

Modul 3 – Android App Entwicklung mit Kotlin

Wiederholung

Datum: 30.05.2022

Gliederung

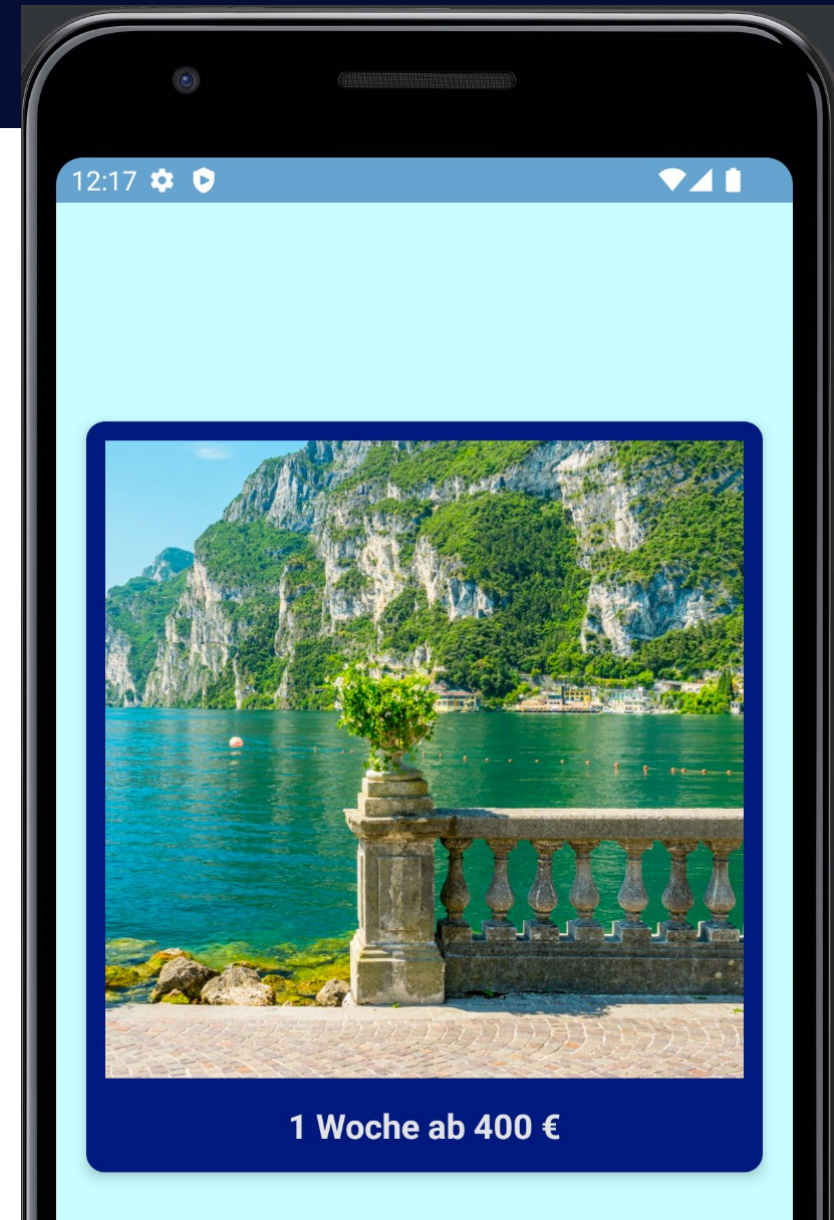
- Was haben wir bis jetzt gelernt?
- Was kommt noch?
- Offene Fragen
- Euer Feedback



Quelle: <https://innovationbydesign.pressbooks.com/chapter/ideate/>

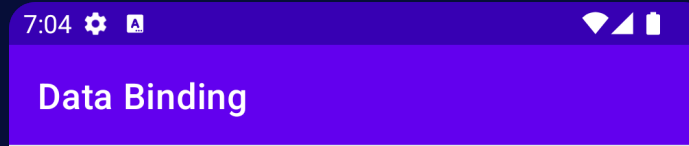
Activities und Intents

- Intents
- Unterschied implizit und explizit
- DetailActivity
- ShareIntent



Activity Lifecycle

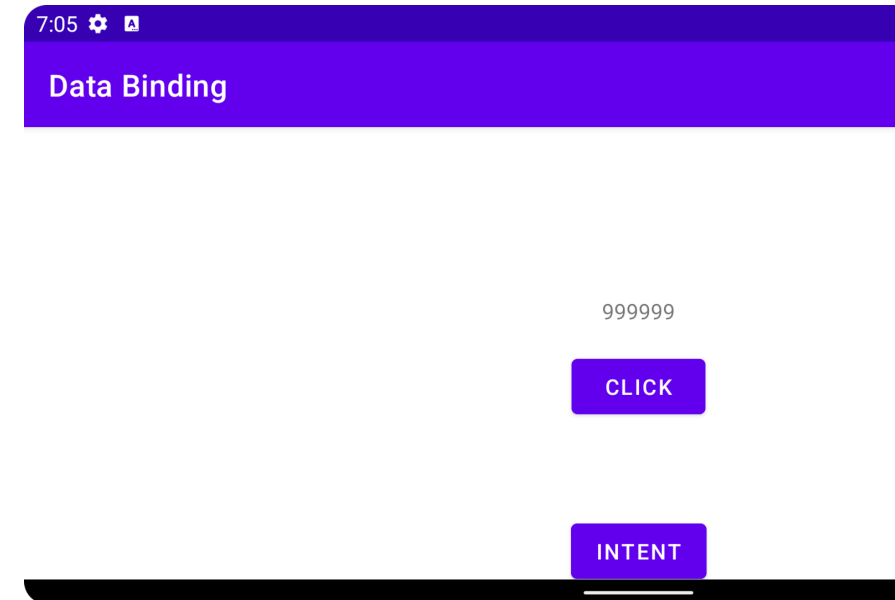
- Activity Lifecycle
- Zustände und Events
- SaveInstanceState



999999

CLICK

INTENT



Fragmente

- Fragmente und Lifecycle
- Navigation Component



Navigation Tests

- Fragmente und Navigation Component
- Instrumented Tests
- Navigation Test

```
@RunWith(JUnit4::class)
class NavigationTest {

    lateinit var navController: TestNavHostController

    lateinit var offerListScenario: FragmentScenario<HomeFragment>

    @Before
    fun setup() {
        // Test NavController wird bereitgestellt
        navController = TestNavHostController(ApplicationProvider.getApplicationContext())

        // Das HomeFragment wird in die MainActivity geladen
        offerListScenario = launchFragmentInContainer<HomeFragment>(themeResId = R.style.Theme_RecyclerView)

        // Der NavGraph in den NavController geladen und der NavController ins Fragment
        offerListScenario.onFragment { it: HomeFragment
            navController.setGraph(R.navigation.nav_graph)
            Navigation.setViewNavController(it.requireView(), navController)
        }
    }

    @Test
    fun navigateToOffer() = runBlocking { this: CoroutineScope

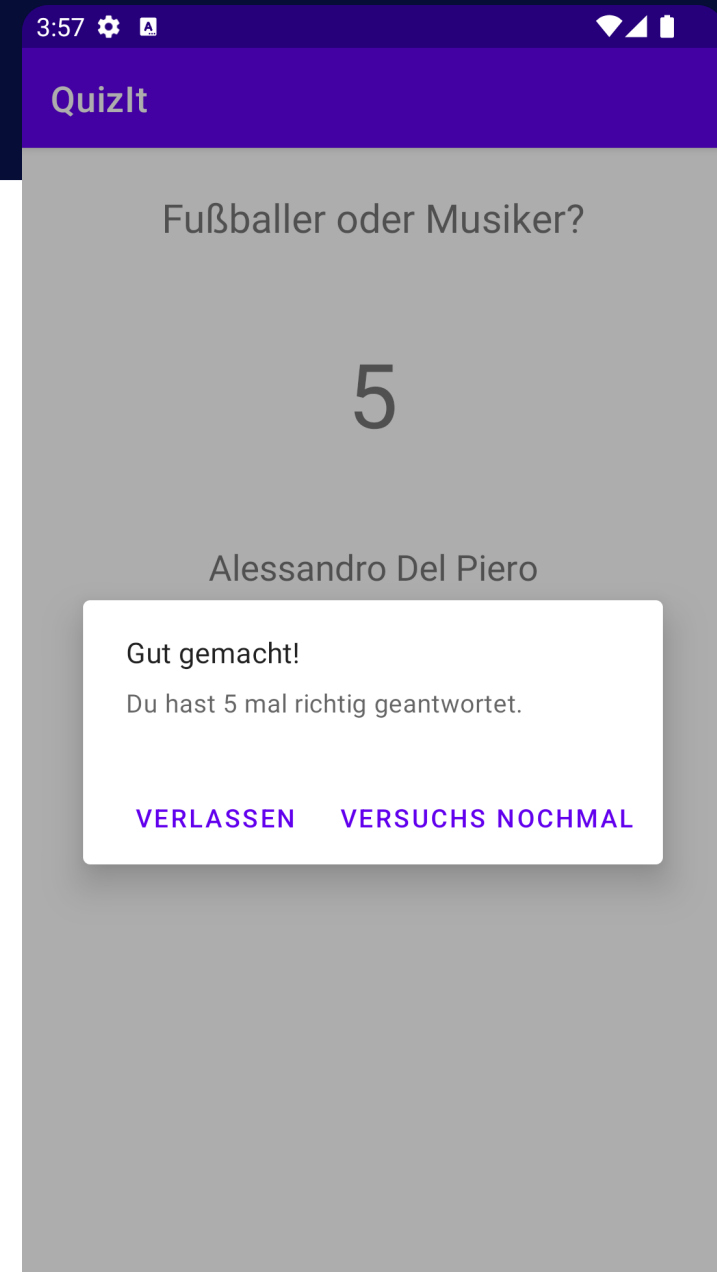
        // GIVEN - RecyclerView mit Angeboten

        // WHEN - Klick auf Angebot 10
        onView(withId(R.id.recyclerViewHome))
            .perform(
                RecyclerViewActions
                    .actionOnItemAtPosition<RecyclerView.ViewHolder>(position: 10, click())
            )

        // THEN - NavController navigiert zum DetailFragment
        assertEquals(navController.currentDestination?.id, R.id.detailFragment)
    }
}
```

Architektur und ViewModel

- Architekturprinzipien
- MVVM
- ViewModel



Live Data

- Live Data
- Einsatz in MVVM
- Data Binding im Layout

Antonio
Vivaldi

Franco

Bellini

VERSUCHS NOCHMAL

VERLASSEN

Andrea
Pirlo

Shared ViewModel

- Shared ViewModel
- Backstack und BackButton
- ViewModel Tests

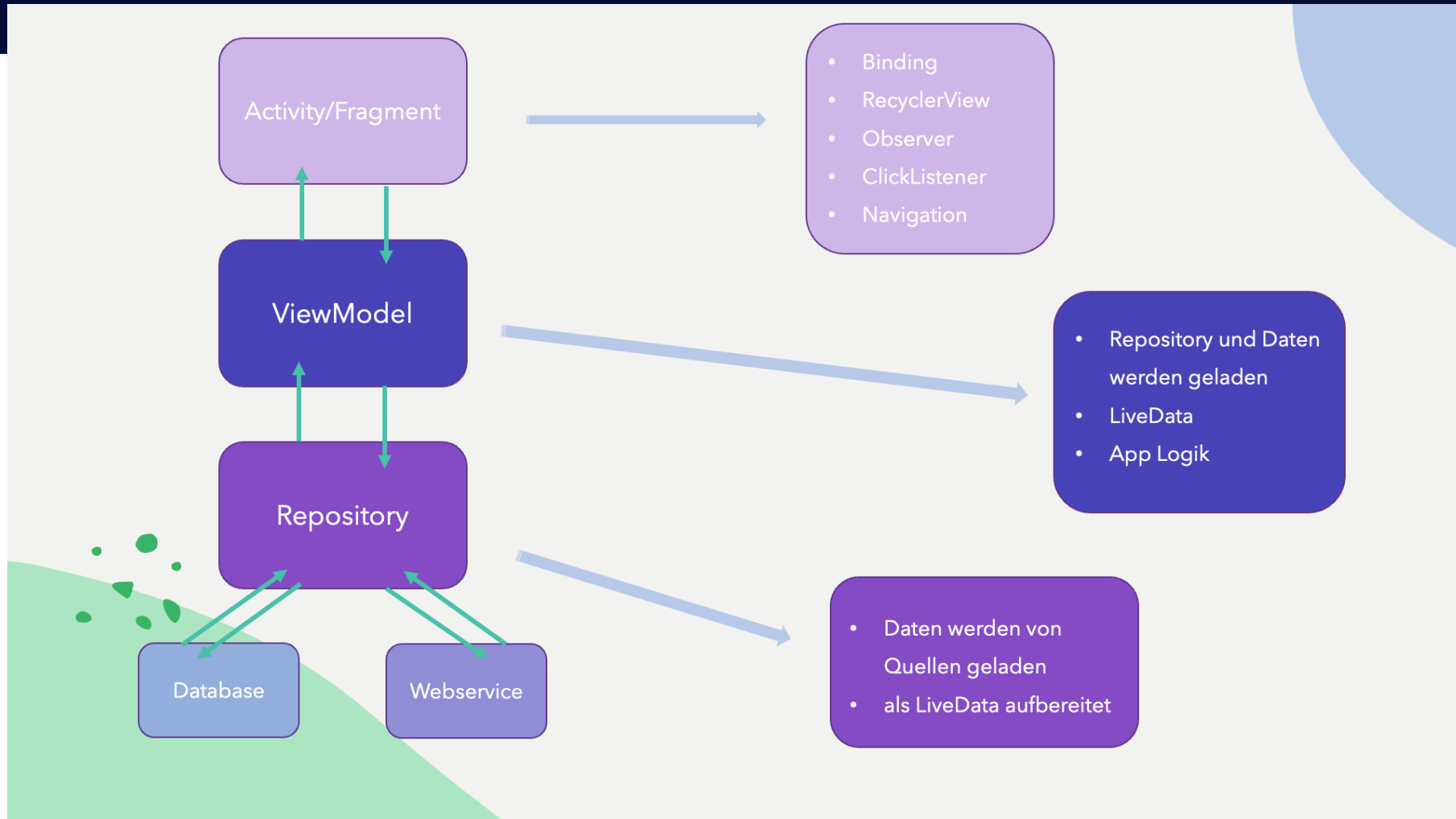
5:39

← Ergebnis

Andrea

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    val navHostFragment = supportFragmentManager  
        .findFragmentById(R.id.nav_host_fragment) as NavHostFragment  
    navController = navHostFragment.navController  
  
    setupActionBarWithNavController(navController)  
}  
  
override fun onSupportNavigateUp(): Boolean {  
    return navController.navigateUp() || super.onSupportNavigateUp()  
}
```

MVVM



Was kommt noch?

- Coroutines
- Webservices und API Calls
- Caching Data



Quelle: <https://schulschwestern.de/wp-content/uploads/2022/01/Bild1-1.jpg>

Offene Fragen



Quelle: <https://healthblog.uofmhealth.org/cancer-care/what-did-my-doctor-say-what-to-do-when-you-dont-understand>

Feedback



Quelle: <https://mediamakeameme.org/created/accept-feedback.jpg>



Gut gemacht!

Quelle: <https://www.rawpixel.com/search/fist%20bump?page=1&sort=curated>