ПРАКТИЧЕСКАЯ РАБОТА 9.
“ВЗАИМОДЕЙСТВИЕ ПРИЛОЖЕНИЕ СО СТОРОННИМИ ПРИЛОЖЕНИЯМИ И НАОБОРОТ”

В некоторых ситуациях необходимо направить пользователя из одного приложения в другое. К таким ситуациям относится необходимость: открыть приложение, отображающее карту, открыть приложение «Телефон» и др. Эта задача реализуется с помощью Intent. При создании такого типа Intent используется список предопределенных действий, например Intent.ACTION_VIEW, Intent.ACTION_SEND и др. Полный список действий доступен: https://developer.android.com/reference/android/content/Intent При использовании такого типа Intent нет необходимости указывать класс приложения напрямую. Система выбирает из установленных приложений позволяющее открыть запрашиваемый тип Intent.

В зависимости от Intent, который создает приложение, передаваемые данные могут быть Uri, другие типы или не быть вовсе.

Примеры создания различных типов Intent

Открыть приложение «Телефон»

```
Uri number = Uri.parse("tel:5551234");
Intent callIntent = new Intent(Intent.ACTION_DIAL, number);
```

Открыть карту

```
// Map point based on address
Uri location =
Uri.parse("geo:0,0?q=1600+Amphitheatre+Parkway,+Mountain+View,+C
alifornia");
// Or map point based on latitude/longitude
// Uri location = Uri.parse("geo:37.422219,-122.08364?z=14"); //
z param is zoom level
Intent mapIntent = new Intent(Intent.ACTION_VIEW, location);
```

Открыть веб-страницу

```
Uri webpage = Uri.parse("https://www.android.com");
Intent webIntent = new Intent(Intent.ACTION_VIEW, webpage);
```

Добавить данные в Intent

Некоторые виды Intent требуют «дополнительных» данных, которые предоставляют другие типы данных, например строку. Вы можете добавить один или несколько фрагментов дополнительных данных, используя различные методы putExtra(). По умолчанию система определяет соответствующий тип MIME, передаваемых в Intent, на основе включенных данных Uri. Если вы не включаете Uri в Intent, обычно следует использовать setType() для указания типа данных, связанных с Intent. Установка типа MIME дополнительно указывает, какие виды действий должны получать Intent.

Отправить email и прикрепить вложение

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);
// The intent does not have a URI, so declare the "text/plain"
MIME type
emailIntent.setType(HTTP.PLAIN_TEXT_TYPE);
emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[]
{"jan@example.com"}); // recipients
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Email subject");
emailIntent.putExtra(Intent.EXTRA_TEXT, "Email message text");
emailIntent.putExtra(Intent.EXTRA_STREAM,
Uri.parse("content://path/to/email/attachment"));
// You can also attach multiple items by passing an ArrayList of
Uris
```

Создать событие в календаре

```
// Event is on January 23, 2021 -- from 7:30 AM to 10:30 AM.
Intent calendarIntent = new Intent(Intent.ACTION_INSERT,
Events.CONTENT_URI);
Calendar beginTime = Calendar.getInstance();
beginTime.set(2021, 0, 23, 7, 30);
Calendar endTime = Calendar.getInstance();
endTime.set(2021, 0, 23, 10, 30);
calendarIntent.putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME,
beginTime.getTimeInMillis());
calendarIntent.putExtra(CalendarContract.EXTRA_EVENT_END_TIME,
endTime.getTimeInMillis());
calendarIntent.putExtra(Events.TITLE, "Ninja class");
calendarIntent.putExtra(Events.EVENT_LOCATION, "Secret dojo");
```

Для запуска Intent используется команда startActivity(intent)

Запуск приложения из сторонних приложений

Если ваше приложение может выполнять действие, которое может быть вызвано из другого приложения, ваше приложение должно быть готово отвечать на запросы действий, указав соответствующий intent-filter в Activity.

Например, если вы создаете социальное приложение, которое может обмениваться сообщениями или фотографиями с друзьями пользователя, вы должны поддерживать Intent.ACTION_SEND. Затем, когда пользователи инициируют действие «поделиться» из другого приложения, ваше приложение появляется в качестве опции в диалоговом окне выбора (также известном как «диалог значений неоднозначности»), как показано на рис. 1.

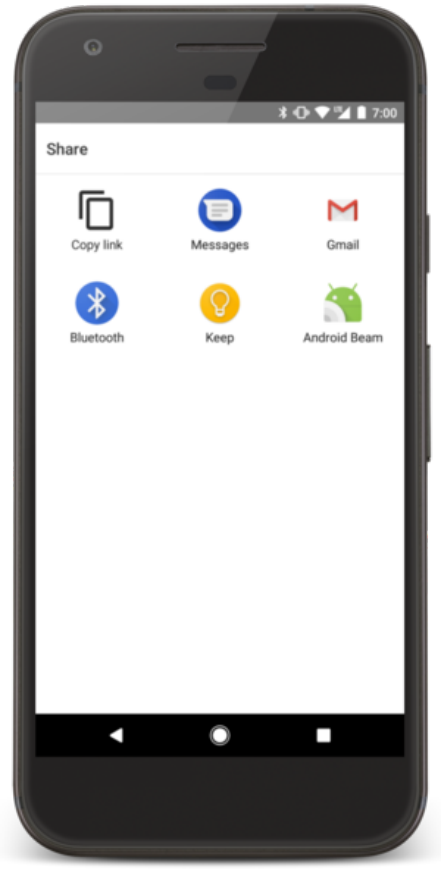

Рисунок 1. Пример экрана при выборе действия «Поделиться»

Чтобы разрешить другим приложениям запускать ваше Activity таким образом, вам нужно добавить элемент <intent-filter> в файл манифеста для соответствующего элемента <activity>.

Когда ваше приложение установлено на устройстве, система идентифицирует ваши intent-filter и добавляет информацию во внутренний каталог intent, поддерживаемый всеми установленными приложениями. Когда

приложение вызывает startActivity() или startActivityForResult() с неявным intent, система определяет, какое действие (или действия) может ответить на этот intent.

Добавить фильтр Intent

Чтобы правильно определить, какие Intent может обрабатывать ваше приложение и Activity, необходимо указать типы действий и данные, которые может принять приложение.

Система может отправить заданное Intent действию, если у этого действия есть intent-filter, удовлетворяющий критериям intent:

Action

Строка с названием выполняемого действия. Обычно это одно из определённых в системе значений, например ACTION_SEND или ACTION_VIEW.

Укажите это в своем intent-filter с помощью элемента <action>. Значение, указанное в этом элементе, должно быть полным строковым именем действия, а не константой API.

Data

Описание данных, передающихся в Intent. Указывается в фильтре intent с помощью элемента <data>. Используя один или несколько атрибутов в этом элементе, вы можете указать только тип MIME, только префикс URI, только схему URI или комбинацию этих и других, которые указывают на принятый тип данных.

Примечание. Если вам не нужно объявлять специфику данных Uri (например, когда ваша активность обрабатывает другие виды «дополнительных» данных вместо URI), вы должны указать только атрибут android:mimeType для объявления типа данных, которые обрабатывает ваша активность, например text/plain или image/jpeg.

Category

Предоставляет дополнительный способ описания действия, обрабатывающего intent, обычно связанного с жестом пользователя или местоположением, из которого оно было запущено. Существует несколько различных категорий, поддерживаемых системой, но большинство из них используются редко. Однако все неявные intent по умолчанию определяются с помощью CATEGORY_DEFAULT. Укажите это в своем intent-filter с помощью элемента <category>. В своем intent-filter вы можете объявить, какие

критерии принимает ваше Activity, объявив каждый из них с соответствующими элементами XML, вложенными в элемент <intent-filter>. Например, Activity с intent-filter, который обрабатывает intent ACTION_SEND, когда типом данных является текст или изображение:

```
<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>
```

Каждый intent-filter указывает только одно действие и один тип данных, но можно объявить несколько экземпляров элементов <action>, <category> и <data> в каждом <intent-filter>.

Если любые две пары действий и данных взаимоисключающие в своем поведении, следует создать отдельные intent-filter, чтобы указать, какие действия допустимы в сочетании с какими типами данных.

Например, предположим, что ваше приложение обрабатывает как текст, так и изображения для Intent ACTION_SEND и ACTION_SENDTO. В этом случае необходимо определить два отдельных intent-filter для двух действий, поскольку Intent ACTION_SENDTO должно использовать Uri данных для указания адреса получателя с помощью схемы URI отправки или отправки. Например:

```
<activity android:name="ShareActivity">
    <!-- filter for sending text; accepts SENDTO action with sms
URI schemes -->
    <intent-filter>
        <action android:name="android.intent.action.SENDTO"/>
        <category
android:name="android.intent.category.DEFAULT"/>
        <data android:scheme="sms" />
        <data android:scheme="smsto" />
    </intent-filter>
    <!-- filter for sending text or images; accepts SEND action
and text or image data -->
```

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

Обработка Intent в приложении

Чтобы решить, какое действие приложение должно выполнить при получение Intent из другого приложения, при запуске Activity необходимо получить текущий Intent через getIntent. Обычно это следует делать во время таких вызовов, как onCreate() или onStart(). Например:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.main);

    // Get the intent that started this activity
    Intent intent = getIntent();
    Uri data = intent.getData();

    // Figure out what to do based on the intent type
    if (intent.getType().indexOf("image/") != -1) {
        // Handle intents with image data ...
    } else if (intent.getType().equals("text/plain")) {
        // Handle intents with text ...
    }
}
```

Задание

1. Реализовать функционал, с помощью которого можно поделиться текстовой информацией из своего приложения при помощи стороннего приложения;
2. В соответствии с предметной областью, реализовать возможность отправки данных из стороннего приложения в разрабатываемое.