

آموزش زبان برنامه نویسی جاوا

دستورات شرطی

جلسه هشتم

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



- قبل از این که به سراغ پیاده سازی دستورات شرطی در محیط کدزنی برویم یک نگاه الگوریتمی و فلوچارتی به این موضوع می کنیم که درک بهتری نسبت به این مبحث داشته باشیم:
به مثال های زیر توجه کنید:
اگر امروز تعطیل باشد، علی به مدرسه نمی رود.
اگر هوا آفتابی باشد، والیبال بازی میکنیم.
اگر هوا صاف باشد، هواپیما پرواز می کند.
اگر دریا طوفانی باشد، کشتی حرکت نمی کند.
دستورات شرطی با اگر شروع می شود و یک شرط رو درون خود دارد که اگر برقرار (true) باشد نتیجه مورد نظر اتفاق می افتد در غیر این صورت عکس نتیجه مورد نظر اتفاق می افتد.
در مثال اگر هوا صاف باشد، هواپیما پرواز می کند
در صورتی هواپیما پرواز میکند که شرط یعنی صاف بودن هوا برقرار (true) باشد. در غیر این صورت اگر هوا عکس صاف بودن یعنی طوفانی (false) باشد هواپیما پرواز نمیکند.
در نگاهی دیگر: اگر هوا صاف باشد، هواپیما پرواز می کند. در صورتی هواپیما پرواز میکند که هوا صاف باشد: شرط صاف بودن هوا برقرار و درست (true) باشد و اگر هوا طوفانی باشد (شرط هوا عکس جمله شرطی صاف بودن یعنی هوا طوفانی است که شرط نادرست (false) می باشد) در نتیجه هواپیما پرواز نمیکند.
می توان جملات شرطی رو به صورت زیر خلاصه کرد:
اگر (شرط برقرار بود) نتیجه
در صورتی نتیجه اجرا می شود که شرط برقرار (درست یا true) باشد.

مثال:

امروز هوا بارانی است.

اگر (امروز هوا آفتابی باشد) علی به باشگاه می رود.

شرط نادرست (false) می باشد و نتیجه اجرا نمی شود یعنی علی به باشگاه نمی رود چون امروز هوا بارانی است و آفتابی نیست.

مثال:

نمره جعفر ۱۱ هست.

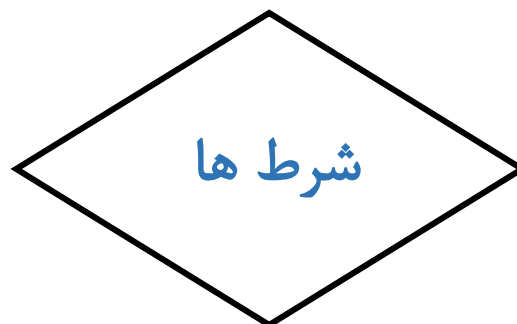
اگر (نمره جعفر بزرگتر یا مساوی ۱۰ باشد) جعفر قبول است.

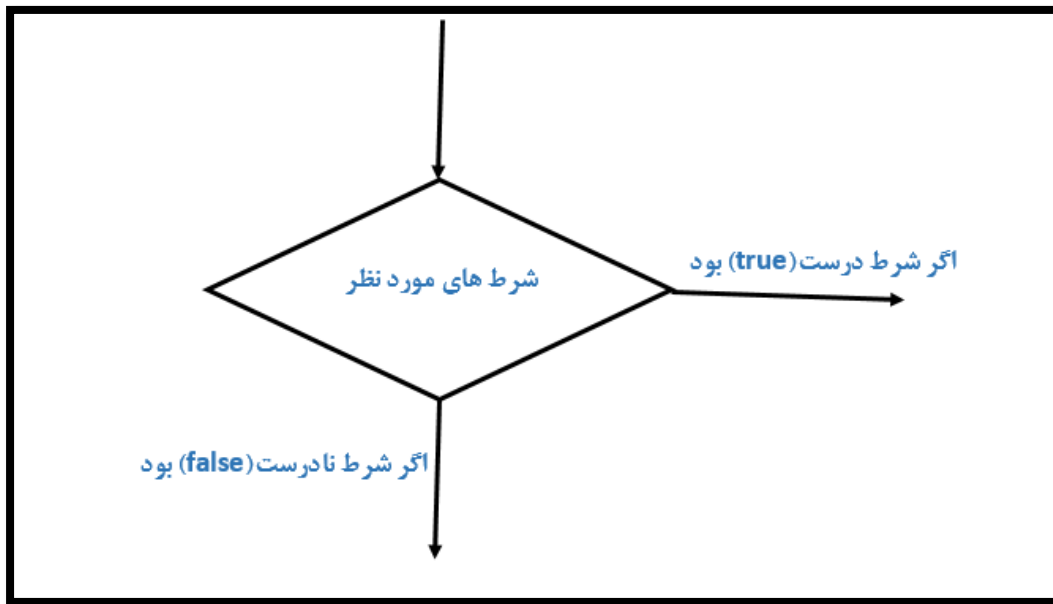
شرط درست (true) می باشد و نتیجه اجرا می شود یعنی جعفر قبول می شود چون نمره جعفر ۱۱ هست و از ۱۰ بیشتر است.

- پس جملات شرطی با یک "اگر" شروع و درون خود یک شرط دارد که اگر درست (true) باشد نتیجه اجرا می شود در غیر این صورت یعنی شرط نادرست (false) و نتیجه اجرا نمی شود.

مبحث فلوچارت عبارت شرطی:

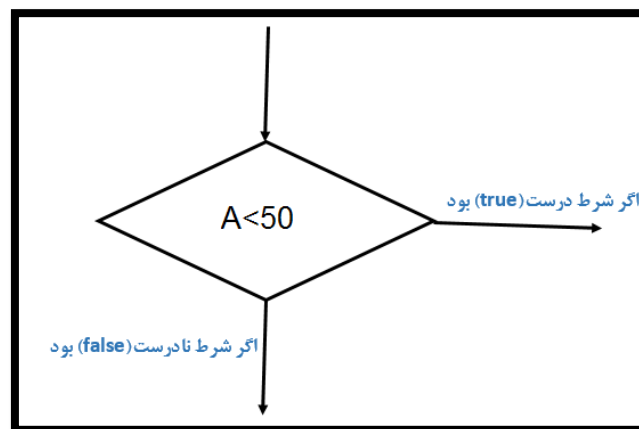
- در فلوچارت علامت لوزی برای دستورات شرطی استفاده می شود.
- همان طور که در مثال های بالا بررسی کردیم انجام عملیات (اجرای نتیجه) منوط به برقرار بودن شرط است.
- دستورات شرطی نوعی حالت تصمیم گیری دارند.
- در زیر به فلوچارت همراه با مثال چگونگی کار عبارت های شرطی پرداخته شده است:





شکل (۱)

- با توجه به شکل (۱) فلوجارت دستورات شرطی مشاهده میکنیم که هر دستور شرطی یک شرط دارد که اگر درست بود یک تصمیمی میگیرد و در صورت نادرست بودن تصمیمی دیگر



شکل (۲)

- در شکل (۲) اگر $A=10$ باشد شرط درست (true) و اگر $A=100$ باشد شرط نادرست (false) می باشد.
- خوب ابتدا برای این ذهنیت و درک بهتری نسبت به دستورات شرطی داشته باشیم از الگوریتم و فلوجارت استفاده کردیم حال بریم به سراغ پیاده سازی دستورات شرطی در قالب کد نویسی.

دستور شرطی if

شکل این دستور شرطی در برنامه نویسی به صورت زیر است:

```
if(Boolean_expression){
    //Executes when the Boolean expression is true
}
```

- در برنامه نویسی هم مثل حالت جمله فارسی در دستور شرطی از کلمه کلیدی "if" ابتدای دستور شرطی می آوریم.
- جلوی کلمه کلیدی "if" شرط مورد نظر رو درون پرانتز قرار می دهیم.
- بعد از پرانتز دو بلوک {} باز و بسته شده نشان دهنده محدوده اجرای دستورات مورد نظر با توجه به درست (true) بودن شرط می باشد.
- نکته : اگر یک دستور داشتیم نیاز به نوشتن {} نیست و زمانی که دستورات ما بیش از یک دستور باشد الزامی هست که دستورات خود رو درون بلوک {} قرار بدهیم.
- درون پرانتز شرط مورد نظر (Boolean_expression) روش بررسیش بصورت زیر است:
اگر شرط درست بود true بر می گرداند و دستور شرط اجرا می شود
اگر شرط نادرست بود false بر میگرداند و دستور شرط اجرا نمی شود به همین دلیل از Boolean_expression در مثال استفاده شده که بررسی شرط تنها دو جواب منطقی true یا false دارد.
خب از توضیحات بگذریم بریم سراغ مثال مربوط به این سرفصل:
مثال:

```
package iran;

public class Eighth_Session {

    public static void main(String[] args) {

        int a = 10;
        if (a < 50)
            System.out.println("true " + a + "<" + 50);

    }

}
```

خروجی:

true 10<50

```
int a = 10;
    if (a < 50)
        System.out.println("true " + a + "<" + 50);
```

- متغیر **a** از نوع صحیح تعریف شده و بهش مقدار ۱۰ نسبت داده شده است.
- خط بعد یک دستور شرطی داریم که عبارت شرط ما درون پرانتز (**a < 50**) است در صورتی پیام مورد نظر چاپ می شود که مقدار **a** کوچک تر از ۵۰ باشد که با توجه به مقدار **a=10** شرط برقرار (**true**) می باشد و پیام چاپ می شود.
- همون طور که مشاهده میکنید چون یک دستور در صورت درست بودن شرط برای اجرا داریم نیاز به قرار دادن دستور مورد نظر درون {} نیست.

مثال:

```
package iran;

public class Eighth_Session {

    public static void main(String[] args) {

        int a = 10;
        if (a < 50) {
            System.out.println(+a + "<" + 50);
            System.out.println("true ");
        }

    }
}
```

خروجی:

10<50
true

- همون طور که مشاهده میکنید چون بیش از یک دستور در صورت درست بودن شرط برای اجرا داریم نیاز به قرار دادن دستور مورد نظر درون {} می باشد.

مثال:

```

package iran;

public class Eighth_Session {

    public static void main(String[] args) {

        int a = 60;
        if (a <= 50)
            System.out.println(+a + "<" + 50);
        System.out.println("Salam ");

    }

}

```

خروجی:

Salam

- چون شرط نادرست (false) می باشد دستور `System.out.println(+a + "<" + 50);` اجرا نمی شود اما چون از بلوک `{}` استفاده نشده تنها شرط رو یک دستور تاثیر می گذارد و محدودش شما دو دستور نمی شود پس دستور `System.out.println("Salam ");` اجرا می شود.

برای درک بهتر به محدوده بندی مثال های زیر توجه کنید:

```

package iran;

public class Eighth_Session {
    public static void main(String[] args) {
        int a = 60;
        if (a <= 50)
            System.out.println(+a + "<" + 50);

        System.out.println("Salam ");
    }
}

```

- هر بلوک که باز و بسته می شود محدوده اختیارات هر دستور رو مشخص میکند.

```
package iran;
```

```
public class Eighth_Session {
```

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

```
        int a = 60;
```

```
        if (a != 50){
```

```
            System.out.println(+a + "!=" + 50);
```

```
            System.out.println("True ");
```

```
        }
```

```
        System.out.println("salam");
```

```
    }
```

خروجی:

```
60!=50
True
salam
```

- همان طور که مشاهده می کنید چون دستورات اجرایی شرط ما بیش از یک دستور هست درون دو { } بلوک قرار گرفته است. و چون دستور `System.out.println("salam");` خارج از این بلوک { } قرار گرفته بدون هیچ پیش شرطی اجرا می شود.

• دستور شرطی if-else :

با یک مثال شروع می کنیم

مثال : اگر باقیمانده عدد m بر ۲ برابر با صفر بود عدد m زوج می باشد در غیر این صورت عدد m فرد می باشد!!!

در مثال بالا یک جمله شرطی هست زیرا با اگر شروع شده است .

شرط(باقیمانده عدد m بر ۲ برابر صفر باشد) عدد m را زوج اعلام کن

در غیر این صورت(یعنی زمانی که شرط برقرار نباشد) عدد m را فرد اعلام کن.

*پس اگر شرط اول ما برقرار نبود با آوردن عبارت "در غیر این صورت" می گیم که یک کار دیگه برامون انجام بده مثلا اگر عددمون زوج نبود فرد اعلامش کن.

دستور شرطی `if-else` هم همین کار را برای ما انجام می دهد.

if (اگر) شرط برقرار بود این کارو بکن.....
else (در غیر این صورت) زمانی که شرط بالا برقرار نبود این کار رو انجام بده
شکل قالب پیاده سازی if-else دستور بصورت زیر است:

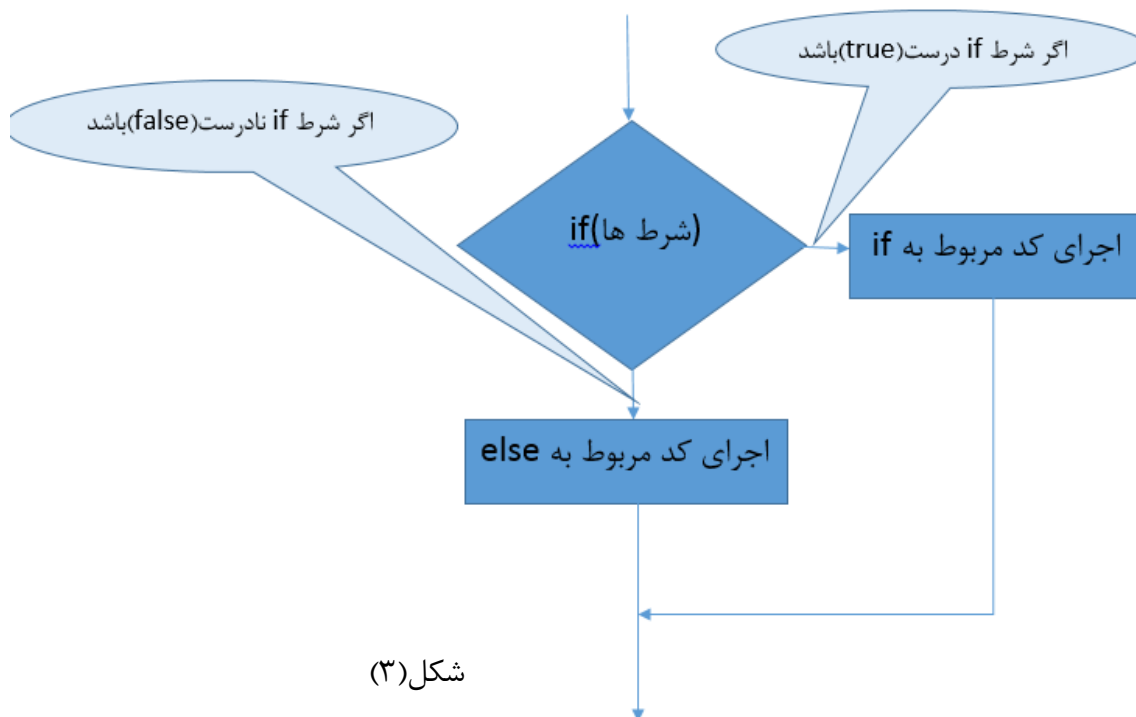
```

if(Boolean_expression){
    //Executes when the Boolean expression is true
    // اگر شرط درون پرانتز درست(true) بود بیا داخل این بلوک {} و دستورات داخلش رو اجرا کن.
} else{
    //Executes when the Boolean expression is false
    // اگر شرط درون پرانتز نادرست(false) بود بیا داخل این بلوک {} و دستورات داخلش رو اجرا کن.
}

```

- **Boolean_expression** همون عبارت شرط درون پرانتز جلوی **if** هست که اگر شرط درست باشه **true** میشه و وارد بلوک **if** می شود، اگر شرط درون پرانتز نادرست باشه **false** می شود و دستورات درون بلوک **if** دیگر اجرا نمی شود و وارد بلوک **else** می شود و دستورات درون آن را اجرا می کند.
- **else** نیز اگر برای اجرا بیش از یک دستور داشته باشد باید دستوراتش درون بلوک {} قرار گیرد.

فلوچارت نحوه کارکرد این دستور رو در شکل (۳) مشاهده میکنید:



شکل (۳)

مثال :

```
package iran;

public class Eighth_Session {

    public static void main(String[] args) {

        int a = 9;

        if (a % 2 == 0)
            System.out.println(a + " zoj mibashad");
        else
            System.out.println(a + " fard mibashad");

    }
}
```

- شرط زوج بودن یک عدد این هست که آن عدد بر ۲ بخش پذیر باشد یعنی باقیمانده عدد بر ۲ برابر صفر شود در غیر این صورت آن عدد فرد می باشد.

```
if (a % 2 == 0)
```

- خوب شرط زوج بودن عدد بررسی شده است اگر شرط برقرار بود پیام زیر را چاپ میکند:

```
System.out.println(a + " zoj mibashad");
```

- اگر شرط برقرار نبود بدون اجرای پیام فوق دستور شرط **else** یعنی در غیر این صورت که پیامش حاوی فرد بودن عدد هست رو چاپ میکند:

```
System.out.println(a + " fard mibashad");
```

مثال:

```
package iran;

public class Eighth_Session {

    public static void main(String[] args) {

        int a = 9;
        int b = 100;
        if (a > b) {
            System.out.print(a);
            System.out.println("=max");
            System.out.print(b);
            System.out.println("=min");
        }

        else {
            System.out.print(b);
            System.out.println("=max");
            System.out.print(a);
            System.out.println("=min");
        }

    }

}
```

- چون دستورات بیش از یک دستور هست درون بلوک {} قرار داده می شود.
- اگر $a > b$ بود، a را max و b را min اعلام کن در غیر این صورت b را max و a را min اعلام کن.

دستور شرطی if...else if...else :

با یک مثال شروع می کنیم:

فرض کنید مقدار متغیر $a=5$ باشد.

اگر $a==5$ بود پیام "A" را چاپ کن.

در غیر این صورت اگر $a < 7$ بود پیام "B" را چاپ کن

در غیر این صورت اگر باقیمانده تقسیم a بر ۲ برابر صفر بود پیام "Zoj" را چاپ کن.

در غیر این صورت پیام "fard bodan" را چاپ کن.

- اگر شرط برقرار نباشد و چند حالت ممکن در شرط ممکن است رخ دهد از این دستورات شرطی استفاده میکنیم.
- در مثال زوج و فرد بودن، شرط ما از دو حالت خارج نیست اگر عدد باقیمانده اش بر ۲ صفر می شد عدد زوج در غیر این صورت عدد فرد است اما گاهی در شرط بیش از دو حالت ممکن احتمال رخ دادنش وجود داشته باشد در این صورت از دستورات شرطی if...else if....else استفاده میکنیم.

مثال :

```
package iran;

public class Eighth_Session {

    public static void main(String[] args) {

        int a = 5;
        if (a == 5)
            System.out.println("A");
        else if (a < 7)
            System.out.println("B");
        else if (a % 2 == 0)
            System.out.println(a + " is zoj");
        else
            System.out.println(a + " is fard");

    }
}
```

خروجی:

A

- شرط اول برقرار بود و پیام "A" چاپ شد.
 - در این دستورات شرطی تنها کد مربوط به یک شرط اجرا می شود. اگر مقدار a را به ۹ تغییر بدهیم خروجی بصورت زیر خواهد شد:
- خروجی:

9 is fard

- با تغییر دادن کد و مقدار متغیر تمرین کنید.

مثال :

```

package iran;

public class Eighth_Session {

    public static void main(String args[]) {
        int x = 30;

        if (x == 10) {
            System.out.print("Value of X is 10");
        } else if (x == 20) {
            System.out.print("Value of X is 20");
        } else if (x == 30) {
            System.out.print("Value of X is 30");
        } else {
            System.out.print("This is else statement");
        }
    }
}

```

خروجی:

Value of X is 30

- پس اگر چند شرط برای یک متغیر وجود داشت از این دستوره‌های شرطی متوالی استفاده میکنیم.
 - در مقایسه برابری مقدار دو متغیر از دو مساوی "==" استفاده میکنیم.
 - می‌توانیم مثل سایر دستورات شرطی کد مربوط به هر شرط رو درون دو بلوک {} قرار دهیم.
- ❖ راه دیگری برای زمانی که چند شرط ممکن داشته باشیم و نخواهیم پشت سر هم از دستور `if..else if..else` استفاده کنیم وجود دارد که بصورت زیر است:

دستور شرطی switch-case :

وقتی چند شرط ممکن داشتیم و تعداد این شرط‌ها زیاد شد مثلاً بیشتر از ۱۰ تا، پیاده‌سازی شرط‌ها با دستورات شرطی `if..else if....else` خسته‌کننده می‌شود `switch-case` جایگزین مناسبی برای این دستورات شرطی می‌باشد تنها بایک تفاوت این که شرط‌ها رو بصورت برابری مقایسه میکنند!!! نگران نباشید با مثال مفهوم روشن می‌شود:

شکل پیاده‌سازی این دستور بصورت زیر است:

```

switch(expression){
case value :
//Statements
break; //optional
case value :
//Statements
break; //optional
//You can have any number of case statements.
default : //Optional
//Statements
}

```

- ❖ عبارت درون پرانتز (expression) جلوی کلمه کلیدی switch مقداری که با سایر گزینه ها مقایسه می شود میگیرد، که گزینه انتخابی ما برای مقایسه با سایر مقادیر می باشد.
- ❖ case های ما هر کدام یک مقدار دارند و یک دستور برای اجرا که پیاده سازی بصورت زیر است:

```

case value :
//Statements

```

- ❖ گزینه انتخابی ما که درون پرانتز (expression) جلوی کلمه کلیدی switch قرار دارد یکی یکی با مقدار case ها مقایسه می شود اگر برابر بودند خیلی هم خوب دستور مربوطه به آن case اجرا می شود.

:break

- ❖ اگر گزینه انتخابی درون پرانتز با مقدار case برابر نبود به سراغ case بعد می رویم اگر برابر بود بعد از اجرای دستور مربوط به آن case دیگر نباید پیش روی کنیم و سراغ case بعد برویم چون جواب گرفتیم و شرط برقرار شده برای این کار از دستور break استفاده می کنیم که وظیفه اش متوقف کردن بررسی سایر case های موجود می باشد و کاری میکنه که از این روند خارج شویم.

:default

- ❖ اگر گزینه انتخابی درون پرانتز با هیچ کدام از case ها برابر نبود بصورت پیش فرض دستورات درونش اجرا می شود. و با break کردن از روند مقایسه خارج می شویم.
- ❖ پیاده سازی default اختیاری هست تنها برای مواردی بکار می رود که احتمال دارد برابر بودن گزینه انتخابی درون پرانتز با هیچ کدام از case ها وجود نداشته باشد.

❖ بعد از پیاده سازی هر دستور **case** باید از **break** استفاده کنیم وگرنه با وجود برابر بودن گزینه انتخابی درون پرانتز با مقدار یک **case** دستورات **case** های بعدی هم اجرا می شود زیرا ما از روند مقایسه خارج نشده ایم. توجه شود گفتیم دستورات بعد از **case** ای که مقدارش با گزینه انتخابی برابر است و دستورات قبل از این **case** حتی با وجود عدم پیاده سازی **break** اجرا نمی شود.

مثال:

```
package iran;

public class Eighth_Session {

    public static void main(String args[]) {

        int a = 2;
        switch (a) {

            case 1:
                System.out.println("1");
                break;
            case 2:
                System.out.println("2");
                break;
            default:
                System.out.println("value not found");
                break;

        }

    }
}
```

خروجی:

2

- گزینه انتخابی ما مقدارش ۲ می باشد.
- گزینه انتخابی ابتدا با مقدار **case** اولی که مقدارش ۱ می باشد مقایسه می شود در صورت عدم برابری به سراغ **case** دومی می رود که مقدارش ۲ می باشد مقایسه می شود مشاهده میکنیم که مقدار گزینه انتخابی با این **case** برابر می باشد در نتیجه دستورات **case 2:** اجرا می شود که حاوی چاپ پیام "۲" هست بعد برای متوقف کردن روند مقایسه **break** می شود.
- اگر مقدار **a** را به ۸ تغییر بدهیم خروجی بصورت زیر خواهد شد:

```
value not found
```

- اگر break رو در تمامی case ها حذف کنیم و a=1 قرار دهیم کد و خروجی آن بصورت زیر خواهد شد:

```
package iran;

public class Eighth_Session {

    public static void main(String args[]) {

        int a = 1;
        switch (a) {

            case 1:
                System.out.println("1");

            case 2:
                System.out.println("2");

            default:
                System.out.println("value not found");

        }

    }

}
```

خروجی:

```
1
2
value not found
```

- به دلیل عدم پیاده سازی break و برابر بودن گزینه انتخابی با case 1 تمامی دستورات این case و case های بعد از آن اجرا شده است.
- مقدار a=2 قرار می دهیم خروجی را بصورت زیر مشاهده خواهیم کرد:

```
2
value not found
```


- همان طور که گفتیم بدلیل نبود break در هر case دستورات case برابر با گزینه انتخابی و case های بعد از آن اجرا می شود دستورات case های قبل اجرا نمی شود.

- میدونید که دستورات switch-case را با if..else if..else هم پیاده سازی کنیم.
- در زیر یک مثال با دو روش متفاوت که دارای یک جواب می باشد پیاده سازی شده است:

switch-case

```
package iran;

public class Eighth_Session {

    public static void main(String args[]) {
        int a = 2;
        switch (a) {

            case 1:
                System.out.println("1");
                break;
            case 2:
                System.out.println("2");
                break;
            default:
                System.out.println("value not fount");
                break;
        }
    }
}
```

if..else if..else

```
package iran;

public class Eighth_Session {

    public static void main(String args[]) {
        int a = 2;
        if (a == 1)
            System.out.println("1");
        if (a == 2)
            System.out.println("2");
        else
            System.out.println("value not fount");
    }
}
```

```
}
```

خروجی هر دو مثال :

2

نکته:

نوع متغیر هایی که در **switch-case** مقایسه می شوند شامل همه متغیر ها نمی شود!!!!!! آنها برخی از متغیر ها رو می توانیم در این ساختار شرطی مقایسه کنیم:

متغیر هایی که در این ساختار قابل مقایسه هستند:

int
short
byte
char

متغیر هایی که در این ساختار قابل مقایسه نیستند به دلیل بزرگی !!!

long
double
float

مثال:

```
package iran;

public class Eighth_Session {

    public static void main(String args[]) {

        char grade = 'C';

        switch (grade) {
            case 'A':
                System.out.println("Excellent!");
                break;
            case 'B':
            case 'C':
                System.out.println("Well done");
                break;
            case 'D':
                System.out.println("You passed");
            case 'F':
```

```

        System.out.println("Better try again");
        break;
    default:
        System.out.println("Invalid grade");
    }
    System.out.println("Your grade is " + grade);
}
}

```

خروجی:

```

Well done
Your grade is C

```

مثال پیشتر.....

```

package iran;

public class Eighth_Session {

    public static void main(String args[]) {

        int a = 8;
        int b = 18;
        if (a == 8 || b == 12)
            System.out.println("ok");

    }
}

```

خروجی:

ok

- ما همزمان می توانید چند شرط درون پرانتز رو بررسی کنیم.
- در این مثال گفته شده اگر مقدار **a** برابر ۸ یا مقدار **b** برابر ۱۲ بود پیام "ok" را چاپ کن.
- روند بررسی شرط در این مثال به صورت زیر بوده است:

(true || false)=true

در جدول های زیر مقایسه های منطقی بین دو متغیر در برنامه نویسی را مشاهده میکنید:

عملگر **and (&&)**: در صورتی **and** دو متغیر **true** هست که مقدار دو متغیر **true** باشد. جدول (۱)

ردیف	مقدار متغیر A	مقدار متغیر B	عملگر منطقی and (و) بین دو متغیر	نتیجه
۱	false	false	&&	false

true	&&	true	true	۲
false	&&	true	false	۳

جدول (۱)

عملگر **or** (||): در صورتی **or** دو متغیر **true** هست که حداقل مقدار یکی از دو متغیر **true** باشد. جدول (۲)

ردیف	مقدار متغیر A	مقدار متغیر B	عملگر منطقی or (یا) بین دو متغیر	نتیجه
۱	false	false		false
۲	true	true		true
۳	false	true		true

جدول (۲)

عملگر **not** (!): مقدار منطقی متغیر را نقض می کند یعنی اگر **true** باشد **false** می کند و بالعکس. جدول (۳)

ردیف	مقدار متغیر A	عملگر منطقی not (!)	نتیجه
۱	true	!	false
۲	false	!	true

جدول (۳)

مثال:

```
package iran;

public class Eighth_Session {

    public static void main(String args[]) {

        int a = 8;
        int b = 18;
        if ((a == 8 || b == 12) && a == 9)
            System.out.println("ok");
        System.out.println((a == 8 || b == 12) && a == 9);
    }
}
```

خروجی:

false

- نتیجه مقایسه شرط بصورت **if((true)&&false)** که نتیجه می دهد **if(false)** و در نتیجه به دلیل نادرست بودن شرط کد مربوط به شرط اجرا نمی شود.
- نتیجه **false** بودن شرط نیز در پایان چاپ شده است.

یک مبحث یا کد برنامه نویسی که اولین بار باهاش برخورد میکنیم برای خود من هم سخت هست!!! چطور آسون میشه؟!!! تنها با در محیط eclipse تست کردن و تغییر در کد ایجاد کردن و تمرین کردن و خود را به چالش کشیدن.

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.