

T.C. DÜMLUPINAR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

NESNEYE YÖNELİK ANALİZ VE TASARIM VİZE SINAVI

23.03.2015

SÜRE: 70 DAKİKA

SORULAR

```
public interface A {}  
public interface B {}  
public class C implements A, B{}  
public abstract class D implements A{}  
public class E extends D{}  
public class F extends E implements B{}
```

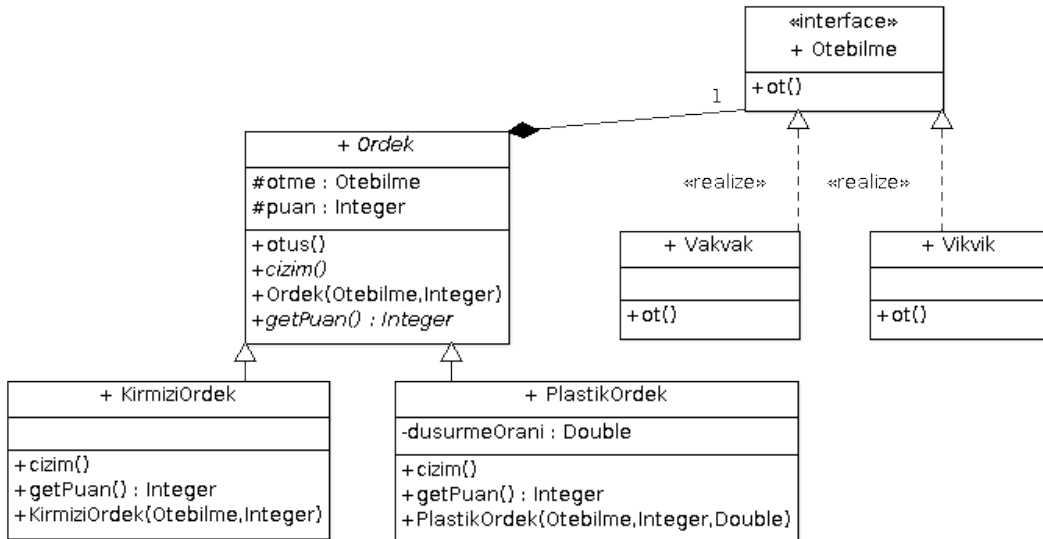
1. Yukarıdaki verilen sınıf sıradüzenine göre aşağıda verilen seçeneklerde yer alan ifadelerin Java dilinde kabul edilip edilmeyeceğini, her bir seçenek için ayrı ayrı olmak üzere “Derleme Hatası / Çalışma Zamanı Hatası / Doğru” anahtar kelimelerinin birisini kullanarak belirttikten sonra eğer hatalı ise nedenini kısaca açıklayınız. (Her biri 3 puan)

I. A a = new E();	VI. B b4 = new F(); C c4 = (C)b4;
II. B b = new E();	VII. B b5 = new C(); A a5 = (A)b5;
III. A a3 = new D();	VIII. D d4 = (D)new A();
IV. A a4 = new F(); B b2 = (B)a4;	IX. D d5 = (A)new C();
V. A a2 = new E(); D d = (D) a2;	X. B b6 = new F(); D d6 = (D) b6;

2. Kapsülleme (Encapsulation) ile bilgi saklama (information hiding) prensipleri arasındaki ilişkiyi ve aralarındaki farkı belirtiniz. Kapsülleme yapan fakat bilgi saklama işlemini gerçekleştiremeyen basit örnek bir sınıfı Javada kodlayarak gösteriniz. (20 puan)

3. Aşağıdaki UML diyagramının ürettiği Java/C# kodunu yazınız. (30 puan)

Not: Ördek soyut sınıf ile bu sınıf içerisindeki çizim() ve getPuan() metodları soyut metodlardır. Sınıflara ait constructorlar (yapılandırıcı metodlar) ile Ördek sınıfındaki otus() metodunun içi dolu olmalıdır.



4. Aşağıdaki java kodunun UML sınıf diyagramı nasıl olmalıdır? (25 puan)

```
public interface A {
    public String methodA();
    public int methodB(double val);
}

public interface B {
    public void methodK();
}

public class C implements A{
    private F instanceF;

    public C(){
        instanceF = new F();
    }

    public String methodA() {
        return "a method";
    }

    public int methodB(double val) {
        return (int)val;
    }
}

public abstract class D implements A{
    public String methodA() {
        return "a method";
    }

    public int methodB(double val) {
        return (int)val;
    }

    protected abstract void methodC(int val);
}

public class E extends D{
    protected void methodC(int val) {
        System.out.println(val+" in methodC");
    }

    public void methodD(C sample){
        //do something with sample C
    }
}

public class F extends E{
    private B[] items;
    private int index;

    public F(){
        items = new B[3];
    }

    public void addNewItem(B item){
        items[index++] = item;
    }
}
```