

HomeWork_8_Sonuç_Raporu

TURGUT YILDIZ

05.06.2020

CEVAPLAR

1. Q1 uygulamasının think() metodunda try{} bloğu kullanılmış fakat bu metod catch{} veya finally{} bloğu ile devam ettirilmesi gerekmektedir.

CEVAP : D

2. Önce try{} bloğu eğer bu bloğa girmezse daha sonra catch{} bloğu en son finally bloğu{} okunarak ilerlenir.

CEVAP : B

3. RuntimeException classı diğer sub Exception classlar gibi Exception classını extend eder. Burada **C ve D** seçenekleri kalır.

Daha sonra Throwable classından Exception classı extend edilir. D seçeneği b şekildedir. C şıkkında RuntimeException parent Throwable class ise subclass olarak belirtilmiş bu kesinlikle yanlış oluyor.

Ayrıca bu soru ile Error classıda Throwable classın sub classı oluyor. Bu soru ile öğrendiğimiz nokta bu kısımdır.

CEVAP : D

4. Error daha derlenme aşamasında kodun hiç derlenmeyeceği aşamadır. Error varsa kod derlenmeyeceği için catch edilemeyecektir.

CEVAP : A

5. Q5 uygulamasına bakıldığında bu soru yerel değişkenlerin tanımlandığı bloklar ile çözülebilir. Try{} bloğundaki bir değişken catch veya finally bloğunda tanımlı değildir. Bu soru çalışan bir kod yapabilmek için global bir değişken kullanılmalıdır.

CEVAP : D

6. ClassCastException = alt argümanda bir classa tip dönüşümü yaparken çevrilemeyen bir durum oluşursa. (Runtime)

ArrayIndexOutOfBoundsException = dizi sınırları aşıldığında (Run time)

IllegalArgumentException = hatalı argüman döndürüldüğünde (Run time)

IOException = bağlantının çalıştırılma aşamasında kapanma durumu (Compile time)

IOException hatası alınabilmesi için ilgili kodun çalışmayacak şekilde yazılması gerekir.

CEVAP : B

7. Eğer bir metod yakalayamadığı bir istisnaya neden olursa, metodu çağıranların kendilerini istisnaya karşı koruyabilmeleri için çağıran metodun bu davranışı belirtmesi gerekir. Yazdığımız Java programında açıkca throw ifadesini kullanarak bir istisna fırlatmak mümkündür.

<https://okanuzun.com/java-throw-ve-throws-kavrami/>

CEVAP : A

8. Javada önce subclass daha sonra parentclass okunacak şekilde derlendiğinden ve IOException Exception classının subclassı olduğundan IOException kesinlikle önce okunmalıdır.

CEVAP : B

9. Q9 uygulamasında kod derlenmemektedir. Bu kodun derlenmeme nedeni yanlış hata referansı kullanılmasıdır.

CEVAP : D

10. Q10 uygulaması try-catch-finally kod bloklarını kullanan bir uygulamadır. //g3 satırı derlenebilmesi için main() metodu ya throw Exception refere etmeli veya try-catch blokları kullanarak metoda erişmelidir.

CEVAP : C

11. ArithmeticException ve NullPointerException RuntimeException classının subıdır. RuntimeException ise Exception classının subudur ve bütün exception türleri bu classtan kalıtılır.

CEVAP : B

12. Q12 uygulamasına bakıldığında dizi sınırı aşılması durumu söz konusu değildir. bundan dolayı try{} bloğunda throw olarak belirtilen kod cacth(Throwable) {} bloğudur.

CEVAP : A

13. Önceki yaptığım uygulamalardaki gibi örneğin Q12 final tanımlandığında {} blok yapısı gereklidir.

CEVAP : C

14. Q14 uygulamasında IOException hatası kullanılmıştır. Daha sonra başka bir catch bloğunda bu classın subclassı olan FileNotFoundException hatası kullanılmıştır. İlk hata tüm programı kapsayacağından tekrar kullanılması hata verecektir.

CEVAP : C

15. Try bloğu yalnız kullanılmaz ya catch yada finally de kullanılmalıdır.Q12 uygulamasındaki 2'si birdende kullanılabilir.

CEVAP : C

16. Exceptionlar program ilgili haa yoluna girdiğinde fırlatılır. Yan program çalışmaya devam ederken gelir. Exception'ı encapsüle edebilirsek exception bölümü çalışmayacağından hatada vermez ve program çalışmaya devam eder.

CEVAP : B

17. Q17 uygulamasında catct bloğu parametre bölümünden sonra işleteceği kodları süslü parentez { } içinde yazılır.

CEVAP : D

18. 11. Sorudaki gibi bir mantıkla bu soru çözülebilir. Exceptionlar genelde daha dar olan subclasslar için parentclassı orada çağırıldıysa kaldırılmalıdır. Exception kaldırılması ile diğer kodlarda sorun olmaz ama Exception throwu kllanılırsa diğer bütün sub exceptionlar geçersiz olacaktır.

CEVAP : B

19. Exceptionlar java.lang kütüphanesinde tanımlanmıştır. Ama java.lang kütüphanesi otomatik tanımlanan bir kütüphane olduğundan tekrar deklare etmeye gerek yoktur.

CEVAP : C , şaşırtmalı soru ise CEVAP : D

20. Q20 uygulamasında catch bloğu için öncesinde (Exception e) gibi bir ifade olmalıdır. Belli bir parametre almalıdır.

CEVAP : C

21. –

22. Q22 uygulamasında kullanıcı tarafından girilen tüm Exceptionlar throw ile deklare edilmelidir. Çünkü bu hataları derleyici bilmemektedir bunları derleyiciye bildirmemiz gerekmektedir.

CEVAP : B

23. Birden fazla catch bloğu varsa ilk refere edilmiş olan catch çağırılır daha sonra diğer catchler işletilmez varsa finally işleilir.

CEVAP : A

24. Q24 uygulamasında Try'da bir nesne oluşturulmuşsa bunu ya o metoda throw olarak yada ayrıca bir catch açıp PrintStackTrace exceptionnı çalıştırılması gerekmektedir. Bu hali ile kod çalışmayacaktır.

CEVAP : C

25. –

26. Varolmayan bir değere veya nesneye ulaşmaya çalıştığımızda nullpointerexception döner.sonsuz döngüde ise stackoverflowexception döner.

CEVAP : B

27. Exception deklare etmek kullanıcının kendi exception ifadesini yadığında, bir exceptiondan kurtulmaya çalışıldığından ve birden fazla catch kullanıldığında istediğimiz hatayı almak istediğimizde throw olarak eklememiz gerekir.
CEVAP : C
28. Q28 uygulamasında try-catch-finally sıralamasından farklı bir kod yazımı yapılmıştır. Bu sıralamanın dışındaki kod yazımı hatalıdır.
CEVAP : D
29. Try bloğunun 0 veya 1 tane finally 0 veya çok sayıda ise catch bloğu olabilir.
CEVAP : A
30. Q30 uygulaması derlenir fakat 5/0 tanımsız değerinden dolayı ArithmeticException haası döner.
CEVAP : D
31. Q5 uygulamasında yaptığım uygulamada eğer catch ve finally blokları varsa ve bunlardan hangisine girilir diye soruluyorsa finally bloğu işletilir.
CEVAP : B
32. Q32 uygulamasında try – catch bloğu kullanmadan throw kullanıldığından //m1 satırı hatalıdır.
CEVAP : A
33. Q33 uygulamasında subclass ile yapılan 2. Exception obje ataması cast edilememektedir. ClassCastException hatası verir.
CEVAP : A
34. –
35. Q35 uygulamasında yapılan örnekte hiçbirine uymayan farklı ir çıktı sunmuştur.
CEVAP : D
36. ClassCastException RuntimeException dan türetildiğinden önce ClassCastException catch bloğunda yer almalıdır. Diğer türlü RuntimeException ClassCastException'ı kapsadığından ClassCastException'nın bulunduğu catch oknmayan bir kod satırı olur.
CEVAP : A
37. Bu soruda en uygun olan şık C'dir çünkü exception çalışan bir kodda önemli hatalar alınacaksa ve hatanın nedeninin ne olduğu ile ilgileniliyorsa Exception kullanılır.
CEVAP : C
38. Q38 uygulamasında Exception RunTimeException'nın parenti olduğundan uygulamadi gibi eklenmelidir.
CEVAP : C

39. Genel anlamda exceptionlar try-catch blokları ile belirtilirse uygulamada hangi hatanın alındığı ve bu hata karşısında ne yapacağımızı belirleriz. Bundan dolayı NullPointerException’da try-catch bloğu ile kullanılırsa hata karşısında uygulama hatayı bize bildirir ve görmezden gelebilir.

CEVAP : D

40. Q40 uygulaması Exception clasından türetilen RuntimeException üretilirse kod derlenir.

CEVAP : D

41. Q41 uygulamasına bakıldığında try-catch-finally bloğu kullanıldığından finally bloğu kesin olarak işletilir bundan dolayı RuntimeException alınır.

CEVAP : C

42. Q42 uygulamasında return bir exception olmadığından RuntimeException hatası dışında bir hata vermez.

CEVAP : D

43. Q43 uygulamasında catch bloğunun imza kısmına değişken verilmemiştir bundan dolayı kod derlenmez. Eğer değişken Error’dan türetilirse kod derlenir.

CEVAP : D

44. Q44 uygulamasında finally bloğu kesin çalışacaktır. Exception nesnesi try-catch te catch bloğunda yakalanıyor. Bundan dolayı RuntimeException ve finally içindeki değer çıktı alınır.

CEVAP : D

45. Hem IllegalArgumentException hemde ClassCastException Exception’ın subclassıdır. Bundan dolayı 2 Exception’da çağrılabilir.

CEVAP : C

46. –

47. Q47 uygulamasında try-catch te new ile bir nesne oluşturup bunu throws ile hata mesajı şeklinde tanımlamak yanlıştır.

CEVAP : D

48. Lang sınıfında veri türleri örneğin NullPointerException gibi bellek tahsisi hataları bilgisi barındırır. Bundan dolayı yetersiz memory için atılabilecek Exception bu sınıfta yer alır.

CEVAP : D

49. Q49 uygulamasında eğer catch bloğuna düşülürse burada yazılımın kontrol edeceği 2 tane catch { } vardır. Bunların değişkenide aynı isimde olduğundan derleme hatası verecektir.

CEVAP : C

50. Q50 uygulamasında Exception classı newlenerek yeni bir nesne oluşturuluyor ve bundan finally bölümünde hata mesajı vermesi isteniyor. Bu hata mesajını alabilmek için metoda ve o metodu kullanan main'e throw olarak Exception classı eklenmelidir. Veya throw olarak zorlamadan new önündeki throw ifadesi kaldırılırsa kod derlenir.

CEVAP : B