МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Отчёт по лабораторной работе № 4 по курсу «Программирование мобильных систем»**

«Вызов Активности с помощью явного намерения и получение результатов работы. Использование неявных Намерений. Получение данных из Намерения»

ВЫПОЛНИЛ студент группы 16-ИТ-3

Яблонский А.С.

ПРОВЕРИЛ преподаватель

Лукьянов А.О.

Полоцк, 2019 г.

**ЦЕЛЬ РАБОТЫ:** Научиться вызывать Активность с использованием явного намерения и получать результаты её работы. Научиться использовать неявные Намерения и получать данные из Намерения.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ:**

1. Дайте определение понятию «Намерение» в ОС Android.

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

1. Для чего могу применяться Намерения?

Намерения могут применяться для:

• объявления о желании (необходимости) вашего приложения запуска какой-то Активности или Сервиса для выполнения определенных действий.

• Извещения о том, что произошло какое-то событие.

• Явного запуска указанного Сервиса или Активности.

1. Перечислите и охарактеризуйте методы для запуска Активности.

Чтобы запустить нужную Активность, вызывается метод **startActivity**(someIntent).

В конструкторе Намерения можно явно указать класс Активности, которую требуется запустить, или действие, которое нужно выполнить. Во втором случае система автоматически подберет нужную Активность, используя механизм, называемый Определением Намерений (Intent Resolution). Метод startActivity находит запускает Активность, наиболее подходящую вашему Намерению.

По окончании работы запущенной таким образом Активности запустившая ее Активность не получает никаких извещений о результатах обработки Намерения. Если требуется получать результаты, используется метод **startActivityForResult**.

1. Опишите механизм возврата результата работы Активности.

Активность, запущенная с помощью метода **startActivity**, полностью независима от запустившей ее Активности и, соответственно, завершает свою работу, никому об этом не сообщая. В то же время, вы можете запускать Активности, «связанные» со своим «породителем». Такой способ отлично подходит для ситуаций, когда «дочерняя» Активность должна обработать ввод пользовательских данных и предоставить результаты обработки «родительской» Активности. Запускаемые таким образом (с помощью метода **startActivityForResult**) Активности должны быть «зарегистрированы» в Манифесте приложения.

В отличие от метода **startActivity**, метод **startActivityForResult** требует явного указания еще одного параметра – *кода запроса (request code).* Этот параметр используется вызывающей Активностью для определения того, какая именно дочерняя Активность завершила работу и (возможно) предоставила результаты:

*private static final int BUY\_BEER = 1;*

*Intent intent = new Intent(this, MyOtherActivity.class);*

*startActivityForResult(intent, BUY\_BEER);*

1. Перечислите и охарактеризуйте параметры, передаваемые обработчику onActivityResult.

• **Request code**. Использованный при запуске дочерней активности код запроса.

• **Result code**. Код возврата.

• **Data**. Намерение, используемое для упаковки возвращаемых данных.

1. Дайте определение понятиям «Неявное намерение» и «Фильтр Намерений».

**Неявные намерения** используются для запуска Активностей для выполнения заказанных действий в условиях, когда неизвестно (или безразлично), какая именно Активность (и из какого приложения) будет использоваться.

Для определения того, какой именно компонент должен быть запущен для выполнения действий, указанных в Намерениях, Android использует **Фильтры Намерений** (Intent Filters). Используя Фильтры Намерений, приложения сообщают системе, что они могут выполнять определенные действия (action) с определенными данными (data) при определенных условиях (category) по заказу других компонентов системы.

1. Перечислите и охарактеризуйте основные атрибуты Фильтра Намерений.

Для регистрации компонента приложения (Активности или Сервиса) в качестве потенциального обработчика Намерений, требуется добавить элемент **<intent-filter>** в качестве дочернего элемента для нужного компонента в Манифесте приложения. У элемента **<intent-filter>** могут быть указаны следующие дочерние элементы (и соответствующие атрибуты у них):

• **<action>**. Атрибут android:name данного элемента используется для указания названия действия, которое может обслуживаться. Каждый Фильтр Намерений должен содержать не менее одного вложенного элемента <action>. Если не указать действие, ни одно Намерение не будет «проходить» через этот Фильтр. У главной Активности приложения в Манифесте должен быть указан Фильтр Намерений с действием android.intent.action.MAIN

• **<category>**. Сообщает системе, при каких обстоятельствах должно обслуживаться действие (с помощью атрибута android:name). Внутри <intent-filter> может быть указано несколько категорий. Категория android.intent.category.LAUNCHER требуется Активности, которая желает иметь «иконку» для запуска. Активности, запускаемые с помощью метода startActivity, обязаны иметь категорию android.intent.category.DEFAULT

• **<data>**. Дает возможность указать тип данных, которые может обрабатывать компонент. <intent-filter> может содержать несколько элементов <data>. В этом элементе могут использоваться следующие атрибуты:

• *android:host* : имя хоста (например, www.specialist.ru)

• *android:mimetype* : обрабатываемый тип данных (например, text/html)

• *android:path* : «путь» внутри URI (например, /course/android)

• *android:port* : порт сервера (например, 80)

• *android:scheme* : схема URI (например, http)

1. Перечислите и охарактеризуйте этапы проверки соответствия Намерения и Активности.

**Проверка действий**. После этого шага остаются только компоненты приложений, у которых в Фильтрах Намерений указано действие Намерения. В случае, если действие в Намерении отсутствует, совпадение происходит для всех Фильтров Намерений, у которых указано хотя бы одно действие.

• **Проверка категорий**. Все категории, имеющиеся у Намерения, должны присутствовать в Фильтре Намерений. Если у Намерения нет категорий, то на данном этапе ему соответствуют все Фильтры Намерений, за одним исключением, упоминавшимся выше: Активности, запускаемые с помощью метода startActivity, обязаны иметь категорию android.intent.category.DEFAULT, так как Намерению, использованному в этом случае, по умолчанию присваивается данная категория, даже если разработчик не указал ничего явно. Из этого исключения, в свою очередь, есть исключение: если у Активности присутствуют действие android.intent.action.MAIN и категория android.intent.category.LAUNCHER, ему не требуется иметь категорию android.intent.category.DEFAULT.

• **Проверка данных**. Здесь применяются следующие правила:

• Намерение, не содержащее ни URI, ни типа данных, проходит через Фильтр, если он тоже ничего перечисленного не содержит.

• Намерение, которое имеет URI, но не содержит тип данных (и тип данных невозможно определить по URI), проходит через Фильтр, если URI Намерения совпадает с URI Фильтра. Это справедливо только в случае таких URI, как mailto: или tel:, которые не ссылаются на реальные данные.

• Намерение, содержащее тип данных, но не содержащее URI подходит только для аналогичных Фильтров Намерений.

• Намерение, содержащее и тип данных, и URI (или если тип данных может быть вычислен из URI), проходит этот этап проверки, только если его тип данных присутствует в Фильтре. В этом случае URI должен совпадать, либо(!) у Намерения указан URI вида content: или file:, а у Фильтра URI не указан. То есть, предполагается, что если у компонента в Фильтре указан только тип данных, то он поддерживает URI вида content: или file:.

1. Как определить Намерение, вызвавшее запуск Активности?

Поскольку объекты типа Intent служат, в том числе, для передачи информации между компонентами одного или нескольких приложений, может возникнуть необходимость в работе с объектом Намерения, вызвавшим Активность к жизни.

Для получения доступа к этому объекту используется метод getIntent. Пример: *Intent intent = getIntent();*

1. Как получить доступ к данным, содержащимся в Намерении?

• метод getAction возвращает действие Намерения

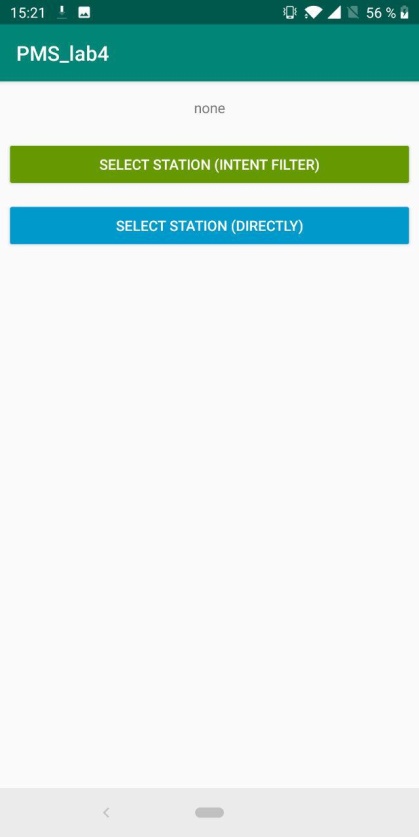
• метод getData возвращает данные Намерения (обычно URI)

• набор методов для разных типов вида get**ТИП**Extra позволяет получить доступ к типизированным значениям, хранящимся в свойстве extras Намерения.

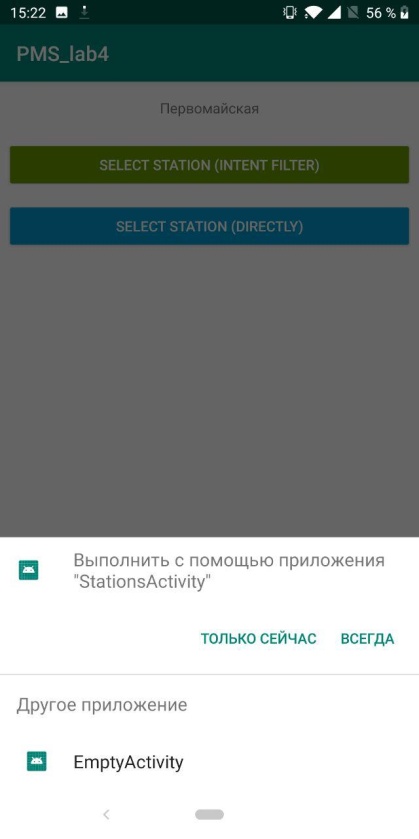
**ОПИСАНИЕ ПРОДЕЛАННОЙ РАБОТЫ:**

1. Создан новый проект **MetroPicker**.
2. Добавлена вспомогательная Активность **StationsActivity** для отображения и выбора станций метро.
3. Отредактирован файл разметки **res/layout/main.xml**: добавлена кнопка выбора станции метро, присвоены идентификаторы виджетам TextView и Button для того, чтобы на них можно было ссылаться в коде.
4. Установлен обработчик нажатия на кнопку в главной Активности для вызова списка станции и выбора нужной станции.
5. Написан нужный обработчик для установки выбранной станции метро в виджет TextView родительской Активности (метод setText виджета TextView позволяет установить отображаемый текст). А также обработана ситуация, когда пользователь нажимает кнопку «Назад» (в этом случае «никакой станции не выбрано» и главная Активность должна известить об этом пользователя).
6. Протестировано созданное приложение, проверена реакция на различные действия потенциальных пользователей.
7. Модифицированы методы onCreate Активностей так, что с помощью Toast они показывают действие вызвавшего их Намерения.

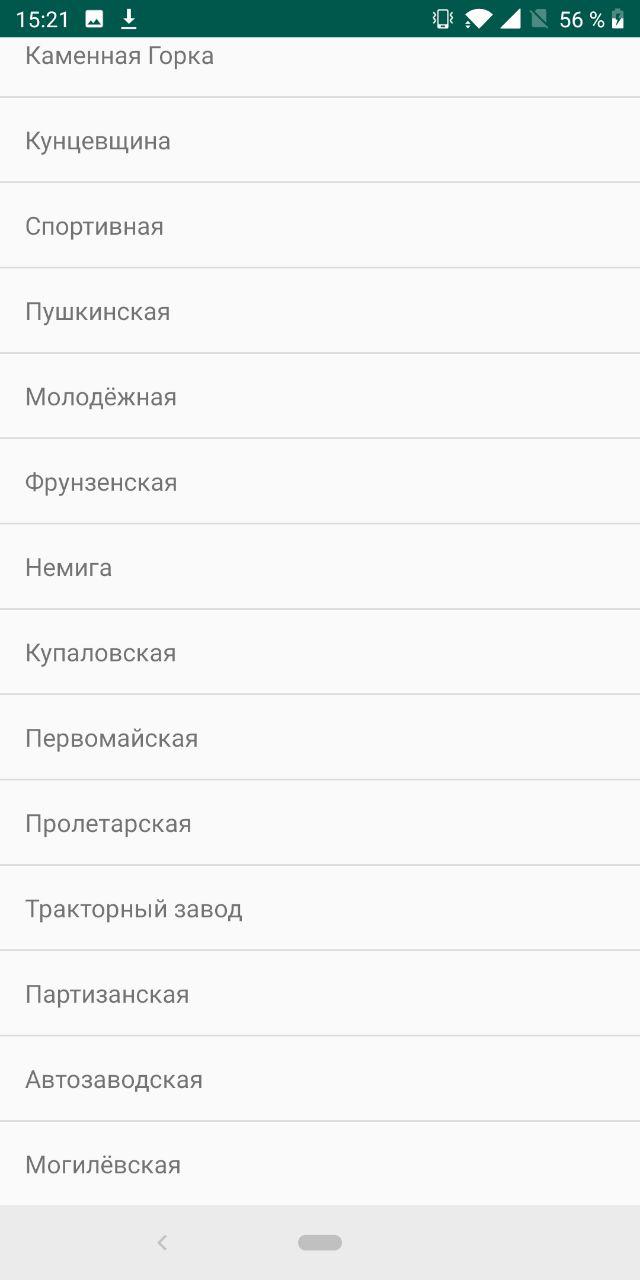
**РЕЗУЛЬТАТЫ ВЫПОЛЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ**:



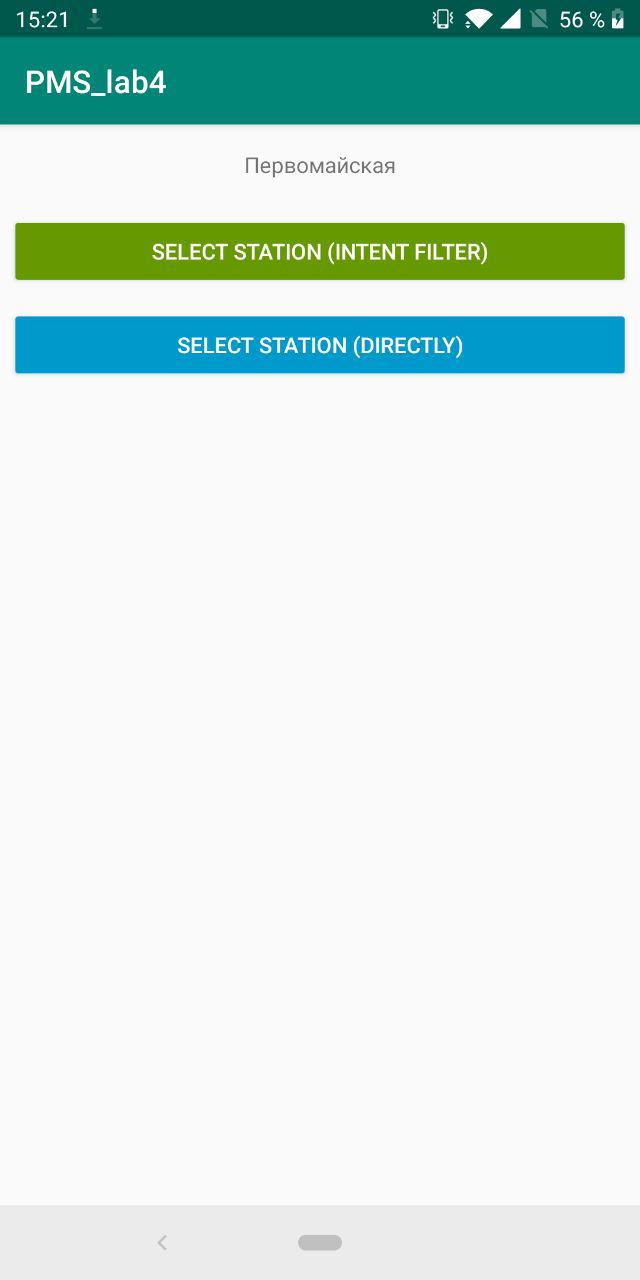
**Рисунок 1** — страница при запуске приложения



**Рисунок 2** — результат нажатия на кнопку «выбрать станцию неявно»



**Рисунок 3** — активити выбора станции



**Рисунок 4** — результат выбора определенной станции

**Вывод:** в результате выполнения лабораторной работы я научился вызывать Активность с использованием явного намерения и получать результаты её работы, а также использовать неявные намерения и получать данные из них.