Java Chess

202168052 강현

목차

- 클래스 설명
- 기물 이동 과정 설명
- 느낀 점

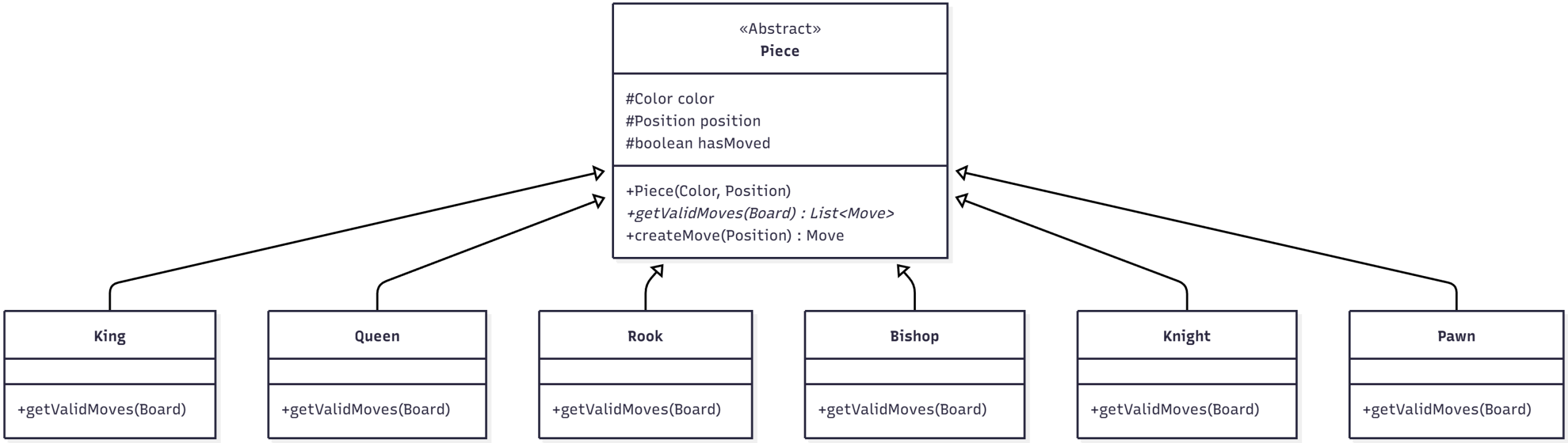
Position, Move 클래스

- Position 클래스 : 체스판 위의 좌표를 나타내는 가장 작은 단위의 객체입니다
- Equals() 를 오버라이딩을 통해 재정의하여 주소가 달라도 위치가 같으면 같은 위치로 재정의 했습니다
- 좌표가 체스판 위에 있는지 확인하는 메서드를 통해 스스로 검사하여 에러를 방지합니다

- Move 클래스: 출발지와 목적지를 묶은 기물의 이동 정보를 저장하는 클래스입니다 사용자가 입력한 경로가 유효한 이동 목록에 존재하는지 **비교 및 검증하는 기준**이 됩니다

Piece 클래스

- 추상 클래스 Piece와 6개의 기물 클래스(Knight,Rook,Pawn,Bishop,Queen,King)으로 구성되어 있으며, getValidMoves메서드를 통해 이동 가능한 방향을 반환받을 수 있습니다
- Knight : 점프하는 8방향 확인 Pawn: 일반적인 전진과 대각선 전진 확인
- Rook/Bishop/Queen : 특정한 방향으로 쭉 뻗어나가는 반복문 체크



Board, Player 클래스

- Board 클래스 : 8 X 8 배열의 체스판을 실제로 관리하는 클래스로, 실제 배열에서 기물 이동과 잡힌 기물 처리(executeMove()), 콘솔에 체스판 출력 (display()) , Game 클래스에서 이동 유효 시뮬레이션을 위해 체스판 복제(copy()) 를 수행합니다

- Player 클래스 : 콘솔 창을 통해 들어오는 입력을 처리하는 클래스로, getMoveFromUser,getMoveToUser 메서드를 통해 유효한 입력을 확인합니다

Game 클래스

- 모든 클래스를 이용하고 조합하여 게임을 실제로 작동하게 하는 클래스로 규칙 판정 및 특수 규칙 처리, 게임 진행 및 상태 관리를 하는 클래스입니다

- getStandardLegalMoves() 메서드를 통해 킹을 위험하게 만드는 기본적인 수를 시뮬레이션하고, getAllLegalMovesForPiece() 를 이용하여 앙파상, 캐슬링도 같이 시뮬레이션합니다. 이 외에handlePromotion() 메서드를 통해서 프로모션을 구현합니다

- **start()**: display() (출력) → handleTurn() (입력 및 검증) → switchPlayer() (턴 교체) 반복하고 updateGameState() 를 통해 매 턴 종료시 승패를 결정합니다

기물 이동 과정

- Main에서 Game을 시작함.
- Game은 Player에게 입력을 시킵니다.
- Player가 받아온 좌표로 Game이 Piece에게 이동이 가능한 위치를 물어봅니다.
- Game은 그 이동이 안전한지(왕이 안 죽는지) Board를 복사해서 시뮬레이션합니다.
- 검증되면 Board에게 기물을 옮기라고(executeMove) 명령합니다.

실제 구동 사진

```
    0   1   2   3   4   5   6   7
  ---------------------------------
0 | r | . | b | q | k | b | n | r |
  ---------------------------------
1 | . | p | p | p | p | . | p | p |
  ---------------------------------
2 | . | . | R | . | . | . | . | . |
  ---------------------------------
3 | p | . | . | . | . | p | . | . |
  ---------------------------------
4 | P | . | . | . | . | . | . | . |
  ---------------------------------
5 | . | . | . | . | . | . | . | . |
  ---------------------------------
6 | . | P | P | P | P | P | P | P |
  ---------------------------------
7 | . | N | B | Q | K | B | N | R |
  ---------------------------------
흑(B)팀의 기물 이동
이동할 기물의 위치 (y x):
```

```
    0   1   2   3   4   5   6   7
  ---------------------------------
0 | r | n | b | q | k | b | n | r |
  ---------------------------------
1 | . | p | p | p | p | p | p | . |
  ---------------------------------
2 | . | . | . | . | . | . | . | . |
  ---------------------------------
3 | p | . | . | . | . | . | . | . |
  ---------------------------------
4 | P | . | . | . | . | . | . | p |
  ---------------------------------
5 | . | R | . | . | . | . | . | . |
  ---------------------------------
6 | . | P | P | P | P | P | P | P |
  ---------------------------------
7 | . | N | B | Q | K | B | N | R |
  ---------------------------------
백(W)팀의 기물 이동
이동할 기물의 위치 (y x):5 1
이동 가능한 위치: Position[y=4, x=1]
Position[y=3, x=1]
Position[y=2, x=1]
Position[y=1, x=1]
Position[y=5, x=0]
Position[y=5, x=2]
Position[y=5, x=3]
Position[y=5, x=4]
Position[y=5, x=5]
Position[y=5, x=6]
Position[y=5, x=7]

목표 위치 (y x):1 1
```

느낀 점

- 복잡한 체스를 구현하면서 실제 프로그래밍을 할때 신경써야할 부분이 많다는것을 느꼈고, 앞으로 부족한 실력을 연습을 통해 보완해야 할것같다는 생각이 들었습니다

- 자바 및 프로그램 구현 실력이 늘어난 것 같아 좋은것 같습니다