



Switch Statement

Lesson:
JAVA Chapter 06

Switch Statement

Switch Case yapısı **if else** yapısının karışıklık oluşturacağı yerlerde tercih edilir, girilen veriye göre sınırsız işlem yaptırılmak isteniyorsa bu yapı kullanılır.

switch - case Nedir? Ne işe Yarar? Nasıl Kullanılır?

bir değer sonucuna göre işlem sağlamak istiyorsak ve bunu if-else komutunun karmaşıklaştığı bir noktada yaparken **switch-case** yapısı tercih edilir, bu yapı ile sonsuz işlem tanımlama yapılabilir, girilen bir değere göre "**case**" ler oluşturulur ve bunlara yapılacak işlem tanımlanır.

1. **switch** sözcüğünden hemen sonra gelen ifade **parantez içinde yer almali**.
2. **case** anahtar sözcüklerini izleyen ifadeler **değişken içermemelidir**.
3. **break** komutu ile sonlandırılır, uyuşmayan ifade de kullanılacak işlem ve ya mesaj seçilir
4. İki **case** sabiti aynı değeri alamaz.

switch deyiminin çalışma prensibi basittir. **switch** ifadesi hangi **case'i** izleyen sabitle çakışıyorsa, programın kontrolü o case'i izleyen kısma geçer. Bu case'den sonraki deyimler de kontrol edilmeden çalıştırılır.

Switch/case yapısında, case lerin çalıştırılıp sadece birini çalıştırıp bundan sonra **switch/case** yapısını terketmesini isiyorsak bu durumda her case seçeneğinden sonra **break** komutu yerleştirmeliyiz.



Switch Statement

```
public class Cases{
    public static void main(String args[]) {
        Scanner giris = new Scanner(System.in);
        int secim;

        System.out.println("Secim yapiniz (1 - 2 - 3)");
        secim=giris.nextInt();

        switch (secim) {
        case 1 :
            System.out.println("A sinifi dergilerin listesi");
            break;

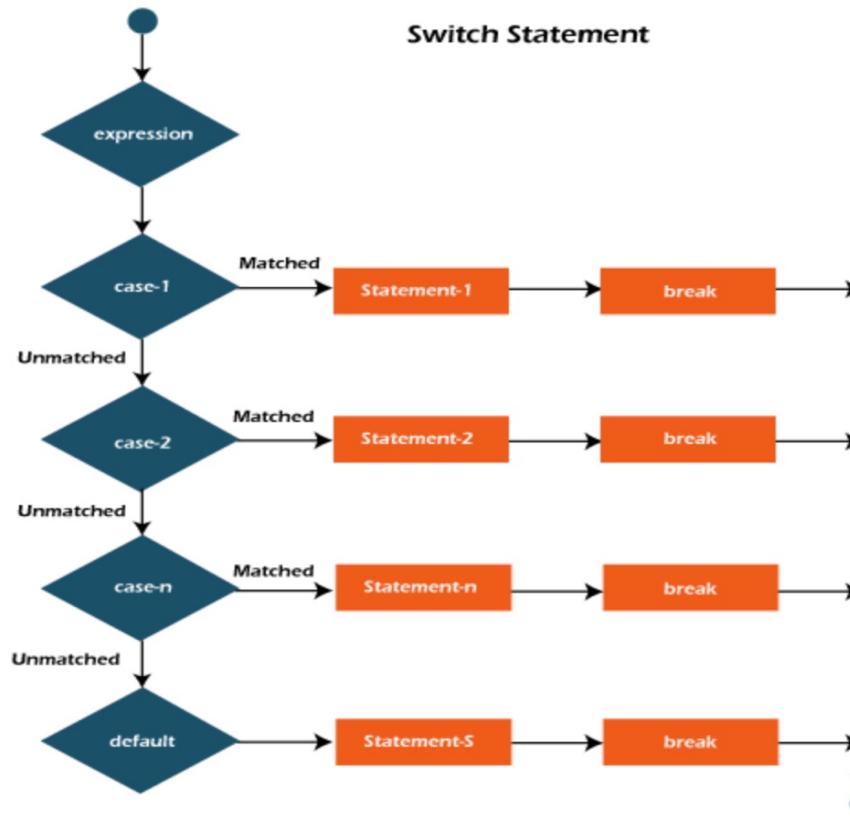
        case 2 :
            System.out.println("B sinifi dergilerin listesi");
            break;

        case 3 :
            System.out.println("C sinifi dergilerin listesi");
            break;

        default :
            System.out.println("Hatali secim! 1, 2 ya da 3'e basiniz.");
            break;
    }
}
```



Switch Statement



Switch Statement

```
int day = 4;
switch (day) {
    case 6:
        System.out.println("Today is Saturday");
        break;
    case 7:
        System.out.println("Today is Sunday");
        break;
    default:
        System.out.println("Looking forward to the Weekend");
}
// Outputs "Looking forward to the Weekend"
```

```
int day = 4;
switch (day) {
    case 1:
        System.out.println("Monday");
        break;
    case 2:
        System.out.println("Tuesday");
        break;
    case 3:
        System.out.println("Wednesday");
        break;
    case 4:
        System.out.println("Thursday");
        break;
    case 5:
        System.out.println("Friday");
        break;
    case 6:
        System.out.println("Saturday");
        break;
    case 7:
        System.out.println("Sunday");
        break;
}
// Outputs "Thursday" (day 4)
```



Switch Statement

break komutu yapacagimiz islem bittiginde switch statement'in sonuna gitmemizi saglar.

Java istenen case'e gittikten sonra **break** komutunu gorene kadar tum case'leri calistirir.

default komutu basta tanimlanan degisken icin hic bir case calismazsa calistirmak isedigimiz kodlari yazdigimiz bolumdur.

(If else statements da en sonda yazdigimiz else gibi calisir)

Switch Statement'da long,double,float ve boolean **kullanilamaz**



THANKS!

Any questions?

HalUk Bilgin | JAVA Backend Developer

BAŞARI GAYRETE AŞIKTIR ☺

