

به نام خدا

ساختارهای if, if-else, if-else if

در این مقاله به سراغ یکی از ساختارهای کنترلی جاوا می رویم که به نوعی از پرکاربردترین آن ها نیز به شمار می رود.

در حقیقت در پایان این مقاله قادر خواهید بود تا کنترل خطوط برنامه را مدیریت کنید.

برای درک بهتر به این مسئله توجه داشته باشید که در برنامه هایی که تاکنون نوشتیم ،

دستورات برنامه به ترتیب و از بالا به پایین اجرا می شدند. ولی قطعا همواره مواقعی پیش

می آید که ما می خواهیم تحت شرایط خاص، برنامه مان به گونه های متفاوتی کار کرده و

دستورات هر موقعیت را بخواهیم که جداگانه اجرا کنیم.

به عنوان مثال فرض کنید که مسئله ی بسیار ساده ای داریم که طی آن قرار است برنامه ی

شما نمره ای را از ورودی بخواند و طبق آن به عنوان یک پیغام در خروجی بنویسد که آیا

این نمره ، نمره ی قبولی است یا نه. آیا با دستوراتی که تا حالا آموخته ایم، می توانیم چنین

برنامه ای را درست کنیم؟

ساختار تصمیم گیری if :

نمای کلی این دستور به شکل زیر است:

```
if (boolean expression) statement;
```

```
or
```

```
if (boolean expression) {
```

statement or statements

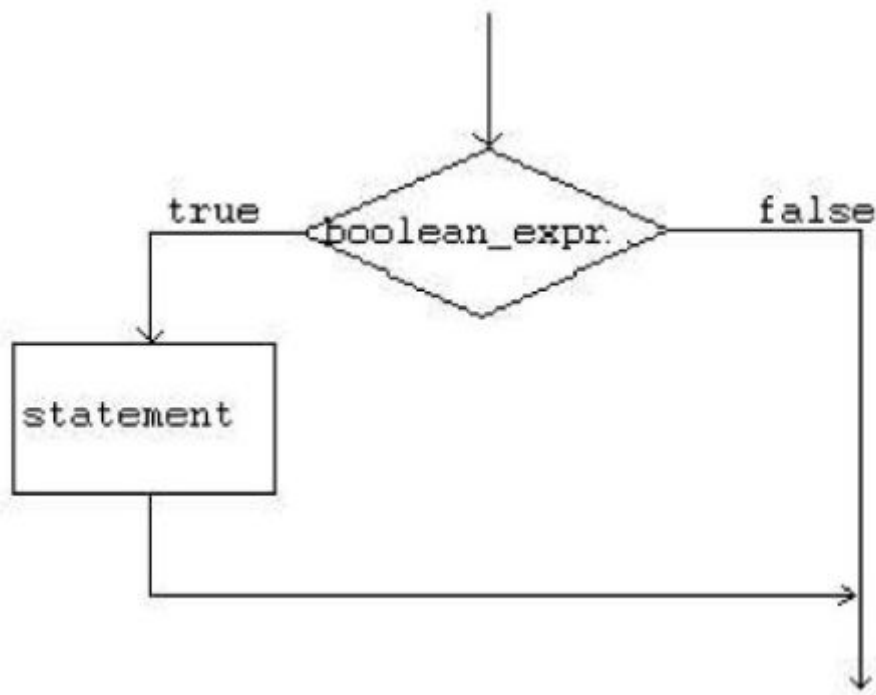
}

در عبارات مذکور، **if** یک کلمه ی کلیدی (**keyword**) است. در بین دو پرانتز یک عبارت بولین قرار می گیرد؛ به عبارت دیگر عبارتی که درون جفت پرانتزها قرار می گیرد، فقط باید به صورت **true** یا **false** ارزیابی شود.

در جایی که عبارت **statement** نوشته شده است، یک دستور از برنامه قرار می گیرد؛ و این دستورات در صورتی انجام می شوند که عبارت بولین درون پرانتزها ، **true** ارزیابی شود. و در غیر این صورت برنامه این بخش را نادیده می گیرد و به سراغ ادامه ی برنامه می رود.

تفاوت دو عبارت فوق در اینست که اگر در صورت **true** ارزیابی شدن عبارت بولین ، فقط می خواهیم که یک دستور اجرا شود، می توانیم دستور خود را در بیرون از یک جفت آکولاد قرار بدهیم ولی اگر تعداد دستورات مورد نظر ، که قرار است در صورت **true** ارزیابی شدن عبارت بولین اجرا شوند، بیش از یک دستور باشد، باید دستورات مورد نظر را درون یک بلوک (جفت آکولاد) از برنامه قرار بدهیم.

فلوچارت زیر می تواند به درک بهتر چگونگی عملکرد ساختار کنترلی **if** کمک کند:



حالا فرض کنیم می خواهیم همان برنامه ای را بنویسیم که در ابتدا به آن اشاره کردیم. برای این کار لازم است ، قطعه کد زیر را به عنوان برنامه ی کوچک خود، مورد استفاده قرار بدهیم:

```

int grade=18;
if (grade<10) System.out.println("You failed");
if (grade>=10) System.out.println("You are passed");
  
```

در برنامه ی مذکور، مقدار متغیر صحیح grade که با مقدار ۱۸ مقدار دهی اولیه شده است، در دستور if اول مورد مقایسه قرار می گیرد. بعد از آن چون عبارت بولین $grade < 10$ به صورت false ارزیابی می شود، عبارت مقابل آن اجرا نشده و کنترل برنامه به سراغ عبارت بعدی می رود. در این بخش چون ساختار تصمیم گیری دیگری وجود دارد، مجدداً یک عبارت بولین دیگر مورد بررسی قرار می گیرد. در این قسمت چون عبارت

grade >= 10 به صورت true ارزیابی می شود، دستور جلوی آن اجرا می شود. بنا براین عبارت You are passed در خروجی چاپ می شود.

حالا فرض کنیم که متغیر grade با عدد صحیح ۷ مقدار دهی اولیه شده باشد، در این صورت چه اتفاقی رخ می دهد؟

اگر متغیر مذکور با عدد صحیح ۲۵ مقدار دهی اولیه شده باشد چطور؟ (فکر نمی کنید برنامه ی خوبی نوشته ایم!) در حقیقت برنامه ی خوب، برنامه ای است که قبل از پردازش داده هایش، مقادیر ورودی خود را کنترل کند.

ساختار تصمیم گیری if-else:

در حقیقت از این ساختار وقتی استفاده می کنیم که دو دسته دستور (دستورات) را داشته باشیم. به طوریکه می خواهیم فقط یک دسته از آن ها اجرا شوند یعنی یا فقط دستور (دستورات) دسته ی اول اجرا شوند و یا فقط دستور (دستورات) دسته ی دوم. به این منظور توجه شما را به نمای کلی این ساختار جلب می کنم:

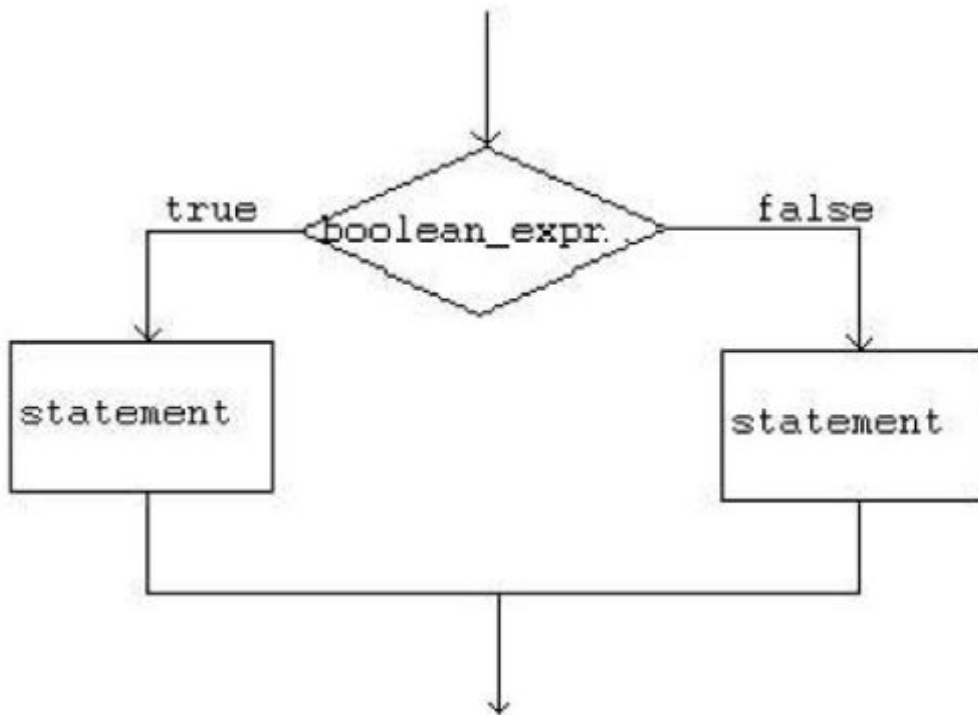
```
if (boolean expression){
    statements1;
    ....
    ....
}
else{
    statements2;
    ....
    ....
```

}

در ساختار فوق، کلماتی که با رنگ آبی مشخص شده اند، به عنوان کلمات کلیدی شناخته می شود. پس از آن باید گفت که ابتدا عبارت بولین که در درون یک جفت پرانتز قرار دارد، ارزیابی می شود. سپس در صورتی که این عبارت `true` ارزیابی شود دستور یا دستوراتی که در بلوک اول قرار دارند (`statements1`) اجرا می شوند و در صورتی که عبارت بولین مذکور `false` ارزیابی شود، دستور یا دستوراتی که در بلوک دوم قرار دارند (`statements 2`) اجرا می شوند.

لازم به یاد آوری است که در هر قسمتی از ساختار فوق اگر فقط یک دستور داشته باشیم، می توانیم آن را خارج از یک جفت آکولاد بیاوریم. اما به کار بردن آکولادها خوانایی برنامه هایمان را افزایش می دهد.

برای داشتن درکی بهتر، به فلوجارت زیر توجه کنید:



WWW

و حالا برنامه ی کوچک و بسیار ساده ای را که در بالا مطرح کردیم را به گونه ای دیگر می نویسیم. (هر دوی این قطعه کد ها کار مشابهی را انجام می دهند):

```
int grade=18;  
if (grade<10) System.out.println("You failed");  
else System.out.println("You are passed");
```

تنها تفاوت این قطعه کد با قطعه کدی که بدون else نوشتیم در این است که در قطعه کد دومی ، فقط یک بار عمل ارزیابی یک عبارت boolean انجام می شود.

ساختار تصمیم گیری if-else-if :

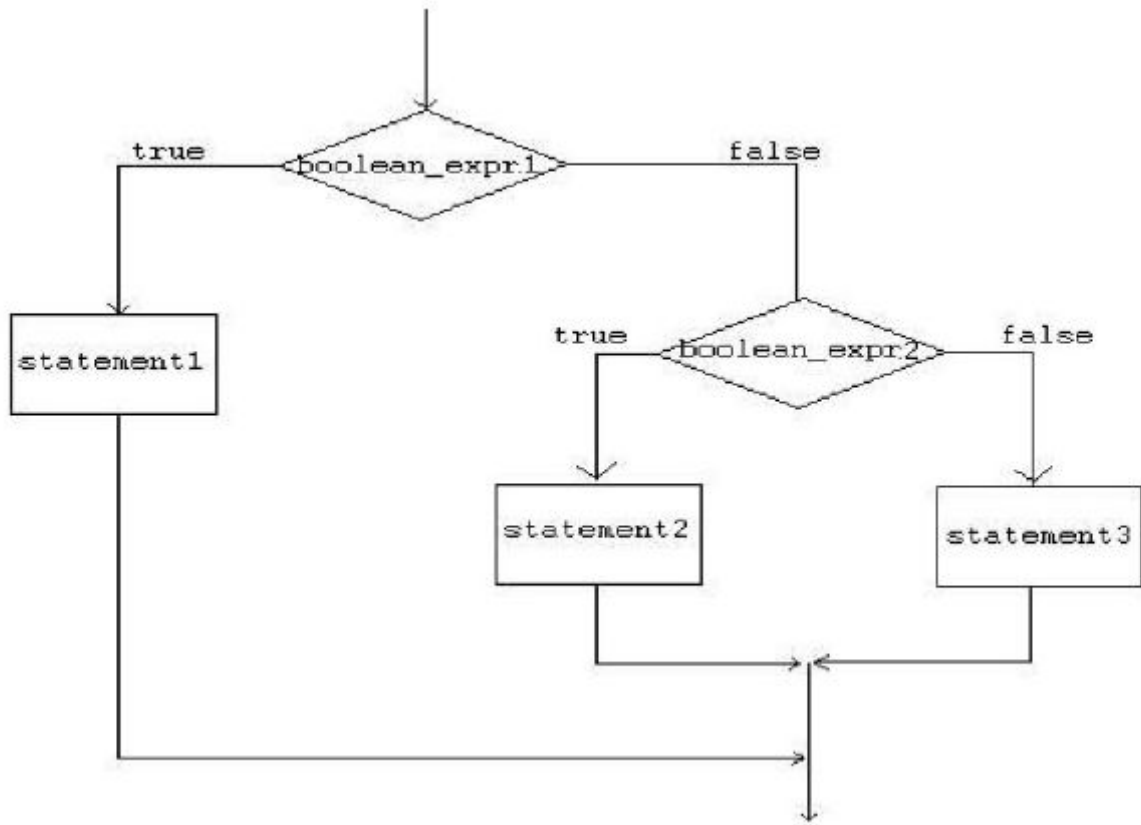
ساختار فوق به ما کمک می کند تا کنترل های بیشتری را بر روی برنامه ی خود اعمال کنیم. نمای کلی استفاده از این ساختار به صورت زیر است:

```
if (boolean expression 1){  
    statements 1;  
    ....  
    ....  
}  
else if(boolean expression 2){  
    statements 2;  
    ....  
    ....  
}  
else{  
    statements 3;  
    ....
```

```
....  
} // end of this code fragment
```

در عبارات مذکور، ابتدا 1 boolean expression ارزیابی می شود. در صورتی که این عبارت به صورت true ارزیابی شود، دسته عبارات 1 statements اجرا می شوند و کنترل برنامه به بعد از تمام این قطعه کد منتقل می شود. اما در صورتی که boolean expression 1 به صورت false ارزیابی شود، برنامه 1 statements را نادیده گرفته و به سراغ عبارت else if آمده و 2 boolean expression را ارزیابی می کند. در صورتی که این عبارت بولین true ارزیابی شود، دستورات 2 statements اجرا می شوند و کنترل برنامه به بعد از تمام این قطعه کد منتقل می شود ولی در صورتی که 2 boolean expression به صورت false ارزیابی شود، برنامه به سراغ بلوک بعد از کلمه ی کلیدی else رفته و محتویات این بلوک را اجرا می کند.

برای اینکه تصور بهتری از چگونگی عملکرد این ساختار داشته باشید، به فلوجارت زیر توجه کنید:



برای درک بهتر فرض کنید می خواهیم برنامه ی کوچک خودمان را به گونه ای تغییر بدهیم که نمره ی موجود را به صورت جزئی تر بررسی کند. به طوری که اگر نمره کمتر از ۱۰ باشد، پیغام مردودی را چاپ کند. اگر نمره بین ۱۰ تا ۱۲ باشد، پیغامی را بنویسد که در آن عبارت بیشتر تلاش کن به کاربر اعلان شود و در نهایت اگر نمره بیشتر از ۱۲ باشد، عبارت خوب است. را چاپ کند.

حالا به مثال زیر توجه کنید:

```

int grade=12;
if (grade<10){
    System.out.println("You failed");
}
else if(grade<=12){

```



```

System.out.println("Try more");
}
else {
System.out.println("Good !");
}

```

در قطعه کد مذکور، عملکرد برنامه به این شکل است که ابتدا عبارت $grade < 10$ مورد ارزیابی قرار می‌گیرد. سپس از آنجائیکه که این عبارت غلط ارزیابی می‌شود، عبارت **You failed** در خروجی چاپ نمی‌شود. اما از آنجا که این عبارت غلط ارزیابی می‌شود، کنترل برنامه (با علم به این که متغیر $grade$ محتوی عددی بزرگتر یا مساوی ۱۰ دارد) به سراغ `else if(grade <= 12)` می‌رود و با درست ارزیابی کردن عبارت بولین داخل پرانتز، عبارت **Try more** در خروجی این برنامه چاپ می‌شود و بعد از آن کنترل برنامه به بعد از این قطعه کد منتقل می‌شود.

حالا اگر در قطعه کد مذکور متغیر $grade$ با مقدار دهی اولیه شود، چه اتفاقی می‌افتد؟

اگر $grade$ با مقدار دهی اولیه شود چطور؟

اگر $grade$ با مقدار دهی اولیه شود چطور؟ (باز هم می‌بینیم که ممکن است برنامه ای که ما در بالا نوشته ایم، علی‌رغم اینکه با داده‌های اشتباهی کار می‌کند، نتایج به ظاهر درستی را به ما برگرداند!)

خطاهای رایج:

- نوشتن elseif به جای else if یکی از رایج ترین خطاهاست.
- و همچنین به کار بردن علامت انتساب = به جای == در booleans expression

تهیه شده در www.java4every1.wordpress.com

ایمیل تماس با مدیریت وبلاگ: blogsofmine@gmail.com

در پایان مثل همیشه از خوانندگان عزیز خواهش می کنم تا با نظرات و انتقادات خودشان بنده را در امر هرچه بهتر دنبال کردن فرآیند آموزشی در وبلاگ یاری کنند.