

# “WELCOME BTCP THEATER”

- 나의 작고 소중한 첫 과제 -

개발 언어 : Java 11

개발 기간 : 2021.11.10 ~ 2021.11.12(총 3일)

새롭게 MVC 모델을 배우며 즐겁게 작업하였던 첫 과제였기 때문에 별거없지만 기록하여 남기고 싶었다..

## | 순서

1. 기획
2. 요구사항
3. 구현
4. 개발
5. 회고

## | 기획

- 주제 : 극장 영화 관리 시스템 만들기
- 사용자 대상 : 극장 영화 매니저, 영화 평론가, 영화 관람객

## | 요구사항 - [권한]

- 극장 영화 매니저 : 영화 관리 권한, 극장 관리 권한, 극장별 상영정보 관리 권한
- 영화 평론가 : 상영되고 있는 영화 리스트 확인, 등록된 극장 리스트 확인 가능하며 극장별로 상영되고 있는 영화 목록과 정보 확인 가능. 현재 극장에서 상영되고 있는 영화의 평점을 영화 평론가로서 매길 수 있음
- 영화 관람객 : 상영되고 있는 영화 리스트 확인, 등록된 극장 리스트 확인 가능하며 극장별로 상영되고 있는 영화 목록과 정보 확인 가능. 현재 극장에서 상영되고 있는 영화의 평점을 일반 관람객으로서 매길 수 있음

## | 요구사항 - [기능]

- 극장별 극장 정보와 현재 극장에서 상영되고 있는 영화 목록 리스트 확인
- 극장을 선택하면 해당 극장의 정보를 확인할 수 있음
- 영화를 선택하면 해당 영화의 상영 정보를 확인할 수 있음
- 영화별 영화 정보와 해당 영화가 상영되고 있는 극장, 영화 평점 정보, 감상평을 영화 평론가, 관람객 별로 확인

## | 구현

## [관리자]

## 1. 개봉 영화 등록

```
<WELCOME BTCP THEATER>
1.로그인 2.회원가입 3.프로그램 종료
> 1
아이디를 입력해 주세요.(뒤로가기 X번)
> master
비밀번호를 입력해 주세요.
> master
정상적으로 로그인 되었습니다.
```

```
-----
[안녕하세요. Master님(관리자)]
-----
```

```
1.영화보기 2.극장보기 3.회원정보 4.로그아웃
> 1
```

```
-----
현재 등록된 영화가 없습니다.
-----
```

```
1.개봉영화등록 2.뒤로가기(메인메뉴)
```

```
-----
1.개봉영화등록 2.뒤로가기(메인메뉴)
```

```
> 1
```

```
영화 제목을 입력해 주세요.
```

```
> 날아라 슈퍼보드
```

```
영화 줄거리를 입력해 주세요.
```

```
> 돼지
```

```
영화 등급을 선택해 주세요.
```

```
[1.전체관람가 2.12세관람가 3.15세관람가 4.18세관람가]
```

```
> 3
```

```
새로운 영화를 등록하시겠습니까? Y/N
```

```
> y
```

```
정상적으로 영화가 등록되었습니다^^
```

```
-----
1번. 날아라 슈퍼보드(15세관람가)
-----
```

```
1.개봉영화등록 2.영화정보보기 3.뒤로가기(메인메뉴)
```

```
>
```

관리자로 로그인 후 영화보기 > 개봉영화등록 후 정보를 입력하면 등록된다.  
이제 등록한 영화를 상영할 극장을 등록해보자.

## | 구현

## [관리자]

## 2. 극장 등록

1. 영화보기 2. 극장보기 3. 회원정보 4. 로그아웃

&gt; 2

현재 등록된 극장이 없습니다.

1. 극장등록하기 2. 뒤로가기(메인메뉴)

&gt; 1

극장 이름을 입력해 주세요.

&gt; 대한극장

극장 위치를 입력해 주세요.

&gt; 홍대입구

극장 전화번호를 입력해 주세요.(예:123-456-7899)

&gt; 123-1111-2222

극장 정보를 등록하시겠습니까? Y/N

&gt; y

정상적으로 등록되었습니다.

1. 극장등록하기 2. 극장 및 상영정보보기 3. 뒤로가기(메인메뉴)

&gt; 2

상세보기할 극장의 번호를 입력해 주세요.(뒤로가기 0번)

&gt; 1

&lt;극장 상세 정보&gt;

극장이름 : 대한극장

극장위치 : 홍대입구

극장번호 : 123-1111-2222

1. 상영정보보기 2. 극장정보수정 3. 극장정보삭제 4. 뒤로가기(극장목록)

&lt;극장 상세 정보&gt;

극장이름 : 대한극장

극장위치 : 홍대입구

극장번호 : 123-1111-2222

1. 상영정보보기 2. 극장정보수정 3. 극장정보삭제 4. 뒤로가기(극장목록)

&gt; 1

해당 극장에는 현재 등록된 상영정보가 없습니다.

영화 등록과 동일하게 극장 등록을 할 수 있고 극장 등록 후 정보보기를 들어가면 해당 극장의 상세 정보와 현재 상영 정보를 볼 수 있다.

지금은 등록된 상영 정보가 없기에 이제 상영 영화를 등록해보자.

## | 구현

## [관리자]

## 3. 극장에 상영 영화 등록

```

-----
해당 극장에는 현재 등록된 상영정보가 없습니다.
-----
1. 상영정보등록 2. 뒤로가기(극장상세정보)
> 1
-----
<현재 개봉한 영화 LIST>
-----
1번. 날아라 슈퍼보드(15세관람가)
-----
상영할 영화의 번호를 선택해 주세요.
> 1
시작시간을 입력해 주세요.(예:08:11)
> 09:23
상영시간을 입력해 주세요.(단위:분)
> 120
해당 상영정보를 등록하시겠습니까? Y/N
> y
정상적으로 상영정보가 등록되었습니다.

```

```

-----
<현재 극장에서 상영중인 영화 LIST>
-----
1. 날아라 슈퍼보드
-----
1. 상영정보등록 2. 상영영화 상세보기 3. 뒤로가기(극장상세정보)
> 2
상세보기하려는 상영영화 번호를 입력해 주세요.
> 1
-----
<상영 상세 정보>
-----
상영극장 : 대한극장
상영영화 : 날아라 슈퍼보드
영화시작시간 : 09시 23분
상영시간 : 120분
-----

```

상영정보등록으로 들어가면 아까 등록한 현재 개봉한 영화 목록을 확인할 수 있고,

상영할 영화를 선택 후 상영 정보를 등록하면 해당 극장에 상영 영화가 등록된다.

이제 해당 극장에서 상영중인 영화 리스트에 방금 등록한 영화가 보이고 상영 상세 정보도 확인할 수 있다.

## | 구현

## [영화 평론가]

## 1. 개봉 영화별 정보 확인

1번. 날아라 슈퍼보드(15세관람가)  
 2번. 아기공룡 둘리(전체관람가)  
 3번. 못말리는 황구(12세관람가)  
 4번. 도라이몽(18세관람가)

상세보기할 영화번호를 입력해 주세요. (뒤로가기 0번)  
 > 4

<영화 상세 정보>

영화제목 : 도라이몽  
 영화등급 : 18세관람가  
 영화줄거리 : 주머니에 모든 것이 있는 캐릭터  
 영화총평점 : 0.0점/5점(0명)

<현재 해당 영화를 상영하고 있는 극장 LIST>

1. 롯데 시네마

<현재 해당 영화를 상영하고 있는 극장 LIST>

1. 롯데 시네마

1. 감상평보기 2. 뒤로가기(영화목록)

> 1

등록된 감상평이 없습니다.

1. 새 감상평 등록 2. 뒤로가기(영화상세정보)

현재 개봉한 영화 목록을 볼 수 있으며 영화의 상세 정보와 평점, 현재 상영하고 있는 극장 목록을 확인할 수 있다.

이제 영화 평론가로서 평점과 감상평을 등록하러 가보자.



## | 구현

## [영화 평론가]

## 2. 평점 및 감상평 등록

등록된 감상평이 없습니다.

1. 새 감상평 등록 2. 뒤로가기(영화상세정보)

> 1

평점을 입력해 주세요. (0~5점)

> 3

감상평을 입력해 주세요.

> 괜찮습니다.

감상평을 등록하시겠습니까? Y/N

> y

정상적으로 감상평이 등록되었습니다^^

<영화평론가 감상평> 평점:3.0점/5점

1. [작성자:G1]- "괜찮습니다." (3점) (작성일:22.06.16 11:36:26)

<일반관람객 감상평> 평점:0.0점/5점

<영화 상세 정보>

영화제목 : 못말리는 평구

영화등급 : 12세관람가

영화줄거리 : 찡구는 개구쟁이

영화총평점 : 2.7점/5점(3명)

<현재 해당 영화를 상영하고 있는 극장 LIST>

1. 롯데 시네마

평점과 감상평을 입력하면 영화 평론가 감상평 목록에 추가되어 바로 보여진다.

평점은 일반 관람객 평점과 영화 평론가 평점이 구분되도록 하였다.

\*영화의 상세정보에서 보여지는 평점은 일반 관람객과 영화 평론가 평점 합쳐져 내어진 평점이다.

# | 구현

## [영화 평론가]

### 3. 극장별 상영 영화 확인

1. 영화보기 2. 극장보기 3. 회원정보 4. 로그아웃

> 2

1. 대한극장 (홍대입구)

2. ggV (서울역)

3. 롯데 시네마 (강남)

극장 및 상영정보 원하시는 극장번호를 입력해 주세요. (뒤로가기 0번)

> 2

<극장 상세 정보>

극장이름 : ggV

극장위치 : 서울역

극장번호 : 02-222-3322

1. 상영정보보기 2. 뒤로가기 (극장목록)

> 1

<현재 극장에서 상영중인 영화 LIST>

4. 아기공룡 둘리

상세보기하려는 상영영화 번호를 입력해 주세요. (뒤로가기 0번)

> 4

<상영 상세 정보>

상영극장 : ggV

상영영화 : 아기공룡 둘리

영화시작시간 : 09시 11분

상영시간 : 50분

극장 목록을 확인할 수 있고 극장별 정보를 볼 수 있다.

현재 상영되고 있는 영화목록도 보여지며 영화를 선택하면 상영 정보도 확인할 수 있다.

## | 구현

## [일반 관람객]

## 1. 개봉 영화 정보 확인

## &lt;영화 상세 정보&gt;

영화제목 : 못말리는 짱구  
 영화등급 : 12세관람가  
 영화줄거리 : 짱구는 개구쟁이  
 영화총평점 : 2.5점/5점 (2명)

## &lt;현재 해당 영화를 상영하고 있는 극장 LIST&gt;

1. 롯데 시네마

<영화평론가 감상평> 평점:2.5점/5점

2. [작성자:G1]- "짱구가 너무 나빠요" (2점) (작성일:22.06.16 11:38:06)

5. [작성자:G2]- "짱구 머리 해보고 싶어요" (3점) (작성일:22.06.16 11:41:25)

<일반관람객 감상평> 평점:0.0점/5점

1. 새 감상평 등록 2. 감상평 수정 3. 감상평 삭제 4. 뒤로가기 (영화상세정보)

> 1

평점을 입력해 주세요. (0~5점)

> 3

감상평을 입력해 주세요.

> 내 유치원 시절이 생각나네요

감상평을 등록하시겠습니까? Y/N

> y

정상적으로 감상평이 등록되었습니다^^

<영화평론가 감상평> 평점:2.5점/5점

2. [작성자:G1]- "짱구가 너무 나빠요" (2점) (작성일:22.06.16 11:38:06)

5. [작성자:G2]- "짱구 머리 해보고 싶어요" (3점) (작성일:22.06.16 11:41:25)

<일반관람객 감상평> 평점:3.0점/5점

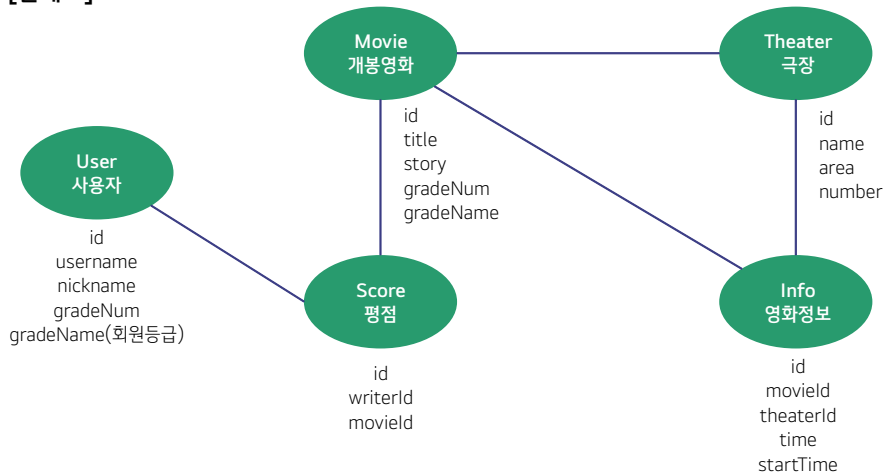
6. [작성자:hong]- "내 유치원 시절이 생각나네요" (3점) (작성일:22.06.16 11:43:42)

영화 평론가와 동일하게 영화 상세 정보를 들어가서 감상평을 적을 수 있고  
 감상평 적을 시 일반 관람객 감상평으로 보여진다.

극장 정보에서 현재 상영 영화와 정보를 볼 수 있음은 영화 평론가와 동일하므로 생략하겠다.

## | 개발

## [관계도]



## | 개발

### [util] - ScannerUtil 클래스 생성

- 유효성 검사를 해줄 기본적인 메소드들을 모아둔 클래스

예) 5. 사용자가 입력한 값이 숫자로만 이루어져있으면 true, 그 외에는 false가 반환되는 메소드 - 정규표현식 사용

```
private static boolean validateNumber(String s) {  
    String regEx = new String("^\\d+$");  
    if(s.matches(regEx)) {  
        return true;  
    }  
    regEx = new String("^-\\d+$");  
    if(s.matches(regEx)) {  
        return true;  
    }  
    return false;  
}
```

## | 개발

### [DTO]

- 기본 생성자, 매개변수 있는 생성자 생성
- setter, getter 사용
- equals() 메소드 사용하여 값들 비교

\*equals() 메소드 사용 이유 : 동일한 데이터(객체)인지 확인하기 위하여 -> id로 구분하기로 설정

```
public boolean equals(Object o) {  
    if(o instanceof TheaterDTO) {  
        TheaterDTO t = (TheaterDTO)o;  
        return id == t.id;  
    }  
    return false;  
}
```

# | 개발

## [Controller]

- UserController에서 기본 생성자를 통해 기본적으로 영화 평론가와 매니저 계정 생성되도록 함

```
public UserController() {  
    list = new ArrayList<>();  
    map = new HashMap<String, String>();  
    nextId = 1;  
  
    // 영화평론가 3명  
    for(int i=1 ; i<=3 ; i++) {  
        UserDTO u = new UserDTO();  
        u.setUsername("g"+i);  
        u.setPassword("g"+i);  
        map.put("g"+i, "g"+i);  
        u.setNickname("G"+i);  
        u.setGradeNum(2);  
  
        insert(u);  
    }  
  
    // 관리자 1명  
    UserDTO u = new UserDTO();  
    u.setUsername("master");  
    u.setPassword("master");  
    u.setNickname("Master");  
    map.put("master", "master");  
    u.setGradeNum(3);  
  
    insert(u);  
}
```

## | 개발

### [Controller]

- ArrayList를 사용하여 데이터가 동적 배열로 저장되게 함



# | 개발

## [View]

- 콘솔창에 보여지는 부분을 담당

```
// 영화 목록 매서드
public void showMenu() {
    while(true) {
        ArrayList<MovieDTO> list = movieController.selectAll();

        if(list.isEmpty()) {
            System.out.println("-----");
            System.out.println("현재 등록된 영화가 없습니다.");
            System.out.println("-----");

            if(login.getGradeNum() == 3) {
                String message = new String("1. 개봉영화등록 2. 뒤로가기(메인메뉴)");
                int userChoice = ScannerUtil.nextInt(scanner, message, 1, 2);

                if(userChoice == 1) {
                    movieRegister();
                } else if(userChoice == 2) {
                    break;
                }
            }

        } else {

            String message = new String("뒤로가기하려면 0번을 눌러주세요.");
            int userChoice = ScannerUtil.nextInt(scanner, message, 0, 0);
            break;

        }
    }
} else {
    System.out.println("-----");
    for(MovieDTO m : list) {
```

## | 개발

### [View]

- setter로 서로의 view 클래스들을 공유함

```
public void setScoreViewer(ScoreViewer scoreViewer) {  
    this.scoreViewer = scoreViewer;  
}
```

```
public void setUserViewer(UserViewer userViewer) {  
    this.userViewer = userViewer;  
}
```

```
public void setInfoViewer(InfoViewer infoViewer) {  
    this.infoViewer = infoViewer;  
}
```

## | 회고

지금 보면 정말 간단하고 어떻게 보면 하찮을(?) 수 있는 결과물이다.

하지만 당시의 나는 이 과제에 열심히 임했고 정말 재밌게 했고 더 많은 기능을 넣어보고 싶었지만 시간 제한상 넣지 못한 부분이 아쉬움으로 남았다.

이 ppt에 하나하나 기록하면서 내가 setter, getter, MVC, 코드 구성 등에 대해 어떠한 생각으로 했고 처음에 어떻게 접했는지가 기억이 난다.

생성자에 대해 배웠을 때 되게 모호한 느낌을 받았는데 Spring, Spring Boot에서 의존성에 관련하여 쓰이는 것을 나중에 배우게 되며 좀 더 뚜렷해졌던 그런 과정들이 있었다.

앞으로의 나의 개발자 인생도 초심을 잃지 않고 기억하며 성장해 나가고 싶다.

The end