Panel 1

Last homework

2D Arrays

Code to manipulate 2D Arrays

Tick-Tack-Toe program

HW: design main method

Quiz next - 0

Panel 2

Quiz - (part I)

1. True or false:

a) A two-dimensional array defined as int A[][] = new int[5][4] has room for at most 9 integers.

b) If double A[][] = new double[4][5]; then A[3] is an array.

c) If char A[][] = new char[9][10], then the index of the last column is 9.

d) For a two dimensional array, the number of rows must always be bigger than the number of columns.
Panel 3

Quiz (Part 2)

1. If \( A[][] = \text{new int}[4][5] \), then what's the code to add the number in the upper right to the one in the lower left corner of the corresponding table (you may use unnamed constants in this example).

2. Suppose \( A \) is a 2-dimensional array of integers with 4 rows and 5 columns. Write a segment of Java code that finds the sum of all entries in the table.

Panel 4

Design the main method for Tic Tac Toe

```java
main

Need to create board
Need to have 'X' and 'O'

'O' makes move < check if 'O' wins

'X' makes move < check if 'X' wins

check if tie <

repeat if nobody won
```

```
O X O
O X X
X O O
```
Panel 5

Classes

A class is the fundamental structure in Java. It is composed of two sections, fields to contain data and methods to manipulate data or perform an action. Every class represents a new reference type that can be used by other classes. Classes are defined using the syntax:

```java
[public] class ClassName [extends ClassName] [implements InterfaceList] {
    /* list of fields */
    /* list of methods */
}
```

where `ClassName` is the name of the class, `extends` indicates that the class is derived from another class and `implements` indicates that the class has attributes in common with one or more interfaces.

![Representation of a Class](image)

Panel 6

```java
Main for Tic Tac Toe:

Field: char board[3][3]; // array for N, initializes
   create Board(); // the board via new
   show Board();
   while (! game Over)
   {
      make move ('o'); < show Board();
      has won ('o'); // checkIfTie();
      if (gameOver) make move ('x'); < show Board();
      has won ('x'); // checkIfTie();
      checkIfTie();
   }
```

```

X | 0 | X
0 | X | 0
X | 0 | X
```
Panel 7

```java
createBoard()
{
    // ask user for N
    board = new char[N][N];
    // initialize each slot in the board
    for (int row = 0; row < N; row++)
    {
        for (int col = 0; col < N; col++)
        {
            board[row][col] = ' _ ';
        }
    }
}
```

Panel 8

```java
showBoard();
did that!

makeMove (char player)

// Print: "player , your move!

for row ? add/subtract 1?

for col ?

check if legal move

loop if necessary!
```
Panel 9

```
has Won(char player)
{
  game Over = check Rows (player);
  if (!game Over)
    game Over = check Cols (player);
  else if (!game Over)
    game Over = check Main Diag (player);
  else if (!game Over)
    game Over = check Minor Diag (player);
}
```

Panel 10

```
boolean check Main Diag (char player)
{
  int sum = 0;
  for (int i = 0; i < board length; i++)
    {
      if (board [i][i] == player)
        sum ++;
    }
  return (sum == board length);
}
check Minor Diag
check Rows
check Cols
```

Left to do!
Next assignment:

Finish Tic Tac Toe

by next Monday

Friday is QA session!