01

01-b. TabLayout

androidx 뷰 활용

12-2 탭 레이아웃 – 탭 버튼 구성

- 탭 레이아웃은 탭tab으로 구분하는 화면에서 탭 버튼을 배치하는 레이아웃

12-2 탭 레이아웃 – 탭 버튼 구성

• 탭 레이아웃 등록

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <FrameLayout
        android:id="@+id/tabContent"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```

• 코드에서 탭 버튼 정의

```
binding.tabs.run {
    val tab1: TabLayout.Tab = newTab()
    tab1.text="Tab1"
    addTab(tab1)

    val tab2: TabLayout.Tab = newTab()
    tab2.text="Tab2"
    addTab(tab2)

    val tab3: TabLayout.Tab = newTab()
    tab3.text="Tab3"
    addTab(tab3)
}
```

12-2 탭 레이아웃 – 탭 버튼 구성

- 탭 버튼을 코드에서 정의하지 않고 레이아웃 XML 파일의 TabItem으로 정의

• XML 파일에서 탭 버튼 정의

```
<com.google.android.material.tabs.TabLayout
    android:id="@+id/tabs"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <com.google.android.material.tabs.TabItem
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tab1" />
    <com.google.android.material.tabs.TabItem
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tab2" />
    <com.google.android.material.tabs.TabItem
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tab3" />
</com.google.android.material.tabs.TabLayout>
```

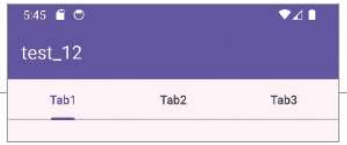
12-2 탭 레이아웃 – 탭 버튼 구성

- 탭 버튼의 이벤트 핸들러

• 탭 버튼 이벤트 처리

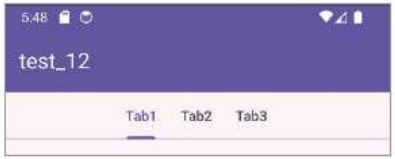
```
binding.tabs.addOnTabSelectedListener(object: TabLayout.OnTabSelectedListener {
    // 탭 버튼을 선택할 때 이벤트
    override fun onTabSelected(tab: TabLayout.Tab?) {
        val transaction = supportFragmentManager.beginTransaction()
        when (tab?.text) {
            "Tab1"-> transaction.replace(R.id.tabContent, OneFragment())
            "Tab2"-> transaction.replace(R.id.tabContent, TwoFragment())
            "Tab3"-> transaction.replace(R.id.tabContent, ThreeFragment())
        }
        transaction.commit()
    }
    // 선택된 탭 버튼을 다시 선택할 때 이벤트
    override fun onTabReselected(tab: TabLayout.Tab?) {

    }
    // 다른 탭 버튼을 눌러 선택된 탭 버튼이 해제될 때 이벤트
    override fun onTabUnselected(tab: TabLayout.Tab?) {

    }
})
```

▶ 실행 결과



12-2 탭 레이아웃 – 탭 버튼 구성

- 탭 버튼 정렬하기
  - tabGravity는 탭 버튼을 정렬하는 속성



- 스크롤 설정하기
  - tabMode 속성은 탭 버튼을 스크롤할 수 있는지를 설정
  - fixed은 스크롤을 지원하지 않는다는 의미
  - scrollable로 설정하면 탭 버튼이 왼쪽부터 나열되고 모두 출력할 수 없다면 자동으로 가로 스크롤

12-2 탭 레이아웃 – 탭 버튼 구성

- 뷰 페이저 연동하기
  - TabLayout과 ViewPager2를 등록한 후 코드에서 TabLayoutMediator를 이용해 둘을 연동

• 탭 레이아웃과 뷰 페이저 등록

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical">
    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:tabMode="scrollable">
    </com.google.android.material.tabs.TabLayout>
    <androidx.viewpager2.widget.ViewPager2
        android:id="@+id/viewpager"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```

• 탭 레이아웃과 뷰 페이저 연동

```
TabLayoutMediator(tabLayout, viewPager) { tab, position ->
    tab.text = "Tab${(position + 1)}"
}.attach()
```

감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.

윌리엄 셰익스피어
William Shakespeare