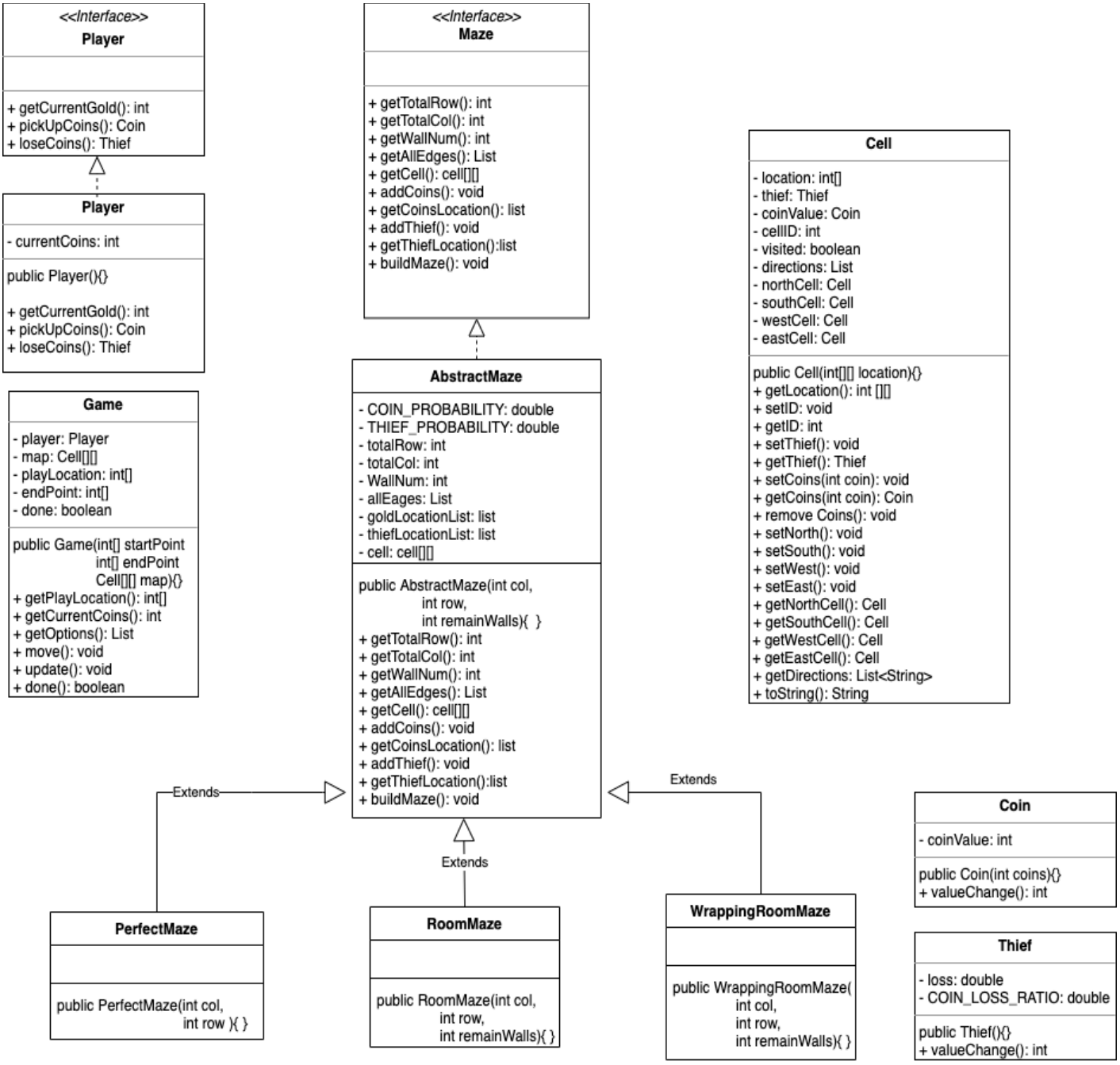
HW3:

UML:



Test Case:

1. Test Maze
   //Check maze Construction: toString()

   //Check maze information:
   assertEquals(1, maze.getNumOfRows())
   assertEquals(1, maze.getNumOfCols())
   assertEquals(0, maze.getRemainWalls())
   assertEquals(false, maze.isWrapped())

```
//test negative numOfRows/numOfCol: throw IllegalArgumentException
Perfect maze = new PerfectMaze(1, -2) ;

//test invalid remain walls
PerfectMaze maze = new PerfectMaze(3,2,3) //should be between 0 – 2

//test negative numOfRows/numOfCol
assertEquals(1, maze.getNumOfRows())
assertEquals(1, maze.getNumOfCols())

//Check gold information:
assertEquals(2, maze.getGoldLocation().get[0](2))

//Check thief information:
assertEquals(2, maze.getThiefLocation().get[0](2))
```

2. Test Player:

```
//test current gold
assertEquals(0, player.getCurrentGold());
// testCollectGold
// testLoseGold
//start point/ goal point have negative num
Play.setStartingPoint(0,-2);
```

3. Test Game:

```
//testPlayLocation. After each move
   assertEquals(0, MazeGame.getPlayerLocation()[1]);
   MazeGame.move(SOUTH);
   assertEquals(1, newGame.getPlayerLocation()[0]);
// testCurrentGold, after moving
```