

1. Cevap B. switch deyimi içinde yer alan ifade integral değişkenler olarak ifade edilen int, short, byte türü bir değişken, String, karakter ya da enumeration tipi bir ifade olabilir. Bu nedenle int, char ve String ifadeler uygundur, double değildir.  
Kaynak: <https://en.cppreference.com/w/cpp/language/switch>
2. Cevap A. Java'da üçlü kontrol yapısına sokulan meal değişkenin değeri 6'dan küçük olduğundan dolayı tip değişken pre-decrement operatörü ile çalışarak önce değeri 1 azalacak sonra işletilecektir. Sonuçta 1 değeri ile yüklenmiş olacaktır.  
Kaynak: <https://www.geeksforgeeks.org/java-ternary-operator-with-examples/>  
<https://www.geeksforgeeks.org/interesting-facts-increment-decrement-operators-java/>
3. Cevap C. İlk satırda değeri "john" olarak bir String yaratılmıştır. İkinci satırda ise, new operatörü ile değeri "john" olan bir String'in referansı jon isimli referansa yüklenmiştir. == operatörü ile eşitlikleri kontrol edildiğinde false değeri döner. Çünkü, jon referansının gösterdiği değer, new operatörü ile oluşturulan String'in bellekteki yeridir. Bu nedenle eşit değildirler ve false döner. equals() metodu true döner. Çünkü refere ettikleri yerdeki değerler aynıdır.  
Kaynak: <https://www.geeksforgeeks.org/difference-equals-method-java/>  
<https://www.tutorialspoint.com/differences-between-and-equals-method-in-java>
4. Cevap B. Öncelikle kodda sözdizimi (syntax) hatası vardır. else if bloğunun süslü parantezleri yanlış konumlandırılmıştır. plan değişkenine yeni değer ataması yapılırken önce plan++ ile 2 değeri elde edilir. Post-increment operatörü değişkenin kendi değerini değiştirmez. Ardından elde edilen 2 değeri ile plan değişkeninin pre-decrement operatörü ile işletilmiş hali toplanır. plan değişkeninin değeri halen 1 olduğundan bu aşamada bir azaltılır ve değişkenin değeri de 0 (sıfır) olarak güncellenecektir. Sonuçta 2+0 işlemi sonucu 2 değeri plan değişkenine atanır ve else if bloğu icra edilir. "Plan B" ifadesi ekran çıktısı olarak basılır.  
Kaynak: <https://www.baeldung.com/java-unary-operators>
5. Cevap C. switch yapısındaki default deyimi tercihe bağlıdır. Gerek duyulmuyorsa implemente edilmeyebilir. default deyiminin switch yapısının case'lerinin en sonunda olması gerektiği ile ilgili bir kısıt yoktur. Sıralama olarak switch yapısı içinde herhangi bir yerde konumlandırılabilir. default deyimi, switch yapısında yer alan hiçbir case'e uyulmadığı durumda icra edilir. İcrası için bu case'lerden birine girmemelidir. Bu nedenle A, B ve D seçeneklerinde verilen ifadeler yanlıştır.  
Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>
6. Cevap B. Kod parçasının ilk satırında üçlü kontrol yapısı vardır. Burada 5 sayısı 5'ten küçük ya da eşit mi kontrolü vardır ve true döneceği için 1+2 işlem sonucu olan 3 değeri thatNumber değişkenine yüklenir. if deyimi içinde thatNumber değişkeni pre-increment operatörü ile önce 1 artırılır ve thatNumber değişkeninin yeni değeri 4 olur. Sonrasında 4<4 kontrolü yapılır ve sonuç false olacağı için bir alt satırdaki thatNumber += 1; işlemi icra edilmez. thatNumber değişkeninin sonra değeri 4'tür.  
Kaynak: <https://www.geeksforgeeks.org/java-ternary-operator-with-examples/>
7. Cevap B. switch bloğundaki tüm case'leri ve default deyimini atlayarak bloktan çıkmak için kullanılan operatör break operatörüdür. Java'da exit isimli bir operatör bulunmamaktadır. goto ifadesi rezerve edilen bir kelime olmasına karşın Java'da bunun da karşılığı olan bir operatör yoktur. continue ifadesi ise for, while, do-while gibi döngülerde kullanılır.  
Kaynak: <https://www.javatpoint.com/java-break>  
<https://www.javatpoint.com/java-switch>
8. Cevap C. Üçlü kontrol yapılarında hiçbir surette koşul operatörünün sağında her iki ifade de işletilmez. Yalnızca birisi işletilebilir. Bu nedenle A seçeneği yanlıştır. Üçlü kontrol yapısında kullanılan parantezler her ne kadar okumayı kolaylaştırıp hatayı önlemeye yardımcı olsa da zorunlu değildir. Üçlü kontrol yapısında koşula sokulan deyim sadece boolean olabilir. Böylece

doğru ise ilk, değilse ikinci işlem icra edilir. Bu nedenle B ve D seçenekleri de yanlıştır. Üçlü kontrol ifadeleri çoğunlukla çok kısa olan if else yapılarını daha kısa ifade etmek için kullanılır. Bu nedenle if-then-else yapısına kolayca çevrilebilir. Tam tersi de yapılabilir. Böylece C seçenektir.

Kaynak: <https://www.geeksforgeeks.org/java-ternary-operator-with-examples/>

9. Cevap C. Kod derlenmez. Java’da Logical And operatörü (&&) argümanları boolean olmalıdır. Bu nedenle 4 numaralı satırdaki hatadan dolayı kod derlenmez.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>  
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op2.html>

10. Cevap A. İlk olarak if kontrolünde pterodactyl değişkeninin 3’e göre modu alınır. 6 değerinin 3 bölümünden kalan değer 0’dır.  $0 \geq 1$  işlem sonucu false döner. if bloğu süslü parantezle implemente edilmediği için true olması durumunda sadece bir alt satır icra edilecekti. false döndüğü için triceratops++ komutu işletilmez, bir alt satırdaki triceratops—işletilerek değişken 2 değeri ile yüklenir. Sonuç olarak ekran çıktısı 2 olacaktır.

Kaynak: <https://www.geeksforgeeks.org/java-ternary-operator-with-examples/>  
<https://www.geeksforgeeks.org/interesting-facts-increment-decrement-operators-java/>

11. if-then yapısında else ifadesi gerekli değildir. A seçeneği yanlıştır. if-then yapısında ifade false dönerse hiçbir surette then kısmında yer alan komutlar icra edilmez. B seçeneği de yanlıştır. if-then yapısı bir nesneye casting yapmak için gerekli değildir. Bu nedenle C seçeneği de yanlıştır. if-then yapısında süslü parantezler kullanılmazsa sadece bir satır, süslü parantez varsa blok içinde kalan tüm komutlar icra edilecektir.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>

12. Cevap D. flair değişkeninin değeri 15’tir. Dolayısıyla ilk if kontrolünde,  $15 \geq 15$  işlem sonucu ve  $15 < 37$  işlem sonucu true dönerler. Bu iki işlem sonucu AND operatörü ile icra edilir ve sonuç true döner. İlk olarak ekrana “Not enough” yazacaktır. Bir sonraki if kontrolüne girmez. Buradaki else bloğu,  $\text{if}(\text{flair} == 37)$  ile başlayan bloğa aittir ve if bloğuna girmediği için buraya girecek ve ekrana “Too many” yazacaktır. Sonuç olarak ekranda “Not enoughToo Many” ifadesi görülür.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>

13. Cevap B. Bir case değeri final değer ya da bir literal ifade olabileceği için A ve C seçenekleri doğrudur. Case değeri, switch deyimi argümanı aynı tipte ya da yükseltilebilen bir değer olmak zorundadır. Bu nedenle D seçeneği de doğrudur. Case deyimleri içinde break operatörü zorunlu değildir. Bu nedenle B seçeneği yanlıştır.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>

14. Cevap D. Verilen doğruluk Logical And (&&) operatörüne aittir.

Kaynak: [https://www.cs.utah.edu/~germain/PPS/Topics/truth\\_tables\\_and\\_logic.html](https://www.cs.utah.edu/~germain/PPS/Topics/truth_tables_and_logic.html)  
<https://www.w3resource.com/java-tutorial/java-logical-operators.php>

15. Cevap C. if deyimine argüman olarak tipin boolean olması gerekmektedir. Bu nedenle kod 4. satırdan dolayı derlenmez.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>

16. Cevap B. Pre-increment operatörü değeri ya da değişkeni 1 artırarak yeni değer ile yükler. Post-decrement operatörü değeri ya da değişkenin değerini 1 azaltır ancak orijinal değer ile döner.

Kaynak: <https://www.geeksforgeeks.org/interesting-facts-increment-decrement-operators-java/>

17. Cevap B. Java’da operatör önceliğine göre en öncelikli olan operatör atama (=) operatörüdür. winner değişkenine değer yüklenmesi işleminde operatör önceliğine göre önce parantez içinde tiger+lion işlemi icra edilir. Elde edilen değer ile 2 ile çarpılarak lion değişkeninin değeri ile

toplanarak winner değişkenine atanır.  $winner = 3 + 2 * (2 + 3)$  işlem sonucu olan 13 değeri ekran çıktısı olarak basılır.

Kaynak: <https://introcs.cs.princeton.edu/java/11precedence/>

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>

18. Cevap B. dayOfWeek değişkeninin int olarak kabul edilmesinden dolayı saturday değişkeninin tipi int ya da int'e yükseltilebilen bir değişken olmalıdır. Bu nedenle char, byte, short veya int olabilir. Ancak B seçeneğinde verilen long tipinin hem switch deyimine argüman olamaması hem de int tipine veri kayıpsız çevrim yapılamayacağından dolayı long tipi geçerli olamaz.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>

19. Cevap D. Verilen kod parçacığı dinner değişkenine değer atanmak istenen satırdan dolayı derlenmez. Kodun derlenmeme nedeni C seçeneğinde ifade edilen parantez eksikliğinden dolayı değil, içte yer alan üçlü kontrol yapısındaki `<< day ? "Takeout" : "Salad" >>` ifadesinde day değişkenine yerine boolean bir ifade olması gerektiğindendir. A ve B seçenekleri kodun derlenmemesinden dolayı, C seçeneği ise kodun derlenmeme nedeninin yanlış ifade edilmesinden dolayı yanlıştır.

Kaynak: <https://www.geeksforgeeks.org/java-ternary-operator-with-examples/>

20. Cevap C. main() metodu içine girişte ilk adımda, leaders değişkenine değer atanması işleminde ilk olarak en içteki  $(1 + 2 / 5)$  işlem sonucunun 2 katının 10 fazla atanmak istenmektedir. Ancak kapama parantezi unutulmasında dolayı kod derlenmez. Eğer kapama unutulmamış olsaydı, leaders değişkenine işlemler sonucu 30 değeri yüklenecekti. followers değişkenine de 60 değeri yüklenirdi. Bu işlemler sonucu, ekran çıktı basma işleminde leaders + followers işlem toplamı 90 sayısı, 10 değerinden büyük olduğunda dolayı üçlü kontrol yapısında false dönerek "Too many" ifadesini ekran çıktısı olacak basacaktı.

Kaynak: <https://www.geeksforgeeks.org/java-ternary-operator-with-examples/>

21. Cevap B. Java + operatörünü soldan işletir. Bu neden print() metoduna parametre verilen  $5 + 6 + "7" + 8 + 9$  ifadesinde önce  $5 + 6$  işlemini icra eder. İki değer de nümerik olduğundan 11 sonucu elde edilir. Sonrasında 11 sayısı ile "7" String değeri toplar. Türleri farklı olduğundan dolayı 11 sayısı String gibi düşünülerek 7 ile birleştirilir. "117" + 8 + 9 işlemi icra edilir. Bu aşamadan sonra solda kalan değer String bir değer olacağından dolayı kalan tüm ifadeler String olarak birleştirilir ve ekran çıktısı 11789 olur.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op1.html>

22. Cevap B. İki sayı arasındaki farkı bulmakta kullanılan operatör çıkarma (-) operatörüdür. Bir sayının başka bir sayıya bölümünde kalanı bulmakta kullanılan operatör ise mod alma (%) operatörüdür.

Kaynak: [https://www.tutorialspoint.com/java/java\\_basic\\_operators.htm](https://www.tutorialspoint.com/java/java_basic_operators.htm)

23. Cevap B. dog değişkenine 11, cat değişkenine 3 değeri atanmıştır. partA değişkenine  $11 / 3$  işlem sonucu 3 değeri yüklenir. partB değişkenine  $11 \% 3$  işlemi sonucu 2 değeri yüklenir. newDog değişkenine  $2 + 3 * 3$  işlem sonucu olan 11 değeri yüklenir. \* operatörü, + operatöründen önceliklidir. Ekran çıktısı olarak 11 basılır.

24. Cevap B. flavors değişkeninin değeri 30 olduğu için case 30 ifadesindeki komutlar icra edilir. case ifadesinde break operatörü bulunmadığı için case 40 ifadesindeki komutlar da işletilir. Burada da break olmadığından dolayı default case de icra edilir. Böylece eaten değişkeni önce bir artar, sonra 2 artar ve en son 1 azalarak 2 değeri ile yüklenir.

25. Cevap C. travel() metodundaki return ifadesinde üçlü kontrol yapısı kullanılmıştır. Burada distance < 1000 işlemi sonucunda dönen değer false olması durumunda dönülmesi gereken değer String türünden olmadığı için kod derlenmeyecektir.

26. Cevap A. Java'da == ve equals() arasındaki fark şudur: == operatörü bir referans karşılaştırması yapar. Yani her iki nesne de aynı bellek konumuna işaret eder. equals() metodu, nesnelerdeki değerlerin karşılaştırmasını değerlendirir.

Bu nedenle referans eşitliği için == operatörü, değer eşitliği için equals() metodu kullanılır. Doğru seçenek B'dir.

Kaynak: <https://ahmetkilinc.net/javada-string-karsilastirmasi-yapmak/>  
<https://www.geeksforgeeks.org/difference-equals-method-java/>

27. Cevap B. myTestVariable değişkeni null ile yüklü olmayan herhangi bir değişken olduğundan dolayı null değeri eşitliğini equals() metodu ile kontrol edersek false döner. Eğer null olan bir String ile equals() kontrolü yapılırsa runtime'da NullPointerException fırlatır.

Kaynak: <https://www.javatpoint.com/java-string-equals>

28. Cevap D. "else if (streets && intersections > 1000)" ifadesinde söz dizimi hatası vardır. streets değişkeni boolean olmalı ya da bir sayı ile karşılaştırılarak true/false dönen bir ifade yazılmalıdır. Örneğin, else "if (streets > 1000 && intersections > 1000)" şeklinde yazılıysaydı ekran çıktısı olarak "11" yazacaktı.

29. Cevap B. Java'da & operatörü bitset olarak verilen iki değer için tüm bitlerini AND işlemine sokarak yeni bir değer elde eder. && operatörü ise 2 tane boolean değişkeni ya da değeri AND işlemine sokarak true ya da false değeri elde eder.

Kaynak: <https://www.tutorialspoint.com/Differences-between-and-and-and-and-operators-in-Java>

30. Cevap C. İlk iki satırda değişkenlere ilk değer ataması yapılmıştır. 3. Satırda x değişkeninin değeri, w değişkeninin false olmasından dolayı y++ işlemi ile 6 olarak güncellenir. Sonraki adımda w değişkeni, z değişkeninin değeri ile güncellenerek yeniden true olarak atanır. Ekran çıktısı alınan satırda önce (x+y) yani 6+5 işlem sonucu 11, sonra bir boşluk karakteri, sonrasında ise w değişkeninin true olmasından dolayı 5 değeri ekrana basılarak "11 5" olarak bir çıktı alınır.

31. Cevap A. bob değişkeni oluşturulduktan sonra oluşturulan notBob değişkenine bob değişkeninin referans ettiği adres bilgisi verilmektedir. Bu nedenle bu iki değişken aslında aynı yeri referans ettiği için == operatörü ya da equals() metodu ile kontrol edildiklerinde her ikisi de true döner. Ekran çıktısı "true true" olacaktır.

Kaynak: <https://www.javatpoint.com/java-string-equals>

32. Cevap B. Önce içteki parantez işlemi icra edilir ve 2 değeri üretilir. Sonrasında \* operatörü önceliklidir 6\*3 işlemi icra edilir. 12+18%2 işlemi elde edilmiş olur. % operatörü öncelik alır ve 18%2=0 sonucu elde edilir. En son olarak da 12+0 işlem sonucu 12 değeri elde edilir.

Kaynak: <https://introcs.cs.princeton.edu/java/11precedence/>  
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>

33. Cevap D.

XOR (^) Operator	p = true	p = false
q = true	false	true
q = false	true	false

Tablo-1: XOR Operatörü Doğruluk Tablosu

Kaynak: <https://www.baeldung.com/java-xor-operator>

34. Cevap C. Soruda main() metoduna komut satırı argümanı gönderimiyle ilgili çıktı sorulmaktadır. Kod derlendikten sonra çalıştırılırken "java [program\_adi] [parametre\_listesi]" şeklinde çalıştırılır. Eğer program adından sonra ilk parametre için sound verirse ekran çıktısı sound, logic verirse ekran çıktısı logic olacaktır. Parametre verilmezse boş çıktı üretir. Hiç parametre girmemek ya da 2'den fazla girmek kodun runtime'da hata fırlatmasına sebep olmaz.

Kaynak: <http://www.mathcs.emory.edu/~cheung/Courses/170/Syllabus/09/command-args.html>

35. Cevap C. Verilenlerden C seçeneğindeki operatörler 12. seviyede birlikte yer almaktadırlar.  
Kaynak: <https://introcs.cs.princeton.edu/java/11precedence/>
36. Cevap D. XOR (^) operatörü, 2 boolean operand ile icra ettirilir. Her iki operandın farklı olması durumunda true, aynı olması durumunda false döner. A seçeneğinde ifade edilen operandlardan birisi true ise daima true sonucu üretmesi yanlış ifadedir. B seçeneğindeki ifade de ^^ operatörü ile aynı anlamda olması yanlış ifadedir. C seçeneğinde ifade edilen bilgi, her iki operandın true olması durumunda false döneceği için yanlıştır. Bu operatör 2 tane boolean değer ile uygulanır.  
Kaynak: <https://www.baeldung.com/java-xor-operator>
37. Cevap C. Verilen Venn şeması x ve y kümelerini OR işlemine tabi tutulması ile elde edilir. z kümesi kullanılmadığı için dahil edilmez.  
Kaynak: <http://inet.mountsaintvincent.edu/library2/venn.htm>
38. Cevap D. switch yapısında case ifadelerine sabit bir değer verilmesi gerekmektedir. Söz dizimi hatasından dolayı kod derlenmez. Cevap verilen seçeneklerinden hiçbiridir.  
Kaynak: <https://riptutorial.com/java/example/26261/constant-expressions>
39. Cevap C. Verilen bir sayının 8.1 değerinden küçük ve 5.21 sayısından büyük eşit aralığı ifade etmek için kullanılacak operatörler sırasıyla < ve >= operatörleridir.  
Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/opsummary.html>
40. Cevap B. main() metodu ilk adımında turtle değişkenine atanan değer, işlemler sonucu 30 olacaktır. Sonraki adımda hare değişkenine atanan değer, üçlü kontrol ifadesinde turtle değişkeninin 30 değerinde olması ve 5 ile kıyaslanması sonucu false dönerek 25 olacaktır. Ekran çıktısı alınan satırda ise turtle = 30, hare = 25 değerleri kıyaslanır. Sonuç false olacağından dolayı ekran çıktısı B seçeneğindeki gibi olacaktır.
41. Cevap A. Verilen kodda main() metodu içinde bir ekran çıktısı basılmaktadır. getResult() metoduna sırasıyla 5, 1, 0 ve 2 değerleri gönderilmekte ve dönen sayısal değerler toplanmaktadır. Gönderilen 4 değer her birine karşılık 0 sonucu dönecek ve 0+0+0+0 = 0 sonucu elde edilecektir. Ekran çıktısının sonundaki "" ifadesi kendisinden önceki ifade ile String olarak birleştirme yapmaktadır. 0+"" işlem sonucu String olarak 0 ifadesini ekrana basar.
42. Cevap A. main() metodunda TestGame sınıfından oluşturulan nesne referansı ile runTest() metoduna sırasıyla false ve true parametreleri gönderilmektedir. runTest() metodunda ilk olarak spinner = roller kontrolü yapılmaktadır. roller değişkenindeki true değeri spinner'a yüklenir ve if kontrolü içinden "up" sonucu geri döner. Ekran çıktısı "up" olur. Eğer if (roller == spinner) olarak implemente edilmiş olsaydı, else bloğu çalışacak ve roller'a false yüklendiği için "middle" değeri ekrana basılacaktı.
43. Cevap D. İki operandtan bir tanesinin true olması durumunda true sonucunu garanti eden operatör OR (||) operatörüdür. boolean bir değeri tersine çeviren operatör ise DEĞİL (!) operatörüdür.  
Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/opsummary.html>
44. Cevap A. movieRating değişkenine değer atanması işleminde ilk olarak characters değişkeninin 4'ten küçük ya da eşit olup olmadığı kontrol edilir. characters değişkeninde 5 değeri yüklü olduğundan dolayı false döner ve üçlü kontrol yapısının ikinci kısmı icra edilir. İkinci kısımda yine bir üçlü kontrol yapısı mevcuttur. Burada ise story değişkeninin 1'den büyük olup olmadığı kontrolü sonucu true dönerek movieRating değişkenine 2 değeri yüklenir. movieRating noktalı bir sayı olduğu için ekran çıktısı alındığında ya da Debug modda incelendiğinde 2.0 değeri ile yüklendiği görülür.
45. Cevap B. switch yapısından herhangi bir (sıfır ya da fazla) sayıda case ifadesi ve en fazla bir tane default ifadesi yazılabilir. Hiç case ifadesi yazılmayabilir ve default ifadesi implemente edilmeyebilir.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>

46. Cevap A. Komut satırı argümanı verilerek test edildiğinde ilk parametre “sunny” değeri olursa kontroller sonucu ekran çıktısı olarak “Go Outside” basılacaktır. Farklı herhangi bir parametre verilmesi durumunda ise “Stay Inside” basılacaktır. Ancak herhangi bir parametre verilmeden run edilirse weather parametresine değer yüklenmeyeceği için bu dizinin boyutu olmayacak ve 0. indis diye bir yer olmayacağından dolayı ArrayIndexOutOfBoundsException fırlatacaktır. Bu hatta illegal indis hatasıdır.

Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/lang/ArrayIndexOutOfBoundsException.html>

47. Cevap D. int veri tipi için DEĞİL (!) operatörü geçersiz bir argümandır. Bu nedenle “!2” işleminde hata alınır. Bunlardan hiçbirisi ekrana basılmaz.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

48. Cevap C. Her iki boşluğa da true gelecektir.

OR (  ) Operator	w = true	w = false
z = true	false	true
z = false	true	false

Tablo-2: OR Operatörü Doğruluk Tablosu

Kaynak: [https://www.cs.utah.edu/~germain/PPS/Topics/truth\\_tables\\_and\\_logic.html](https://www.cs.utah.edu/~germain/PPS/Topics/truth_tables_and_logic.html)

49. Cevap A. -, +, /, \* ve % operatörleri aynı seviye yer almaktadırlar.

Kaynak: <https://introcs.cs.princeton.edu/java/11precedence/>

50. Cevap C. Verilen p1 numaralı satırdan dolayı derlenmez. game değişkeni String bir değişkendir. Üçlü kontrol ifadesinde game değişkenine int türünden bir değer gönderilmek istenmektedir. Üçlü kontrol ifadesinde bu türlerin aynı olması gerektiği için bu satırdan kod derlenmez.

Kaynak: <https://www.baeldung.com/java-ternary-operator>