

HomeWork_6_Sonuç_Raporu

TURGUT YILDIZ

22.05.2020

CEVAPLAR

1. Protected aynı paket içindeki subclassların erişimine izin verir. Package-private paketteki her işleme izin vermektedir.

CEVAP : C

2. this() metodu mevcut sınıfın kurucusunu çağırmak için oluşturulmuş metoddur.

CEVAP : A CAVEP : B

3. Q3 uygulamasına bakıldığında dönüş tipi tanımlanmış bir metoda gövde bölümünde bir return ifadesi olması gerekir. Eğer return ifadesi if-else içerisinde olursa derleyici bunu derlemeyecektir.

CEVAP : D

4. Dönüş tipi olmayan bir metod() çağırılmıştır. Bundan dolayı derlenmeyecektir.

CEVAP : D

5. Öncelikle bir dilin pass-by-value ya da pass-by-reference olduğundan söz edilirken burada o dilin metodlara/fonksiyonlara parametre aktarırken parametreleri metoda ne şekilde aktardığı kastedilmektedir. Pass-by-reference demek parametrenin gerçek adresini geçmek anlamına gelirken pass-by-value ise sadece değerini bir kopya ile geçmek şeklinde tanımlanabilir. Her metodun bir stack'i vardır ve metoda geçilen parametreler bu stack'e push edilir, java'da kopyalanır. Nesnenin referansının metoda geçilmesi pass-by-reference yapıldığı anlamına gelmez, java'da referans da kopyalanarak geçilir, java pass-by-value'dir.

<https://kodcu.com/2012/06/java-pass-by-value/>

CEVAP : B

6. get() metodu değer döndürdüğünden void olmamalıdır. set() metodu ise dışarıdan bir para metre almalıdır. Dönüş tipide olmamalıdır.

CEVAP : C

7. süper() ve this() metodları aynı anda kullanılamaz sadece this() veya 2 ayrı süper() metodu ile kullanılmalıdır.

CEVAP : B

8. soruda A seçeneğinde Public değil public olarak tanımlanmalıdır. Metodun dönüş değeri integer olduğundan int,long gibi değişken tanımlamaları kullanılmalıdır.

CEVAP : B

9. soruda her zaman erişim sağlanacak ifadeyi sormaktadır. Public ve static 2 side her zaman erişim sağlamaktadır.

CEVAP : A, C

10. Q10 uygulamasında oluşturulan classın önce parametresiz constructor() metodu çalışır fakat bu sonuca etki etmez. Çünkü bu classın değişkeni çağırılmaktadır. Daha sonra this(4) metodu ile parametrelili kurucu çalıştırılır.

CEVAP : A

11. Package-private ifade aynı pakkette erişime izin vermektedir.

CEVAP : B

12. Get() metodları return dönüş tipine sahiptir ve public erişimi ile erişilir.

Get() metodun bir dönüş tipi olmamalıdır.

CEVAP : A CEVAP : D

13. no-argument constructor class ilk çağırıldığında çalışacak olan alanlardan birisidir. Eğer class parametre ile çağırılırsa parametrelili olan constructor çalışır.

CEVAP : C

14. bir metoddaki en fazla 1 tane varyans içerir ve bu son eleman olmalıdır.

CEVAP : A

15. Q15 uygulamasında slalom() metodu çağırıldığında metod en son nesneyi null a eşitler fakat final olan değişkenler hayatını sürdürmeye devam edecektir.

mySkier = ski-age=18

myName = "Rosie"

mySpeed = "0"

CEVAP : C

16. Overloading için metod() ismi aynı fakat parametrelerde eksiklik veya fazlalık olması gerekmektedir.

CEVAP : B

17. Encapsulation programın değişmesini önlediği gibi daha pratik kod geliştirmeye yardımcı olur. Fakat performans ve eş zamanlı çalışma konusunda bir artışı yoktur.

CEVAP : D

18. Metodlar dizilerin elemanlarını değiştirebilir. Çünkü adres ile iletilmektedir.

CEVAP : A

19. Classın bir metodu erişmek için class.metod şeklinde erişilir.

CEVAP : B

20. Return ifade ile void, byte veya String tip döndürülmez. Long,int gibi sınırları ve tanımlaması olan değişkenler döner.

CEVAP : D

21. Q21 uygulamasında final bir değışkenden değışiklik yapılması derlemede hata verecektir. 2 farklı yerde değışiklik yapılırsa derleme gereklenir.
CEVAP : C
22. Süper() metodu parent class'taki constructor() metodu çağırırken kullanılır. This() metodu ile karıştırmamalıdır. This() metodu doğrudan referans ile overloading olan constructor() metodlar arasında önceliğı belirleyebilir.
CEVAP : D
23. Package-private bir ifade olduğundan paket içerisinde erişim sağlanabilir.
CEVAP : B
24. Setter() ve getter() metodlar class içindeki private ve protected değışkelere doğrudan erişimi engellemek için oluşturulan metodlardır. 3 seçenekte doğrudur.
Kodun çalışabilmesi için min. Kod düzeni sorulmuş yalnız 1 olmalıdır set ve get metoda gerek yoktur.
CEVAP : C CEVAP : A
25. Metodlar sayı ile başlayamaz ayrıca “_” veya “\$” ifadeleri içerebilir. “-” ifadesini içeremez.
CEVAP : A
26. Q26 uygulamasında hem final hem de static bir metod yapılmaya çalışılmıştır, static ifadesi silinirse kod derlenir.
CEVAP : D
27. Nesnede yapılan bir değışiklik çağırılan değışkenin adresinde yansır nerede çağırılırsa orada yeni değeri sergilenir.
CEVAP : B
28. Q28 uygulamasında private bir değışken aynı zamanda final() ile kullanılmış bu değışkenin setter() metodu bulunmaktadır. Kodun derlenmesi için final ifadesi silinmelidir.
CEVAP : C
Muhtemel çıktı için ;
CEVAP : A
29. –**CEVAP : A**
30. Başka bir paketdeki static metoda erişmek için ;
import static clothes.Store.getClothes; şeklinde erişim sağlanabilir. Clothes paketi içindeki store sınıfının getClothes() metoduna erişilebilir.
CEVAP : C
31. 23. Sorudaki gibi bir metod veya değışken package-private yapılması için önüne herhangi bir erişim denetleyicisi konulmaz.
CEVAP : D

32. Q32 uygulamasında kurucu olarak düşünölen bir void metodda süper() metodu oluşturulmuştur. Void ifadesi silinirse kod derlenir.
CEVAP : B
33. Instance metodlar yerel static değışkenlere erişebilmektedir.
Static metodlar instance değerlere erişemezler.
CEVAP : A
34. Short dönüş tipine sahip bir metodda return ifadesine short veya daha küçük bir dönüş tipi eklenmelidir.
CEVAP : B
35. Overloading metodlar aynı isimde farklı parametre isminde olmalıdır.
CEVAP : C
36. Q36 uygulamasında 2 satırın kaldırılması ile derleme sağlanır.
CEVAP : B CEVAP : C
37. Q37 uygulamasında void bir metod kurucu olamaz. Kurucu metodun dönüş tipi olmadan tanımlanması gerekir.
CEVAP : D
38. Public metod her yerden erişim sağlanırken private metod aynı class içinden veya setter getter metodlar ile erişim sağlanır.
CEVAP : A
39. Q39 uygulamasında debugger ile bakıldığında phone nesnesinin size değışkeni ilk satırda set edildikten sonra hiç değıştirilmemektedir.
CEVAP : A
40. This.classname.metod ifadesi yanlış bir kullanımdır. This class ismini nitelediğinden 2. Kez classa ulaşmaya çalışıyoruz bundan dolayı hatalı erişimdir.
CEVAP : B
41. 3 tane parametre alan bir metod tanımlanmış 1. Parametre = int 2. Parametre = String 3. Parametre = String olmalıdır.
CEVAP : C
42. Static ifadelere classın tüm instance larından erişilebilir.
CEVAP : D
43. Hem this hemde süper metodları aynı anlama gelebilmektedir.
CEVAP : A
44. Q44 uygulamasında hiçbir final değışkeninin kaldırılmasına gerek yoktur. Sadece height değışkeni static yapılmalıdır.
CEVAP : D

45. Q45 uygulamasında extends yapılan sınıfın erişeceği değer süper olarak tanımlanmadığından derlenmeyecektir.

CEVAP : D

46. Q46 uygulamasında dönüşümler göz önüne alındığında byte ile int toplanır ve int bir sonuç döner. 2+1 'in byte olması için () içinde yazılması gerekirdi = (byte)(2+1) gibi

CEVAP : A

47. Q47 uygulamasında long değeri int olarak cast etmek gerekir. Diğer türlü int değeri gönderildiğinde overloading bir int metod() yoksa kod derlenmeyecektir.

CEVAP : C

48. Metod isimleri (\$) ile başlayabilir fakat (/), (%) ifadeleri ile hem başlayamaz hem barındıramaz. Ayrıca isimlendirmede (#) ifadesi barındırılmamalıdır.

CEVAP : A

49. Protected korumasına o class, altclasslardan ve bulunulan paketten erişilebilir.

CEVAP : B

50. Q50 uygulamasındaki ve newQ50 paketindeki uygulama gibi; Static olmadan bir paket eklenmesi gerekir ayrıca eklenen paketteki classtan bir nesne oluşturulması ile bu classa ait tüm metodlar kullanılabilir.

CEVAP : D