**CLASS:** Node

Package model

**IMPLEMENTATION**

*The class Node represents valid positions for a piece on the game board. Nodes are indexed starting at the top left corner of the board then continuing clockwise, going from outside to inside.*

**INHERITS**

None

**USES**

java.util.ArrayList

**ACCESS** **PROGRAMS**

Node (position, x, y : integer; joints : ArrayList of Integer ) : Node

*Declares a node with a position in respect to other nodes, coordinates with respect to the board, and a list of connected nodes*

Node (position, x, y : integer; coordinates : integer[] ) : Node

*Declares a node with a position in respect to other nodes, x and y coordinates with respect to the board, and a position with respect to the coordinate system used by the pieces*

Node (position, x, y : integer; coordinates : integer[] ) : Node

*Declares a node with a position in respect to other nodes, as well as x and y coordinates with respect to the board*

getX ( ) : integer

*Returns the x-coordinate of the node*

getY ( ) : integer

*Returns the y-coordinate of the node*

getCoords ( ) : integer[]

*Returns the coordinate of the node*

getConnections ( ) : ArrayList of type Integer

*Returns a list of all nodes connected to the current node*

setConnections (ArrayList of type integer) : void

*Sets the connections of the current node*

addConnection (pos : integer) : void

*Adds a connection from the current node to the one with position 'pos'*

isConnected (first, second : Node) : boolean

*Returns whether the two nodes are connected based on their list of connections*

**RESTRICTIONS**

*None*