

# СОФТВЕРСКИ КВАЛИТЕТ И ТЕСТИРАЊЕ

## ДОМАШНА ЗАДАЧА 6

Фисник Лимани, 151027

### 1. Класата **Movie**

- **String id**
- **List<User> users**
- **MovieRatingService movieRatingService**
- **getAverageRating()**

```
public double getAverageRating() {  
    double total = 0.0;  
  
    for(User user: users) {  
        total += movieRatingService.getRateFromUser(user.getUsername());  
    }  
  
    return users.size() > 0 ? total / users.size() : total;  
}
```
- **isThisMovieRatedBy(String username)**

```
public boolean isThisMovieRatedBy(String username) {  
    if(movieRatingService.getRateFromUser(username) == null){  
        return false;  
    }  
    return true;  
}
```

### 2. Класата **User**

- **String username;**
- **getNumberOfRatedMovies(List<Movie> movies)**

```
public int getNumberOfRatedMovies(List<Movie> movies) {  
    int total = 0;  
    for(Movie movie: movies) {  
        if(movie.isThisMovieRatedBy(username)){  
            total += 1;  
        }  
    }  
    return total;  
}
```

### 3. Интерфејс **MovieRatingService**

```
public interface MovieRatingService {  
    Integer getRateFromUser(String username);  
}
```

### 4. Класата **MockitoTesting**

- **setUpForTestingAverageRate** – иницијализираме вредности за соодветните варијабли за да го тестираме функцијата `averageRate`

```
public void setUpForTestingAverageRate() {  
    this.movie = new Movie( id: "movie1");  
  
    movieRatingService = mock(MovieRatingService.class);  
    this.movie.setMovieRatingService(movieRatingService);  
}
```

- **testAverageRate1()**
  - креираме 5 корисници кој го оценувале филмот
  - мокираме оценка 4 од првиот корисник, оценка 3 од вториот, оценка 4 од третиот, оценка 4 од четвртиот и оценка 5 од петтиот
  - треба да врати просечна оценка 4
- **testAverageRate2()**
  - креираме 5 корисници кој го оценувале филмот
  - мокираме оценка 2 од првиот корисник, оценка 4 од вториот, оценка 3 од третиот, оценка 3 од четвртиот, оценка 1 од петтиот
  - треба да врати  $(2+4+3+3+1)/5.0$
- **testAverageRate3()**
  - нема корисници кој го оценувале филмот
  - треба да врати 0

- **setUpForTestingTotalNumberOfMoviesRatedByUser()**
  - Креираме 5 Movie објекти
  - Креираме 5 MovieRatingService mock инстанци и ги додадеме по една за секој филм посебно
  - Креираме 3 User објекти
  - Оценките ги додадеме вака:
    - За првиот филм дале оценка првиот и вториот user
    - За вториот филм дале оценка првиот и вториот user
    - За третиот филм дал оценка вториот user
    - За четвртиот филм дале оценка првиот и вториот user
    - За петтиот филм дал оценка вториот user
- **testTotalNumberOfMoviesRatedByFirstUser()**
  - Тестираме првиот user на колку филмови им дал оценка
  - Треба да врати 3
- **testTotalNumberOfMoviesRatedBySecondUser()**
  - Тестираме вториот user колку филмови ги оценил
  - Треба да врати 5
- **testTotalNumberOfMoviesRatedByThirdUser()**
  - Тестираме третиот user колку филмови ги оценил
  - Треба да врати 0

## 5. OUTPUT ОД ИЗВРШУВАЊЕТО НА ТЕСТОВИТЕ

`testAverageRate:`

`Test1: Passed successfully`

`Test2: Passed successfully`

`Test3: Passed successfully`

`testTotalNumberOfMoviesRatedByUser`

`Test1: Passed successfully`

`Test2: Passed successfully`

`Test3: Passed successfully`