Operation #8 – Connect Four
CS 1400 - Satterthwaite

Goal: To gain more experience working with two dimensional arrays, as well as working with methods.

Assignment: Implement a Connect 4 game with the following requirements:

- Name the game `ConnectFour`.
- Allow the user to input their name, they will be playing against the computer.
- The grid must have 6 rows and 7 columns.
- User only inputs the column he or she wants to drop a tile into.
- Your program must be able to compute winners if 4 consecutive tiles are found horizontally, vertically, or diagonally.
- Validate input to only allow for integers when a user enters a row and column, handle any errors thrown from incorrect input.
- Handle errors if a user enters an integer that is larger than your grid (i.e. 9 would throw an error because there are only 7 columns).
- Create a method called `checkRows` that will check if any rows contain four tiles in a row.
- Create a method called `checkColumns` that will check if any columns contain four tiles stacked on top of each other.
- Create a method called `checkDiagonals` that will check if four tiles connect diagonally.

Strategies: The `TicTacToe` program gives a great template to start with. You can find it in this week's module on Canvas.

One of the harder parts of this assignment is handling how to drop a tile into a column. So you need to do a while loop, and keep looping searching down the rows until it finds one with tile OR it can't search anymore because it hit the bottom. Then place the tile in the prior open spot.

Computing a winner takes a bit of work. The `TicTacToe` program demonstrates how to use a `for` loop to find row winners. You want to use something similar. Except, instead of just searching down the left-most column, you need to search down column 0, 1, 2, and 3. This will require an outer `for` loop for the rows, and an inner `for` loop for the columns. The idea is this:

```
for rows 0 through 5:
 for columns 0 through 3:
   if (the value isn't null
      AND the row/col value equals the row/col+1 value
      AND the row/col value equals the row/col+2 value
      AND the row/col value equals the row/col+3 value)
       set a winner found boolean variable to true.
```

Columns work in a similar way. You will need to look from rows 0 through 2 and columns 0 through 6.

Diagonals are slightly trickier. To go down and to the right, you need to look from rows 0 through 2 and columns 0 through 3. To go down and to the left, you need to look from rows 0 through 2 and columns 3 through 6.