UML Class Diagram for Board class

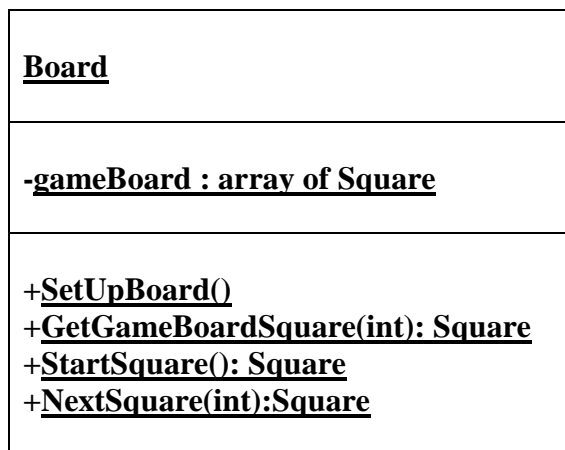| Board |
| --- |
| **-gameBoard : array of Square** |
| **+SetUpBoard()**
+GetGameBoardSquare(int): Square
+StartSquare(): Square
+NextSquare(int):Square |

Underlining in UML class diagram indicates static modifier.

A number of constants should be declared in this class to represent

- **The Start and Finish square's number, ie 0 & 55**
- **The number of playing squares on the board, 54**
- **Any other magic numbers associated with the gameBoard**

SetUpBoard():

Initialises each element (square) (1-54) of the gameBoard to be the appropriate type either an "ordinary" square, a Win square, a Lose square or Chance square.

- **Squares 10, 20, 30, 40 and 50 are "Lose" squares.**
- **Squares 5, 15, 25, 35 and 45 are "Win" squares.**
- **Squares 6, 12, 18, 24, 36, 42, 48 and 54 are "Chance" squares**
- **All other numbered Squares are "ordinary" squares.**

Element 0 is the Start square and element 55 is the Finish square.

GetGameBoardSquare(int): returns the square object with the specified number supplied as the parameter

StartSquare(): returns the Start square (element 0 of gameBoard)

NextSquare(int): returns the square after the specified square number supplied as the parameter. Note the Finish square (element 55) does not have a "Next Square".

Both GetGameBoardSquare(int) and NextSquare(int) should throw a general exception, `Exception`, **with an appropriate error message if the parameter is not a valid square number.**

This class is completely specified, no more additional variables or methods are to be added. Though you are expected to have additional named constants so that the class is not full of magic numbers. Most of these constants will be public so that other classes can access them.