

به نام خدا

ساختار تصمیم گیری Switch

در این مقاله سعی می کنیم تا به اختصار نحوه ی استفاده از ساختار تصمیم گیری switch و ویژگی ها و امکاناتی که دارد را شرح بدهیم.

در مقاله ی قبلی به این مسئله اشاره کردیم که چگونه میتوانیم با استفاده از کلمات کلیدی if و else و نیز ترکیب های مختلفی که می توانیم از آن ها درست کنیم، یک ساختار تصمیم گیری درست کنیم.

حالا در این قسمت شما نحوه ی دیگری از راه های ممکن را فرا می گیرید تا بتوانید در مواقع خاصی از ساختار تصمیم گیری switch استفاده کنید.

ساختار های if این امکان را به ما می دهند تا برای مواقعی که می خواهیم بر حسب مقادیر مختلف یک متغیر ، دستورات خاصی اجرا شوند ، از آن ها استفاده کنیم ؛ اما اگر متغیرهایی که می خواهیم آن ها را مورد ارزیابی و مقایسه قرار دهیم ، مقادیر مشخصی را به خود بگیرند ، می توانیم از ساختار switch به جای ساختار های if استفاده کنیم .
برای درک بهتر ابتدا به نمای کلی این دستور توجه کنید:

```
switch (switch_expression){  
    case case_selector 1:  
        statements 1; // first block  
    ....
```

```

....
break;
case case_selector 2:
statements 2; // second block
....
....
break;
.
.
default:
statements 3; // default block
....
....
} // end of switch statement

```

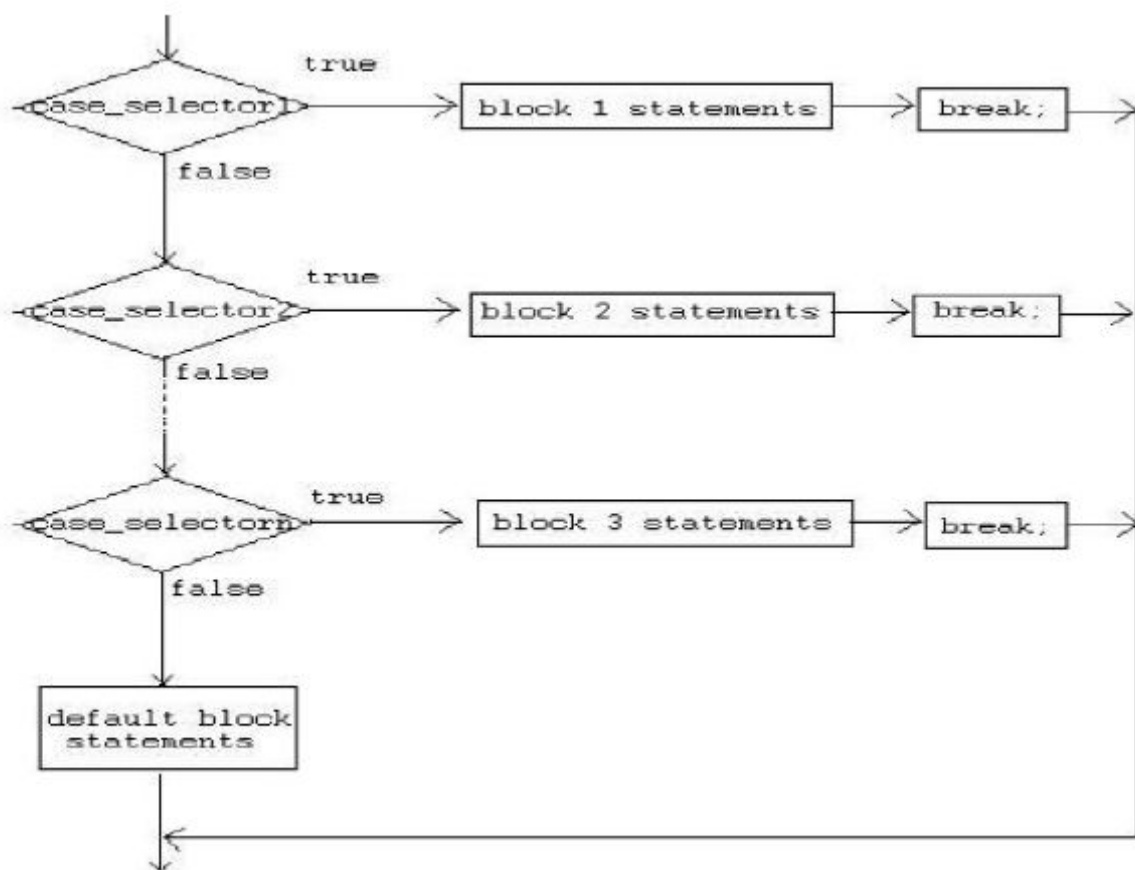
در نمای بالا، کلمات آبی رنگ همان کلمات کلیدی ساختار می باشند. به علامت هایی که با رنگ قرمز مشخص شده اند توجه کنید تا آن ها را با ; اشتباه نگیرید.

در جایی که به عنوان switch_expression مشخص شده است، فقط می توانیم نام یک متغیری را بیاوریم که یا از نوع char و یا از نوع int و انواع داده های آن باشد. (long, short).

در جاهایی که تحت عنوان case_selector مشخص شده اند، مقادیری می آیند که می خواهیم در صورتی که متغیر ما این مقدار را با خود داشت ، دستورات خاصی قرار بگیرند.

در صورتی که هر کدام از `case_selector` ها با مقداری که `switch_expression` در خود دارد، مطابقت داشته باشد، دستورات بعد از همان `case_selector` تا قبل از اولین عبارت `break` مورد اعمال قرار می گیرند.

وقتی یک عبارت `switch` در برنامه رخ می دهد، ابتدا مقدار `switch_expression` ارزیابی می شود و بعد از آن کنترل برنامه به `case_selector` ای می رود که مقدار آن مساوی مقدار `switch_expression` باشد. و اگر هیچ کدام از `case_selector` ها مساوی مقدار `switch_expression` نباشند، دستوراتی که در بلوک `default` قرار دارند، تا قبل از آکولاد اجرا می شوند.
به این فلوجارت توجه کنید:



مثال) فرض کنید در کلاسی نمرات کمی و کیفی طبق جدول زیر به دانش آموزان ابلاغ می شود:

بازه ی نمره ی کمی	نمره ی کیفی
20 - 17	A
17 - 15	B
15 - 12	C
12 - 10	D
10 - 0	E

حالا می خواهیم برنامه ای داشته باشیم که با دادن نمره ی کیفی به آن ، بازه ی نمره ی کمی را تعیین کند. و پیغامی را چاپ کند که متناسب با نمره ی دانش آموز باشد. به نمونه برنامه ی زیر توجه کنید:

```
public class grade{
    public static void main(String [] args){
        char grade=B;
        switch (grade){
            case A:
                System.out.println("Your grade is between 20 and 17");
                System.out.println("Exellent!");
                break;
            case B:
                System.out.println("Your grade is between 17 and 15");
                System.out.println("Good!");
                break;
            case C:
                System.out.println("Your grade is between 15 and 12);
                System.out.println("Try more");
                break;
            case D:
```

```

        System.out.println("Your grade is between 12 and 10");
        System.out.println("You are weak");
        break;
    case E:
        System.out.println("Your grade is less than 10");
        System.out.println("Sorry, you failed");
        break;
    default:
        System.out.println("This grade is unknown");
    } // end of switch statement
} // end of main method
} // end of grade class

```

در این برنامه به محض اینکه کنترل برنامه به عبارت switch می رسد، مقدار متغیر grade را ارزیابی می کند. اگر این متغیر حاوی مقدار A باشد، بازه ی نمره ی کمی به همراه پیغام Excellent! در خروجی چاپ می شود. اگر این متغیر حاوی مقدار B باشد، بازه ی نمره کمی به همراه پیغام Good چاپ می شود و به همین ترتیب ادامه پیدا می کند. اگر مقداری که متغیر grade در خود دارد، معادل هیچ کدام از مقادیر A , B , C , D , E نباشد، پیغامی مبنی بر ناشناخته بودن این نمره در خروجی ثبت می شود: This grade is unknown

✓ حالا می توانیم کاملا با توجه به شرایط مسئله ای که می خواهیم در برنامه مان به حل آن پردازیم، تصمیم بگیریم که از کدام ساختار تصمیم گیری بهتر است که استفاده کنیم؛ if یا switch . در ساختارهای تصمیم گیری if می توانیم مقداری که متغیر مورد

نظرمان دارد را در بین بازه ای از مقادیر مقایسه کنیم ولی در ساختار switch فقط می توانیم روی مقادیر یکتا بررسی خود را انجام بدهیم.

✓ بر خلاف ساختار if در ساختار switch لازم نیست که دستورات خود را در درون این ساختار، داخل یک بلوک قرار بدهیم. بلکه فقط کافی است که آن ها را بین کلمات کلیدی گفته شده قرار بدهیم.

تهیه شده در: www.java4every1.wordpress.com

ایمیل تماس با مدیریت وبلاگ: blogsofmine@gmail.com

مثل همیشه پذیرای نظرات، پیشنهادات و انتقادات شما هستیم.