

1. C

Modifierlerden `protected`, aynı paketteki alt sınıfların ve üyelerin erişimine izin verir, `package-private` ise aynı paketteki üyelere erişime izin verir.

2. B

`super()` ifadesi üst sınıftaki constructorları çağırırken kullanılır, `this()` ise aynı sınıftaki yapıları çağırarak için kullanılır.

3. D

`sell()` diye bir fonksiyon olmadığı için derlenmeyecektir.

4. D.

`Nested()` fonksiyonu bir değer döndürmediği halde `print` yapılmaya çalışıldığı için hata verecektir.

5. ???

6. C.

`Get()` fonksiyonları bir değer döndürmelidir. `Set()` fonksiyonları da bir değer almalıdır.

7. B.

`This()` ve `super()` aynı constructor da bulunamazlar.

8. B.

A şıkkındaki `Public` büyük harfle başladığı için yanlış.

9. ??

10. A.

```
public class Q10 {
    private int rope = 1;
    protected boolean outside;
    public Q10() {
        this(4); // p1
        outside = true;
    }
    public Q10(int rope) {
        this.rope = outside ? rope : rope + 1;
    }
    public static void main(String[] bounce) {
        System.out.print(new Q10().rope);
    }
}
```

11. ??

12. ???

13. C

14. A.

Bir metod en fazla bir varargs parametresi içerebilir ve varargs listede son argüman olmalıdır.

15. ??

16.C ??

17. D

Encapsulation performansla ilgili bir durum değildir.

18. A

19. B.

20. D

21. C.

```
public final class Q21 {  
    public final static int finish(int score) {  
        final int win = 3;  
        int result = score++ < 5 ? 2 : win;  
        return result += win;  
    }  
  
    public static void main(final String[] v) {  
        System.out.print(finish(Integer.parseInt(v[0])));  
    }  
}
```

Score ve result da final ifadeler kaldırilmaliir.

22. D.

23. B.

signature metodu package-private dir. Yani aynı paketteki sınıflar tarafından erişilebilir

24. ??

25. A.

B seceneginde – ifadesi yanlis, C secenegi java da kayitli bir isim oldugu icin yanlis ,D secenegi rakamla basladigi icin yanlis

26. D.

```
public final static int getNumberOfChickens() {  
    return 10;  
}
```

Seklinde olabilirdi. İnt metod isminden once kullanilmali.

27. ?

28. C

```
private final Object contents;
```

final dan dolayi derlenmeyecektir.

29. A

JavaBean metodu get, set ve is prefixlerini kulanabilir.

30. C

A seceneginde staic ve import yerlerinin degismesi gerekiyot.

31. D.

Class icindekilerin pakete ait oldugunu belirtmek icin kullanılacak ozel bir modifier yoktur. Eger mdifier kullanilmamissa zate paket ozel oldugunu gsterir.

32. B.

```
public class Q32 {  
    private int inThe = 4;  
  
    public void Stars() {  
        super(); //sadece bu satirda derleme hatasi alinmaktadır  
    }  
  
    public Q32(int inThe) {  
        this.inThe = this.inThe;  
    }  
  
    public static void main(String[] endless) {  
        System.out.print(new Q32(2).inThe);  
    }  
}
```

33. A.

instance metot ya da constructor in tum static degiskenkere erisimi vardır.

34. B

```
public class Q34 {  
    public short calculateDistance(double lat1, double lon1, double lat2, double lon2) {  
        // iserttt here  
        return new Byte((byte) 6);  
    }  
}
```

35. C.

Overloaded metotlar ayni isimli olmalı ama farklı parametre listeleri olmalıdır.

Return tipleri aynı da olabilir farklı da.

36. B

```
public class Q36 {  
    // private static final String monday;  
    String tuesday;  
    // final static wednesday = 3;  
    final protected int thursday = 4;  
}
```

37. D

38. A

39. A.

```
public class Q39 {  
    private int size;  
    public Q39(int size) {  
        this.size = size;  
    }  
    public static void sendHome(Q39 p, int newSize) {  
        p = new Q39(newSize);  
        p.size = 4;  
    }  
    public static final void main(String... params) {  
        final Q39 phone = new Q39(3);  
        sendHome(phone, 7);  
        System.out.print(phone.size);  
    }  
}
```

40. B.

```
public class Q40 {  
    public static void water() {  
    }  
  
    public void get() {  
        this.Drink.water();  
    }  
}
```

41. C

Metotun 3 parametresi var int,string, string

42. D

static variable'lara sinifin butun instance larinfan erisilebilir.

43. A.

Constructor in ilk satiri this() 'te olabilir.

44. D

45. D

46. A.

```
public class ChooseWisely {  
    public ChooseWisely() {  
        super();  
    }  
}
```

```

    }
    public int choose(int choice) {
        return 5;
    }
    public int choose(short choice) {
        return 2;
    }
    public int choose(long choice) {
        return 11;
    }
    public static void main(String[] path) {
        System.out.print(new ChooseWisely().choose((byte) 2 + 1));
    }
}

```

47. C.

getScre Long turunde ama m2 satirinda int degisken gonderilmis.

48. A.

Metotlar bir harf, \$ ifadesi, ya da _ ifadesi ile baslamalidir

49. B.

protected modifier aynipaket icindeyse herhangi bir siniftan erisilebilir.

50. D