

**BURSA TEKNİK ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**BLM0121 – Nesneye Yönelik Programlama**  
**Bütünleme Sınavı**

Ad&Soyad	: <b>CEVAP ANAHTARI</b>
Öğrenci Numarası	:

Akademik yıl : 2023-2024  
Dönem : Bahar  
Tarih : 27 Haziran 2024 – 13:00  
Sınav süresi : 90 dakika  
Öğr. görevlisi : Doç. Dr. Ergün GÜMÜŞ

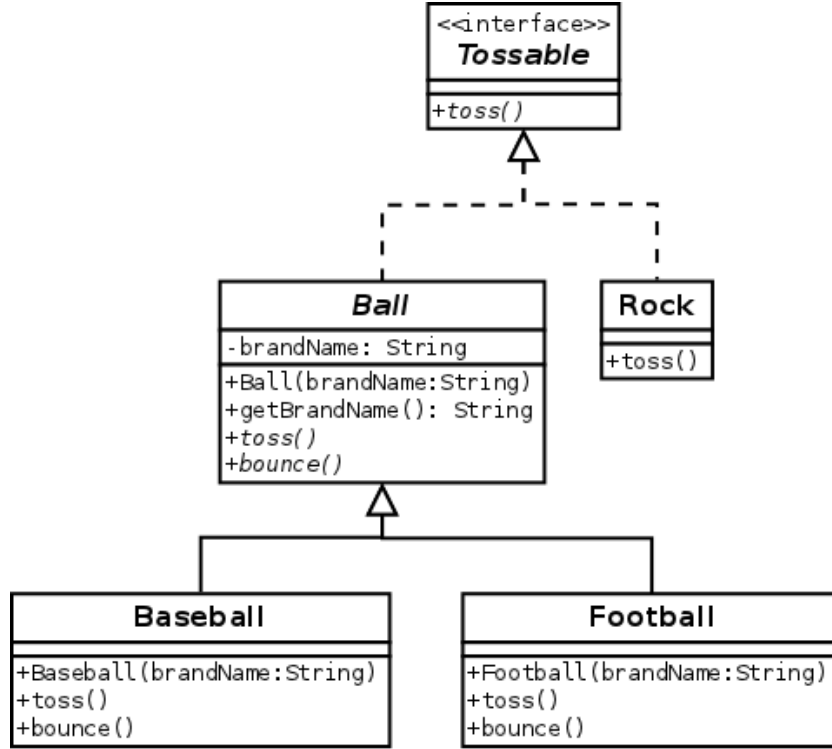
Soru	1	2	3	4	5	Toplam
Puan	25	20	25	15	15	100
Not						

**KURALLAR**

- Sınava başlamadan önce Ad&Soyad ve Öğrenci numarası alanlarını doldurunuz.
- Sınav öncesinde ve süresince sınav gözetmenlerinin tüm uyarılarına uymanız gerekmektedir.
- Sınav öncesinde cep telefonlarınızı KAPATINIZ!
- Soruları yanıtlamak için sadece sınav kâğıdınızla beraber verilen kâğıtları kullanmanız gerekmektedir. Yanıtlarınız açık ve okunaklı olmalıdır.
- Sınav boyunca masanızın üzerinde bulunabilecek malzemeler sadece sınav kâğıdınız, kalem ve silgidir.
- Sınav süresince herhangi bir nedenle birbirinizle konuşmak ve malzeme (silgi, kalem, kâğıt vb.) alışverişi yasaktır.
- Bu kuralların herhangi birine uymamak kopya çekmeye yönelik bir hareket olarak değerlendirilir ve ilgili makamlara bildirilir.

## Sorular

1)



Yukarıda görülen UML diyagramını inceleyiniz. İtalik yazılan sınıf ismi ve metod isimlerine dikkat ediniz.

Buna göre,

i) Diyagramda görülen tüm yapılara ait Java kodlarını yazınız.

ii) Bir test sınıfı oluşturup bu sınıftaki `main()` metodu üzerinden bir adet Baseball topu, bir adet Football topu ve bir adet taş yaratınız.

iii) Yarattığınız nesneler üzerinden `toss()` ve `bounce()` metodlarını çağırınız. Yaptığınız çağrılar sonucunda aşağıdaki konsol çıktısını elde etmeniz gerekmektedir. Kodlarınızı bu çıktıları alacak şekilde geliştiriniz.

### Konsol çıktısı:

```
A marka beyzbol topu atildi
A marka beyzbol topu sekti
B marka futbol topu atildi
B marka futbol topu sekti
Tas atildi
```

```
interface Tossable{
    public void toss();
}
```

```
abstract class Ball implements Tossable{
    private String brandName;
    public Ball(String brandName) {
        this.brandName=brandName;
    }
    public String getBrandName() {
        return brandName;
    }
    public abstract void toss();
    public abstract void bounce();
}
```

```
class Rock implements Tossable{
    public void toss() {
        System.out.println("Tas atildi");
    }
}

class Baseball extends Ball{
    public Baseball(String brandName) {
        super(brandName);
    }
    public void toss() {
        System.out.println(getBrandName()+" marka beyzbol topu atildi");
    }
    public void bounce() {
        System.out.println(getBrandName()+" marka beyzbol topu sekti");
    }
}

class Football extends Ball{
    public Football(String brandName) {
        super(brandName);
    }
    public void toss() {
        System.out.println(getBrandName()+" marka futbol topu atildi");
    }
    public void bounce() {
        System.out.println(getBrandName()+" marka futbol topu sekti");
    }
}

public class deneme2{
    public static void main(String[] args) {
        Ball[] toplar= {new Baseball("A"),new Football("B")};
        for(Ball top:toplar){
            top.toss();
            top.bounce();
        }
        Rock tas=new Rock();
        tas.toss();
    }
}
```

2) Elimizde "isimler.txt" adında bir dosya bulunmaktadır. Bu dosyanın içinde bu yıl doğan erkek ve kız bebeklere verilen isimler ve bu isimlerin sayısı bulunmaktadır. Dosyamızın ilk 5 satırı aşağıdaki gibidir:

1	Liam	20,802	Olivia	15,270
2	Noah	18,995	Emma	13,527
3	Oliver	14,741	Charlotte	12,596
4	James	11,670	Amelia	12,311
5	Elijah	11,452	Sophia	11,944

Her satırda tab karakterleri ile ayrılmış 5 adet değer bulunmaktadır. İlk değer sıra numarasını, 2. ve 3. değerler bu yıl en çok konan erkek bebek ismini ve sayısını, 4. ve 5. değerler ise bu yıl en çok konan kız bebek ismini ve sayısını göstermektedir. Dosyada kaç satır olduğu bilinmemektedir. Buna göre kullanıcıdan alınan bir ismi bu dosyada arayan, bulursa ismin erkek veya kız ismi olduğu bilgisiyle beraber kaçınıcı satırda olduğunu raporlayan bir Java kodu yazınız. Programınızın çıktısı aşağıdakilere benzemelidir:

Deneme1	Deneme2
Aradığınız ismi giriniz: Amelia Aradığınız isim bir kız bebek ismi olup 4. sirada!	Aradığınız ismi giriniz: Noah Aradığınız isim bir erkek bebek ismi olup 2. sirada!

Deneme3
Aradığınız ismi giriniz: Murat Aradığınız isim bulunamadi!

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class deneme2{
    public static void main(String[] args) throws FileNotFoundException {
        File f=new File("C:\\isimler.txt");
        Scanner input=new Scanner(f);
        Scanner input2=new Scanner(System.in);

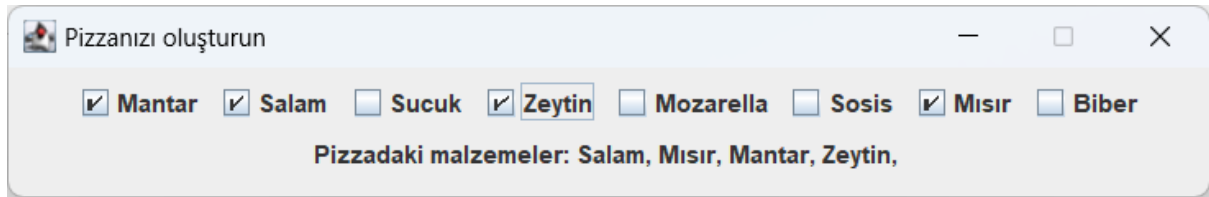
        System.out.print("Aradığınız ismi giriniz: ");
        String aranan=input2.next();

        String[] kelimeler;
        boolean bulundu=false;

        while(input.hasNext()) {
            kelimeler=input.nextLine().split("\t");
            if(kelimeler[1].equals(aranan)){
                System.out.println("Aradığınız isim bir erkek bebek ismi olup
"+kelimeler[0]+" sirada!");
                bulundu=true;
                break;
            }
            else if(kelimeler[3].equals(aranan)){
                System.out.println("Aradığınız isim bir kız bebek ismi olup
"+kelimeler[0]+" sirada!");
                bulundu=true;
                break;
            }
        }

        if(!bulundu)
            System.out.println("Aradığınız isim bulunamadi!");
    }
}
```

3) Bu soruda Swing kütüphanesini kullanarak bir pizza malzemesi seçimi ekranı yapmanız istenmektedir. Pencere 600x100 boyutunda olup FlowLayout düzenine sahip olmalıdır. Pencerede 8 adet malzemeyi simgeleyen 8 checkbox ve bir adet label bulunmalıdır. Kullanıcı checkboxları tıkladıkça bellekteki malzeme listesi güncellenerek (ekleme olabileceği gibi silme de olabilir) anında label'a aktarılmalıdır. Bellekteki malzeme listesini bir ArrayList ile tutabilirsiniz. Aşağıda, sırasıyla Salam, Mısır, Mantar ve Zeytin seçildiğindeki sonuç görülmektedir.



```
import java.awt.FlowLayout;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.util.ArrayList;
import javax.swing.*.*;

class Pencere extends JFrame implements ItemListener{
    JLabel jlbl=new JLabel("");
    String[] malzemeler= {"Mantar", "Salam", "Sucuk", "Zeytin", "Mozarella", "Sosis", "Mısır",
"Biber"};
    JCheckBox[] jcbler=new JCheckBox[malzemeler.length];
    ArrayList<String> pizza=new ArrayList<String>();

    Pencere(){
        super("Pizzanızı oluşturun");
        int en=600;
        int boy=100;
        this.setSize(en, boy);
        this.setResizable(false);
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
        this.setLayout(new FlowLayout());

        for(int i=0;i<jcbler.length;i++){
            jcbler[i]=new JCheckBox(malzemeler[i]);
            jcbler[i].addItemListener(this);
            this.add(jcbler[i]);
        }
        this.add(jlbl);
        this.setVisible(true);
    }

    public void itemStateChanged(ItemEvent e) {
        JCheckBox tiklanan=(JCheckBox)e.getSource();
        if(tiklanan.isSelected())
            pizza.add(tiklanan.getText());
        else
            pizza.remove(tiklanan.getText());

        String t="Pizzadaki malzemeler: ";
        for(String s:pizza) {
            t+=s+" ";
        }
        jlbl.setText(t);
    }
}

public class deneme2{
    public static void main(String[] args) {
        new Pencere();
    }
}
```

4) Aşağıdaki kavramları anlaşılır bir şekilde açıklayınız. Gerekli görürseniz kısa kod örnekleri veriniz.

a) Aggregation / Composition (Toplama/Bileşim):

İki sınıf arasında varoluşsal bağımlılık söz konusu değilse aggregation vardır. Örneğin, Araba sınıfını miras alan Motor sınıfı. Motorun imal edilmesi (var olması) için illa arabanın imal edilmesi şart değildir.

İki sınıf arasında varoluşsal bağımlılık söz konusu ise composition vardır. Örneğin, İnsan sınıfını miras alan Bacak sınıfı. İnsan olmazsa Bacak var olamaz.

b) Ragged Array (Düzensiz dizi):

Satırlarındaki sütun sayısı değişken olan, genellikle iki boyutlu olan dizilerdir. Örneğin,

```
int[][] dizi = { {1,2,3}, {4,5}, {6} };
```

c) public, private, protected düzenleyicileri:

Bkz: Kitaptaki Table 11.2

5) "liste" isimli bir ArrayList'te sayısı bilinmeyen miktarda Kus ve Tavsan nesnesi bulunduğunu varsayalım. Buna göre "liste"yi parametre olarak alıp içindeki Kus ve Tavsan nesnelerinin sayısını sayan ve konsola yazdıran "saydir" metodunun gövdesini tamamlayınız. Konsol çıktısı şu formatta olmalıdır:

```
>> Listede 14 tane Kus, 86 tane Tavsan var.
```

```
class Kus extends Object{ }
class Tavsan extends Object { }

class Test{
    public static void saydir(ArrayList<Object> liste){
        int k = 0, t = 0;
        for(int i = 0; i < liste.size(); i++) {
            Object nesne = liste.get(i);
            if(nesne instanceof Kus)
                k++;
            else
                t++;
        }
        System.out.println("Listede "+k+" tane Kus, "+t+" tane
Tavsan var.");
    }
}
```