

آرایه های چند بعدی

آرایه‌های چند بعدی، آرایه‌هایی هستند که برای دسترسی به هر یک از عناصر آنها باید از چندین اندیس استفاده کنیم. یک آرایه چند بعدی را می‌توان مانند یک جدول با تعدادی ستون و ردیف تصور کنید. با افزایش اندیسها اندازه ابعاد آرایه نیز افزایش می‌یابد و آرایه‌های چند بعدی با بیش از دو اندیس به وجود می‌آیند. نحوه ایجاد یک آرایه با دو بعد به صورت زیر است :

```
datatype[][] arrayName = new datatype[lengthX][lengthY];
```

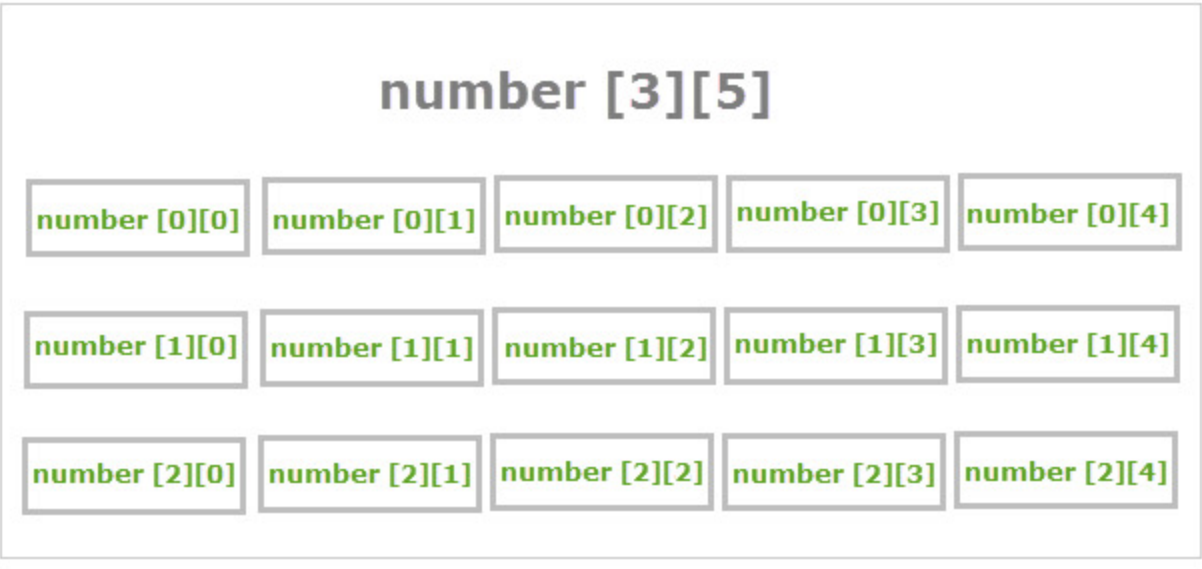
و یک آرایه سه بعدی به صورت زیر ایجاد می‌شود :

```
datatype[][][] arrayName = new datatype[lengthX][lengthY][lengthZ];
```

می‌توان یک آرایه با تعداد زیادی بعد ایجاد کرد به شرطی که هر بعد دارای طول مشخصی باشد. به دلیل اینکه آرایه‌های سه بعدی یا آرایه‌های با بیشتر از دو بعد بسیار کمتر مورد استفاده قرار می‌گیرند اجازه بدهید که در این درس بر روی آرایه‌های دو بعدی تمرکز کنیم. در تعریف این نوع آرایه ابتدا نوع آرایه یعنی اینکه آرایه چه نوعی از انواع داده را در خود ذخیره می‌کند را مشخص می‌کنیم. سپس دو جفت کروشه قرار می‌دهیم. به تعداد کروشه‌ها توجه کنید. اگر آرایه ما دو بعدی است باید 2 جفت کروشه و اگر سه بعدی است باید 3 جفت کروشه قرار دهیم. سپس یک نام برای آرایه انتخاب کرده و بعد تعریف آنرا با گذاشتن کلمه new ، نوع داده و طول هر بعد آن کامل می‌کنیم. در یک آرایه دو بعدی برای دسترسی به هر یک از عناصر به دو مقدار نیاز داریم یکی مقدار X و دیگری مقدار Y که مقدار x نشان دهنده ردیف و مقدار Y نشان دهنده ستون آرایه است البته اگر ما آرایه دو بعدی را به صورت جدول در نظر بگیریم یک آرایه سه بعدی را می‌توان به صورت یک مکعب تصور کرد که دارای سه بعد است و x طول، Y عرض و Z ارتفاع آن است. یک مثال از آرایه دو بعدی در زیر آمده است :

```
int[][] numbers = new int[3][5];
```

کد بالا به کامپایلر می‌گوید که فضای کافی به عناصر آرایه اختصاص بده (در این مثال ۱5 خانه). در شکل زیر مکان هر عنصر در یک آرایه دو بعدی نشان داده شده است.



مقدار 3 را به x اختصاص می‌دهیم چون 3 سطر و مقدار 5 را به Y چون 5 ستون داریم اختصاص می‌دهیم. چطور یک آرایه چند بعدی را مقدار دهی کنیم؟ چند راه برای مقدار دهی به آرایه‌ها وجود دارد. یک راه این است که مقادیر عناصر آرایه را در همان زمان تعریف آرایه، مشخص کنیم :

```
datatype[][] arrayName = { { r0c0, r0c1, ... r0cX },
                             { r1c0, r1c1, ... r1cX },
                             .
                             .
                             { rYc0, rYc1, ... rYcX } };
```

به عنوان مثال :

```
int[][] numbers = { { 1, 2, 3, 4, 5 },
                    { 6, 7, 8, 9, 10 },
                    { 11, 12, 13, 14, 15 } };
```

و یا می‌توان مقدار دهی به عناصر را به صورت دستی انجام داد مانند :

```
array[0][0] = value;
array[0][1] = value;
array[0][2] = value;
array[1][0] = value;
array[1][1] = value;
array[1][2] = value;
array[2][0] = value;
array[2][1] = value;
array[2][2] = value;
```

همانطور که مشاهده می‌کنید برای دسترسی به هر یک از عناصر در یک آرایه دو بعدی به سادگی می‌توان از اندیسهای X و Y و یک جفت کروشه مانند مثال استفاده کرد.

گردش در میان عناصر آرایه‌های چند بعدی

گردش در میان عناصر آرایه‌های چند بعدی نیاز به کمی دقت دارد. یکی از راههای آسان استفاده از حلقه foreach و یا حلقه for تو در تو است. اجازه دهید ابتدا از حلقه foreach استفاده کنیم.

```
1 package myfirstprogram;
2
3 public class MyFirstProgram
4 {
5     public static void main(String[] args)
6     {
7         int[][] numbers = { { 1, 2, 3, 4, 5 },
8                             { 6, 7, 8, 9, 10 },
9                             { 11, 12, 13, 14, 15 }
10        };
11
12        for (int[] number : numbers)
13        {
14            for (int num : number)
15            {
16                System.out.print(num + " ");
17            }
18        }
19    }
20 }
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

مشاهده کردید که گردش در میان مقادیر عناصر یک آرایه چند بعدی چقدر راحت است. به وسیله حلقه foreach نمی‌توانیم انتهای ردیفها را مشخص کنیم. برنامه زیر نشان می‌دهد که چطور از حلقه for برای خواندن همه مقادیر آرایه و تعیین انتهای ردیف‌ها استفاده کنید.

```
1 package myfirstprogram;
2
3 public class MyFirstProgram
4 {
5     public static void main(String[] args)
6     {
7         int[][] numbers = { { 1, 2, 3, 4, 5 },
8                             { 6, 7, 8, 9, 10 },
9                             { 11, 12, 13, 14, 15 }
10        };
11
12        for (int row = 0; row < numbers.length; row++)
13        {
14            for (int col = 0; col < numbers[row].length; col++)
15            {
16                System.out.print(numbers[row][col] + " ");
17            }
18
19            //Go to the next line
20            System.out.println();
21        }
22    }
23 }
```

1 2 3 4 5
6 7 8 9 10
11 12 13 14 15

همانطور که در مثال بالا نشان داده شده است با استفاده از یک حلقه ساده for نمی‌توان به مقادیر دسترسی یافت بلکه به یک حلقه for تو در تون نیاز داریم. در اولین حلقه for (خط ۱2) یک متغیر تعریف شده است که در میان ردیف‌های آرایه (row) گردش می‌کند. این حلقه تا زمانی ادامه می‌یابد که مقدار ردیف کمتر از طول اولین بعد باشد. در این مثال از خاصیت lenght کلاس Array استفاده کرده‌ایم این خاصیت طول آرایه را در یک بعد خاص نشان می‌دهد. به عنوان مثال برای به دست آوردن طول اولین بعد آرایه که همان تعداد ردیف‌ها می‌باشد از دستور numbers.length استفاده کرده‌ایم.

در داخل اولین حلقه for حلقه دیگری تعریف شده است (خط ۱4). در این حلقه یک شمارنده برای شمارش تعداد ستونهای (columns) هر ردیف تعریف شده است و در شرط داخل آن بار دیگر از خاصیت lenght استفاده شده است، ولی این بار به نوعی دیگر از آن استفاده می‌کنیم. همانطور که مشاهده می‌کنید ابتدا نام آرایه را نوشته‌ایم و سپس یک اندیس به آن اختصاص داده‌ایم و بعد از خاصیت lenght استفاده نموده‌ایم :

```
numbers[row].length
```

استفاده از row به عنوان اندیس باعث می‌شود که به عنوان مثال وقتی که مقدار ردیف (row) صفر باشد، حلقه دوم از [0][0] تا [0][4] اجرا شود. سپس مقدار هر عنصر از آرایه را با استفاده از حلقه نشان می‌دهیم، اگر مقدار ردیف (row) برابر 0 و مقدار ستون (col) برابر 0 باشد مقدار عنصری که در ستون ۱ و ردیف 1 (numbers[0][0]) قرار دارد نشان داده خواهد شد که در مثال بالا عدد ۱ است.

بعد از اینکه دومین حلقه تکرار به پایان رسید، فوراً دستورات بعد از آن اجرا خواهند شد، که در اینجا دستور System.out.println() که به برنامه اطلاع می‌دهد که به خط بعد برود. سپس حلقه با اضافه کردن یک واحد به مقدار row این فرایند را دوباره تکرار می‌کند. سپس دومین حلقه for اجرا شده و مقادیر دومین ردیف نمایش داده می‌شود. این فرایند تا زمانی اجرا می‌شود که مقدار row کمتر از طول اولین بعد باشد. حال بیایید آنچه را از قبل یاد گرفته‌ایم در یک برنامه به کار ببریم. این برنامه نمره چهار درس مربوط به سه دانش آموز را از ما می‌گیرد و معدل سه دانش آموز را حساب می‌کند.

```
1 package myfirstprogram;
2
3 import java.util.Scanner;
4 import java.text.MessageFormat;
5
6 public class MyFirstProgram
7 {
8     public static void main(String[] args)
9     {
10        Scanner input = new Scanner(System.in);
11
12        double[][] studentGrades = new double[3][4];
13        double total;
14
15        for (int student = 0; student < studentGrades.length; student++)
16        {
17            total = 0;
18
19            System.out.println(MessageFormat.format("Enter grades for Student {0}", student + 1));
20
21            for (int grade = 0; grade < studentGrades[student].length; grade++)
22            {
23                System.out.print(MessageFormat.format("Enter Grade #{0}: ", grade + 1));
24                studentGrades[student][grade] = input.nextDouble();
25                total += studentGrades[student][grade];
26            }
27
28            System.out.print(MessageFormat.format("Average is {0}", (total / studentGrades[student].length)));
29            System.out.println();
30        }
31    }
32 }
```

Enter grades for Student 1
Enter Grade #1: 92
Enter Grade #2: 87
Enter Grade #3: 89
Enter Grade #4: 95
Average is 90.75

Enter grades for Student 2
Enter Grade #1: 85
Enter Grade #2: 85
Enter Grade #3: 86
Enter Grade #4: 87
Average is 85.75

Enter grades for Student 3
Enter Grade #1: 90
Enter Grade #2: 90
Enter Grade #3: 90
Enter Grade #4: 90
Average is 90.00

در برنامه بالا یک آرایه چند بعدی از نوع double تعریف شده است (خط ۱2). همچنین یک متغیر به نام total تعریف می‌کنیم که مقدار محاسبه شده معدل هر دانش آموز را در آن قرار دهیم. حال وارد حلقه for تو در تو می‌شویم (خط ۱5). در اولین حلقه for یک متغیر به نام student برای تشخیص پایه درسی هر دانش آموز تعریف کرده‌ایم. از خاصیت lenght هم برای تشخیص تعداد دانش آموزان استفاده شده است. وارد بدنه حلقه for می‌شویم. در خط ۱2 مقدار متغیر total را برابر صفر قرار می‌دهیم. بعداً مشاهده می‌کنید که چرا این کار را انجام دادیم. سپس برنامه یک پیغام را نشان می‌دهد و از شما می‌خواهد که شماره دانش آموز را وارد کنید (۱ + student). عدد ۱ را به student اضافه کرده‌ایم تا به جای نمایش 0 Student، 1 Student شروع شود، تا طبیعی‌تر به نظر برسد. سپس به دومین حلقه for در خط 2۱ می‌رسیم. در این حلقه یک متغیر شمارنده به نام grade تعریف می‌کنیم که طول دومین بعد آرایه را با استفاده از studentGrades[student].length به دست می‌آورد. این طول تعداد نمراتی را که برنامه از سؤال می‌کند را نشان می‌دهد. برنامه چهار نمره مربوط به دانش آموز را می‌گیرد. هر وقت که برنامه یک نمره را از کاربر دریافت می‌کند، نمره به متغیر total اضافه می‌شود.

وقتی همه نمره‌ها وارد شدند، متغیر total هم جمع همه نمرات را نشان می‌دهد. در خط 28 معدل دانش آموز نشان داده می‌شود. معدل از تقسیم کردن total (جمع) بر تعداد نمرات به دست می‌آید. از studentGrades[student].length هم برای به دست آوردن تعداد نمرات استفاده می‌شود.