Лекция №3.
по курсу «Мобильное программирование»

Москва 2019

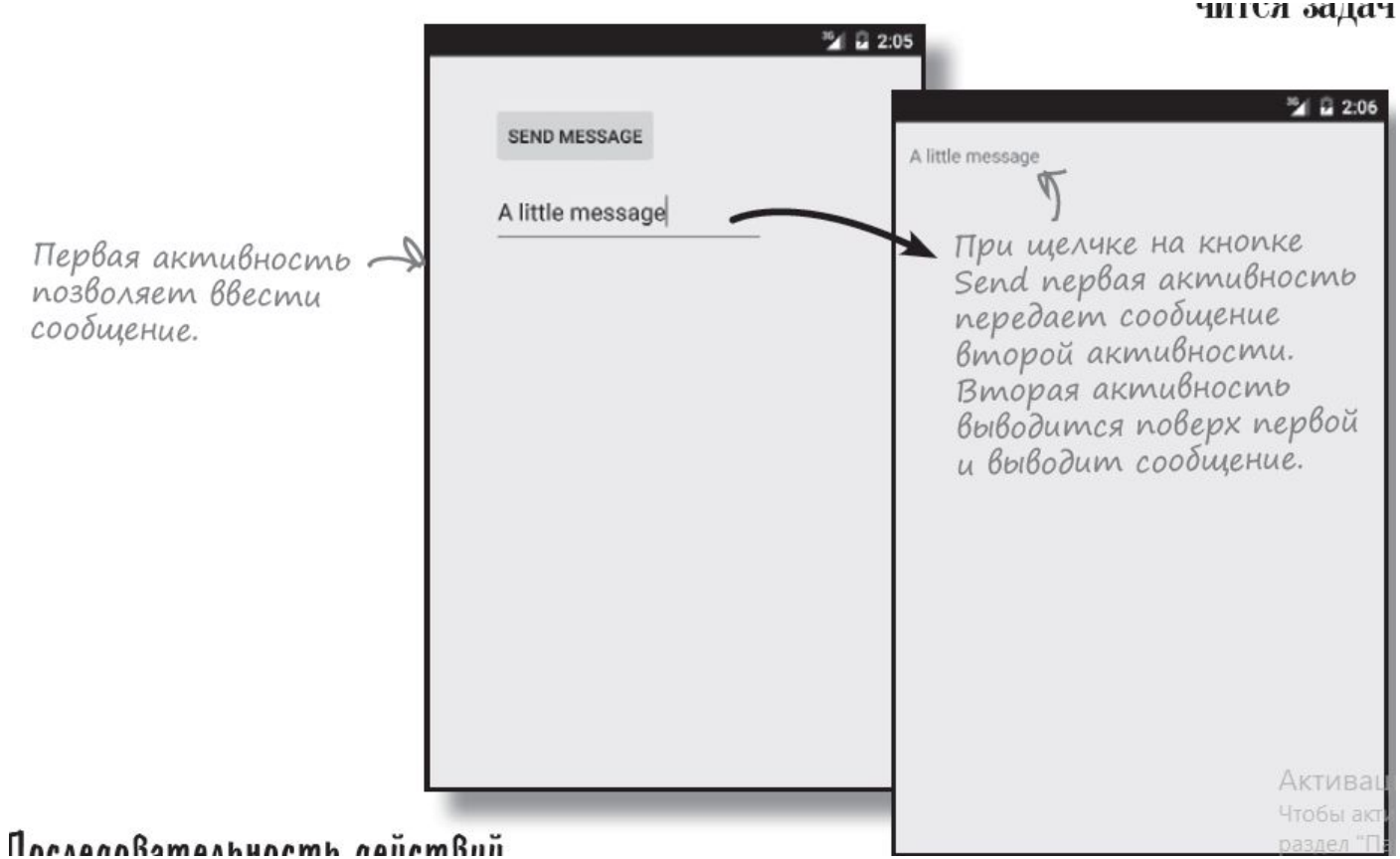
ДЗ. Программа «Угадай число»

Программа генерирует случайным образом число. Если введенное число больше чем сгенерированное число , то выводится «Больше», если меньше, то «Меньше». Если введенное число совпадает с сгенерированным, то выводится равно.

НЕСКОЛЬКО АКТИВНОСТЕЙ

Активность (отвечает за взаимодействие пользователя с информцией на экране)

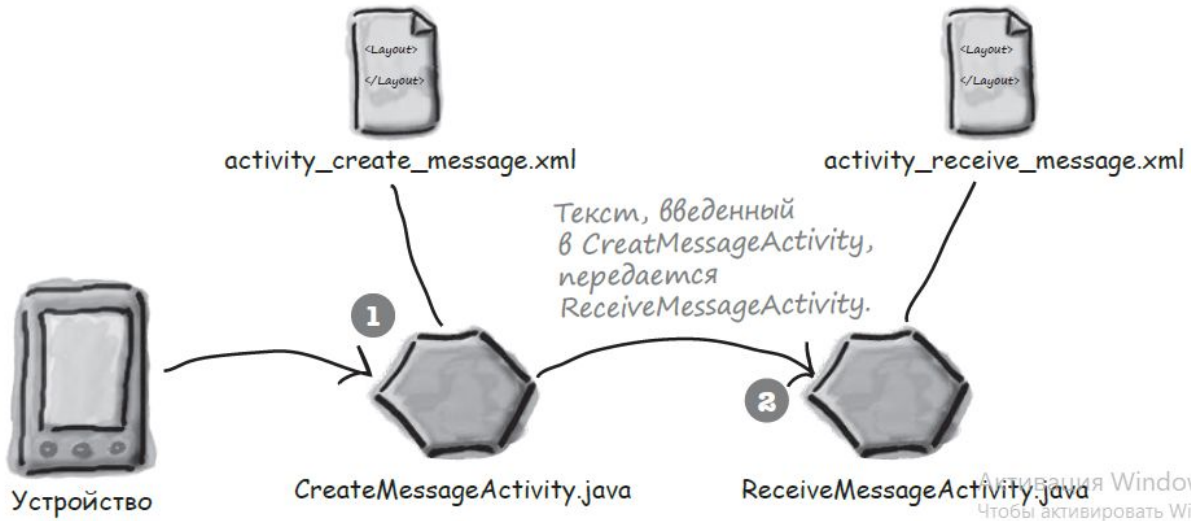
Активность – одна целенаправленная операция, которая может выполняться пользователем.



ПОСЛЕДОВАТЕЛЬНОСТЬ ДЕЙСТВИЙ

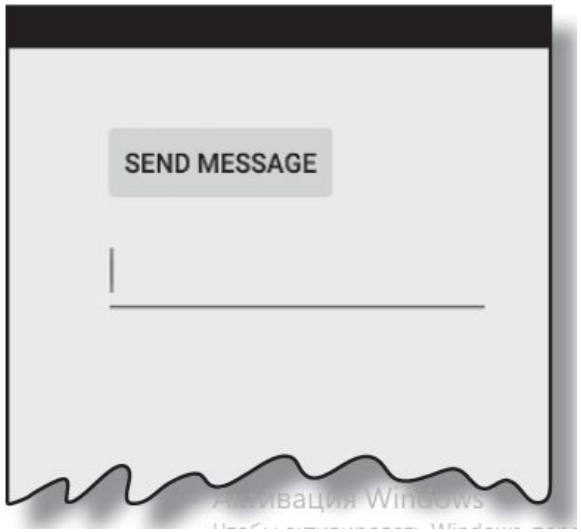
1. Создание базового приложения с одной активностью и макетом.
2. Добавление второй активности и макета.
3. Организация вызова второй активности из первой.
4. Организация передачи данных из первой активности во вторую.

2 **Пользователь щелкает на кнопке в CreateMessageActivity.**
Кнопка запускает активность `ReceiveMessageActivity`, которая использует макет *activity_receive_message.xml*.



НЕСКОЛЬКО АКТИВНОСТЕЙ

Создаем проект Android Studio Messenger, Добавляем кнопку и компонент EditText



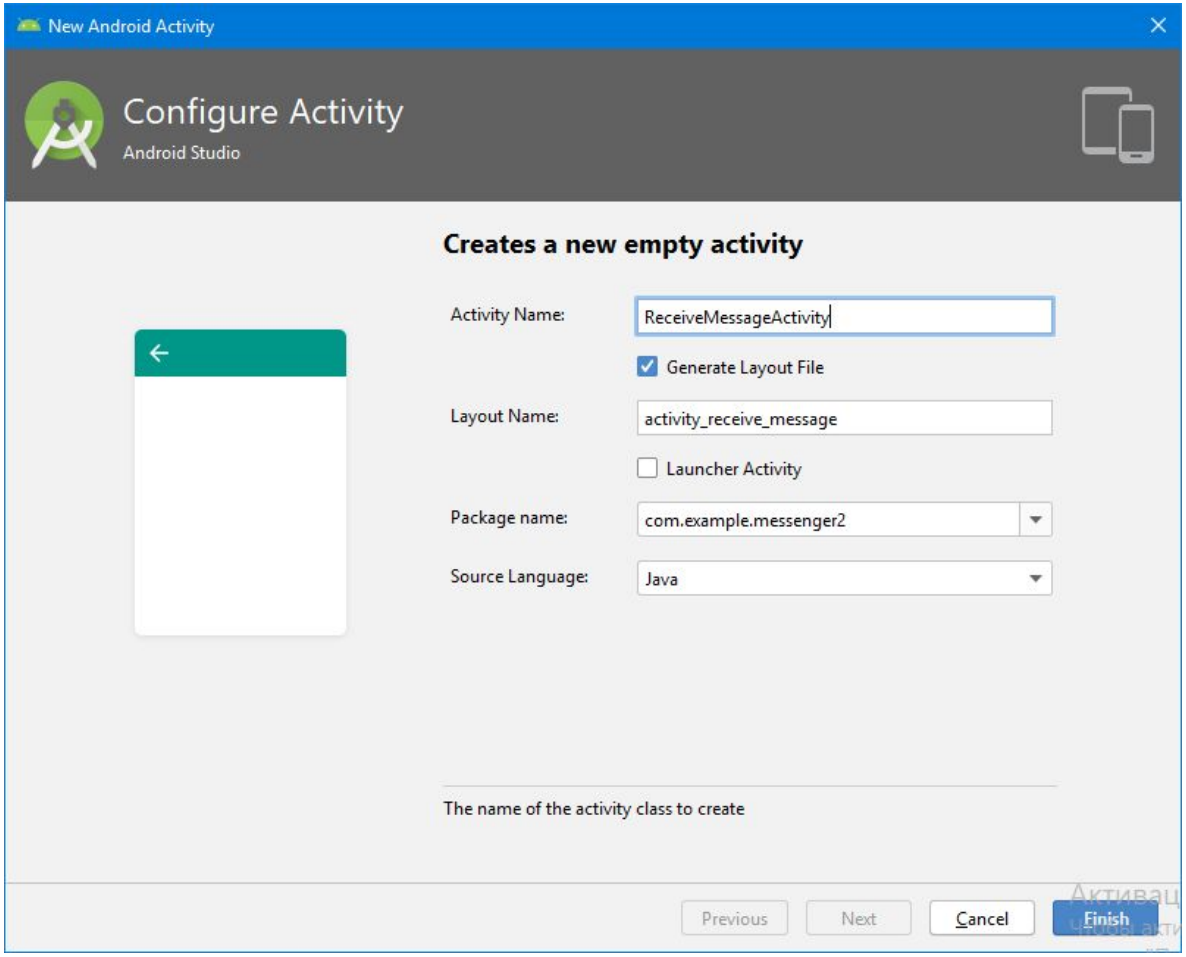
Создание 1-й активности

Создание 2-й активности

Вызов 2-й активности

Передача данных

ДОБАВЛЯЕМ ВТОРУЮ АКТИВНОСТЬ



Настройки приложения

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Messenger2"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".ReceiveMessageActivity"></activity>
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

активность может использоваться для запуска приложения, активность является главной активностью приложения.

ИНТЕНТ

ИНТЕНТ (Интент можно рассматривать как своего рода «намерение выполнить некую операцию». 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.)

Процедура создания и отправки интента состоит всего из двух строк кода. Для начала создайте интент:

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

Первый параметр сообщает Android, от какого объекта поступил интент; для обозначения текущей активности используется ключевое слово `this`. Во втором параметре передается имя класса активности, которая должна получить интент.

После того как интент будет создан, он передается Android следующим вызовом:

```
startActivity(intent);
```

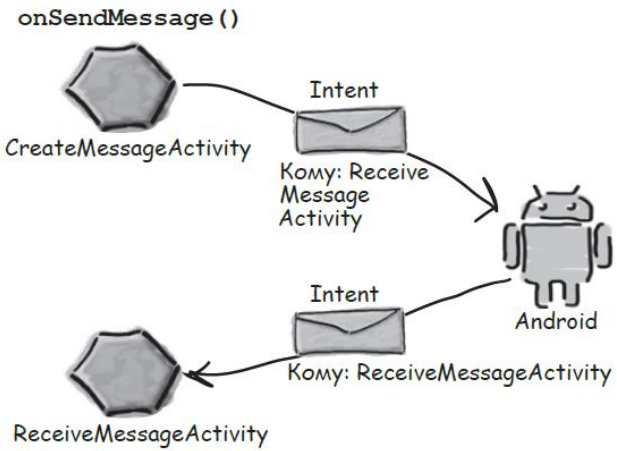
startActivity() запускает активность, указанную в интенте...

ИНТЕНТЫ

Этот вызов приказывает Android запустить активность, определяемую интентом. При получении интента Android убеждается в том, что все правильно, и приказывает активности запуститься. Если найти активность не удалось, инициируется исключение **ActivityNotFoundException**.

```
//Вызвать onSendMessage() при щелчке на кнопке
public void onSendMessage(View view) {
    Intent intent = new Intent(this, ReceiveMessageActivity.class);
    startActivity(intent);
}
```

Запустить активность ReceiveMessageActivity.



ПЕРЕДАЧА ТЕКСТА ВТОРОЙ АКТИВНОСТИ

1. Изменить макет *activity_receive_message.xml* так, чтобы он мог использоваться для вывода текста. Сейчас он представляет собой макет по умолчанию, сгенерированный мастером.
2. Обновить активность *CreateMessageActivity.xml*, чтобы она получала введенный пользователем текст. Этот текст должен быть добавлен в интент перед его отправкой.
3. Обновить код *ReceiveMessageActivity.java*, чтобы он выводил текст, отправленный в интенте.

putExtra() включает в интент дополнительную информацию

```
Intent intent = new Intent(this, Target.class);
intent.putExtra("сообщение", значение);
```

Как получить дополнительную информацию из интента

getIntent() возвращает интент, запустивший активность; из полученного интента можно прочитать любую информацию, отправленную вместе с ним. Например, если вы знаете, что интент включает строковое значение с именем “message”, используйте следующий вызов

```
Intent intent = getIntent();
String string = intent.getStringExtra("message");
int intNum = intent.getIntExtra("name", default_value);
```

Код первой активности

```
Button btnSend;
EditText etMessage;

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    btnSend = (Button) findViewById(R.id.btnSend);
    etMessage = (EditText) findViewById(R.id.etMessage);

}

public void onSendMessage(View view) {

    String messageText = etMessage.getText().toString();
    Intent intent = new Intent(this, ReceiveMessageActivity.class);
    intent.putExtra(ReceiveMessageActivity.EXTRA_MESSAGE, messageText);
    startActivity(intent);
}
```

Код второй активности

```
public class ReceiveMessageActivity extends AppCompatActivity {
    TextView tvMessage;
    public static String EXTRA_MESSAGE = "message";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_receive_message);
        tvMessage = (TextView) findViewById(R.id.tvMessage);
        Intent intent = getIntent();
        // извлечь из него сообщение вызовом
        //getStringExtra().
        String messageText = intent.getStringExtra(EXTRA_MESSAGE);
        tvMessage.setText(messageText);

    }
```

Запуск активностей из других приложений

Как узнать, какие активности доступны на устройстве пользователя?

Как узнать, какие из этих активностей подходят для того, что мы собираемся сделать?

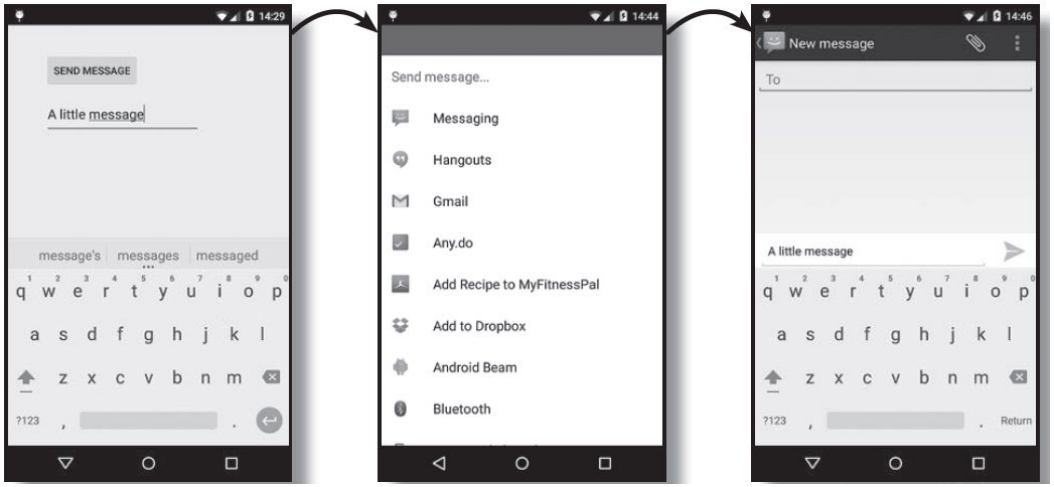
Как узнать, как использовать эти активности?

Интенты

Действия — стандартный механизм, при помощи которого Android узнает о том, какие стандартные операции могут выполняться активностями. Например, Android знает, что все активности, зарегистрированные для действия send, могут отправлять сообщения.

А теперь нужно научиться создавать интенты, использующие действия для получения набора активностей, которые могут использоваться для выполнения стандартных функций — например, для отправки сообщений.

1. **Создать интент с указанием действия**
2. **Разрешить пользователю выбрать используемое приложение.**



Неявные и явные интенты

Intent intent = new Intent(this, ReceiveMessageActivity.class);

Интенты называются **явными если вы явно сообщаете Android, какой класс должна запустить система**

Если требуется выполнить некоторое действие и вас не интересует, какой активностью оно будет выполнено, создайте **неявный интент**

Intent intent = new Intent(действие);

Android предоставляет целый ряд стандартных вариантов действий. Например, действие Intent.ACTION_DIAL используется для набора номера, Intent.ACTION_WEB_SEARCH — для выполнения веб-поиска, а Intent.ACTION_SEND — для отправки сообщений.

Если вы хотите создать интент для отправки сообщения, используйте команду следующего вида:

Intent intent = new Intent(Intent.ACTION_SEND);

Добавление дополнительной информации

После определения действия в интент можно включить дополнительную информацию. Допустим, вы хотите добавить текст, который образует тело отправляемого сообщения

intent.setType("text/plain");

intent.putExtra(Intent.EXTRA_TEXT, текст);

где текст — отправляемый текст.

Неявные и явные интенты

Эти атрибуты актуальны для Intent.ACTION_SEND, а не для всех возможных действий.
если вы хотите также указать тему сообщения, используйте вызов вида
intent.putExtra(Intent.EXTRA_SUBJECT, тема);
где тема — тема сообщения.

```
public void onSendMessage(View view) {

    String messageText = etMessage.getText().toString();

    // Вместо того, чтобы создавать
    //интент, предназначенный кон-
    //кретно для ReceiveMessageActivity,
    //мы создаем интент с указанием
    //действия отправки
    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.setType("text/plain");
    intent.putExtra(Intent.EXTRA_TEXT, messageText);
    startActivity(intent);
}
```

Фильтр интентов

Фильтр интентов сообщает Android, какие активности могут обработать те или иные действия

При получении интента система Android должна определить, какая активность (или активности) может этот интент обработать. Этот процесс называется **разрешением интента.**

При использовании *неявного интента система Android использует информацию, содержащуюся в интенте, для определения того, какие компоненты могут его получить. Для этого Android проверяет фильтры интентов, содержащиеся в экземплярах AndroidManifest.xml всех приложений*

Фильтр интентов указывает, какие типы интентов могут обрабатываться каждым компонентом


```
<activity android:name="ShareActivity">
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="text/plain"/>
        <data android:mimeType="image/*"/>
    </intent-filter>
</activity>
```

Фильтр интентов

Фильтр интентов также включает категорию. Категория предоставляет дополнительную информацию об активности: например, может ли она запускаться браузером или является ли она главной точкой входа приложения. Фильтр интентов должен включать категорию android.intent.category.DEFAULT, если он собирается принимать неявные интенты. Если активность не имеет фильтра интентов или не включает категорию с именем android.intent.category.DEFAULT, это означает, что активность не может запускаться неявным интентом. Она может быть запущена только явным интентом с указанием полного имени компонента


```
<activity android:name="ShareActivity">
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="text/plain"/>
        <data android:mimeType="image/*"/>
    </intent-filter>
</activity>
```

Фильтр интентов

После того как сравнение интента с фильтрами интентов, назначенных компонентам, будет завершено, Android смотрит, сколько совпадений удалось найти. Если найдено только одно совпадение, Android запускает компонент (в нашем случае это активность) и передает ему интент. Если будет найдено несколько совпадений, Android просит пользователя выбрать один из вариантов

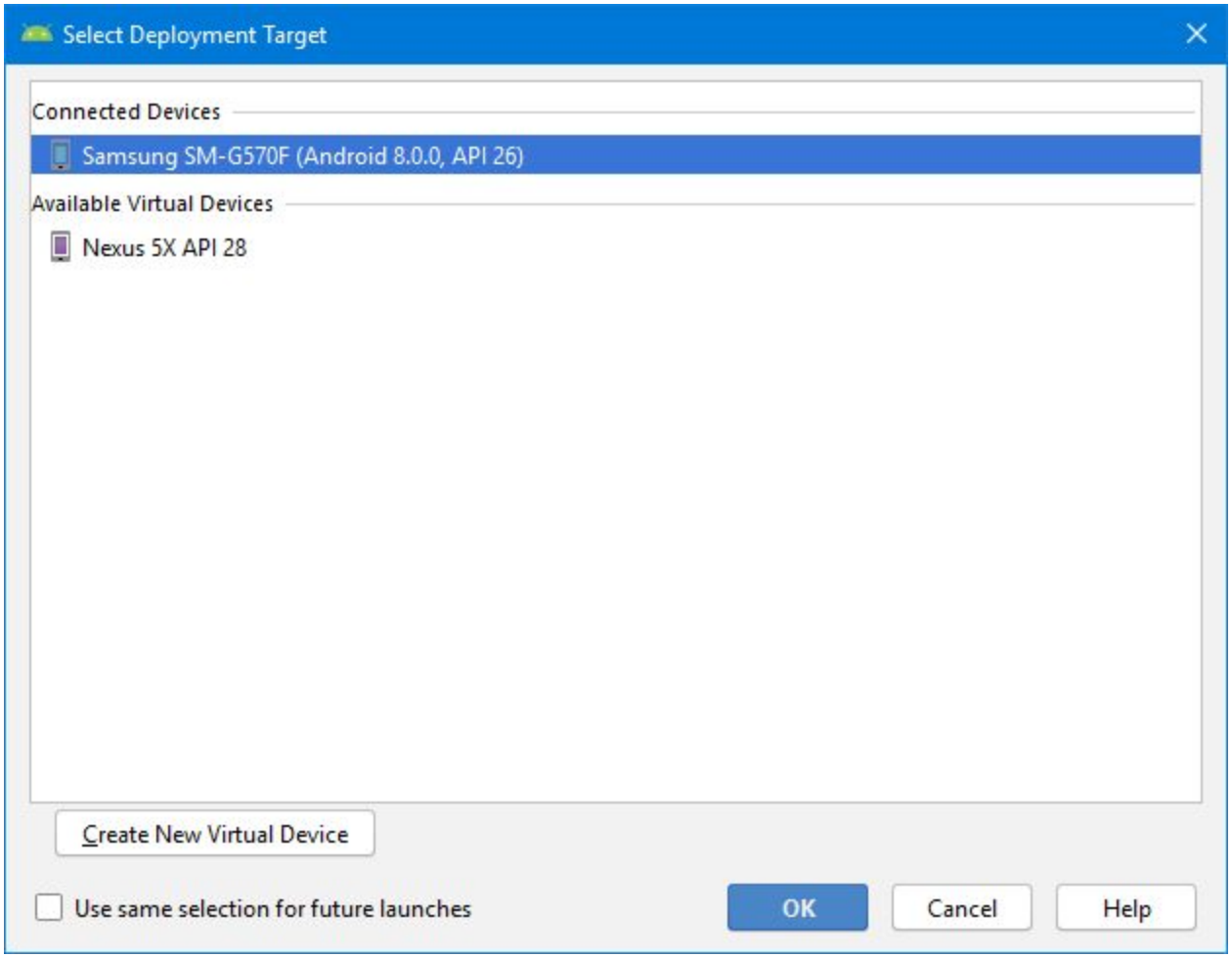
```
<activity android:name="ShareActivity">
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="text/plain"/>
        <data android:mimeType="image/*"/>
    </intent-filter>
</activity>
```

Сообщает Android, что активность может обрабатывать ACTION_SEND.

Фильтр интентов должен включать категорию DEFAULT; в противном случае он не сможет получать неявные интенты.

Типы данных, которые могут обрабатываться активностью.

Отладка на реальном устройстве



Вывод диалоговое окно выбора

метод Intent.createChooser(). Он получает уже созданный интент и «упаковывает» его в диалоговое окно выбора.

Вызвать метод createChooser() можно так:

Intent chosenIntent = Intent.createChooser(intent, "Send message...");

Метод createChooser() возвращает новый объект Intent. Он представляет собой новый явный интент, предназначенный для активности, выбранной пользователем. Он содержит всю дополнительную информацию, передававшуюся в исходном интенте, включая весь текст.

Чтобы запустить активность, выбранную пользователем, нужно вызвать:

startActivity(chosenIntent);

GridLayer

Панель контейнер RelativeLayout не всегда удобно так как смещение одного приводит к сдвигу всех остальных