class Nim

- This class is designed to represent the "board" and "game state" for a game of Nim. The game board will be represented as an array of `int`, where each array index represents a pile of matches and the number stored at each index represents the number of matchsticks in that pile.

Instance variables

- `int piles[]`: an array that represent the piles of matchsticks
- `int player`: represents the current player (the player about to make a move). Valid values for `player` can be either `0` or `1`.

Constructors

- `public Nim(int numPiles, int minMatches, int maxMatches)`:
  - The `numPiles` parameter represents the number of piles of matches in the game. A valid game must have at least one pile of matches.
  - The `minMatches` parameter represents the minimum number of matches that can be in a pile at the start of a new Nim game (inclusive)
  - The `maxMatches` parameter represents the maximum number of matches that can be in a pile at the start of a new Nim game (inclusive)
- `public Nim(int numPiles)`: constructs a Nim game with `numPiles` piles of matches, and between 1 and five matches in each pile.
  - The `numPiles` parameter represents the number of piles of matches in the game. A valid game must have at least one pile of matches.
- `public Nim()`: constructs a Nim game with 3 piles of matches, and between 1 and 5 matches in each pile.

Methods

- `protected void populateEmptyBoard(int minMatches, int maxMatches)`
  - Randomly generates a board
  - The `minMatches` parameter represents the minimum number of matches that can populate any given pile
  - The `maxMatches` parameter represents the maximum number of matches that can populate any given pile
- `public boolean isValidMove(int whichPile, int numMatches)`:
  - Returns true if the move is valid (a correctly specified pile has at least as many matches as requested), and false otherwise.
  - `whichPile` specifies which pile to remove matches from (0-indexed)
  - `numMatches` specifies how many matches to move
- `public void makeMove(int whichPile, int numMatches)`:
  - Removes the specified number of matches from the specified pile (without checking the legality)
  - `whichPile` The pile to remove matches from
  - `numMatches` How many matches to remove
- `public boolean isGameOver()`:
  - Returns true if the game is over (i.e., no more matches in any piles).
- `public String toString()`:
  - A string representation of the current game. The String contains the current player, as well as a row for each pile of matches. A pile is represented by a numeric index and a series of vertical bars, one per match.