آموزش مقدماتی

زبان جاوا

(به زبانی ساده و مصور)

نویسنده: بهنام خانی

WWW.CODE-ACADEMY.IR

به نام خدا

در این کتاب آموزشی شما با مقدمات جاوا به زبانی ساده آشنا می شوید. این کتاب پیشنیاز شماست برای یادگیری برنامه نویسی اندروید. دانلود کنید و با تمرین این پیشنیاز را با موفقیت فرابگیرید. در ضمن کسانی که تصمیم دارند زبان برنامه نویسی جاوا را به منظور برنامه نویسی برنامه های دسکتاپ و ... فرا بگیرند هم می تواند شروع بسیار خوبی باشد.

چه ابزارهایی برای استفاده از این کتاب لازم است؟

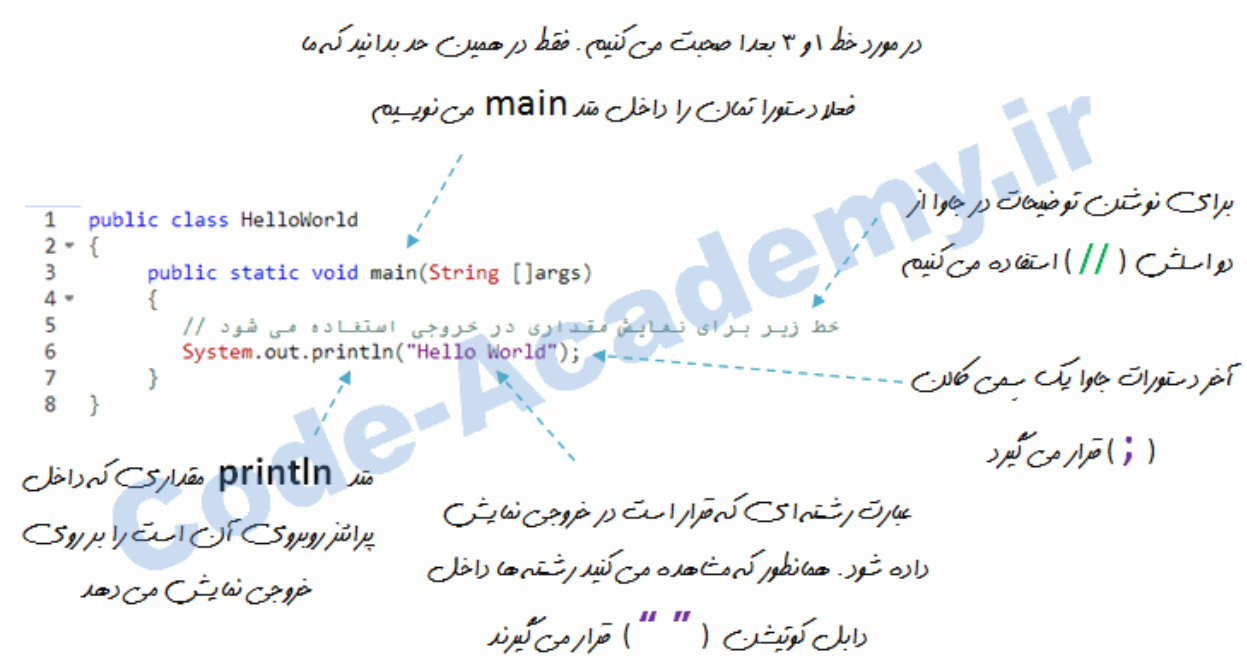
شما با دانلود و نصب اکلیپس و JDK می توانید شروع به یادگیری کنید. ولی با این حال می توانید تمرین ها و آموزش های این قسمت را با استفاده از وب سایت هایی که کامپایلرهای آنلاین جاوا را به صورت رایگان دراختیار کاربران قرار داده اند استفاده کنید. در زیر آدرس یکی از این وب سایت ها آماده است. کافی است شما کد مورد نظرتان را در ویرایشگر متن این وب سایت ها بنویسید و آنرا اجرا و نتیجه آنرا مشاهده کنید .**با این وجود به شما توصیه می کنم که از محیط های توسعه مثل اکلیپس استفاده کنید.**

کامپایلر آنلاین جاوا

برای استفاده از این کامپایلر، پس از وارد کردن کد مورد نظرتان، ابتدا دکمه Compile و بعد از آن دکمه Execute را کلیک کنید.

آشنایی با ساختار جاوا

برای اینکه با ساختار زبان جاوا آشنا شوید کار خود را با یک مثال شروع می کنیم. در زیر برنامه ساده ای را مشاهده می کنید که عبارت Hello World را در خروجی نمایش می دهد.



ممکن است مفاهیم و کدهای بالا برای شما نا مفهوم باشد. نگران نباشید و بر روی خط ششم مثال دقت کنید. در این خط ما با استفاده از متد println عبارتی را بر روی خروجی نمایش می دهیم. خط پنجم این برنامه هم یک خط توضیح است. توضیحات قابلیت اجرایی ندارند و فقط به عنوان کمک کننده به برنامه نویس در مورد نحوه عملکرد برنامه استفاده می شوند.

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید. این برنامه عبارت Hello World را در خروجی نمایش می دهد:

```
public class HelloWorld
{
public static void main(String []args)
{
// خط زیر برای نمایش مقداری بر روی خروجی استفاده می شود
System.out.println("Hello World");
}
}
```

خروجی این برنامه به صورت زیر می باشد:

```
Hello World
```

در ضمن شما می توانید کدهای تمرینی خودتان را نیز در همین قسمت وارد نمایید. توصیه من به کسانی که به تازگی شروع به برنامه نویسی کرده اند این است که تا جای ممکن کد ها را مطالعه و آنها را اجرا کنند. همینطور تمریناتی که در ادامه هر آموزش آمده است را انجام داده و نتیجه آنرا مشاهده نمایند.

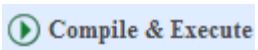
تمرین ۱: برای درک بیشتر مفهوم یک تمرین برای شما در نظر گرفته ام، لطفا بر اساس مراحل آنرا انجام دهید:

۱- کد زیر را کپی کنید

```
public class HelloWorld
{
public static void main(String []args)
{
        //  کد تمرین را در خط زیر اضافه کنید

}
}
```

- بر روی دکمه اجرای کد برنامه کلیک کنید و در پنجره باز شده، کد بالا را جایگزین کد موجود در پنل سمت چپ کنید

۳- در زیر خط توضیح کدی اضافه کنید که عبارت Hello Java را در خروجی نمایش دهد

۴- برنامه را با استفاده از دکمه 

اجرا نمایید.

خروجی این برنامه به صورت زیر می باشد:

```
Hello Java
```

نکته: معمولا به این مورد زیاد برخورده ام که کارآموزان برای دستیابی به جواب به صورت شتابزده عمل می کنند. لطفا در صورتیکه در دفعات اول موفق به اجرای تمرین نشدید از تلاش دست بر نداشته

و باز هم امتحان کنید. یکی از رمزهای مهم در اینکه شما بتوانید برای اندروید برنامه بنویسید همین است، تلاش !!! به هر حال جواب تمرین در زیر آمده است

جواب تمرین

```
public class HelloWorld
{
      public static void main(String []args)
      {
         // کد تمرین را در خط زیر اضافه کنید
         System.out.println("Hello Java");
      }
}
```

اعتقاد من این است که برنامه نویس باید با اصطلاحات تخصصی برنامه نویسی آشنا و کم کم بر روی آن مسلط شود. بنابراین چه در این دوره آموزشی و چه در دوره اندروید من اصطلاحات هر بخش با معنی آن را در اختیار کارآموزان قرار داده ام.

Structure: ساختار

Output: خروجی

Slash: نشان ممیز

Semicolon: ; ویرگول بدین شکل

Quote: نقل قول

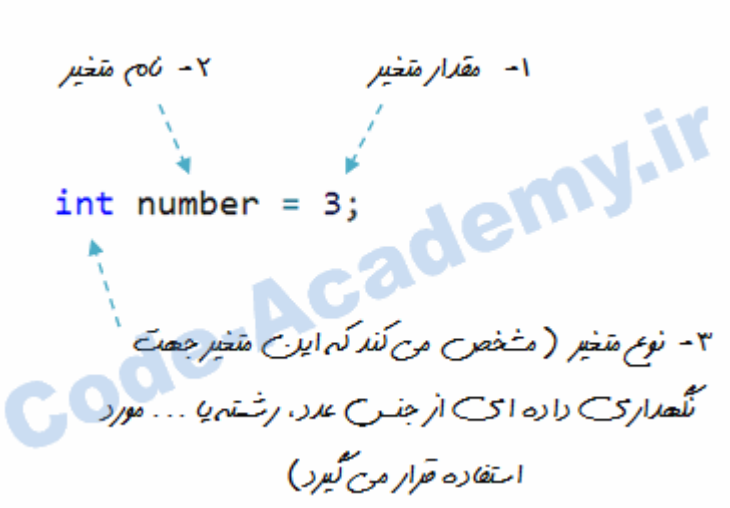
Method: روش، رویه

Compile: ترجمه یک برنامه به زبان ماشین

Execute: اجرای برنامه

متغیر چیست؟

برنامه ها برای نگهداری داده ها از متغیرها استفاده می کنند. متغیر مکانی در حافظه است که برای نگهداری یک مقدار مورد استفاده قرار میگیرد. هر متغیر از سه قسمت تشکیل شده است.



برای استفاده از متغیرها بایستی آنها را در برنامه تعریف کنیم. در شکل بالا متغیری از نوع int (نوعی که اعداد صحیح – بدون ممیز – را نگهداری می کند) تعریف کرده، نام آنرا number گذاشته و مقدار ۳ را در آن قرار داده ایم.

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

```
public class VariableSample
  {
    public static void main(String[] args)
    {
      int number = 3;
      System.out.println(number);
    }
  }
```

همینطور که مشاهده می کنید در متد main متغیری از نوع int تعریف کرده، نام آنرا number گذاشته و مقدار ۳ را در آن قرار داده ایم. سپس در خط بعدی مقدار این متغیر را در خروجی نمایش می دهیم. خروجی این برنامه به صورت زیر می باشد:

```
3
```

مثال ۲

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

```
public class VariableSample
  {
    public static void main(String[] args)
    {
      int age = 30;
      System.out.println("My age is: " + age);
    }
  }
```

همینطور که مشاهده می کنید در متد main متغیری از نوع int تعریف کرده، نام آنرا age گذاشته و مقدار ۳۰ را در آن قرار داده ایم. سپس در خط بعدی مقدار این متغیر را در خروجی نمایش می دهیم. خروجی این برنامه به صورت زیر می باشد:

```
My age is: 30
```

در مثال بالا رشته My age is: به مقداری که داخل متغیر age می باشد چسبانده شده و در خروجی نمایش داده می شود.

تمرین ۱ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که :

۱- متغیری از جنس int تعریف ، نام آنرا page گذاشته و مقدار ۱۰۱۳ را در آن قرار دهید

۲-از متغیر بالا به گونه ای استفاده کنید که خروجی برنامه شما به صورت زیر باشد

```
Page 1013 is about Android
```

جواب تمرین

```
public class VariableSample
  {
    public static void main(String[] args)
    {
      int page = 1013;
      System.out.println("Page " + page + " is about
Android");
    }
  }
```

Variable: متغیر

Data: داده

Integer: عدد صحیح

Declare: تعریف کردن --> Variable declarae: تعریف متغیر

Initialize: مقداردهی اولیه --> Variable initialize: مقدار دهی اولیه متغیر

String: رشته

انواع داده عددی

انواع داده را می توان مثل ظروفی با اندازه های مختلف در نظر بگیرید. یک ظرف ممکن است بتواند یک لیتر آب را در خود نگهداری کند، ظرفی دیگر دو لیتر و یا اینکه ظرف دیگری قابلیت نگهداری چهار لیتر آب را داشته باشد.

در جاوا نیز نوع ها ظرفیت های مختلفی دارند. در جدول زیر نوع های عددی (که فقط می توان اعداد را در آنها ذخیره کرد) با ظرفیت آنها آورده شده است:

نوع داده	ظرفیت	مثال
byte	۱ بایت یعنی از -128 تا 127	byte b1 = 43;
short	۲ بایت یعنی از -32,768 تا 32,767	short s1 = 1500;
int	٤ بایت یعنی از - 2,147,483,648 تا 2,147,483,647	int i1 = 1500000;
long	۸ بایت یعنی از -9,223,372,036,854,775,808 تا 9,223,372,036,854,775,807	long l1 = 4000000000;
float	٤ بایت برای نگهداری اعداد اعشاری	float f1 = 15.2f;
double	۸ بایت برای نگهداری اعداد اعشاری	double d1 = 15.2;

چند نکته را در مورد این جدول به خاطر بسپارید:

۱- برای اعدادی که از سه رقم بیشتر هستند جداکننده نگذارید!

اشتباه است a = 10,000;

۲- از نوع float و double برای نگهداری اعداد اعشاری استفاده می کنیم.

۳- جاوا به صورت پیش فرض اعداد اعشاری را از نوع double در نظر می گیرد. ما برای اینکه به جاوا اعلام کنیم می خواهیم عدد اعشاری را از نوع float نگهداری کنیم بایستی نوع double را به نوع float تبدیل کنیم، به این دو روش:

روش اول: در جلوی مقدار، حرف f را قرار دهید.

```
float f = 4.5f;
```

روش دوم: تبدیل نوع double به float به صورت زیر می باشد:

```
float f = (float) 4.5;
```

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

```
public class VariableSample
  {
    public static void main(String[] args)
    {
        // Initialize variables
      byte b1 = 43;
      short  s1 = 1500;
      int  i1 = 1500000;
      long  l1 = 400000000;
      float  f1 = 15.2f;
      double  d1 = 15.2;

      // Print variables in output
      System.out.println("b1 is: " + b1);
      System.out.println("s1 is: " + s1);
      System.out.println("i1 is: " + i1);
      System.out.println("l1 is: " + l1);
      System.out.println("f1 is: " + f1);
      System.out.println("d1 is: " + d1);
    }
  }
```

خروجی این برنامه به صورت زیر می باشد:

```
b1 is: 43
s1 is: 1500
i1 is: 1500000
l1 is: 400000000
```

```
f1 is: 15.2
d1 is: 15.2
```

تمرین ۱ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که :

۱- متغیری از جنس float تعریف، نام آنرا f1 گذاشته و به روش اول مقدار ۱۲,۲۵ را در آن قرار دهید و سپس آنرا نمایش دهید.
۲- متغیری از جنس float تعریف، نام آنرا f2 گذاشته و به روش دوم مقدار ۱۲,۲۵ را در آن قرار دهید و سپس آنرا نمایش دهید.

خروجی برنامه بایستی به صورت زیر باشد

```
f1 is: 12.25
f2 is: 12.25
```

جواب تمرین ۱

```
public class VariableSample
  {
    public static void main(String[] args)
    {
        // Initialize variables
      float f1 = 12.25f;
      float f2 = (float) 12.25;

      // Print variables in output
      System.out.println("f1 is: " + f1);
      System.out.println("f2 is: " + f2);

    }
     }
```

Data Type: نوع داده

Cast: تبدیل نوع

Seperator: جدا کننده

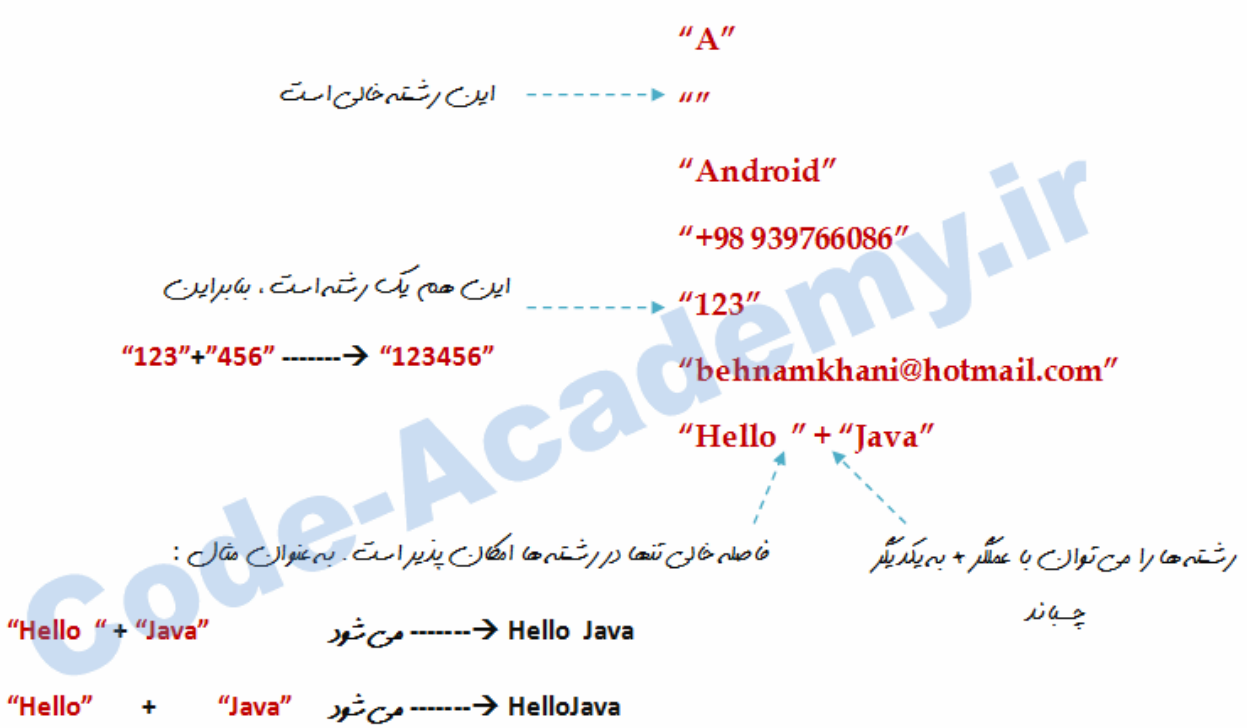
Storage Capacity: ظرفیت نگهداری

Range: بازه، محدوده

نوع داده رشته ای

در جاوا و در برنامه نویسی اندروید از رشته ها به صورت گسترده استفاده می شود. رشته در واقع مجموعه ای کارکترها است. رشته برای نگهداری متن به کار برده می شود. مانند رشته ای برای نگهداری ایمیل، رشته ای برای نگهداری نام کاربر و...

رشته در داخل علامت نقل قول یا کوتیشن قرار می گیرد. در زیر همگی نوعی رشته هستند:



چند نکته را در مورد رشته ها به خاطر بسپارید:

۱- نحوه تعریف رشته ها به صورت زیر می باشد. در نظر داشته باشید که بایستی String را با حرف S بزرگ تایپ کرد. چراکه جاوا یک زبان حساس به کوچک و بزرگ بودن حروف است:

```
String a = "android";
```

۲- رشته ها دارای طول هستند که در واقع طول آنها همان تعداد کارکترهای موجود در آن رشته است. برای بدست آوردن طول یک رشته از متد length() استفاده می کنیم:

```
int len = a.length();     // مقدار ۷ در متغیرذخیره می شود
```

در مورد اینکه چرا بعد از نام متغیر از length() استفاده کردیم در بخش شی گرایی صحبت خواهیم کرد.

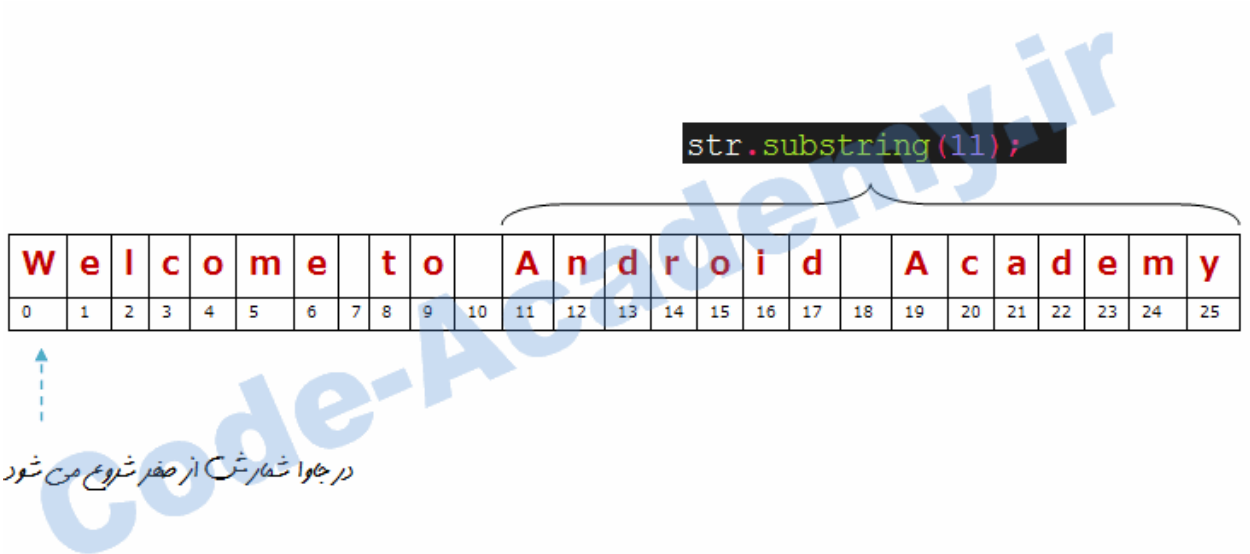
۳- برای بدست آوردن یک زیر رشته از درون یک رشته می توان از متد substring() استفاده کرد. این متد به دو روش عمل می کند:

روش اول: در روش اول این متد یک پارامتر می گیرد. این پارامتر در واقع یک عدد است. با این پارامتر به متد substring() می گوییم که از کارکتر n ام تا آخر رشته را به عنوان زیر رشته برای ما برگرداند. به این تکه کد دقت کنید:

```
String str = "Welcome to Android Academy";
String subStr = str.substring(11);
```

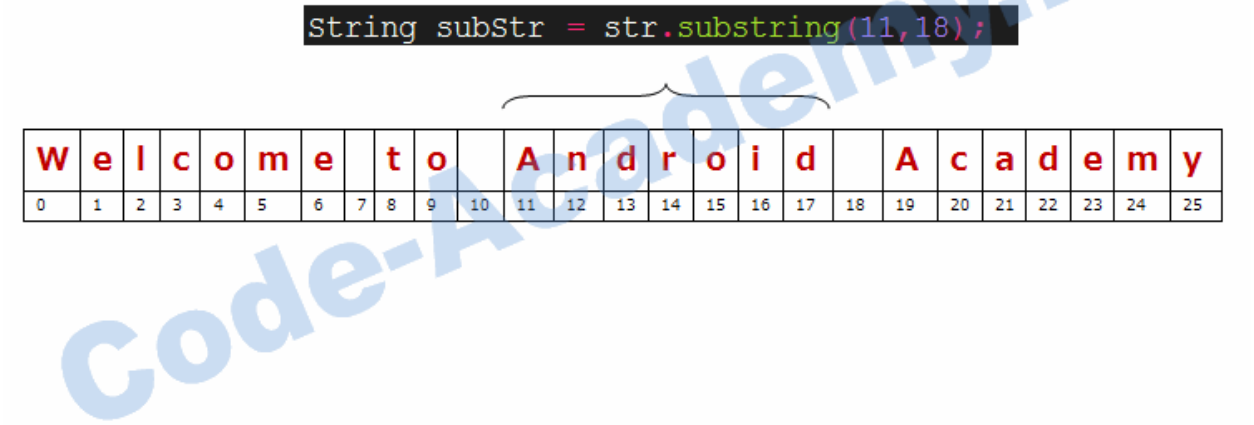
در خط اول این تکه کد متغیری به نام str تعریف کرده و مقدار " Welcome to Android Academy" را در آن قرار می دهیم

در خط دوم این تکه کد متغیری به نام subStr تعریف کرده و متد substring() را با یک پارامتر صدا می زنیم. وقتی این پارامتر را – که همان عدد ۱۱ است – به این متد می دهیم، متوجه می شود که باید از خانه دوازدهم تا آخر رشته را به عنوان نتیجه برگرداند:



بنابراین مقدار Android Academy در متغیر subStr قرار می یگرد.

روش دوم: در روش دوم این متد دو پارامتر می گیرد. پارامتر اول بیانگر این است که از اندیس nام شروع کن و تا اندیس jام که همان پارامتر دوم است را نتخاب کن :




```
String subStr = str.substring(11,18);
```

در کد بالا مقدار Android در متغیر subStr قرار می گیرد.

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید. این برنامه رشته ای تعریف کرده، مقدار Hello Java را در آن قرار می دهد. سپس طول این رشته را در متغیر دیگری ریخته و آنرا در خروجی نمایش می دهد:

```
public class HelloWorld
{
     public static void main(String []args)
     {
        String a = "Hello Java";
        int len = a.length();    //  مقدار عددی ۱۰ در متغیر
ذخیره می شود
        System.out.println("length is " + len);
     }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
length is 10
```

مثال ۲

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید. این برنامه متغیر رشته ای به نام str **تعریف و مقدار** Welcome to Android Academy **را در آن قرار می دهد. سپس با استفاده از متد** substring() **و با دو روش زیر رشته هایی از آن استخراج می کند:**

```
public class HelloWorld
{
     public static void main(String []args)
     {
        String str = "Welcome to Android Academy";
        String subStr1 = str.substring(11);
        String subStr2 = str.substring(11,18);

        System.out.println("subStr1 is " + subStr1);
        System.out.println("subStr2 is " + subStr2);
     }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
subStr1 is Android Academy
subStr2 is Android
```

تمرین ۱ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که :

۱- متغیری به نام str تعریف و مقدار Welcome to Android Academyرا به آن اختصاص دهید.

۲- چهار متغیر به نام sub1,sub2,sub3,sub4 تعریف و هر کدام از کلمات جمله متغیر str را با استفاده از متد substring() بدست آورید و در این چهار متغیر قرار دهید. سپس در خروجی تمام چهار متغیر را نمایش دهید.

خروجی برنامه بایستی به صورت زیر باشد

```
sub1 is: Welcome
sub2 is: to
sub3 is: Android
sub4 is: Academy
```

نکته: معمولا به این مورد زیاد برخورده ام که کارآموزان برای دستیابی به جواب به صورت شتابزده عمل می کنند. لطفا در صورتیکه در دفعات اول موفق به اجرای تمرین نشدید از تلاش دست بر نداشته و باز هم امتحان کنید. یکی از رمزهای مهم در اینکه شما بتوانید برای اندروید برنامه بنویسید همین است، تلاش !!! به هر حال جواب تمرین در زیر آمده است

جواب تمرین ۱

```
public class HelloWorld
{
     public static void main(String []args)
     {
        String str = "Welcome to Android Academy";
        String sub1 = str.substring(0,7);
        String sub2 = str.substring(8,10);
        String sub3 = str.substring(11,18);
        String sub4 = str.substring(19);

        System.out.println("sub1 is: " + sub1);
        System.out.println("sub2 is: " + sub2);
        System.out.println("sub3 is: " + sub3);
        System.out.println("sub4 is: " + sub4);
     }
}
```

Character: کارکتر

String concatenation: چسباندن رشته

Empty: خالی

Case sensitive: حساس به حروف کوچک و بزرگ

Sub string: زیر رشته

Index: شاخص، خانه

Return: برگرداندن

Parameter: پارامتر، مقدار معلوم و مشخص

نوع داده بولین

این نوع داده فقط می تواند مقدار true یا مقدار false را نگهداری کند. از این نوع داده بیشتر در دستورات کنترلی که در بخش بعدی در مورد آن صحبت کرده ام استفاده می شود.مقدار پیش فرض این نوع داده false است.

```
boolean isDay = true;
boolean isNight = false;
```

در این نوع داده، ما متغیرها را به صورت سوالی نامگذاری می کنیم. مثلا در بالا متغیر isDay بیانگر این سوال است که آیا الان روز است؟ و ما مقدار بله را در آن قرار داده ایم.

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

```
public class HelloWorld
{
     public static void main(String []args)
     {
        boolean isDay = true;
        boolean isNight = false;

        System.out.println("Is day right now? "+ isDay );
        System.out.println("Is night right now? "+ isNight);
     }
}
```

خروجی این برنامه به صورت زیر می باشد:

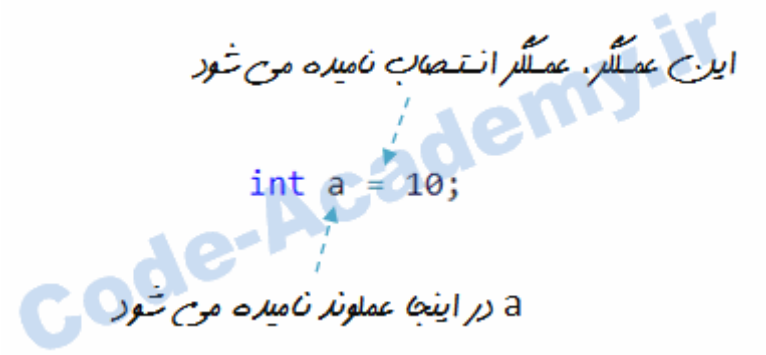
```
Is day right now? true
 Is night right now? false
```

عملگرها

در این قسمت تصمیم دارم شما را با انواع عملگرها در زبان جاوا آشنا کنم. هر کدام از این عملگرها بر روی یک، دو یا سه عملوند تاثیر گذاشته و نتیجه ای را بر می گردانند.

عملگر انتصاب

از این عملگر جهت مقدار دهی متغیرها استفاده می شود



عملگرهای ریاضی

+عملگر جمع
-عملگر تفریق
*عملگر ضرب
/عملگر تقسیم
%عملگر باقیمانده

عملگرهای تساوی و رابطه ای

==عملگر مساوی
=!عملگر نامساوی
<عملگر بزرگتر
=<عملگر بزرگتر یا برابر با
><عملگر کوچکتر
=><عملگر کوچکتر یا برابر با

عملگرهای شرطی

AND &&عملگر
OR ||عملگر

عملگرهای یگانی

++عملگر افزایش
--عملگر کاهش

در قسمت های قبلی از عملگر انتصاب بارها استفاده کردیم. در مثال زیر تصمیم دارم از عملگرهای ریاضی استفاده کنیم. عملگرهای تساوی و شرطی را نیز در قسمت بعدی استفاده خواهیم کرد.

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

```
public class HelloWorld
{
     public static void main(String []args)
     {
          // تعریف متغیرها و انتصاب مقادیر
        int a = 5;
        int b = 3;

        // انجام عملیات ریاضی با عملگرها ریاضی
        int sum = a + b + 2;
        int minus = a - b;
        int multiply = a * b;
        int division =  a / b;
        int remainder = a % b;

        // نمایش نتیجه
        System.out.println("a + b = " + sum);
        System.out.println("a - b = " + minus);
        System.out.println("a * b = " + multiply);
        System.out.println("a / b = " + division);
        System.out.println("a % b = " + remainder);
     }
}
```

خروجی برنامه بالا به صورت زیر خواهد بود:

```
a + b = 10
a - b = 2
a * b = 15
a / b = 1
```

```
a % b = 2
```

تمرین ۱ : برنامه مثال یک را در اکلیپس یا کامپایلر آنلاین به گونه ای تغییر دهید که:

نتیجه تقسیم ۵ بر ۳ را به صورت اعشاری نمایش دهد

خروجی شما بایستی به صورت زیر باشد:

```
a + b = 10
a - b = 2
a * b = 15
a / b = 1.6666666
a % b = 2
```

Operator: عملگر

Operand: عملوند

Assign: اختصاص دادن

Arithmetic Operators: عملگرهای ریاضی

Assignment Operator: عملگر انتصاب

Equality and Relational Operators: عملگرهای تساوی و رابطه ای

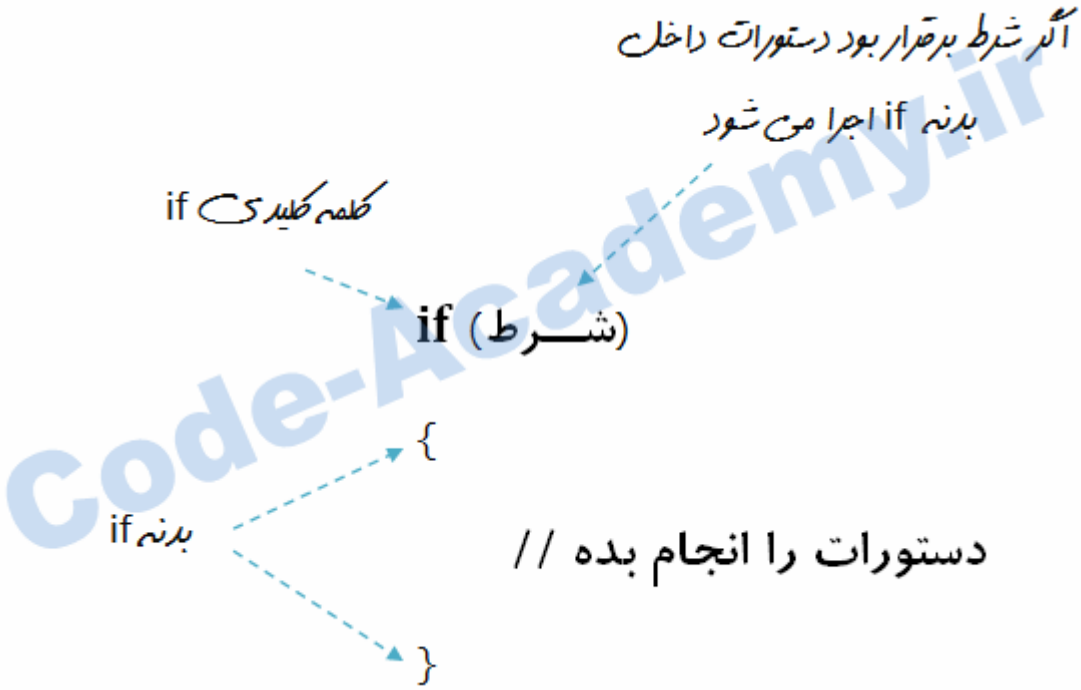
Conditional Operators: عملگرهای شرطی

Unary Operators: عملگرهای یگانی

دستور کنترلی IF ،IF...ELSE

دستور if یکی از دستورات پایه ای در زبان جاوا می باشد.تا الان برنامه هایی که شما نوشته اید پشت سر هم و به صورت خطی اجرا می شد. با استفاده از این دستور می توان حالت های مختلفی را در برنامه بررسی و نسبت به هر کدام از این حالت ها دستور مناسبی را اجرا نمود.

شکل کلی استفاده از این دستور به صورت زیر می باشد:



مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

```
public class HelloWorld
{
     public static void main(String []args)
     {
       int userAge = 18;

       if (userAge <= 18)
       {
           System.out.println("User age is 18 or younger");
       }
       if (userAge > 18)
       {
           System.out.println("User age is older than 18");
       }
     }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
User age is 18 or younger
```

همینطور که مشاهده می کنید شما دو حالت برای سن کاربر با استفاده از دو دستور if در نظر گرفتید. در ادامه به شما در مورد دستور if...else خواهم گفت. با این دستور شما می توانید کد بالا را به شکل بهتری بنویسید:

```
if (شـــرط)
{
    دستورات را انجام بده //
}
else
{
    در غیر اینصورت این دستورات را انجام بده//
}
```

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید. در این مثال تصمیم به استفاده از if...else را دارم.

```
public class HelloWorld
{
     public static void main(String []args)
     {
       int userAge = 18;

       if (userAge <= 18)
       {
           System.out.println("User age is 18 or younger");
       }
       else
       {
           System.out.println("User age is older than 18");
       }
     }
                                                                    }
```

خروجی این برنامه به صورت زیر می باشد:

```
User age is 18 or younger
```

در مثال ۲ ما از دستور if به همراه else استفاده کردیم.

مثال ۳

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید. در این مثال حالت های مختلفی که شرط برقرار می شود را مشاهده می کنید

```
public class HelloWorld
{
     public static void main(String []args)
     {
        boolean b1 = true;
        if (b1 == true)
        {
            System.out.println("b1 is true");
        }

        boolean b2 = true;
        if (b2)    //  شما می توانید در مقادیر بولی از این روش
نیز استفاده کنید
        {
            System.out.println("b2 is true");
        }


        int a1 = 9;
        int a2 = 4;
        if (a1 > a2)
        {
            System.out.println("a1 is greater than a2");
        }

        String s1 = "HELLO";
```

```
        String s2 = "hello";
        if (s1 == s2)     //  جاوا یک زبان حساس به بزرگ و کوچک
بودن حروف است
        {
            System.out.println("s1 is equal to s2");
        }
        else
        {
            System.out.println("s1 is not equal to s2");
        }
    }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
b1 is true
b2 is true
a1 is greater than a2
s1 is not equal to s2
```

استفاده از else if

این دستور در واقع زمانی به کار برده می شود که تعداد حالات بررسی زیاد می باشد. شکل کلی این دستور به صورت زیر می باشد

```
if (شـــرط)
{
        // دستورات را انجام بده
}
else if (اگر این شرط برقرار بود)
{
        // در غیر اینصورت این دستورات را انجام بده
}
else if (اگر این شرط برقرار بود)
{
        // در غیر اینصورت این دستورات را انجام بده
}
```


تمرین ۱ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که :

۱- متغیری از نوع int و به نام day در برنامه تعریف و مقدار ۴ را به آن اختصاص بدهید.

۲- با استفاده از else if مقدار متغیر day را بررسی و روز متناظر با عدد داخل این متغیر را در خروجی نمایش دهید.

خروجی برنامه بایستی به صورت زیر باشد

```
3shanbe
```

جواب تمرین ۱

```
public class HelloWorld
{
     public static void main(String []args)
     {
        int day = 4;

        if (day == 1)  //  اگر دستور یک خط بود میتوانید بدنه مشخص نکنید
            System.out.println("shanbe");
        else if (day == 2)
            System.out.println("1shanbe");
        else if (day == 3)
            System.out.println("2shanbe");
        else if (day == 4)
            System.out.println("3shanbe");
        else if (day == 5)
            System.out.println("4shanbe");
        else if (day == 6)
            System.out.println("5shanbe");
        else
            System.out.println("jome");
```

```
        }
}
```

Control Flow: کنترل روند اجرا

Linear: خطی

Nested If: if تو در تو

Condition: شرط

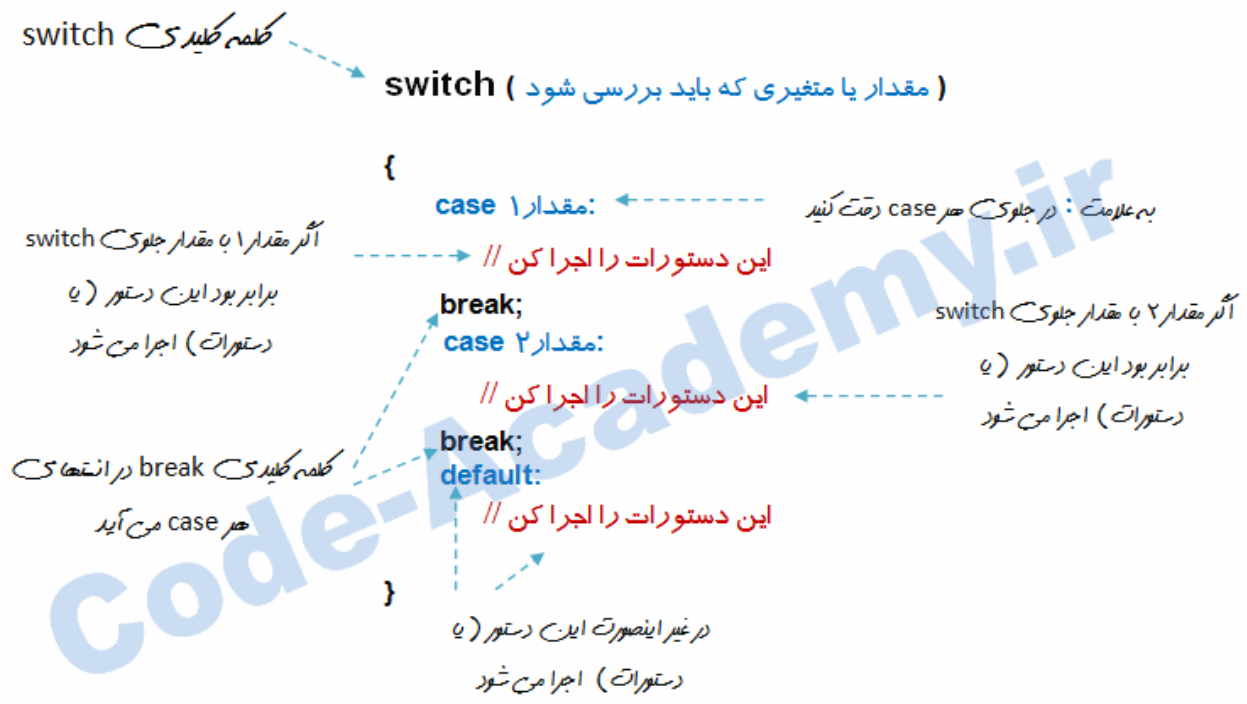
Block: بدنه

Keyword: کلمه کلیدی

دستور کنترلیSWITCH

دستور دیگری که برای کنترل روند اجرای برنامه استفاده می شود، دستور switch می باشد. این دستور بیشتر در مواقعی به کار می رود که تعداد حالات بررسی یک موضوع زیاد بوده و استفاده از if باعث پیچیدگی کار می شود.

شکل کلی استفاده از این دستور به صورت زیر می باشد:



روش استفاده از این دستور به این صورت است که ابتدا مقداری که می خواهیم بر اساس آن تصمیم بگیریم را در داخل پرانتز روبروی switch قرار می دهیم. سپس با استفاده از کلمه کلیدیcase ،

حالت های مختلف را بررسی می کنیم. در صورتی که حالتی (case) با مقدار روبروی switch برابر بود، دستورات داخل آن اجرا شده و پس از رسیدن به کلمه break از switch خارج می شویم.

استفاده از default در داخل switch به صورت اختیاری است. در صورتی که از این کلمه کلیدی استفاده کنیم به دستور switch گفته ایم که اگر هیچکدام از حالت هاcase) ها (با مقدار روبروی پرانتز switch برابر نبود، دستورات بعد از default را اجرا کن.

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

```
public class HelloWorld
{
     public static void main(String []args)
     {
        int user = 19;

        switch(user)
        {
            case 18:
                System.out.println("User is 18");
                break;
            case 19:
                System.out.println("User is 19");
                break;
            case 20:
                System.out.println("User is 20");
                break;
            default:
                System.out.println("User is not in 18, 19, 20");
```

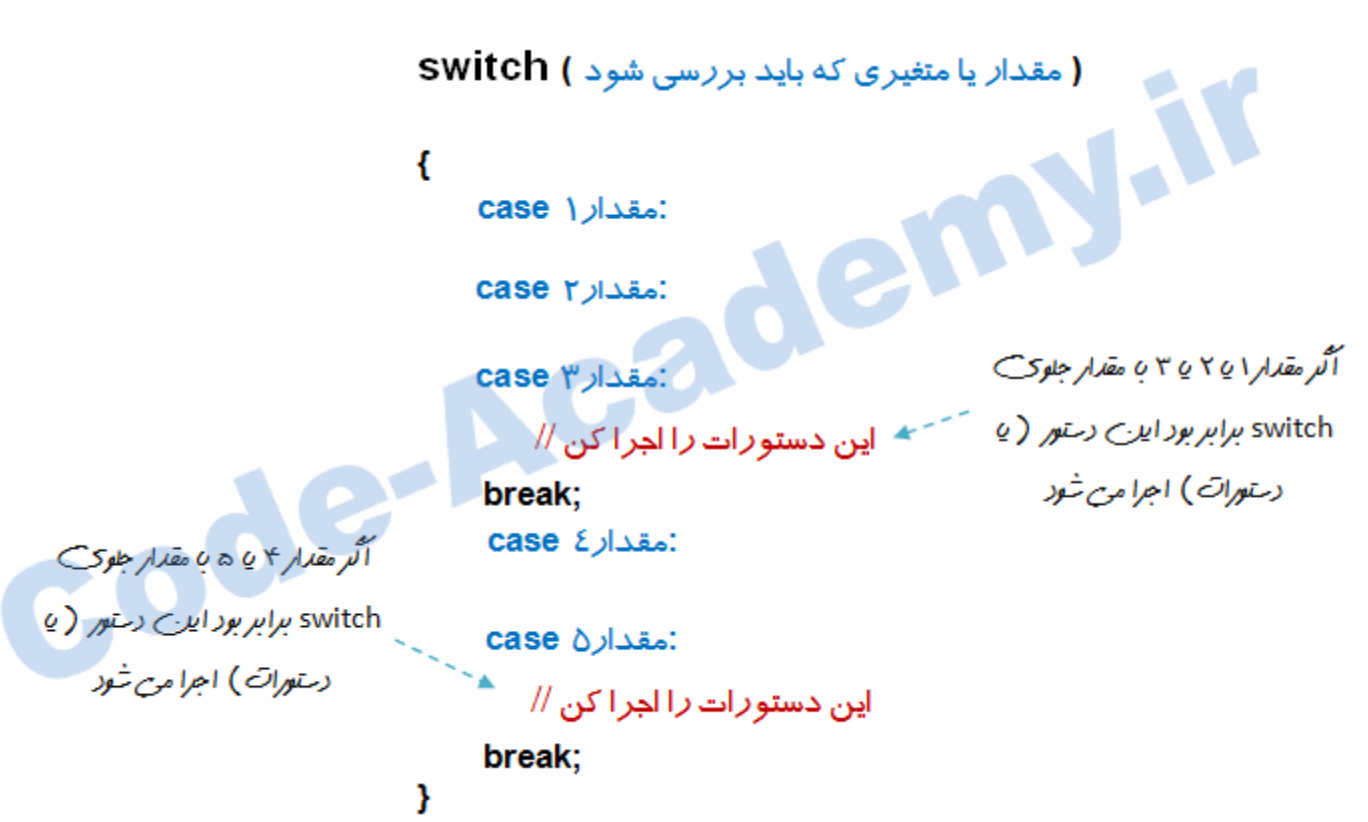
```
                break;
            }
        }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
User is 19
```

در این برنامه مقدار متغیر user در دستور switch بررسی می شود و در صورتی که این مقدار برابر ۱۸،۱۹،۲۰ بود، پیام مناسب را در خروجی نمایش می دهد و در غیر اینصورت پیغام را در خروجی چاپ می کند.

در صورتیکه بخواهید به صورت بازه ای مقداری را بررسی کنیم از این روش از Switch استفاده می کنیم:



در شکل بالا گفته ایم اگر مقدار جلوی کلمه switch برابر با مقدار ۲،۱ یا ۳ بود یک دستور و اگر برابر ۴ یا ۵ بود دستور دیگری را اجرا کن.

تمرین ۱ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که :

1-متغیری از نوع int و به نام day در برنامه تعریف و مقدار ۴ را به آن اختصاص بدهید.

-2با استفاده از switch مقدار متغیر day را بررسی و روز متناظر با عدد داخل این متغیر را در خروجی نمایش دهید.

خروجی برنامه بایستی به صورت زیر باشد

```
3shanbe
```

تمرین۲ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که :

-1متغیری از نوع int و به نام childAge در برنامه تعریف و مقدار ۷ را به آن اختصاص بدهید.
-2با استفاده از switch مقدار متغیر childAge را بررسی و گروه سنی متناظر با آنرا چاپ کند.

از ۳ تا ۶ سال گروه سنیA
از ۷ تا ۹ سال گروه سنیB

خروجی برنامه بایستی به صورت زیر باشد

```
Group B
```

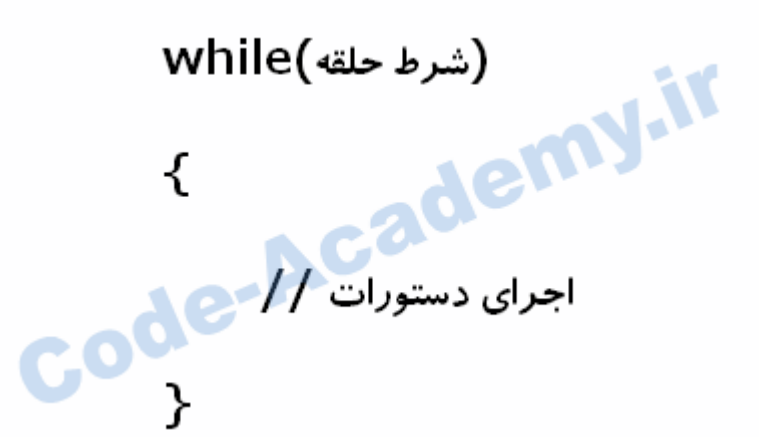
و این هم کد این تمرین:

```
public class HelloWorld
{
     public static void main(String []args)
     {
        int childAge = 7;

        switch(childAge)
        {
            case 3:
            case 4:
            case 5:
            case 6:
                System.out.println("Group A");
                break;
            case 7:
            case 8:
            case 9:
                System.out.println("Group B");
                break;

        }
     }
}
```

استفاده از حلقه WHILE و DO-WHILE

دستور while تا زمان برقراری شرط (یعنی true بودن آن) یک بلاک کد را به صورت مرتب اجرا می کند.

شکل کلی پیاده سازی این دستور به صورت زیر می باشد:



دستور while عبارت داخل پرانتز روبرویش را ارزیابی می کند. در صورتیکه این عبارت مقدار true را بر گرداند، دستور یا دستورات داخل بدنه while اجرا می شوند. پس از اجرای کد داخل بلاک، دستور while بار دیگر عبارت داخل پرانتزش را بررسی می کند. در صورتی که true بود بار دیگر کد داخل بدنه اش را اجرا و در صورت false بودن کد داخل بلاک دیگر اجرا نمی شود.

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید. این برنامه از ۰ تا ۹ را در خروجی نمایش می دهد.

روش کار این برنامه به صورت زیر است:

۱- متغیری به نام i تعریف و مقدار صفر را در آن قرار می دهیم

۲- شرط اجرای بدنه while را کوچک بودن i از عدد ۱۰ قرار می دهیم

۳- در صورتیکه شرط برقرار باشد، مقدار i نمایش داده شده و یک واحد به i اضافه می کنیم

۴- در صورتیکه باز هم مقدار i از ۱۰ کوچکتر بود، دستورات بلاک while دوباره اجرا می شوند.

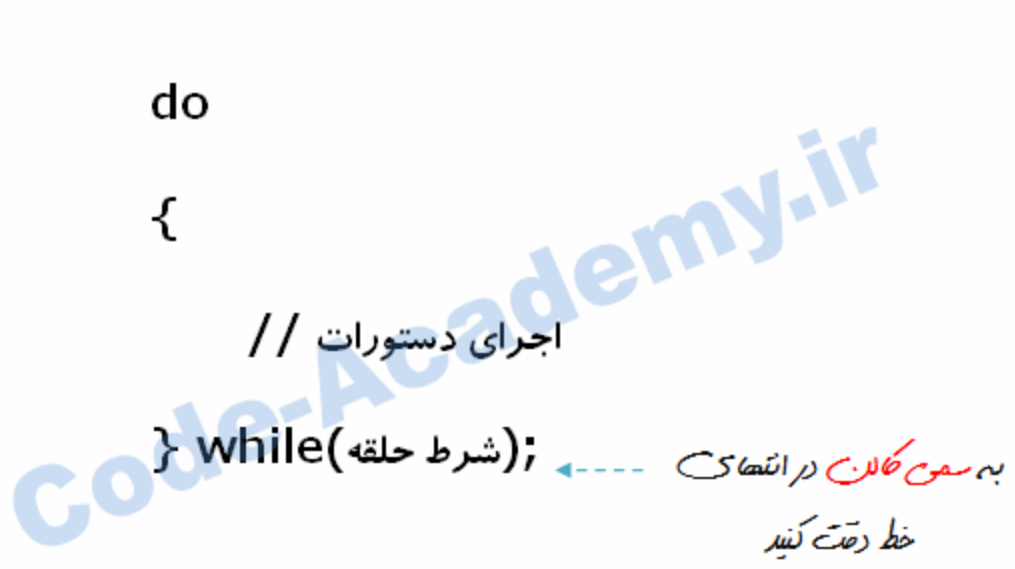
```
public class HelloWorld
{
     public static void main(String []args)
     {
        int i = 0;
        while (i < 10)
        {
            System.out.println(i);
            i = i + 1 ;    // i++;
        }
     }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
0
1
2
3
4
```

```
5
6
7
8
9
```

در جاوا شکل دیگری از دستور while وجود دارد. نام این دستور do-while می باشد. شکل کلی استفاده از این دستور به صورت زیر می باشد.



تفاوت اصلی بین while و do-while در این است که در do-while شرط true بودن عبارت را در پایین بدنه خود بررسی می کند. به همین دلیل در do-while کد داخل بلاک آن حداقل یکبار اجرا می شود.

مثال ۲

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید. در این مثال تصمیم به استفاده از do-while را دارم:

```
public class HelloWorld
{
     public static void main(String []args)
     {
        int i = 0;
        do
        {
            System.out.println(i);

        } while(i != 0);
     }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
0
```

همینطور که در مثال بالا مشاهده می کنید با اینکه شرط اجرای بدنه مخالف بودن i با عدد صفر است، ولی مقدار i در خروجی چاپ می شود. چراکه شرط در انتهای دستور قرار گرفته است. .

تمرین ۱ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که با استفاده از دستور while از ۱۰ تا ۱۰۰ را در خروجی نمایش دهد .

جواب تمرین ۱

```
public class HelloWorld
{
     public static void main(String []args)
     {
        int i = 10;
        while( i <= 100)
        {
            System.out.println(i);
            i = i + 1 ;    // i++;
        }
     }
}
```

Condition: شرط

While: تازمانیکه

Statement: دستور

Evaluate: ارزیابی

Expression: عبارت

Implement: پیاده سازی

استفاده از حلقه FOR

با استفاده از حلقه for می توان به سادگی در بین بازه ای از اعداد پیمایش کرد. فرم کلی استفاده از این دستور به صورت زیر می باشد:

```
for(مقداردهی اولیه ; شرط حلقه ; گام حلقه)
{
// اجرای دستورات
}
```


در هنگام استفاده از دستور for موارد زیر را در نظر داشته باشید:

در قسمت مقدار دهی، حلقه for مقداردهی اولیه می شود. این مقدار دهی فقط یکبار و در زمان شروع اجرای حلقه اتفاق می افتد.

وقتی شرط برقراری حلقه برقرار نباشد، یعنی مقدار false .را برگرداند، حلقه به اتمام می رسد

گام حلقه در هر بار اجرای حلقه می تواند کاهش یا افزایش کند

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید. این برنامه از ۱ تا ۱۰ را در خروجی نمایش می دهد. روش کار این برنامه به صورت زیر است:

۱- ابتدا حلقه مقدار دهی اولیه شده و بنابراین مقدار ۱ در متغیر i قرار داده می شود.

۲- در صورتیکه شرط حلقه برقرار بود (یعنی $i < 11$ بود) بدنه دستور for اجرا می شود.

۳- پس از اجرای بدنه، گام حلقه یک واحد افزایش پیدا می کند

۴- شرط حلقه بررسی می شود و در صورت برقراری بدنه for باز هم اجرا می شود

۵- مرحله ۳ و ۴ تا جایی ادامه پیدا می کند که i کوچکتر از ۱۱ نباشد

```
public class HelloWorld
{
     public static void main(String []args)
     {
          for(int i=1; i<11; i++)
          {
               System.out.println(i);
          }
     }
}
```

مثال ۲

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید. این برنامه از ۱۰ تا ۱ را در خروجی نمایش می دهد:

```
public class HelloWorld
{
     public static void main(String []args)
     {
          for(int i=10; i>0; i--)
          {
               System.out.println(i);
          }
     }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
10
9
8
7
6
5
4
3
2
1
```

در این برنامه به مقدار دهی اولیه، شرط برقراری حلقه و گام حلقه دقت کنید.

تمرین ۱ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که با استفاده از دستور for اعداد زوج بین ۲۰ تا ۴۰ را نمایش دهد .

خروجی این برنامه باید به صورت زیر باشد:

```
20
22
24
26
28
30
32
34
36
38
40
```

جواب تمرین ۱

```
public class HelloWorld
{
     public static void main(String []args)
     {
          for(int i=20; i<= 40; i += 2 )
          {
               System.out.println(i);
          }
     }
}
```

Iterate : پیمایش کردن

range : بازه

initialization: مقدار دهی اولیه

increment: افزایش

decrement : کاهش

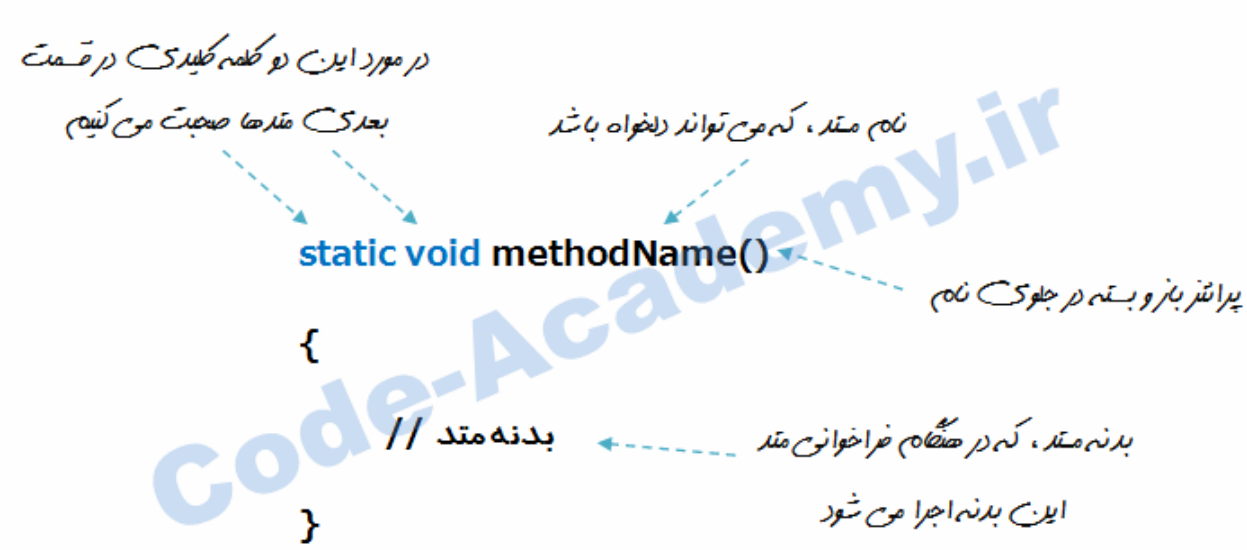
evaluate : بررسی کردن

متد و نحوه پیاده سازی آن

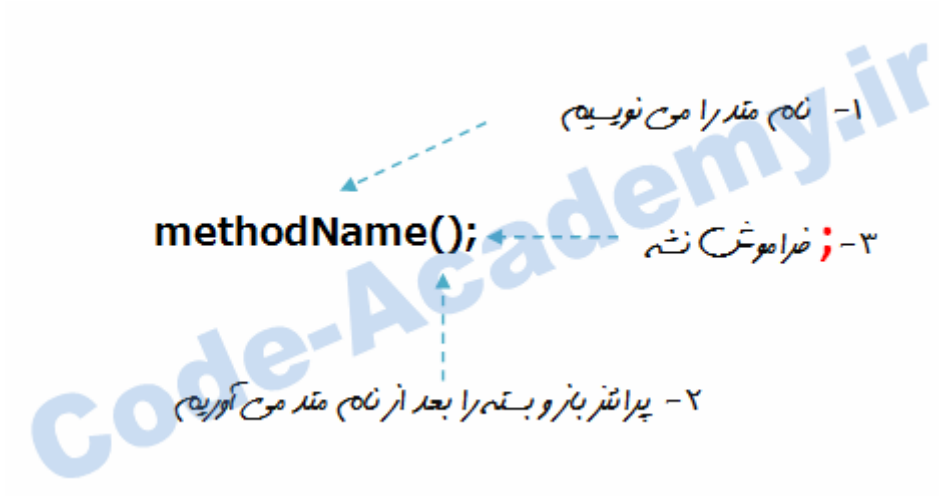
در ساده ترین تعریف می توان گفت متدها کاری را برای ما انجام می دهند. مثلا متد println برای ما مقداری را در خروجی نمایش می دهد. این متد در کتابخانه کلاس جاوا (System.out) قرار گرفته است. به این معنی که پیاده سازی این متد قبلا صورت گرفته و ما فقط از آن استفاده می کنیم. ولی ما می توانیم خودمان هم متد نوشته و از آن استفاده کنیم.

برای استفاده از متدها دو مرحله لازم است، ۱- تعریف متد و ۲- فراخوانی (استفاده از) متد

شکل کلی تعریف متد به صورت زیر است:



شکل کلی فراخوانی متد به صورت زیر است:



در مثال زیر نحوه تعریف متد و همینطور نحوه فراخوانی متد آورده شده است!

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

در این مثال متدی می نویسیم که با فراخوانی آن اعداد ۱ تا ۱۰ را در خروجی چاپ نماید

```
public class HelloWorld
{
     public static void main(String []args)
     {
        printNumbers();
     }

     static void printNumbers()
     {
        for(int i = 1; i<=10 ; i++)
        {
            System.out.println(i);
        }
     }
}
```

همینطور که مشاهده می کنید متدی به نام printNumbers **تعریف و آنرا از داخل متد** main **فراخوانی کردیم.**

خروجی این برنامه به صورت زیر می باشد:

```
1
2
3
4
5
6
7
8
9
10
```

تمرین ۱ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که :

۱- متدی به نام printNumbers بنویسید که اعداد زوج بین ۲۰ تا ۴۰ را در خروجی نمایش دهد.

۲- این متد را از داخل متد main فراخوانی کنید.

خروجی این برنامه باید به صورت زیر باشد:

```
20
22
24
26
28
30
32
34
36
38
40
```

جواب تمرین ۱

```
public class HelloWorld
{
     public static void main(String []args)
     {
        printNumbers();
     }

     static void printNumbers()
     {
        for(int i = 20; i<=40 ; i =  i+ 2)
        {
            System.out.println(i);
        }
     }
}
```

Method : متد، رویه

Class library : کتابخانه کلاس

Initialization: مقدار دهی اولیه

Method declaration: تعریف متد

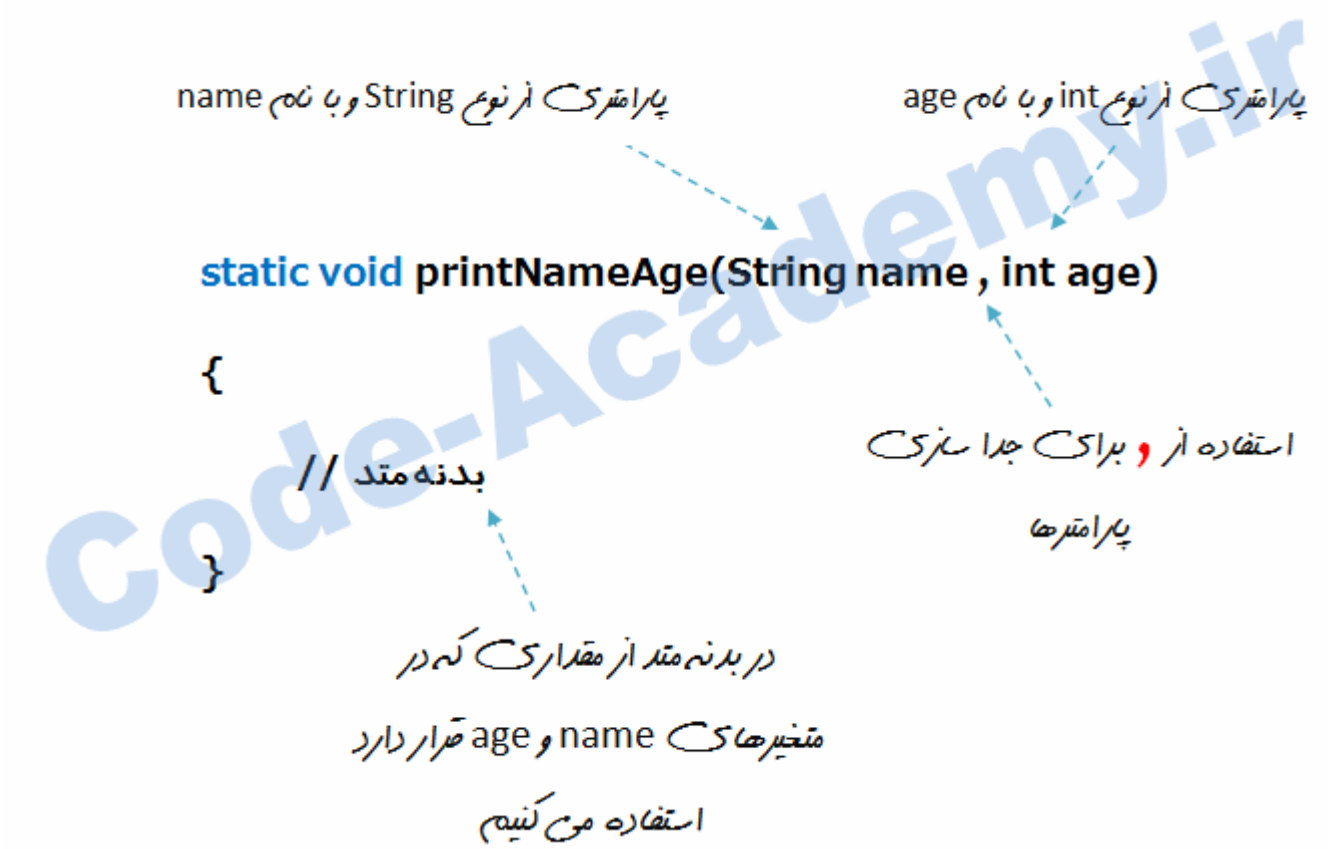
Invoke : فراخوانی

Void : هیچ

ارسال پارامتر به متد

تا به اینجا نحوه تعریف متد و فراخوانی آن را آموزش دادم. در این قسمت در مورد متدهایی صحبت خواهم کرد که یک یا چند پارامتر ورودی می گیرند. پارمتر یعنی ورودی برای متد. در واقع متد اگر پارامتر ورودی داشته باشد از مقدار آن در بدنه متد استفاده خواهد کرد.

برای تعریف متدهایی که یک یا چند پارامتر ورودی می گیرند به صورت زیر عمل می کنیم:



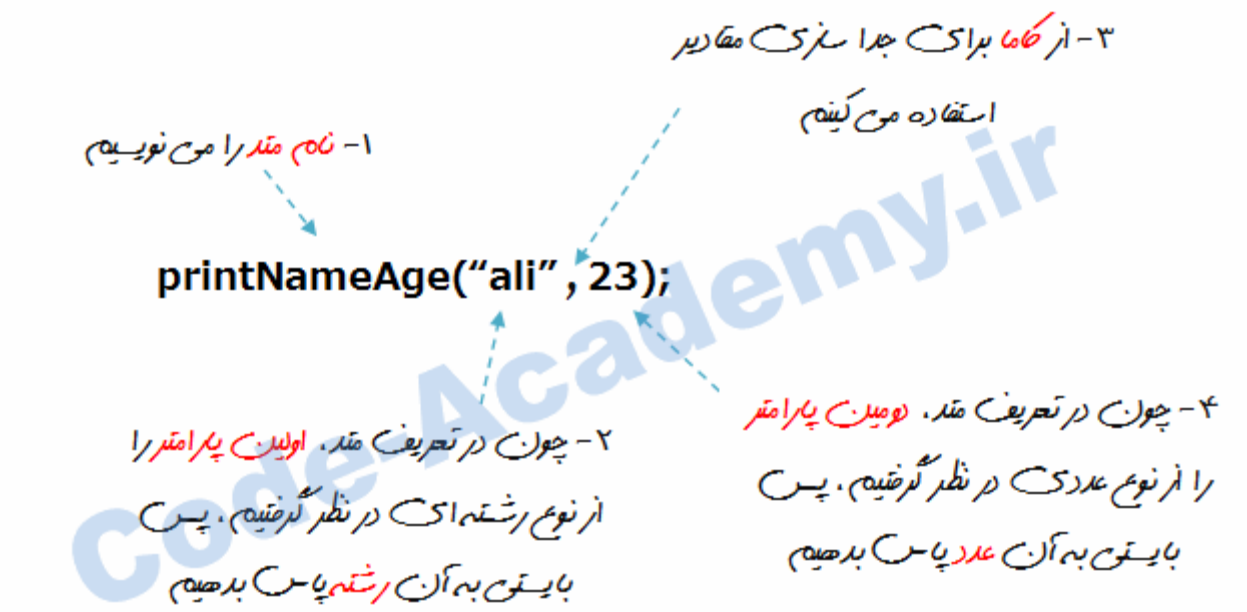
در شکل بالا متدی تعریف کردیم که:

نام آن printNameAge **است**

این متد دو پارامتر می گیرد

پارامتر ها با کاما از یکدیگر جدا شده اند

و اما نحوه فراخوانی متد هایی که پارامتر می گیرند به این صورت است:



برای درک هر چه بهتر به مثال زیر توجه کنید

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

در این مثال متدی نوشته ایم که نام و سن را گرفته و در خروجی نمایش می دهد.

```
public class HelloWorld
{
     public static void main(String []args)
     {
        printNameAge("ali",30); // فراخوانی متد
        printNameAge("sara",24); // فراخوانی متد
        printNameAge("negar",20); // فراخوانی متد
     }

     // تعریف متد
     static void printNameAge(String name,int age)
     {
         System.out.println("Name is " + name + " and age is " +
age);
     }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
Name is ali and age is 30
Name is sara and age is 24
Name is negar and age is 20
```

در این مثال ابتدا متدی تعریف کردیم که دو پارامتر می گیرد. سپس در متد main سه بار این متد را فراخوانی کردیم. چند نکته در مورد فراخوانی متدهایی که پارامتر می گیرند را در نظر داشته باشید:

اگر فرضا متد شما دو پارامتر می گرفت، شما بایستی در هنگام فراخوانی، دقیقا دو پارامتر برای آن ارسال کنید

ترتیب در نوشتن مقدار مهم است. مثلا در برنامه بالا نمی توانید اول سن را فرستاده و سپس نام را وارد کنید

```
printNameAge(24,"sara");
```

به نوع پارامترها دقت کنید. مثلا در برنامه بالا نمی توانید یک مقدار رشته ای به جای عددی به متد ارسال کنید

```
printNameAge("ali","30");
```

مثال ۲

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

فرض کنید می خواهید برای سه کودک رده سنی شان را بین A و B مشخص کنید:

از ۳ تا ۶ سال گروه سنیA

از ۷ تا ۹ سال گروه سنیB

```
public class HelloWorld
{
     public static void main(String []args)
     {
        int saraAge = 4;
        int nedaAge = 7;
        int pedramAge = 8;

        printAgeGroup(saraAge);   //فراخوانی متد
        printAgeGroup(nedaAge); //فراخوانی متد
        printAgeGroup(pedramAge); //فراخوانی متد
     }


    // تعریف متد
    static void printAgeGroup(int age)
    {
        switch(age)
        {
            case 3:
            case 4:
            case 5:
            case 6:
                System.out.println("Group A");
                break;
            case 7:
            case 8:
            case 9:
                System.out.println("Group B");
```

```
                break;

            }
        }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
Group A
Group B
Group B
```

در کل با استفاده از متدها:

یک کد را یک مرتبه نوشته و می توانیم بارها از آن در برنامه استفاده کنیم

خوانایی کد ما بالاتر می رود، زیرا از نوشتن کدهای تکراری جلوگیری می شود

تغییرات آسان تراست، چرا که کافی است فقط بدنه متد را تغییر دهید

تمرین ۱ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که :

1-این برنامه متدی به نام sum داشته باشد که دو عدد را گرفته و با هم جمع کند، سپس حاصل جمع را نمایش دهد

2-متد sum را با اعداد ۳ , ۵ فراخوانی کنید

3-متد sum را با اعداد ۴ , ۱۲ فراخوانی کنید

خروجی این برنامه باید به صورت زیر باشد:

```
8
16
```

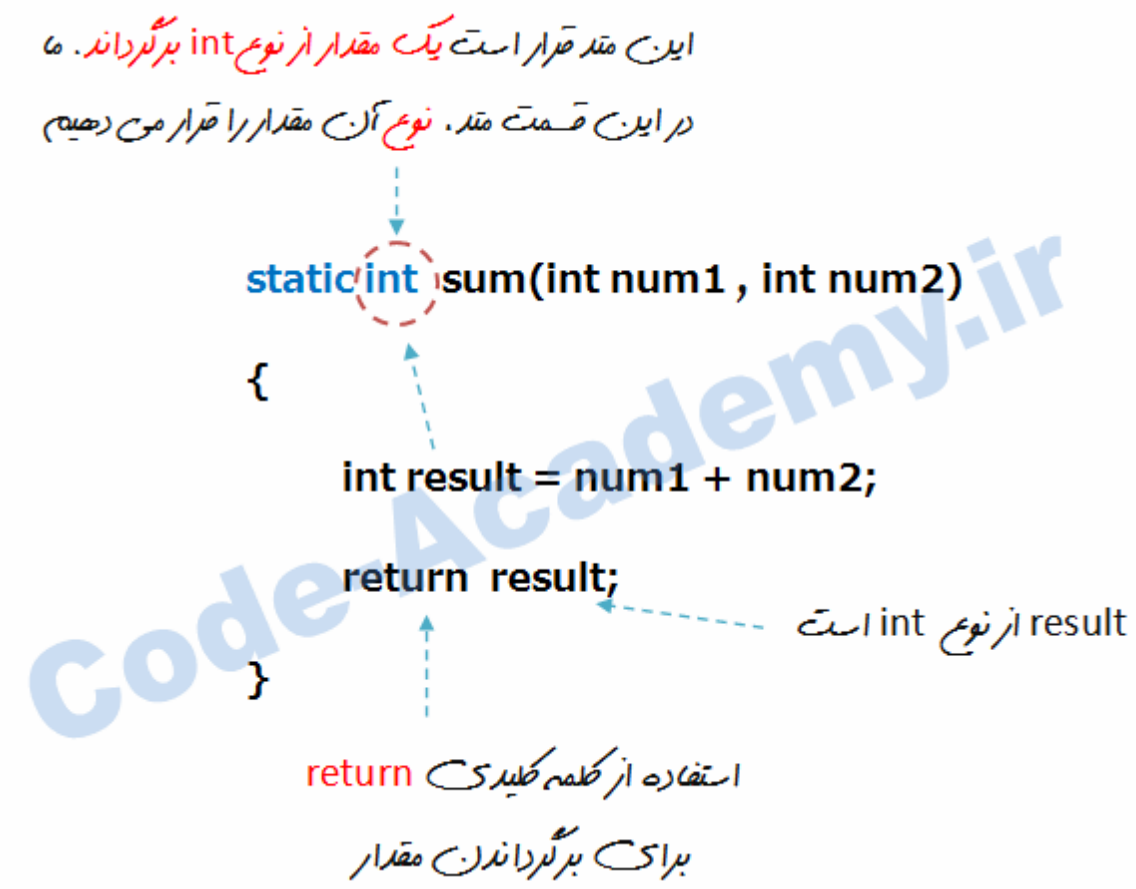
جواب تمرین ۱

```
public class HelloWorld
{
    public static void main(String []args)
    {
        sum(3,5);  //فراخوانی متد
        sum(4,12); //فراخوانی متد
    }


    // تعریف متد
    static void sum(int num1, int num2)
    {
        int result = num1 + num2;
        System.out.println(result);
    }
}
```

ارسال پارامتر به متد و دریافت مقدار از آن

در این قسمت به بررسی متدهایی می پردازیم که یک یا چند ورودی گرفته و سپس بر روی آن عملیاتی انجام داده و نتیجه آن عملیات را به عنوان خروجی متد، برای ما باز می گردانند.

برای تعریف این متد ها، به صورت زیر عمل می کنیم:



در شکل بالا متدی تعریف کردیم که:

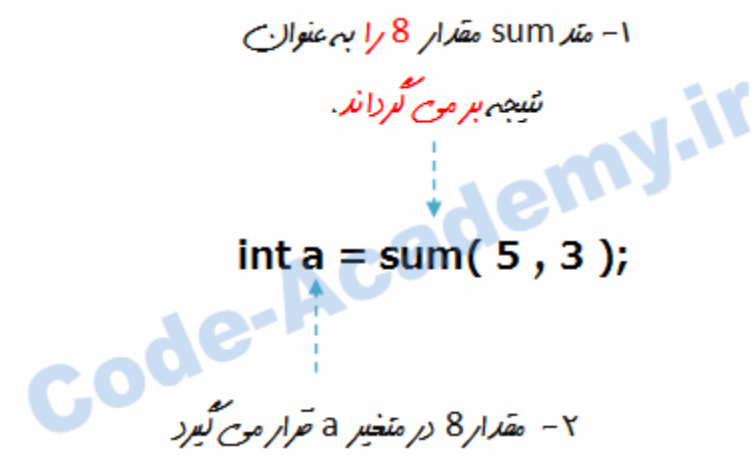
نام آن sum است

این متد دو عدد را به عنوان پارامتر می گیرد

یک متغیر از نوع int و به نام result تعریف کرده، مقدار جمع دو پارامتر را در آن قرار می دهد.

با استفاده از کلمه کلیدی return مقدار result را به عنوان نتیجه بر می گرداند

و اما نحوه فراخوانی متد هایی که مقداری بر می گردانند به این صورت است:



برای درک هر چه بهتر به مثال زیر توجه کنید

مثال ۱

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

در این مثال متدی نوشته ایم که دو عدد صحیح) از نوع (int گرفته و حاصل جمع آنرا برای ما بر می گرداند.

```
public class HelloWorld
{
    public static void main(String []args)
    {
        int a = sum(3,5);  //فراخوانی متد
        System.out.println(a);
    }


    // تعریف متد
    static int sum(int num1, int num2)
    {
        int result = num1 + num2;
        return result;
    }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
8
```

مثال ۲

کد زیر را در اکلیپس یا کامپایلر آنلاین اجرا کنید.

متدی بنویسید که دو رشته گرفته، آن دو رشته را به همدیگر چسبانده و نتیجه را بر گرداند

```
public class HelloWorld
{
    public static void main(String []args)
    {
        String s = ConcatStrings("sara ","ahmadi");
//فراخوانی متد
        System.out.println(s);
    }


    // تعریف متد
    static String ConcatStrings(String s1, String s2)
    {
        String result = s1 + s2;
        return result;
    }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
sara ahmadi
```

تمرین ۱ : برنامه ای در اکلیپس یا کامپایلر آنلاین بنویسید که :

متدی بنویسید که دو عدد اعشاری) از نوع (double گرفته و حاصل جمع دو عدد ۲,۷۵ و ۵ را برای ما بر می گرداند.

خروجی این برنامه باید به صورت زیر باشد:

```
7.75
```

جواب تمرین ۱

```
public class HelloWorld
{
    public static void main(String []args)
    {
        double a = sum(2.75,5);   //فراخوانی متد
        System.out.println(a);
    }


    // تعریف متد
    static double sum(double num1, double num2)
    {
        double result = num1 + num2;
        return result;
    }
}
```

تمرینات

پیاده سازی ماشین حساب ساده

در این تمرین تصمیم دارم از متدها استفاده کنم. برنامه ما ۴ متد دارد که برای انجام چهار عمل اصلی ریاضی یعنی عملیات ضرب، تقسیم، جمع، تفریق استفاده می شود.

```
public class HelloWorld
{
     public static void main(String []args)
     {
        int r = sum(10,25);
        System.out.println(r);

        int r1 = minus(140,40);
        System.out.println(r1);

        int r2 = division(40,10);
        System.out.println(r2);

         int r3 = multiply(6,8);
        System.out.println(r3);

     }

       // متد جمع
        public static int sum(int a,int b)
        {
            int result = a + b;
            return result;
        }

        // متد تفریق
        public static int minus(int a,int b)
        {
            int result = a - b;
```

```
            return result;
        }

        // متد تقسیم
        public static int division(int a,int b)
        {
            int result = a / b;
            return result;
        }

        // متد ضرب
        public static int multiply(int a,int b)
        {
            return a * b;    //  می توان از این حالت نیز برای
برگرداندن مقدار استفاده کرد
        }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
35
100
4
48
```

در این تمرین از متدها استفاده کردیم. اگر به متد multiply دقت کنید متوجه می شوید که برای برگرداندن مقدار از روش دیگری استفاده کرده ام. در این روش مستقیما حاصل ضرب دو متغیر a و b را به عنوان خروجیِ متد قرار داده ام.

برنامه محاسبه عدد بزرگتر در بین دو عدد

در این برنامه متدی می نویسیم که دو عدد را به عنوان پارامتر ورودی گرفته و عدد بزرگتر را به عنوان نتیجه برگرداند.

```
public class HelloWorld
{
    public static void main(String []args)
    {
        int maxNumber = getMax(7,12);
        System.out.println("Max number is: " + maxNumber);
    }

    static int getMax(int num1,int num2)
    {
        int max = 0;

        if (num1 > num2)
            max = num1;
        else
            max = num2;

        return max;
    }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
Max number is: 12
```

برنامه تعیین زوج یا فرد بودن یک عدد

در این برنامه متدی می نویسیم که یک عدد گرفته و زوج یا فرد بودن آن عدد را بر می گرداند

```
public class HelloWorld
{
    public static void main(String []args)
    {
        boolean b = isEven(12);

        if (b == true)
        {
            System.out.println("This number is even!");
        }
        else
        {
            System.out.println("This number is odd!");
        }
    }

    // نام متد به صورت سوالی نوشته شده. آیا زوج است؟
    static boolean isEven(int num)
    {
        boolean result;

        int a = num % 2 ;

        if (a == 0)
        {
            result = true;
        }
        else
        {
            result = false;
        }
```

```
        return result;
    }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
This number is even!
```

با استفاده از عملگر باقیمانده (٪) می توان با قیمانده عددی را بدست آورد. حال اگر باقیمانده تقسیم عددی بر ۲ برابر صفر باشد آن عدد زوج است. در غیر اینصورت آن عدد فرد است

در این قسمت جواب متد isEven **برای عدد ۱۲، در متغیر** b **ریخته می شود. اگر** b **برابر** true **شد یعنی عدد زوج است و در غیر اینصورت عدد فرد بوده است.**

برنامه تعیین زوج یا فرد بودن یک عدد

در این برنامه متدی می نویسیم که یک عدد گرفته و زوج یا فرد بودن آن عدد را بر می گرداند

```
public class HelloWorld
{
    public static void main(String []args)
    {
        boolean b = isEven(12);

        if (b == true)
        {
            System.out.println("This number is even!");
        }
        else
        {
            System.out.println("This number is odd!");
        }
    }

    // نام متد به صورت سوالی نوشته شده. آیا زوج است؟
    static boolean isEven(int num)
    {
        boolean result;

        int a = num % 2 ;

        if (a == 0)
        {
            result = true;
        }
        else
        {
            result = false;
        }
```

```
        return result;
    }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
This number is even!
```

برنامه تبدیل رشته به عدد

همانطور که قبلا گفتم اگر عددی داخل کوتیشن باشد، خاصیت عددی خود را از دست داده و تبدیل به یک رشته می شود. به عنوان مثال "۱۲" یک رشته است. ما نمی توانیم بر روی رشته ها عملیات ریاضی انجام دهیم. اگر فرضا بخواهیم بر روی عدد داخل این رشته عملیات ریاضی انجام دهیم، باید ابتدا عدد داخل آن را استخراج کنیم. در این برنامه متدی می نویسیم که یک رشته با محتوای عددی بگیرد و عدد داخل آنرا برای ما برگرداند. سپس این عدد را در ۲ ضرب کرده و نمایش می دهیم.

```
public class HelloWorld
{
    public static void main(String []args)
    {
        int num = getNumberFromString("12");

        // عدد استخراج شده در ۲ ضرب می شود
        num = num * 2;

        System.out.println(num);
    }

    static int getNumberFromString(String str)
    {
        // این دستور برای استخراج عدد از رشته به کار می رود
        int number = Integer.parseInt(str);
        return number;
    }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
24
```

با دستور زیر می توانیم عدد صحیح را از داخل رشته استخراج کنیم.

```
int number = Integer.parseInt(str);
```

محاسبه فاکتوریل

در این تمرین متدی می نویسیم که عددی به عنوان پارامتر گرفته و با استفاده از یک حلقه for فاکتوریل آن عدد را بدست بیاورد. به عنوان مثال محاسبه فاکتوریل عدد ۵ به صورت زیر می باشد.

$$5! = 1 * 2 * 3 * 4 * 5 = 120$$

```
public class HelloWorld
{
    public static void main(String []args)
    {
        int result = factorial(5);
        System.out.println("Factorial of 5 is: " + result);
    }

    static int factorial(int num)
    {
        int f = 1;

        for(int i=1 ; i<=num ; i++)
        {
            f = f * i;
        }

        return f;
    }
}
```

خروجی این برنامه به صورت زیر می باشد:

```
Factorial of 5 is: 120
```

و در پایان شما را دعوت به استفاده از آموزش های ویدئوئی موجود در وب سایت کد آکادمی می کنم

code-academy.ir

با آرزوی موفقیت
بهنام خانی
کد آکادمی