

1. Cevap B. Java'da main() metoduna komut satırı argümanları için verilen parametre "String[]" paramName" yada "String... paramName" şeklinde gösterimlerle tanımlanır. Her iki kullanımda da java.lang.String sınıfından bir dizi parametre olarak gönderilir.
Kaynak: <https://docs.oracle.com/javase/8/docs/technotes/guides/language/varargs.html>
<https://www.baeldung.com/java-varargs>
2. Cevap B. Komut satırı argümanlarından ilkinde ulaşmak için boşluğa gelmesi gereken ifade B seçeneğindedir. Java'da dizilerin ilk elemanı 0. indiste konumlanır.
Kaynak: <https://www.geeksforgeeks.org/variable-arguments-varargs-in-java/>
3. Cevap D. Java'da tüm diziler Object sınıfına ait olduğundan dolayı ne uppercase dizisi, ne de lowercase dizisi ilkel tipte değildir.
Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>
https://www.tutorialspoint.com/java/java_arrays.htm
4. Cevap C. Verilen ifadeler ilki geçerli bir dizi deklarasyonu değildir. Diğer iki ifade geçerli dizi deklarasyonlarıdır.
Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>
https://www.tutorialspoint.com/java/java_arrays.htm
5. Cevap C. printStormNames metodu parametre olarak dizi beklemektedir. Ancak gönderilen parametre dizi değildir. Bu nedenle C seçeneğindeki metod çağırısı derlenmez.
Kaynak: <http://www.mathcs.emory.edu/~cheung/Courses/170/Syllabus/09/array-param.html>
6. Cevap A. Java'da bir dizinin boyutu yani dizi kaç eleman yer aldığı bilgisi length özelliği ile elde edilir.
Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>
<https://www.geeksforgeeks.org/length-vs-length-java/>
7. Cevap C. Java'da çok boyutlu diziler C seçeneğinde gösterilen şekilde tanımlanır. Java maksimum 255 boyutlu dizi oluşturmaya olanak sağlamaktadır.
Kaynak: <https://www.geeksforgeeks.org/multidimensional-arrays-in-java/>
<https://stackoverflow.com/questions/4060538/maximum-number-of-dimensions-in-a-java-array>
<https://www.quora.com/How-many-maximum-dimensions-can-an-array-have-in-C-C++-or-Java>
8. Cevap B. Soruda tanımlanan tek boyutlu dizi 7 elemana sahiptir ve length değeri 7 olduğundan dolayı 7 satır çıktı üretecektir.
9. Cevap B. Java'da bir dizide ikili arama yapmak için java.util.binarySearch() metodu ve diziyi sıralamak için java.util.sort() metodu kullanılabilir.
Kaynak: https://www.tutorialspoint.com/java/util/arrays_sort_object_index.htm
https://www.tutorialspoint.com/java/util/arrays_binarysearch_object.htm
10. Cevap B. java.util.Arrays.sort() metodu parametre olarak verilen ifadeyi türü Object olduğu için alfabetik olarak sıralar. Dolayısıyla sıralama 1, 9, 10 değil, 1, 10, 9 olacaktır.
Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>
11. Cevap B. Java bir dizinin ilk elemanı 0'inci indiste, son elemanı (boyut-1)'inci indisinde konumlanır. Sorudaki dizinin ilk elemanı trains[0] ve son elemanı trains[trains.length-1] olacaktır.
Kaynak: <https://www.geeksforgeeks.org/find-first-and-last-element-of-arraylist-in-java/>
<https://www.tutorialspoint.com/how-to-get-first-and-last-elements-from-arraylist-in-java>
12. Cevap C. Verilenlerden lion ve bear dizileri için verilen deklarasyonlar geçerlidir. tiger ve ohMy dizilerinin deklarasyonlarında dizi boyutu belirtildiği için bunlar geçersizdir. İlk değer ataması

ile yapılan dizi deklarasyonlarında boyut verilmez, süslü parantezler içine verilen initial değerlerle dizi boyutu ayarlanmış olur.

Kaynak: <http://grails.asia/java-string-array-declaration>

13. Cevap B. Verilenlerden sadece tiger dizi deklarasyonu geçerlidir. İlk değer yüklemesi olmadan yapılan dizi tanımlamasında boyut belirtilmeli ve boyut ifadesi veri tipinden sonra gelmelidir.

Kaynak: <http://www.hudatutorials.com/java/basics/java-arrays/java-float-array>

14. Cevap C. binarySearch() metodu ile kesin bir sonuç almak için önce dizi sıralanmalıdır. Bunu en doğru şekilde ifade eden seçenek C seçeneğidir.

Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>

15. Cevap A. Java'da bir dizi eleman eklendikçe ya da dolduğu anda otomatik olarak genişlemez. Bu nedenle A seçeneği diziler için doğru ifade değildir. Bir dizide aynı elemandan başka bir indekste bulunabilir. Diziler sort() metodu ile sıralı elemanlardan oluşacak şekilde düzenlenebilir. Dizin ilk elemanı 0'ıncı indekste barınır. Bu bilgilere göre B, C ve D seçenekleri doğrudur.

Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>

16. Cevap C. Verilen kod parçasında ilk olarak m3 numaralı satırda ArrayIndexOutOfBoundsException hatası alınır. Verilen 2 boyutlu dizinin ilk boyutu 1 elemanlı, ikinci boyutu 2 elemanlıdır. Dolayısıyla ilk boyut için [1], ikinci boyut için [2] ve sonrası indeksler dizinin dışındadır.

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

17. Cevap B. Verilen os dizisi parametresi Object olduğu için alfabetik olarak sıralanır ve sonuç Linux, Mac, Windows şeklinde olur. Dizin sort() metodu ile sıralanması sonrası binarySearch() metodu ile "Mac" ifadesi arandığında sıralı dizinin 1'inci indisinde bu ifade olacağı için binarySearch() metodu 1 değeri ile geri döner ve ekran çıktısı 1 olur.

18. Cevap A. Verilen kod parçasının r1 satırında hatalı bir dizi tanımlaması vardır ve bundan dolayı kodun derlenmesi önlenir. 2 boyutlu dizinin boyutu [3, 3] şeklinde değil [3][3] şeklinde olacaktır. Bu hata giderildikten sonra ise r2 satırında dizinin 3'üncü indeksine değer atanmaya çalışılması ArrayIndexOutOfBoundsException hatasına yol açacaktır.

Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>

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

19. Cevap B. İlk olarak oluşturulan lotto dizisi bir objedir. 4 elemanlı olarak oluşturulan dizinin tüm elemanları null olarak atanmıştır ve elemanları obje değildir. Fakat, dizi deklarasyonu sonrası dizinin ilk 2 indeksine değer ataması yapılmaktadır. Atanan bu değerler yeni birer referansa sahip olduklarından 2 yeni obje elde edilmiş olunur. Çalışma zamanında 3 tane nesne yaratılmış olur.

20. Cevap B. Verilen dizi deklarasyonlarından alpha ve beta dizi deklarasyonları geçersizdir. Dizi tanımlaması yapılırken köşeli parantezler veri tipinden önce gelemez. Ya veri tipinden sonra ya da dizi isminden sonra yazılabilir. Bu nedenle gamma, delta ve epsilon dizilerine ait deklarasyonları geçerlidir.

Kaynak: <https://www.geeksforgeeks.org/array-declarations-java-single-multidimensional/>

21. Cevap B. Verilen blocks dizisinin 0. indisinde 3 eleman, 1. indisinde 1 eleman ve 2. indisinde 2 eleman yer almaktadır. Bu diziyi en doğru şekilde ifade eden grafiksel gösterim B seçeneğindedir. 3x3 boyutlu bu dizinin diğer boşta kalan indeksleri yaratılmadığı için bu noktalar dizi dışıdır. Kullanılması durumunda ArrayIndexOutOfBoundsException hatası fırlatır.

22. Cevap D. Verilen metod derlenir. Ancak boş ya da null olmayan bir dizi ile çağrıldığında metodun ilk satırında dizinin sondan bir sonraki indeksine veri yazmaya çalışması nedeniyle ArrayIndexOutOfBoundsException fırlatır. Bu nedenle cevap D seçeneğidir.

23. Cevap C. Dizinin yaratılması sonrası for döngüsü içinde dizinin boyutun size() metodu ile elde edilmeye çalışılmaktadır. Java'da dizilerin boyutları length özelliği ile Collections framework'ün sınıflarında yer alan veri yapılarının boyutları size() metodu ile elde edilir. Bu nedenle derleme hatası vardır. Kod derlenmez.
Kaynak: <https://stackoverflow.com/questions/20192843/difference-between-size-and-length-methods>
24. Cevap C. Hem bools dizisi hem moreBools dizisi 3 boyutlu olur. Ancak ikisi de ön yüklenmediği için boyutların genişlikleri belirsizdir.
Kaynak: <https://www.geeksforgeeks.org/array-declarations-java-single-multidimensional/>
25. Cevap C. Verilen dizi tek boyutlu ve 2 elemanlı olarak oluşturulmuş ve değer ataması yapılmamıştır. Bu dizinin 2 elemanı da null olarak belirlenecektir. Ekran çıktısı alınan dizi ismi Object türünden bir nesnenin referansı olduğu için dizi ismi ile çıktı alındığında C seçeneğindeki benzer şekilde bir adres bilgisi verecektir.
Kaynak: <https://users.scala-lang.org/t/what-is-a-ljava-lang-string-and-how-to-print-its-content-human-readably/1088/2>
26. Cevap B. Verilen kod parçasında r2 numaralı satırda 3. indekste yer alan yere değer yazılmak istenmektedir. Dizi 3x3 boyutlu matris olarak tanımlanmış ve 3. indeksli noktalar dizinin dışındadır. Bu nedenle bu noktalara değer yazılmaya çalışılması sonucu ArrayIndexOutOfBoundsException hatası fırlatacaktır.
Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/lang/ArrayIndexOutOfBoundsException.html>
27. Cevap D. String... original şekliyle yapılan dizi tanımlaması metoda parametre olarak verildiğinde geçerlidir. Bu nedenle copy dizisi tanımlanırken "..." ile tanımlama yapılamaz ve kod derlenmez.
Kaynak: <https://www.geeksforgeeks.org/variable-arguments-varargs-in-java/>
<https://softwareengineering.stackexchange.com/questions/267534/string-args-vs-string-args>
28. Cevap D. Kod derlenir. Ancak çalışma zamanında 8. satırdan dolayı ArrayStoreException fırlatır. Bu hata dizi elemanına uyumsuz türde veri atanmak istendiğinde alınmaktadır. Verilen kodda 2x2 boyutlu game dizisi tek boyutlu olarak tanımlanan obj dizisine atanmaktadır. Bu tehlikeli bir durum olsa da Java buna izin verdiği için derleme hatasına yol açmamaktadır.
Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/lang/ArrayStoreException.html>
<https://www.geeksforgeeks.org/arraystoreexception-in-java/>
http://javayaz.com/?page_id=100
29. Cevap C. Verilen dizide olmayan bir değer arandığında negatif değeri ile dönüş yapılır. Verilen kod çalıştırıldığında 0. indeksten başlanarak arama yapılacak bulamadığı eleman adedince 0'dan düşerek -3 sonucunu dönecektir.
Kaynak: <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Arrays.html>
30. Cevap B. Verilen kod örneği terminalde derlenip komut satırı argümanı olarak Wolfe değeri çalıştırıldığında Wolfie sonucunu ekrana basacaktır. Program adından sonraki her bir ifade names isimli komut satırı argüman listesinin sıralı indekslerine atanacaktır.
Kaynak: <https://docs.oracle.com/javase/tutorial/essential/environment/cmdLineArgs.html>
31. Cevap C. Verilen kod örneği komut satırında derlenip 1 ve 2 komut satırı argümanları ile çalıştırıldığında main() metoduna gönderilen dizinin boyutu 1 ve 2 değerleri ile 2 olacaktır. Terminal ekranı çıktısı 2 olacaktır.
Kaynak: <https://docs.oracle.com/javase/tutorial/essential/environment/cmdLineArgs.html>
32. Cevap B. Verilen kod örneğinde komut satırı argümanları sırasıyla seed ve flower'dır. Dolayısıyla args[0] = seed'dir ve one isimli String değişkenine "seed" değeri yüklenir. Sonrasında dizi sort() metodu ile sıralandığında 0. indekste "flower", 1. indekste "seed" değeri

olacaktır. `binarySearch()` metodu ile `one` değişkenine yüklenen “seed” değeri dizinin 1. indeksinde yer aldığı için ekran çıktısı 1 olacaktır.

33. Cevap D. Verilen ilk 3 dizi tanımlaması birbirinin aynısıdır, gösterim olarak farklıdır. İlk 3 dizi de 2 boyutlu ilk boyutu 2 ikinci boyutu 1 elemanlı olacak tasarlanmıştır. Ancak son dizinin ilk boyutu 1 ikinci boyutu 2 elemanlıdır.

Kaynak: <https://www.oreilly.com/library/view/learning-java-4th/9781449372477/ch04s07.html>

34. Cevap C. Bir dizinin her bir elemanına kendi nümerik indeksi ile erişilir. Bu nedenle A ve B seçeneklerindeki gibi yöntemlerde dizi elemanına erişilemez. Java’da dizi indekslemesi 0 ile başlayacağından dolayı `z` değerine erişebilmek için kullanılması gereken yöntem `dimension[2][2]` şeklinde olacaktır.

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

35. Cevap D. Java’da bir dizide indeksleme 0 ile başlar `diziAdi.length` değerinin 1 eksiğine kadar devam eder. Verilen kod örneğinde `days.length` değeri 7 dir. `for` döngüsünde ise 1’den başlanarak 7 dahil olmak üzere 7 iterasyon boyunca çalıştırıldığında ilk 6 satırda normal şekilde alt alta sırasıyla,

Monday
Tuesday
Wednesday
Thursday
Friday
Saturday

değerleri basıldıktan sonra `ArrayIndexOutOfBoundsException` fırlatır. Çünkü `days[7]` değeri dizi dışındadır.

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

36. Cevap C. Verilen kod terminalde derlenerek komutu satırı argümanı olarak Wolfie ile çalıştırıldığında `ArrayIndexOutOfBoundsException` hatası fırlatır. Çünkü, argüman olarak girilen Wolfie değeri `args` dizisinin 0. indeksine konumlanır. 1. İndeks dizi dışında olacağı için bu hatayı fırlatır.

37. Cevap D. Verilen kodda herhangi bir söz dizimi hatası yoktur. Bu nedenle sorunsuz olarak derlenir. Çalıştırıldığında dizinin belirtilen indekslerine “X” değeri yazılır. Ekran çıktısı alınan satırda `ticTacToe` dizisinin boyutu 3 ve yanına “ in a row!” ifadesi yazılacaktır.

38. Cevap D. Dizinin eleman sayısını gösteren özellik `length`’dir. Kodda `length()` metodu ile çekilmeye çalışılmaktadır ve bu yüzden derleme hatası alınır.

39. Cevap B. Verilen kod örneğinde `boolean[][]` olarak tanımlama yapılırken yanına yazılan dizi otomatik olarak 2 boyutlu olacaktır. Bununla beraber dizi ismi yanına eklenen her bir köşeli parantez çiftli ile dizi boyutu 1 artarak büyür. Bu nedenle `bools` dizisi 3 boyutlu olarak, `moreBools` dizisi 2 boyutlu olarak tanımlanmış olunur.

Kaynak: <https://www.geeksforgeeks.org/array-declarations-java-single-multidimensional/>

40. Cevap B. Komut satırında herhangi bir argüman verilmediği için `args` dizisinde eleman yoktur. Yine de `args` dizisi `sort()` metodu için geçerli bir dizidir ve sıralama sonucu “[]” şeklinde dönecektir. Bu dizi ekran çıktısı olarak basıldığında B seçeneğindeki ifade görülecektir.

41. Cevap D. Verilen kod parçasındaki `os` dizisi sıralanmadan `binarySearch()` metoduna gönderildiği için “The output is not defined.” şeklinde çıktı alınır.

42. Cevap B. Verilen kodda `int` türünden `game` dizisinin `[3][3]` indeksine “X” şeklinde String değer atanmaya çalışılan noktada `type mismatch` hatası ile karşılaşılır ve bu nedenle kod derlenmez.

Kaynak: <https://docs.oracle.com/javase/7/docs/api/org/omg/CORBA/DynAnyPackage/TypeMismatch.html>

43. Cevap A. Verilen listing dizisi 2x1 boyutunda bir dizi olarak tanımlanmış ve tanımlandığı anda ön yüklenmiştir. listing[0][0]="Book", listing[1][0]="Game" ve listing[1][1]="29.99" değerleri ile yüklü olacaktır. listing dizisi 2 elemanlı olduğundan length değeri 2'dir. 0. indekste bir eleman vardır ve burada length değeri 1'dir Dolayısıyla ekran çıktısı "2 1" şeklinde olacaktır.
44. Cevap C. Verilen kod derlendikten sonra komut satırı argümanı verilmeden çalıştırıldığında names dizisinin 0. indeksine herhangi bir değer atanmadığı ve bu değere erişilmeye çalışıldığı için ArrayIndexOutOfBoundsException hatası fırlatır.
45. Cevap A. Verilen kod sorunsuz şekilde derlenir ve çalışır. Döngüde 1. indeksten başlanarak 7. indeksin bir öncesine kadar toplam iterasyon boyunca dönerek alt alta sırasıyla,
- Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
- Değerleri basılır.
46. Cevap B. Verilen kod komut satırında derlendikten sonra "1 2" değeri ile çalıştırıldığında program bu değeri String türünden 1 adet eleman olarak algılar. Bu nedenle length değeri 1 olur ve ekrana 1 değeri basılır.
47. Cevap A. Verilen kod derlenip çalıştığında dizinin ilk elemanı olan "Linux" ifadesi dizide arandığında binarySearch() metodu 0 indeks değerini döner. Ekran çıktısı 0 olacaktır.
48. Cevap A. Bir metod içinde, bir dizi parametresi ve bir varargs parametresi aynı şekilde ele alınır. Çağırın kod için bir dizi parametresi daha kısıtlayıcıdır. Her iki tür de bir dizi alabilir. Ancak, yalnızca bir varargs parametresinin tek tek parametreleri otomatik olarak bir diziye dönüştürmesine izin verilir. Bu nedenle, l ifadesi doğrudur ve cevap Seçenek A'dır.
49. Cevap B. A seçeneğindeki num1a ve num1b dizileri 4 boyutludur. C seçeneğindeki num3a ve nums3b dizileri 4 boyutludur. D seçeneğindeki num4a ve nums4b dizileri 4 boyutludur. Ancak B seçeneğindeki nums2a dizisi 4 boyutlu iken, nums2b dizisi 3 boyutludur.
50. Cevap C. Verilen kodda binarySearch() metodundan dönen int türünden değer String bir değişkene atanmaya çalışıldığı anda type mismatch hatası alınacaktır ve kod derlenmeyecektir. Değişkenin türü int olarak güncellendiğinde sıralanmış dizinin ilk elemanı dizi de arandığında 0. indekste yer aldığı için sonuç 0 olacaktır.

The Zen of Python

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than **right** now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea -- let's do more of those!