**FastRxJava 4기 Syllabus**

**강의제목:** 프로젝트로 정복하는 RxJava 4기

**강의개요**: RxJava2를 기초개념부터 활용까지 이론강의와 더불어 실전 예제를 통해 익혀갑니다. RxJava2를 학습하면서 필요한 프로그래밍 기초지식도 함께 배웁니다.

**시간**: 3/21~4/25 (4/4 휴강) 5주 코스 매주 토요일 오후 2-5시 3시간씩 진행

**장소**: 강남 미왕빌딩 10층 F 강의실

**사전지식:**

* Kotlin 언어
* Activity, Fragment, 생명주기 등 안드로이드 기본지식
* 안드로이드 커스텀 리스트 UI 구현 지식
* Retrofit, Glide 등 라이브러리를 사용하여 문서를 보고 API를 받아올 수 있는 분

**강의목표**:

* ReactiveX 패러다임 이해와 활용
* RxJava2의 스트림, Observable, Operator, 스캐쥴러, 에러처리, 메모리 관리와 같은 기본 이론 이해
* RxJava2 실전 프로그래밍 경험
* RxJava2를 활용하기 위한 프로그래밍 기초지식 학습

**커뮤니케이션 채널**: https://open.kakao.com/o/gVp3UW2b

**Github Repo**: <https://github.com/maryangmin/fastcampus-rxjava2>

**강의방식**:

* 과제 없습니다.
* 강의 중 질문은 익명 질문 플랫폼 sli.do 활용합니다. 강의 전 오픈채팅으로 링크 전달

**강의일정**:

|  |  |
| --- | --- |
| 주차 | 강의 |
| 1 | 1. 강의 OT 2. ReactiveX 프로그래밍 개요  * 비동기 프로그래밍   + 프로그램의 Client-Server-DB 요청 흐름   + Callback 방식   + Event 방식 * 디자인 패턴   + OOP(Object Oriented Programing) 개념   + 디자인 패턴 개념   + 아키텍쳐 개념과 디자인 패턴과의 차이   + Observer 패턴, Iterator 패턴 * Observable, Observer  1. 함수형 프로그래밍 개요  * 명령형 프로그래밍 * 스트림 개념 * Operator 개념  1. Observable와 Observer  * Single, Maybe, Completable 등 Observable Type * Observer의 onNext, onError 등 method들  1. Operator  * map, flatmap, filter 등 핵심 operator * 다양한 Operator를 스스로 공부할 수 있는 문서 보는 방법 * RxJava는 이벤트 기반 비동기 프로그래밍 라이브러리 입니다. 이벤트 기반은 무엇인지, 비동기 프로그래밍은 무엇인지, RxJava 활용법을 관통하는 함수형 프로그래밍은 무엇인지, 라이브러리를 이해하기 위한 주요 기본 개념들을 알아봅니다. * RxJava 설계에 활용된 디자인 패턴을 설명합니다. 디자인 패턴이 무엇인지, OOP가 무엇인지, 설명을 이해하기 위한 기본 개념들을 함께 다룹니다. * RxJava를 구성하는 가장 기본 클래스인 Observable, Observer, Operator을 배웁니다. RxJava의 모든 시작과 활용은 위 3개의 클래스로 이어집니다.   프로젝트   1. Retrofit Call을 RxJava로 바꾸기 2. ViewModel에서 API를 호출하고, RxJava로 UI까지 잇기 3. API 응답 데이터 필터링 해보기 |
| 2 | 1. Cold & Hot Observable  * Cold & Hot Observable 개념 * Cold에서 Hot 만들기. ConnectableObservable 개념 * Publish, Behaivor 등 Subject 종류 * subject 활용해서 RxBus 만들기   + EventBus 개념   + RxBus 구현   + RxBus 활용 Case     - RxBus로 Data 전달하기     - RxBus 주의할 점     - interface, onActivityResult() 등 Data 전달 방법  1. Backpressure  * Backpressure 개념 * Flowable과 Buffer, Drop 등 Backpressure 전략 * Operator로 Backpressure 처리하기  1. RxJava2로 쓰레드 관리하기  * 프로세스와 쓰레드 개념 * 쓰레드 풀 개념 * Java ExecutorService 개념 * RxJava2 스캐쥴러 개념 및 io, computaion 등 종류 * 안드로이드 메인 쓰레드 개념 * subscribeOn, observeOn 이해 및 쓰레드 활용 * 구독할 때 이벤트를 받는 Cold Observable와, 언제나 이벤트를 받는 Hot Observable 2가지 종류의 Observable에 대해 알아봅니다. Hot Observable 실무 활용 케이스를 설명합니다. * 매우 많은 이벤트를 처리할 때 RxJava는 어떻게 관리할까요? Backpressure를 활용하여 많은 수의 이벤트를 관리하는 방법을 설명합니다. * RxJava의 가장 큰 장점 중 하나인, Operator을 활용한 쉬운 쓰레드 관리에 대해 알아봅니다. 쓰레드가 무엇인지, 기본 개념부터 차근차근 설명합니다.   프로젝트   1. 여러개의 체크박스 동작 stream으로 이어보기 2. zip으로 여러 api 이어보기 |
| 3 | 1. 구독(subscribe) 동작 원리 1 (RxJava2 내부코드 분석)  * Observable.just()/create() 활용 및 이해 * upstream, downstream 이해 및 코드단위 분석 * operator chaining 이해 * 디자인 패턴   + Decorator 패턴 * RxJava는 어떻게 동작할까요? 단순히 사용해왔던 메소드들의 동작원리를 디자인 패턴부터 소스코드 단위까지 깊이 분석해봅니다. * 동작을 상세하게 분석하여 얻은 이해를 바탕으로, 기존에는 애매하게 다가왔던 RxJava의 동작을 정확하게 예상해봅시다.   프로젝트   1. RxJava2로 쓰레드 관리 문제풀이 2. 이슈상세 기능 구현하기 |
| 4 | 1. 메모리 관리  * 메모리 누수의 정의   + 오브젝트 레퍼런스     - 의존성 개념     - OOP Tip. is 관계 / has 관계   + 가비지 컬렉터   + 메모리 누수가 발생하는 이론적인 상황   + 안드로이드의 전형적인 메모리 누수 Case   + RxJava2의 전형적인 메모리 누수 Case * 메모리 누수 방지 방법   + dispose 이해   + CompositeDisposable 활용 * Single, Completable 자동 dispose 설명   + 메모리 관리 dispose 활용 의미 이해  1. 구독(subscribe) 동작 원리 2 (RxJava2 내부코드 분석)  * 자동 dispose 원리 이해   + fromCallable() 이해   + Single, Completable 속성 심화 이해  1. Exception 처리  * Observer.onError() 개념 * onError() 처리 Best Practice   + 스트림 관리 Best Practice   + RxBus 활용해서 에러처리하기 * 종료된 에러 스트림 이어가기 * RxJavaPlugin 활용 * RxJava를 잘못 다룸으로써 발생할 수 있는 메모리 누수가 언제 발생하는지, 왜 발생하는지, 어떻게 관리할 수 있을지 알아봅니다. 메모리 누수가 무엇을 의미하는지에 대한 기본 개념을 함께 설명합니다. * 좋은 서비스를 만들기위해서는 Exception을 잘 처리해야 합니다. RxJava는 어떻게 Exception을 관리하는지 살펴봅니다.   프로젝트   1. CompositeDisposable로 BaseActivity에서 stream들 dispose 해보기 2. Exception을 다른 Exception으로 Wrapping 하기 3. BaseObserver와 RxJavaPlugin으로 DefaultErrorHandler 만들기 |
| 5 | 1. 테스트  * 소프트웨어 테스트 개념 * assert, verify, mocking 등 테스트 구현 개념 * Testable Architecture   + MVP / MVVM * RxJava2 테스트 구현 * RxJava2 테스트를 위한 스캐쥴러 변경   + 쓰레드 동작 복습   + 테스트 스캐쥴러 활용   + RxJavaPlugin 활용  1. RxBinding UI 처리 2. Room with Rx  * Database 개념 * 테이블, column, row, 쿼리 개념 * 관계형 DB, nosql 개념 * ORM 개념 * Room 개념 및 활용 * Room을 Rx로 활용하기 * 좋은 소프트웨어를 만들기 위해서는 테스트가 필수입니다. 테스트가 왜 필요한지, 어떻게 구현하는지 등 테스트에 대한 기본 개념부터, RxJava는 어떻게 테스트 하는지, 무엇을 주의해야할지에 대해 설명합니다. * RxJava는 단순히 하나의 라이브러리가 아니라 구글이 공식적으로 지원하는 안드로이드 앱 개발의 필수 도구입니다. RxBinding, Room 등 다른 라이브러리와 함께 활용하여 UI와 DB 처리를 RxJava로 쉽게 풀어낼 수 방법을 설명합니다.   프로젝트   1. 테스트 구현하기 2. UI Interaction을 Rx로 처리하기 |

\*수강생분들의 수준에 따라 난이도 조정 및 순서변경이 있을 수 있습니다

**환경**:

* 노트북 (OS 무관)
* Android Studio (버전 무관. 최신을 추천.)
* Git, Github 연동

**강사정보**: 이승민 / rfrost77@gmail.com

**강사이력**: 7년차 현업 안드로이드 개발자

* (현) 뱅크샐러드 Frontend Team Lead
* (현) GDE (Google Developers Experts) Android Korea
* (전) 드라마앤컴퍼니 안드로이드 개발자 - 명함관리 앱 리멤버 개발
* <https://www.linkedin.com/in/maryangmin/>