آموزش زبان برنامه نویسی جاوا

عملگرهای پایه‌ای جاوا

جلسه هفتم

نویسنده : رحمان زارعی

جاوا را ساده،آسان و شیرین بنوشید!!!!

جاوا مجموعه ای از عملگرهای ها رو برای دستکاری متغیر ها فراهم کرده است.

عملگرهای های پایه ای در جاوا به بخش های زیر تقسیم می شود:

- عملگرهای محاسباتی(Arithmetic Operators)
- عملگرهای رابطه ای(Relational Operators)
- عملگرهای منطقی(Logical Operators)
- عملگرهای وظیفه ای(Assignment Operators)

عملگرهای محاسباتی(Arithmetic Operators):

عملگرهای محاسباتی همان عملگرهایی هستند که در محاسبات جبری ریاضی نظیر تفریق ،ضرب،جمع،تقسیم و....استفاده می شوند.

پایین در جدول(۱) به این عملگرها همراه با مثال می پردازیم:

فرض کنید متغیر صحیح A دارای مقدار ۱۰و متغیر صحیح B دارای مقدار ۲۰ می باشد

```
A=10;
B=20;
```

- به +،*،/،-،%،^و.... عملگر می گویند.
- به A,B,c,f,g,e,R و....در کل به متغیر ها عملوند می گویند.

خیلی سخت نگیرید چیز خاصی نیست تنها برای نام بردنشون از این اصلاحات استفاده میکنیم که منظورمون رو برسونیم.

شماره	معرفی عملگر محاسباتی همراه با مثال
۱	+ (جمع) مثال: A+B می دهد ۳۰
۲	- (تفریق) مثال: A-B می دهد 10-
۳	* (ضرب) مثال: A*B می دهد ۲۰۰
۴	/ (تقسیم) مثال:B/A می دهد ۲
۵	% (باقیمانده) مثال:B%A می دهد ۰
۶	++ (افزایش) به مقدار موجود عملوند(متغیر) ۱ اضاف می کند.(عملوند+۱=عملوند) مثال:B++ می دهد ۲۱ B++ معادل ;B=B+1 می باشد.
۷	-- (کاهش) از مقدار موجود عملوند(متغیر) ۱ کم می کند.(1-عملوند=عملوند) مثال:B-- می دهد 19 B-- معادل ;B=B-1 می باشد.

جدول(۱)

عملگرهای رابطه ای(Relational Operators):

پایین در جدول(۲) به معرفی این عملگر همراه با مثال پرداخته شده است.

فرض کنید متغیر صحیح A دارای مقدار ۱۰و متغیر صحیح B دارای مقدار ۲۰ می باشد.

A=10;

B=20;

شماره	معرفی عملگر رابطه ای همراه با مثال
۱	== (برابر) برای برسی مقدار دو عملوند(متغیر) که آیا مقدارشون برابر هست یا خیر. اگر دو متغیر برابر باشد شرط درست(true) و اگه برابر نبودن شرط نادرست(false) می باشد. مثال: (A==B) نادرست(false)می باشد.
۲	=! (برابر نیست یا نابرابری) مقدار دو متغیر (عملوند) رو از نظر نابرابری بررسی میکند . اگر دو متغیر نابرابر باشند شرط درست(true) و اگر برابر باشند شرط نادرست(false) می باشد. مثال: (A!=B) درست(true) می باشد.
۳	> (بزرگتر از) اگر مقدار متغیر سمت چپ از مقدار متغیر سمت راست بزرگ تر باشد شرط درست(true)و برقرار می باشد. مثال: (A>B) نادرست(false) می باشد.
۴	< (کمتر از) اگر مقدار متغیر سمت چپ از مقدار متغیر سمت راست کوچک تر باشد شرط درست(true)و برقرار می باشد مثال: (A<B) درست(true) می باشد.
۵	=>(بزرگ تر مساوی از) اگر مقدار متغیر سمت چپ از مقدار متغیر سمت راست بزرگ تر یا مساوی باشد شرط درست(true)و برقرار می باشد. مثال: (A>=B) نادرست(false) می باشد

ادامه در صفحه بعد...

جدول(۲)

۶	=>کوچکتر مساوی از) اگر مقدار متغیر سمت چپ از مقدار متغیر سمت راست کوچک تر یا مساوی باشد شرط درست(true)و برقرار می باشد. مثال: (A<=B) درست(true) می باشد

جدول(۲)

عملگرهای منطقی(Logical Operators):

پایین در جدول(۳) به معرفی این عملگرها همراه با مثال پرداخته شده است.

فرض کنید متغیر بولی(boolean) A دارای مقدار true و متغیر بولی(boolean) B دارای مقدار false می باشد.

```
boolean A=true;
boolean B=false;
```

شماره	معرفی عملگر منطقی همراه با مثال
۱	&& (و)(and) اگر مقدار هر دو عملوند(متغیر) درست(true) باشند شرط درست(true) می باشد. مثال: (A && B) نادرست(false)می باشد.
۲	|| (یا)(or) اگر حداقل مقدار یکی از دو متغیر درست(true)باشد شرط درست(true) می باشد. مثال: (A || B) درست(true) می باشد.
۳	! (نقیض)(not) مقداری منطقی که متغیر دارد رو برعکس(نقضش)می کند. اگر مقدار یک متغیر درست(true) باشد بعد از اعمال ! مقدارش عکس و نادرست(false)می شود. مثال: (!A) درست(true) می باشد. مثال: (!A && B) درست(true) می باشد.

جدول(۳)

عملگرهای وظیفه ای(Assignment Operators):

پایین در جدول(۴) به معرفی این عملگرها همراه با مثال پرداخته شده است.

شماره	معرفی عملگر وظیفه ای همراه با مثال
۱	= اختصاص(نسبت) دادن مقدارهای سمت راست عملگر مساوی به سمت چپ عملگر مساوی مثال:C = A + B مقدار A + B ریخته (اختصاص داده)(نسبت داده) می شود به متغیر C
۲	+= مقدار متغیر سمت چپ مساوی را با مقدار متغیر سمت راست مساوی جمع بزن بعد نتیجه را به متغیر سمت چپ مساوی اختصاص(نسبت) بده. مثال: C += A معادل C = C + A
۳	-= مقدار متغیر سمت چپ مساوی را از مقدار متغیر سمت راست مساوی کم کن بعد نتیجه را به متغیر سمت چپ مساوی نسبت (اختصاص) بده. مثال: C -= A معادل C = C - A
۴	*= مقدار متغیر سمت چپ مساوی را در مقدار متغیر سمت راست مساوی ضرب کن بعد نتیجه را به متغیر سمت چپ مساوی نسبت (اختصاص) بده مثال: C *= A معادل C = C * A
۵	/= مقدار متغیر سمت چپ مساوی را بر مقدار متغیر سمت راست مساوی تقسیم کن بعد نتیجه را به متغیر سمت چپ مساوی نسبت (اختصاص) بده مثال: C /= A معادل C = C / A
۶	%= باقیمانده تقسیم مقدار متغیر سمت چپ مساوی را بر مقدار متغیر سمت راست مساوی به دست بیاور بعد نتیجه را به متغیر سمت چپ مساوی نسبت (اختصاص) بده مثال: C %= A معادل C = C % A

جدول(۴)

اولیت عملگر ها در جاوا:

عملگر ها برای تاثیر گذاشتن بر متغیر نسبت به هم اولیت هایی دارند.

به مثال زیر توجه کنید:

```
x = 7 + 3 * 2
```

بنظر شما جواب x چند میشه؟!!!۲۰ نه اشتباه است!!!!

در اینجا اگر ترتیب اولیت های عملگر ها رو رعایت کنیم جواب ۱۳ می شود!!!!!!!

عملگر * (ضرب) اولیت بیشتری نسبت به عملگر + (جمع) دارد!!!!!

پس ابتدا ضرب ۲*۳ انجام می شود و سپس نتیجه اش با ۷ جمع می شود.

در برنامه نویسی به همین صورت به انواع عملگر ها با توجه به اولیتی که دارند ترتیب اثر داده می شود.

در جدول(۵) در زیر اولیت انواع عملگرها نسبت به هم مشخص شده اند.

- در جدول زیر اولیت ها از بالا به پایین کمتر می شود یعنی عملگر هایی که در ابتدای سطر جدول هستند بالاترین اولیت و عملگر هایی که آخر سطر جدول هستند کمترین اولیت رو نسبت به هم دارند.

نکته: ما هنوز عملگر های دیگه ای هم داریم اما در این آموزش سعی کردیم عملگر هایی که بیشترین کاربرد را دارد آموزش بدیم و از مطالب اضافی و خسته کننده برهیز شده است.برای اطلاعات بیشتر می توانید سرچ بزنید!!!!!!

دسته بندی	عملگر	اولیت
پسوند (Postfix)	() [] . (عملگر نقطه)	از راست به چپ
ضربی (Multiplicative)	* / %	از چپ به راست
افزودنی (Additive)	+ -	از چپ به راست
رابطه ای (Relational)	> >= < <=	از چپ به راست
برابری (Equality)	== !=	از چپ به راست
و (and)	&&	از چپ به راست
یا (or)	||	از چپ به راست
وظیفه ای (Assignment)	= += -= *= /= %=	از راست به چپ

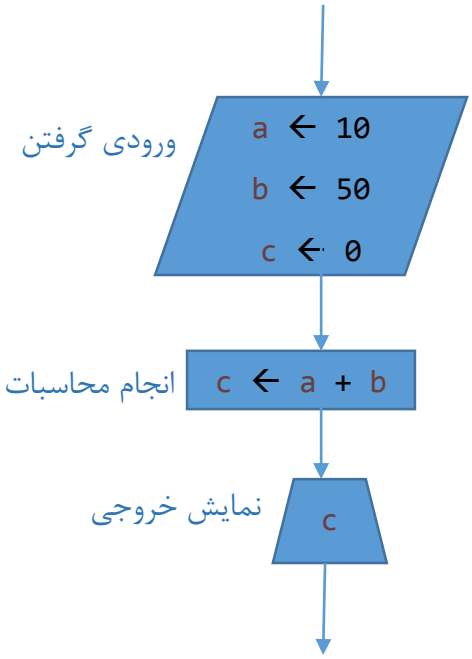
جدول(۵)

خب از مقدمه چینی و توضیح مفاهیم گذشتیم رسیدیم به ایستگاه مثال!!!!!

اول قصد داشتم مستقیم برم سراغ مثال و بین مثال ها به مفاهیم بپردازیم!!! اما گفتم اول ۰ تا ۱۰۰ مفاهیم مربوط به این جلسه رو بگم بعد مثال کار کنیم اگه نکته ای هم جا مونده باشه تو مثال ها با هم بررسیش می کنیم.

مثال۱:

```
package iran;

public class Example2 {

	public static void main(String[] args) {

		int a = 10;
		int b = 50;
		int c = 0;
		c = a + b;
		System.out.println("c=" + c);

	}
}
```



فلوچارت مثال بالا!!!

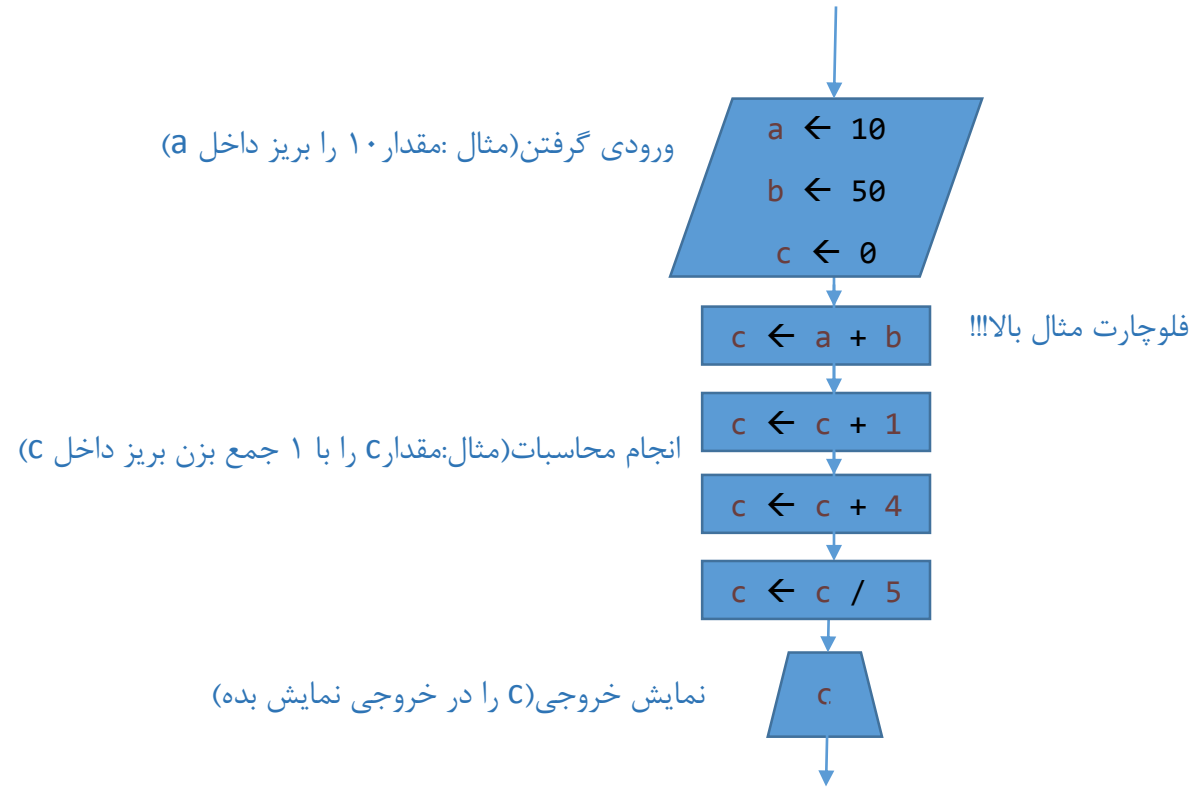
خروجی:

```
c=60
```

در مثال بالا با استفاده از عملگر جمع(+) حاصل جمع مقدار دو متغیر a و b داخل متغیر c ریخته شد و مقدار متغیر c از 0 به 60 تغییر یافت و با دستور چاپ در خروجی کنسول نمایش داده شد.

مثال۲:

```
package iran;

public class Example2 {

	public static void main(String[] args) {

		int a = 10;
		int b = 50;
		int c = 0;
		c = a + b;
		c++;
		c+=4;
		c/=5;
		System.out.println("c=" + c);

	}
}
```



خروجی:

```
c=13
```

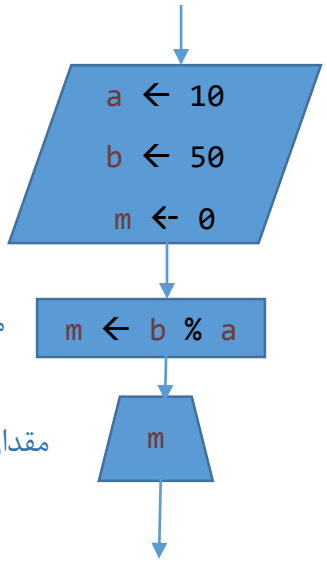
مثال۳:

```
package iran;

public class Example2 {

	public static void main(String[] args) {

		int a = 10;
		int b = 50;
	int m=0;
	m=b%a;
		System.out.println("m=" + m);

	}
}
```

در ورودی مقادیر به متغیر ها نسبت داده می شود

محاسبات:نتیجه باقیمانده تقسیم متغیر b بر a را بریز داخل متغیر m

مقدار متغیر m رو در خروجی نمایش بده



فلوچارت مثال بالا!!!!

خروجی:

```
m=0
```

یک مثال خوب همراه با توضیح!!!!!!!

```
1. package iran;

2. public class Example2 {

3. public static void main(String args[]) {
4. int a = 10;
5. int b = 20;
6. int c = 25;
7. int d = 25;
8. System.out.println("a + b = " + (a + b));
9. System.out.println("a - b = " + (a - b));
10.  System.out.println("a * b = " + (a * b));
11.  System.out.println("b / a = " + (b / a));
12.  System.out.println("b % a = " + (b % a));
13.  System.out.println("c % a = " + (c % a));
14.  System.out.println("a++   = " + (a++));
15.  System.out.println("a--   = " + (a--));
16.  System.out.println("d++   = " + (d++));
17.  System.out.println("++d   = " + (++d));
18.  }

19.  }
```

خروجی:

```
a + b = 30
a - b = -10
a * b = 200
b / a = 2
b % a = 0
c % a = 5
a++   = 10
a--   = 11
d++   = 25
++d   = 27
```

- در زیر توضیحات مربوط به این کد آمده است:

خطوط ۴ تا ۷:

```
int a = 10;
int b = 20;
int c = 25;
int d = 25;
```

- تعریف و مقدار دهی مستقیم به متغیر هایی که از نوع int می باشد.

خطوط ۸ تا ۱۳:

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

- در این کد یک رشته که بین دو "" آمده است و به حاصل جمع دو متغیر متصل می شود.
- ما می توانید محاسبات بر روی دو متغیر رو مستقیم چاپ کنیم برای این کار محاسبات رو مستقیم درون دستور زیر قرار می دهیم:

```
System.out.println(a+b)
```

اگر در کنار محاسبات از رشته(String) استفاده کردیم ،برای این که محاسبات صحیح انجام شود باید محاسبات رو درون دو پرانتز قرار داد: (در مثال بعد به چرا؟؟؟؟؟؟؟؟!!!!!!!!و به چه دلیل؟؟؟؟؟؟؟؟؟؟!!!!!!!!!! این مسئله می پردازیم!!!!!)

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

پس دستور زیر:

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

معادل دستور پایین می باشد:

```
int out=a+b;

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

بقیه دستورات به همین شکل هست و عملگرد محاسبه تغییر کرده است تا این که به خط زیر می رسیم:

خط ۱۴:

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

- اینم قالب و محاسبات مستقیم مثل خطوط قبل است فقط شکل محاسبه تفاوت دارد
- همان طور که در این جلسه توضیح داده شده a++ معادل a=a+1; می باشد تفاوتی از لحاظ مقدار نمی کند.
- فقط یک سوال؟!!!! فرض کنید مقدار a=10; می باشد،مقدار خروجی دستور زیر چیست؟؟!!

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

خب در نگاه اول میگیم چون++a یعنی a=a+1 و مقدار a=10 و با این محاسبه مقدار a می شود ۱۱ پس در خروجی باید ۱۱ چاپ شود!!! اما جواب خیر است!!!!!!برای یافتن جواب به نکته زیر توجه کنید:

نکته:

در دستور a++ چه اتفاقی می افتد؟!!!!

نگاه اول:تکرار میکنم !!!!! که a++ معادل a=a+1 می باشد.پس مقدار a که ۱۰ بود باید بشود ۱۱!!!

نگاه دوم:**اولیت** کار دست ما میده!!!!در دستور زیر:

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

ابتدا اولیت با چاپ مقدار متغیر a هست و سپس عملیات محاسباتی ++ اعمال می شود یعنی ابتدا:

مقدار a که ۱۰ می باشد چاپ می شودو سپس یکی به مقدار a اضاف و مقدارش می شود ۱۱

پس در خروجی مقدار اولیه a که ۱۰ می باشد چاپ می شود و مقدار نهایی a به ۱۱ تغییر پیدا میکند.

خط۱۵:

حال دستور خط ۱۵ که بصورت زیر است رو بررسی میکنیم:

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

- خب شکل و فرم و قالب همه چیزش مثل خطوط قبل می باشد.
- مقدار a که ۱۰ بود با a++ به ۱۱ مقدارش تغییر میکند حال باز شبیه دستور خط قبل با این تفاوت که این مقدار کاهشی هست ابتدا مقدار a که ۱۱ هست چاپ می شود و سپس یکی از مقدار a کاسته و به ۱۰ تغییر مقدار پیدا میکند.

خط۱۶:

خط ۱۶ هم شبیه خط ۱۴ هست.

خط۱۷:

این خط نسبت به خطوط قبل یک تفاوتی داره و اینه که ابتدا ++ (به اصطلاح پلاس پلاس)شده و سپس مقدارش نمایش داده شده با این تفاسیر مقدار d ابتدا ۲۵ بوده و خط ۱۶ مقدارش به ۲۶ تغییر می کند حال در خط ۱۷ ابتدا یکی به مقدار d اضافه می شود و مقدار d به ۲۷ تغییر پیدا میکند و سپس مقدارش چاپ می شود.

```
System.out.println("++d    = " + (++d));
```

نکته :تفاوت ++d و d++ بصورت زیر است:

++d: ابتدا مقدار متغیر d و سپس یکی به مقدار متغیر d اضافه کن.

d++:ابتدا یکی به مقدار متغیر d اضافه کن و سپس مقدار متغیر d

در پایان می توانید کد زیر رو برای خود کپی و در محیط Eclipse تستش کنید و با ایجاد تغییرات در کد،خود را به چالش بکشید.

```
package iran;

public class Example2 {

     public static void main(String args[]) {
          int a = 10;
          int b = 20;
          int c = 25;
          int d = 25;
          int out = a + b;
          System.out.println("a + b = " + (a + b));
          System.out.println("a - b = " + (a - b));
          System.out.println("a * b = " + (a * b));
          System.out.println("b / a = " + (b / a));
          System.out.println("b % a = " + (b % a));
          System.out.println("c % a = " + (c % a));
          System.out.println("a++   = " + (a++));
          System.out.println("a--   = " + (a--));
          System.out.println("d++   = " + (d++));
          System.out.println("++d   = " + (++d));
     }
}
```

مثال آخر.....

یادتون باشه گفتم در یک مثال به تفاوت چاپ محاسبه مستقیم مقدار ،مقدار با رشته(String)،رشته (String) می پردازم حالا وقتش رسید!!!!!

۱. چاپ محاسبه مستقیم دو متغیر :

- **حاصل جمع a+b چاپ می شود**

```
package iran;

public class Test {

	public static void main(String[] args) {
		int a = 5;
		int b = 20;
		System.out.println(a + b);

	}
}
```

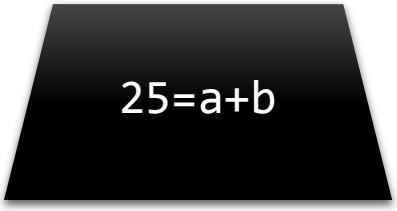
خروجی:

۲. چاپ ابتدا محاسبه دو متغیر و سپس رشته(String) :

- **حاصل جمع a+b محاسبه و به رشته "a+b=" وصل و سپس چاپ می شوند**

```
package iran;

public class Test {

	public static void main(String[] args) {
		int a = 5;
		int b = 20;
		System.out.println(a + b + "=a+b");

	}

}
```

خروجی:

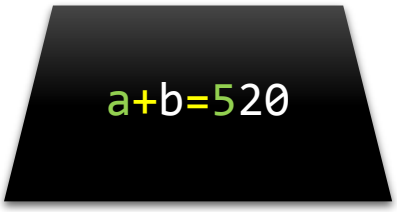


۳.چاپ ابتدا رشته(String) و سپس محاسبه دو متغیر:

بگذارید بعد از دیدن مثال براتون توضیح بدم!!!!

```
package iran;

public class Test {

    public static void main(String[] args) {
        int a = 5;
        int b = 20;
        System.out.println("a+b="+ a + b);

    }
}
```

خروجی:



بر خلاف تصور حاصل جمع a+b که باید ۲۵ می شد اتفاق نیفتاد!!!!! و تنها تک به تک مقدار بعد از رشته(String) چاپ شدند نه نتیجه محاسبه!!!!

استثناً

وقتی بعد از رشته(String) محاسبات به صورت عملگر جمع (+)صورت میگیرد بعد از چاپ رشته(String)

علامت جمع (+)بین متغیر ها به عنوان یک اتصال دهنده خوانده می شود و تنها مقدار متغیر ها رو به هم وصل می کندو عملیات محاسباتی جبری صورت نمی گیرد این استثنا برای بعضی از عملگرها نظیر جمع (+) یا حتی برای تفریق(-) خطا کامپایل میده و....دیده می شود!!!!!

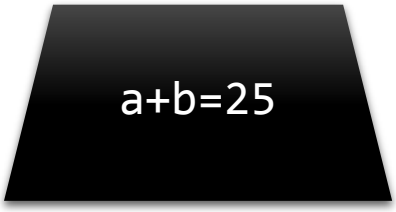
برای رفع این استثنا:

تنها کافی است که محاسبات جبری خود رو درون یک پرانتز کنار یک رشته پیاده سازی کنیم!!!

مثل مثال زیر:

```
package iran;

public class Test {

	public static void main(String[] args) {
		int a = 5;
		int b = 20;
		System.out.println("a+b=" + (a + b));

	}

}
```

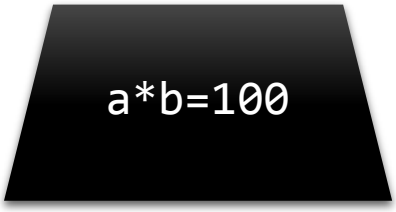
خروجی:



حالا برای این که ثابت کنیم این استثنا تنها به برخی از عملگر ها نظیر جناب جمع (+) و تفریق(-) تعلق دارد و برای بعضی دیگر از عملگر های محاسباتی مشکلی پیش نمیاد به مثال های زیر توجه کنید:

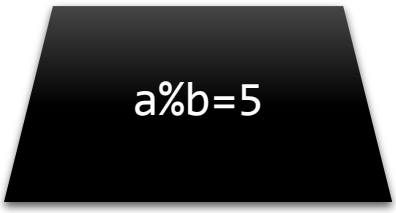
```java
package iran;

public class Test {

	public static void main(String[] args) {
		int a = 5;
		int b = 20;
		System.out.println("a*b=" + a*b);

	}

}
```

خروجی:




```java
package iran;

public class Test {

	public static void main(String[] args) {
		int a = 5;
		int b = 20;
		System.out.println("a%b=" + a%b);

	}

}
```

خروجی:



اگر توجه کرده باشید محاسبات ضرب و باقیمانده بدون پرانتز انجام دادیم.

حالا به مثال تفریق توجه کنید (محاسبات تفریق در کنار رشته(String) هم باید درون پرانتز قرار گیرد)

```java
package iran;

public class Test {

	public static void main(String[] args) {
		int a = 5;
		int b = 20;
		System.out.println("a-b=" + (a-b));
	}
}
```

خروجی:

```
a-b=-15
```

حال به آخرین مثال توجه کنید:

```java
package iran;

public class Test {

	public static void main(String[] args) {
		int a = 5;
		int b = 20;
		System.out.println(b + "/" + a + "=" + b / a);
	}
}
```

خروجی:

```
20/5=4
```

- برای تقسیم هم بدون پرانتز محاسباتش مشکلی نداشت،اما سری که درد نمیکنه دستمال نمی بندند خواهشا هنگامی که محاسباتی داشتید در کنار یک رشته درون پرانتز قرارش بدید که مشکلی پیش نیاید ☺

مفاهیم این جلسه رو خوب تمرین کنید چون در مثال های جلسات بعد از این مفاهیم استفاده می شود.

فقط تمرین کنید، فقط با تمرین هست که تو برنامه نویسی راه می افتید!!!!!!!

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.