



BİL 141 Java Programlama Final Sınavı

Öğrenci Numarası:

Adı Soyadı:.....

Puanlama:	1-5: 3 puan	11-12: 10 puan	15-16: 5 puan (bonus)
	6-10: 5 puan	13-14: 20 puan	Toplam: 110 puan

1. Aşağıdakilerden hangisi `short` veri türünün değer aralığıdır?

- a) $-2^7 \dots 2^7 - 1$ b) $0 \dots 2^8 - 1$ **c) $-2^{15} \dots 2^{15} - 1$** d) $0 \dots 2^{16} - 1$

2. `float x = (float) 2.0;` ifadesindeki `(float)` ile yapılan işlem ne olarak adlandırılır?

- a) tür dönüşümü** b) veri dönüşümü c) sınıf dönüşümü d) değişken dönüşümü

3. Java derleyicisi, java kaynak kodunu,**JVM**..... isimli sanal makineye ait**byte**.... koda dönüştürür

4. **Farklı parçalardan (yani birden fazla java kaynak kod dosyalarından) oluşan programın bütün haline getirilmesine bağlama /linking** denir

5. Bir sınıfın değişkenleri ve metodlarının, yeni oluşturulacak bir sınıfa aktarılabilmesine **kalıtım/inheritance** denir

6. Aşağıdaki ifadelerin başına doğru ise D, yanlış ise Y yazınız.

- .**D.** Alt düzey bilgisayar dillerine **makine dili** ve **assembly dili** denir
- .**Y.** `String` **basit/temel** java veri türüdür
- .**Y.** Farklı parçalardan (yani birden fazla java kaynak kod dosyalarından) oluşan programın bütün haline getirilmesine **derleme/compiling** denir
- .**D.** Bir sınıf içinde **hiçbir** nesne kurucu tanımlanmaz ise, java derleyici/compiler, bu sınıf için varsayılan nesne kurucuyu otomatik olarak oluşturur
- .**Y.** Soyut/abstract sınıflardan alt sınıf türetmek mümkün **değildir**

7. Aşağıda verilen kod parçasının çıktısı nedir?

```
int a = 10, b = 5;
a *= b;
a += 20;
a = a % 6;
System.out.println("a:" + a);
```

a:4



8. Aşağıdaki **while** döngüsü ile hazırlanmış **program parçasını**, **for** döngüsü kullanarak yazınız.

```
int toplam = 0;
int i = 100;
while (i >= 1)
{
    toplam += i;
    i = i - 2;
}
System.out.println("toplam:"+toplam);
```

```
int toplam = 0;
for (int i = 100; i >= 1; i = i - 2;)
    toplam += i;
System.out.println("toplam:"+toplam);
```

9. Aşağıdaki programın sonucu ne olur?

```
int k = 2;
while (k < 5)
{
    System.out.println(k + " " + j);
    k += k % 2;
}
```

j değişkeni tanımlı olmadığı için derleyici/compiler hatası oluşur

10. Aşağıda java kodu verilen sınıfı UML tarzında gösteriniz.

```
public class Saat {
    private int saat;
    private int dakika;
    private int saniye;

    public Saat(int s, int d, int sn)
    {
        this.saat = s;
        this.dakika = d;
        this.saniye = sn;
    }
    public Saat()
    {
        this.saat = 0;
        this.dakika = 0;
        this.saniye = 0;
    }
    private void Sifirla()
    {
        this.saat = 0;
        this.dakika = 0;
        this.saniye = 0;
    }
}
```

Saat

- saat: int
- dakika: int
- saniye: int

+ Saat(int, int, int)
+ Saat()
- Sifirla(): void



11. Aşağıdaki çıktıyı oluşturan java kod parçasını, yazdırma komutu olarak sadece `System.out.print("*");` `System.out.print(" ");` ve `System.out.println();` kullanarak yazınız.

```
*****
*****
*****
*****
*****
****
***
**
*
```

```
for(int satir=0; satir<9; satir++)
{
    for(int bosluk=0; bosluk<satir; bosluk++)
        System.out.print(" ");
    for(int yildiz=9; yildiz>satir; yildiz--)
        System.out.print("*");
    System.out.println();
}
```

12. Bir metin dosyasına satırlar halinde kaydedilmiş Numara, Ad ve Soyad bilgilerini okuyup ekrana alt alta yazan bir program yazınız

```
import java.io.*;
public class Cevap12
{
    public static void main(String[] args)
    {
        // asagidaki minimum olması gereken kod

        BufferedReader girisAkimi =
            new BufferedReader(new FileReader("ornek.txt"));

        String numara = girisAkimi.readLine();
        String ad = girisAkimi.readLine();
        String soyad = girisAkimi.readLine();

        girisAkimi.close();

        System.out.println("Numara: " + numara);
        System.out.println("Ad : " + ad);
        System.out.println("Soyad : " + soyad);
    }
}
```



13. İki boyutlu dizileri kullanarak aşağıdaki metotları java'da kodlayınız.

- public double** [][] matrisYap(int m, int n)
Not: matrisin içini Math.Random()*100 kullanarak doldurabilirsiniz.
- public double** [][] matrisTopla(double[][] a, double [][] b)
- public double** [][] matrisTranspose(double [][] m)
- public double** matrisOrtalama(double [][] m)

```
public static double[][] matrisYap(int m, int n)
{
    double[][] matris = new double[m][n];
    for(int i=0; i<m; i++)
        for(int j=0; j<n; j++)
            matris[i][j] = Math.random()*100;

    return matris;
}

public static double[][] matrisTopla(double[][] a, double [][] b)
{
    // a ve b matrislerinin aynı boyutta olması onkosul olmak üzere
    int m = a.length;
    int n = a[0].length;
    double[][] matris = new double[m][n];
    for(int i=0; i<m; i++)
        for(int j=0; j<n; j++)
            matris[i][j] = a[i][j] + b[i][j];

    return matris;
}

public static double[][] matrisTranspose(double [][] m)
{
    int r = m.length;
    int c = m[0].length;
    double[][] matris = new double[c][r];
    for(int i=0; i<c; i++)
        for(int j=0; j<r; j++)
            matris[i][j] = m[j][i];

    return matris;
}

public static double matrisOrtalama(double [][] m)
{
    int r = m.length;
    int c = m[0].length;
    double toplam = 0.0;
    for(int i=0; i<r; i++)
        for(int j=0; j<c; j++)
            toplam += m[i][j];

    return toplam/(r*c);
}
```



14. Değişik yiyecek çeşitleri için bir sınıf tanımlanacaktır. Bu sınıf; yiyeceklerin isim, renk, ve ("agac", "saksi" ve "toprak" değerlerinden birini alabilecek) yetiştiği yer bilgisini tutacaktır. Ancak, bu sınıftan nesnelere oluşturulmayacak/oluşturulamayacak, sınıf daha sonra türetilen sınıflara temel teşkil edecektir. Ayrıca, sınıf değişkenlerine, sınıf dışından erişilemeyecek, ve tüm bu sınıf değişkenlerine değer atayan ve sınıf dışından erişilebilecek tek bir nesne kurucu (constructor) tanımlanacaktır. Yine sınıf değişkenleri sayısında parametre alan ve bunları sınıf değişkenlerine atayan bir set metodu, ve isim ile renk değişkenleri için parametre alan ve bunları bu iki sınıf değişkenine atayan diğer bir set metodu olmak üzere 2 set metodu tanımlanırken; sınıf değişkenlerinin her biri için ayrı ayrı get metodları tanımlanacaktır. Son olarak, Yiyecek sınıfı içerisinde YiyecekTuru isimli ve dışardan erişilebilen bir metod tanımlanacaktır. Bu metod parametre almayacak ve değer olarak "belirsiz" döndürecek.

- Yiyecek isimli bu sınıfı tanımlayan java kodunu yazınız.
- Yiyecek sınıfından Sebze ve Meyve sınıfları türetilecektir. Sebze ve Meyve sınıfları için sadece isim ve renk değerlerini parametre olarak alan nesne kurucular tanımlanacaktır. Sebze nesne kurucusu, nesnelere oluştururken yetiştiği yer başlangıç değerini "toprak" yapacak, Meyve nesne kurucusu ise "agac" yapacaktır. Yine bu sınıflar YiyecekTuru isimli metodu **yeniden yapılandıracağız**; Sebze sınıfı nesnelere üzerinden bu metod çağrıldığında "sebze" değerini, Meyve sınıfı nesnelere üzerinden bu metod çağrıldığında ise "meyve" değerini döndürecek. Sebze ve Meyve sınıflarını bu şekilde tanımlayan java kodunu yazınız.
- Bu sınıf ve metodları kullanarak **kırmızı lahana** ve **yeşil elma** için nesnelere oluşturan ve yetiştikleri yeri yazan java kod parçasını yazınız.
- Bir Yiyecek nesnesini parametre olarak alan ve örneğin bu nesne **kırmızı lahana** için oluşturulan nesne ise "lahana bir sebzedir", **yeşil elma** için oluşturulan nesne ise "elma bir meyvedir" , yazan **static** bir metod yazınız.

```
public abstract class Yiyecek { // abstract, soyut class olduğunu belirtiyor
    private String isim;
    private String renk;
    private String yer;

    public Yiyecek(String isim, String renk, String yer)
    {
        Set(isim, renk, yer);
    }

    public void Set(String i, String r, String y) // 1. set metodu
    {
        Set(i,r);
        yer = y;
    }
    public void Set(String i, String r) // 2. set metodu
    {
        isim = i;
        renk = r;
    }
    //get metodları
    public String GetIsim() { return isim; }
    public String GetRenk() { return renk; }
    public String GetYer() { return yer; }

    public String YiyecekTuru()
    {
        return "belirsiz";
    }
}
```



Cevap 14
Devam

b.

```
public class Sebze extends Yiyecek
{
    public Sebze(String i, String r)
    {
        super(i,r,"toprak");
    }

    public String YiyecekTuru()
    {
        return "sebze";
    }
}
public class Meyve extends Yiyecek
{
    public Meyve(String i, String r)
    {
        super(i,r,"agac");
    }

    public String YiyecekTuru()
    {
        return "meyve";
    }
}
```

c.

```
Sebze s = new Sebze("lahana","kirmizi");
String isim = s.GetIsim();
String renk = s.GetRenk();
String yer = s.GetYer();
System.out.println(renk + " " + isim + "\'nın yetistigi yer: " + yer);

Meyve m = new Meyve("elma","yesil");
isim = m.GetIsim();
renk = m.GetRenk();
yer = m.GetYer();
System.out.println(renk + " " + isim + "\'nın yetistigi yer: " + yer);
```

d.

```
public static void YiyecekYazdir(Yiyecek y)
{
    System.out.println(y.GetIsim() + " bir " + y.YiyecekTuru() + "dir.");
}
```



15. Aşağıdaki a ve b değişkenleri için her zaman $-2^{31} < (a + b) < 2^{31}$ koşulu sağlanıyor ise; bu değişkenlerin değerlerini 3. bir değişken kullanmadan yer değiştiren java kod parçasını yazınız. Sizin kodunuzdan önce örneğin a=3, b=11 değerlerinde ise, sizin kodunuzdan sonra a=11, b=3 değerlerinde olmalı. **İpucu:** Yukarıdaki koşul, a+b değerinin her zaman **int** türü değişkenlerin alabileceği minimum ve maksimum değerler içerisinde kaldığı anlamına gelir.

```
int a,b;
...
... // bu noktada a=3 b=11 degerinde ise

a = a + b; // bu noktada a=14 olur
b = a - b; // bu noktada b=3 olur
a = a - b; // bu noktada a=11 olur

// boylece bu noktada a=11 b=3 olur
```

16. Metod tanımlarının yeniden yapılandırılması (overriding) ile metodlarının fazladan yüklenmesi (overloading) arasındaki farkı örnek kod ile açıklayınız.

14. sorunun cevabında `Yiyecek` sınıfı içinde tanımlanan `Set` isimli 2 metod (farklı sayıda ve/veya türde parametre kullanmaları nedeniyle) fazladan yükleme (yani overloading) için örnektir.

Yine aynı sorunun cevabında, `Yiyecek` sınıfında tanımlanan `YiyecekTuru` metodu, kalıtım yolu ile `Sebze` ve `Meyve` sınıflarına aktarılmasına rağmen; bu sınıflar bu metodu `Yiyecek` sınıfındaki (yani temel sınıftaki şekli ile - parametre sayısı ve tipleri aynı olmak üzere) yeniden tanımladıkları için `Sebze` ve `Meyve` sınıflarındaki `YiyecekTuru` metodları, yeniden yapılandırılma (yani overriding) örnekleridir.