

آموزش زبان برنامه نویسی جاوا

ورودی گرفتن

جلسه دهم

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



در برنامه نویسی برای مقدار دهی کردن به متغیر ها گاهی نیاز است که از کاربر درخواست شود که مقدار مورد نظر خود را به متغیر بدهد.

برای جا انداختن این مفهوم به مثال زیر توجه کنید:

فرض کنید شما می خواهید یک برنامه ساده برای جمع زدن دو عدد بنویسید!!!!

دو راه وجود دارد!!!!

راه اول :

```
package iran;

public class Sum {

    public static void main(String[] args) {

        int a, b;
        a = 10;
        b = 15;
        int sum = 0;
        sum = a + b;
        System.out.println(a + " + " + b + " = " + sum);

    }
}
```

خروجی:

10 + 15 = 25

- در این برنامه ساده دو متغیر تعریف شده که ابتدا به آن ها مقدار دهی شده و سپس با تعریف متغیر دیگری به نام SUM مجموع آن دو متغیر داخل متغیر SUM ریخته شده است. و در پایان مجموع دو متغیر چاپ می شود.
- در این جا به هنگام کد زنی دو مقدار ۱۰ و ۱۵ به ترتیب به متغیر a و b داده شده است! اما.... اگر بخواهیم مقدار دلخواه به این دو متغیر بدهیم و باهم جمع شوند باید چکار کنیم؟! یعنی میشه هنگام اجرا برنامه از کاربر درخواست بشه که با کیبورد مقدار عددی دلخواه به این دو متغیر بدهد؟!!!!!! جواب مثبت می باشد!!! ادامه آموزش رو دنبال کنید.

کتابخانه های (پکیج های) آماده جاوا

کتابخانه ها در زبان های برنامه نویسی چه کاربردی دارند؟!

کتابخانه های هر زبان برنامه نویسی کد هایی که از قبل توسط برنامه نویسان زده شده و آماده در اختیار شما قرار داده می شود. پس دیگر نیازی نیست خودمان رو برای برخی دستورات به زحمت بیاندازیم چون می توانیم از کتابخانه موجود در زبان جاوا استفاده کنیم.

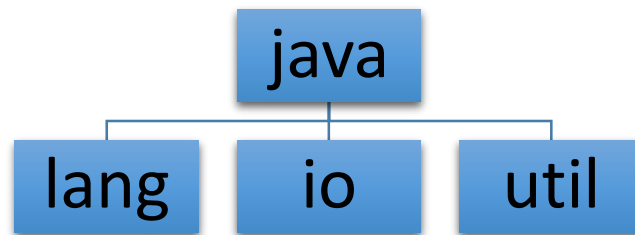
یکی از نقاط قوت جاوا داشتن کتابخانه های بزرگ می باشد هر کتابخانه شامل کدهایی می باشد که کار خاصی رو برای ما انجام می دهد.

ما با استفاده از یک کتابخانه در برنامه خود می توانیم به تمامی متدها و دستورات آن استفاده کنیم.

- به این کتابخانه ها پکیج هم می گویند که شامل مجموعه ای از class ها , interface ها و... می باشد. و هر کلاس هم مجموعه از ویژگی ها و متدها را دارا می باشد.
- در مورد interface ها جلسات آینده صحبت خواهیم کرد.
- پس ما می توانیم با توجه به نیاز از این پکیج های آماده جاوا در برنامه خود استفاده کنیم. خوب چطور به این پکیج ها دسترسی داشته باشیم؟!
- روش ساخت پکیج ها برای دسته بندی و مرتب سازی و نظم بخشیدن به کلاس هامون رو در جلسات قبل بررسی کردیم، حالا در این آموزش قصد داریم روش استفاده از پکیج های آماده جاوا را بررسی کنیم.
- در جاوا قبلا تعداد پکیج هایی تعریف شده که ما به هنگام برنامه نویسی می توانیم از آنها استفاده کنیم نظیر:

java.lang
java.io
java.util

.....9



چطور از این پکیج ها در برنامه مون استفاده کنیم؟


ما با استفاده از کلمه کلیدی "import" می توانیم به این پکیج ها درسی پیدا کنیم.


فرض کنیم می خواهیم به تمامی کلاس های آماده موجود در پکیج util دسترسی پیدا کنیم :


```
import java.util.*;
```


ترتیب پیاده سازی دستور دسترسی به این پکیج بصورت زیر است:


- 1) **import**
- 2) **import java**
- 3) **import java.**
- 4) **import java.util**
- 5) **import java.util.**
- 6) **import java.util.*;**

در مورد شماره ۴ با کاراکتر نقطه (.) به نام پکیج مورد نظر خود که util می باشد اشاره کردیم. 

خب در شماره ۵ با کاراکتر نقطه (.) قصد داریم به یکی از کلاس های موجود در این پکیج اشاره کنیم . 

در مورد شماره ۶ اگر کلاس خاصی از این پکیج رو در نظر ندارید با تایپ * بعد از نقطه می توانید به تمامی کلاس های موجود در این پکیج دسترسی پیدا کنید. 

در تصویر (۱) بخشی از کلاس های موجود در این پکیج رو مشاهده می کنید که با کلیک بر روی یکی از آن ها می توانید از ویژگی ها و متدهای آن استفاده کنید. 

با زدن دکمه ترکیبی **ctrl+space** بعد از نقطه (.) این کلاس ها به شما پیشنهاد داده می شود!!! یکی از ویژگی های برنامه Eclipse همین می باشد که نیاز به حفظ کردن برخی دستورات خاص رو ندارید و این برنامه به شما این دستورات رو پیشنهاد می دهد و شما می توانید از آن ها استفاده و مفهیم ان ها را با آزمون و خطا یاد بگیرید!!!!!! 

همان طور که می بینید تنها این پکیج شامل ۱۰ ها کلاس می باشد که با توجه به نیاز و ضرورت می توان از شون استفاده کرد پس نیازی نیست همه آن ها را یاد بگیرید مگر این که نیاز شما به آن ها بیافتد. و معمولا در آموزش های برنامه نویسی با توجه به مفهیم زیاد و مهم یک مثال روی یک دستور می زنند و بقیه دستورات برای یاد گیری به خود شما و نیاز شما بستگی دارد!!!

کلاس های پیشنهادی با زدن دکمه ترکیبی **ctrl+space**

```

1 package iran;
2 import java.util.
3 public class Sum
4
5     public static
6
7     int a,b;
8     a=10;
9     b=15;
10    int sum=0
11    sum=a+b;
12    System.out
13
14
15
16    }
17
18 }
19

```

Scanner - java.util
 AbstractCollection - java.util
 AbstractList - java.util
 AbstractMap - java.util
 AbstractQueue - java.util
 AbstractSequentialList - java.util
 AbstractSet - java.util
 ArrayDeque - java.util
 ArrayList - java.util
 Arrays - java.util
 Base64 - java.util
 BitSet - java.util
 Calendar - java.util
 Collection - java.util
 Collections - java.util
 Comparator - java.util
 ConcurrentModificationException - java.util
 Currency - java.util
 Date - java.util
 Deque - java.util
 Dictionary - java.util
 DoubleSummaryStatistics - java.util
 DuplicateFormatFlagsException - java.util
 EmptyStackException - java.util
 Enumeration - java.util
 EnumMap - java.util
 EnumSet - java.util
 EventListener - java.util
 EventListenerProxy - java.util

Press 'Ctrl+Space' to show Template Proposals

تصویر (۱)

پس اگر نیاز به کلاس خاصی داشتید نام آن را از بین کلاس ها پیدا کرد در غیر این صورت با گذاشتن کاراکتر * می توانید به تمامی کلاس های موجود در پکیج دسترسی پیدا کنید.

دستورالعمل دسترسی به سایر پکیج ها هم شبیه به همین مثال پکیج util می باشد.

مکان پیاده سازی دسرسی به پکیج ها همان طور که در تصویر (۱) و (۲) مشاهده میکنید بالا قبل از تعریف کلاس می باشد.

ما می توانید به هر تعداد دلخواه پکیج مورد نیاز به اصطلاح import کنیم. تصویر (۲)

```

1 package iran;
2 import java.util.*;
3 import java.util.Objects;
4 public class Sum {
5
6     public static void main(String[] args) {
7

```

تصویر (۲)

مکان استفاده از پکیج های حاوی کلاس های آماده جاوا و استفاده از دو پکیج که یکی همه کلاس ها رو مورد استفاده قرار داده و دیگری تنها از یک کلاس استفاده کرده است.

پس دو نوع پکیج در جاوا وجود دارد:

۱. پکیج هایی خود ما برای کلاس هامون ایجاد میکنیم

۲. پکیج هایی که از قبل آماده وجود دارد.

خب با پکیج های و کلاس های آماده جاوا و روش استفاده از آن ها آشنا شدیم. فقط این نکته رو بیاد داشته باشید که جاوا کلاس ها و پکیج های آماده خیلی خیلی زیادی داره و ما در این آموزش نمی توانیم تک به تک بهشون بپردازیم چون منطقی هم نیست ، من خودم هیچ کدوم از این دستورات رو نمی دونم!!!! یعنی در کل اصلا جاوا را حفظ نکردم!!!! پس چطور می تونم برنامه نویسی جاوا کنم؟!!!! جوابش اینه که من هر بار که قصد برنامه نویسی رو دارم و با توجه به نیاز به پکیج یا کلاسی خاص اون رو سرچ میکنم و ازش استفاده میکنم به همین راحتی!!!! و پیشنهاد من به شما اینه لازم نیست دستورات و... جاوا رو حفظ کنید و شما باید مهارت این رو به دست بیاورید که هر بار در برنامه نویسی بدونید به چه چیز نیاز دارید و اون رو سرچ کنید و ازش استفاده کنید!!!! به همین راحتی!!!! چون اگر بدون هدف قصد یادگیری تمامی مفاهیم رو داشته باشید دل زده و خسته و نا امید می شوید. ما در این آموزش قصد داریم با مفاهیم مهم و ضروری جاوا آشنا شویم یاد گیری جزییات با شماست!!!! یعنی امکان یک کلاس آماده ۲۰ متد داشته باشه که ما تنها به یک متد آن بپردازیم و اون ۱۹ متد هم اگر نیاز بهش پیدا کردید دیگه بستگی به خودتون داره هرچند دیگه کار سختی نیست برای یادگیریشون!!!! هر چند در جریان یادگیری سوال داشتید یا به مشکلی برخوردید ما به شما پاسخ می دهیم. خب بریم سراغ ادامه آموزش 😊

در آینده خیلی خیلی با این پکیج ها و کلاس های آماده در جاوا نیاز پیدا میکنیم!!!!!!!



حالا که با پکیج هاو کلاسهای جاوا آشنا شدیم به سراغ ورودی گرفتن در جاوا می رویم.

- برای ورودی گرفتن و مقدار دهی به متغیر ها از کلاس آماده Scanner استفاده میکنیم.
- از واژه کلاس آماده استفاده کردیم!!!! پس باید ذهنمون بره به طرف پکیج های آماده جاوا!!!!
- کلاس Scanner یک کلاس در پکیج java.util می باشد.
- این کلاس به کاربر اجازه می دهد که مقدار از انواع نوع متغیر ها را از ورودی نظیر وردی کیبورد بخواند.
- متدهای زیادی در کلاس Scanner وجود دارد که شما می توانید از آن ها استفاده کنید.
- ما در این جلسه تنها به یک زیر مجموعه کوچک از این کلاس برای خواندن ورودی از کیبورد می پردازیم.
- روش استفاده (import) کردن کلاس Scanner که از پکیج util می باشد مثل سایر import کردن پکیج ها بصورت زیر است:

```
import java.util.Scanner;
```

- همان که مشاهده میکنید تنها از یک کلاس از پکیج util به نام Scanner استفاده کردیم.
- ما در جلسه ۹ در مبحث شی گرایی دریافتیم که برای دسترسی به تمامی ویژگی ها و رفتار های یک کلاس باید از آن شی ساخت.
- خب Scanner هم یک کلاس می باشد که برای دسترسی و استفاده از متد های درونش باید ازش یک شی بسازیم.

```
Scanner input=new Scanner(System.in);
```

- نام کلاس Scanner
- نام شی ساخته شده از کلاس input (همان طور که می دانید تعیین نام برای اشیا و متغیر ها به دلخواه می باشد)
- استفاده از کلمه کلیدی new
- استفاده از سازنده Scanner(System.in) ، کلاس Scanner
- لازم است بدانید که کلاس Scanner هشت سازنده با پارامتر های متفاوت دارد که پارامتر سازنده ای که برای خواندن از کیبورد استفاده می شود از نوع InputStream می باشد. یعنی System.in از نوع InputStream است، اما زیاد به این جزئیات توجه نمی کنیم و سرمون بکار خودمون تنها برنامه نویسی مون رو میکنیم!!!! 😊
- حالا با شی input می توانیم به متد های مورد نظر خود برای ورودی دسترسی پیدا کنیم.

مثال: فرض کنید می خواهیم مثل همان مثال اول مقدار عددی صحیح به دو متغیر خود بدهیم و مجموع آن دو رو چاپ کنیم با این تفاوت که مقدار دو متغیر عددی صحیح از کاربر درخواست شود که با کیبورد وارد کند!!!

```
package iran;

import java.util.Scanner;

public class Sum {

    public static void main(String[] args) {
        int a, b;
        Scanner input = new Scanner(System.in);
        System.out.println("pleas Enter number a :");
        a = input.nextInt();
        System.out.println("pleas Enter number b :");
        b = input.nextInt();
        int sum = 0;
        sum = a + b;
        System.out.println(a + " + " + b + " = " + sum);
    }
}
```

خروجی : در کنسول جاوا از کاربر درخواست می شود که به دو متغیر **a** و **b** مقدار بدهد تست این برنامه بصورت زیر است:

```
pleas Enter number a :
10
pleas Enter number b :
50
10 + 50 = 60
```

```
package iran;
```

◀ پکیجی که کلاس ما در آن قرار دارد.

```
import java.util.Scanner;
```

◀ Import کردن پکیج `java.util` برای دسترسی به کلاس `Scanner` و دستورات درونش.

```
public class Sum {
```

◀ تعریف کلاس خود به نام `Sum` که با حرف بزرگ شروع شده است.


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

برای اجرای دستورات خود نیاز به متد main داریم

```
int a, b;
```

تعریف دو متغیر از نوع عدد صحیح

```
Scanner input = new Scanner(System.in);
```

ساختن شی از کلاس Scanner

```
System.out.println("pleas Enter number a :");
```

چاپ پیامی در کنسول به کاربر برای این مقداری را با کیبورد وارد کند.

```
a = input.nextInt();
```

یکی از متد های دریافت ورودی در کلاس Scanner، nextInt() می باشد که با شی (input) ساخته شده از این کلاس به آن اشاره و دسترسی پیدا میکنیم. و مقداری که از کیبورد دریافت میکند میریزه داخل متغیر a

```
System.out.println("pleas Enter number b :");
b = input.nextInt();
```

این دو دستور نیز مشابه دستور قبل می باشد.

```
int sum = 0;
sum = a + b;
```

تعریف متغیری از نوع صحیح به نام sum که مقدارش صفر است و حاصل جمع دو متغیر a,b را داخل sum می ریزیم.

```
System.out.println(a + " + " + b + " = " + sum);
```

در پایان حاصل جمع دو متغیر در خروجی چاپ می شود.

Tip! برای دریافت انواع ورودی از کیبورد نظیر انواع عدد صحیح ، انواع عدد اعشاری ، کاراکتر ، رشته و.... متدهای مربوطه آن ها بصورت زیر است یعنی وقتی که شما یک شی از کلاس Scanner ساختید می توانید از طریق آن شی و کاراکتر نقطه (.) به این متد ها اشاره و دسترسی پیدا کنید: (البته ما به متد های معروف و پر کاربرد این کلاس می پردازیم در صورت نیاز خود که خارج از این متد ها می باشد می توانید سرچ بزنید!!! چون اگر ما بخواهیم به تمامی دستورات و متدهای درون یک کلاس تک به تک بپردازیم هم از حوصله آموزش خارج میشه هم خسته کننده و گیج کننده می شود . چون هدف از این آموزش ها تنها ایجاد حرکت در شماسست و یادگیری به اراده و تلاش خودتان بستگی دارد)

- فرض کنید شی ساخته شده از کلاس Scanner ، input نام دارد.

```
Scanner input = new Scanner(System.in);
```

int nextInt()

نکته: این متد نوعش int می باشد و مقدار عدد صحیح int برمی گرداند برخلاف متد های از نوع void که مقدار برنمی گردانند. می توان به سایر متدهای زیر که نوعش فرق میکند تعمیم داد.

بخشی از متد های پر کاربرد کلاس Scanner

متدها همراه با توصیف	ردیف
<p>int nextInt()</p> <pre>int a=input.nextInt();</pre> <p>این متد برای گرفتن ورودی از نوع عدد صحیح int می باشد</p>	۱
<p>long nextLong()</p> <pre>long a=input.nextLong();</pre> <p>این متد برای گرفتن ورودی از نوع عدد صحیح long می باشد</p>	۲
<p>short nextShort()</p> <pre>short a=input.nextShort();</pre> <p>این متد برای گرفتن ورودی از نوع عدد صحیح short می باشد</p>	۳
<p>float nextFloat()</p> <pre>float a=input.nextFloat();</pre> <p>این متد برای گرفتن ورودی از نوع عدد اعشاری short می باشد</p>	۴
<p>double nextDouble()</p> <pre>double a=input.nextDouble();</pre> <p>این متد برای گرفتن ورودی از نوع عدد اعشاری double می باشد</p>	۵
<p>byte nextByte()</p> <pre>byte a=input.nextByte();</pre> <p>این متد برای گرفتن ورودی از نوع عدد صحیح byte می باشد</p>	۶
<p>boolean nextBoolean()</p> <pre>boolean a=input.Boolean();</pre> <p>این متد برای گرفتن ورودی از نوع مقدار boolean می باشد</p>	۷

String a=input.next();	String next()	۸
char a=input.next().charAt(0);	next().charAt(0)	۹

این متد برای گرفتن ورودی از نوع رشته (String) می باشد

این متد برای گرفتن ورودی از نوع کاراکتر (char) می باشد

لازم است بدونید در ورودی گرفتن و ریختن مقدار به درون یک متغیر استفاده از نوع متد ورودی بستگی به نوع متغیری دارد که قراره داخلش مقدار بریزید مثلا اگر متغیر شما از نوع `int` هست باید از متد نوع `int` یا اگر متغیر از نوع رشته (`String`) باشد باید متد ورودی از نوع `String` باشد و...



با پکیج ها و روش ورودی گرفتن با کیبورد و کلاس `Scanner` و متد های پر کاربرد آن آشنا شدید حالا برای تمرین بیشتر مثال زیر را برای خودتون کار کنید و با ایجاد تغییرات در آن خود را به چالش بکشید:

در مثال زیر از انواع ورودی گرفتن از کیبورد بکار گرفته شده است:

```
package iran;
import java.util.Scanner;

public class Sum {
    public static void main(String[] args) {
        int b;
        char ch;
        String str;
        boolean bln;
        float flt;
        double dbl;
        long lng;
        short sht;
        Scanner input = new Scanner(System.in);
        System.out.println("please enter value of int");
        b = input.nextInt();
        System.out.println(b + " printing");
        System.out.println("please enter value of char");
        ch = input.next().charAt(0);
        System.out.println(ch + " printing");
        System.out.println("please enter value of String");
        str = input.next();
    }
}
```

```
System.out.println(str + " printing");
System.out.println("please enter value of boolean");
bln = input.nextBoolean();
System.out.println(bln + " printing");
System.out.println("please enter value of float");
flt = input.nextFloat();
System.out.println(flt + " printing");
System.out.println("please enter value of short");
sht = input.nextShort();
System.out.println(sht + " printing");
System.out.println("please enter value of double");
dbl = input.nextDouble();
System.out.println(dbl + " printing");

}

}
```

خروجی: درخواست به کاربر برای وارد کردن مقدار، نمونه تست انجام شده این برنامه که هر بار با گرفتن مقدار پشت سر آن مقدار چاپ می شود.

```
please enter value of int
245454
245454 printing
please enter value of char
@
@ printing
please enter value of String
bjhghjfk546498@%&$*%#*((
bjhghjfk546498@%&$*%#*(( printing
please enter value of boolean
false
false printing
please enter value of float
1.005
1.005 printing
please enter value of short
125
125 printing
please enter value of double
555.00009
555.00009 printing
```



با پکیج های آماده جاوا و روش استفاده از آن ها و ورودی گرفتن از کیبورد آشنا شدیم این جلسات رو گام به گام یاد بگیرید چون هرچی بریم جلو مقدمات و پایه جاوا رو که یاد گرفتید مثال های جامع تر و بزرگی تر که کلیه این مفاهیم را در بر دارد را بررسی خواهیم کرد. با تمرین کردن خود را به چالش بکشید منم هستم سوالی داشتید پاسخ گو ام 😊



شما در این آموزش نسخه JAVA SE رو آموزش می بینید نسخه ای که پیشنهاد برنامه نویسی اندروید و JAVA EE و حتی با یاد گیریش ، یاد گیری سایر زبان های برنامه نویسی کار سختی نیست چون ساختار و اصول زبان های برنامه نویسی شبیه به هم می باشد.



در مورد گرافیک و کار فریم ها و باتن ها و لیبل ها و پنل ها و انواع دکمه ها و..... بله جاوا داره مفاهیم پایه ای که یاد گرفتیم سراغ گرافیک و برنامه نویسی نرم افزار های تحت ویندوز رو هم خواهیم داشت.



بازار کار جاوا: شما با یاد گیری جاوا SE خود را آماده یادگیری اندروید یا جاوا EE خواهید کرد که بازار کار خوبی دارند.



در مورد برنامه نویسی جاوا و در کل همه زبان های برنامه نویسی باید صبر و تلاش فراوان داشت برای یادگیری.



این را بدانید که همه مفاهیم جاوا را در قالب آموزش نمی توان بیان کرد چون خسته کننده و از حوصله خارج می شود اما مفاهیم اصلی و مهم که شما را به حرکت وا می دارد و باعث شود قدرت تشخیص نیاز های خود برای حل یک مسئله رو پیدا کنید و نیاز های خود را برطرف کنید آموزش داده می شود.



تنها به مطالعه تئوری اکتفا نکنید حتما عملا کد بزنید خودتون رو به زحمت بیاندازید تلاش کنید و با تمرین کردن خودتون رو به چالش دعوت کنید. در ضمن سرچ زدن برای رفع مشکل خود را فراموش نکنید!!!!!!

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 از کانال

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.