Kotlin Flow

😵 비동기

🥴 코루틴

1. Kotlin Flow

- Kotlin Flow는 Kotlin에서 **비동기적인 데이터 스트림을 처리하기 위한 라이브러리**
- 이는 Kotlin의 코루틴을 기반으로 하여 비동기적인 작업을 보다 효율적으로 다룰 수 있도록 도와줌
- Kotlin Flow는 비동기적인 작업을 연속적으로 처리하고, 데이터 스트림을 처리하는데 사용됨

1-1) 특징

1. **비동기적인 데이터 스트림 처리**
    - Kotlin Flow를 사용하면 데이터를 비동기적으로 생성하고, 전달하며, 처리할 수 있음
    - 이는 네트워크 호출, 파일 I/O 작업, 데이터베이스 쿼리 등과 같은 비동기 작업을 처리하는 데 유용
2. **선언적인 API**
    - Kotlin Flow는 선언적은 API를 제공하여 비동기 코드를 간결하게 작성할 수 있도록 함
    - 데이터를 처리하고 조작하기 위한 다양한 연산자들(예: map, filter, reduce 등)을 제공하여 고드의 가독성을 향상시킴
3. **취소 및 에외 처리**

- Kotlin Flow는 코루틴의 취소 및 여외 처리 메커니즘과 함께 동작함
- 이를 통해 안전하고 예외 상황에 대처할 수 있는 비동기 코드를 작성할 수 있음

4. **데이터 흐름 제어**
    - Kotlin Flow를 사용하여 데이터 흐름을 제어할 수 있음
    - 데이터 소스의 속도를 제어하거나 데이터를 버퍼링 하거나 데이터 흐름을 중지하고 다시 시작하는 등의 작업을 수행할 수 있음

1-2) 예시 코드

- Flow를 사용하여 1~5까지의 숫자를 생성하고 각 숫자를 두 배로 변환하는 작업 수행

```
import kotlinx.coroutines.flow.Flow
import kotlinx.coroutines.flow.flow
import kotlinx.coroutines.flow.collect
import kotlinx.coroutines.runBlocking

// Flow를 생성하는 함수
fun simpleFlow(): Flow<Int> = flow {
        // Flow에서 값을 순차적으로 보낼 수 있음
        for (i in 1..5){
                emit(i) // 숫자를 Flow에 보냄
        }
}

fun main() = runBlocking<Unit> {
        // Flow를 수집하고 처리하는 코드
        simpleFlow().collect { value ->
            // 각 숫자를 두 배로 변환하여 출력
            println("결과 : ${value * 2}")
    }
}
```

▼ 코드 설명

- `simpleFlow` 함수 : Flow 생성, Flow 빌더 함수인 `flow` 를 사용하여 Flow를 정의하고, 숫자 1부터 5까지를 Flow에 보냄
- `collect` 함수 : Flow를 수집하여 각 요소를 처리. collect 함수는 suspending함수이므로, runBlocking 빌더를 사용하여 메인 스레드에서 실행될 수 있도록 함

- `collect` 함수 내부 : 각 요소를 받아서 두 배로 변환 후 출력

▼ 실행 결과

```
결과 : 2
결과 : 4
결과 : 6
결과 : 8
결과 : 10
```

결론

- Kotlin Flow는 Kotlin에서 비동기적인 데이터 스트림을 처리하기 위한 강력하고 유연한 라이브러리로,

  코루틴과 함께 사용하여 비동기 작업을 보다 쉽게 다룰 수 있도록 지원