**Домашнее задание к занятию «3.2. Организация навигации (перемещение между Activity)»**

**О документации**

Актуальная информация об изменениях в мире Android:

1. [Выступления разработчиков](https://www.youtube.com/user/androiddevelopers).
2. Документация.
3. Исходный код.

**Задача. Editing**

**Задача**

На лекции мы создали пост в отдельной Activity. Вам нужно добавить функцию редактирования, тоже не в MainActivity. Вы можете сделать это по-разному:

1. Создать отдельное Activity для редактирования.
2. Совместить создание и редактирование в одной Activity, но тогда нужно будет отличать создание от редактирования.

Опубликуйте изменения в вашем проекте на GitHub. Убедитесь, что apk собирается с помощью GitHub Actions и при установке в эмуляторе приложение работает корректно.

**Задача. YouTube Video**

**Задача**

На Intent в Android строится значительная часть взаимодействия между приложениями. Мы можем задействовать другие приложения для отображения нужного нам контента или выполнения действий и т. д.

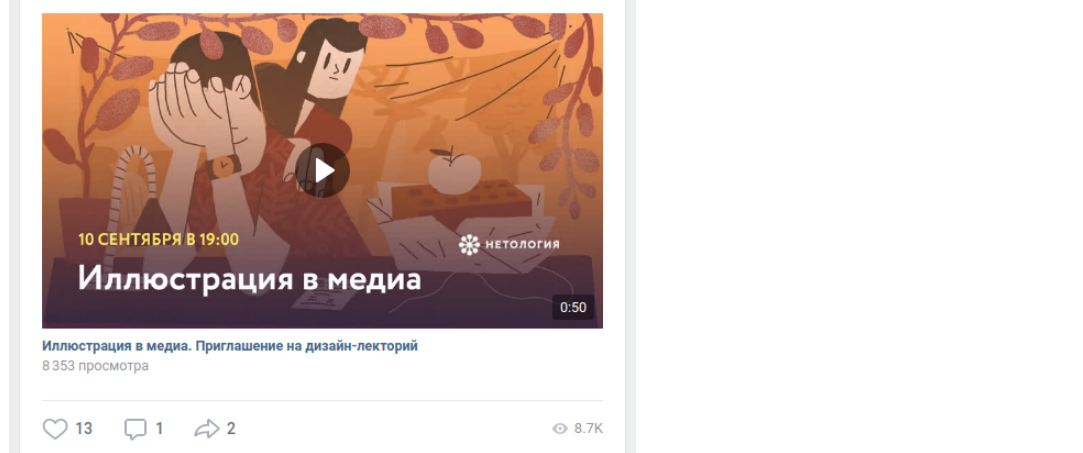
Изучите самые распространённые Intent [по ссылке](https://developer.android.com/guide/components/intents-common).

Идея: мы хотим сделать в классе Post необязательное поле video, в которое пользователь может вставить ссылку на YouTube-видео.

Пока мы сами не умеем проигрывать видео, но можем подключить для этого другие приложения.

Проанализируйте, как другие приложения отображают интерфейс постов с видео и как оно проигрывается.

Часто это выглядит так:

[](https://github.com/netology-code/and2-homeworks/blob/master/09_navigation/pic/video.png)

Вам нужно добавить в разметку поста отдельный блок, который будет показываться при наличии ссылки на видео. При нажатии на этот блок запускать неявный Intent со ссылкой, чтобы система его обработала и отобразила пользователю видео в браузере или в приложении YouTube (при наличии).

Советы по реализации

1. Вместо обложки видео поставьте произвольную картинку-заглушку и кнопку Play.
2. Перехватывайте все клики на кнопке и на обложке: пользователь не обязательно должен попадать в саму кнопку.
3. Для открытия внешнего приложения достаточно:
   * URL вида: "https://www.youtube.com/watch?v=WhWc3b3KhnY" или сокращённой версии;
   * передать этот URL в Uri.parse: Intent(Intent.ACTION\_VIEW, Uri.parse('url'));
   * стартовать Activity с созданным Intent.

Если вас заинтересовала работа с внешними Intent в Android, вам могут быть полезны методы [Intent.resolveActivity](https://developer.android.com/reference/android/content/Intent#resolveActivity(android.content.pm.PackageManager)) и [PackageManager.queryIntentActivities](https://developer.android.com/reference/android/content/pm/PackageManager#queryIntentActivities(android.content.Intent,%20int)). Они позволяют получить данные о том, какая активити будет запущена после отправки неявного Intent в метод startActivity. Так можно получить список приложений, которые обрабатывают Intent, отправляемые из вашего приложения.

Для работы этих методов на Android 11 и выше необходимо в AndroidManifest.xml добавить параметр queries, чтобы указать системе, что вы будете делать подобные запросы, и перечислить их параметры. Для просмотра "активитей", которые обрабатывают Intent.ACTION\_VIEW, нужно прописать:

<manifest>

<application>

...

</application>

<queries>

<intent>

<action android:name="android.intent.action.VIEW" />

<data android:scheme="\*" />

</intent>

</queries>

</manifest>

С помощью этих методов можно увидеть, какое приложение и какая активити в нём обработает ваш Intent. Для Intent-просмотра видео на эмуляторе получается такой результат:

queryIntentActivities: [ResolveInfo{ebaf4b0 com.google.android.youtube/.UrlActivity o=1 m=0x508000}]

resolveActivity: ComponentInfo{com.google.android.youtube/com.google.android.youtube.UrlActivity}

Опубликуйте изменения в вашем проекте на GitHub. Убедитесь, что apk собирается с помощью GitHub Actions и при установке в эмуляторе приложение работает корректно.