

## HW\_2\_SONUC\_RAPORU

**MÜH.TĞM. MUHAMMET YURTSEVEN**

**Anlaşılmayan** : Soru27,Soru34,Soru38, Soru 46

Soru 1- Cevap B : Switch yapısı Verilen değişken değerlerine göre işlem yapılmasını sağlar. Tam sayılar ile kullanılabilir fakat double, float gibi sayılarla kullanılmaz. Her koşuldan sonra break; deyimi kullanılmalıdır. Aksi halde sağlanan koşuldan sonraki tüm koşullar java tarafından doğruluğu kontrol edilmeksizin çalıştırılır.

<https://medium.com/gokhanyavas/javada-kontrol-yap%C4%B1lar%C4%B1-c7bf9336f5d7>

Soru 2 Cevap A :

Burada ;

int meal= 5;

int tip = 2;

int total = meal + (meal>6 ? ++tip : --tip); //Burada meal değeri 5'ten küçük olduğu için sağ taraf değerlendirilir ve yeni tip değeri 1 olur. Bu yüzden Cevap A seçeneğidir.

Soru 3- Cevap C

Soru 4 – Cevap D

Soru 5 – Cevap C : switch deyimi, tek bir ifadenin değerine göre sınırsız sayıda çalıştırma yolu belirlemeyi sağlayan bir komuttur. switch sözcüğünden hemen sonra gelen ifade parantez içinde yer almalı ve bir tamsayı ifade olmalıdır. case anahtar sözcüklerini izleyen ifadeler tamsayı sabit türünde ifadeler olmalıdır, yani değişken içermemelidir.

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

case sabitlerinden hiçbirisi ifade ile uyuşmuyorsa, programın akışı default'u izleyen kısma geçer. (Bu kısım mevcutsa) default'un en sonda olması şart değildir. Fakat en sona koymak iyi bir programlama stilidir. İki case sabiti aynı değeri alamaz.

Switch/case yapısında, programın kontrolünün ifadenin değerine göre case seçeneklerinden sadece birini çalıştırıp bundan sonra switch/case yapısını terketmesini istiyorsak bu durumda her case seçeneğinden sonra break komutu yerleştirmeliyiz.

---

Soru 6 – Cevap B :

long thatNumber = 5 >= 5 ? 1+2 : 1\*1; // İfade doğru olduğundan dolayı 1. Yer çalışır.

if(++thatNumber < 4) // Yukarıdan gelen değer 3 ve bir artırıldığında 4 olacaktır ve 4'ten küçük olmayacağı için yeni thatNumber değerimiz 4'tür.

thatNumber += 1;

---

Soru 7 – Cevap B :

Break deyiminin iki farklı uyarlaması bulunur ;

1. Etiketli Break : Label hangi döngünün başına konulmuş ise o döngüyü sona erdirir.
2. Etiketsiz Break : Yazıldığı döngüyü sona erdirir.

Continue ifadeleri de break ifadeleri gibi iki çeşide ayrılır:

1. Etiketli Continue : Label(Etiket) hangi döngünün başına konulmuş ise o döngüde pas geçer.
2. Etiketsiz Continue : Yazıldığı döngüde pas geçer.

go to : Aynı anda tek bir ifade kullanarak çoklu döngüleri bitirmek içindir.

exit: Programı bitirir.

---

Soru 8 – Cevap C : Her iki ifadeden yalnızca biri değerlendirilir. Parantez kullanımı zorunlu değildir sadece ayırt etmek için kullanılır. Yalnızca boolean ifadelere izin verilir. Bu ifadeler If-Then-Else yerine kullanıldığından Cevap C'dir

---

Soru 9- Cevap C

---

Soru 10- Cevap A

---

Soru 11- Cevap D :

- Önce ifade1 kontrol edilir; bu ifade doğru ise, Deyim1 uygulanır ve sonra Deyim\_n+1'e geçilir.
- İfade1 yanlış ise, o takdirde, ifade2 kontrol edilir. Bu ifade doğru ise, bu durumda sadece Deyim2 uygulanır ve sonra Deyim\_n+1'e geçilir.
- İfade2 yanlış ise, o takdirde, ifade3 kontrol edilir. Bu ifade doğru ise, bu durumda sadece Deyim3 uygulanır ve sonra Deyim\_n+1'e geçilir.

Bu işlemler daha fazla sayıda *else if* bulunduğu durumda aynı şekilde devam eder.

*if*'ler ile kontrol edilen ifadelerin tümü yanlış ise, bu durumda programın kontrolü *else*'i izleyen Deyim\_n'ye geçer ve bu deyim çalıştırılır. Daha sonra kontrol Deyim\_n+1'e geçer. Yukardaki yapı içinde Deyim1, Deyim2 ...Deyim\_n ile belirttiğimiz birimler tek bir deyimden ibaret olabileceği gibi, { } sembolleri içine alınmış deyimler grubundan da oluşabilir.

Boolean ifadesi yanlış ise hedef değerlendirilmez.

<https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/if-else-if-basamak-yapisi>

---

Soru 12- Cevap D

---

Soru 13- Cevap B : Bir case değeri değişmez veya sabit olmalıdır. case deyimi bir break deyimi ile bitirilebilir, ancak bu zorunlu değildir. Cevap B seçeneğidir.

---

Soru 14 Cevap D : A ve B seçenekleri sayısal operatör oldukları için değildir. && (ve) operatörü doğru olan seçenektir.

---

Soru 15- Cevap C:

---

Soru 16- Cevap B : Birinci pre-increment [++v] yeni değer oluştururken , ikinci post-decrement [v--] orijinale çevirir.

---

Soru 17- Cevap B

---

Soru 18- Cevap C: Uzun girdiler olabileceğinden veri kaybını önlemek için en uygun olan seçenek long seçeneğidir.

---

Soru 19- Cevap C : Java'da, if-then deyimlerinde veya üçlü işlemlerde boole ifadeleri yerine sayısal değerlerin kabul edilmediğini unutmayın. Parantez olması zorunlu değildir.

---

Soru 20- Cevap C

---

Soru 21- Cevap B

---

Soru 22- Cevap B : Çıkarma işlemi “-” işlemi ile gerçekleşirken, mod alma işlemide % ile gerçekleşmektedir.

---

Soru 23- Cevap B

---

Soru 24- Cevap B

---

Soru 25- Cevap C

---

Soru 26- Cevap A : İki nesne aynı ise == kullanılır. Aynı string nesnesiyse equals() true döndürür.

Soru 27- Cevap

---

Soru 28- Cevap D

---

Soru 29- Cevap B: & operatörü iki tarafında doğru olması gerekir && operatöründe bir tarafın doğru olması yeterlidir. Bu neden cevap B seçeneğidir.

---

Soru 30- Cevap C :

```
int x = 10, y = 5;
```

```
boolean w = true, z = false;
```

```
x = w ? y++ : y--;w = !z; //burada yeni x değeri 6 olacaktır.
```

```
System.out.print((x+y)+" "+(w ? 5 : 10)); // ilk taraf 6+5 ten 11 gelir. İkinci taraf doğru  
oldugundan 5 gelecektir. Cevap 11 5
```

Soru 31- Cevap A

Soru 32- Cevap B:  $12 + 6 * 3 \% (1 + 1)$  ilk önce parantez içi yapılır. Ondan sonra  
çarma işlemi gelir ve sonuç olarak  $12+18\%2$  ve  $12+0$  dan cevap 12 olacaktır.

Soru 33- Cevap D : XOR ^ operatörü, p ve q farklıysa true, aynı ise false olarak değerlendirir.

Soru 34- Cevap

Soru 35- Cevap: C /,\*,% sıralama bu şekilde olacaktır.

Soru 36- Cevap D : (XOR) ^ işleci, sonucu belirlemek için her iki işlenenin de  
değerlendirilmesini gerektirir. (XOR) ^ işlevi boole ifadelerde kullanılır.

Soru 37- Cevap C : En uygun seçenek z kullanılmadığı için.  $x || y$ .

Soru 38- Cevap

Soru 39- Cevap C : Soruya uyan şık C seçeneğidir. < ve >=

Soru 40- Cevap B

---

Soru 41- Cevap A

---

Soru 42- Cevap A

---

Soru 43- Cevap D : İşlenenlerden herhangi biri doğru ise || (OR(VEYA)) operatörü , tersine döndürme işlemide ! operatörü ile yapılır.

---

Soru 44- Cevap A :

```
int characters = 5;
```

```
int story = 3;
```

```
double movieRating = characters <= 4 ? 3 : story > 1 ? 2 : 1;
```

Burada ilk bölüm yanlış olduğundan ikinci bölüme geçer ve doğru olduğundan sonuç 2 döndürür.

---

Soru 45- Cevap B : herhangi bir sayıda case değeri olabilir , en fazla bir tane varsayılan değeri olabilir.

---

Soru 46- Cevap

---

Soru 47- Cevap D :  $(5 + (!2 + 8) * 3 - 3 \% 2) / 2$  // Burada !2 ifadesindeki ! bir sayıya uygulanamayacağından işlem gerçekleşmez.

---

Soru 48- Cevap C: || Operatöründe sadece birinin doğru olması yeterlidir. Bundan dolayı cevap C seçeneğidir.

---

Soru 49- Cevap A: Doğru cevap +,/, \* seçeneğidir.

---

Soru 50- Cevap C