



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

---

**Институт информационных технологий (ИИТ)**

**Кафедра математического обеспечения и стандартизации ИТ**

## **ОТЧЁТ ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №7**

**по дисциплине**

**«Разработка мобильных приложений»**

Отчет представлен к

рассмотрению:

Студенты группы ИНБО-04-20

« » февраля 2022 г.

\_\_\_\_\_  
(подпись)

Ло В.Х.

Преподаватель

« »

2022 г.

\_\_\_\_\_  
(подпись)

Фандеев И.И.

Москва, 2022г.

## СОДЕРЖАНИЕ

ОТЧЕТ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ.....	3
1. Запуск другого приложения.....	3
2. Получения результата выполнения явления .....	5
3. Разрешаем другим приложениям запускать ваши явления .....	7
ВЫВОД.....	9
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	9

# ОТЧЕТ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОЙ РАБОТЫ.

## 1. Запуск другого приложения

### 1.1 Создание неявного намерения

Вот как создать интент, инициирующее телефонный звонок с помощью Uri, указывающий номер телефона:



Рисунок 1 – Создание неявного намерения

### 1.2 Просмотр карты

Ниже приведены несколько других намерений, задающих пары Uri и действий:



Рисунок 2 – Просмотр карты

### 1.3 Просмотр web-страницы.

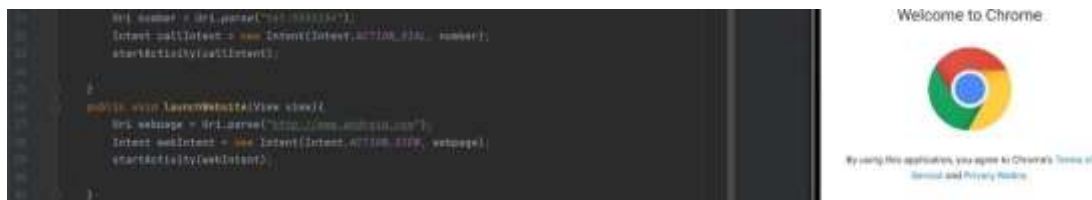


Рисунок 3 – Просмотр web-страницы.

### 1.4 Отправка письма с вложением:

Другие виды неявных интентов требуют «дополнительные» данные, которые предоставляют данные различных типов, таких как строки. Вы можете добавить один или несколько кусков дополнительных данных, используя различные putExtra() методы.

По умолчанию, система определяет соответствующий MIME тип, необходимый интенту, на основе Uri данных. Если вы не добавили Uri в интент, вы должны, как правило, использовать setType() для указания тип данных,

связанных с интендом. Установка MIME типа в дальнейшем определяет, какие виды activity должны получить это намерение.



Рисунок 4 – Отправка письма с вложением

## 1.5 Создание события в календаре:

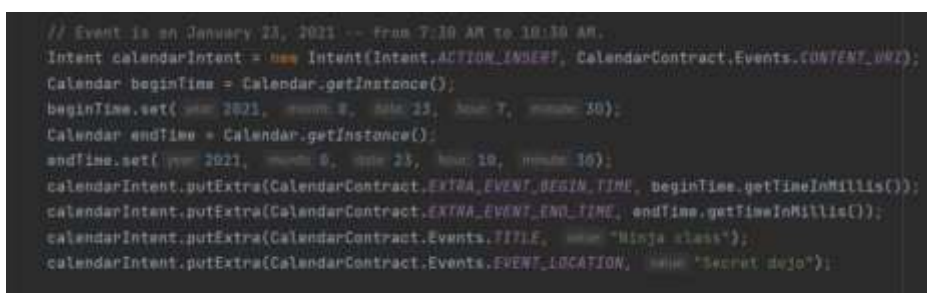


Рисунок 5 – Создание события в календаре

## 1.6 Проверка наличия приложения для получения намерения

Чтобы проверить существование явления, способного принять намерение, вызовите метод `queryIntentActivities()` для получения списка явлений, способных обрабатывать ваше намерение. Если метод вернет не пустой список (объект типа `List`), можете использовать ваше намерение.

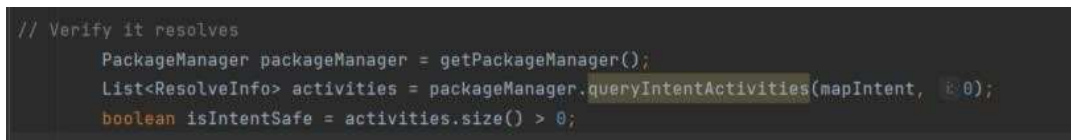


Рисунок 6 – Проверка наличия приложения для получения намерения

Чтобы показать такой диалог, создайте намерение используя метод `createChooser()` и передайте его в метод `startActivity()`

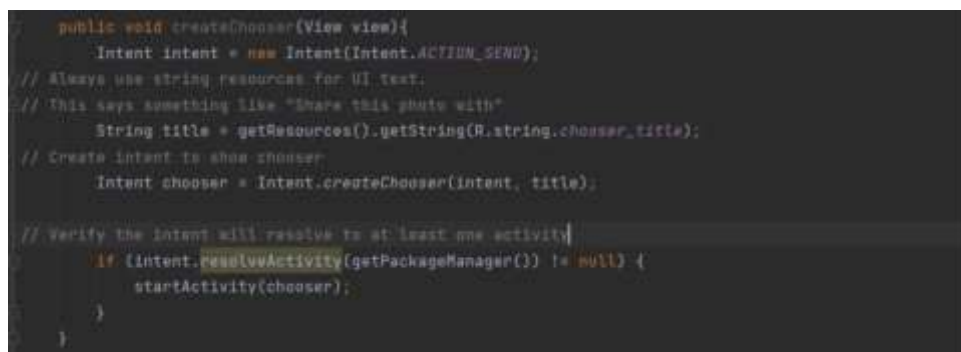


Рисунок 7 – createChooser()

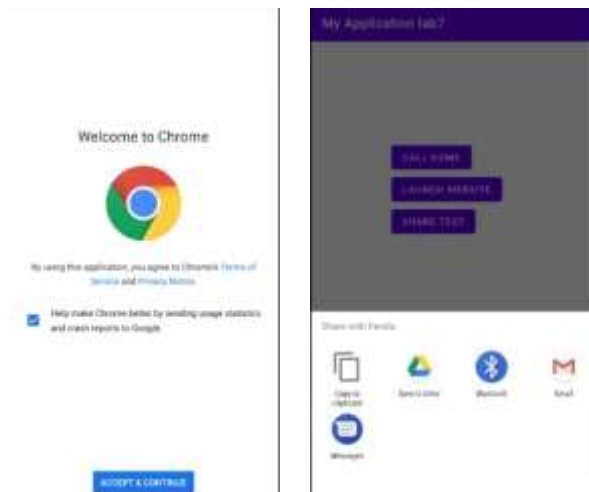


Рисунок 8 – Диалоги

## 2. Получения результата выполнения явления

### 2.1 Запуск явления

Нет ничего особенного в намерениях, которые вы будете использовать для получения результата, просто необходимо передавать дополнительный целочисленный аргумент в метод `startActivityForResult()`.

Данный аргумент представляет из себя “код запроса”, который позволяет отличить один запрос от другого. При передаче результата работы, в метод обратного вызова также передается код запроса, чтобы вы могли правильно обработать полученные данные.

```
private void pickContact() {
    Intent pickContactIntent = new Intent(Intent.ACTION_PICK, Uri.parse("content://contacts"));
    pickContactIntent.setType(ContactsContract.Contacts.CONTENT_TYPE); // Show user only contacts w/ phone numbers
    startActivityForResult(pickContactIntent, PICK_CONTACT_REQUEST);
}
```

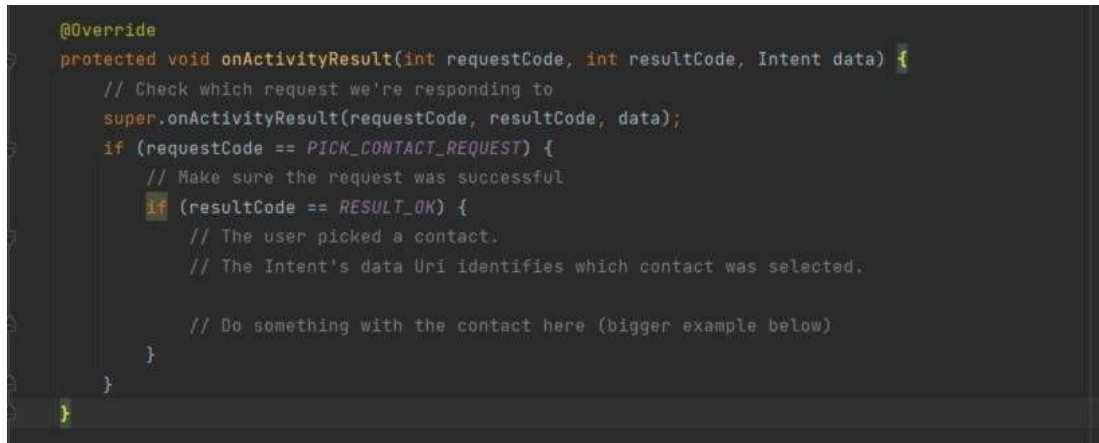
Рисунок 9 – Запуск явления

### 2.2 Получение результата

Если пользователь закрыл стороннее явление и вернулся в ваше, система вызывает метод `onActivityResult()`. В метод передается три аргумента:

- Код запроса, который вы указали при вызове `startActivityForResult()`.
- Код возврата явления. Это может быть или `RESULT_OK`, если операция завершена успешно, или `RESULT_CANCELED`, если пользователь просто вернулся или операция не была успешно завершена по каким-либо причинам.

- Намерение, в котором содержится результат выполнения.



```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // Check which request we're responding to
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == PICK_CONTACT_REQUEST) {
        // Make sure the request was successful
        if (resultCode == RESULT_OK) {
            // The user picked a contact.
            // The Intent's data Uri identifies which contact was selected.

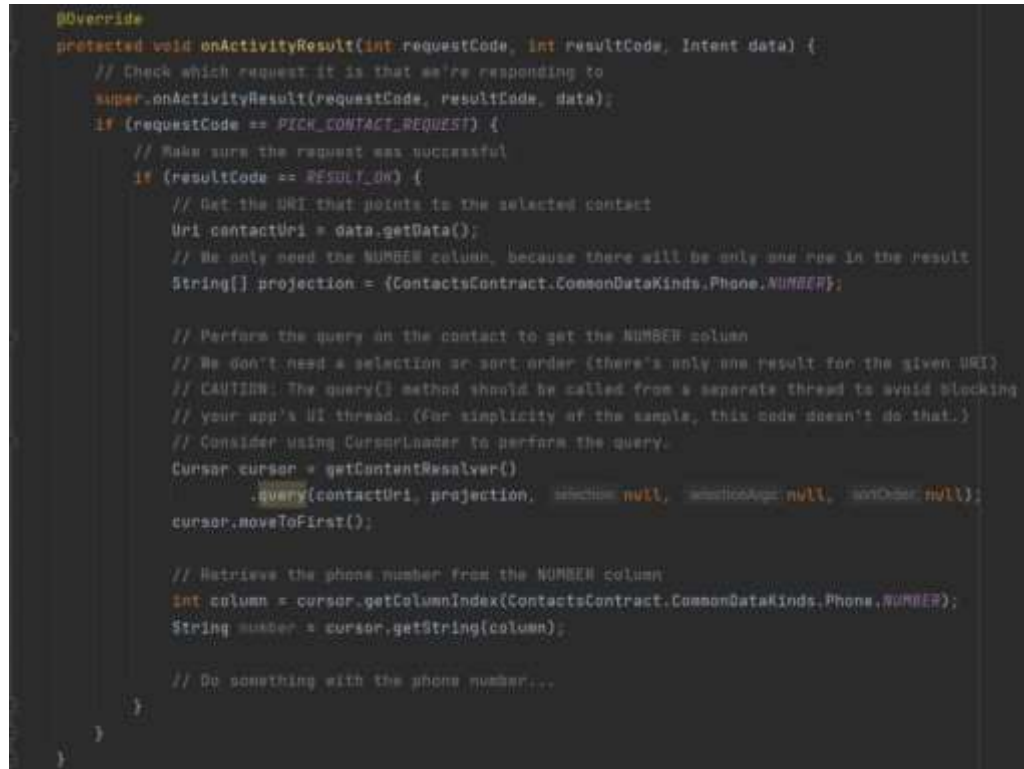
            // Do something with the contact here (bigger example below)
        }
    }
}

```

Рисунок 10 – Получение результата

## 2.3 Чтение контактов

Код выше показывает, как получить результат из приложения Контакты, но не показывает подробно как прочитать данные из полученного результата, поскольку это требует дополнительно разобраться с поставщиками контента. Однако, если вам интересно, ниже показано как получить номер телефона выбранного контакта:



```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // Check which request it is that we're responding to
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == PICK_CONTACT_REQUEST) {
        // Make sure the request was successful
        if (resultCode == RESULT_OK) {
            // Get the URI that points to the selected contact
            Uri contactUri = data.getData();
            // We only need the NUMBER column, because there will be only one row in the result
            String[] projection = {ContactsContract.CommonDataKinds.Phone.NUMBER};

            // Perform the query on the contact to get the NUMBER column
            // We don't need a selection or sort order (there's only one result for the given URI)
            // CAUTION: The query() method should be called from a separate thread to avoid blocking
            // your app's UI thread. (For simplicity of the sample, this code doesn't do that.)
            // Consider using CursorLoader to perform the query.
            Cursor cursor = getContentResolver()
                .query(contactUri, projection, selection null, selectionArgs null, sortOrder null);
            cursor.moveToFirst();

            // Retrieves the phone number from the NUMBER column
            int column = cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER);
            String number = cursor.getString(column);

            // Do something with the phone number...
        }
    }
}

```

Рисунок 11 – Чтение контактов

## 3. Разрешаем другим приложениям запускать ваши явления

### 3.1 Добавление фильтра намерений

Для того, чтобы правильно указать какие намерения может обработать ваше явление, каждый добавленный фильтр должен конкретно указывать действия и данные, которое принимает явление. Система может передать объект намерения в явление только в том случае, если оно имеет фильтр, включающий все критерии полученного намерения.

Каждое поступающее намерение указывает только одно действие и только один тип данных, но можно указать несколько элементов `<action>`, `<category>` и `<data>` в каждом `<intent-filter>`

Если любые две пары действий и данных, взаимоисключают поведение друг друга, вы должны создать разные фильтры, чтобы точно определить, приемлемые пары действий с типами данных.

Предположим, что ваше явление может обрабатывать и текст и изображения для обоих действий `ACTION_SEND` и `ACTION_SENDTO`. В данном случае, вы должны определить два разных фильтра для каждого из действий, поскольку намерение `ACTION_SENDTO` должно использовать `Uri`, чтобы определиться с адресом получателя, используя схему `URI send` или `send_to`.

```
<activity
  android:name="ShareActivity"
  android:exported="true">
  <!-- 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="mailto" />
  </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>
```

Рисунок 12 – Добавление фильтра намерений

## 3.2 Управление намерениями в вашем явлении

Для того, чтобы решить, какие действия следует выполнить в вашем явлении, вы можете использовать объект типа `Intent`, который запустил его.

После запуска явления, вызовите метод `getIntent()`, чтобы получить намерение, запустившее явление. Вы можете сделать это в любой момент жизненного цикла явления, однако лучше сделать это в методах `onCreate()` или `onStart()`.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    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 ...
    }
}
```

Рисунок 13 – Управление намерениями в вашем явлении

## 3.3 Возвращение результата

Если вы хотите вернуть результат в стороннее явление, вызвавшее ваше, просто вызовите метод `setResult()`, чтобы передать результат через новое намерение. Если операция отменена и нужно вернуть пользователя обратно в его приложение, вызовите метод `finish()`, чтобы закрыть и уничтожить ваше явление.

```
// Create intent to deliver some kind of result data
Intent result = new Intent(Intent.ACTION_MAIN, Uri.parse("content://result_uri"));
setResult(Activity.RESULT_OK, result);
finish();
```

Рисунок 14 – Возвращение результата



## **ВЫВОД**

В ходе выполнения практической работы были изучены:

- Взаимодействие с другими приложениями.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Лекции по дисциплине «Разработка мобильных приложений» / И. В. Сеницын, МИРЭА — Российский технологический университет, 2022.