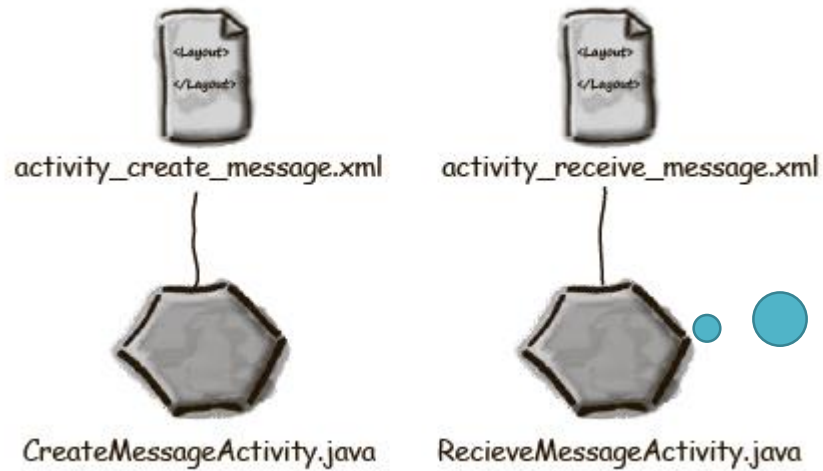


Android Development for Beginners

Лекция 3
Множественные Activity, Intent,
Жизненный цикл Activity

Каждая активность должна быть объявлена



Если информация обо мне не включена в AndroidManifest.xml, то с точки зрения системы я не существую и никогда не буду выполняться.

```
<application
...
...>
<activity
    android:name="имя_класса_активности"
    android:label="@string/метка_активности"
    ...
...>
...
</activity>
...
</application>
```

Каждая активность должна быть объявлена в элементе <application>.

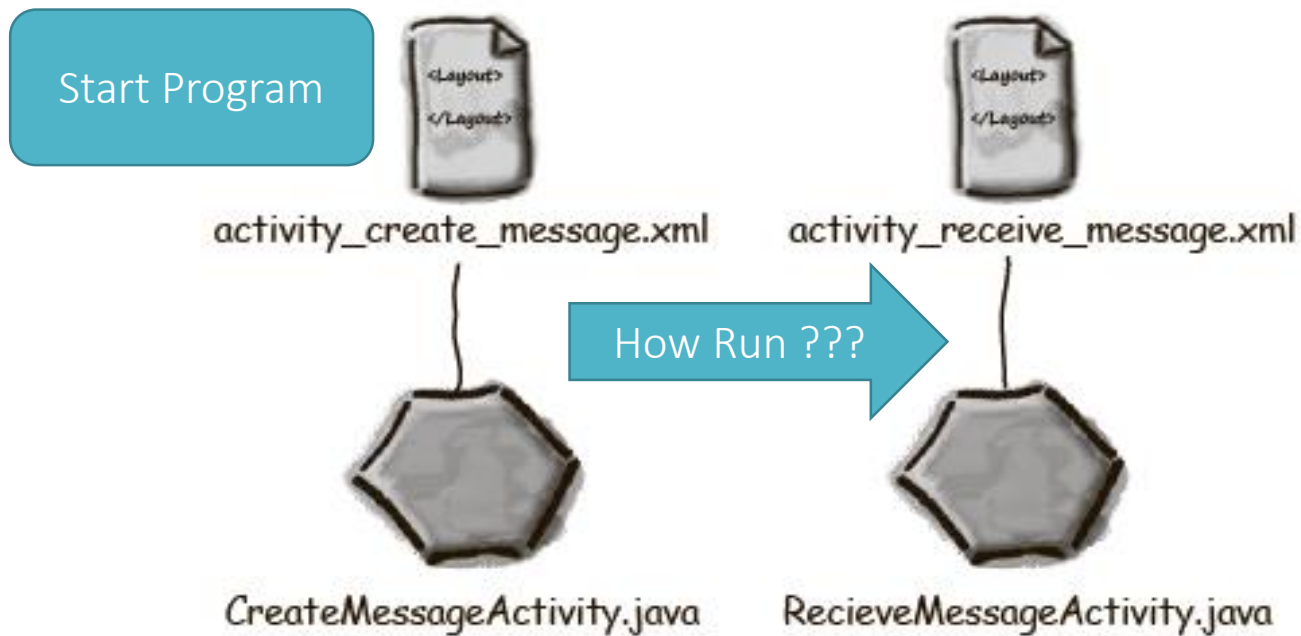
Эта строка обязательна.

Эта строка не обязательна, но Android Studio генерирует ее за нас.

Активность также может обладать другими свойствами.



Интент — разновидность сообщения



Интент Это разновидность сообщений, позволяющая связать разнородные объекты (например, активности) на стадии выполнения. (Если одна активность хочет запустить другую, она отправляет для этого интент системе Android. Android запускает вторую активность и передает ей интент).



Интент — разновидность сообщения

```
public void onSendMessage(View view) {  
    .....  
    .....  
    Intent intent = new Intent(this, ReceiveMessageActivity.class);  
    .....  
    startActivity(intent);  
}
```

new Intent (Context packageContext, Class cls)

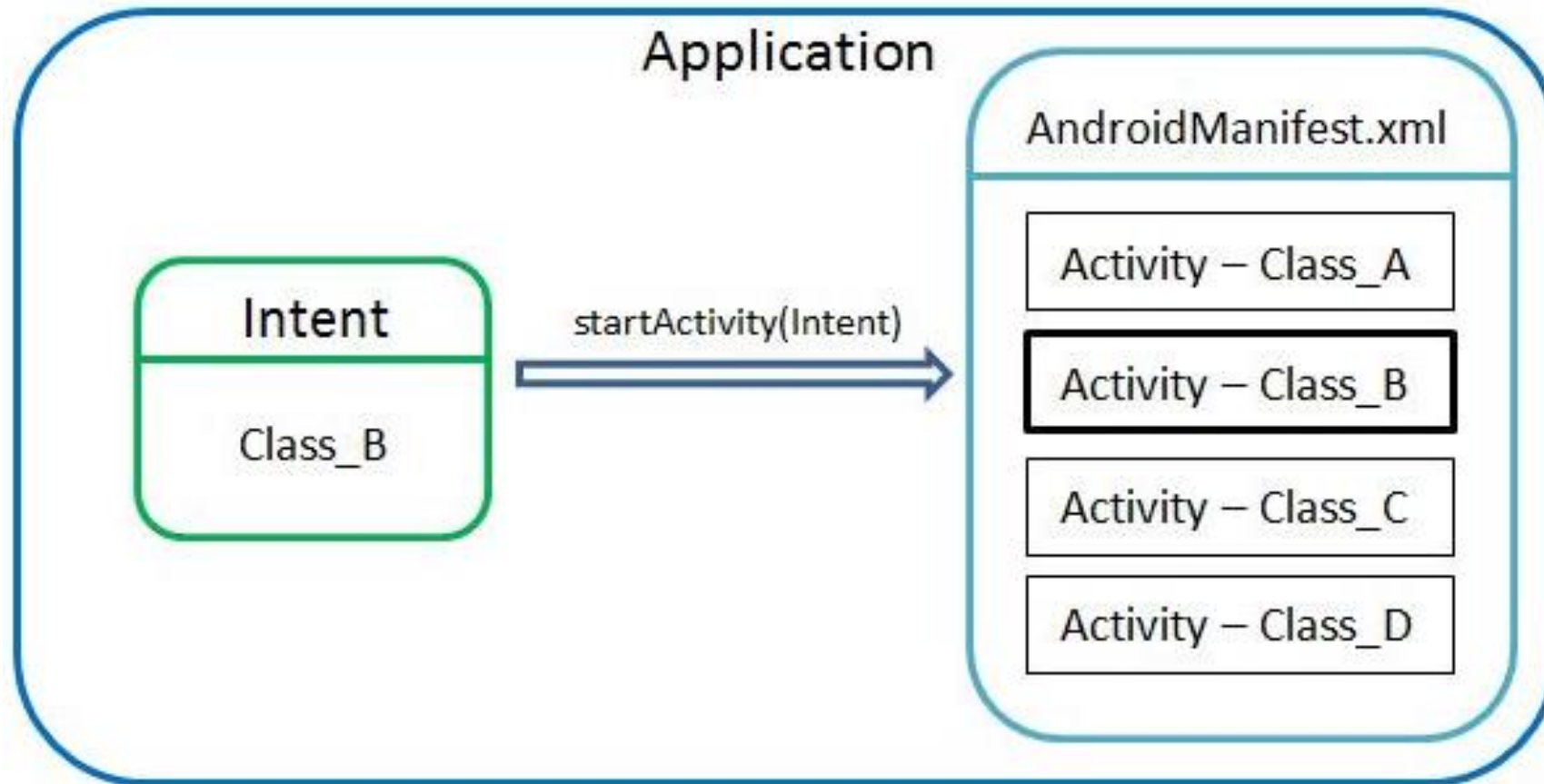
Context — это объект, который **предоставляет** доступ к **базовым функциям** приложения таким как: доступ к ресурсам, к файловой системе, вызов Activity и т.д (**Activity** является **подклассом Context**)

Второй параметр — **имя класса** Activity которое нужно запустить (Если указан в Manifest)



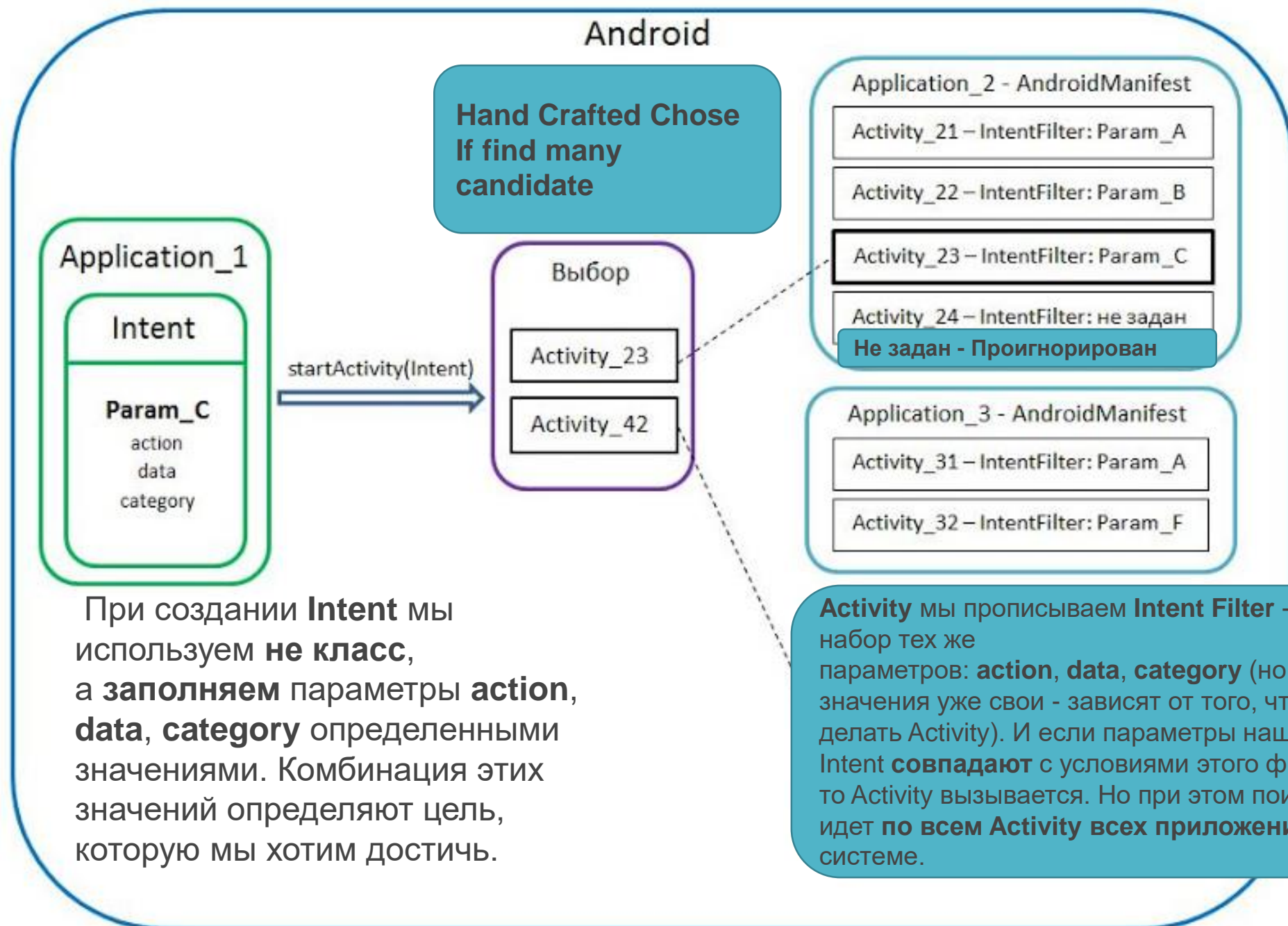
Интент — ЯВНЫЙ ВЫЗОВ

Вызов Activity с помощью такого Intent – это **ЯВНЫЙ** вызов. Т.е. с помощью класса мы **ЯВНО** указываем какое Activity хотели бы увидеть. Это обычно используется **внутри одного приложения**





Интент — Не Явный вызов



Intent — IntentFilter

```
public class MainActivity extends Activity implements OnClickListener {

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button btnTime = (Button) findViewById(R.id.btnTime);
        Button btnDate = (Button) findViewById(R.id.btnDate);

        btnTime.setOnClickListener(this);
        btnDate.setOnClickListener(this);
    }

    @Override
    public void onClick(View v) {
        Intent intent;

        switch(v.getId()) {
            case R.id.btnTime:
                intent = new Intent("ru.startandroid.intent.action.showtime");
                startActivity(intent);
                break;
            case R.id.btnDate:
                intent = new Intent("ru.startandroid.intent.action.showdate");
                startActivity(intent);
                break;
        }
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="ru.startandroid"
    <uses-sdk android:minSdkVersion="10"></uses-sdk>
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".MainActivity" android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"></action>
                <category android:name="android.intent.category.LAUNCHER"></category>
            </intent-filter>
        </activity>
        <activity android:name="ActivityTime">
            <intent-filter>
                <action android:name="ru.startandroid.intent.action.showtime"></action>
                <category android:name="android.intent.category.DEFAULT"></category>
            </intent-filter>
        </activity>
        <activity android:name="ActivityDate" android:label="Date basic">
            <intent-filter>
                <action android:name="ru.startandroid.intent.action.showdate"></action>
                <category android:name="android.intent.category.DEFAULT"></category>
            </intent-filter>
        </activity>
        <activity class=".NoteEditor" android:label="@string/title_note">
            <intent-filter android:label="@string/resolve_edit">
                <action android:name="android.intent.action.VIEW" />
                <action android:name="android.intent.action.EDIT" />
                <category android:name="android.intent.category.DEFAULT" />
                <data android:mimeType="vnd.android.cursor.item/vnd.google.note" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```



Intent — IntentFilter

Intent ([String](#) action)

Intent ([String](#) action, [Uri](#) uri)

Intent ([Context](#) packageContext, [Class](#)<?> cls)

SW
Ca

```
intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://developer.android.com"));
startActivity(intent);
break;
case R.id.btnMap:
    intent = new Intent();
    intent.setAction(Intent.ACTION_VIEW);
    intent.setData(Uri.parse("geo:55.754283,37.62002"));
    startActivity(intent);
    break;
case R.id.btnCall:
    intent = new Intent(Intent.ACTION_DIAL);
    intent.setData(Uri.parse("tel:12345"));
    startActivity(intent);
    break;
```

- ACTION_ANSWER — Открывает активность, которая связана с входящими звонками. Это действие обрабатывается стандартным экраном для приема звонков;
- ACTION_CALL — инициализирует обращение по телефону;
- ACTION_DELETE — Запускает активность, с помощью которой можно удалить данные, указанные в пути URI внутри намерения;
- ACTION_EDIT — Отображает данные для редактирования пользователем;
- ACTION_VIEW — Наиболее распространенное общее действие. Для данных, передаваемых с помощью пути URI в намерении, ищется наиболее подходящий способ вывода. В

```
<activity class=".NoteEditor" android:label="@string/title_note">
    <intent-filter android:label="@string/resolve_edit">
        <action android:name="android.intent.action.VIEW" />
        <action android:name="android.intent.action.EDIT" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="vnd.android.cursor.item/vnd.google.note" />
    </intent-filter>
</activity>
```

```
public class ReceiveMessageActivity extends Activity {

    public static final String EXTRA_MESSAGE = "message";

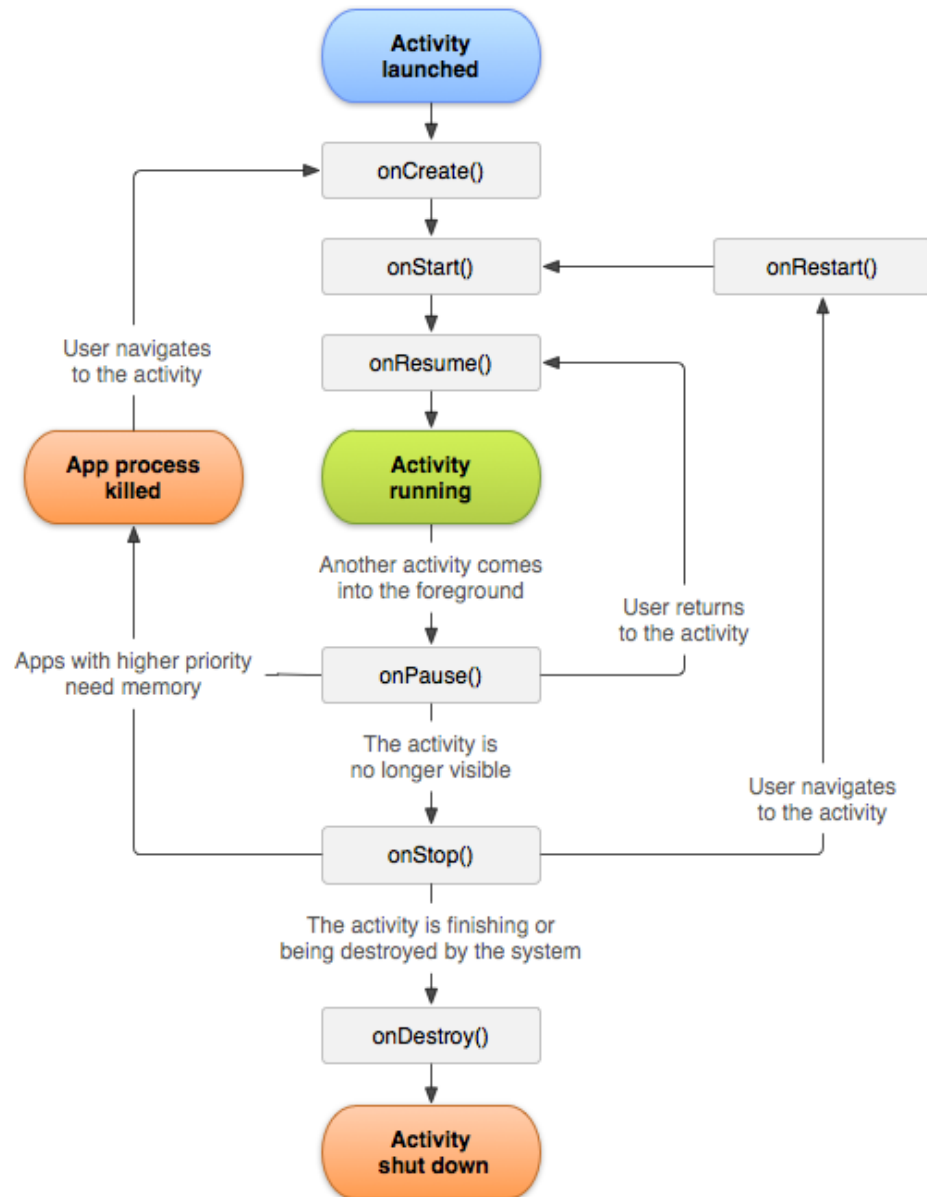
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_receive_message);
        Intent intent = getIntent();
        String messageText = intent.getStringExtra(EXTRA_MESSAGE);
        TextView messageView = (TextView) findViewById(R.id.message);
        messageView.setText(messageText);
    }
}
```

Имя дополнительного значения,
передаваемого в интеннте.

Получить интеннт
и извлечь из него
сообщение вызовом
getStringExtra().

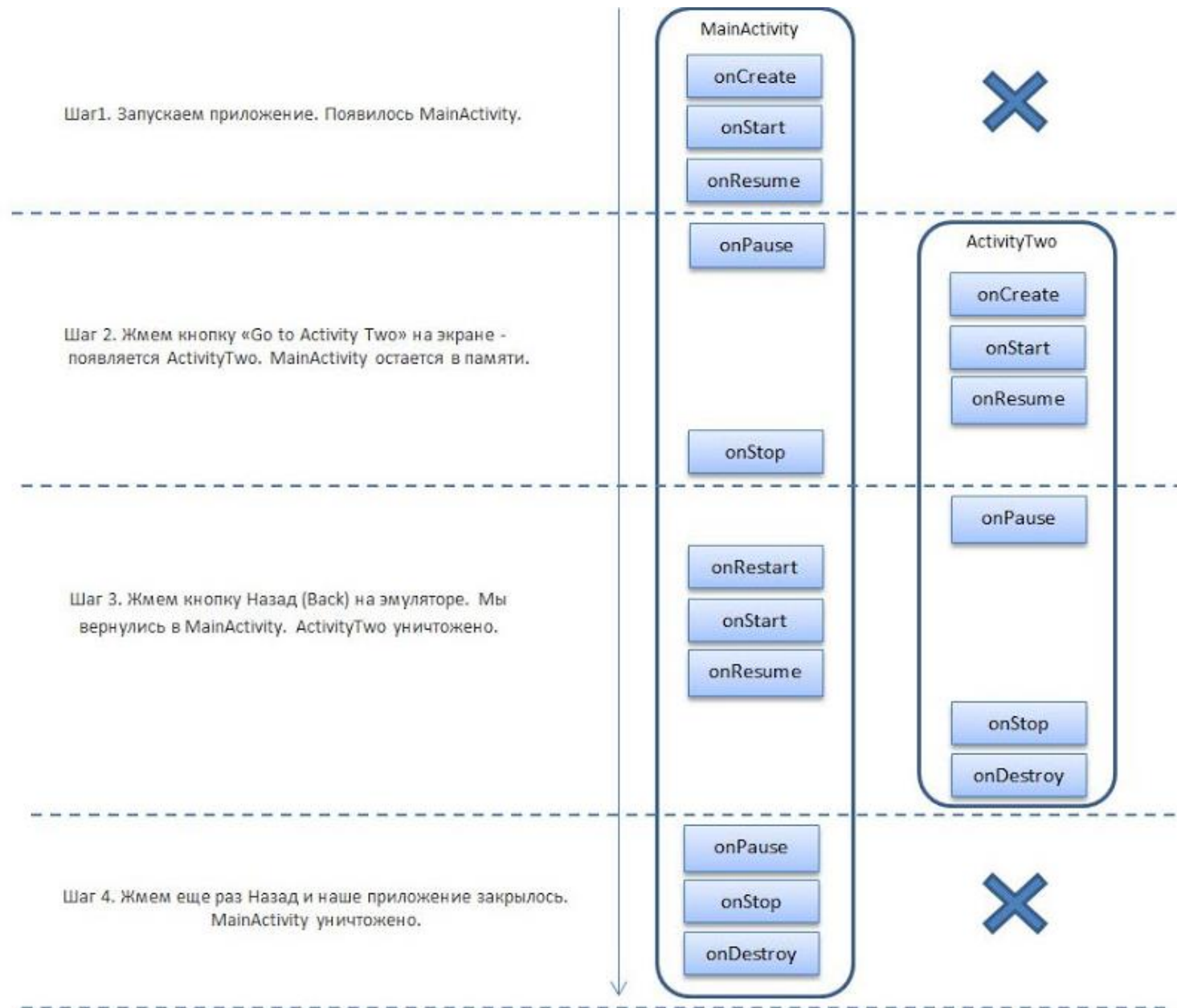
Добавить текст в надпись
с идентификатором message.

Activity Lifecycle.



```
public class Activity extends ApplicationContext {  
    protected void onCreate(Bundle savedInstanceState);  
  
    protected void onStart();  
  
    protected void onRestart();  
  
    protected void onResume();  
  
    protected void onPause();  
  
    protected void onStop();  
  
    protected void onDestroy();  
}
```

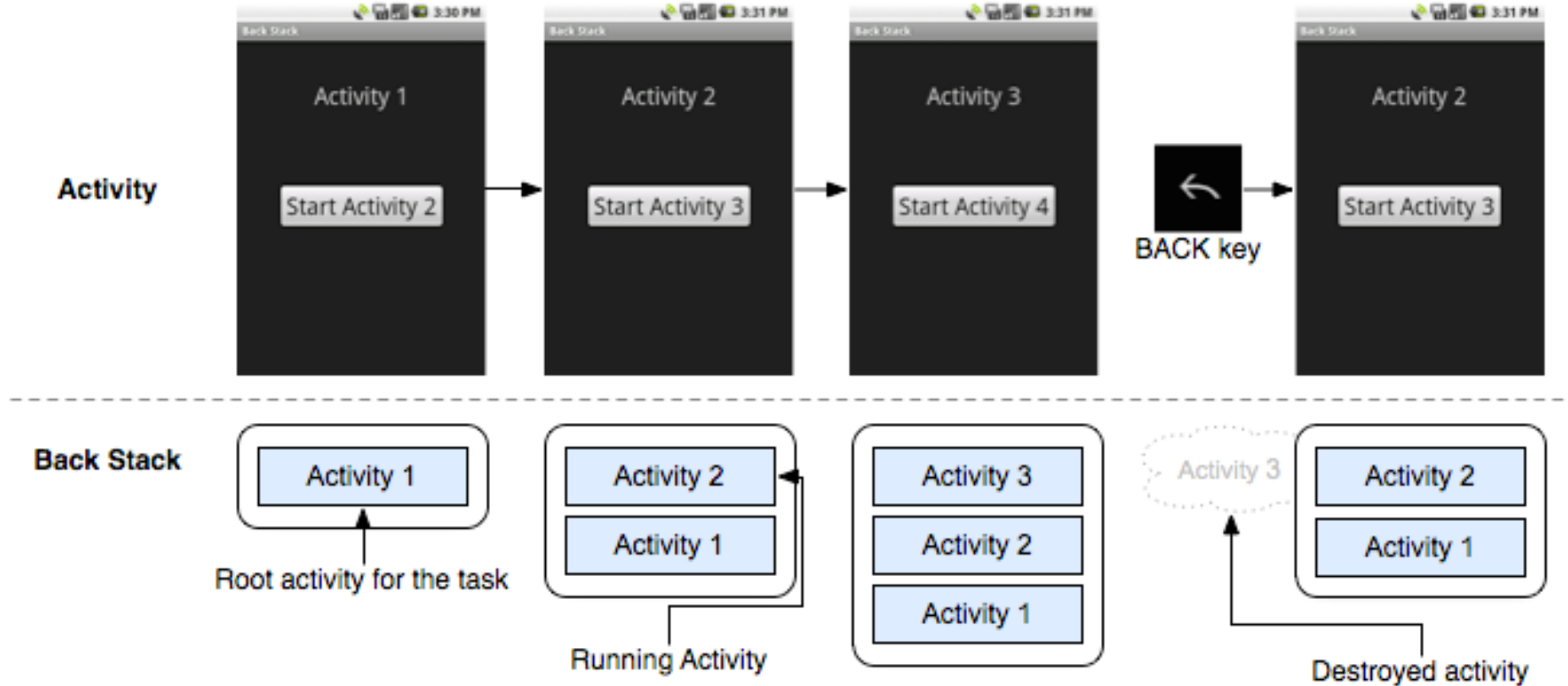
Activity Lifecycle.



TASK - STACK ACTIVITY

Активность – одна целенаправленная операция, которая может выполняться пользователем. Если объединить несколько активностей для выполнения чего-то более сложного, получится задание (**TASK**)

В итоге список аналогий
браузера и Android таков:
Браузер – Android
Вкладка с историей
посещений – Task
Страница – Activity
Ссылка – Intent



TASK - STACK ACTIVITY

Task – группа из нескольких Activity, с помощью которых пользователь выполняет определенную операцию. Обычно стартовая позиция для создания Task – это экран Домой (Home).

Находясь в Home вы вызываете какое-либо приложение из списка приложений или через ярлык. Создается Task. И Activity приложения (которое отмечено как MAIN в манифест-файле) помещается в этот Task как корневое. Task выходит на передний фон. Если же при вызове приложения, система обнаружила, что в фоне уже существует Task, соответствующий этому приложению, то она выведет его на передний план и создавать ничего не будет.

Когда Activity_A вызывает Activity_B, то Activity_B помещается на верх (в топ) Task и получает фокус. Activity_A остается в Task, но находится в состоянии Stopped (его не видно и оно не в фокусе). Далее, если пользователь жмет Back находясь в Activity_B, то Activity_B удаляется из Task и уничтожается. А Activity_A оказывается теперь на верху Task и получает фокус.

В каком порядке открывались (добавлялись в Task) Activity, в таком порядке они и содержатся в Task. Они никак специально не сортируются и не упорядочиваются внутри. Набор Activity в Task еще называют back stack.

Задание и полезные ссылки

- Создать приложение из нескольких активностей
- По нажатию кнопки открывать разные активности
- Дописать в ваш калькулятор Activity для результата.
- По нажатию на кнопку запустить внешнее приложение(например Google Maps)
- Создать загрузочный экран. Вызывать его лишь один раз.

Полезные ссылки:

<https://developer.android.com/guide/index.html>

<http://javarush.ru/>

<http://startandroid.ru/>

<http://developer.alexanderklimov.ru/android/>