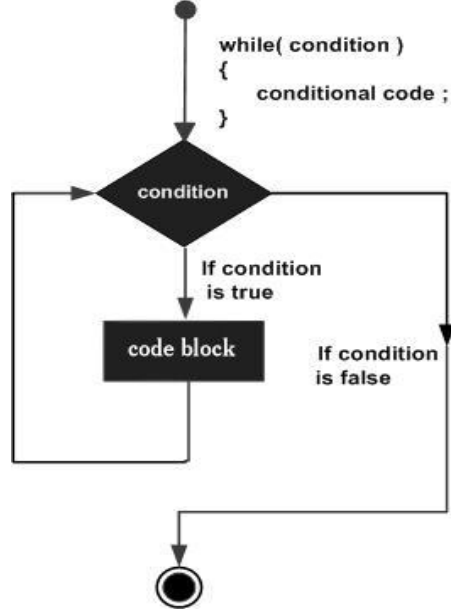


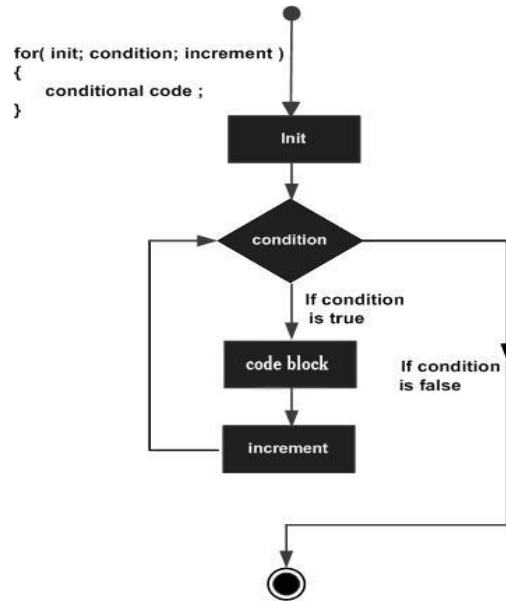
1. Cevap D. boolean bir ifadenin doğruluğuna göre döngüye giren döngü while döngüsüdür.



Şekil-1: while Döngüsü Diyagramı

Kaynak: https://www.tutorialspoint.com/java/java_while_loop.htm

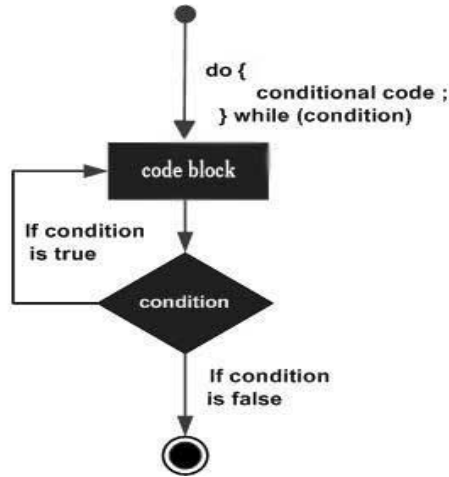
2. Cevap B. Bir indeks ve time counter kullanarak inşa edilen en bilinen döngü geleneksel for döngüsüdür.



Şekil-2: for Döngüsü Diyagramı

Kaynak: https://www.tutorialspoint.com/java/java_for_loop.htm

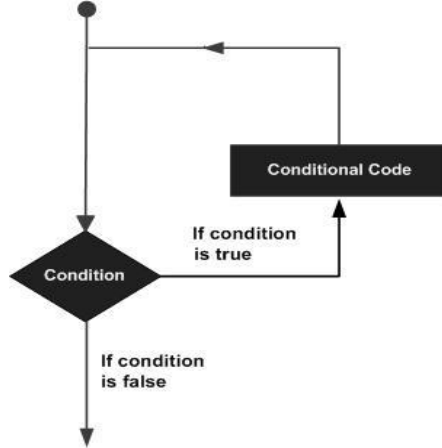
3. Cevap A. do-while döngüsünde döngü içine en az bir kez girilir ve while kısmında yer alan kontrol true olduğu sürece döngü çalışır. do-while döngüsünde gövdede yer alan kodların en az bir defa çalışması garanti edilir.



Şekil-3: do-while Döngüsü Diyagramı

Kaynak: https://www.tutorialspoint.com/java/java_do_while_loop.htm

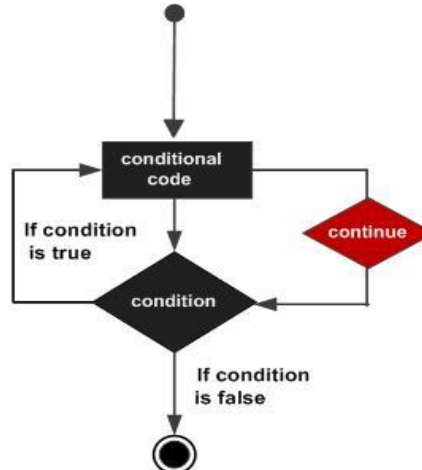
4. Cevap C. foreach döngüsünde for döngüsünden farklı olarak kendisine verilen bir indekse ya da referansa ihtiyaç olmaksızın tüm elemanlar içinde gezilerek gövdede yer alan kodlar işletilir. Bu işlemler sırasında değer bir değişkende tutulur.



Şekil-4: for-each Döngüsü Diyagramı

Kaynak: <https://docs.oracle.com/javase/8/docs/technotes/guides/language/foreach.html>
<https://www.tutorialspoint.com/foreach-in-Java>

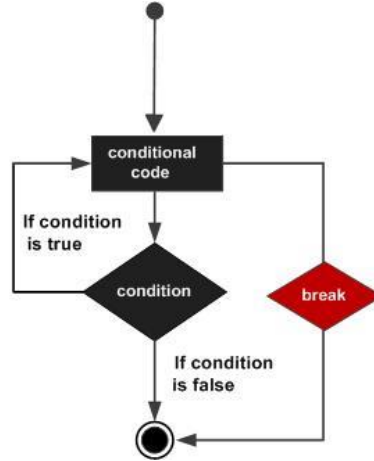
5. Cevap B. Java'da continue deyimi döngüde belirli bir koşul kontrolüyle sonraki iterasyona geçmeyi sağlayan anahtar sözcüktür.



Şekil-5: continue Anahtar Sözcüğü Akış Diyagramı

Kaynak: https://www.tutorialspoint.com/java/java_continue_statement.htm

6. Cevap A. Java’da döngüyü hızlıca kırarak çıkmak için kullanılan anahtar sözcük break deyimidir.



Şekil-6: break Deyimi Akış Diyagramı

Kaynak: https://www.tutorialspoint.com/java/java_break_statement.htm

7. Cevap B. Java’da geleneksel for döngüsü genel olarak 3 ifadeden oluşur. Bunlar; ön yükleme ifadesi, kontrol ifadesi ve takipçiyi güncelleme ifadesidir.

Kaynak: <https://www.javatpoint.com/java-for-loop>
<https://www.geeksforgeeks.org/loops-in-java/>

8. Cevap C. Geleneksel for döngüsünde iterasyona başlanılan dizinin ilk yani 0. indeksinden ya da son indeksinden başlanabilir. Bu nedenle iki ifade de teknik olarak doğrudur.

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

9. Cevap A. foreach döngüsünde iterasyon ilk indeksten başlanarak icra edilir.

Kaynak: <https://docs.oracle.com/javase/8/docs/technotes/guides/language/foreach.html>
<https://www.baeldung.com/java-iterate-list>

10. Cevap A. Java’da bir iterasyon işletildikten sonra bir koşula bağlı olarak döngüye devam edildiği yapı do-while döngüsünde bulunmaktadır. Geleneksel for döngüsü ve foreach döngüleri sadece boolean bir şarta olarak inşa edilmediği için B ve C seçenekleri yanlıştır. D seçeneğindeki while döngüsü ise hiçbir iterasyon icra edilmeden koşul kontrolüne girdiği için soruda verilen şartı sağlamaz ve bu yüzden D seçeneği de doğru değildir.

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

11. Cevap B. while döngüsünde koşul ifadesi kısmında boolean bir ifade yer almalıdır. Buraya int türünden bir ifade verildiği için kod derlenmez.

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

12. Cevap B. for döngüsü listenin son elemanından geriye doğru iterasyona koyulacak şekilde yapılandırılmıştır. Bu nedenle ekran çıktısı olarak önce “cup,” sonra “can,” basılarak sonuçta “cup,can,” şeklinde çıktı üretecektir.

Kaynak: <https://www.geeksforgeeks.org/loops-in-java/>

13. Cevap A. Listedeki ilk eleman “glass” ifadesine “,” karakteri eklendikten sonra döngüde yer alan break komutu ile listenin diğer elemanları iterasyona sokulmadan döngüden çıkılır ve “end” ifadesi yazılır. Bu nedenle ekran çıktısı “glass,end” şeklinde olacaktır.

Kaynak: https://www.tutorialspoint.com/java/java_break_statement.htm

14. Cevap A. İçi boş olan letters değişkenin boyutu 0’dır. while döngüsünde değişkenin boyutu 2 olana kadar işlem yapılır. Her aşamada letters değişkenine “a” karakteri eklenir. 2. karakter eklendikten sonra değişken boyutu 2 olur ve döngüden çıkılarak değişkende yer alan “aa” değeri konsol çıktısı olarak basılır.

15. Cevap D. Döngüde takipçi değişken args isimli komut satırı argümanları dizisinin boyutundan başlayarak 0'dan büyük olduğu sürece artacak şekilde implemente edildiği için bu döngü sonsuz döngüye dönüşmektedir. Bu nedenle sürekli olarak alt alta "args" yazacaktır.
16. Cevap B. count değişkeni bir instance değişkeni olduğundan dolayı default değeri 0'dır. main() metodunda yer alan while döngüsünde koşul kontrolünde stops dizisinin boyutu 4'tür ve $4 > 0$ olduğu için break komutu görene kadar ya da count değişkeni 4'ü geçene kadar döngü devam edecektir. İlk iterasyonda stops[0]'da "Washington" değeri vardır ve if kontrolündeki ifade false döner ve if gövdesindeki kodlar işletilmez. count değişkeni 1 değeri ile güncellenir. İkinci iterasyonda stops[1]'de "Monroe" değeri vardır ve if kontrolündeki ifade true dönerek break komutu işletilir. break komutu işletilmesi öncesinde count değişkeni 2 değeri güncellenir ve sonuçta ekran çıktısı 2 olacaktır.
17. Cevap C. Dıştaki do-while döngüsünde tanımlanan count değişkeni yerel bir değişkendir ve döngü bloğu dışından erişilemez. Buradaki söz dizimi hatasından dolayı kod derlenmez. Eğer count değişkeni tanımı dıştaki do-while döngüsü öncesinde tanımlanırsa ekran çıktısı 2 olur.
18. Cevap D. for döngüsünde yer alan üç segment de boş bırakılabilir. Bu teknik olarak sonsuz döngü anlamına gelecektir. Bu döngüden çıkabilmek için döngü de break deyimi kullanılmış olması gerekir.
- Kaynak: <https://stackoverflow.com/questions/3792183/for-loop-missing-initialization>
19. Cevap C. Java'da geleneksel for, while ve do-while döngüleri sonsuz döngü yaratılabilir. foreach döngüsü ya da gelişmiş for döngüsünde parametre olarak bir dizi ya da veri yapısı olarak bu veri seti üzerinde indeks sıralı olarak iterasyona gireceği için bu döngü tipinde sonsuz döngü yaratılamaz.
- Kaynak: <https://www.baeldung.com/infinite-loops-java>
20. Cevap A. Verilen kod örneğinde 2 elemanlı drinks listesinde yer alan String değerler sırayla ekrana basılmakta ve değer sonuna "," karakteri eklenmektedir. Böylece A seçeneğindeki gibi "can,cup," ibaresi ekran çıktısı olacaktır.
21. Cevap D. Verilen kodda do-while döngüsü implementasyonunda yazım hatası vardır. Bir do-while döngüsü tanımlanırken do ve while sözcükleri arasında süslü parantezlerle gövde kurgulanır. Buradaki parantezler süslü parantezle değiştirildiğinde döngü bir kez çalışarak "helium" değerini konsol satırına yazar ve while kontrolünde ifade false olduğu için çıkarak döngü sonlanır.
- Kaynak: Kaynak: https://www.tutorialspoint.com/java/java_do_while_loop.htm
22. Cevap B. A veya C seçeneklerindeki gibi bir foreach yapısı yoktur. Doğru söz dizimi B seçeneğindeki gibidir.
- Kaynak: <https://www.geeksforgeeks.org/for-each-loop-in-java/>
23. Cevap C. Verilen kod örneğindeki letters satırına dönebilmesi verilenlerden break komutu ve break numbers komutu kullanılabilir. break letters komutu verilirse her iki döngüden de çıkılmış olunur ve tek iterasyon yapılmış olur.
- Kaynak: <https://www.decodejava.com/java-labelled-break.htm>
24. Cevap B. Verilenlerden sadece continue letters komutu ile kod akışı tamamlanır. continue komutu bulunduğu döngüde bir sonraki iterasyona devam edeceği için çalışmayı etkilemez. continue numbers komutu ile dıştaki döngü için bir sonraki iterasyona geçiş demektir. Bu nedenle sadece continue letters ile 1 olacaktır.
- Kaynak: <https://www.journaldev.com/983/java-continue-statement>
25. Cevap C. Döngü hiçbir çıktı üretmeden sonlanır. Çünkü, singer değişkeninin değeri 0'dır ve while kontrolündeki ifadede $0 > 0$ işlem sonucu false dönerek döngüye girilmeden kod parçası geçer.

26. Cevap C. foreach döngüsünde dizilere veya Java Collections'ta yer alan ArrayList sınıfına izin verilir. Dolayısıyla StringBuilder türüne izin verilmez.
Kaynak: <https://www.geeksforgeeks.org/for-each-loop-in-java/>
27. Cevap B. balloonInflated değişkeni başta false iken do-while döngüsüne girildiğinde true yapılarak konsola "inflate-" ifadesi yazılmaktadır. Ardından while bloğunda yer alan koşul ifadesinde true ifadesinin değili alınarak döngüden çıkılır ve "done" ifadesi yazılır. Böylece ekran çıktısı "inflate-done" olur.
28. Cevap D. letters değişkeni başta olarak tanımlanmış ve while döngüsündeki koşulda ise bu değişkenin boyutu 3 olmadığı sürece iterasyon yapılacak şekilde döngü kurulmuştur. Değişkene her defasında 2 karakter eklendiği için değişken boyutu hiçbir zaman 3'e eşit olmayacağı için buradaki while döngüsü sonsuz döngüdür.
29. Cevap B. for döngüsünde ilk ifade ön yükleme (initialization) ifadesi, ikinci ifade boolean dönen bir koşul ifadesi ve üçüncü ifade ise geleneksel olarak kullanılan bir counter değişkeninin güncellendiği ifadedir. Doğru sıralama B seçeneğidir.
Kaynak: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html>
30. Cevap B. Başta count değişkeni değeri 10 ve chars isimli listede henüz eleman yoktur. do-while döngüsünde ilk iterasyonda chars listesine bir karakter eklenir ve chars listesinde tek eleman olduğu için for döngüsü bir iterasyon yapar, count değişkeni 1 azalarak 9 değeri ile yüklenir. do-while döngüsünün 2. iterasyonunda liste boyutu 1 tane daha artarak 2 olur ve bu defa for döngüsü 2 defa dönerek count değişkeni 2 azalarak 7 değeri ile yüklenir. Sonraki iterasyonda chars listesi 3 elemana ulaşır, for döngüsü 3 defa döner ve count değişkeni 3 azalarak 4 değeri ile yüklenir. Dördüncü iterasyonda chars listesinde 4 eleman olduğunda count değişkeni for döngüsünün 4 defa dönmesi sonucu 4 azalarak 0 değeri ile yüklenir ve artık while koşulunda 0>0 işlem sonucu false olacağı için do-while döngüsünden çıkılır. Son durumda chars listesinde 4 tane eleman olduğu için ekran çıktısı liste genişliği olan "4" olacaktır.
31. Cevap A. for döngüsünde i değişkeni 10 olarak yüklenmiştir. İçteki while döngüsünde ise i değişkeni değeri 3'ten küçük olana değin 3'er azalarak en son 1 değeri ile yüklenir. i=1 olduğu anda while döngüsünden çıkılarak k değişkeni değeri 1 artar. Daha sonra i değişkeni bu defa da for döngüsünde 1 azaltılır ve 0 değeri ile yüklendiği anda for döngüsünden de çıkılarak ekran çıktısı olarak k değişkeni değeri "1" ekran çıktısı olur.
32. Cevap D. Java'da gelişmiş for döngüsü ilk elemandan başlanarak iterasyonu gerçekleştirir. Verilen kod örneğindeki ifadede listenin son elemanından geriye doğru iterasyon yapıldığı için B seçeneğindeki kod parçası bu kodun eşiti değildir. A ve C seçeneklerinde ise söz dizimi hatasından dolayı geçersizdir.
Kaynak: <https://www.geeksforgeeks.org/for-each-loop-in-java/>
33. Cevap C. Verilen kodda for döngüsü süslü parantezlerle bloklanmadığı için break deyimi derleme hatasına yol açar. break deyimi gelişigüzel yazılarak main() metodunu kıramaz. Bu nedenle break deyimi blok içine alınırsa ekran çıktısı "glass,end" şeklinde olacaktır.
34. Cevap C. for döngüsünde counter değişkenlerinin güncelleme ifadeleri arasında ";" değil "," karakteri kullanılmadığı için kod derlenmez. Buradaki yazım hatası giderilirse ekran çıktısı nycTourLoops ve times dizilerinin sırayla ilk ve ikinci indekslerinde yer alan değerlerin birleştirilmesi sonucu "Downtown Day-Uptown Night-" şeklinde olacaktır.
35. Cevap D. Kod sorunsuz şekilde derlenir. Ancak çalışma zamanında for döngüsünde kullanılan i isimli counter değişkenin ilk değeri args dizisinin boyutu kadar olduğu için, ekran çıktısı basılan satırda args[3] değerine ulaşılmaya çalışılacaktır. Bu indeks değeri dizi dışında olduğu için uygulama ArrayIndexOutOfBoundsException fırlatır.
36. Cevap B. Kod örneğinde tie değişkeni ilk olarak null olarak tanımlanır. while döngüsü kontrolünde tie==null ifadesi true olduğu için döngüye girer ve tie="shoelace" olarak

güncellenir ve ikinci iterasyonda tie değişkeni artık null olmadığı için döngüden çıkılır. Ekran çıktısı “shoelace” olur.

37. Cevap C. Buradaki do-while döngüsünün iptal edilmesi durumunda break deyiminden dolayı artık derlenmez hale gelecektir. Java’da break komutu döngü veya switch yapısı dışında kullanılamaz.

Kaynak: <https://www.cis.upenn.edu/~matuszek/General/JavaSyntax/break.html>

38. Cevap C. count değişkeni instance değişkeni olarak tanımlanmıştır ve default değeri 0’dır. stops dizisinin boyutu ise 4’tür. main() metodu içindeki while döngüsü count’ın değeri 4 olana kadar döner ve 4 olduğunda döngüden çıkarak ekrana “4” değeri basılır.

39. Cevap C. do-while döngüsünde while bloğunda yer alan kontrolün boolean bir ifade olması gerekmektedir. Buraya parametre olarak verilen builder değişkeni StringBuilder sınıfının bir nesne referansı olduğu için geçersizdir.

40. Cevap A. Başlangıçta count=0’dır. Dıştaki do-while döngüsü sonsuz bir döngüdür ve bu döngü içinde break komutu görüldüğü anda döngü kırılacaktır. İçteki do-while döngüsünde ise count değişkeni 2 olana kadar artar ve 2 olduğu anda döngüden çıkar. break komutu ile dıştaki döngü kırılır ve ekran çıktısı olarak “2” değeri basılır.

41. Cevap C. Kodun sorun şekilde derlenir ve sonsuz döngüye yol açmayacak şekilde çalışması için koda eklenmesi gereken komut “break t;” dir. Çünkü, her iki döngüyü de bırakabilmesi için dıştaki while döngüsünü de kırması gerekmektedir.

42. Cevap B. Verilen kod sorunsuz derlenir ve çalışır. Verilen dizilerin önce ilk daha sonra ikinci elemanları birleştirilerek “Downtown Day-Uptown Night-” şeklinde ekran çıktısı üretecektir.

43. Cevap B. Kod sorunsuz olarak derlenir ve çalışır. exams listesi 2 elemanlıdır ve iç içe 2 döngü nedeniyle 4 satır olarak ekran çıktısı verir.

44. Cevap B. Döngü başladığında ilk olarak ön yükleme ifadesi olan alpha çalışır. alpha sonrası boolean ifade olan beta kontrolü yapılır. beta ifadesi true ise döngü gövdesindeki kodlar işletilir. Soruda beta’nın false olduğu verildiğine göre döngü girmez ve sadece alpha ve beta çalışmış olur.

45. Cevap B. Verilen döngüde delta ifadesi tam olarak bir defa çalıştığına göre işlem sırası alpha, beta, delta, gamma, beta şeklinde olacaktır. İlk olarak alpha çalışır, çünkü ön yükleme ifadesidir. Sonrasında beta ifadesi kontrol edilir. beta ifadesi true ise gövdeye girilerek delta işletilir. delta ifadesinde beta==false olacak şekilde revize yapılmış olsa gerek ki, delta bir kez çalışmaktadır. delta ifadesi işletildikten sonra gamma ifadesi ile counter değişken güncellemesi yapılır. Sonra tekrar beta ifadesi kontrol edilir. beta ifadesinin false olmasından dolayı uygulama sonlanır.

46. Cevap C. Verilenlerden C seçeneğindeki do-while döngüsü 6 iterasyon yaparken diğer ifadeler 5 iterasyon yapmaktadır. for döngüleri başlangıç indeksleri farklı olsa da birisi 0-4 indeksleri arasında diğeri 1-5 indeksleri arasında 5’er iterasyon yapar. D seçeneğindeki while döngüsü de k’nın 0, 1, 2, 3 ve 4 değerleri için 5 iterasyon yapar. Ancak do-while döngüsü 0-5 arasında toplam iterasyon yapar.

47. Cevap D. Verilen kod parçasındaki while döngüsü yanındaki “;” komutu ile sonsuz döngüye dönüşmektedir. Döngüye verilen tie değişkeni sürekli null kalacağı ve döngüden çıkılmayacağı için bu döngü sonsuz döngüdür. Eğer while döngüsü için break komutu koyarsak tie değişkenine “shoelace” değeri yüklenir ve bu değer ekrana basılır.

48. Cevap C. Verilen kod örneğinde for isimli bir etiket tanımlanmış ve kullanılmaktadır. Java’nın rezerve edilen anahtar sözcükleri değişken, metot, etiket tanımlarında kullanılamayacağı için kod derlenmez. Verilen koddaki “for” etiket ismi yerine “for_label” gibi bir ifade yazılırsa ekran çıktısı “5” olacaktır.

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

49. Cevap D. Verilen kod örneğinde `baloonInflated` değişkeni döngü içindeki if kontrolünde `true` yapıldıktan sonra `do-while` döngüsü sonsuz döngüye dönüşür ve döngüden artık çıkılmaz. Bu nedenle ekrana `"inflate-"` yazıldıktan sonra artık sonsuz döngüde devam edilir. Eğer bir `break` komutu ile çıkılırsa `"inflate-done"` yazılarak uygulama sonlanır.
50. Cevap B. Verilen kod örneğinde döngünün ilk yükleme ifadesi kısmına tanımlama yapılırken yazılabilecek ifade B seçeneğindeki gibidir. A, C ve D seçeneklerindeki ifadelerin hiçbiri söz dizimi kurallarına uymadıkları için derleme hatasına neden olur.