

# TDD 시험 안내

평가 내용 : TDD 를 적용한 TV Channel Controller 구현  
배점 : 120점/200점

# 시험 준비 – TDD TV

---

- Repository : TDD\_TV
- 평가 코드는 Org 내 Repo 생성, 구현 진행
  - Repository : 과정명\_실습명\_개인번호 (ex: TDD\_TV\_01, TDD\_TV\_02,...)
  - description : Author 성명 -> ex: Author : 주연임
  - dev branch 생성하여 dev branch 로 commit push

# 시험 설명 – TDD TV 채널 관리 모듈

- 셋탑 박스에서 리모컨으로부터 입력을 받아 채널을 관리하고, 튜너를 Set 하는 모듈을 만들고자 한다.
  - 기능  
: 0~9의 숫자 값으로 채널 이동, 이전 채널, 채널 업/다운, 채널 검색, 선호 채널 추가, 다음 선호 채널.
  - 채널은 0~99 까지 있다
  - Tuner 업체에서 Channel tuner 모듈을 제공한다
    - seekCH() : 현재 채널에서 숫자가 증가하는 방향으로 시청 가능한 채널을 검색하여, 채널을 검색된 채널로 변경하고 검색된 채널 값을 반환한다.
    - setCH(ch) : 지정된 채널값으로 현재 채널을 변경한다.
    - getCurrentCH() : 현재 설정된 채널값을 반환한다.
  - 본 실습에서 Tuner는 입력에 대해 정확하게 동작한다고 가정하며, Tuner는 만들어야 하는 모듈의 DOC(Depend On Component)일 뿐, Test를 만들고 동작을 확인해야 하는 대상은 아니다. (TunerTest.java 참조)
- TDD 를 적용하여, 잘 구조화된 Controller 모듈을 구현하세요.

# 시험 설명 – TDD TV 채널 관리 모듈

- 숫자 버튼 : 숫자 값에 의한 채널 변경
  - 리모컨 '1'을 누르고 '확인' 을 누르면 1번 채널로 변경된다
  - 리모컨 '1'을 누르고 '2' 를 누르면 12번 채널로 변경된다.
  - '1','2','3','4' 를 연속적으로 누를 경우,  
12번 채널로 변경되었다가 34번 채널로 변경된다.
  - '4','5','6' 을 연속적으로 누를 경우: 45번 채널로 변경
    - 이후 숫자와 확인 버튼을 누르면 해당 숫자의 번호로 이동(6번 또는 6\_번)
    - 그 외의 버튼을 누르면 6 은 무효화
  - '0','7' 를 누를 경우 7번 채널로 변경된다.

# 시험 설명 – TDD TV 채널 관리 모듈

- 이전 채널 : 바로 이전 채널로 변경
  - 리모컨 이전 채널을 누르면 이전에 채널로 변경된다.

4->64->12 의 순서로 채널을 변경한 경우,

"이전 채널" 을 누르면 64로 변경되고,

다시 "이전 채널 " 을 누르면 12로 변경된다.(이후에는 64 <-> 12 를 반복한다.)

- 이전 채널이 없는 경우 채널이 변경되지 않는다

# 시험 설명 – TDD TV 채널 관리 모듈

- 선호 채널 : 선호 채널 목록에 추가/삭제
  - 채널을 시청 중에 선호채널을 누르는 경우
    - 선호 채널이 아닌 채널은 선호 채널에 저장된다
    - 이미 선호 채널인 채널은 선호채널에서 삭제된다
    - '12' 번 채널 상태에서 '선호 채널'을 누르면 12 를 선호 채널 목록에 추가된다.  
만약 '12' 번이 선호 채널 목록에 있다면, 선호 채널 목록에서 삭제한다.
    - '12'->선호채널->'08'->선호채널->'37'->선호채널->'08'->선호채널->'06'->선호채널  
의 순서로 버튼을 누를 경우, '6, 12, 37' 이 선호 채널에 저장된다.

# 시험 설명 – TDD TV 채널 관리 모듈

- 다음 선호 채널 : 선호 채널 목록 내 이동
  - 다음선호채널을 누르면, 선호 채널 목록 내에서 현재 채널보다 큰 값 중 가장 작은 채널로 변경된다.
    - 1, 4, 12, 56 이 선호 채널에 저장되어 있는 경우,
    - 6번 채널을 시청 중 다음선호채널을 누르면 12번 채널로 변경된다
    - 56번 채널 시청 중 다음선호채널을 누르면 1번 채널로 변경된다
  - 선호 채널 목록이 없는 상태에서 다음선호채널 버튼을 누르면, 채널은 변경되지 않는다.

# 시험 설명 – TDD TV 채널 관리 모듈

- 채널 검색 : 시청 가능한 채널 검색
  - 리모컨 채널 검색을 누르면, 시청 가능한 모든 채널을 검색하여 저장한다.
  - 채널의 검색은 Tuner 의 seekCH() 의 반복 호출 결과를 저장한다.
  - Tuner.seekCH() 의 호출로 찾은 채널 값들을 저장한다.
    - Tuner.seekCH() 은 현재 채널값 이후 채널 중 시청 가능한 하나의 채널값만 반환한다.
    - Tuner.getCurrentCH() = '78' 이고, 그 다음으로 시청 가능한 채널이 '83' 이라면, Tuner.seekCH() 은 '83' 을 반환하고 현재의 채널을 '83' 으로 변경한다.



# 시험 설명 – TDD TV 채널 관리 모듈

- 채널 업/다운에 관한 요구 사항
  - 채널 검색에 의해 저장된 채널 목록이 없는 경우
    - 현재 채널이 6인 경우, 채널 업은 7 로 변경되며, 채널 다운은 5 로 변경
    - 현재 채널이 99인 경우, 채널 업은 0 으로 변경되며, 채널 다운은 98 로 변경
    - 현재 채널이 0인 경우, 채널 업은 1 로 변경되며, 채널 다운은 99 로 변경
  - 채널 검색에 의해 저장된 채널 목록이 있는 경우
    - 채널 업/다운을 누르면 저장된 채널 목록 중 다음/이전 채널로 가야 한다.
    - 저장된 채널이 4, 6, 14, 93 일 때.  
현재 채널이 6인 경우 채널 업을 누르면 14로 변경, 다운을 누르면 4로 변경.  
현재 채널이 95인 경우 채널 업을 누르면 4로 변경, 다운을 누르면 93으로 변경.
    - 현재 채널이 0 인 경우 채널 업을 누르면 4로, 다운을 누르면 93으로 변경.

# 시험 설명 – TDD TV 채널 관리 모듈

- 개인별 Test
  - 개인별 하나의 repository 생성
  - 주어진 소스를 바탕으로, TDD 를 적용하여 TDD\_TV 구현.
  - 가능한 TDD 가 적용되는 중간 과정이 잘 드러나게 commit-push 해 주세요.
- 목표
  - Coverage가 높은 Test case 를 작성할 수 있다.
  - Test case 를 효과적으로 작성할 수 있다.
  - 작성한 module(SUT)가 DOC인 Tuner를 정확하게 동작시키는 것을 확인할 수 있다.
  - 시간이 오래 걸리는 Tuner 의 동작에 대해 빠른 Test 를 만들 수 있다

# TunerTest.java 1/4

---

```
class TunerTest {  
    Tuner tuner;  
  
    @BeforeEach  
    void setup(){  
        tuner = new Tuner();  
    }  
  
    @Test  
    @DisplayName("초기 채널값 확인 테스트")  
    void initChannel(){  
        int initCh = tuner.getCurrentCH();  
        assertTrue(initCh >=0 && initCh <=99);  
    }  
}
```

# TunerTest.java 2/4

//...이어서

```
@ParameterizedTest
@CsvSource({"0", "4", "5", "12", "99"})
@DisplayName("채널설정테스트 : 유효한 채널인 경우")
void testSetChForValidChannel(String channel) {
    tuner.setCH(channel);
    assertEquals(Integer.parseInt(channel), tuner.getCurrentCH());
}
```

```
@ParameterizedTest
@CsvSource({"-12", "-2", "-0", "100", "9999"})
@DisplayName("채널설정테스트 : 유효하지 않은 채널인 경우")
void testSetChForInvalidChannel(String channel) {
    assertThrows(IllegalArgumentException.class, () -> tuner.setCH(channel));
}
```

# TunerTest.java 3/4

```
//...이어서
```

```
@Test
```

```
@DisplayName("채널검색테스트 : 시작 채널로부터 10개 채널 검색 테스트")
```

```
void testSeekCh10times() {
```

```
    ArrayList<String> seekChannel = new ArrayList<>();
```

```
    for (int i=0; i<10; i++){
```

```
        String seekCh = tuner.seekCH();
```

```
        int ch = Integer.parseInt(seekCh);
```

```
        assertTrue(0<= ch && ch <=99);
```

```
        seekChannel.add(seekCh);
```

```
    }
```

```
    assertEquals(10, seekChannel.size());
```

```
    System.out.println(seekChannel.toString());
```

```
}
```

# TunerTest.java 4/4

```
//...이어서
```

```
@Test
```

```
@DisplayName("채널검색테스트 : 시작 채널값을 99로 지정하고 10개 채널 검색 테스트")
```

```
void testSeekCh10timesAfterSetCH() {
```

```
    ArrayList<String> seekChannel = new ArrayList<>();
```

```
    tuner.setCH("99");
```

```
    for (int i=0; i<10; i++){
```

```
        String seekCh = tuner.seekCH();
```

```
        int ch = Integer.parseInt(seekCh);
```

```
        assertTrue(0<= ch && ch <=99);
```

```
        seekChannel.add(seekCh);
```

```
    }
```

```
    assertEquals(10, seekChannel.size());
```

```
    System.out.println(seekChannel.toString());
```

```
}
```

```
}
```

수고하셨습니다.