

## HomeWork\_3\_Sonuç\_Raporu

TURGUT YILDIZ

01.05.2020

### CEVAPLAR

1. Switch-case yapısı float ve double değişken yapılarını ondalıklı gösterime sahip olduğundan desteklememektedir.  
**CEVAP : B**
2. Q2 uygulamasındaki gibi “?” ifadesi doğru ise 1. alan değilse 2. alanı döndürür.  
**CEVAP : D**
3. Q3 uygulamasında yeni bir nesne oluşturulduğundan ilk karşılaştırma “FALSE” 2. karşılaştırma ise 2. Satırda yapılan işlemde birbirlerine atanan değerlerden dolayı “TRUE” döndürür.  
**CEVAP : C**
4. Eğer yazılan kod yanlışlık yapıldıysa derlenmez. Q4 uygulamasındaki gibi düzeltilmeden sonra sonuç aşağıdaki gibi çıkacaktır.  
“plan++” operatörü önce işlem yapıp daha sonra ifadeyi 1 arttırır.  
“--plan” operatörü önce değeri 1 azaltır daha sonra işlemi gerçekleştirir.  
**CEVAP : B**
5. “default” ifadesi “switch-case” ifadesinde isteğe bağlıdır, kontrol için herhangi bir değer alamaz.  
**CEVAP : C**
6. “?” operatörü “TRUE” değer için 1. bölümü çalıştıracaktır. Bundan dolayı ilk adımda değişkenin değeri 3’tür. Daha sonra “++” operatörü değişkenin değerini 1 arttırır ve değer “4” olur.  
**CEVAP : B**
7. “break” deyimi switch-case ifadesinden veya döngülerden çıkmak için kullanılan bir anahtar deyimdir.  
**CEVAP : B**
8. “?” ifadesi yalnızca bool ifade döndürür. “true” ise 1. ifade “false” ise 2. ifade döndürülür. If-else ifadesi yerine kullanılabilen bir ifadedir.  
**CEVAP : C**
9. Q9 uygulamasında çalışan bir kod ile derleme yapılmış. Programın soruda verilen hali ile çalışmayacağı uygulama üzerinde belirtilmiştir.  
**CEVAP : C**

10. Bir sayının “%” işlemi yapıldığından hangi sayıya göre mod alınıyorsa o sayı ile bölümünden kalan sayıyı döndürür. “6 % 3 = 0 ” olduğundan if bloğuna girilmez ve “triceratops” değeri 1 azaltılarak ekrana çıkarılır.

**CEVAP : A**

11. A : “if-else” bloğunda “else” ifadesi olmayabilir.

B : “if-else” ifadesinde değerlendirilen ifade ”false” dönerse o blok değerlendirilmez.

C : Bir nesneyi değerlendirmek için “if-else” ifadesi olmasına gerek yoktur.

D : “if-else” ifadesi değerlendireceği bloğu sadece 1 satır veya {} arasındaki tüm satırları çalıştırır.

**CEVAP : D**

12. Bir “if-else” bloğu yalnızca 1 if’ten oluşabilir. Bundan dolayı . Q12 uygulamasındaki ilk if bloğu çalışır. 2.ifadede else bloğu çalışacaktır. Cevabın “Not enoughToo many” olması gerekmektedir.

**CEVAP : D**

13. “switch-case” ifadesi “break” deyimi kullanılarak sonlandırılmak zorunda değildir.

**CEVAP : B**

14. Mantıksal operatörlerden ve(&&) ifadesi ile tabloya ulaşılabilir.

**CEVAP : D**

15. İf deyiminde boolean ifadeler değerlendirilir. Q15 uygulamasına bakıldığında integer değerden dolayı kod derlenmez.

**CEVAP : C**

16. Pre-fix “++v” ifade önce değeri artırır daha sonra işlem yapılır. Post-fix “v++” önce işlem yapılır daha sonra değeri artırılır.

**CEVAP : B**

17. Q17 uygulamasındaki gibi.  $3 + 2 * (2 + 3) = 13$  değerine ulaşılır.

**CEVAP : B**

18. Q18 uygulamasındaki gibi byte ve integer değerler “switch-case” ifadesinde kullanılırken long ifade desteklenmemektedir.

**CEVAP : B**

19. `time > 10 ? day ? "Takeout" : "Salad" : "Leftovers";` // day ifadesi bir integer değer olduğundan ve mantıksal boolean bir sonuç dönmediğinden derlenmez.

**CEVAP : D**

20. Q20 uygulamasına bakıldığında “)” eksiği olduğundan derlenmeyecektir. Derlenseydi “Too many” olurdu.

**CEVAP : C**

21. Q21 uygulamasında String ifadeye kadar olan değerler toplanır ve String ifade ile yanyana gelerek “117” değerine ulaşılır daha sonraki her bir “+” operatörü sonraki gelen değeri String ifade ile birleştirerek ekrana “11789” değerine çıkarır.

**CEVAP : B**

22. “-” operatörü iki sayıyı birbirinden çıkarır.

“%” operatörü iki sayıyı böler ve kalanı verir.

“/” operatörü iki sayıyı böler ve bölüneni verir.

**CEVAP : B**

23. Q23 uygulamasına bakıldığında sonucun 11 olduğu görülür.

**CEVAP : B**

24. Q24 uygulamasına bakıldığında “switch-case” yapısında case içine herhangi bir kontrol konulmazsa sonraki case adımına geçilir.

**CEVAP : B**

25. Q25 uygulamasına bakıldığında “?” ifadesinde dönüş yapılacak iki ifadede aynı türden olmalıdır. (String,String veya int, int) gibi.

**CEVAP : C**

26. Sorunun son ifadesinde apples ifadesinin kesinlikle oranges ifadesine eşitliği döndürülecektir deniliyor bundan dolayı ilk ifadede “apples == oranges” kontrolü yapılır. Doğru ise 2. Seçenek sağlanır.

**CEVAP : A**

27. Soruda **myTestVariable** değişkeninin null olmadığı söyleniyor. Bu durumda **myTestVariable.equals(null)** ifadesinin null değere eşit olup olmadığı kontrol edildiğinde false döner.

**CEVAP : B**

28. Q28 uygulamasında else if içine yazılmış olan streets ifadesi boolean sonuç verecek bir karşılaştırma yapılırsa kod derlenir.(Örneğin ben streets>500 ifadesi ile bir karşılaştırma yaptığımda çıktı “11” olur.) Soruda kod derlenmez olacaktır.

**CEVAP : D**

29. -

30. Q30 uygulamasına bakıldığında int bir değer null harici bir değere sahip olması onu “true” olarak değerlendirir. “!” tersleme yaparak boolean ifadelerde kullanılır.

**CEVAP : C**

31. Q31 uygulamasında hem “==” operatörü hem de equals() metodu kullanılmış ve “true true” çıktısı görülmüştür.

**CEVAP : A**

32. Java operatör işlemlerini prefix yöntemiyle yaptığından  $[12 + 6 * 3 \% (1 + 1)]$  işlemi soldan sağa doğru yapılır.  $[12 + 18 \% 2] \Rightarrow [12 + 0] \Rightarrow 12$  olur.

**CEVAP : B**

33.  $p \wedge q$  XOR yani özel veya kapısıdır  $p'q + pq'$  şeklindedir.  $[(1 \ 1) \text{ ve } (0 \ 0)]$  değerlerinde “false”  $[(1 \ 0) \text{ ve } (0 \ 1)]$  değerlerinde “true” döner.

**CEVAP : D**

34. -

35. –

36. 33. Soruda belirtildiği gibi “ $\wedge$ ” ifadesi XOR kapısıdır. Bundan dolayı 2 boolean ifadeyi karşılaştırır.

**CEVAP : D**

37. Sorudaki Venn şeması A birleşim B ifadesi yani A veya B mantık kapısı ile ifade edilebilir.  $[A \cup B]$

**CEVAP : C**

38. Case ifadesinden sonra değişken ismi getirilirse derleme gerçekleşmeyecektir.

**CEVAP : D**

39.  $[x \geq 5.21 \ \& \ x < 8.1]$  şeklinde yapılır.

**CEVAP : C**

40. Q40 uygulamasında işlem öncelikleri ve “?” ifadesi takip edilirse “Turtle wins!” olur.

**CEVAP : B**

41. Q41 uygulamasında  $(x > 5)$  ifadesi için 5 ten büyük bir değer gönderilmediği için “?” operatörünün false yani 2. tarafı döndü. Bundan dolayı sonuç=0

**CEVAP : A**

42. Q42 uygulamasına bakıldığında atama “=” ve karşılaştırma “==” operatörleri arasındaki fark bu soruyu çözdürür. Atama işlemi yapıldığında “true” değer döneceği için if bloğu içindeki işlemler yapılır.

**CEVAP : A**

43. q veya p işleminde herhangi bir eleman true olursa sonuçta true olur. “!” operandı ise o anki bool ifadeyi tersine çevirir.

**CEVAP : D**

44. Q44 uygulamasındaki gibi ilk “?” karşılaştırması “false” döner. Bundan dolayı 2. Taraftaki “?” operatörlü kısım işletilir ve bu kısım “true” dönerek `movieRating` değerini 2.0 yapar.

**CEVAP : A**

45. “Switch-case” ifadesinde herhangi bir sayıda case ve en fazla 1 tane default ifadesi olabilir.

**CEVAP : B**

46. Q46 uygulamasına bakıldığında [java.lang.ArrayIndexOutOfBoundsException](#) hatası ile karşılaşıldı ve bu hatanın uzunluğu olmayan bir dizinin karşılaştırma operatörlerinden biri ile işleme tutulması sonucu ortaya çıkmıştır.

**CEVAP : A**

47. [ (5 + (!2 + 8) \* 3 - 3 % 2)/2 ] ifadesinde !2 ile bir integer değere boolean bir ifade operatörü uygulandığından kod derlenmez.

**CEVAP : D**

48. “||” veya kapısıdır. Her 2 değişkeninde “false” olduğu durumlarda “false” diğer durumlarda “true” döner.

**CEVAP : C**

49. “-” çıkarma işlemi ile aynı önceliğe sahip “+” toplama işlemidir. Bu operatörlerden daha öncelikli ama “%” mod operatöründen önceliksiz operatörler ise “\* ve /” operatörleridir.

**CEVAP : A**

50. Q50 uygulamasına bakıldığında String bir ifadeye integer değer ataması yapılmaya çalışılmış ve uygulama //p1 satırında hata verip derlenmemiştir.

**CEVAP : C**