBroadcast(intent, null)`
  - `sendOrderedBroadcast(intent, null, myBroadcastReceiver, null, Activity.RESULT_OK, null, null);`
- «Липкие» трансляции  
(требуется полномочие `BROADCAST_STICKY`)
  - `sendStickyBroadcast`
  - `removeStickyBroadcast`

## Ожидающие намерения (Pending Intents)

// Start an Activity

```
Intent startActivityIntent = new Intent(this, MyOtherActivity.class);  
PendingIntent.getActivity(this, 0, startActivityIntent, 0);
```

// Broadcast an Intent

```
Intent broadcastIntent = new  
Intent(NEW_LIFEFORM_DETECTED);  
PendingIntent.getBroadcast(this, 0, broadcastIntent, 0);
```

# Адаптеры

Интерфейс **Adapter**

Наследники:

- ArrayAdapter<Type t>
- SimpleCursorAdapter
- BaseAdapter, CursorAdapter, HeaderViewListAdapter, ListAdapter, ResourceCursorAdapter, SimpleAdapter, SpinnerAdapter, WrapperListAdapter
- Пользовательские адаптеры поддерживающие привязку к представлениям, наследникам абстрактного класса AdapterView

```
public class MyArrayAdapter extends ArrayAdapter<MyClass> {
    int resource;
    public ToDoItemAdapter(Context context, int resource, List<MyClass> items) {
        super(context, resource, items);
        resource = _resource;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LinearLayout newView;
        MyClass classInstance = getItem(position);
        // TODO Retrieve values to display from the
        // classInstance variable.
        // Inflate a new view if this is not an update.
        if (convertView == null) {
            newView = new LinearLayout(getContext());
            String inflater = Context.LAYOUT_INFLATER_SERVICE;
            LayoutInflater vi = (LayoutInflater)getContext().getSystemService(inflater);
            vi.inflate(resource, newView, true);
        } else {
            newView = (LinearLayout)convertView;
        }
        // TODO Retrieve the Views to populate
        // TODO Populate the Views with object property values.
        return newView;
    }
}
```

# Использование адаптеров

```
ArrayList<String> myStringArray = new ArrayList<String>();
ArrayAdapter<String> myAdapterInstance;

int layoutID = android.R.layout.simple_list_item_1;
myAdapterInstance = new ArrayAdapter<String>(this, layoutID , myStringArray);

myListView.setAdapter(myAdapterInstance);

String uriString = "content://contacts/people/";
Cursor myCursor = managedQuery(Uri.parse(uriString), null, null, null);

String[] fromColumns = new String[] {People.NUMBER, People.NAME};

int[] toLayoutIDs = new int[] { R.id.nameTextView, R.id.numberTextView};

SimpleCursorAdapter myAdapter;
myAdapter = new SimpleCursorAdapter(this,
                                   R.layout.simplecursorlayout,
                                   myCursor,
                                   fromColumns,
                                   toLayoutIDs);

myListView.setAdapter(myAdapter);
```

# ИСПОЛЬЗОВАНИЕ АДАПТЕРОВ

```
String uriString = "content://contacts/people/";
Cursor myCursor = managedQuery(Uri.parse(uriString), null, null, null);

String[] fromColumns = new String[] {People.NUMBER, People.NAME};

int[] toLayoutIDs = new int[] { R.id.nameTextView,
R.id.numberTextView};

SimpleCursorAdapter myAdapter;
myAdapter = new SimpleCursorAdapter(this,
                                   R.layout.simplecursorlayout,
                                   myCursor,
                                   fromColumns,
                                   toLayoutIDs);

myListView.setAdapter(myAdapter);
```

# Internet resources

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

```
String myFeed = getString(R.string.my_feed);
try {
    URL url = new URL(myFeed);

    URLConnection connection = url.openConnection();
    HttpURLConnection httpConnection = (HttpURLConnection)connection;

    int responseCode = httpConnection.getResponseCode();
    if (responseCode == HttpURLConnection.HTTP_OK) {
        InputStream in = httpConnection.getInputStream();
        [ ... Process the input stream as required ... ]
    }
}
catch (MalformedURLException e) {}
catch (IOException e) {}
```



# Диалоговые окна

- **Класс Dialog и его производные**

Специальные диалоговые окна:

- *AlertDialog*
- *CharacterPickerDialog*
- *DatePickerDialog*
- *TimePickerDialog*
- *ProgressDialog*

- **Activity в модальном режиме**

```
<activity android:name="MyDialogActivity"  
android:theme="@android:style/Theme.Dialog">  
</activity>
```

```
static final private int TIME_DIALOG = 1;
```

```
@Override
```

```
public Dialog onCreateDialog(int id) {  
    switch(id) {  
        case (TIME_DIALOG) :  
            AlertDialog.Builder timeDialog = new AlertDialog.Builder(this);  
            timeDialog.setTitle("The Current Time Is...");  
            timeDialog.setMessage("Now");  
            return timeDialog.create();  
        }  
        return null;  
    }  
}
```

```
@Override
```

```
public void onPrepareDialog(int id, Dialog dialog) {  
    switch(id) {  
        case (TIME_DIALOG) :  
            SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss");  
            Date currentTime = new Date(java.lang.System.currentTimeMillis());  
            String dateString = sdf.format(currentTime);  
            AlertDialog timeDialog = (AlertDialog)dialog;  
            timeDialog.setMessage(dateString);  
            break;  
        }  
    }  
}
```

# Просмотр данных о землетрясениях (демо)

## Работа с файлами

```
String FILE_NAME = "tempfile.tmp";
```

```
// Create a new output file stream that's private to this application.  
FileOutputStream fos = openFileOutput(FILE_NAME, Context.MODE_PRIVATE);  
// Create a new file input stream.  
FileInputStream fis = openFileInput(FILE_NAME);
```

Типы режимов доступа к файлам

- MODE\_APPEND
- MODE\_PRIVATE
- MODE\_WORLD\_READABLE
- MODE\_WORLD\_WRITEABLE

Доступ к файлам из ресурсов:

```
Resources myResources = getResources();  
InputStream myFile = myResources.openRawResource(R.raw.myfileName)
```

Другие операции: `deleteFile`, `fileList` (также см. `java.io.File`)

# Настройки

- Общие (SharedPreferences)
- Состояние приложения/активности (Bundle)

Поддерживаемые типы:

- boolean
- string
- float
- long
- integer

# Общие настройки

```
// Get the stored preferences
int mode = Activity.MODE_PRIVATE;
final String MY_PREFS = "MY_PREFS";
SharedPreferences mySharedPreferences = getSharedPreferences(MY_PREFS, mode);

// Retrieve the saved values.
boolean isTrue = mySharedPreferences.getBoolean("isTrue", false);
float lastFloat = mySharedPreferences.getFloat("lastFloat", 0f);
int wholeNumber = mySharedPreferences.getInt("wholeNumber", 1);
long aNumber = mySharedPreferences.getLong("aNumber", 0);
String stringPreference = mySharedPreferences.getString("textEntryValue", "");
...
SharedPreferences.Editor editor = mySharedPreferences.edit();

// Store new primitive types in the shared preferences object.
editor.putBoolean("isTrue", true);
editor.putFloat("lastFloat", 1f);
editor.putInt("wholeNumber", 2);
editor.putLong("aNumber", 31);
editor.putString("textEntryValue", "Not Empty");

// Commit the changes.
editor.commit();
```

```

<PreferenceScreen
    xmlns:android="http://schemas.android.com/apk/res/android">
    <PreferenceCategory android:title="@string/inline_preferences">
        <CheckBoxPreference
            android:key="checkbox_preference"
            android:title="@string/title_checkbox_preference"
            android:summary="@string/summary_checkbox_preference" />
    </PreferenceCategory>
    <PreferenceCategory
        android:title="@string/dialog_based_preferences">
        <EditTextPreference android:key="edittext_preference"
android:title="@string/title_edittext_preference"
android:summary="@string/summary_edittext_preference"
android:dialogTitle="@string/dialog_title_edittext_preference" />
        <ListPreference android:key="list_preference" android:title="@string/title_list_preference"
android:summary="@string/summary_list_preference"
android:entries="@array/entries_list_preference"
android:entryValues="@array/entryvalues_list_preference"
android:dialogTitle="@string/dialog_title_list_preference" />
    </PreferenceCategory>
    <PreferenceCategory android:title="@string/launch_preferences">
        <PreferenceScreen
            android:fragment="com.example.android.apis.preference.PreferenceWithHeaders$Prefs1Frag
mentInner" android:title="@string/title_fragment_preference"
            android:summary="@string/summary_fragment_preference">
            <extra android:name="someKey" android:value="somePrefValue" />
        </PreferenceScreen>
    </PreferenceCategory>
</PreferenceScreen>

```

## Работа с PreferenceActivity

```
public class MySettings extends PreferenceActivity {
    SharedPreferences prefs;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.preferences);
        prefs = this.getSharedPreferences("MySettings",
            Activity.MODE_PRIVATE);
    }
}
```

```
<activity android:name=".MySettings"
          android:label="My Settings">
    ...
</activity>
```

```
Intent i = new intent(this, MySettings.class);
startActivityForResult(i, SHOW_PREFERENCES);
```



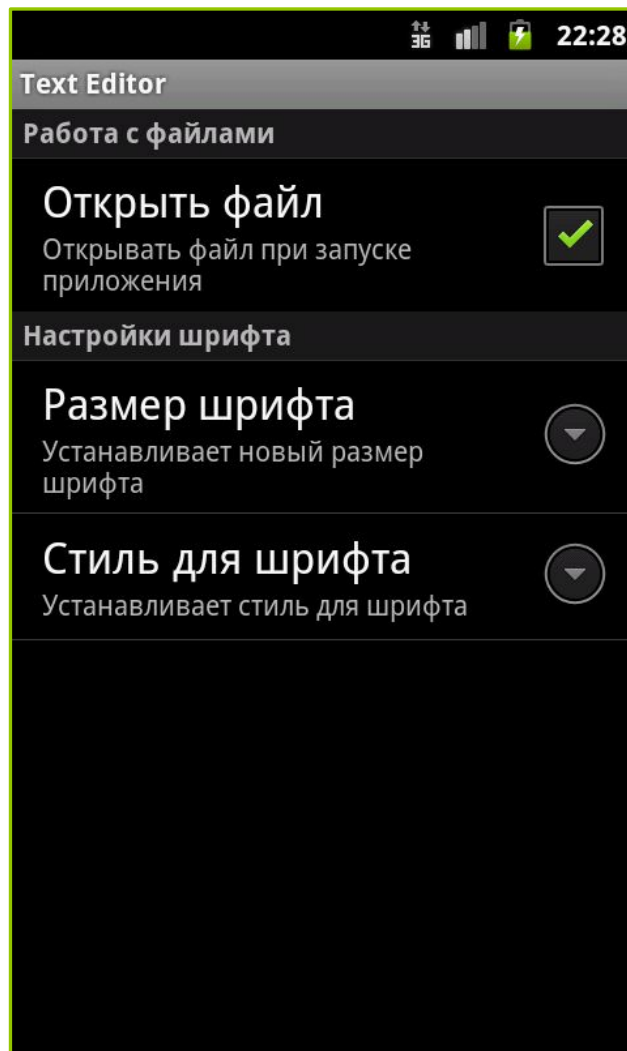
## Работа с общими настройками

```
Context context = getApplicationContext();
SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(context);
// TODO: use get<preference type> methods
```

```
public class MyActivity extends Activity implements
OnSharedPreferenceChangeListener {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        // Register this OnSharedPreferenceChangeListener
        Context context = getApplicationContext();
        SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(context);
        prefs.registerOnSharedPreferenceChangeListener(this);
    }

    public void onSharedPreferenceChanged(SharedPreferences prefs, String key) {
        // TODO Check the shared preference and key parameters and change UI or
        // behavior as appropriate.
    }
}
```

# Пример PreferenceActivity



## На следующей лекции

- SQLite и источники данных
- Геокодирование & Карты
- Геолокационные сервисы

# Задание «Записная книжка»



**Разработать приложение с графическим пользовательским интерфейсом, поддерживающее**

- создание,
- редактирование,
- удаление,
- поиск

заметок.

Два варианта хранения заметок:

- А) в базе SQLite;
- Б) с использованием файловой системы.

**Срок выполнения: 2 недели**

