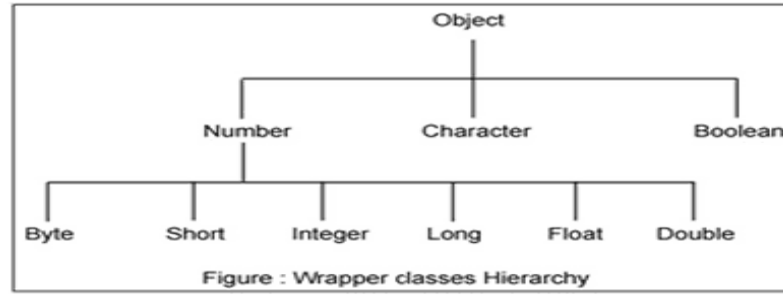


1. Cevap A. A seçeneğinde bir değişken tanımlanması sonrasında farklı türde bir değişken tanımlaması yapılırken noktalı virgül (;) kullanılmaması nedeniyle bu deklarasyon geçersizdir. B, C ve D seçeneklerinde aynı türden birden fazla değişken tek satırda deklare edilmektedir ve değişkenlere ilk değer ataması yapılabilir.
Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
2. Cevap D. İlk değer ataması yapılmayan chair değişkeninin sonuna, tanımlandığı satırın bir alt satırında ilk değer ataması yapılan table değişkeni eklenmek istenmektedir. Ancak, Java'da ilk bir string değişkeni başka bir string değişkene eklemek için null olarak tanımlamak ya da "" ifadesi boş olarak deklare etmek gerekir.
Kaynak: <https://www.baeldung.com/java-string-initialization>
3. Cevap B. String türünde oluşturulan bir instance değişkenin ilk değeri null olarak yüklenir. Bu nedenle doğru seçenek B'dir. Boş bir String değer ile yüklenmez. Instance değişkenlerinin default değerleri vardır. Ayrıca deklare edildiği yerde yüklenmesine ihtiyaç olmadığı için A, C ve D seçenekleri yanlıştır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>
4. Cevap B. Java'da isimlendirme standardı olarak ilk karakter büyük veya küçük olmak üzere bir harf, alt çizgi (_) ya da dolar (\$) karakteri olabilir. Özel karakterler veya sayılarla başlayamaz. Bu nedenle verilenlerden "2blue" ifadesi bir isim olmayacaktır. A, C ve D seçeneklerindeki ifadeler birer değişken ismi olabilir.
Kaynak: <https://www.tutorialspoint.com/Java-Naming-conventions>
5. Cevap B. Java'da sınıf (class) isimleri büyük harf ile başlar ve küçük harf ile devam eder. Sınıf adı birden fazla kelimedenden oluşacaksa, her kelimenin ilk harfi büyük devamındaki harfleri küçük harf olacaktır. Bu nedenle Java isimlendirme kurallarına en uygun seçenek B'dir. Diğer seçenekler bu standardın dışındadır.
Kaynak: <https://www.oracle.com/technetwork/java/codeconventions-135099.html>
6. Cevap C. Verilen metot yapılarından ilki olan int türü parametre olarak String değer döndürmesi amaçlanan ifade derlenmeyecektir. Çünkü ilkel olan int türü toString() metodunu implemente etmemektedir. Buradaki problemi int türünden değişkene boşluk ekleyerek çözebiliriz. toString() metodu Object sınıfında implemente edildiği için buradan türeyen referans sınıflara ait instance'lar bu metodu çağırabilecektir.
Kaynak: <https://www.geeksforgeeks.org/object-tostring-method-in-java/>
<https://www.wideskills.com/java-tutorial/java-tostring-method>
7. Cevap C. Java'da nümerik değişkenlerde alt çizgi değerinin başında bulunmamaktadır. Bu nedenle C seçeneğindeki ifade derlenmez.
Kaynak: <https://docs.oracle.com/javase/8/docs/technotes/guides/language/underscores-literals.html>
8. Cevap C. Java'da kapsayıcı (wrapper) sınıflar, ilkel veri tiplerini nesneye ve nesneleri ilk veri tiplerine dönüştürmek için mekanizma oluşturan sınıflardır. 8 ilkel veri türüne karşılık gelen 8 tane kapsayıcı sınıf vardır. Bunlar; Byte, Short, Integer, Long, Float, Double, Boolean ve Character sınıflarıdır. int, ilkel bir veri tipidir. Java'da Int şeklinde bir veri tipi yoktur. Object sınıfı ise herhangi bir ilkel türe map etmediği için kapsayıcı sınıf değildir.



Kaynak: <https://www.javatpoint.com/wrapper-class-in-java>

<https://www.injavawetrust.com/pure-java-33-wrapper-classes-01/>

9. Cevap C. Kod parçasındaki integer türünde değişkenler yaratılmak istenmektedir. Ancak integer şeklinde bir değişken tipi olmamasından dolayı kod derlenmez. Bu tip, Integer olarak güncellenirse sonuç B seçeneği olurdu.

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

10. Cevap C. Java'da new anahtar sözcüğü ile bir sınıfın kurucu metodu çağrılabilir veya bir sınıftan bir instance örneklenebilir. Bu ifadeye en uygun karşılık C seçeneğidir. Bu anahtar sözcükle mevcut bir nesneden yeni bir tane kopya edilmez. İlkel bir türden örnekleme yapmaz. Bir nesnenin referansını yeni bir tane nesneye aktarmaz. Bu nedenle A, B ve D seçenekleri doğru değildir.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/javaOO/objectcreation.html>

<https://www.javatpoint.com/new-keyword-in-java>

11. Cevap D. Java'da bir veri tipi kendisinden daha geniş bir veri tipine atanabilir. Ancak daha büyük veri tipleri daha küçük olan tiplere atanırken veri kayıpları olabilir. Bu bilgilere göre seçenekler incelenir. A seçeneğinde float bir değer double türünden d1 değişkenine atanmıştır ve derleme hatasına yol açmaz. B seçeneğinde 5.0d olarak belirtilmese de 5.0 değeri double değerdir ve double türünden d2 değişkenine atanmıştır ve derleme hatasına neden olmaz. C seçeneğinde float bir değer float türünden f1 değişkenine atanmıştır ve derleme hatasına neden olmaz. Ancak, D seçeneğinde double bir değer float türünden f2 değişkenine atanmak istenmiştir ve 32 bitlik alana 64 bitlik veri yazılamayacağından dolayı p4 satırında derleme hatası verir.

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

12. Cevap A. Java'da ilkel veri tiplerinin genişlikleri ve aralıkları aşağıdaki tabloda olduğu gibidir. Tabloya göre bakıldığında sıralama byte, char, float ve double olacaktır. D seçeneğinde gösterilen bigint şeklinde bir ilkel tip ise yoktur.

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

Tablo-1: İlkel Veri Türleri

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
https://www.w3schools.com/java/java_data_types.asp

13. Cevap D. Java'da bir sınıfa ait örnek değişkenleri, metotlar ve kurucu metotların sıralamasının bir önemi yoktur. Tüm seçenekler geçerlidir ve doğru D seçeneğidir.

Kaynak: <https://checkstyle.sourceforge.io/apidocs/com/puppycrawl/tools/checkstyle/checks/coding/DeclarationOrderCheck.html>

14. Cevap B. x2 numaralı satırda iki farklı tipte değişkeni tanımlaması yapılmaktadır. Ancak int türünden değişken tanımlı yapıldıktan sonra double türünden değişken tanımlaması öncesi kullanılacak olan operatör noktalı virgül olmalıdır. Virgül operatörü sadece aynı türden çoklu değişken tanımlamaları yapılacaksa kullanılabilir. Eğer, bu derleme problemi çözülmüş olsaydı, x3 numaralı satırda değer ataması yapılmamış olan hot değişkeni ile title değişkenin toplanması sırasında hata alınacaktı ve Cevap C seçeneği olacaktı. Fakat ilk planda cevap B seçeneğidir.

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

15. Cevap C.

Java'da iki tür initialization blok vardır; static initialization block ve instance initialization block. **Static Initialization Block:** static anahtar sözcüğü ve süslü parantezlerle implemente edilir. Bir sınıf ilk yüklendiği anda sadece bir kez çalışır. Bir sınıf içerisinde birden fazla static initialization block yazılabilir.

Instance Initialization Block: Anahtar sözcük olmaksızın süslü parantezlerle implemente edilir. Sınıftan her bir örnek (instance) oluşturulduğunda çalışır.

Verilen kod örneğinde 2 ve 7 numaralı satırlardaki bloklar instance initialization blocks'tır. 6 numaralı satırdaki ifade static initialization block'tır. 3 ve 5 numaralı satırlar arasındaki ifade kurucu (constructor) metottur. Bu bilgilere göre kod incelendiğinde 2 tane instance initializer olduğu görülmektedir.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/javaOO/initial.html>
<https://www.injavawetrust.com/pure-java-32-initialization-blocks/>

16. Cevap A. defaultValue değişkeni, yerel değişken olmasından dolayı otomatik olarak initialize edilmez. Bu nedenle, ilkel veri tipi ne olursa kod derlenmez. Eğer bu değişken örnek(instance) değişkeni olarak tanımlanmış olsaydı, int ve short türü için 0 sonucunu; double türü için 0.0 sonucunu üretecekti ve cevap C seçeneği olacaktı.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
<https://www.tutorialspoint.com/default-value-of-primitive-data-types-in-java>

17. Cevap A.

finalize() metodu Object sınıfında yer alır. Object sınıfı, Java'da sınıf hiyerarşisinin tepesinde yer alır ve tüm sınıflar Object sınıfını kalıttığı için her sınıfın bir örneği (instance) bu metodu bulunduracaktır.

Garbage Collection mekanizması, Garbage Collector(GC) ile heap'ten ilgili nesneyi silmeden önce, finalize() metodunu çağırır. Bunun nedeni, nesne silinmeden önce yapılaması gereken bir işlem olabilir. Dosya kapatma işlemi ya da silinmek üzere olan nesne silinmeden kurtarılabilir. finalize() metodu GC tarafından sadece bir kez çağrılabilir. Eğer GC bir nesneyi silmeye karar verirse ve finalize() metodu kod içerisinde çağrılmadıysa, GC finalize() metodunu çağırma garantisi eder. GC, her nesneyi silmeyi garanti etmez. Dolayısıyla finalize() metodu hiç çalışmayabilir.

Bu bilgilere göre finalize() metodu bir ya da daha fazla çağrılabilir.

Kaynak: [https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#finalize\(\)](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html#finalize())
<https://www.injavawetrust.com/pure-java-38-the-finalize/>

18. Cevap D. Verilenlerden String sınıfı, kapsayıcı(wrapper) sınıf değildir.

Kapsayıcı sınıflar; Byte, Short, Integer, Long, Float, Double, Boolean ve Character sınıflarıdır.

Kaynak: <https://www.javatpoint.com/wrapper-class-in-java>

<https://www.injavawetrust.com/pure-java-33-wrapper-classes-01/>

19. Cevap C. Verilen kod parçasının diyagrama dökülmüş hali C seçeneğidir. Link sınıfının instance'ları oluşturulduktan sonra link2 ve link3 çift yönlü olarak birbirine bağlanmıştır. Sonrasında link1 ve link3 instance'larının null olarak atanmıştır.

20. Cevap C. Verilen noktalı sayı f harfi ile belirtilmediği için bu sayı double'dır. Boşluğa double yazılır. float yazabilmek için f harfi ile markalanmalıdır. double türü 8 byte ve float türü 4 byte olduğundan dolayı double değer float veri tipine yazılamaz. Daha büyük veri, daha küçük veri tipine yazılırken casting yapılabilir ancak bu durumda veri kayıpları olacaktır.

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

21. Cevap B. Kaynakta verilen isimler Java'da değişken ismi olarak kullanılamaz. Bir değişkene int şeklinde bir isim verilemeyeceği kodda ilk derleme hatasına yol açacak satır k2 satırıdır.

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

22. Cevap B. foo bir sınıfın örneğine referans ediyorsa bir nesnedir. Demek ki, bar isminde bir özellik implemente edilmiştir. Bu nedenle bar bir instance değişkenidir, yerel değişken değildir. Doğru olmayan seçenek sorulduğuna göre A'daki bilgi doğrudur ve A seçeneği yanlıştır. B'deki bilgi yanlış ve sorunun cevabı B seçeneğidir. Okuma yazma işlemleri için bar değişkeni kullanılır. C ve D seçenekleri de doğru ifadedir.

23. Cevap C. Java'da isimlendirme standardı olarak ilk karakter büyük veya küçük olmak üzere bir harf, alt çizgi (_) ya da dolar (\$) karakteri olabilir. Özel karakterler veya sayılarla başlayamaz. Bu nedenle verilenlerden "5MainSt" ifadesi bir isim olmayacaktır. A, B ve D seçeneklerindeki ifadeler birer sınıf ismi olabilir.

Kaynak: <https://www.tutorialspoint.com/Java-Naming-conventions>

24. Cevap D. Soruda verilen d değişkeni, new anahtar sözcüğü ile türetildiğine ilkel bir tipte değildir. Dolayısıyla her iki boşluğa Double sınıf ismi yazılmalıdır. Ayrıca, alt çizgi(_) karakteri iki nümerik dijit arasına gelebilir. Nokta yanında yer almasından dolayı hata verecektir.

Kaynak: <https://docs.oracle.com/javase/8/docs/technotes/guides/language/underscores-literals.html>

25. Cevap C. Yerel olarak tanımlanan String türünden bir değişkene ilk değer ataması yapılmazsa otomatik olarak değer atanmaz. Değer atanmadığı gibi bu değişkenin değeri null ya da boşluk da değildir. Ayrıca, deklare edildiği yerde initialize edilmemesi de derleme hatasına yol açmaz. Bu bilgilere göre A, B ve D seçenekleri yanlış, C seçeneği doğrudur.

Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>

26. Cevap C. İlkel veri tipleri yerel değişken olarak tanımlandıklarında default değerle initialize edilmezler. Ancak örnek(instance) değişkeni olarak tanımlandıklarında default değerle initialize edilirler. Verilen veri tiplerinden short, int ve long tiplerinin default değerleri 0 olduğundan dolayı doğru seçenek C'dir. double değişkeninin default değeri 0.0 olduğu için hariç tutulur.

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

<https://www.tutorialspoint.com/default-value-of-primitive-data-types-in-java>

27. Cevap B. Sadece bir nesne üzerinde metotlar çağrılabilir. Bu nedenle A seçeneği yanlıştır. valueOf() metodu bir stringden kapsayıcı bir sınıfın instance'ına dönüştürmek için kullanılır. Bu nedenle C seçeneği yanlıştır. D seçeneğinden bahsedildiği üzere Java'da ilkel bir veri tipini doğrudan ArrayList'e ekleme yoktur. Autoboxing yapılarak ilkel veri önce nesneye dönüştürülür. Daha sonra ArrayList'e eklenir. B seçeneği ise bir autoboxing örneğidir. Java ilkel veri tipini kapsayıcı sınıfa dönüştürürken otomatik olarak değer ataması yapar.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/data/autoboxing.html>

<https://www.geeksforgeeks.org/autoboxing-unboxing-java/>

28. Cevap C. Kod derlenmez. Çünkü, i değişkeni ilkel bir veri tipidir. Java ilkel veri tiplerinin metot çağrısı yapmasına izin vermez. Bu nedenle kod son satırda derleme hatası verecektir.
Kaynak: <http://www.cs.toronto.edu/~reid/web/javaparams.html>
29. Cevap D. Terminale "bounce" basması için TennisBall sınıfının kurucu metodunu çağırmak gerekmektedir. Bir sınıfın kurucu metoduna çağrı yapmak için new anahtar sözcüğü ve kurucu metodun adı yazılır. Doğru yazım, D seçeneğindeki "new TennisBall();" ifadesidir.
Kaynak: <https://www.geeksforgeeks.org/constructor-chaining-java-examples/>
<https://docs.oracle.com/javase/tutorial/java/javaOO/objectcreation.html>
30. Cevap A. I numaralı ifadede cat ve dog değişkenlerine "animal" değeri atanmaktadır. II numaralı ifadede dog değişkeninin veri tipi belirtilmediği için söz dizimi(syntax)/sentaks hatası vardır ve dog değişkenine istenen değer atanamayacaktır. III numaralı ifadede cat değişkenine ilk değer ataması yapılmadığı için bu değişken içerisinde herhangi bir değer yoktur. IV numaralı ifadede tanımlanan iki değişken arasında noktalı virgül işareti kullanılması gerekirken virgül kullanılmıştır ve söz dizimi/sentaks hatasıdır. Buradaki virgül, noktalı virgüle evrilirse söz dizimi hatası düzeltilir, ancak yine de istenen değer cat değişkenine atanmaz. Söz dizimi hatasını gidermenin bir diğer yöntemi de dog değişkeni önündeki String kelimesinin silinmesidir. Bu durumda yalnız I numaralı seçenek istenen değerlerin set edildiği seçenektir.
Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
<https://www.geeksforgeeks.org/variables-in-java/>
31. Cevap C. Java'da ilkel veri tipi ve kapsayıcı sınıf ismi arasında farklılık olan iki ilkel veri tip char ve int veri tipleridir. Bu veri tiplerinin kapsayıcı sınıfları Character ve Integer sınıflarıdır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/sql/Wrapper.html>
<https://www.javatpoint.com/wrapper-class-in-java>
<https://www.injavawetrust.com/pure-java-33-wrapper-classes-01/>
32. Cevap A. String türünden instance değişkenine null atanabilir. Bu yüzden A doğrudur. Sınıf dışından değeri set edilebilir. Bu nedenle B yanlıştır. Kurucu metot dışından değeri değiştirilebildiği için C de yanlıştır. Programın her çalışmasında istendiği kadar değeri değiştirilebilir. D seçeneği de yanlıştır.
Kaynak: <https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>
33. Cevap A. İlkel veri tipleri küçük harflerle başlarlar. null değeri atanamaz ve kendi ilkel veri türümüzü oluşturamayız. String sınıfı ilkel bir veri tipi değildir. Bu nedenle B, C ve D seçenekleri yanlıştır.
Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
34. Cevap D. garbage collector, çöp toplama işlemi için zorlanamaz. Bu nedenle A, B ve C seçeneklerinde gibi yöntemlerle çöp toplama işlemi başlamaz. Doğru cevap D seçeneğidir.
35. Cevap C. main() metodu sonunda 3 değişken de "apple" değeri ile değer atanmış olacaktır. Bu nedenle 2 tane String nesnesi çöp toplama(garbage collection) için uygun olacaktır.
Kaynak: <https://www.javatpoint.com/Garbage-Collection>
<https://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html>
36. Cevap B. Double sınıfının kurucu metodu, double ilkel veri tipi ismiyle çağrılmaz. Bu nedenle A ve C seçenekleri ve dolaylı olarak D seçeneği yanlış olacaktır. Sadece B seçeneği ile kod derlenir hale gelir.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/Double.html>
37. Cevap B. main() metodu içinde new anahtarı sözcüğü ile InitOrder sınıfı kurucu metoduna çağrı yapılmaktadır. İlk planda first değişkeni, "instance" değeri ile set edilse de, kurucu metot çalıştığı anda "constructor değeri yüklenecektir."
Kaynak: <https://docs.oracle.com/javase/tutorial/java/javaOO/objectcreation.html>

38. Cevap C. İlkel veri tipleri null değeri ile yüklenemezler. Bu nedenle ilk ifade derlenmez, diğer iki ifade derlenir.
Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
39. Cevap C. Bir instance değişkenine yalnızca o sınıftan üretilen instance'ın metotlarından başvurulabilir. Static bir değişkene herhangi bir yöntemden başvurulabilir.
Kaynak: <https://www.geeksforgeeks.org/static-methods-vs-instance-methods-java/>
40. Cevap B. Nümerik veri tipinde alt çizgi sembolojisi iki sayısal karakter arasında olabilir. B seçeneğindeki ifade derlenmez.
Kaynak: <https://docs.oracle.com/javase/8/docs/technotes/guides/language/underscores-literals.html>
41. Cevap A. Seçeneklerde verilen ilkel veri tiplerinin boyutları byte için 1 byte, short için 2 byte, int için 4 byte ve long için 8 byte'tır. Böylece doğru sıralama A seçeneğindeki gibidir.
Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
https://www.w3schools.com/java/java_data_types.asp
42. Cevap A. Bir sınıftan üretilen nesnenin bir özelliğini set etmek için nesne_adi.ozellik_adi şeklinde bir notasyon izlenir. Doğru cevap A seçeneğindeki gibidir.
Kaynak: https://www.tutorialspoint.com/java/java_variable_types.htm
43. Cevap B. Burada tanımlanan finalizer() metodu garbage collector'a ait metot değildir. Toy sınıftan üretilen nesneden çağrılmadığı sürece hiç çalışmaz ve "clean-" ifadesi terminale basılmaz. Toy sınıftan üretilen her nesnenin play metoduna çağrısında "play-" ifadesi basılır. Sonuç B seçeneğindeki gibi olacaktır. System.gc() metodu kullanılmayan nesneleri JVM'den temizleyecektir.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/System.html#gc-->
44. Cevap A. Verilen örnek setBeakLength() metoduna parametre olarak gönderilen Penguin sınıfı türünden p değişkeninin beakLength özelliği yüklenmektedir. Doğru yöntem, A seçeneğindeki gibidir.
45. Cevap B. parseInt() metodu bir ilkel veri dönerken, valueOf() metodu bir obje döner. Bu nedenle ilk boşluğa ilkel veri tipi, ikincisine Kapsayıcı sınıfı yazılacaktır.
Kaynak: <https://docs.oracle.com/javase/tutorial/java/data/autoboxing.html>
<https://www.geeksforgeeks.org/autoboxing-unboxing-java/>
46. Cevap B. 9. satıra gelindiğinde 3 nesnenin de referansı vardır. Ancak bir sonraki adımda diana nesnesi yerine zoe nesnesi gelir ve diana garbage collection için uygun hale gelir. Bu nedenle, main() metodu sonunda bir tane nesne bu işlem için uygun olacaktır.
Kaynak: <https://www.geeksforgeeks.org/garbage-collection-java/>
47. Cevap C. Java'da kurucu metotların dönüş tipi olmaz. Parametrelili ve parametresiz kurucu metot olarak iki türde kurucu metot vardır. Metodun adı sınıf adı ile aynıdır. Bu bilgilere uyan tanımlama C seçeneğindeki ifadedir.
Kaynak: <https://www.geeksforgeeks.org/constructors-in-java/>
<https://www.javatpoint.com/java-constructor>
48. Cevap A. Verilen kod parçasında car ve doll isimli nesne referansları null yapılırsa çöp toplama için uygun hale geleceklerdir. Kodda ne olursa play() metodu 2 defa çalışacağından dolayı "play-play-" şeklinde çıktı verir. Eğer bu nesne referansları null yapılırsa garbage collector finalize() metodu ile her çağrı için "clean-" ifadesini çıktı olarak verebilir. Bu neden B, C ve D muhtemel ekran çıktılarıdır. A seçeneği değildir.
49. Cevap B. Her kapsayıcı sınıfın, ilkel metodunun eşdeğerini aldığı bir kurucu metodu vardır. Doğru cevap B seçeneğidir. Diğer seçenekler ilkel bir veri tipinin kapsayıcısına autoboxing yapılmadan dönüştürülmesinde söz konusu olmayan durumlar olduğu için A, C ve D seçenekleri yanlıştır.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/data/autoboxing.html>
<https://www.geeksforgeeks.org/autoboxing-unboxing-java/>

50. Cevap C. main() metodunda new anahtar sözcüğü ile Sand sınıfı kurucu metoduna çağrı yapılır ve önce "a" değerini basar. main() metodu içindeki Sand() kurucu metodu çağrısı ardından Sand sınıfının run() metoduna çağrı yapar. run() metodu içinde yeniden new anahtar sözcüğü ile kurucu metoda çağrı yapılır ve "a" değeri basılır. run() metodu içinde sonraki adımda ise kurucu metot olmayan Sand() metodu çağrılır ve "b" değeri basılır. Sonuçta terminalde "aab" değeri görülecektir.