

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BLM0121 – Nesneye Yönelik Programlama
Yılıçi Sınavı

Ad&Soyad	: CEVAP ANAHTARI
Öğrenci Numarası	:

Akademik yıl : 2023-2024
Dönem : Bahar
Tarih : 19 Nisan 2024 – 17:00
Sınav süresi : 90 dakika
Öğr. görevlisi : Doç. Dr. Ergün GÜMÜŞ

Soru	1	2	3	4	5	Toplam
Puan	25	15	20	20	20	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) Aşağıdaki Java kodunda bulunan hatalı satırları tespit ederek ilgili satırların yanına, olması gereken doğru halini yazınız.

```
01. private class Elbise {    //private sınıf olmaz!
02.     private String[] renkler;
03.     private double uzunluk;
04.     private int beden;
05.     private static int sıra = 0;
06.     private static int standart_elbise_sayaci = 0;
07.
08.     Elbise(){
09.         standart_elbise_sayaci++; //ilk satırda olamaz!
10.         this(new String[]{"Beyaz", "Turkuaz"}, 150.5, 38); //{}
11.     }
12.
13.     Elbise(String[] renkler, double uzunluk, int beden){
14.         this.renkler = renkler;
15.         this.uzunluk = uzunluk;
16.         this.beden = beden;
17.         elbiseDik();
18.     }
19.
20.     private void elbiseDik(){
21.         System.out.println(++this.sıra + ". elbise dikiliyor...");
22.
23.         for(renk: renkler)    //String renk:
24.             System.out.print(renk + ", ");
25.
26.         System.out.print("renklerinde; " +
27.                         uzunluk + " boyunda; " +
28.                         beden + " beden elbise dikildi.\n");
29.     }
30.
31.     public static void main(String[] args){
32.         new Elbise();
33.         new Elbise(new String[]{"Mavi", "Kırmızı"}, 145, 42); //{}
34.     }
35. }
```

Doğru çalışsaydı yukarıdaki kodun çıktısı:

1. elbise dikiliyor...
Beyaz, Turkuaz, renklerinde; 150.5 boyunda; 38 beden elbise dikildi.
2. elbise dikiliyor...
Mavi, Kırmızı, renklerinde; 145.0 boyunda; 42 beden elbise dikildi.

2) 1. soruda hatalarını düzelttiğiniz Java kodunun UML sınıf diyagramını çiziniz.

Elbise

```
-----  
- renkler: String[ ]  
- uzunluk: double  
- beden: int  
- sira: int  
- standart_elbise_sayaci: int  
-----
```

```
Elbise()  
Elbise(renkler: String[ ], uzunluk: double, beden: int)  
- elbiseDik(): void  
+ main(args: String[ ]): void
```

3) 1. sorudaki main metodunu aşağıdaki kurallara uyarak değiştiriniz:

- Bir elbise minimum 2, maksimum 4 adet renk içerebilir. Bu renklerin hepsi birbirinden farklıdır.
- Elbisede kullanılacak renkler şu kümeden seçilmelidir: Mavi, Kırmızı, Beyaz, Turkuaz, Sarı, Mor
- Bir elbise minimum 140.0 cm, maksimum 180.0 cm olabilir.
- Bir elbise minimum 36 beden, maksimum 46 beden olabilir.

Bu kurallara bağlı olarak; rasgele sayıdaki rasgele renklere sahip, rasgele uzunlukta ve rasgele bedende 100 adet elbise üreten bir kod bloğu yazınız.

```
String[] renk_listesi = {"Mavi", "Kırmızı", "Beyaz", "Turkuaz", "Sarı", "Mor"};  
java.util.Random r = new java.util.Random();  
  
int beden, rs, indis, k;  
double uzunluk;  
String[] rl;  
  
for(int i = 0; i < 100; i++) {  
    rs = r.nextInt(3) + 2; //2-4 arası bir sayı  
    rl = new String[rs];  
  
    for(int j = 0; j < rs; j++) {  
        indis = r.nextInt(renk_listesi.length);  
  
        for(k = 0; k < j; k++)  
            if(rl[k].equals(renk_listesi[indis]))  
                break;  
  
        if(k == j)  
            rl[j] = renk_listesi[indis];  
        else  
            j--;  
    }  
  
    uzunluk = r.nextDouble() % 40 + 140;  
    beden = r.nextInt(11) + 36;  
    new Elbise(rl, uzunluk, beden);  
}
```

4) Aşağıdaki ifadelerin durumuna göre Doğru veya Yanlış alanını "X" sembolüyle işaretleyiniz.

İfade	Doğru	Yanlış
Bir yavru (alt) sınıf birden fazla ata (üst) sınıftan aynı anda <u>doğrudan</u> miras alamaz.	X	
Bir yavru sınıf, mirasçısı olduğu ata sınıfın private türdeki değişkenlerini miras alır.	X	
Miras alınan ata sınıftaki private türdeki metotlara yavru sınıftan <u>doğrudan</u> erişilebilir.		X
Java dilinde bütün sınıflar "Object" sınıfının doğrudan veya dolaylı mirasçısıdır.	X	
String sınıfından yaratılan iki nesnenin içerikleri == operatörü ile karşılaştırılmaz.	X	
Ata sınıftaki bir metot, yavru sınıftaki bir metoda/değişkene erişmek için "super" ifadesini kullanabilir.		X
boolean türdeki yerel değişkenlerin varsayılan değeri "false" dur.		X
Bir konstrüktörde hem "super(...)" hem de "this(...)" çağrısı bulunamaz.	X	
Yavru sınıfta, ata sınıftakiyle aynı imzaya, geri dönüş parametresine ve erişim belirtecine sahip bir metot varsa "method overloading" söz konusudur.		X
Değiştirilemez (immutable) sınıflarda accessor (getter) metotlar bulunabilir.	X	

5) Java dilinde bir oyun yazalım. Oyunumuzda bir top ile gemileri vurup hasar veriyoruz. Aşağıdaki sınıfları inceleyiniz.

```

class Gemi{
    int saglik = 100;
    boolean batti = false;
}

class Top{
    int hasar = 20;
    void atisYap(Gemi hedef) {
        if(!hedef.batti){
            hedef.saglik -= hasar;
            if(hedef.saglik <= 0)
                hedef.batti = true;
        }
    }
}

```

Buna göre 10 gemiden oluşan bir filoya toplam 50 atış yapılmasını sağlayan bir oyun kodu parçası yazınız (ayrı bir Test sınıfındaki main() metodunun içine). Bunun için bir adet "top" nesnesi ve 10 elemanlı bir "gemiler" dizisi oluşturunuz. Topunuz 50 atışın her birinde filodaki rasgele bir gemiyi hedef almalıdır (batmış bir gemiyi de hedef alabilirsiniz, atışın boşa gitmesi de doğaldır). Atışlar tamamlandıktan sonra kodunuz aşağıdakine benzer bir savaş raporu yazdırmalıdır.

Rapor: 0. geminin sagligi 20 | 1. geminin sagligi 20 | 2. gemi batti! | 3. gemi batti! | 4. gemi batti! | 5. gemi batti! | 6. geminin sagligi 60 | 7. geminin sagligi 20 | 8. gemi batti! | 9. geminin sagligi 60 |

```

public class Test{
    public static void main(String[] args) {
        Top top = new Top();
        Gemi[] gemiler = new Gemi[10];

        for(int i = 0; i < gemiler.length; i++)
            gemiler[i] = new Gemi();

        for(int atis = 0; atis < 50; atis++)
            top.atisYap(gemiler[(int)(Math.random()*gemiler.length)]);

        System.out.print("Rapor: ");
        for(int i = 0; i < gemiler.length; i++) {
            if(gemiler[i].batti)
                System.out.print(i+". gemi batti! | ");
            else
                System.out.print(i+". geminin sagligi "+gemiler[i].saglik+" | ");
        }
    }
}

```