

Exercise walk-through

2D Array Creation and Access ■ 4.11

eXercise

You must be able to

- create a **two-dimensional array** 二维数组 arranged in rows and columns
- access an element using two indices, `grid[row][col]`, both from 0
- find the number of rows and columns

Method — Creating and accessing

```
k+ktinto[o]o[o]+w ngrid+w o+=w knew+w k+ktinto[l+m+mi3o]o[l+m+mi4o]p;+w
ngrido[l+m+mi1o]o[l+m+mi2o]+w o+=w l+m+mi9p;+w                                c+c1// row
```

Both indices start at 0.

Method — Dimensions

- `grid.length` — the number of **rows**.
- `grid[0].length` — the number of **columns**.

Practice 2.1 ■ 1 mark

State how to access the element in row 1, column 2 of `grid`.

Solution 2.1

Method: Creating and accessing

Worked steps

```
grid[1][2] B1.
```

Practice 2.2 ■ 2 marks

For `int[] [] g = new int[3][5];`, state the number of rows and columns.

Solution 2.2

Method: Dimensions

Worked steps

3 rows and 5 columns **B1** **B1**.

Practice 2.3 ■ 1 mark

State how to find the number of rows of a 2D array `g`.

Solution 2.3

Method: Dimensions

Worked steps

```
g.length B1.
```

Exam-style 3.1 ■ 1 mark

A 2D array element is accessed with

A `grid[row]`

B `grid[row][col]`

C `grid(row, col)`

D `grid.get(row)`

Solution 3.1

A `grid[row]`

B `grid[row][col]` 

C `grid(row, col)`

D `grid.get(row)`

Worked steps

B.

Exam-style 3.2 ■ 1 mark

`grid.length` gives the number of

- A** columns
- B** rows
- C** total elements
- D** bytes

Solution 3.2

Method: Dimensions

A columns

B rows



C total elements

D bytes

Worked steps

B.

Exam-style 3.3 ■ 1 mark

```
int[] [] g = new int[4][6];
```

- (a) State the number of rows.
- (b) State the number of columns.
- (c) State how to access the element in row 0, column 0.

Solution 3.3

Worked steps

(a) 4 **B1**. (b) 6 **B1**. (c) $g[0][0]$ **B1**.