

1.C

Cinema class'ı theater package'inin icinde bulunmaktadir amaMovie class'i burada degildir. Bu yuzden Cinema classindan extend edilmeyecalisilan satirda ve name string'inin kullanilmaya calisilgi satirda hata alinacaktır.

2.B ?????????

"abstract" means: "Implements no functionality", and "static" means: "There is functionality even if you don't have an object instance". And that's a logical contradiction.

3.C

Ayni isimde iki metot bulundu icin hata alinacaktır.

4.A

5.A

İki tane class olamaz.

6.A

Abstract class olaak yazildiginda diger ekip istedginde senin kodunu dahil ederek kullanabilecektir.

7.D

The public type ElectricCar must be defined in its own file hatasi alinacak.

8.D

Java normalde multiple inharitance izin vermez yani extend dedikten sonra tek isim yazabiliriz. Ama interface kullanilarak da inharitance yapilabilir.

9.

10.C

11.C

Computer class'indaki final modifier'indan dolayi derlenmeyecektir. Eger modifier kaldırılırsa cvap 3 olacaktır.

12.??

13. B

İnterface metotlarda static modifiyeri kullanilabilir.

14.D

15.C ,B

16.D

Ship class'ina height private ldugu icin diger class dan erisilemiyr.

17.D

18.C

Cannot instantiate the type IsoscelesRightTriangle hatasi alinacatir.

19.D

Play metotu hem Horn hem de Woodward da oldugu icin kod derlenmeyecektir.

20.C

Bir class abstract class'i extent ettiginde de interfece'i impement edebilir.

21.A

```
package paper;
public class Encyclopedia extends Book {
    public static String material = "cellulose";

    public Encyclopedia() {
        super();
    }

    public String getMaterial() {
        return super.material;
    }

    public static void main(String[] pages) {
        System.out.print(new Encyclopedia().getMaterial());
    }
}
```

22.A ???

23. D

protected, package-private, and public access modifiers can each be applied to abstract methods.

24.D

Mars Planetten extend , Sphere den implement edilmelidir.

25.??

26. B

Interface variables public, static, and final.

27.C

```
package race;
public class BlueCar extends Car {
```

```

    {
        System.out.print("4");
    }

    public BlueCar() {
        super("blue");
        System.out.print("5");
    }

    public static void main(String[] args) {
        new BlueCar();
    }
}

```

28.C

Overloaded methodsyani asiri yuklenmis metotta ayni isimden bircok metot olusturabiliriz ama parametreleri farkli olacak. Override metot ise ust siniftan extend ettigimiz metot icinde degisiklikler yapabiliriz. Yani ikisinde ortak olan isimlerinin ayni olmasidir.

29. A

```

package sports;
public class SoccerBall extends Ball implements Equipment {
    public SoccerBall() {
        super(5);
    }

    public Ball get() {
        return this;
    }

    public static void main(String[] args) {
        Equipment equipment = (Equipment) (Ball) new SoccerBall().get();
        System.out.print(((SoccerBall) equipment).size);
    }
}

```

30.B ??

31.B

Rectangle class'ında getEqualSides metodu static oldugu icin x2 satirinda hata alınacaktır.

32.C

class Rotorcraft abstract metot icerdigi icin abstract yapılmalıdır.Bu sekilde derlenemez.

33.B ?

33.D ?

35.D

Bird classından final kaldırilmali.

Public int fly () fonksiyonuna return parametresi eklenmeli

Public void fly() metodu static yapılmalı

36.B

Sadece interface default metot icerebilir.

37.C

İki tane implemet yapılmaya çalışılmış burada hata alınıyor. Birisinin override edilmesi gerekiyor.

38.A

Static, final ve private metotlar virtual metot olamıyorlar.

39.B

40.A

```
public class InfiniteMath extends ComplexMath {  
    public final double secret = 8;  
  
    public static void main(String[] numbers) {  
        Math math = new InfiniteMath();  
        System.out.print(math.secret);  
    }  
}
```

41.??

42.C

```
package Q42;  
public class Zoologist {  
    Canine animal;  
  
    public final void setAnimal(Dog animal) {  
        this.animal = animal;  
    }  
  
    public static void main(String[] furryFriends) {  
        new Zoologist().setAnimal(new Wolf());  
    }  
}
```

43.C

44.A

```
package party;  
  
abstract public class Ballroom extends House {  
    protected abstract Object getSpace();  
  
    public static void main(String[] squareFootage) {  
        System.out.print("Let's start the party!");  
    }  
}
```

45.A

46.B

47.D

48.A

```
package orchestra;

public class Violin extends StringInstrument implements MusicCreator {
    public Long play() {
        return 12L;
    }
    //ayrca Long oldugunu belirtmekte gerekityor.
}
```

49.B

50.??