

Panel 1

Last time

- 2D Arrays
- Code to manipulate 2D Arrays
- Tic Tac Toe program

HW: design main method

Quiz next →

1

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.

2

Panel 3

Quiz (part 2)

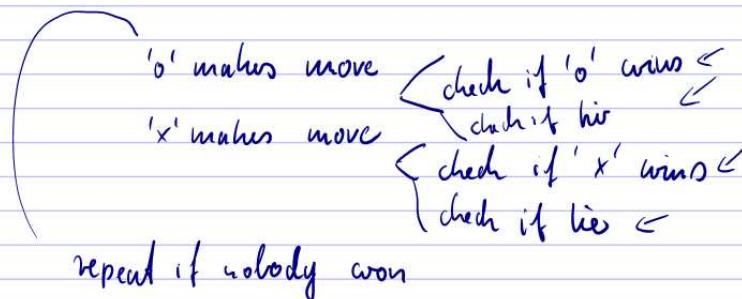
1. If `int A[][] = new int[4][5]`, then what's the code to add the number in the **upper right** to the one in the **lower left** corner of the corresponding table (you may use unnamed constants in this example).
2. Suppose `A` is a 2-dimensional array of integers with 4 rows and 5 columns. Write a segment of Java code that finds the sum of all entries in the table.

3

Panel 4

Design the main method for Tic Tac Toemain:

Need to create board
 Need to have 'x' and 'o'



o	x	o
o	x	x
x	o	o

4

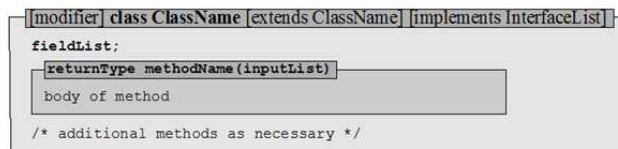
Panel 5

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:

```
[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.



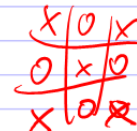
Representation of a Class

Panel 6

main for Tic Tac Toe : Board is $n \times n$ array of chars

```
Field: char board[][];
→ some game char = false;
   create Board();
   ⇒ show Board();
   while (! game Over)
   {
     make Move ('o');
     has Won ('o');
     check If Tie();
     make Move ('x');
     has Won ('x');
     check If Tie();
   }
```

notes for M, initializes the board via new
 if (! game Over)
 more often?



Panel 7

```

createBoard()
{
    // ask user for M
    board = new char [M][M];
    // initialize each slot in the board
    for (int row = 0; ... )
    {
        for (int col = 0; ... )
        {
            board [row][col] = ' _ '
        }
    }
}

```

minus sign

7

Panel 8

show Board()

did that!

make Move (char player)

// print: "player, your move"

ask for row ? add/subtract 1?
 ask for col ?
 check if legal move
 loop if necessary!

X	O	X
X	O	

8

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);
}

```

9

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);
}

```

a ₀₀	a ₀₁	a ₀₂	a ₀₃
a ₁₀	a ₁₁	a ₁₂	a ₁₃
a ₂₀	a ₂₁	a ₂₂	a ₂₃
a ₃₀	a ₃₁	a ₃₂	a ₃₃

{ check Minor Diag } left to do!
 { check Rows
 check Cols }

10

Panel 11

Next assignment:

finish Tic Tac Toe

by next Monday

Friday is QA session!