Эргономика мобильных приложений

Преподаватель: **Меженная Марина Михайловна**

к.т.н., доцент,
доцент кафедры инженерной психологии и эргономики
а 609-2
mezhennaya@bsuir.by

Лекция 5:
Всплывающие сообщения. Логирование.
Явный и неявный вызов нового Activity.

1. Всплывающие сообщения (класс Toast).

2. Логирование (Log).

3. Явный вызов нового Activity. Intent.

4. Неявный вызов нового Activity. Intent Filter.

Всплывающие сообщения

Синтаксис: **Toast.makeText(context, text , duration).show();**

Статический метод makeText создает View-элемент Toast.
Параметры метода:
- **context** – объект, который предоставляет доступ к базовым функциям приложения, т.к. Activity является подклассом Context, то в качестве объекта от Context используем текущую Activity, т.е. this.
- **text** – текст, который надо показать.
- **duration** – продолжительность показа (Toast.LENGTH_LONG – длинная (3,5 сек), Toast.LENGTH_SHORT – короткая (2 сек)).

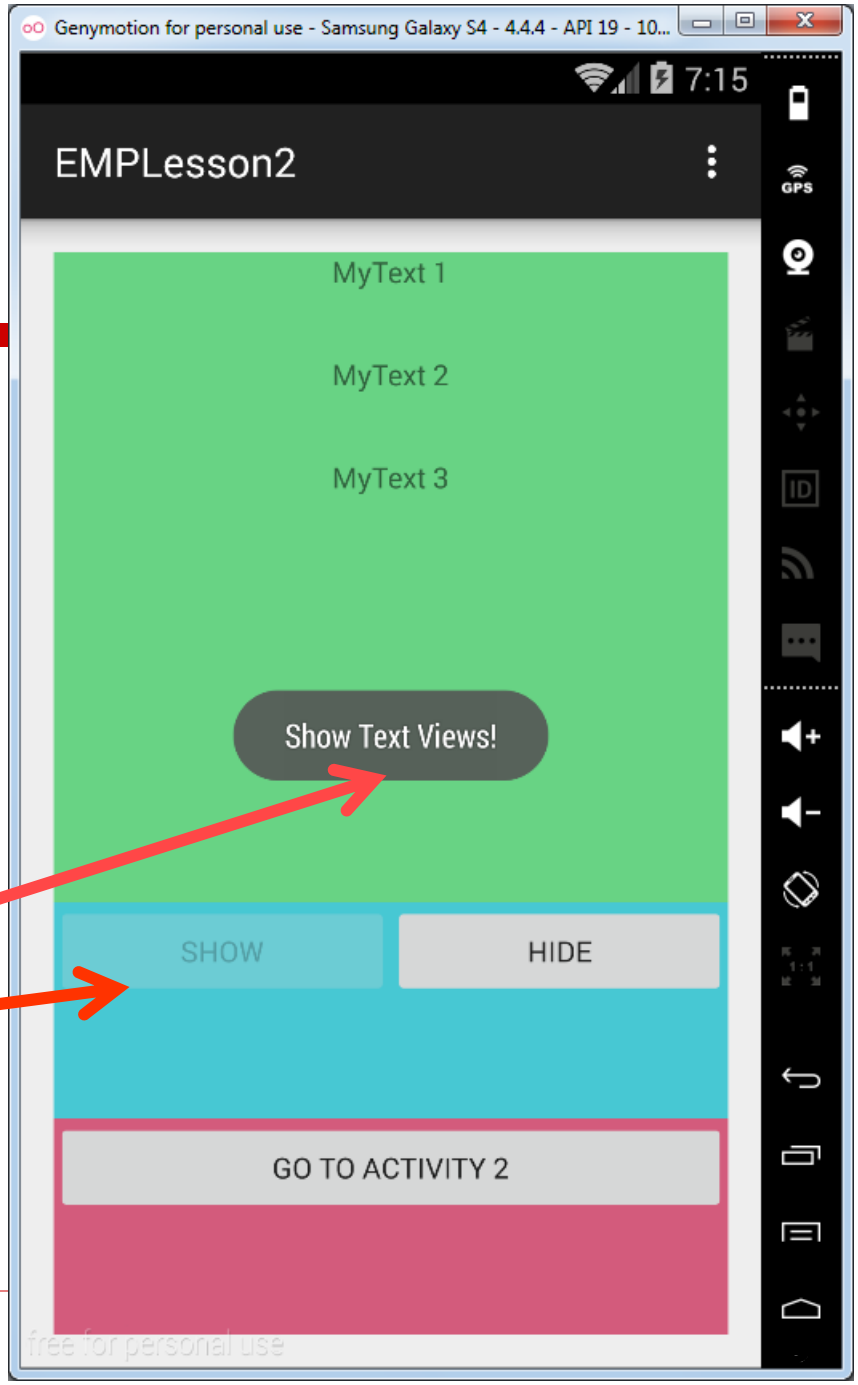
Чтобы Toast отобразился на экране, вызывается метод show().

Всплывающие сообщения

Toast.makeText (this,
"Show Text Views!",
Toast.LENGTH_SHORT).show();

Всплывающие сообщения

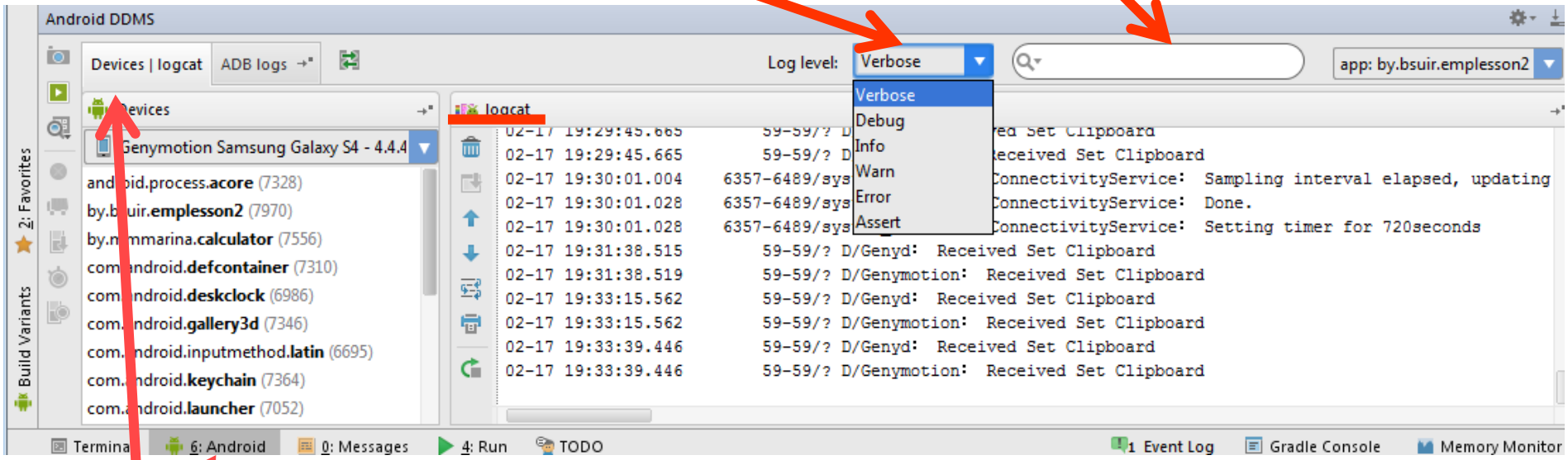
```
Toast toast = Toast.makeText(this,
"Show Text Views!",
Toast.LENGTH_SHORT);

toast.setGravity(Gravity.CENTER, 0, 0);

toast.show();
```



Логи

Фильтр для поиска логов по тэгу

Уровни логов по степени важности



Вкладки для отображения окна logcat

Логи

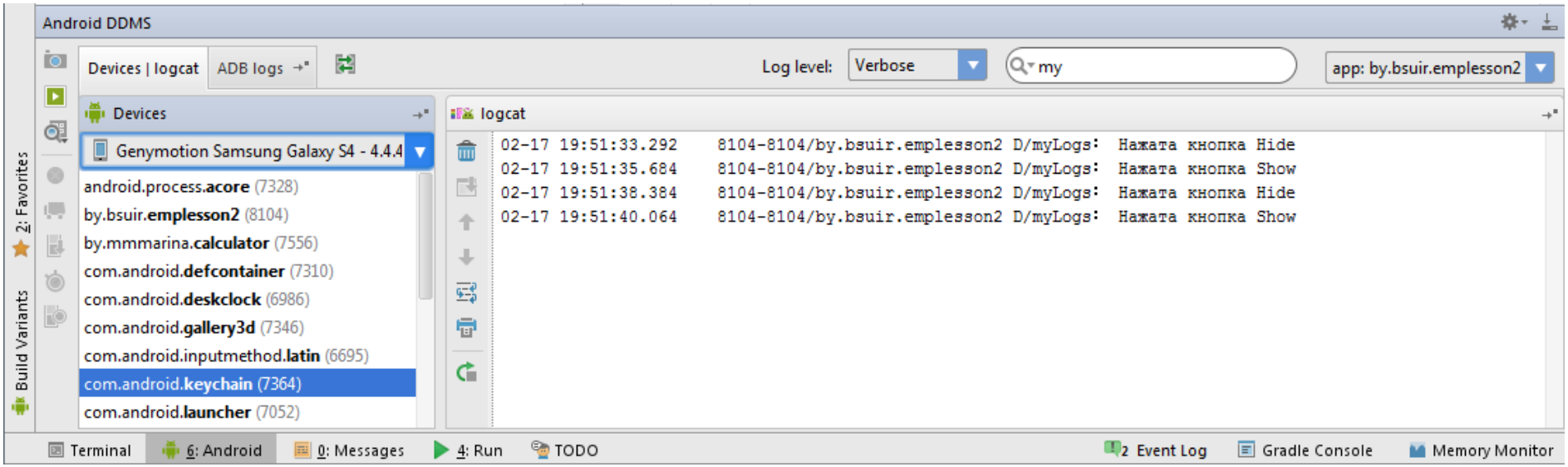
Класс Log и его методы:
Log.v(),
Log.d(),
Log.i(),
Log.w(),
Log.e().

Синтаксис методов: Log.d(tag, text).

Логи

```
private static final String TAG = "myLogs";

…

 public void onClick(View v) {
        switch (v.getId()) {
           case R.id.btnShow:
                Log.d(TAG, "Нажата кнопка Show");
                break;
           case R.id.btnHide:
                Log.d(TAG, "Нажата кнопка Hide");
                break;
        }
}
```

Логи


Android DDMS
Devices | logcat
ADB logs
Log level:
Verbose
my
app: by.bsuir.emplesson2
Devices
Genymotion Samsung Galaxy S4 - 4.4.4
android.process.acore (7328)
by.bsuir.emplesson2 (8104)
by.mmmarina.calculator (7556)
com.android.defcontainer (7310)
com.android.deskclock (6986)
com.android.gallery3d (7346)
com.android.inputmethod.latin (6695)
com.android.keychain (7364)
com.android.launcher (7052)
logcat
02-17 19:51:33.292 8104-8104/by.bsuir.emplesson2 D/myLogs: Нажата кнопка Hide
02-17 19:51:35.684 8104-8104/by.bsuir.emplesson2 D/myLogs: Нажата кнопка Show
02-17 19:51:38.384 8104-8104/by.bsuir.emplesson2 D/myLogs: Нажата кнопка Hide
02-17 19:51:40.064 8104-8104/by.bsuir.emplesson2 D/myLogs: Нажата кнопка Show
2: Favorites
Build Variants
Terminal
6: Android
0: Messages
4: Run
TODO
2 Event Log
Gradle Console
Memory Monitor

Быстрый способ убрать логи после отладки

```
public class L {

private static final boolean TRIGGER = true;
private static final String TAG = "myLogs";

        public static void d(String message) {
            if (TRIGGER) {
                Log.d(TAG, message);
            }
        }
}
```

Быстрый способ убрать логи после отладки

```
//private static final String TAG = "myLogs";

…

 public void onClick(View v) {
        switch (v.getId()) {
           case R.id.btnShow:
               L.d("Нажата кнопка Show");
               break;
           case R.id.btnHide:
               L.d("Нажата кнопка Hide");
               break;
        }
}
```

Явный вызов нового Activity

1.Создаем новый xml файл разметки (например, activity_second.xml).
2.Создаем новый java-класс под второе Activity (например, SecondActivity).
3.Прописываем логику SecondActivity.java (наследуемся от суперкласса, подключаем разметку).

```
public class SecondActivity extends ActionBarActivity{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
    }
}
```

Явный вызов нового Activity

4.Регистрируем новое Activity в Манифесте:

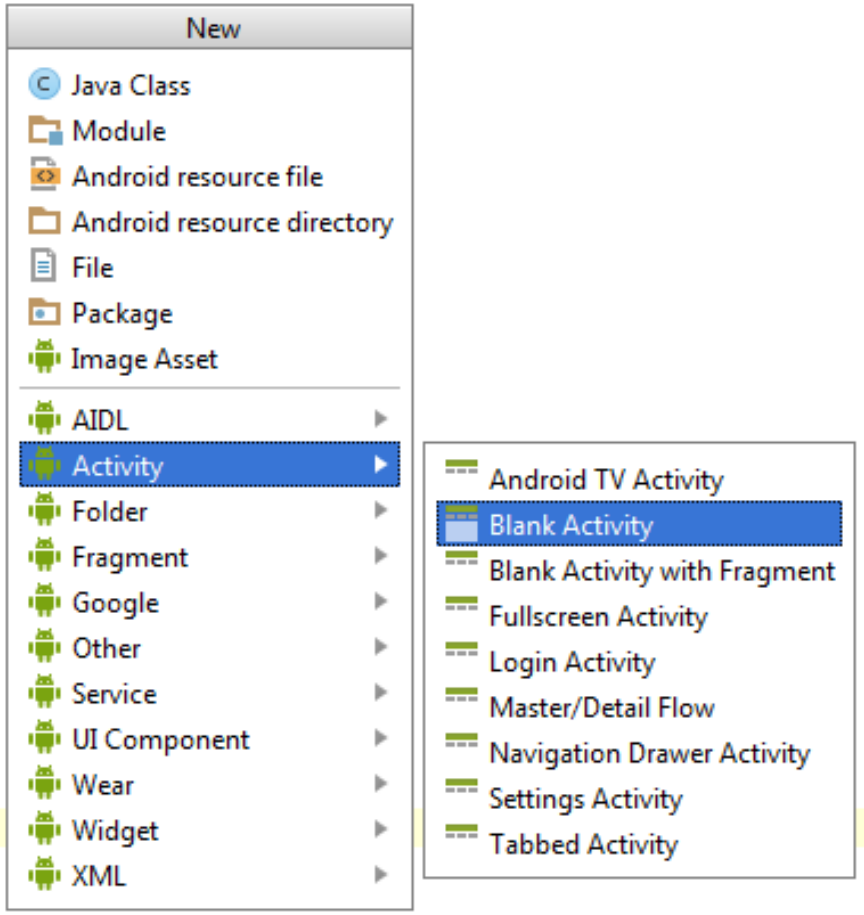
```
<activity
    android:name=".MainActivity"
    android:label="@string/app_name">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

```
<activity
    android:name=".SecondActivity"
    android:label="@string/app_name">
</activity>
```

Явный вызов нового Activity

Программный способ:

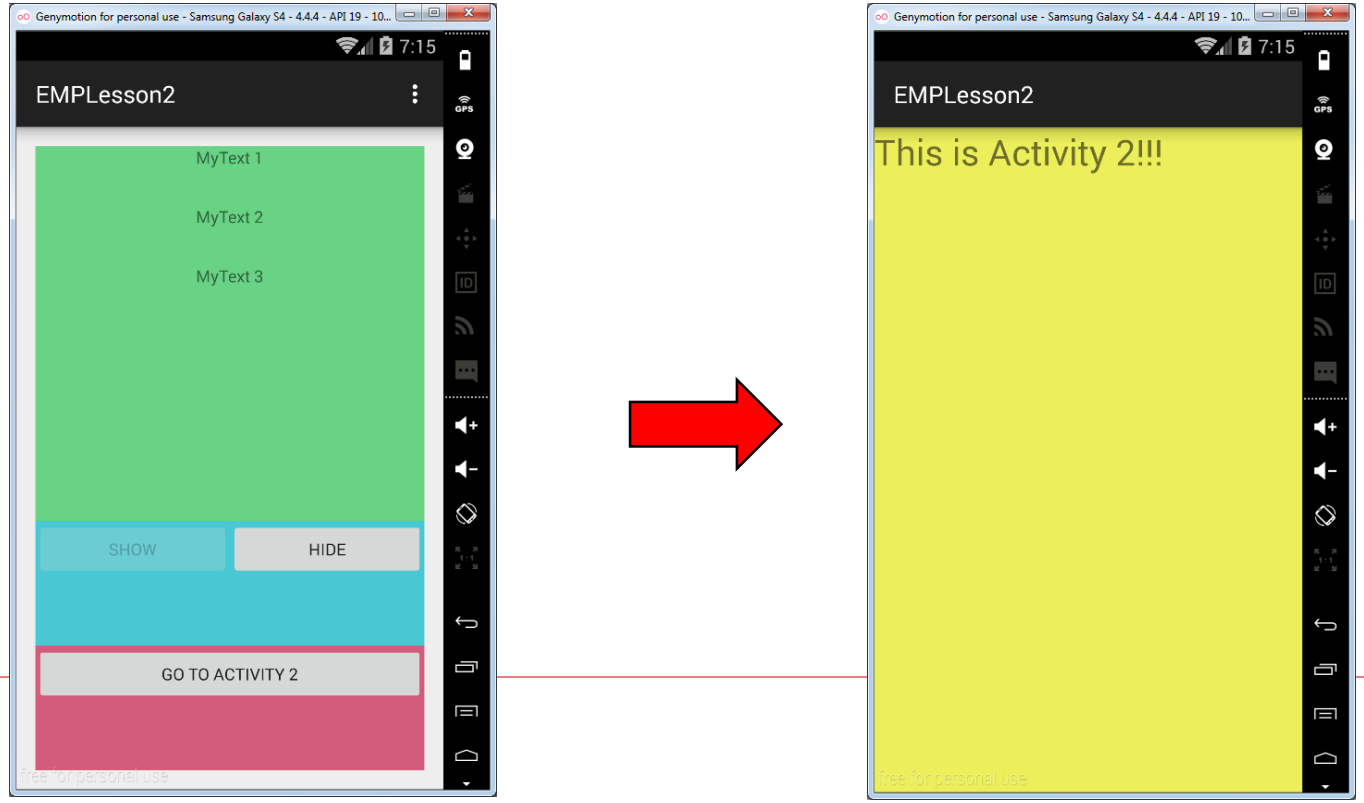
Alt + Insert →



Явный вызов нового Activity

5.Реализуем вызов нового Activity из основного:

```
Intent intent = new Intent(this, SecondActivity.class);
startActivity(intent);
```



Явный вызов нового Activity

Намерение (Intent) - это механизм для описания одной операции - выбрать фотографию, отправить письмо, сделать звонок, запустить браузер и перейти по указанному адресу.

В Android-приложениях многие операции работают через намерения.

Наиболее распространенный сценарий использования намерения - запуск другой активности в своём приложении.

Явный вызов нового Activity

С помощью класса Intent мы **явно** указываем какое Activity хотели бы увидеть (это обычно используется внутри одного приложения):

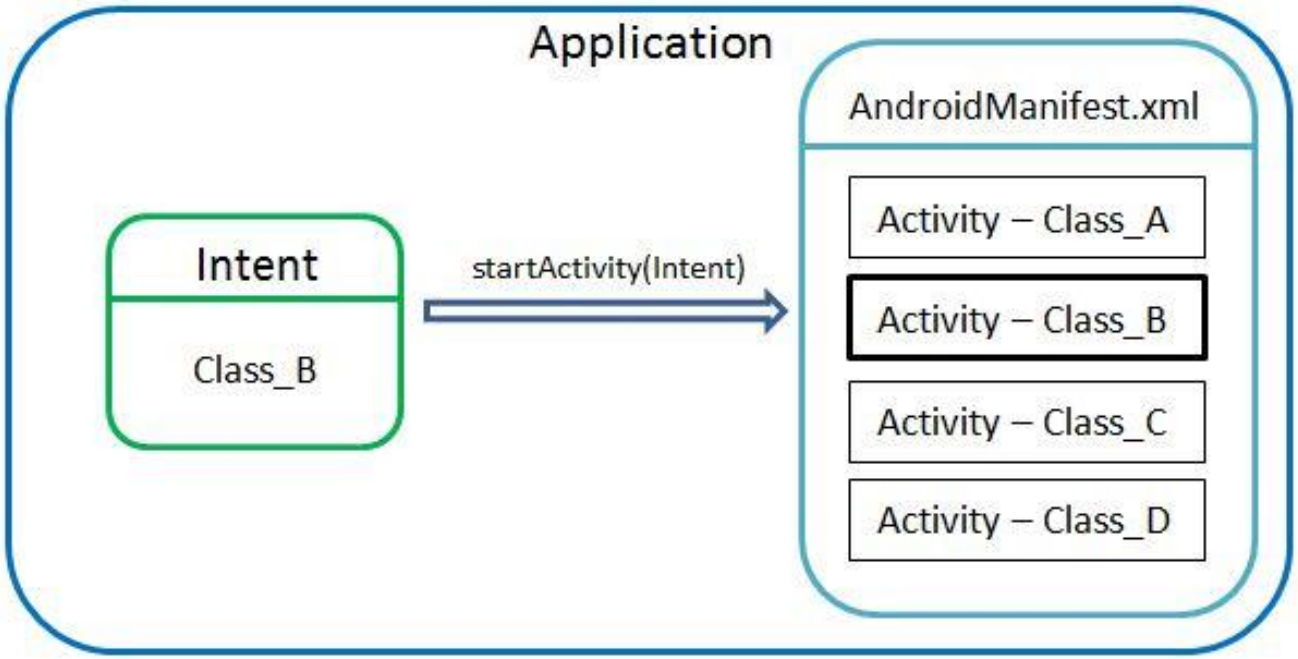
Intent (Context packageContext, Class cls)

Context – это объект, который предоставляет доступ к базовым функциям приложения таким как: доступ к ресурсам, к файловой системе, вызов Activiy и т.д.

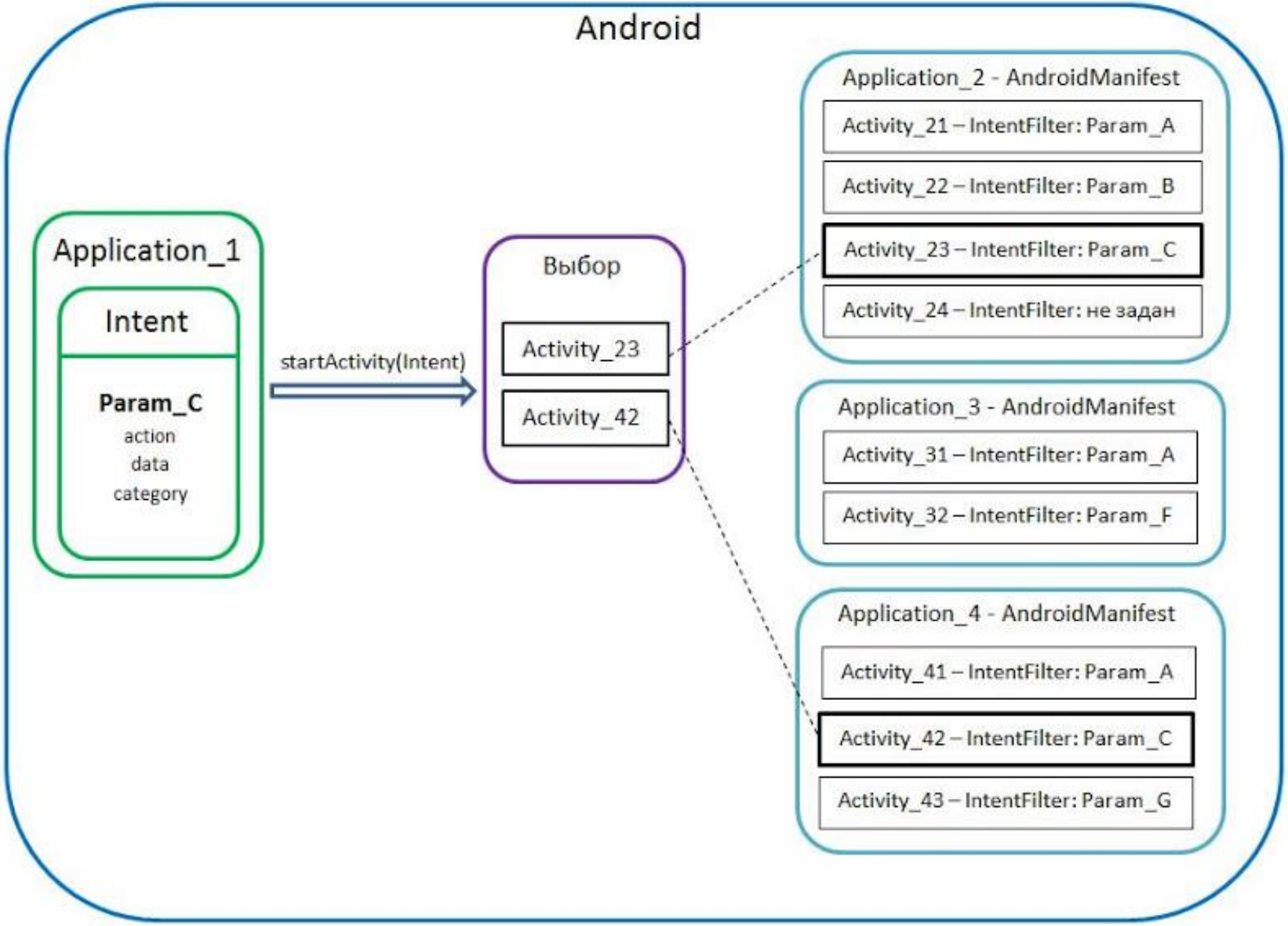
Class – имя класса, указанное в манифест-файле.

Явный вызов нового Activity

Создаем Intent, в качестве параметра передаем ему класс Class_B. Далее вызываем метод startActivity с созданным Intent в качестве параметра. Метод проверяет AndroidManifest на наличие Activity связанной с классом Class_B и если находит, то отображает. Все это в пределах одного приложения.



Неявный вызов нового Activity



Неявный вызов нового Activity

```
<activity
    android:name=".SecondActivity"
    android:label="@string/app_name">
    <intent-filter>
      <action android:name="android.intent.action.SecondActivity" />
      <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

```
intent = new Intent("android.intent.action.SecondActivity");
startActivity(intent);
```