I. start *not in JUnit testing*
    A. LOOP Iterator
        1. Zero
            a) When no arguments are entered, Gomoku runs its default game (19 by 19 board, 5 in a row to win)
        2. One
            a) When one argument is entered, Gomoku creates a default board (19 by 19) with the argument entered as the number in a row needed to win
        3. Many
            a) When three arguments are entered, Gomoku creates a game in which the number in a row needed to win is the first argument, the number of rows is the second argument, and the number of columns is the third argument
        4. First
            a) When the iterator runs through the first time, all metrics are adjusted correctly
        5. Middle
            a) The iterator only runs once so the middle is the first
        6. Last
            a) The iterator only runs once so the last is the first
    B. CONDITIONAL If number of arguments is 2
        1. True: If the number of arguments is 2 the first argument is the number of rows and the second argument is the number of columns
            a) CONDITIONAL If columns <= 0
                (1) True: The number of columns becomes 19
                (2) False: Nothing happens
            b) CONDITIONAL If rows <= 0
                (1) True: The number of rows becomes 19
                (2) False: Nothing happens
        2. False: Nothing happens
    C. CONDITIONAL If rows <= 0
        1. True: The number of rows becomes 19
        2. False: Nothing happens
    D. CONDITIONAL If columns <= 0
        1. True: The number of columns becomes 19
        2. False: Nothing happens
    E. LOOP Forming the rows of the board
        1. Zero
            a) When the user inputs for zero rows, 19 is the default so 19 rows are formed
        2. One
            a) When the user inputs for one row, one row is formed

        3. Many
            a) When the user inputs for 10 rows, 10 rows are formed
        4. First
            a) The first row is formed when the user wants 10 rows
        5. Middle
            a) The fourth row is formed when the user wants 10 rows
        6. Last
            a) The tenth row is formed when the user wants 10 rows
    F. LOOP Forming the columns of the board
        1. Zero
            a) When the user inputs for zero columns, 19 is the default so 19 columns are formed
        2. One
            a) When the user inputs for one column, one column is formed
        3. Many
            a) When the user inputs for 10 columns, 10 columns are formed
        4. First
            a) The first column is formed when the user wants 10 columns
        5. Middle
            a) The fourth column is formed when the user wants 10 columns
        6. Last
            a) The tenth column is formed when the user wants 10 columns

II. numberInLine
    A. LOOP Counting the number of pieces in a row
        1. Zero
            a) When counting the number of pieces in a row on an empty square, the number of empty spaces in a row will be returned
        2. One
            a) When counting the number of pieces in a row when there is one, one is returned
        3. Many
            a) When counting the number of pieces in a row when there are three, three is returned
        4. First
            a) When counting the number of pieces in a row when broken after the first piece, 1 is returned
        5. Middle
            a) When counting the number of pieces in a row when broken after the 2nd piece, 2 is returned
        6. Last
            a) When counting the number of pieces in a row when broken after the 3rd piece, 3 is returned

III. isOpen

A. LOOP Going until the end of a line of pieces
    1. Zero
        a) When there are zero in a row, true is returned because the space itself is empty
    2. One
        a) When there is one in a row, true or false is returned dependent on the emptiness
    3. Many
        a) When there are three in a row, true or false is returned dependent on the emptiness
    4. First
        a) After the first piece, the next space is open, true is returned
    5. Middle
        a) After the second piece, the next space is open, true is returned
    6. Last
        a) The last time through the loop, when the next space is an opposing piece, false should be returned

B. CONDITIONAL Determining if the end of a line of pieces is empty
    1. True: When a series of pieces in a row ends with an empty space, true is returned
    2. False: When a series of pieces in a row ends with either an opponent's piece or the edge of the board, false is returned

IV. isFourFour

A. LOOP Going through all of the directions
    1. Zero
        a) If there aren't any groups of four in a row that will be formed, the number of fours in a row remains zero, and true is returned
    2. One
        a) If there is one group of four in a row that is formed, the number of fours in a row becomes one, and true is returned
    3. Many
        a) If there are 2 groups of four in a row being formed, the number of fours in a row becomes 2, and false is returned
    4. First
        a) The first element of the Direction enum is evaluated (LEFT)
    5. Middle
        a) The fourth element of the Direction enum is evaluated (DOWN)
    6. Last
        a) The last element of the Direction enum is evaluated (DOWNRIGHT)

B. CONDITIONAL If a pair of directions forms a four-four
    1. True: The number of four in a rows formed increments by one
    2. False: Nothing happens

C. CONDITIONAL If the four-four rule has been broken
    1. True: False is returned
    2. False: True is returned

V. isThreeThree
  A. LOOP Going through all of the directions
    1. Zero
      a) If there aren't any groups of open three in a row that will be formed, the number of three in a row remains zero and true is returned
    2. One
      a) If there is one group of open three in a row that is formed, the number of three in a row becomes one and true is returned
    3. Many
      a) If there are 2 groups of three in a row being formed, the number of three in a row becomes 2 and false is returned
    4. First
      a) The first element of the Direction enum is evaluated (LEFT)
    5. Middle
      a) The fourth element of the Direction enum is evaluated (DOWN)
    6. Last
      a) The last element of the Direction enum is evaluated (DOWNRIGHT)
    7.
  B. CONDITIONAL If a direction is open
    1. True: The number in a row in that direction pair is incremented by the number in a row in that direction and the number in that direction pair evaluated is incremented by one
    2. False: Nothing happens
  C. CONDITIONAL If a pair of directions forms a three-three
    1. True: The number of three in a rows formed increments by one
    2. False: Nothing happens
  D. CONDITIONAL If the three-three rule has been broken
    1. True: False is returned
    2. False: True is returned

VI. isWon
  A. LOOP Going through all of the directions
    1. Zero
      a) If there aren't any groups of five in a row that will be formed, nothing happens
    2. One
      a) If there is one group of five in a row that will be formed, the winner is returned
    3. Many

            a) If there are 2 groups of five in a row that will be formed, the winner is returned
        4. First
            a) The first element of the Direction enum is evaluated (LEFT)
        5. Middle
            a) The fourth element of the Direction enum is evaluated (DOWN)
        6. Last
            a) The last element of the Direction enum is evaluated (DOWNRIGHT)
    B. CONDITIONAL The game has been won
        1. True: The winner is returned
        2. False: EMPTY is returned

VII. handle *not in JUnit testing*
    A. CONDITIONAL If moves are allowed
        1. True: The board and the logic behind the board are changed
            a) CONDITIONAL If there isn’t a piece in the space and no rule is violated
                (1) True: The game may have been won, a piece will be placed, and the turn will switch to the next player
                    (a) CONDITIONAL If the game has been won
                        (i) True: A statement is printed indicating who won and no moves can be made subsequently
                        (ii) False: Nothing happens
                    (b) CONDITIONAL If the current player is white
                        (i) True: A white piece appears in that space on the board and the turn changes to black
                        (ii) False: A black piece appears in that space on the board and the turn changes to white
                (2) False
        2. False: Nothing happens

For the two methods without JUnit testing, I tested the if statements and for loops by playing the game as they dealt with user input.