

1. Cevap C. String nesneleri immutable(değiştirilemez) iken StringBuilder nesneleri mutable(değiştirilebilir)'dir. StringBuilder sınıfı multi-thread desteği sağlamaz. StringBuilder için == operatörü değerleri değil, referansların eşitliğini kontrol ettiği için B seçeneği yanlıştır. String tipinde bir değişken yaratılırken her defasında yeni bir nesne üretilmiş olur. StringBuilder bir referans yarattığı için yaratılan nesne sayısı azaltılmış olur. Bu nedenle C seçeneği doğrudur.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html>
<https://www.geeksforgeeks.org/string-vs-stringbuilder-vs-stringbuffer-in-java/>
<https://www.geeksforgeeks.org/stringbuilder-class-in-java-with-examples/>
<https://www.geeksforgeeks.org/difference-equals-method-java/>
2. Cevap D. String türünde bir değişken yaratılırken String sınıfı kurucu metodu çağırmak yerine literal kullanarak doğrudan atayabilir. A seçeneğindeki bilgi doğrudur ve A doğru seçenek değildir. String, string havuzunda yeniden kullanılabilir olacağı için B seçeneğindeki bilgi doğrudur, ancak cevap değildir. String sınıfı final'dır ve immutable'dır. Bu sebeple C seçeneği yanlıştır ve D seçeneği doğrudur.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>
Javadoc of java.lang.String
3. Cevap D. Verilen değerlerin hepsi aynıdır. Soruda StringBuilder sınıfındaki metotlar çalışması hakkında örnekleme vardır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html>
4. Cevap B. Verilen kod örneğinde teams isimli StringBuilder referansı değerlerin append edilmesi sonrası "333 806 1601" değeri ekran basılır.
5. Cevap B. Verilenlerden sadece ArrayList sınıfı boş bırakılan yere yazılarak kod derlenebilir. List bir interface'dir ve new anahtar sözcüğü örneklenemez. Object sınıfı, concrete(somut) bir sınıftır. Object sınıfı, List interface'ine cast edilerek kod derlenebilir ancak bu durumda da uygulama çalışma zamanında ClassCastException fırlatır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/List.html>
6. Cevap C. Verilen kod derlenip çalıştırıldığında "hammer", "nail" ve "hex key" değerleri sırasıyla 0, 1 ve 2. İndekslere yerleştirilir. tools.get(i) ile 1. indekste yer alan "nail" değeri ekran çıktısı olur.
7. Cevap C. Verilen kodda sb isimli StringBuilder referansı henüz initialize edilmeden kod içinde kullanılmıştır. Buradan dolayı kod derlenmez.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html>
8. Cevap A. Verilen kod örneğinde museums isimli listeye sırayla "Natural History", "Science" ve "Art" değerleri eklenmektedir. remove(2) metodu ile "Art" değeri listeden çıkarılır. Liste ekrana basıldığında "[Natural History,Science]" değeri ekrana basılır. Olmayan bir indeksten değer silmeye çalışılırsa IndexOutOfBoundsException fırlatır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/List.html#remove-int->
9. Cevap C. Verilen kod örneğinde b isimli StringBuilder nesne referansı ilklendirilirken "12" değeri yüklenmekte ve hemen sonrasında append() metodu ile "3" değeri eklenmektedir. Sonrasında reverse() metodu b isimli referansın içindeki değerleri ters indeksten sıralar ve ekrana "321" değerleri basılır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html>
10. Cevap D. A seçeneğinde autoboxingden bahsedilmektedir. Bu hususun lambda ifadeleri ile ilgisi yoktur. A seçeneği yanlıştır. B ve C seçeneklerindeki durumları Java'da gerçekleştirmek mümkün olmadığından dolayı bu seçeneklerde yanlıştır. Lambda ifadeleri yazılan kodun geç çalışmasına olanak sağladığından doğru cevap D seçeneğidir.
11. Cevap D. Verilen kodda line isimli referans initialize edilirken "-" değeri yüklenir. 6 numaralı satırda line referansına "--" değeri append() metodu eklenir ve "--" değeri anotherLine

referansına yüklenir. Sonuç olarak her iki StringBuilder referansı da aynı değerle yüklenir == operatörü ile işleme sokulduğunda true değeri döner. line ve anotherLine referanslarının boyutları 2 olacaktır. Ekran çıktısı "true 2" şeklinde olacaktır.

12. Cevap D. Verilen referans tiplerden hiçbirisi kodun derlenmesini sağlamaz. ArrayList tipi için add() metodu vardır, ancak get() metodu Object türünden döndüğü String türüne casting yapmak gerekecektir. Ayrıca ArrayList referans türü boyutu size() metodu ile alınır. Bu nedenle ArrayList türü uygun olmayacaktır.

ArrayList<StringBuilder> türü için add() metodu yoktur. get() metodu Object döner ve ArrayList StringBuilder türünden referanslarla doldurulacağı için str referansı StringBuilder türünde yapılabilir. Benzer şekilde boyut size() metodu ile alınır. Bu seçenek de uygun olmayacaktır.

List referans türü için add() metodu vardır. Ancak get() metodu Object ve String'e casting gerekir. Ayrıca size() metodu ile boyut alınır. Bu nedenle seçeneklerden hiçbirisi değildir.

Kaynak: https://www.tutorialspoint.com/java/util/arraylist_size.htm
<https://docs.oracle.com/javase/8/docs/api/java/util/List.html>

13. Cevap D. Predicate interface'i boolean'a map etmektedir. Bu nedenle StringBuilder çıkarılarak sıfır parametrelili lambda haline getirilemez. A seçeneği yanlıştır. B seçeneğinde lambda ifadesinin yapısını bozan bir durum vardır ve B seçeneği de yanlıştır. Lambda ifadesinde -> operatörü sonrası süslü parantez "{" beklenmektedir. Bu nedenle C seçeneği de kodu derlenmez hale getirir. D seçeneğinde return ifadesi kaldırılrsa da yapı bozulmaz. D seçeneği doğrudur.

Kaynak: <https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>
<https://docs.oracle.com/javase/8/docs/api/java/util/function/Predicate.html>
javadoc: boolean java.util.function.Predicate.test(StringBuilder t)

14. Cevap A. Verilen kod örneğinde chars isimli listeye sırayla "a" ve "b" karakterleri eklenmekte, 0. indekse "c" değeri set edilmektedir. Böyle listede "[a,c]" değerleri vardır. Sonrasında 0. indekste yer alan "a" değeri silinir. Böylece liste boyutu 1'e düşer. contains() metodu ile "b" karakteri aranır ve bu değer listede olmadığından false dönerek ekrana "1 false" ifadesi yazılır.

Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/List.html>
https://www.tutorialspoint.com/java/util/arraylist_contains.htm#:~:text=The%20jav a,list%20contains%20the%20specified%20element.

15. Cevap D. String tipi için reverse() metodu yoktur. Bu nedenle kod derlenmez. Kod StringBuilder referans tipi ile düzenlenerek derlenip çalıştırılabilir.

Kaynak: <https://www.baeldung.com/java-reverse-string>
https://www.tutorialspoint.com/java/lang/stringbuilder_reverse.htm

16. Cevap A. Verilenlerden pred1, pred2 ve pred4 geçerli gösterimlerdir. ise isimli lambda ifadesi ise geçersizdir. "String s" ifadesinin parantezler içinde yazılması gerekmektedir.

Kaynak: <http://eherrera.net/ocpj8-notes/04-lambda-built-in-functional-interfaces>

17. Cevap A. Verilen kod örneği lambda ifadesinin uygulanmasını göstermektedir. double bir argüman olarak boolean dönen bir metot vardır. Bu metoda 45 ve 5 değerleri gönderildiğinde 45 > 5 sonucu true olacağından true değerini ready değişkenine yükler ve ekrana true basar.

18. Cevap A. Verilen kod örneğinde String sınıfının concat() metodu ile teams referansına değer birleştirilmektedir. concat() metodu kendisine parametre olarak verilen değeri örnek referansın sonuna ekliyor görünse de bu değer teams'e atanmadığı için korunmamaktadır. Bu nedenle teams referansında "694" değeri değişmemektedir ve ekran çıktısı bu olacaktır.

Kaynak: https://www.tutorialspoint.com/java/lang/string_concat.htm
<https://www.geeksforgeeks.org/java-string-concat-examples/>

19. Cevap A. Verilenlerden sadece ArrayList sınıfı java.util paketinde yer almaktadır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/package-summary.html>
20. Cevap C. A, B ve D seçenekleri "radical robots" şeklinde çıktı üretmektedir. C seçeneğinde "radicalrobots" şeklinde çıktı üretilir. Çünkü, C seçeneği için sb referansı boyutu 8'dir ve 7. indeksten "robots" değeri insert edildiği için ekran çıktısı "radicalrobots " şeklinde yani "radical" ifadesi ve " " (boşluk) arasına insert edildiği için ekran çıktısı diğerlerinden farklı olmaktadır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html#method.summary>
21. Cevap A. Verilen kod örneğinde "Natural History" ve "Science" değerleri array isimli diziye doldurulmaktadır. Sonrasında array dizisi museums listesine liste olarak set edilmekte ve sonrasında 0. indekste yer alan "Natural History" değeri yerine "Art" yazılmaktadır. List sınıfının contains() metodu "Art" değerini listede aradığında bularak true değerini döner ve ekrana basılır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/List.html#contains-java.lang.Object->
22. Cevap D. contains() metodu kendisine parametre olarak verilen değeri referansın için arar ve varsa true yoksa false döner. equals() metodu değer eşitliğini kontrol eder. startsWith() metodu ise kendisine parametre olarak değer, test edilen referansın değer başlangıç kısmıyla kontrol girer. Sonuca bağlı olarak bu değer ile başlıyorsa true, başlamıyorsa false döner.
Bu bilgiler göz önüne alındığında "abc" değerinin aranan metinde yer alması durumu "abc" ye eşit olduğu anlamına gelmez. Ayrıca "abc" ile başlayacağı da garanti değildir. Benzer şekilde "abc" ile başlaması "abc" ye eşit anlamına gelmez. Bunlara göre A, B ve C seçenekleri yanlıştır. Ancak "abc" ile başlaması, "abc" değerini ihtiva etmeyi garanti edeceğinden dolayı D seçeneği yanlıştır.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>
<https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html>
<https://www.geeksforgeeks.org/difference-equals-method-java/>
23. Cevap D. Verilen kod örneğinde chars isimli listenin boyutu 25 numaralı satırda length() metodu ile alınmaya çalışılmaktadır. List sınıfından bir nesnenin boyutu size() metodu ile edilir. 25 numaralı satırdan dolayı kod derlenmez. Cevap seçeneklerden hiçbirisidir.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/List.html#size-->
24. Cevap B. Verilenlerden String sınıfı boşluğa yazılabilir. replace() ve startsWith() metotları String sınıfında implemente edilmiştir. ArrayList ve StringBuilder tipleri, kodu tamamlamak için kullanılamaz.
Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#method.summary>
25. Cevap B. Verilenlerden P pozisyonuna <DataType> şeklinde tip belirtilirken, Q pozisyonuna <> ifadesi herhangi bir ekleme olmadan yazılabilir. Bu nedenle seçeneklerden B şıkkı doğrudur.
26. Cevap D. Verilen lambda ifadesinde Predicate ile eşleşme olması için "(String s)" ifadesi yazılmalıdır. Ancak main() metodu komutu satırı argümanı ile lambda argümanı aynı ismi taşıdığı için C seçeneği de doğru olmayacaktır. Bu nedenle cevap hiçbirisidir.
27. Cevap A. Kod örneğinde verilen line nesne referansında "--" değeri varken, anotherLine nesne referansına "--" değeri line referansına "--" değeri concat edilmek suretiyle eklenmektedir. Bu nedenle == operatörüne girdiklerinde false döner. line isimli referansın boyutu 1'dir ve ekran çıktısı "false 1" şeklinde olacaktır.
28. Cevap C. startsWith() metodu java.lang.String sınıfının bir metodudur. Bu nedenle c değişkeni String'e cast edilmelidir. Casting yapılmadığı için kod derlenmez.
Kaynak: <https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>

<http://tutorials.jenkov.com/java/lambda-expressions.html>

29. Cevap B. Seçeneklerde verilenlerden LocalTimeStamp adında bir sınıf Java'da yer almaz. Diğer sınıflar java.time paketi içinde yer almaktadır. Bu sınıflardan LocalDate, sadece date bilgisine sahiptir. LocalDateTime ve LocalTime sınıfları saat, dakika ve saniye bilgisini içerir.

Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html>

<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDateTime.html>

<https://docs.oracle.com/javase/8/docs/api/java/time/LocalTime.html>

30. Cevap D. Verilen kod örneği sorunsuz olarak derlenir. Ancak çalışma zamanında StringIndexOutOfBoundsException fırlatır. String tipinde olan builder isimli değişkene "54321" değerleri yüklenmiştir. Sonrasında 4. indeksten itibaren olan değerler, yani sadece "1" değeri builder değişkenine yüklenmektedir. Son aşamada charAt(2) metodu ile elde kalan builder değişkeninin 2. indeksinde yer alan değer okunmak istenmektedir. Ancak böyle bir indeks olmadığı için StringIndexOutOfBoundsException fırlatılır.

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

31. Cevap D. Verilen kod parçasında "i->i!=0" ifadesi yerine geçebilecek ifade seçeneklerde süslü parantezlerle gösterilmiştir. Süslü parantezlerle ifade edersek return kelimesi ve ifade sonunda ";;" karakterine ihtiyaç olacaktır. Bu nedenle yerine geçebilecek ifade D seçeneğindeki gibidir.

32. Cevap B. Verilen kod parçasında LocalDate türünden xmas referansına 25.12.2016 tarihi yüklenmektedir. Sonraki satırda LocalDate.plusDays() metodu ile gün sayısı 1 azaltılmakta ancak referansa set edilmemektedir. Bu nedenle xmas referansında yüklü olan dayOfMonth değeri 25 olarak kalmıştır. Bu nedenle ekran çıktısı 25 olacaktır.

Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html#method.summary>

33. Cevap A. Verilen kod parçasında sb isimli StringBuilder türünden referansa "red" değeri append edilmektedir. Sonrasında 0. indekste yer alan değer deleteCharAt() metodu ile silinmekte ve "ed" değeri kalmaktadır. Sonrasında ise delete(1,2) metodu ile belirtilen indeksten başlanarak bitiş indeksine kadar olan değer silinmektedir. Sonunda elde "e" değeri kalır ve ekrana basılır.

Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html#deleteCharAt-int->

34. Cevap B. Verilen lambda ifadesinde clear isimli referansta "clear" değeri eşitlik kontrolü konulmaktadır. Sonrasında test() metodu ile "pink" ve "clear" değerleri eşitliği kontrol edildiğinde eşitsizlikten dolayı false döner.

Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/function/Predicate.html#test-T->

35. Cevap C. Java'da diziler ve String'ler için indeksleme 0'dan başlamaktadır. LocalTime sınıfında date yada timezone bilgisi içermez. Bu nedenle indeksleme 0'dan başlayabilir. Fakat LocalDateTime sınıfında date bilgisi olduğundan dolayı indeksleme 0 yerine 1'den başlar.

Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/time/LocalTime.html>

<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDateTime.html>

36. Cevap C. java.util.function.Predicate bir FunctionalInterface'dir. Predicate'in boolean dönen soyut bir metodu vardır. Bu metodun ismi test() metodudur. Bu bilgilere göre A, B ve D seçenekleri doğru, C seçeneği yanlıştır.

Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/function/Predicate.html>

37. Cevap B. Period sınıfından 2 referans alınmaktadır. period1 isimli referans 3 gün zaman genişliği oluştururken period2 isimli referans 10 günlük zaman genişliği oluşturur. Bu nedenle period2 daha geniş bir zaman gösterimi yapar.

Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/time/Period.html>

38. Cevap B. Verilen kod örneğinde 01.01.2017 tarihi newYears LocalDate sınıfından bir referansa atanmaktadır. Sonrasında 1 günlük bir periyot belirlenmekte ve newYears tarihinden period isimli zaman minus() metodu ile çıkarılmaktadır. Sonuç olarak 12-13-2016 tarihi elde edilerek ekran çıktısı olmaktadır. Zaman gösterimi de DateTimeFormatter sınıfından format isimli referans ile yapılmaktadır.
- Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/time/Period.html>
<https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html>
<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html>
39. Cevap C. Verilen kod örneğinde happy isimli String değişkene “ :) – (: ” değeri yüklenmiştir. Hemen sonra trim() metodu ile happy değişkeninin başında ve sonunda yer alan boşluklar silinmektedir. Daha sonra question isimli değişkene happy değişkeninin ilk karakteri atlanarak 1. indeksten length() -1. indekse kadar olan değer alınır. Böylece question ve really’nin değerleri eşit olabilsin. Bu iki değişkeninin equals() metoduna sokularak true dönmesi için başlardaki ve sonlardaki boşluklar temizlenmektedir. Bu nedenle uygun yordam C seçeneğidir.
- Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>
javadoc java.lang.String.trim()
40. Cevap C. Java.time.Period sınıfı immutable’dır. Period sınıfı, günleri, ayları ve yılları gösterebilir. Tarih şeklindeki zamanlardan gün, ay, yıl gibi zamanları ekle-çıkartabilir. Ancak dakika, saniye, saat şeklinde gösterim yapamaz. Bu nedenle C seçeneğindeki bilgi doğru değildir.
- Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/time/Period.html>
41. Cevap D. StringBuilder türünden oluşturulan builder isimli referansa “54321” değeri yüklenmiştir. Sonraki satırda substring() metodu ile 2. İndeksten sonraki değerleri alınmış, ancak referansa set edilmediği için builder isimli referans değerini korumaktadır. Dolayısıyla charAt() metodu ile hangi indeksteki değer sorgulanırsa builder referansının ilk halindeki indeks değeri eşleşen değer ekrana basılır. Ekran çıktısı “4” olacaktır.
42. Cevap B. Verilen örnekte pennies isimli listeye sırayla 3, 2 ve 1 değerleri 0, 1 ve 2. indekslere sırayla yüklenir. Sonrasında remove() metodu ile 2. indekste yer alan 1 değeri listeden kaldırılır. Böylece listede 3 ve 2 değeri kalır. Burada listede olmayan bir indekse değer ekleme veya olmayan bir indeksten değer silmeye kalkılırsa java.lang.IndexOutOfBoundsException hatası fırlatır.
- Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/List.html>
43. Cevap C. Verilen kod örneğinin en doğru şekilde tamamlanması için boş bırakılan referans tür yerine seçeneklerden StringBuilder referans türü uygundur. ArrayList sınıfında charAt(), insert() ve length() metotları yer almaz. String türünde ise insert() metodu yoktur. StringBuilder kodda yer alan 3 metodu da implemente etmektedir. Bu nedenle cevap C seçeneğidir.
- Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>
<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>
<https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html>
44. Cevap C. LocalTime sınıfına eklenebilecek en küçük zaman birimi nano-saniyedir. Piko-saniye daha küçüktür ancak LocalTime sınıfında implemente edilmemiştir. Milisaniye ve saniye ise daha büyük birimlerdir. Bu nedenle doğru cevap C seçeneğidir.
- Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/time/LocalTime.html>
45. Cevap D. Verilen kod derlenir. Fakat çalışma zamanında java.time.temporal.UnsupportedTemporalTypeException hatası fırlatır. Bu hatanın nedeni DateTimeFormatter.ofPattern() metoduna parametre verirken ay gösteriminin “mm” olarak verilmesinden kaynaklanmaktadır. Doğru gösterim “MM” ile yapılmalıdır.
- Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html>

46. Cevap D. `java.lang.String.replace()` metodunun iki metot imzası vardır. Bunlardan birisi iki tane `char` parametre alırken diğeri `CharSequence` alır. Hem `String` hem `StringBuilder` bu interface'i implemente etmektedir. Bu nedenle verilen seçeneklerden hepsi `String` sınıfındaki `replace()` metoduna parametre olarak geçilebilir.

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

47. Cevap C. Verilen kodda `print()` metodundaki `if` kontrolü içinde `Predicate FunctionalInterface`'inin `test` metoduna çağrı yapılmaktadır. `print()` metoduna parametre olarak `Predicate` interface'i `Integer` türden değer aldığı için `String` ile `Integer` eşitliğine bakılacağından dolayı kod bu satırda derlenmez. Burada `Predicate<String>` şeklinde metot parametresini düzelterek, `print()` metoduna çağrı yapan satırdaki lambda ifadesinde de metoda geçilen parametreyi "`e -> Integer.parseInt(e) < 0`" şeklinde düzenlersek kod derlenir ve çalışır hale gelir. Sonuç olarak da -5 değerini ekrana basar. Yani 1 satır çıktısı olurdu. Ancak mevcut haliyle kod derlenmez.

48. Cevap D. Verilen kod parçası derlenir ancak çalışma zamanında `java.lang.IndexOutOfBoundsException` hatasını fırlatır. Bunun nedeni `magazines` isimli listeye belirli eklemeler yapıldıktan sonra `clear()` metodu ile listenin temizlenmesi sonrası listeye sadece 1 değer eklendikten sonra 1. indeksteki değeri silmeye çalışma sonucu olmaktadır. `clear()` çağrısında liste temizlendiği için sonraki ilk eklenen değer 0. indekste yer alır ve 1. indeks henüz yoktur. Olmayan indeks için `remove()` çağrısı yapılamaz.

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

49. Cevap C. Verilen kodda `witch` değişkenine atanmak istenen değer `String` türünden bir değer değil `char` türünden bir değerdir. Gösterim şekli "`b`" olarak düzeltilirse ekran çıktısı "`black`" şeklinde olacaktır.

50. Cevap C. Verilen kod derlenmez. `LocalDate` sınıfı `immutable` olduğundan dolayı `setter` metotları bulunmaz. Kodda verilen `setYear()` şeklinde bir metoda sahip olmadığı için kod bu satırdan dolayı derlenmez.

Kaynak: <https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html>