Package edu.cmu.cs780.hw3

Class Connect4

java.lang.Object
    edu.cmu.cs780.hw3.Connect4

```
public class Connect4
extends Object
```

Homework 3

Author:

Zihao Yang (zihaoy2@andrew.cmu.edu), Yun Lee (yunl3@andrew.cmu.edu)

Constructor Summary

Constructors

Constructor	Description
`Connect4()`	Initializes the game with an empty board, and randomly decides the starting player.
`Connect4(int[][] newBoard, int checkerCount, int currentPlayer)`	Creates a Connect4 game object using a provided game board, the current number of checkers on the board, and the id of the starting player.

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
boolean	`hasWinner()`	Determines if there is a winner based on the current game board state.
boolean	`hasWinner(int[][] gameBoard)`	Determines if there is a winner on a given game board.
boolean	`isGameOver()`	Determines if the game has reached an end condition or it can be continued.
void	`placeChecker(int columnNum)`	Places a checker in the specified column.
String	`toString()`	Transforms the board into a `String` to be printed in the terminal.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Details

Connect4

```
public Connect4()
```

Initializes the game with an empty board, and randomly decides the starting player.

Connect4

```
public Connect4(int[][] newBoard,
                int checkerCount,
                int currentPlayer)
```

Creates a Connect4 game object using a provided game board, the current number of checkers on the board, and the id of the starting player.

Parameters:

`newBoard` - input game board represented as a 2D array.

`checkerCount` - current number of checkers on the board.

`currentPlayer` - the ID of the player set to play next.

Method Details

isGameOver

```
public boolean isGameOver()
```

Determines if the game has reached an end condition or it can be continued. This method also prints out the game status to the console.

Returns:

`true` if the game has reached a draw or a win condition, `false` if the game is not over.

toString

```
public String toString()
```

Transforms the board into a `String` to be printed in the terminal.

Overrides:

`toString` in class `Object`

Returns:

a `String` of formulated board.

hasWinner

```
public boolean hasWinner()
```

Determines if there is a winner based on the current game board state.

Returns:

`true` if the current player wins, `false` otherwise.

hasWinner

```
public boolean hasWinner(int[][] gameBoard)
```

Determines if there is a winner on a given game board.

Parameters:

`gameBoard` - input Game board represented as a 2D array.

Returns:

`true` if there is a winner, `false` otherwise.

placeChecker

```
public void placeChecker(int columnNum)
```

Places a checker in the specified column. If the chosen column is valid (not full), the current player's checker is added to the game board and the role is switched to the other player. If the chosen column is already full, an appropriate message is displayed.

Parameters:

`columnNum` - the index-based column number (0 to 6 inclusive) where the current player wants to place their checker.