

HomeWork_4_Sonuç_Raporu

TURGUT YILDIZ

08.05.2020

CEVAPLAR

1. Dizi oluřturmanın javadaki bir dięer yntemi (...)'dır.
CEVAP : B
2. Q2 uygulamasında grldę gibi f[0] dizinin ilk elemanına eriřim adresini saęlar.
CEVAP : B
3. Diziler primitive deęildir. New ile belli bir obje oluřturulup daha sonra bu objede iřlemler yapılmaktadır.
CEVAP : D
4. Dizi tanımlamasında křeli parantez [] deęiřkenin sonuna veya tipinin nne konulur.
CEVAP : C
5. Q5 uygulamasına bakıldıęında varargs ifadesi ile oluřturulan dizi tanımlamaları doęrudan string deęer atamasına izin verirken [] ile oluřturulan diziler yalnızca new ile nesne oluřturulup deęer atamasına izin vermektedir.
CEVAP : C
6. Tekrar Q5 uygulamasına bakıldıęında bir dizinin uzunluęu yani eleman sayısı bulunması iin x.length; ifadesi ile bulunur.
CEVAP : A
7. 2 boyutlu diziler;
int[][] blue = new int[2][2]; // řeklinde her boyutun eleman sayısı belirtilecek řekilde oluřturulur.
CEVAP : C
8. Q8 uygulamasındaki gibi dizi.length ile 7 deęeri dner ve ekrana dizinin 7 elamanıda yazdırılır.
CEVAP : B
9. Dizide arama = Arrays.binarySearch() sıralama = Arrays.sort() Q9 uygulamasında bu metodlar denenmiřtir.
CEVAP : B
10. Q10 uygulamasına bakıldıęında String dizisinde sıralama alfabetiktir. Bundan dolayı sıralama 1 – 10 – 9 řeklindedir.
CEVAP : B

11. Dizi indislemesi 0. İndisten başlayarak dizi uzunluğunun 1 eksiği kadar ilerler.

CEVAP : B

12. Dizi oluşturulurken dizinin boyutunu elle giremeyiz. Dizi boyutu tanımlandığı değerlerden elde edilir. Ayrıca diziye boş değer ataması yapılabilmektedir.

CEVAP : C

13. 12. Sorudaki gibi dizi oluşturulurken ya değişken belirtilmeli ya da boyut verilmelidir.

CEVAP : B

14. Q9 uygulamasında denememi yaparken dizide bir arama yapmadan önce sıralanması gerektiğini öğrendim. Çünkü arama işlemi lineer bir sıralama yapılmış dizide sağlıklı sonuç vermektedir.

CEVAP : C

15. Dizinin ilk elemanına dizi[0] şeklinde ulaşılır.

Dizinin farklı elemanlarında aynı değerler olabilir.

Dizilerde sıralama yapılabilir.

Fakat diziler otomatik olarak genişlemez dizi oluşturulmadan önce belli bir genişlik atanır.

CEVAP : A

16. 1x2 boyutunda bir dizi oluşturulmuştur. Maksimum değer atama aralığı = dizi [0] [1] matrix[1][0] = "Is all around you "; // bu satırda dizi boyutu dışına çıkmıştır.

CEVAP : C

17. Q17 uygulamasına bakıldığında dizi sıralanmadan arama yapılmıştır. İlk eleman arandığı için 1 sonucu ile dönecektir.

CEVAP : B

18. Q18 uygulamasında iki boyutlu dizi tanımlamasında hata yapılan satır belirtilmiştir. [3 , 3] şeklinde olmayacak [3] [3] şeklinde olmalıdır.

CEVAP : A

19. Q19 uygulamasında belirttiğim gibi “new” kullanıldığından 3 tane nesne oluşur.

CEVAP : B

20. Değişken isminden önce ‘ [] ’ kullanımı yanlış gösterimdir.

CEVAP : B

21. Q21 uygulamasındaki gibi 2 boyutlu dizi oluşturulmuştur ve dizinin elemanları ekrana çıkarılmıştır. Burada A ve B seçenekleri gibi çıktı alınabilir. Fakat A şıkkında null değişkenlerde gösterilmiş gibidir.

CEVAP : B

22. Q22 uygulamasında olduğu gibi oluşturulan bir dizi tekrardan boyutlandırılmaz.
“ArrayIndexOutOfBoundsException” hatası verecektir.
CEVAP : D
23. Q23 uygulamasında size() metodu kullanılmıştır bu metod array listlerde liste boyunu belirlemek için kullanılır. Kod derlenmesi için lenght() metodu kullanılmalıdır.
CEVAP : C
24. Bool değişken isimlendirmelerinden önce yapılan 3 boyutlu dizi tanımlaması her iki değişken içinde geçerli olacaktır.
CEVAP : C
25. Q25 uygulamasında yapılan gerçeklemede ekrana çıkarma komutu oluşturulan dizinin kütüphane yolu ve adresini göstermektedir.
CEVAP : C
26. Q26 uygulamasında gerçekleştiği gibi 2 boyutlu sınırı 3 eleman olan dizinin [3][3] ile bir değer atanması olmayacaktır.
CEVAP : B
27. Varargs ifadesi metodlarda dizi parametresi tanımlaması için kullanılır. Bir değişken oluşturulamaz.
CEVAP : D
28. Q28 uygulamasında int bir diziye String bir değer atandığından ArrayStoreException hatası verecektir.
CEVAP : D
29. Q29 uygulamasında önce sıralama daha sonra arama yapılmıştır. Dizide olmayan bir eleman eğer sıralam yapılırsa en başa konacaksa dizi uzunluğu 3 olduğundan -2 olurdu 2. Sıraya gelecek olursa -3 3. Sıraya gelecekse -4 olacaktır.
CEVAP : C
30. Q30 uygulamasında görüldüğü gibi [ArrayIndexOutOfBoundsException](#) hatası alacaktır. Dizin boyutu girilmediğinden böyle bir hata alınır.
CEVAP : C B
31. Q31 uygulamasında dizi boyutu parametre bölümüne [] şeklinde girilmiştir. Dizi boyutu 0 dır.
CEVAP : A C
32. Q32 uygulamasında 0 uzunluklu bir dizinin ilk elemanı String değişkene atanıyor. Daha sonra 0 uzunluklu dizi sıralanmak isteniyor. Exception hatası verecektir.
CEVAP : D B

33. İlk 3 seçenekte [2] [1] dizi uzunluğu belirtilmiş fakat son seçenekte [2] [0] dolaylı yoldan 1 boyutlu bir dizi oluşturulur.

CEVAP : D

34. 3x3 boyutunda bir dizi Javada dizi[2][2] olarak belirtilir.

CEVAP : C

35. Q35 uygulamasında dizinin boyutu kadar değil 1 fazlası kadar dönlüdüğünden son döngüde hata verecektir.

CEVAP : D

36. 0 uzunluklu bir dizide dizi[1] ile 2. Elemana ulaşılmaya çalışılmıştır. ArrayIndexOutOfBoundsException hatası alınır.

CEVAP : C

37. Q37 uygulamasında 3x3 boyutunda bir dizinin boyutu satır uzunluğu yani ilk [3] içindeki değeri kadar sonu8ç üretir.

CEVAP : D

38. Dizi uzunluğu veren length() metodu yoktur. Değişken cinsinden değeri üretir.

CEVAP : D

39. boolean[][] bools[], moreBools; ifadesinde “bools” 2 boyuttan ayrıca 1 boyut daha kazanır ve 3 boyutlu bir dizi olur.” moreBools” ise 2 boyutta kalır.

CEVAP : B

40. Q40 uygulamasına bakıldığında boş uzunluksuz bir diziyi String’e çevirdiğimizde “[]” ifadesi ile karşılaşırız.

CEVAP : B

41. Dizi sıralanmadan arama yapılmıştır. Aranan kelime dizide olduğundan dizi 1 olarak dönüş yapacaktır. Kaçınıcı eleman olduğunu belirtmez.

CEVAP : B D

42. Q42 uygulamasında int bir diziyi String değeri atanmaya çalışılmıştır.

CEVAP : B

43. Q43 uygulamasındaki dizinin boyutu [2][1] olan bir dizidir. Uzunluğu 2 dir. İlk satırda 1 eleman olduğundan 1. Satırın uzunluğu 1 dir.

CEVAP : A

44. Q44 uygulamasında uzunluğu olmayan bir dizinin ilk elemanına erişilmeye çalışılmıştır [ArrayIndexOutOfBoundsException](#) hatası alınacaktır.

CEVAP : C

45. Q45 uygulamasında dizinin ilk elemanına erişilemez çünkü dizinin ilk elemanı [0] şeklinde ulaşılmalıdır. Dizinin uzunluğu 7 dir fakat ilk eleman yazdırılmadığından 6 eleman ekrana yazdırılır.

CEVAP : A

46. Q46 uygulamasında Dizi uzunluğu 0 olarak bulunmuştur.

CEVAP : A B

47. Q47 uygulamasında sıralama yapılmadan ekrana çıktı sağlanmıştır. İlk eleman olmasından dolayı 0 döner.

CEVAP : A

48. Derleyici hatası olmadan (String[] arg)'ten (String... arg)'e sorunsuz bir imza geçişi yapılabilir fakat tam tersi mümkün değildir.

CEVAP : A

49. `int[][][] nums1a, nums1b; // ikiside 4D dizidir.`

`int[][][] nums2a[], nums2b; // İlk eleman 4D ikinci eleman 3D dizidir.`

`int[][] nums3a[], nums3b[]; // ikiside 4d dizidir.`

`int[] nums4a[][][], numbs4b[][][]; // ikiside 4d dizidir.`

CEVAP : B

50. Q50 uygulamasında integer döndüren bir metod String bir ifadeye atandığından derlenmeyecektir.

CEVAP : C

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!