**Java**

One and Two Dimensional Array Quiz

**Directions: Answer all questions on your own.**

**1) Complete the following methods: rm length index col is switch**

// print all of the ints in values in any format

public void printValues(int[] values){

for(int i = 0; i<values.length;i++)

System.out.println(values[i] + " ");

}

public int sum(int[] values){

int sum =0;

for(int o=0; o< values.length;o++)

sum+= values[o];

return sum;

}

public int max(int[] values){

int max = -9999;

for(int i=0;i<values.length;i++)

max= Math.max(max,values[i]);

return max;

}

// returns the index of valToFind

// if valToFind is not in values, return -1

public int findIndex(int valToFind, int[] values){

**int index = -1;**

**for(int j=0;j<values.length;j++){**

**if(values[j] == valToFind){**

**index = valToFind;**

**}**

**else{**

**return -1;**

**}**

**}**

**return index;**

**}**

**2)** What is a one dimensional array?

**A one dimensional array is an array that holds values at certain indexes.**

**3)** In the maximum method, what did you initialize your maximum value to? Why?

**-9999. This is because since the number is so low the number that the user enters has to be the new max and that can then be used for comparison for another maximum number.**

**4)** Can the size of an array change?

**No because it has a fixed size.**

**5)** Highlight the objects in real life that you would represent with a two dimensional array in a program.

Grocery List Connect 4 board Seats in a classroom

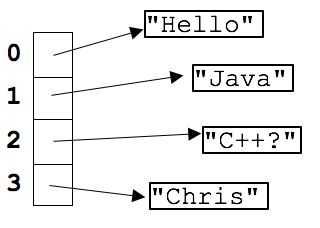
Playlist Test grades for one class Your report card

**6)** What is a two dimensional array?

A two dimensional array is an array that consists of arrays inside of it(refers to array). Which can hold values at certain indexes.

**7)** If you did not highlight one of the items in number 5 list it, and either explain why not, or what data type you would have used.

|  |  |
| --- | --- |
| Item | Explanation |
| Grocery List | Since the grocery list can only hold one set of values(the items to buy) it is a 1 dimensional array. |
| Playlist | Since the playlist can only hold one set of values(the songs to play) it is a one dimensional array. |
| Test Grades for one class | SInce one class can only hold one set of values(the grades for the tests) it is a one dimensional array. |
| Seats in a classroom. | Since one classroom can only hold one value( the seats)) it is a one dimensional array. |
|  |  |
|  |  |

**6)** If this is a picture of a String[4], draw a picture of an int[2][2].

**7)** How can you find the number of rows in a two dimensional array

called grid

**a)** If we are using row major order?

grid.length;

**b)** If we are using column major order?

grid[0].length;

**8)** How can you find the number of columns in a two dimensional array called grid

**a)** If we are using row major order?

grid[0].length;

**b)** If we are using column major order?

grid.length;

**9)** How do you know your answer from 8a?

It tells you the number of columns because we use 0 to specify a row in the array grid. Then .length finds the length of that array which gives us the number of columns.

**10)** Complete the following method that prints out the array grid in row major order.

public static void printRowMajor(int[][] grid){

for(int rows=0; rows<grid.length; rows++)

for(int cols=0; cols<grid[0].length;cols++)

System.out.println(grid[rows][cols]);

}

**11)** Complete the following method that prints out the array grid in column major order.

public static void printColMajor(int[][] grid){

for(int cols=0; cols<grid[0].length;cols++)

for(int rows; rows<grid.length;rows++)

System.out.println(grid[cols][rows]);

}

**12)** Complete the following method that finds the average of row n in grid if we are using row major order.

public static double averageRow(int n, int[][] grid){

double sum =0;

for(int i=0;i<grid[n].length;i++)

sum+=grid[n][i];

return (sum/i);

}

**13)** Complete the following method that finds the sum of all of the positive values in grid that are less than n.

public static int sumPosLessThan(int n, int[][] grid){

int sum =0;

for(int i=0;i<grid.length;i++){

for(int j = 0; j<grid[0].length;j++){

if(grid[i][j]>-1 && grid[i][j]<n){

sum+=grid[i][j];

}

}

}

return sum;

}

**14)** You are creating a Tic Tac Toe game and need to write a method to see if a player has won. The game is represented by a two dimensional array of chars.

Here is an example representation of a game where X has just won:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 | 2 |
| 0 | 'x' | 'o' | 'x' |
| 1 | 'o' | 'x' | 'o' |
| 2 | 'x' | '-' | '-' |

A space has the value 'x' if X has gone in that spot

A space has the value 'o' if O has gone in that spot

A space has the value '-' if no one has gone in that spot

Your method should return 'x' if there are three X's in a row, 'o' if there are three O's in a row, '-' if no one has won yet, and 't' if no one has won yet, but all of the spaces are taken up.

public char checkWin(char[][] board){

//Check hor

for(int i=0;i<board.length;i++)

for(int j=0; board[0].length;j++

{

for(int rows = 0;rows<3;rows++)

{

for(int cols=0;cols<3;cols++){

if( ((board[rows][cols] == 'x') && (board[rows+1][cols+1] == 'x')) && (board[rows+2][cols+2] == 'x') ) {

return 'x';

}

else if( ((board[rows][cols] == 'o') && (board[rows+1][cols+1] == 'o')) && (board[rows+2][cols+2] == 'o') ){

return 'o';

}

}

}

}

// check vert

for(int col = 0; col<board[0].length;col++){

for(int row=0;row<board.length;row++){

if( ((board[col][row] == 'x') && (board[col+1][row+1] == 'x')) && (board[col+2][row+2] == 'x') ) {

return 'x';

}

if( ((board[col][row] == 'o') && (board[col+1][row+1] == 'o')) && (board[col+2][row+2] == 'o') ){

return 'o';

}

}

}

// check diags

if( (((board[0][0] == 'x') && board[1][1] == 'x') && board[2][2] == 'x') ){

return 'x';

}

else if( (((board[0][0] == 'o') && board[1][1] == 'o') && board[2][2] == 'o') ){

return 'o';

}

//diag 2

else if( (((board[2][0] == 'x') && board[1][1] == 'x') && board[0][2] == 'x') ){

return 'x';

}

else if( (((board[2][0] == 'o') && board[1][1] == 'o') && board[0][2] == 'o') ){

return 'o'';

}

else{

return '-';

}

}