UI-фрагменты и Fragment Manager

Фрагмент – модуль пользовательского интерфейса который не может существовать сам по себе.

Он встраивается в Activity, который предоставляет контейнер для его отображения.

У фрагмента есть собственный жизненный цикл, но он зависит от жизненного цикла Activity.

1. Конфигурации устройств и жизненный цикл ресурсов

Что такое конфигурация устройства?

- Набор характеристик устройства: ориентация экрана, плотность пикселей, размер экрана, язык, тип клавиатуры и др.

- При изменении конфигурации (например, поворот экрана) Android уничтожает и пересоздаёт Activity, чтобы загрузить наиболее подходящие ресурсы.

Альтернативные ресурсы

- Android поддерживает квалификаторы ресурсов (например, -land для альбомной ориентации).

- Пример:

  - res/layout/activity\_main.xml — макет по умолчанию (портрет).

  - res/layout-land/activity\_main.xml — макет для альбомной ориентации.

- Файлы должны иметь одинаковые имена, но находиться в разных папках с квалификаторами.

- При повороте система автоматически выбирает нужный макет.

Это позволяет адаптировать UI под разные состояния устройства без изменения кода.

Snackbar — современная замена Toast

Что такое Snackbar?

- Виджет из библиотеки Material Design (`com.google.android.material:material`).

- Отображается внизу экрана, занимает всю ширину.

- Может содержать текстовое сообщение + одну кнопку действия.

- Пользователь может вручную закрыть Snackbar (в отличие от Toast).

3. Создание и отображение Snackbar

Базовое использование (Kotlin)

kotlin

Snackbar.make(view, "Вы нажали кнопку", Snackbar.LENGTH\_LONG).show()

Параметры make():

1. view — корневой View или любой дочерний элемент (лучше использовать CoordinatorLayout).

2. Сообщение — строка.

3. Длительность:

   - Snackbar.LENGTH\_SHORT (~2 сек)

   - Snackbar.LENGTH\_LONG (~3.5 сек)

   - Snackbar.LENGTH\_INDEFINITE — пока не закроют

Важно: Snackbar прикрепляется к переданному View. Для корректной работы рекомендуется использовать CoordinatorLayout как корневой контейнер.

Добавление кнопки действия

kotlin

val snackbar = Snackbar.make(view, "Действие выполнено", Snackbar.LENGTH\_LONG)

snackbar.setAction("Отменить") {

    // Обработка нажатия

}

snackbar.show()

Настройка внешнего вида

- Цвет текста кнопки:

  kotlin

  snackbar.setActionTextColor(Color.BLUE)

Рекомендации по структуре проекта

- При использовании \*\*Basic Activity:

  - activity\_main.xml — содержит CoordinatorLayout, Toolbar, FloatingActionButton.

  - content\_main.xml — содержит основной контент (например, кнопки).

- Кнопки лучше размещать в content\_main.xml, а Snackbar вызывать из MainActivity.

5. Поддержка портретной ориентации

- По умолчанию макет в res/layout/ используется для портретной ориентации.

- Для альбомной ориентации создаётся отдельный макет в res/layout-land/.

- Оба макета должны содержать одинаковые ID виджетов, чтобы код работал одинаково в обеих ориентациях.

При повороте состояние Snackbar не сохраняется автоматически — его нужно воссоздавать при необходимости.