# آموزش زبان برنامه نویسی جاوا

## آرایه دوبعدی

## جلسه سی و هفتم

نویسنده : رحمان زارعی

جاوا را ساده،آسان و شیرین بنوشید!!!!

توجه!!!! این جلسه همان جلسه ۱۵ آموزش جاوا جاوالایک می باشد که در زمینه آرایه یک بعدی آموزش داده شده است.خب چرا باز در قالب یک جلسه جدید ارائه دادید؟! ما در جلسه ۱۵ تنها به آرایه یک بعدی پرداخته بودیم و الان قصد داریم آرایه دو بعدی را آموزش بدیم!!! برای درک بهتر مفهوم آرایه دوبعدی ،تمام آموزش های جلسه ۱۵ را در این جلسه قرار دادیم که ابتدا مروری بر آرایه یک بعدی داشته باشید بعد سراغ آرایه دو بعدی بروید!!!

اگر بر مفاهیم آرایه یک بعدی مسلط هستید!! خب مرور رو بی خیال و سریع بروید سراغ آموزش آرایه دو بعدی موجود در همین جلسه اموزشی ☺

## آرایه یک بعدی!!!!!

جاوا یک ساختمان داده برای ذخیره داده ها به نام آرایه فراهم کرده است.

ساختمان داده ها در کل برای ذخیره اطلاعات استفاده می شوند.اطلاعات چیه؟! اطلاعات منظور همون اطلاعات عددی ،کاراکتری، رشته ای ، اعشاری، شی ، کلاس ها و.... به زبون ساده بهتون بگم چیزی شبیه یک متغیر با ظرفیت و کاربرد بیشتر!!! بیشتر از این هم نیاز نیست بدونیم!!

خب آرایه چیه؟ آرایه ها یک مجموعه پی در پی عناصر با نوع یکسان و اندازه ثابت هستند.به زبان ساده تر! آرایه ها مجموعه ای از متغیر ها با نوع یکسان هستند!!

آرایه ها مثل متغیر ها برای ذخیره داده ها استفاده می شوند اما از آنچه که فکر میکنید مفید تر و کاربردی تر هستند!! اگر یک متغیر تنها می تواند یک مقدار بگیرد یک آرایه با اندازه n می تواند n مقدار بگیرد!!

اندازه آرایه چیه؟! به تعداد خانه های متوالی آرایه که داده ها (عناصر آرایه) درونش قرار میگیرد اندازه آرایه می گویند.

ساختار ذخیره داده ها در آرایه بصورت زیر می باشد:

فرض کنید ما یک آرایه با اندازه ۵ داریم:

| ۰ | ۱ | ۲ | ۳ | ۴ |
|---|---|---|---|---|
| محل قرار گیری داده اول | محل قرار گیری داده دوم | محل قرار گیری داده سوم | محل قرار گیری داده چهارم | محل قرار گیری داده پنجم |

- همان طور که از جدول بالا مشخص هست مانند رشته ها اولین خانه آرایه با شماره ۰ شروع تا ..... آخرین خانه آرایه با شماره (۱- اندازه آرایه) پایان می یابد.که در اینجا اندازه آرایه ۵ بود و خانه ها از ۰ تا ۴ شماره گذاری شده اند.
- به شماره های هر خانه آرایه اندیس های (indexs) آرایه می گویند مثلا اندیس ۲ ،مقدار خانه شماره ۲ آرایه را بر می گرداند.
- هر خانه از آرایه یک متغیر حساب می شود و همه خانه های آرایه از یک نوع یکسان می باشد،بر فرض مثال اگر این آرایه از نوع عدد صحیح باشد تمام عناصر (خانه های آرایه) از نوع عدد صحیح می باشند.

اصلا نگران نباشید آرایه شباهت زیادی با متغیر دارد و مانند متغیر از آن استفاده میکنیم فقط جمع و جورتر و کاربردی تر است!!!! ☺

روش تعریف آرایه:

۱) یک آرایه مانند متغیر هنگام تعریف نوع آن باید مشخص شود.

۲) در کنار نوع آرایه دو کروشه باز و بسته " [] " می کنیم

۳) بعد از تعریف نوع آرایه و قرار دادن دو کروشه باز و بسته ، مانند متغیر یک نام برایش انتخاب میکنیم.

۴) تا اینجا مراحل ۱ تا ۳ بصورت زیر می باشد:

```
Array_Type[]    Array_Name
```

۵) حالا یک مساوی قرار می دهیم:

```
Array_Type[]    Array_Name =
```

۶) سمت راست مساوی از کلمه کلیدی **new** استفاده می کنیم:

```
Array_Type[]    Array_Name = new
```

۷) بعد از کلمه new دوباره نوع آرایه با دو کروشه باز و بسته را پیاده سازی میکنیم، این نوع باید با نوع سمت چپ مساوی یکسان باشد:

```
Array_Type[]    Array_Name = new Array_Type[]
```

۸) حالا درون کروشه سمت راست اندازه (سایز) آرایه را مشخص می کنیم:

```
Array_Type[]    Array_Name =new Array_Type[Array_size];
```

مثال : در زیر یک آرایه از نوع عدد صحیح با نام number و اندازه ۵ تعریف شده است:

```
int[] number=new int[5];
```

این آرایه دارای ۵ خانه که اندیس های خانه (شماره خانه) آن از ۰ تا ۴ می باشد. و از نوع عدد صحیح می باشد.

این آرایه از نوع عدد صحیح معادل متغیر های زیر می باشد:

```
int number0;
int number1;
int number2;
int number3;
int number4;
```

بجای اینکه بیایم ۵ متغیر از نوع عدد صحیح تعریف کنیم و خود را به زحمت بیاندازیم در یک خط میایم یک ارایه از نوع عدد صحیح تعریف می کنیم که دارای ۵ خانه باشد.

حالا تفاوت روش مقدار دهی و دسرسی به خانه های آرایه و متغیر ها را در مثال بالا بررسی می کنیم:

فرض کنید قصد داریم به متغیر های number0,.....,number4 مقدار های ۱۰ تا ۱۴ را نسبت بدهیم که شکل مقدار دهی آنها بصورت زیر می باشد:

```
number0=10
number1=11
number2=12
number3=13
number4=14
```

حالا قصد داریم به خانه های آرایه number به سایز ۵ همین مقادیر ۱۰ تا ۱۴ را نسبت بدهیم:شکل صدا زدن خانه های آرایه و مقدار دهی به هر خانه به صورت زیر است:

```
number[0]= 10;
number[1]= 11;
number[2]= 12;
number[3]= 13;
number[4]= 14;
```

- پس روش صدا زدن و استفاده عناصر موجود در آرایه به صورت زیر است:

۱. نام آرایه
۲. کروشه باز و بسته " [] "
۳. قرار دادن اندیس یا شماره خانه مورد نظر
۴. حال می توان مثل متغیر باهاش رفتار کرد یعنی بهشون مقداری نسبت داد یا مقدارش رو دستکاری و چاپ و... کرد.

حال بعد از مقدار دهی آرایه number ساختار آن را بار دیگر بررسی و مشاهده می کنیم:

| number | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| | 10 | 11 | 12 | 13 | 14 |

اکنون قصد داریم خانه شماره ۳ آرایه number را به عدد ۷۹ تغییر بدهیم کافیست بصورت زیر عمل کنیم:

```
number[3]= 79;
```

تغییرات اعمال شده را بصورت زیر در ساختار آرایه number مشاهده می کنید:

| number | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| | 10 | 11 | 12 | 79 | 14 |

حال قصد داریم خانه شماره ۲ آرایه را چاپ کنیم:

```
System.out.println(number[2]);
```

خروجی:

```
12
```

- همان طور که مشاهده کردید از لحاظ مقدار دهی و شکل استفاده و ..... آرایه ها با متغیر ها فرقی ندارند تنها تفاوت در ساختار آرایه ها می باشد وگرنه آرایه خود مجموعه ای متوالی از متغیر هایی با نوع یکسان می باشد.البته یک تفاوت مهم آرایه با متغیرها دارد که جلوتر به آن می پردازیم.
- حالا دیدید آرایه چه کار بزرگی را برامون انجام میده! هم در تعداد خط هایی که کد میزنیم صرفه جویی میشه هم در انتخاب نام، شاید بگید این چکاریه!! اصلا دوست دارم از متغیر معمولی استفاده کنم !!!! اما فرض کنید در یک برنامه اگر نیاز به ۱۰۰۰ متغیر یا بیشتر از نوع عدد صحیح داشتیم اونقت چکار می کردید؟!!! پس حق بدید که آرایه خیلی مهم و کاربردی هست.
- حال روش تعریف آرایه را یادگرفتیم، اما این رو بدانید که روش های متفاوتی برای تعریف آرایه وجود دارد که انواع روش های ان بصورت زیر است که هیچ فرقی باهم نمی کنند: من رو آرایه از نوع عدد صحیح مثال زدم شما می توانید به جای نوع عدد صحیح هر نوعی بکار ببرید.

1)

```
int[] number=new int[5];
```

2)

```
int number[]=new int[5];
```

3)

```
int number[];
number=new int[5];
```

4)

```
int[] number;
number=new int[5];
```

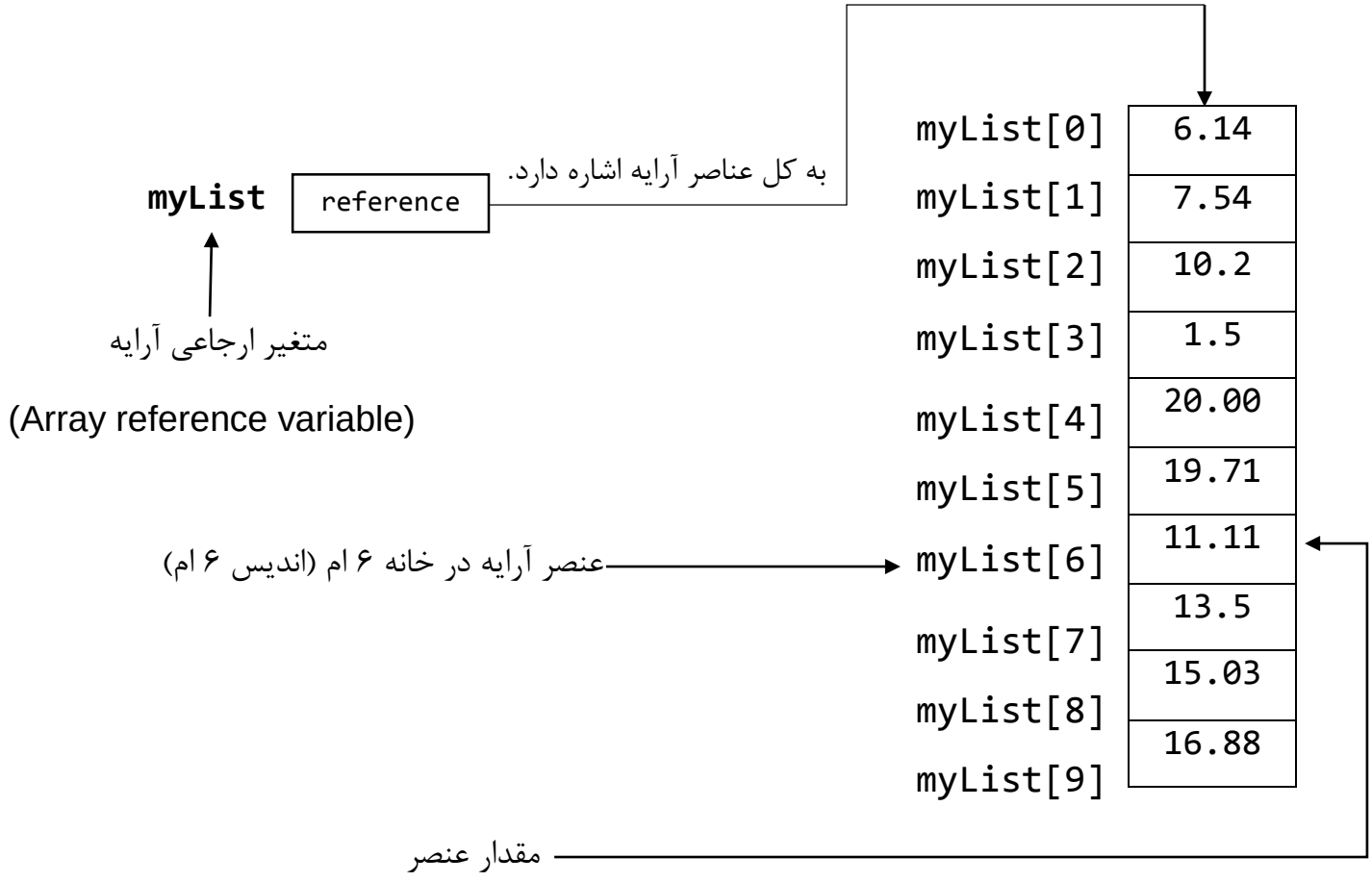
## تفاوت مهم آرایه با متغیر ها:

آرایه یک متغیر ارجاعی (reference variable) می باشد!!!!!!! خب متغیر ارجاعی یعنی چه؟؟؟؟؟؟؟؟؟!!!!!!!!

با مثال به این موضوع می پردازیم پس با ما همراه باشید:

فرض کنید یک لیستی از نمره های دانشجویان در اختیار استاد درس فلسفه می باشد .این لیست ۱۰ خانه دارد که شامل ۱۰ نمره از دانشجویان می باشد. خب ما در اینجا یک آرایه ۱۰ خانه ای از نوع عدد اعشاری داریم:

```
double[] myList = new double[10];
```




در نقشه بالا ما یک آرایه به نام myList که اندازه آن ۱۰ و از نوع عدد اعشاری می باشد، داریم.جدول سمت راست مقادیر عناصر درون آرایه را نشان می دهد. و هر کدام از myList[0] تا .....myList[9] اشاره به مقدار عنصر مورد نظر در خانه ۰ام ..... تا ۹ ام (اندیس ام) آرایه را دارد.خب از متغیر ارجاعی صحبت کردیم ، **myList** یعنی نام آرایه نقش متغیر ارجاعی را بازی می کند!!!خب این یعنی چه؟!! یعنی این متغیر دارنده کل مقادیر عناصر آرایه می

باشد.متغیر ارجاعی **myList** همان طور که در نقشه بالا مشاهده می کنید نه به یک خانه بلکه به کل خانه های آرایه اشاره می کند، از نگاه دیگر این متغیر ارجاعی حاوی آدرس کل آرایه در واحد حافظه می باشد.خب کاربردش چیه؟ فرض کنید یک متد به نام arrayShow داریم که به عنوان ورودی یک آرایه میگیرد و عناصر درون آرایه را چاپ میکند،حال چون آرایه یک متغیر ارجاعی هست نیاز نیست تک به تک ،اندیس به اندیس ،خانه به خانه!!! به آرایه دسرسی پیدا کنیم و به عنوان ورودی به متد بدیم که متد برامون این خانه ها رو چاپ کند،تنها کافیست نام متغیر ارجاعی که همان نام آرایه می باشد را به عنوان ورودی به متد بدهیم و متد با توجه به دستورات درون خود آن را چاپ میکند.

بعد از این که روش دسرسی به خانه های آرایه با یک حلقه for را بررسی کردیم یک مثال از متغیر ارجاعی خواهیم زد.

به مقدار موجود در هر یک ازخانه های آرایه ، عناصر آرایه می گویند.

## پروسه دسرسی به عناصر موجود در آرایه:

گاهی برای مقداردهی، چاپ و دستکاری و... آرایه نیاز هست به عناصر ارایه دسرسی پیدا کنیم.خب راهش چیه؟شاید بگید بصورت دستی یکی یکی به عناصر آرایه دسرسی پیدا می کنیم! اما اگر عناصر آرایه ۱۰۰۰ یا بیشتر بود چی؟! یکی از متداول ترین راه های دسرسی به عناصر آرایه استفاده از حلقه **for** می باشد.

### استفاده از حلقه for برای دسرسی به آرایه :

چرا برای دسرسی به عناصر آرایه از حلقه for استفاده می کنیم؟ زیرا عناصر یک آرایه نوع یکسانی دارند و اندازه آرایه را می دانیم.

قالب حلقه for را که یادتون میاد؟!!

یادآوری:شکل پیاده سازی حلقه for بصورت زیر است:

```
for(int i=0;i<size;i++){


}
```

- متغیر i شمارنده و size مقداری که شمارنده براساس آن تکرار می شد و i++ یکی به مقدار شمارنده می افزود و دستورات درون بدنه حلقه بین دو بلوک اجرا میشد. برای اطلاعات بیشتر به جلسه حلقه ها در همین کانال آموزشی مراجعه فرمایید.

برای دسرسی به خانه های ارایه از طریق حلقه for بصورت زیر عمل میکنیم:

۱) **بخش تعریف شمارنده** :نیاز به شمارنده ای مثل **i** داریم که خانه های آرایه را برای ما بشمارد به تعبیر دیگر این متغیر اندیس های آرایه را تشکیل می دهد و چون آرایه خانه اش از ۰ شروع می شود مقدار اولیه این متغیر برابر با صفر می باشد.

```
for(int i=0;;){

}
```

۲) **بخش مقایسه:**همان طور که می دانید اندازه آرایه ثابت و از قبل می دانیم،مثلا اگر آرایه خود را ۱۰ خانه تعریف کردید ،مقدار سایزی که شمارنده با ان مقایسه می شود را ۱۰ می گذاریم، اگر اندازه آرایه خود را نمی دانید می توانید به شکل زیر عمل کنید:

**نکته:**برای به دست آوردن اندازه آرایه از متد length استفاده می کنیم،چطور؟ نام آرایه + نقطه(.)+متد length :

```
int[] number=new int[5];
int size=number.length;
```

مقدار متغیر size برابر ۵ می باشد.

```
for(int i=0;i<number.length;){

}
```

۳) **بخش پلاس پلاس شدن شمارنده (یکی به مقدار شمارنده افزوده شدن):** در این بخش یکی به مقدار شمارنده افزوده و داخل شمارنده ریخته می شود i++ .

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

}
```

۴) **دستور درون بدنه حلقه (فرض کنید قصد چاپ عناصر درون آرایه را داریم):** در این قسمت نام آرایه با کروشه باز و بسته را درون دستور system.out.print() قرار می دهیم با این تفاوت که اندیس آرایه ، شمارنده ما یعنی i می باشد.

```
for(int i=0;i<number.length;i++){
    System.out.println(number[i]);
      }
```

متغیر i نقش اندیس (شماره) (index) خانه های آرایه را دارد، وقتی

i=0، number[0] و مقدار عنصر درون خانه ۰ ام را به ما می دهد.

i=1، number[1] و مقدار عنصر درون خانه ۱ ام را به ما می دهد.

i=2، number[2] و مقدار عنصر درون خانه ۲ ام را به ما می دهد.

i=3، number[3] و مقدار عنصر درون خانه ۳ ام را به ما می دهد.

i=4، number[4] و مقدار عنصر درون خانه ۴ ام را به ما می دهد.

شرط هم اینه، تا زمانی تکرار کن که i<5 می باشد ، چون number.length مقدار ۵ را برمی گرداند.

مثال: برنامه ای به زبان جاوا بنویسید که از ورودی ۵ عدد بگیرد و ودرون یک ارایه قرار دهد و سپس مقادیر درون ارایه را چاپ کند:

```
package javalike;
import java.util.Scanner;
//https://telegram.me/javalike
//author:Rahman Zarei

public class SampleString {

      public static void main(String[] args) {
            Scanner input = new Scanner(System.in);
            int[] number = new int[5];

            for (int i = 0; i < number.length; i++) {
                  System.out.println("please Enter the number");
                  number[i] = input.nextInt();
            }
            for (int i = 0; i < number.length; i++) {
                  System.out.print(number[i]+"  ");
```

```
            }

      }
}
```

**خروجی: فرض کنید اعداد ورودی ما بصورت تست زیر می باشد:**

```
please Enter the number
10
please Enter the number
265
please Enter the number
10
please Enter the number
47
please Enter the number
200
10  265  10  47  200  عناصر چاپ شده درون آرایه
```

- در صورت سوال گفته ۵ عدد از ورودی میگیریم و درون آرایه قرار می دهیم، پس مشخص هست که باید یک آرایه با سایز ۵ تعریف کنیم. چون نگفته چه عددی ما فرض کردیم که عددمون از نوع صحیح می باشد،پس نوع آرایه خود را int گذاشتیم.
- در این برنامه چون نیاز به ورودی داشتیم کلاس Scanner را import کردیم و ازش شی ساخته و به متد ان دسرسی پیدا کردیم و و در حلقه اول تک به تک به خانه درون آرایه دسرسی پیدا کرده و مقداری که از ورودی گرفتیم داخل این خانه ها ریختیم.
- در حلقه for دوم تک به تک به مقادیر عناصر درون ارایه دسرسی پیدا کرده و آنها را چاپ کردیم.
- برای درک بهتر دسرسی تک به تک به عناصر آرایه به کد زیر و خروجی ان توجه کنید:

```
package javalike;
import java.util.Scanner;
//https://telegram.me/javalike
//author:Rahman Zarei
public class SampleString {
      public static void main(String[] args) {
            Scanner input = new Scanner(System.in);
            int[] number = new int[5];
            for (int i = 0; i < number.length; i++) {
                  System.out.println("please Enter the number");
                  number[i] = input.nextInt();
            }
```

```
            for (int i = 0; i < number.length; i++) {
                  System.out.print("number["+i+"]="+number[i]+"   ");
            }
      }
}
```

**خروجی:**فرض کنید اعداد تست ما در هنگام ورودی دادن بصورت زیر باشد:

```
please Enter the number
56
please Enter the number
4112
please Enter the number
47
please Enter the number
20
please Enter the number
41
number[0]=56  number[1]=4112  number[2]=47  number[3]=20  number[4]=41
```

- در اینجا بصورت نمایشی مقدار هر عنصر از خانه i ام آرایه در خروجی نمایش داده شده است.

## مقداردهی مستقیم به ارایه:

ما می توانیم بعد از تعریف آرایه مستقیم آن را مقداردهی کنیم:

```
int[] number = {20,50,100,19};
```

- ❖ در این جا نه خبری از تعیین سایز آرایه و نه کلمه کلیدی new و.....
- ❖ در این روش دیگر نیاز به تعیین سایز و new کردن آرایه خود ندارید و آرایه شما آماده استفاده می باشد.

**مثال:**

```
package javalike;

import java.util.Scanner;

//https://telegram.me/javalike
//author:Rahman Zarei

public class SampleString {

      public static void main(String[] args) {
            Scanner input = new Scanner(System.in);
            int[] number = { 20, 50, 100, 19 };
```

```
            for (int i = 0; i < number.length; i++) {
                  System.out.print("number[" + i + "]=" + number[i] + "   ");
            }
      }
}
```

**خروجی:**

```
number[0]=20  number[1]=50  number[2]=100  number[3]=19
```

**مثال:**برنامه ای به زبان جاوا بنویسید شامل یک آرایه که عناصر خانه های آن به ترتیب مقادیر ۱۰و۲۰و۳۰و۴۰ باشد سپس مجموع عناصر این آرایه را پیدا و در خروجی چاپ کنید.

```
package javalike;

import java.util.Scanner;

//https://telegram.me/javalike
//author:Rahman Zarei

public class SampleString {

      public static void main(String[] args) {
            Scanner input = new Scanner(System.in);
            int[] number = { 10, 20, 30, 40 };
            int sum = 0;
            for (int i = 0; i < number.length; i++) {
                  sum += number[i];
            }
            System.out.print("sum=" + sum);

      }
}
```

**خروجی:**

```
sum=100
```

## وقتی پارامتر ورودی متد آرایه باشد:

چون ما هنوز در مورد جزییات متدها بطور مفصل اطلاعی نداریم و تنها بصورت سطحی یک بررسی هایی انجام دادیم اجازه بدید این بخش از آرایه را در جلسه ای که بطور کامل به متدها میپردازیم ، مورد بررسی قرار دهیم.

حالا منظور از آرایه یک بعدی چی بود؟!!! وقتی آرایه یک کروشه باز و بسته داشته باشد و تنها یک اندیسه باشد آرایه یک بعدی می باشد.

❖ به این کروشه باز و بسته "[]" اندیس آرایه می گویند.

```
int[] number=new int[5];
```

آرایه بیشتر از یک بعد هم داریم؟

پاسخ مثبت است،بله داریم هر چقدر به تعداد این کروشه باز و بسته اضاف شود به همون تعداد بعدی به آرایه می گویند.

ما بیشتر با آرایه یک بعدی و دو بعدی کار داریم!!! گفتید دو بعدی؟! یک مثال از دو بعدی برامون میزنی؟!!! بله:

```
int[][] number=new int[5][4];
```

این شکل پیاده سازی دو بعدی هست که بهش ماتریس هم می گوییم چیزی شبیه به جدول می باشد که ساختارش برای این مثال بصورت زیر است:

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

تصویر(۱)

این یک آرایه دو بعدی ۵ در ۴ (5*4) می باشد، یعنی ۵ عدد سطر و ۴ عدد ستون دارد.

سطر ها در این آرایه برای درک بهتر با رنگ آبی و ستون ها با رنگ قرمز مشخص کردم. اون رنگ مشکی هم بیخیالش بشید ☺

## آرایه دو بعدی در جاوا:

عناصر درون ارایه دو بعدی برخلاف آرایه یک بعدی در یک سطر قرار ندارند! در این نوع آرایه داده ها در سطر و ستون بر اساس ایندکس ( شماره خانه عنصری از آرایه) ذخیره می شوند.چیزی شبیه ماتریس یا جدول. تصویر(۱)

سینتکس یا نحوه نوشتن آرایه دو بعدی در جاوا بصورت های زیر می باشد:

1. dataType[][] arrayRefVar;

مثال:

```
public class Array2D {

	int[][] arrayRefVar;

}
```

2. dataType [][]arrayRefVar;

مثال:

```
public class Array2D {

	int [][]arrayRefVar;

}
```

3. dataType arrayRefVar[][];

مثال:

```
public class Array2D {

	int arrayRefVar[][];

}
```

4. dataType []arrayRefVar[];

مثال:

```
package swing_javalike;

public class Array2D {
```

```
    int []arrayRefVar[];
}
```

مانند آرایه یک بعدی ، برای استفاده از آرایه دو بعدی در جاوا باید از آن نمونه ایجاد کنیم.برای ساخت یک آرایه دو بعدی که تعداد سطر و ستون آن مشخص می باشد بصورت زیر عمل می کنیم ( برای مثال قصد داریم یک آرایه دو بعدی که تعداد سطر آن ۴ و تعداد ستون آن ۳ و از نوع عدد صحیح int می باشد ایجاد کنیم) :

۱. تعیین نوع آرایه، که در اینجا نوع آرایه را int انتخاب می کنیم:

```
public class Array2D {
   int
}
```

۲. تعیین نام برای آرایه

```
public class Array2D {
   int tabel
}
```

۳. در کنار نام آرایه علامت "[][]" می گذاریم.همان طور که میدانید برای آرایه یک بعدی تنها یک کروشه باز و بسته می کردیم "[]" . تعداد کروشه های باز و بسته مشخص کننده تعداد بعد آرایه می باشد که برای آرایه یک بعدی "[]" و برای آرایه دو بعدی "[][]" و برای آرایه سه بعدی "[][][]" و....

```
public class Array2D {
   int tabel[][]
}
```

۴.علامت "=" (مساوی) را بعد از "[][]" می گذاریم:

```
public class Array2D {
    int tabel[][]=
}
```

۴. سمت راست مساوی از کلمه کلیدی new استفاده می کنیم:

```
public class Array2D {
   int tabel[][]=new

}
```

۵. بعد از کلمه کلیدی new ، دوباره نوع آرایه را تعیین می کنیم:

```
public class Array2D {

   int tabel[][]=new  int

}
```

۶. در کنار نوع آرایه از دو کروشه باز و بسته "[][]" استفاده می کنیم.

```
public class Array2D {

   int tabel[][]=new  int[][]
}
```

۷. حالا تعداد سطر و ستون آرایه دوبعدی خود را مشخص می کنیم:

```
public class Array2D {

   int tabel[][]=new  int[4][3]

}
```

- [] اول تعداد سطر و [] دوم تعداد ستون آرایه دوبعدی ما را تشکیل می دهد.در این مثال آرایه دوبعدی ما از ۴ سطر و ۳ ستون تشکیل شده است.

۸. در پایان علامت ; را آخر آرایه تعریف شده در کنار "[4][3]" می گذاریم.☺

```
public class Array2D {

   int tabel[][] = new int[4][3];

}
```

خب اینم از روش تعریف آرایه دو بعدی در جاوا ☺

حالا قصد داریم با روش مقدار دهی به آرایه دو بعدی آشنا شویم:

فرض کنیم یک آرایه دو بعدی بصورت زیر در برنامه خود تعریف کرده ایم:

```
public class Array2D {

     int table[][] = new int[5][4];

public Array2D(){

}

}
```

برای درک بهتر این آرایه دو بعدی به شکل تصویر(۲) توجه کنید:

**table**

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

تصویر(۲)

طبق تصویر(۲) آرایه دو بعدی ما با نام table دارای ۵ سطر و ۴ ستون می باشد. ( تصویر۲ تنها برای درک نحوه قرار گیری عناصر درون ارایه دو بعدی می باشد)

حالا قصد داریم این ارایه را مقداردهی کنیم.برای این کار بصورت زیر عمل می کنیم: قصد داریم در سازنده کلاس آرایه خود را مقداردهی کنیم.

۱. ابتدا نام آرایه را صدا می زنیم:

```
public class Array2D {

   int table[][] = new int[5][4];
```

```
      public Array2D(){
   table
}
}
```

۲. بعد از نام آرایه باید تعیین کنیم کدام خانه از آرایه را میخواهیم مقداردهی کنیم، آرایه دوبعدی نیز مثل آرایه یک بعدی از تعدادی از خانه تشکیل شده است.راه دسرسی به خانه های آرایه دوبعدی از طریق ایندکس یا شماره خانه های آن می باشد. ایندکس یک خانه آرایه دوبعدی برابر شماره سطر و شماره ستون آن خانه در آرایه دوبعدی می باشد.مثلا ما قصد داریم خانه 1*2 آرایه دوبعدی را مقدار دهی اولیه کنیم. شماره سطر این آرایه برابر ۲ و شماره ستون این آرایه برابر ۱ می باشد.برای این کار بصورت زیر عمل می کنیم:

```
public class Array2D {

   int table[][] = new int[5][4];

public Array2D(){

   table[2][1]
}

}
```

در اینجا دو کروشه باز و بسته "[][]" کنار نام آرایه گذاشتیم ، کروشه اول اشاره به شماره سطر که در اینجا ۲ و کروشه دوم اشاره به شماره ستون خانه آرایه که در اینجا ۱ است دارد این شماره سطر و ستون همان ایندکس خانه مورد نظر در آرایه می باشد.

۳. حال بعد از "[2][1]" علامت مساوی گذاشته و مقداری عدد ۴ را به خانه مورد نظر در آرایه نسبت می دهیم:

```
package swing_javalike;

public class Array2D {

      int table[][] = new int[5][4];

      public Array2D() {
            table[2][1] = 4;

      }

}
```

با این مقدار دهی عدد ۴ در خانه از آرایه که شماره سطر آن ۲ و شماره ستون آن ۱ است قرار میگیرد. تصویر(۳)

**table**

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |
| 2 | | 4 | | |
| 3 | | | | |
| 4 | | | | |

تصویر(۳)

خب حالا که گام به گام با مقداردهی به خانه های آرایه دو بعدی آشنا شدید برای درک بهتر به مثال زیر توجه کنید:

```
package swing_javalike;

public class Array2D {

	int arr[][] = new int[3][3];

	public Array2D() {
		arr[0][0] = 1;
		arr[0][1] = 2;
		arr[0][2] = 3;
		arr[1][0] = 4;
		arr[1][1] = 5;
		arr[1][2] = 6;
		arr[2][0] = 7;
		arr[2][1] = 8;
		arr[2][2] = 9;

	}

}
```

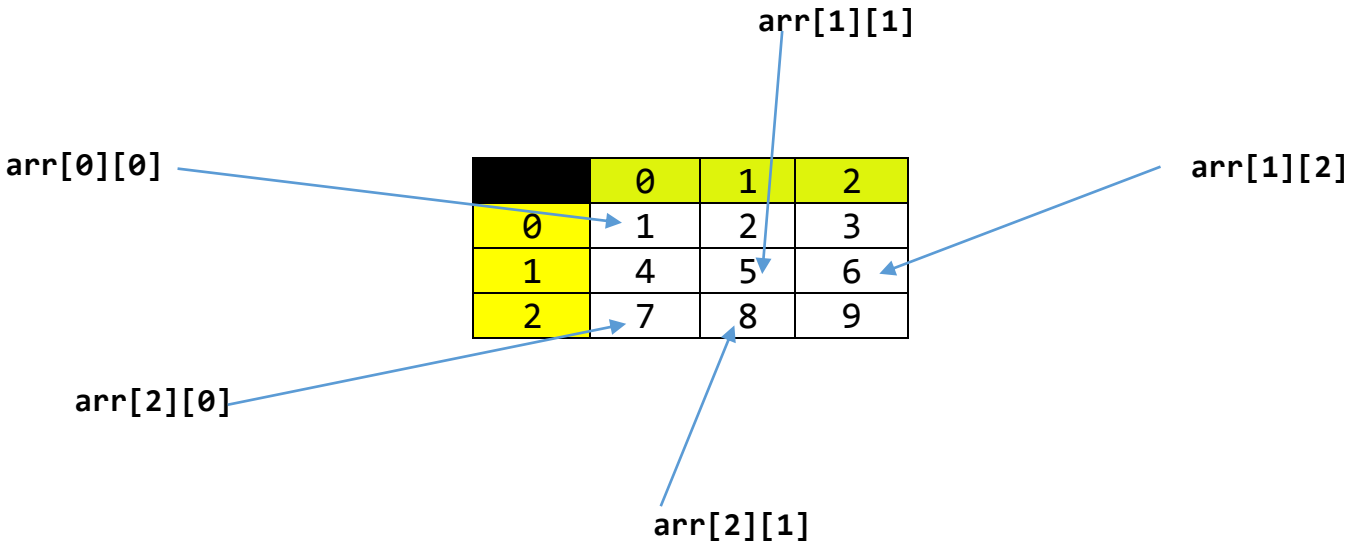
در این مثال یک آرایه دو بعدی 3*3 تعریف کرده ایم و درون سازنده کلاس ، خانه های این آرایه را مقدار دهی اولیه کرده ایم.که این مقادیر بصورت شکل موجود در تصویر(۴) در خانه های ارایه دوبعدی ما قرار می گیرند:

**arr**

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 1 | 4 | 5 | 6 |
| 2 | 7 | 8 | 9 |

تصویر(۴)

- برای درک بهتر دسرسی به خانه های آرایه به شکل زیر توجه کنید




امیدوارم تا اینجا توانسته باشم مفهوم و روشه پیاده سازی آرایه های دوبعدی را برای شما روشن کنم ☺

مثال:در مثال زیر با روش های تعریف و ایجاد ،مقدار دهی اولیه و چاپ عناصر درون آرایه دوبعدی آشنا می شویم:

```
class Testarray3{
public static void main(String args[]){

		// declaring and initializing 2D array
		int arr[][] = { { 1, 2, 3 }, { 2, 4, 5 }, { 4, 4, 5 } };

		// printing 2D array
		for (int i = 0; i < 3; i++) {
			for (int j = 0; j < 3; j++) {
				System.out.print(arr[i][j] + " ");
			}
			System.out.println();
		}

	}
}
```

خروجی:

```
1 2 3
2 4 5
4 4 5
```

- چاپ عناصر درون یک ارایه دوبعدی بصورت یک ماتریس یا جدول شکل می باشد برای درک بهتر سطر ها و ستون هیا آرایه دوبعدی

```
int arr[][] = { { 1, 2, 3 }, { 2, 4, 5 }, { 4, 4, 5 } };
```

- تعریف و مقداردهی اولیه ارایه دو بعدی

```
	for (int i = 0; i < 3; i++) {
			for (int j = 0; j < 3; j++) {
				System.out.print(arr[i][j] + " ");
			}
			System.out.println();
		}

	}
```

- برای چاپ عناصر درون یک آرایه دوبعدی از دو حلقه تو در تو استفاده می کنیم.

برای چاپ عناصر یک ارایه دوبعدی می توانیم بصورت زیر نیز عمل کنیم:

```
package swing_javalike;
class Testarray3{
```

```
public static void main(String args[]){

		// declaring and initializing 2D array
		int arr[][] = { { 1, 2, 3 }, { 2, 4, 5 }, { 4, 4, 5 } };

		// printing 2D array
		for (int i = 0; i < arr.length; i++) {
			for (int j = 0; j < arr[i].length; j++) {
				System.out.print(arr[i][j] + " ");
			}
			System.out.println();
		}

	}
}
```

لطفا نظرتون رو درمورد آموزش آرایه دوبعدی برای ما ارسال کنید .

پیروز و موفق باشید

---

سایت آموزش زبان جاوا به زبان ساده،آسان و شیرین!!!

www.JAVAPro.ir

آموزش جاوا SE را با تجربه شخصی و به زبان خودمونی یاد بگیرید!!!!!

# بازدید ازکانال

# بازدید از سایت

**هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید.**