01-5. Channel

Coroutine

Channel

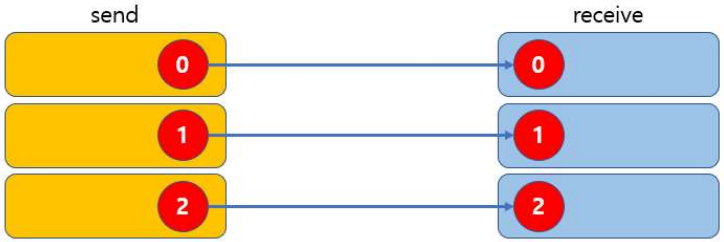
- 코루틴이 실행되면서 발행하는 데이터를 반복적으로 구독하기 위한 방법은 Channel 과 Flow
- Channel 은 데이터를 주고 받기 위한 연결 통로
- Channel 자체가 코루틴이 아니며 단지 코루틴 간의 데이터를 주고 받는 방법
- 데이터를 발행하는 곳을 위한 SendChannel 인터페이스와 데이터를 구독 곳을 위한 ReceiveChannel 인터페이스가 있는데 이 둘을 상속받아 정의한 Channel 인터페이스 타입으로 채널을 이용

```
val channel = Channel<Int>()
……
channel.send(0)
---
channel.consumeEach { }
```

버퍼 이용 - capacity

- Channel 을 이용해 데이터를 발행 하고자 할 때 구독이 준비 안되었을 수도 있다.
- send() 는 어디선가 데이터를 구독 될 때까지 대기상태가 되는 것이고, receive() 는 어디선가 데이터가 발행될 때까지 대기상태가 된다.

- Channe() 함수의 기본 capacity 설정이 RENDEZVOUS 로 되어 있으며 RENDEZVOUS 는 capacity 를 0으로 설정



버퍼 이용 - capacity

- Channel() 함수를 이용하면서 capacity 를 조정하여 채널의 버퍼 사이즈를 조정
- 버퍼 사이즈는 숫자 값으로 지정하거나 아래의 상수 변수로 지정

  - BUFFERED : 버퍼 사이즈 64
  - CONFLATED : 버퍼 사이즈 1, 새로운 데이터가 이전 데이터 대체
  - RENDEZVOUS : 버퍼 사이즈 0
  - UNLIMITED : 수용량에 제한이 없다.

```
Channel<Int>(capacity = 2)
```

버퍼 이용 – onBufferOverflow

- 버퍼가 가득 찼을 때 어떻게 할 것인지 onBufferOverflow 로 제어

  - SUSPEND : 기본 값, send 대기
  - DROP_OLDEST : 가장 오래된 데이터 제거
  - DROP_LATEST : 가장 최신 데이터 제거

```
Channel<Int>( onBufferOverflow = BufferOverflow.DROP_OLDEST )
```

produce()

- produce() 는 코루틴 빌더
- 다른 코루틴 빌더와 차이점은 채널을 이용하기 위해 만들어진 코루틴 빌더

- produce() 에 의해 만들어진 코루틴은 ProducerScope 에 담겨서 실행
- ProducerScope 는 CoroutineScope 를 구현한 것 뿐만 아니라 SendChannel 을 구현
- produce { } 에서는 별도의 Channel 객체를 준비하지 않아도 SendChannel 의 send() 함수를 이용해 데이터를 발행

- produce() 의 리턴 타입은 ReceiveChannel

actor()

- actor() 는 produce() 와 마찬가지로 채널을 이용하기 위한 코루틴 빌더
- actor() 는 CoroutineScope 와 ReceiveChannel 을 구현한 ActorScope 에서 코루틴이 실행
- actor() 의 리턴 타입은 SendChannel
- 외부에서 채널을 통해 데이터를 발행하면 그 데이터를 구독해서 동작하는 코루틴을 만들 때 사용

```
val sendChannel = actor<Int> {
    consumeEach {

    }
}
```

Channel Pipeline

- 채널 파이프라인이란 여러 개의 채널을 연결해서 사용하는 기법
- 하나의 채널에서 발행하는 데이터를 다른 채널에 넘겨서 그곳에서 이용하게 하는 기법

```
fun CoroutineScope.somethingB(receiveChannel: ReceiveChannel<Int>) = produce<Int>{
    ……..
}
```

Fan in, out

- 하나의 채널에서 발행하는 데이터를 여러 코루틴에서 구독하는 것을 Fan-out
- 여러 코루틴에서 데이터를 발행시키는 것을 하나로 모아서 처리하는 것을 Fan-in
- 특별한 기법이라기 보다는 용어이다.

감사합니다

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

윌리엄 셰익스피어
William Shakespeare