

HomeWork_7_Sonuç_Raporu

TURGUT YILDIZ

29.05.2020

CEVAPLAR

1. Q1 ve Q1_extends classlarından oluşan uygulamada hem private yapılmış bir değişkene erişim sağlandığından hem de subclasstan parentclassın bir metoduna erişirken subclassa süper() metodu eklenmelidir.

https://www.w3schools.com/java/java_inheritance.asp

CEVAP : C

2. Abstract classlar public erişim metodu ile kullanılırlar. protected, static ve final olarak kullanılması o classın abstract özelliğinin yok edecektir. Çünkü bu classlar altclasslardan çağrılabilen override yapılabilen classlardır.

CEVAP : D

3. Aynı isimde metodlar farklı parametreler almazsa kod derlenmez. Override olabilmesi için farklı parametreler alınması gerekir.

CEVAP : C

4. Kalıtım alt classların kalıtım aldığı parent classların metodlarını ve özelliklerini kullandığı sistemdir.

CEVAP : A

5. Q5 uygulamasında metod isminden önce class ismi getirilmesi ile ilgili metodun interface mi yoksa subclass mı ait olduğu belirtilmelidir.

CEVAP : C CEVAP : A

6. Henüz üretilmemiş bir class kullanacaksa bunun için üretilmeyen classın interface'ini yazıp onu kullanmak en mantıklısıdır. Hem entegrasyonu hem de kod üretim sürecini kolaylaştırır.

CEVAP : B

7. Q7 uygulamasında parentclassın subclassından bir nesne oluşturuluyor. Yapılan örnekte subclass ve parentclass üzerine ayrı ayrı örnekler yapılmıştır.

CEVAP : B

8. Çoklu kalıtıma izin verilir. Birden fazla parentclass ile kalıtım sağlayabilen subclasslar oluşturulabilir.

CEVAP : D

9. Q9 uygulamasında subclass ve parentclassdaki metod() ların erişimi public olmalı ve dönüş değeri belirtilmelidir.

CEVAP : C

10. **Overriding metodlar sub classları ezmekten ziyade parentclasslardaki metodları overriding ile ezmek yerine geçirmek için kullanılır.**
CEVAP : C
11. Parent classtaki final ifadeden dolayı derlenmeyecektir. Eğer düzgün derlenseydi, subclasstaki 3 değeri cevap olacaktı.
CEVAP : C
12. Q12 uygulamasında subclassın bir nesnesi oluşturuluyor. Daha sonrada bu classtan oluşturulmuş medot() çağrılıyor.
CEVAP : A
13. İnterfaceler publictir. Ayrıca static olarak tanımlanabilir.
CEVAP : B
14. Q14 uygulamasında Sprint classının walk() metodu çalıştırılır.
CEVAP : C
15. 1 interface başka bir interface’i extend edebilir fakat implement edemez. Classa intefaceimplement edilebilir.
CEVAP : B
16. Q16 uygulamasına bakıldığında height değişkeni süper ile erişilmeye çalışılıyor fakat private olarak tanımlanmış. Bu değişkeni protected veya public olarak tanımlarsak erişebiliriz.
CEVAP : D
17. Abstract hem soyut hem de normal olarak tanımlanırken. İnterface yani arayüz sınıflar sadece soyut olarak tanımlanır.
CEVAP : D
18. Q18 uygulaması 3 farklı classtan oluşmuştur. Main() metodunun bulunduğu class abstract yapıldığından bu classla ilgili bir nesne oluşturulursa kodun o satırı hata verecektir. // g3
CEVAP : C
19. Integer, Short ve Number subclass olmadığından boşluğa yazılacak bir subclass yoktur.
CEVAP : D
20. İnterfaceler implemets edilebilir. abstract classlar ise extend edilebilir.
CEVAP : C
21. Q21 uygulamasına bakıldığında extend edildiği classın metodu çağırılır.
CEVAP : A

22. unknownBunny subclass olursa myBunny classı ile aynı yeri gösterebilir.

CEVAP : B

23. Q18 uygulamasındaki gibi abstract classlar protected yapılabilmektedir.

CEVAP : D

24. Q24 uygulamasına bakıldığında interfacer “implements” edilir. Abstractlar “extends” edilir.

CEVAP : D

25. Bir subclassa erişim için superclass’a dolaylı olarak erişip daha sonra subclassa bu referans üzerinden erişilir. Bunada dolaylı erişim denir.

CEVAP : B

26. Interfaceler final, static ve public olabilir. Fakat abstract olamaz.

CEVAP : B

27. Q27 uygulamasında abstract ile extends edilmiş bir proses işletilmiştir. Burada önemli olan önce abstract class işletilir daha sonra subclass işletilir.

CEVAP : C

28. Overloading metodlarda hep aynı isimler kullanılır, sadece parametreler değiştirilir.

CEVAP : C

29. Q29 uygulaması takip edildiğinde sonuca ulaşılacaktır. Size = 5

CEVAP : A

30. Parent classler ve static metodlar hiding değişken ve metod yapılabiliirler.

CEVAP : C

31. Q31 uygulamasında hata x2 satırında gerçekleşir. Static bir metod static olmayan bir metodu override edemediğinden çağırıldığı satırda hata verecektir.

CEVAP : B

32. Q32 uygulamasında ilgili fly() metodu abstract yapılmış fakat bir subclass olmadığından derleme gerçekleşmeyecektir.

CEVAP : C

33. Bir class başka bir classa superclass olarak atanabilir. Altclasslar ise parentclassa extend edilebilir.

CEVAP : B

Bir

34. Concrete classlar abstract olmayan bir abstract subclassları uygulamak için kullanılır.

CEVAP : C

35. 1-) Fly() metodu static veya default olarak tanımlandığından hatalıdır.
2-) Bird sınıfındaki fly metodu return dönüş tipine sahip değildir. hatalıdır.
3-) final bir class extend edildiğinden hatalıdır.

CEVAP : D

36. Hem abstract hemde interface classlar default metod içerebilir.

CEVAP : B

37. Q37 uygulamasında imzası aynı olan metod() ların override edilemediği görülecektir.
Bun çözmek için extend edilen classları süper ile belirtmek gerekir.

CEVAP : C

38. Private, final, static olmayan classlar sanal olarak belirtilir.

CEVAP : A

39. Classlar ve interface extends edilebilir.

CEVAP : B

40. Q40 uygulamasına bakıldığında complexmath subclassındaki secret değeri
çalıştırılacaktır.

CEVAP : B CEVAP : A

41. Burada public ve final özelliklerin override edilebildiği bilinmelidir. Ayrıca private veya erişim sınırlandıran metodlar override edilememektedir.

CEVAP : D

42. Q42 uygulamasına ve extends olan classlara bakıldığında setAnimal metodunun Wolf
classından türetilmesi gerekmektedir.

CEVAP : C

43. İnterfaceler final olamazlar. Soyutlanacak sınıflar final olursa somutlaşacaktır hata
verecektir.

CEVAP : A

44. Q44 uygulaması normal bir şekilde çalışacaktır ve ekrana print içindeki değeri
basacaktır.

CEVAP : A

45. Override metodlar farklı dönüş türüne sahipken overridden metodlar aynı dönüş türüne
sahiptir.

28. soruya bakıldığında bu soruda aynı mantıkla yapılabilir. Override parentclasstan
aynı metodla ve parametrelerle yazılmış başka metodları ezmesine denilmektedir.
Soruda ise bize bu seçenek bize verilmemiştir. Yani son bölüm yanlıştır.

CEVAP : A CEVAP : D

46. Parentclassta bir constructor metod yoksa subclasslarda constructor metod olmalıdır.

CEVAP : B

47. Bir classta nesne oluřturulursa bu bellekte o nesneye ait hangi verilerin olduėunu g sterirken, referance type ise caller tarafından eriřilebilecek verileri belirtir.

CEVAP : D

48. Q48 uygulamasında abstract classtaki play() metodu Long t r nden olduėundan overloading i in ve kodun hata vermemesi i in metod t r ne “Long” yazılır.

CEVAP : A

49. Se eneklere bakıldıėında default interface yapılmasının en  nemli nedeni mevcut interfacelerin eskiye d n k programları kapsamayı gerektirmektedir.

CEVAP : B

50. Q50 uygulamasına bakıldıėında Q50_ext classına throws olarak IOException eklenirse uygulama derlenecektir. Parentclass eėer subclassta eklenmiřse bunu derlemek i in throws b l mleri eksiksiz eklenmelidir.

CEVAP : C