

객체지향 프로그래밍 및 실습

-1 차 프로그래밍 과제-

이름 : 김태석

학번 : 201720183

학과 : 미디어학과

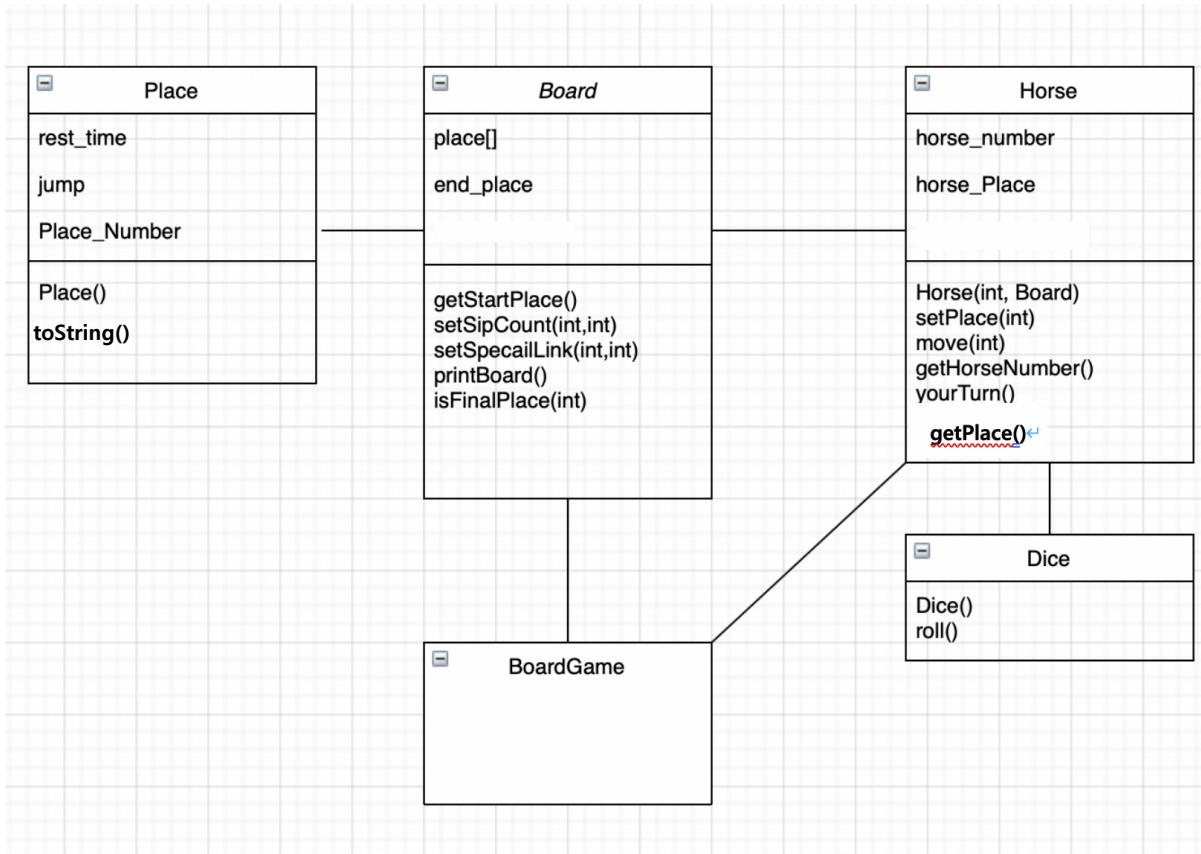
학년 : 2 학년

소개

- 테스트를 실행하는 클래스 외에 각 보드판, 말 등 게임을 구현하기 위한 클래스는 총 네개를 만들었다. Board, Horse, Dice, Place 클래스를 만들어 게임을 진행하기 원활하게 하였다.

구현한 부분	구현하지 못한부분(아쉬웠던 부분)
<ul style="list-style-type: none">- 칸의 크기는 10~20 이어야함.- 말의 개수는 2~4 여야함.- 순위가 다 정해졌을 때 게임을 종료하도록 함.- 시작점, 도착점에는 특별칸이 설치 될 수 없음.- 특별칸 두 곳이 한칸에 동시에 존재 x- Board 를 프린트하는 함수를 만들 때 place.toString 메소드를 이용하여 칸의 정보 모두 출력.	<ul style="list-style-type: none">- 쉬어가는칸, 스페셜링크 칸 중 어떤걸 먼저 실행할지를 정하고 그에 따른 예외성을 따져주는 작업은 구현하지 않았음. 쉬어가는 칸을 먼저 설정하고 스페셜링크 칸을 설정하는 순서만을 구현하였으며, 그에 따른 예외성을 따져줌.- 실제 게임처럼 플레이어들의 턴을 순서대로가 아닌 특정 결과 우승자 (이를테면 가위바위보 혹은 주사위 숫자가 가장 큰 순)순으로 진행하고 싶었지만 구현하지 못하였다.

분석/설계



-class diagram-

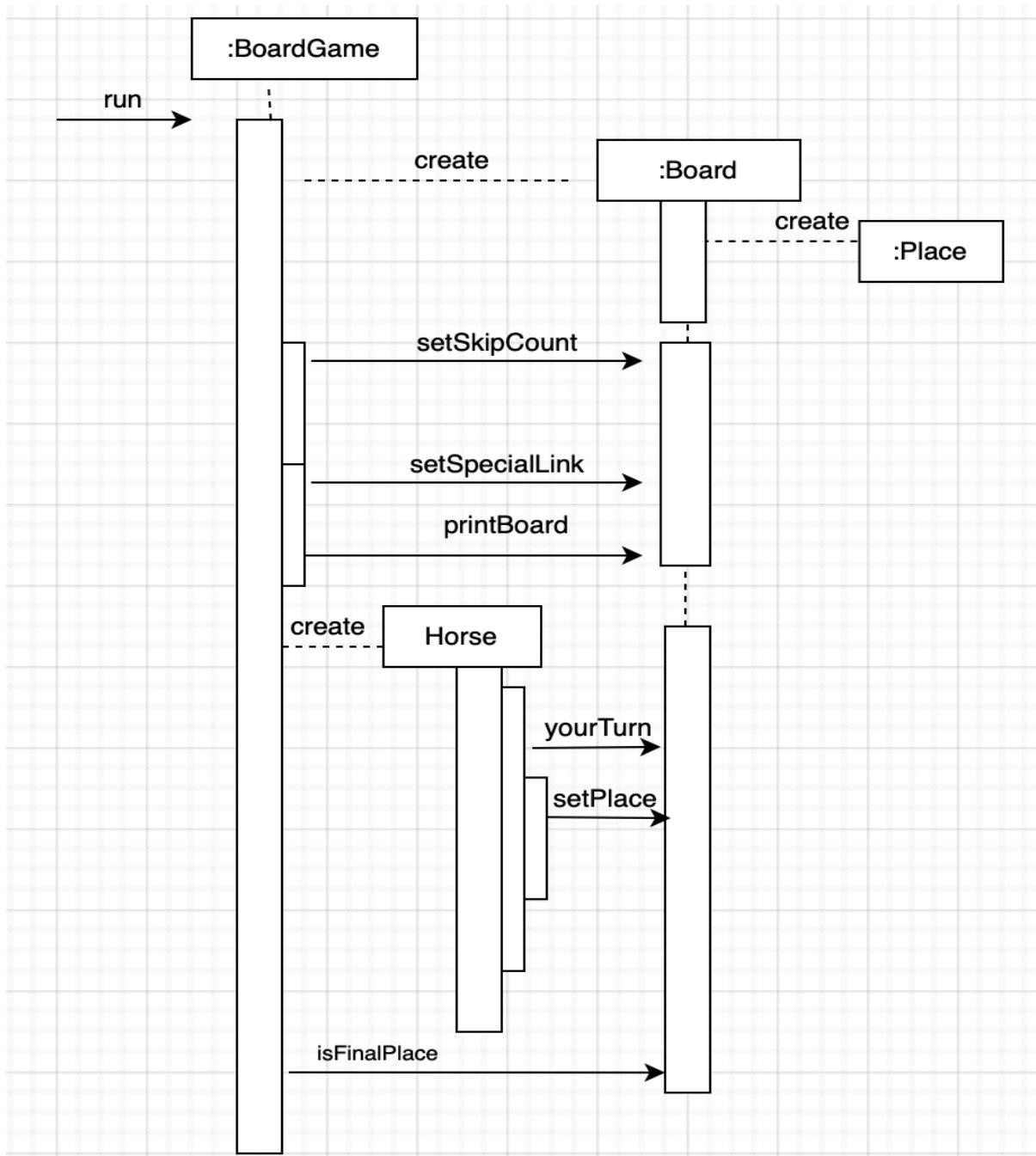
- **Board** : 보드판을 구현하는 클래스이다. getStartPlace, setSkipCount, setSpecialLink, isFinalPlace, printBoard 의 총 5 개의 메소드를 포함하고 있으며, 생성자에는 place 객체를 받아와서 보드판에서의 위치들을 표현할 수 있게 해주었다.
 - getStartPlace : 시작점을 반환하는 메소드이다(시작점은 항상 0 이므로 0 을 반환한다.)
 - setSkipCount : 쉬어가는 지점을 설정해주는 메소드이다. 매개변수로 쉬는칸의 넘버와 쉬어가는 횟수를 받아와서 설정해준다.
 - setSpecialLink : 특별경로로 이동하는 칸이다. 특별경로는 앞으로, 뒤로 둘 다 가능하며, 매개변수로 특별경로로 이동하는칸, 도착하는 칸을 받아온다.
 - printBoard : 보드판을 출력해주는 메소드이다. 일반칸은 _ 으로 표현해주었고, 쉬어가는칸은 * 스페셜링크칸은 **로 출력되게 만들어주었다.
 - isFinalPlace : 매개변수로 받아오는값이 도착점인지 아닌지를 판단하는 메소드이다. 이를통해 순위를 매겨줄 수 있다.
- **Horse** : 말을 구현하는 클래스이다. setPlace, move, getPlace, getHorseNumber, yourTurn 의 총 5 개의 메소드를 포함하고 있으며, 생성자에는 말의번호, 랭크를 포함해주었다.
 - setPlace : 말을 보드판의 칸 위에 놓이게 설정해준다.
 - move : 주사위를 굴려 굴린 수만큼 말이 이동한다.
 - getPlace : 현재 말의 위치를 반환해준다.

--- getHorseNumber : 말의 번호를 반환해준다.

--- yourTurn : 각 말들이 자신의 턴을 실시할 수 있게 해준다. 자신의 턴을 실시할 때 말의 위치가 어디인지에 따라 쉬어가는 횟수에 맞춰 움직일 수도, 혹은 주사위를 굴려 이동할 수도 있다.

- Dice : 주사위를 굴릴 수 있는 메소드를 포함하는 클래스이다. 1~6 까지 수를 랜덤으로 뽑아주는 간단한 방법을 통해 구현하였다.
- Place : 보드판에서의 위치를 표현하는 클래스로 다른 메소드 없이 생성자만 존재한다. 그 안에는 place number, jump, rest time 이 포함되어있는데 일반칸인 경우 place number 는 기본적으로 -1, 쉬어가는 칸의 경우 0, 점프되는 칸의 경우 1 로 설정해주었다. Jump 는 보드판에서 칸이 스페셜링크칸으로 설정된 경우, 매개변수로 받아오는 이동되는 칸의 값을 넣어주기 위해 만들어 놓았고, rest time 도 마찬가지로 보드판에서 칸이 skipcount 칸으로 설정된 경우, 매개변수로 받아오는 쉬어가는 횟수를 넣어주기 위해 만들어 놓았다.

---toString : 칸의 정보를 모두 출력해주는 메소드이다. 인트값의 칸 정보를 한 문장으로 출력해준다.



-sequence diagram-

보드판 객체를 생성한 후 보드판에서의 특별 칸을 설정 해 준다. 보드의 생성자에서는 플레이스를 끌어다 사용하기 때문에 플레이스 객체또한 같이 생성된다. 특별칸 생성이 끝나면 보드판을 출력해주고 그 후 말 객체를 생성 해 준다(말의 객체는 2~4개가 생성되어야 하지만 위의 시퀀스 다이어그램에선 일단 하나로 표현해 주었다.)

말 객체를 생성한 후 말이 보드판 위에서 자신의 턴을 실시 할 수 있게 해준다. 주사위를 여러번 굴려야 하므로 `yourTurn`메소드를 여러번 반복시켜 주어야한다. `yourTurn` 메소드를 실시하고 턴이 끝날 때마다 `setPlace`메소드를 실시해준다. 그 후 `yourTurn`메소드에서 보드판에 도착을 해서 더이상 턴의 실행이 되지 않으면 `isFinalPlace`메소드를 실시하여 말이 보드판 끝에 위치해 있는지 확인 후 게임을 종료해준다.

Test1

```
1차 실행화면  
<terminated> test1 (1) [Java App]  
Dice value : 5  
5  
Dice value : 2  
7
```

2차 실행화면

```
<terminated> test1 (1) [Java Application]  
Dice value : 4  
4  
Dice value : 4  
8
```

3차 실행화면

```
<terminated> test1 (1) [Java Ap]  
Dice value : 4  
4  
Dice value : 3  
7
```

테스트1에서는 보드판을 생성 후 말을 하나 생성하여 주사위를 두번 굴려주어서 각 굴려줄 때마다 주사위값과 함께 말의 위치를 출력해준다. 특별칸이 생성되지 않은 상태이므로 특별칸은 고려해 주지 않는다.

Test2

1차 실행화면

```
<terminated> test2 (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0
*   *   *   *
place0-place_number : -1 rest_time : 0 Jump Jump : 0
place1-place_number : -1 rest_time : 0 Jump Jump : 0
place2-place_number : -1 rest_time : 0 Jump Jump : 0
place3-place_number : -1 rest_time : 0 Jump Jump : 0
place4-place_number : 0 rest_time : 1 Jump Jump : 0
place5-place_number : 1 rest_time : 0 Jump Jump : 10
place6-place_number : -1 rest_time : 0 Jump Jump : 0
place7-place_number : 1 rest_time : 0 Jump Jump : 0
place8-place_number : 0 rest_time : 2 Jump Jump : 0
place9-place_number : 1 rest_time : 0 Jump Jump : 15
place10-place_number : 1 rest_time : 0 Jump Jump : 0
place11-place_number : -1 rest_time : 0 Jump Jump : 0
place12-place_number : 1 rest_time : 0 Jump Jump : 7
place13-place_number : -1 rest_time : 0 Jump Jump : 0
place14-place_number : -1 rest_time : 0 Jump Jump : 0
place15-place_number : 1 rest_time : 0 Jump Jump : 0
place16-place_number : -1 rest_time : 0 Jump Jump : 0
place17-place_number : -1 rest_time : 0 Jump Jump : 0
place18-place_number : -1 rest_time : 0 Jump Jump : 0
place19-place_number : -1 rest_time : 0 Jump Jump : 0
Dice value : 1
1
Dice value : 1
2
```

2차 실행화면

```
<terminated> test2 (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1
*   *   *   *
place0-place_number : -1 rest_time : 0 Jump Jump : 0
place1-place_number : -1 rest_time : 0 Jump Jump : 0
place2-place_number : -1 rest_time : 0 Jump Jump : 0
place3-place_number : -1 rest_time : 0 Jump Jump : 0
place4-place_number : 0 rest_time : 1 Jump Jump : 0
place5-place_number : 1 rest_time : 0 Jump Jump : 10
place6-place_number : -1 rest_time : 0 Jump Jump : 0
place7-place_number : 1 rest_time : 0 Jump Jump : 0
place8-place_number : 0 rest_time : 2 Jump Jump : 0
place9-place_number : 1 rest_time : 0 Jump Jump : 15
place10-place_number : 1 rest_time : 0 Jump Jump : 0
place11-place_number : -1 rest_time : 0 Jump Jump : 0
place12-place_number : 1 rest_time : 0 Jump Jump : 7
place13-place_number : -1 rest_time : 0 Jump Jump : 0
place14-place_number : -1 rest_time : 0 Jump Jump : 0
place15-place_number : 1 rest_time : 0 Jump Jump : 0
place16-place_number : -1 rest_time : 0 Jump Jump : 0
place17-place_number : -1 rest_time : 0 Jump Jump : 0
place18-place_number : -1 rest_time : 0 Jump Jump : 0
place19-place_number : -1 rest_time : 0 Jump Jump : 0
Dice value : 1
1
Dice value : 4
5
```

Test2

3차 실행화면

```
<terminated> test2 (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jd
*   ** - - *   ** - - *   **
place0-place_number : -1 rest_time : 0 Jump Jump : 0
place1-place_number : -1 rest_time : 0 Jump Jump : 0
place2-place_number : -1 rest_time : 0 Jump Jump : 0
place3-place_number : -1 rest_time : 0 Jump Jump : 0
place4-place_number : 0 rest_time : 1 Jump Jump : 0
place5-place_number : 1 rest_time : 0 Jump Jump : 10
place6-place_number : -1 rest_time : 0 Jump Jump : 0
place7-place_number : 1 rest_time : 0 Jump Jump : 0
place8-place_number : 0 rest_time : 2 Jump Jump : 0
place9-place_number : 1 rest_time : 0 Jump Jump : 15
place10-place_number : 1 rest_time : 0 Jump Jump : 0
place11-place_number : -1 rest_time : 0 Jump Jump : 0
place12-place_number : 1 rest_time : 0 Jump Jump : 7
place13-place_number : -1 rest_time : 0 Jump Jump : 0
place14-place_number : -1 rest_time : 0 Jump Jump : 0
place15-place_number : 1 rest_time : 0 Jump Jump : 0
place16-place_number : -1 rest_time : 0 Jump Jump : 0
place17-place_number : -1 rest_time : 0 Jump Jump : 0
place18-place_number : -1 rest_time : 0 Jump Jump : 0
place19-place_number : -1 rest_time : 0 Jump Jump : 0
Dice value : 4
4
Dice value : 5
9
```

테스트2에서는 특별칸을 설정하여서. 주사위를 두번 굴렸을 때 말의 위치를 각각 출력해주는 테스트이다. 하지만 여기서 특별칸을 제대로 넘어가고, 구현한 클래스들을 그대로 사용하기위해선 테스트2의 내용이 일부 수정되어야 한다.(2번째 실행화면을 보면 5번째 칸에 도착하였지만 점프를 하지 않은것을 볼 수 있다.)

본인은 yourTurn 메소드에 특별칸을 따지는 로직을 구성하였기에

```
/*int n= Dice.roll();
System.out.println("Dice value : "+n);
horse.move(n);
p=horse.getPlace();
System.out.println(p);
n=Dice.roll();
System.out.println("Dice value : "+n);
horse.move(n);
p=horse.getPlace();
System.out.println(p);*/
horse.yourTurn();
p=horse.getPlace();
System.out.println(p);
horse.yourTurn();
p=horse.getPlace();
System.out.println(p);
```

이와같이 테스트2의 코드를 일부 수정하여 정상적인 값이 출력 될 수 있도록 변경하였다.

아래의 실행화면을 보면 코드를 수정 한 후 특별칸의 기능을 정상 수행하는 모습을 확인가능하다.

Test2

```
<terminated> test2 (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Cc
* ** - * **
place0-place_number : -1 rest_time : 0 Jump Jump : 0
place1-place_number : -1 rest_time : 0 Jump Jump : 0
place2-place_number : -1 rest_time : 0 Jump Jump : 0
place3-place_number : -1 rest_time : 0 Jump Jump : 0
place4-place_number : 0 rest_time : 1 Jump Jump : 0
place5-place_number : 1 rest_time : 0 Jump Jump : 10
place6-place_number : -1 rest_time : 0 Jump Jump : 0
place7-place_number : 1 rest_time : 0 Jump Jump : 0
place8-place_number : 0 rest_time : 2 Jump Jump : 0
place9-place_number : 1 rest_time : 0 Jump Jump : 15
place10-place_number : 1 rest_time : 0 Jump Jump : 0
place11-place_number : -1 rest_time : 0 Jump Jump : 0
place12-place_number : 1 rest_time : 0 Jump Jump : 7
place13-place_number : -1 rest_time : 0 Jump Jump : 0
place14-place_number : -1 rest_time : 0 Jump Jump : 0
place15-place_number : 1 rest_time : 0 Jump Jump : 0
place16-place_number : -1 rest_time : 0 Jump Jump : 0
place17-place_number : -1 rest_time : 0 Jump Jump : 0
place18-place_number : -1 rest_time : 0 Jump Jump : 0
place19-place_number : -1 rest_time : 0 Jump Jump : 0
Horse1's Dice value : 4
4
Horse1 Rest-off!
4
```

Test3

1 차 실행화면

```
<terminated> test3 (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/C
* * * * *
place0-place_number : -1 rest_time : 0 Jump Jump : 0
place1-place_number : -1 rest_time : 0 Jump Jump : 0
place2-place_number : -1 rest_time : 0 Jump Jump : 0
place3-place_number : -1 rest_time : 0 Jump Jump : 0
place4-place_number : 0 rest_time : 1 Jump Jump : 0
place5-place_number : 1 rest_time : 0 Jump Jump : 10
place6-place_number : -1 rest_time : 0 Jump Jump : 0
place7-place_number : 1 rest_time : 0 Jump Jump : 0
place8-place_number : 0 rest_time : 2 Jump Jump : 0
place9-place_number : 1 rest_time : 0 Jump Jump : 15
place10-place_number : 1 rest_time : 0 Jump Jump : 0
place11-place_number : -1 rest_time : 0 Jump Jump : 0
place12-place_number : 1 rest_time : 0 Jump Jump : 7
place13-place_number : -1 rest_time : 0 Jump Jump : 0
place14-place_number : -1 rest_time : 0 Jump Jump : 0
place15-place_number : 1 rest_time : 0 Jump Jump : 0
place16-place_number : -1 rest_time : 0 Jump Jump : 0
place17-place_number : -1 rest_time : 0 Jump Jump : 0
place18-place_number : -1 rest_time : 0 Jump Jump : 0
place19-place_number : -1 rest_time : 0 Jump Jump : 0
Horse1's Dice value : 5
Horse1's place = 10
Horse2's Dice value : 4
Horse2's place = 4
Horse1's Dice value : 4
Horse1's place = 14
Horse2 Rest-off!
Horse2's place = 4
Horse1's Dice value : 2
Horse1's place = 16
Horse2's Dice value : 2
Horse2's place = 6
Horse1's Dice value : 2
Horse1's place = 18
Horse2's Dice value : 1
Horse2's place = 7
Horse1's Dice value : 3
Horse1's place = 19
Horse 1 wins!
```

2 차 실행화면

```
<terminated> test3 (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/C
* * * * *
place0-place_number : -1 rest_time : 0 Jump Jump : 0
place1-place_number : -1 rest_time : 0 Jump Jump : 0
place2-place_number : -1 rest_time : 0 Jump Jump : 0
place3-place_number : -1 rest_time : 0 Jump Jump : 0
place4-place_number : 0 rest_time : 1 Jump Jump : 0
place5-place_number : 1 rest_time : 0 Jump Jump : 10
place6-place_number : -1 rest_time : 0 Jump Jump : 0
place7-place_number : 1 rest_time : 0 Jump Jump : 0
place8-place_number : 0 rest_time : 2 Jump Jump : 0
place9-place_number : 1 rest_time : 0 Jump Jump : 15
place10-place_number : 1 rest_time : 0 Jump Jump : 0
place11-place_number : -1 rest_time : 0 Jump Jump : 0
place12-place_number : 1 rest_time : 0 Jump Jump : 7
place13-place_number : -1 rest_time : 0 Jump Jump : 0
place14-place_number : -1 rest_time : 0 Jump Jump : 0
place15-place_number : 1 rest_time : 0 Jump Jump : 0
place16-place_number : -1 rest_time : 0 Jump Jump : 0
place17-place_number : -1 rest_time : 0 Jump Jump : 0
place18-place_number : -1 rest_time : 0 Jump Jump : 0
place19-place_number : -1 rest_time : 0 Jump Jump : 0
Horse1's Dice value : 5
Horse1's place = 10
Horse2's Dice value : 4
Horse2's place = 4
Horse1's Dice value : 3
Horse1's place = 13
Horse2 Rest-off!
Horse2's place = 4
Horse1's Dice value : 4
Horse1's place = 17
Horse2's Dice value : 3
Horse2's place = 7
Horse1's Dice value : 1
Horse1's place = 18
Horse2's Dice value : 4
Horse2's place = 11
Horse1's Dice value : 1
Horse1's place = 19
Horse 1 wins!
```

Test3

3 차 실행화면

```
<terminated> test3 (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdl
*   **  *   **  *   **
place0-place_number : -1 rest_time : 0 Jump Jump : 0
place1-place_number : -1 rest_time : 0 Jump Jump : 0
place2-place_number : -1 rest_time : 0 Jump Jump : 0
place3-place_number : -1 rest_time : 0 Jump Jump : 0
place4-place_number : 0 rest_time : 1 Jump Jump : 0
place5-place_number : 1 rest_time : 0 Jump Jump : 10
place6-place_number : -1 rest_time : 0 Jump Jump : 0
place7-place_number : 1 rest_time : 0 Jump Jump : 0
place8-place_number : 0 rest_time : 2 Jump Jump : 0
place9-place_number : 1 rest_time : 0 Jump Jump : 15
place10-place_number : 1 rest_time : 0 Jump Jump : 0
place11-place_number : -1 rest_time : 0 Jump Jump : 0
place12-place_number : 1 rest_time : 0 Jump Jump : 7
place13-place_number : -1 rest_time : 0 Jump Jump : 0
place14-place_number : -1 rest_time : 0 Jump Jump : 0
place15-place_number : 1 rest_time : 0 Jump Jump : 0
place16-place_number : -1 rest_time : 0 Jump Jump : 0
place17-place_number : -1 rest_time : 0 Jump Jump : 0
place18-place_number : -1 rest_time : 0 Jump Jump : 0
place19-place_number : -1 rest_time : 0 Jump Jump : 0
Horse1's Dice value : 5
Horse1's place = 10
Horse2's Dice value : 4
Horse2's place = 4
Horse1's Dice value : 1
Horse1's place = 11
Horse2 Rest-off!
Horse2's place = 4
Horse1's Dice value : 3
Horse1's place = 14
Horse2's Dice value : 5
Horse2's place = 15
Horse1's Dice value : 1
Horse1's place = 15
Horse2's Dice value : 5
Horse2's place = 19
Horse 2 wins!
```

테스트 3 은 둘이서 게임을 진행 하는 상황이다. 칸의 크기, 특별칸들의 설정은 이미 다 주어진 상태에서 주사위를 굴려서 게임을 진행한다. 테스트 3 에서는 구현해놓은 yourTurn 메소드를 사용하였기에 따로 수정해 줄 사안은 없었다. 1,2,3 차 실행화면 모두 각 말들이 특별칸을 거친것을 확인 할 수 있고 그에따라 쉬어가거나, 특정칸으로 점프한 결과를 확인 할 수 있다

Test4

1 차 실행화면

```
<terminated> test4 (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/1
Setting new Board! How many place?
15
How many rest_time place are on the board?
2
How many jumping place are on the board?
3
Set board : rest_time place
Place :
4
Rest Time :
1
Place :
8
Rest Time :
2
Set board : jumping area
Place :
5
Jump To :
10
Place :
9
Jump To :
14
Place :
12
Jump To :
7
    * ** - - * ** - - *
place0-place_number : -1 rest_time : 0 Jump Jump : 0
place1-place_number : -1 rest_time : 0 Jump Jump : 0
place2-place_number : -1 rest_time : 0 Jump Jump : 0
place3-place_number : -1 rest_time : 0 Jump Jump : 0
place4-place_number : 0 rest_time : 1 Jump Jump : 0
place5-place_number : 1 rest_time : 0 Jump Jump : 10
place6-place_number : -1 rest_time : 0 Jump Jump : 0
place7-place_number : 1 rest_time : 0 Jump Jump : 0
place8-place_number : 0 rest_time : 2 Jump Jump : 0
place9-place_number : 1 rest_time : 0 Jump Jump : 14
place10-place_number : 1 rest_time : 0 Jump Jump : 0
place11-place_number : -1 rest_time : 0 Jump Jump : 0
place12-place_number : 1 rest_time : 0 Jump Jump : 7
place13-place_number : -1 rest_time : 0 Jump Jump : 0
place14-place_number : 1 rest_time : 0 Jump Jump : 0
How many players? :
4
How many players? :
4
Horse1's Dice value : 5
Horse1's place = 10
Horse2's Dice value : 4
Horse2's place = 4
Horse3's Dice value : 2
Horse3's place = 2
Horse4's Dice value : 1
Horse4's place = 1
Horse1's Dice value : 3
Horse1's place = 13
Horse2 Rest-off!
Horse2's place = 4
Horse3's Dice value : 1
Horse3's place = 3
Horse4's Dice value : 4
Horse4's place = 10
Horse1's Dice value : 5
Horse1's place = 14
Horse2's Dice value : 3
Horse2's place = 7
Horse3's Dice value : 5
Horse3's place = 8
Horse4's Dice value : 4
Horse4's place = 14
Horse2's Dice value : 1
Horse2's place = 8
Horse3 Rest-off!
Horse3's place = 8
Horse2 Rest-off!
Horse2's place = 8
Horse3's Dice value : 2
Horse3's place = 10
Horse2's Dice value : 5
Horse2's place = 13
Horse3's Dice value : 1
Horse3's place = 11
Horse2's Dice value : 3
Horse2's place = 14
Horse3's Dice value : 3
Horse3's place = 14
Ranking is :
Horse number1 ranking is : 1
Horse number2 ranking is : 3
Horse number3 ranking is : 4
Horse number4 ranking is : 2
Game Over!!
```

Test4

2 차 실행화면

```
<terminated> test4 (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14
Setting new Board! How many place?
10
How many rest_time place are on the board?
1
How many jumping place are on the board?
2
Set board : rest_time place
Place :
5
Rest Time :
1
Set board : jumping area
Place :
6
Jump To :
9
Place :
8
Jump To :
1
place0-place_number : -1 rest_time : 0 Jump Jump : 0
place1-place_number : 1 rest_time : 0 Jump Jump : 0
place2-place_number : -1 rest_time : 0 Jump Jump : 0
place3-place_number : -1 rest_time : 0 Jump Jump : 0
place4-place_number : -1 rest_time : 0 Jump Jump : 0
place5-place_number : 0 rest_time : 1 Jump Jump : 0
place6-place_number : 1 rest_time : 0 Jump Jump : 9
place7-place_number : -1 rest_time : 0 Jump Jump : 0
place8-place_number : 1 rest_time : 0 Jump Jump : 1
place9-place_number : 1 rest_time : 0 Jump Jump : 0
How many players? :
2
Horse1's Dice value : 2
Horse1's place = 2
Horse2's Dice value : 2
Horse2's place = 2
Horse1's Dice value : 3
Horse1's place = 5
Horse2's Dice value : 4
Horse2's place = 9
Ranking is :
Horse number1 ranking is : 2
Horse number2 ranking is : 1
Game Over!!
```

Test4

3 차 실행화면

```
<terminated> test4 (1) [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Content/ place0-place_number : -1 rest_time : 0 Jump Jump : 0
Setting new Board! How many place? 20
How many rest_time place are on the board? 2
How many jumping place are on the board? 2
Set board : rest_time place
Place :
Rest Time :
1
Place :
12
Rest Time :
3
Set board : jumping area
Place :
3
Jump To :
6
Place :
13
Jump To :
16
* **
place0-place_number : -1 rest_time : 0 Jump Jump : 0
place1-place_number : -1 rest_time : 0 Jump Jump : 0
place2-place_number : 0 rest_time : 1 Jump Jump : 0
place3-place_number : 1 rest_time : 0 Jump Jump : 6
place4-place_number : -1 rest_time : 0 Jump Jump : 0
place5-place_number : -1 rest_time : 0 Jump Jump : 0
place6-place_number : 1 rest_time : 0 Jump Jump : 0
place7-place_number : -1 rest_time : 0 Jump Jump : 0
place8-place_number : -1 rest_time : 0 Jump Jump : 0
place9-place_number : -1 rest_time : 0 Jump Jump : 0
place10-place_number : -1 rest_time : 0 Jump Jump : 0
place11-place_number : -1 rest_time : 0 Jump Jump : 0
place12-place_number : 0 rest_time : 3 Jump Jump : 0
place13-place_number : 1 rest_time : 0 Jump Jump : 16
place14-place_number : -1 rest_time : 0 Jump Jump : 0
place15-place_number : -1 rest_time : 0 Jump Jump : 0
place16-place_number : 1 rest_time : 0 Jump Jump : 0
place17-place_number : -1 rest_time : 0 Jump Jump : 0
place18-place_number : -1 rest_time : 0 Jump Jump : 0
place19-place_number : -1 rest_time : 0 Jump Jump : 0
How many players? 3
Horse1's Dice value : 3
Horse1's place = 6
Horse2's Dice value : 4
Horse2's place = 4
Horse3's Dice value : 5
Horse3's place = 5
Horse1's Dice value : 3
Horse1's place = 9
Horse2's Dice value : 3
Horse2's place = 7
Horse3's Dice value : 3
Horse3's place = 8
Horse1's Dice value : 2
Horse1's place = 11
Horse2's Dice value : 3
Horse2's place = 10
Horse3's Dice value : 1
Horse3's place = 9
Horse1's Dice value : 4
Horse1's place = 15
Horse2's Dice value : 1
Horse2's place = 11
Horse3's Dice value : 1
Horse3's place = 10
Horse1's Dice value : 2
Horse1's place = 17
Horse2's Dice value : 1
Horse2's place = 12
Horse3's Dice value : 4
Horse3's place = 14
Horse1's Dice value : 5
Horse1's place = 19
Horse2 Rest-off!
Horse2's place = 12
Horse3's Dice value : 3
Horse3's place = 17
Horse2 Rest-off!
Horse2's place = 12
Horse3's Dice value : 3
Horse3's place = 19
Ranking is :
Horse number1 ranking is : 1
Horse number2 ranking is : 3
Horse number3 ranking is : 2
Game Over!!
```

Test4

테스트4는 1차 프로그래밍 과제의 파이널 테스트를 실시하게 된다.

칸의 수를 입력 받고, 스페셜칸을 직접 설정해주며 플레이어수의 입력을 통하여 몇명이서 게임을 진행하는지 결정 할 수 있다.

다양한 결과를 내기 위하여 총 3번의 실행에서 보드판의 크기, 플레이어의 수, 특별칸의 설정을 모두 달리했으며 그에 따라 결과또한 다양해 진 것을 확인 할 수 있다.

테스트4를 진행하며 가장 중요시 여겼던것은 문제의 조건에 부합하는 값을 받지 못했을 때 다시 값을 입력받게 한 점이었다. 보드판의 크기(10~20), 플레이어의 수(2~4), 특별칸의 설정 조건 등 모든 예외들을 조건을 걸어 두어서 올바르지 못한 값을 입력했을 때 값을 다시 입력 받을 수 있게 로직을 구성하였다. 또한 테스트3에서는 단 둘이 게임을 하였기에 한쪽이 먼저 도착하면 게임이 끝나는 로직으로 순위를 매길 수 있었지만, 테스트4에서는 플레이어의 수가 최대 4명까지 가능하기에 이에따라서 먼저 도착하면 더이상 주사위를 굴리지 않고 도착하는 순서대로 랭크를 카운트 할 수 있는 조건을 걸어두었다.

프로그래밍 시 어려웠던 점

-특별칸을 생성하는 과정이 가장 처음 부딪힌 문제 중 하나였다. 점프칸 설정의 경우 그냥 그 칸의 넘버를 바꿔주어 말의 위치와 점프칸의 경우가 같을 경우 바로 이동해주는 설정을 해 주었지만, 쉬어가는 턴을 설정 해주는걸 어떻게 짜야 할지 구상이 되지 않아서 특별 칸 메소드 생성에만 하루를 쓸게 되었다. 결국 생각 해 낸 것이 특별칸에 도착할 경우 매개변수로 쉬어가는 횟수를 받아서 그 횟수만큼 yourTurn 메소드에서 조건문을 통해 쉬어갈 수 있게 해주었다.

-말이 같은 보드판을 공유하기 때문에 yourTurn 메소드를 실시하면 각 말들이 위치값을 공유하는 문제가 발생하였다. 메소드들의 설계가 잘못된 것 같아 다른 방법으로 코드 대부분을 수정했음에도 위치값을 공유하는 문제가 지속되었다. 해결 방안은, Horse class에서 this를 사용하여 horse class의 method에서 각 변수가 지칭하는 것을 그 생성된 객체 내부의 것으로 설정해주어 문제를 해결 할 수 있었다.

-3인 이상이 게임을 진행했을 때 순위를 매기는 것이 까다로웠다. 2인의 게임과 같은 경우, 한쪽이 먼저 보드판에 도착하면 승패가 결정될 뿐만 아니라, 과제 설명에도 2인의 게임에 대한 스켈레톤이 잡혀있어서 쉽게 해결 할 수 있었지만, 3인 이상의 경우 한사람만이 도착한다고 해서 순위가 1,2,3위 자동으로 매겨지는 것이 아니기 때문에, 1등 2등 3등을 순서대로 매길 수 있어야 했다. 이를 해결하기 위해서 생성된 horse에 랭크값을 기본으로 0으로 설정 해 주었고, 도착하는 순서대로 랭크 카운트를 하나씩 올려주었다. n명이서 진행하는 게임에서는 n-1등까지 등수가 매겨지면 자동으로 n번째는 꼴지가 되며 게임이 종료되지만 n등의 랭크값은 그대로 0으로 남아있어서 랭크값이 0으로 설정된 말은 n등이라고 설정해주는 조건문을 추가해주어서 각 말들이 0이 아닌 랭크값을 가질 수 있게 해주었다.

-예외성을 따지는 문제가 어려웠다. 기본적으로 프로그램이 돌아간다고 하여도 과제의 조건에 부합하는 값을 받아와야 하기 때문에 조건에 부적절한 모든 경우는 예외로 간주해야 했다. 플레이어 수 설정(2~4인), 보드판의 크기(10~20), 특별칸의 조건(점프칸과 쉬는칸이 같은 칸일 수 없으며, 시작점과 끝점에는 위치 할 수 없다, 입력받는 특별칸과 점프칸의 플레이스가 보드판 내에만 존재하여야 한다)등의 모든 예외를 생각하여 조건을 걸어줘야 했던 것이 까다로웠다. 다양한 예외들을 생각하면서 모든 예외를 다 설정 해 주었다고 생각했는데 다른 예외가 발생했을 때 계속해서 코드에 예외를 설정해주었다.

프로그램을 구상하며 배운점

- 가장 첫번째로 배운점은 jar파일 생성 후 터미널(혹은 cmd)을 통해 그를 실행시켜 본 점이다.
처음에는 명령어를 잘못입력하여 실행이 되지 않아 당황했지만, java -jar (테스트파일명).jar로 제대로 입력을 해 주었는데도 실행이 되지 않아서 곤란했다. 하지만 파일이 있는 경로를 제대로 설정해주고, 설치된 jdk 버전이 맞는곳에서 실행시키자 실행이 원활히 되었다.

파일 실행시 올바른 명령어, 올바른 경로 및 환경이 중요하다는 것을 배웠다.

- 주어진 조건에 맞는 코드 짜기 위해서 각 예외들이 모두 처리되어야 한다는 것을 배웠다.
모든 예외를 다 설정 해 주었다고 생각했지만, 하나를 처리하다보면 새로운 예외를 발견하고 또 새로운 것을 발견하는것을 반복하였다.

프로그램을 짠 후 몇번이고 검토하고 실행시키며 확인을 하며 고쳐야 할 것들을 발견하는것이 중요하다는 점을 배웠다.

-다양한 메소드들과 클래스들을 만들며 어느경우에 어떤 메소드를 생성할지, 생성된 메소드를 언제 이용할지 생각해보게 되었다. Boolean 타입의 메소드를 만들것이냐, void 형을 취할것이냐 등 각 메소드의 기능에 알맞는 것을 선택해보는 시간을 가졌고, **만들어진 메소드들을 단순히 호출하는것이 아니라 메소드들 어떻게, 언제 이용하고 활용할지 생각해볼 수 있는 기회가 되었다.**