**Домашнее задание:**

**Жизненный цикл Android компонентов.**

**Ознакомиться с дополнительными полезными материалами по темам лекции.**

1. Жизненный цикл - <https://swiftbook.ru/post/tutorials/android-lifecycle/>
2. ViewModel - <https://habr.com/ru/post/334942/>
3. Task and back stack - <https://www.youtube.com/watch?v=GN-MltHRpi4>
4. Fragment - <https://metanit.com/java/android/8.1.php>
5. FragmentTransaction - <https://startandroid.ru/ru/uroki/vse-uroki-spiskom/175-urok-105-android-3-fragments-dinamicheskaja-rabota.html>

**Задача №1**

**Цель:**

Создание собственного приложения с 2 фрагментами.

**Уточнение:**

Макетов не будет, стилизация интерфейсной части может быть индивидуальной на ваше усмотрение. Главное, реализовать каждый из пунктов, чтобы в конце приложение имело точный функционал.

**Формат сдачи:**

Предоставить проект ссылкой на GitHub.

**Шаги:**

1. Реализовать собственную MainActivity. На активити на весь экран должен размещаться SenderFragment.
2. SenderFragment на интерфейсе должен иметь кнопку c текстом “Send message”. По клику на кнопку должна происходить “отправка сообщения на второй фрагмент”. В этом случае SenderFragment должен быть заменен на ReceiverFragment с передачей ему любого текста сообщения.
3. ReceiverFragment должен отображать текст сообщения на экране. Также на фрагменте должна быть кнопка “Read”, нажатие на нее “отметит сообщение прочитанным” и заменит отображаемый текст на “All messages is read”. Нажатие на системную кнопку назад должно вернуть пользователя на SenderFragment.
4. На ReceiverFragment необходимо сохранять текущее состояние с помощью ViewModel. В качестве полей состояния ViewModel разрешается использовать только MutableLiveData или LiveData.

**Критерии сдачи:**

1. SenderFragment и ReceiverFragment должны быть разными, чтобы визуально их можно было отличить друг от друга.
2. SenderFragment должен отображать кнопку с текстом “Send message” (допускается капсом).
3. Клик на кнопку send message на SenderFragment должен **заменять** SenderFragment на ReceiverFragment.
4. На ReceiverFragment системная кнопка назад должна перемещать пользователя обратно на SenderFragment.
5. При открытии ReceiverFragment должен отображать текст любого сообщения из SenderFragment и кнопку с текстом “Read” (допускается капсом).
6. Нажатие на кнопку “Read” должно заменять текст сообщения на “All messages is read”.
7. После поворота экрана ReceiverFragment должен отображать тот же текст, что был до поворота экрана (до/после клика на кнопку read).

**Задача №2**

**Цель:**

Самостоятельная реализация различных транзакций фрагментов.

**Уточнение:**

Для выполнения задачи вам потребуется завести новый проект или воспользоваться проектом и задачи 1, который будет содержать MainActivity, FragmentA и FragmentB. Fragment A должен будет иметь кнопку, нажатие на которую будет выполнять разные транзакции.

На фрагментах необходимо залогировать следующие методы: onAttach, onCreate, onCreateView, onViewCreated, onStart, onResume, onPause, onStop, onSaveInstaceState, onDestroyView, onDestroy, onDetach.

Логи каждого метода должны содержать его название и название фрагмента. В качестве образца оформления лога - можно взять логи из примеров лекции. Если вы еще не знаете как создавать логи и как их просматривать - внизу будет описание.

**Формат сдачи:**

Необходимо в письменном виде (формат word) по каждому из сценариев дать краткие ответы на вопросы и предоставить логи колбэков, которые будут вызваны в этих сценариях.

**Важное уточнение:**

Предоставляемые логи должны содержать только колбэки жизненного цикла FragmentA и FragmentB. Сценарий записи логов должен быть без лишних действий, т.е. с момента холодного старта приложения (когда оно выгружено из недавних), нажатие на кнопку перехода на FragmentB, поворот экрана. Если вы несколько раз повернете экран, или заблокируете телефон, или свернете приложение и это будет отображено в логах вашего сценария - ответ по сценарию не будет засчитан.

**Сценарий:**

На экране FragmentA в состоянии Resumed, по клику на кнопку выполнить **транзакцию** **X** для FragmentB. На FragmentB повернуть экран.

**Вопросы:**

1. В каком состоянии до поворота экрана будут находяться FragmentA и FragmentB?
2. В каком состоянии после поворота будут находиться FragmentA и FragmentB?
3. Какой колбэк будет обработан последним в каждом из фрагментов?

**Транзакции X:**

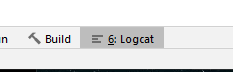
1. add
2. replace
3. add + addToBackStack
4. replace + addToBackStack

**Критерии сдачи:**

1. Логи колбэков во всех сценариях являются верными.
2. Указаны верные конечные состояния фрагментов.
3. Указаны верные последние обработанные колбэки каждого из фрагментов.

**P.S.**

Для просмотра логов вы можете открыть на нижней панели AndroidStudio logcat



Для примера вы можете взять такой вид установки логов 

Найти вы их сможете выбрав следующие фильтры Logcat - устройство или эмулятор, ваше приложение из списка, уровень логов info - Information, в строке поиска указать ваш tag, в моем примере это LifecycleCallbacks и выбрать пункт show only selected application.

