EXCEPTIONLAR VE DOSYA İŞLEMLERİ

Java'da Exceptionlar (istisnalar) beklenmeyen durumları ifade ederler. Dosya bulunamama, giriş çıkış istisnaları ve null pointer istisnası gibi çeşitleri bulunmaktadır. İstisnalar, semantik olarak hata olmayan, ancak çalışma anındaki bazı eksiklikler ve aksilikler nedeniyle çalışması yarıda kalan durumlardır. Bir istisnanın fonksiyona bildirilmesi veya try catch blokları ile yakalanması gerekmektedir. Aşağıdaki kodda iki örneği de görmek mümkündür. Programın daha sorunsuz çalışması adına bu durumların try catch blokları ile ele alınması en doğru yaklaşım olacaktır.

```
public static void main(String[] args) throws Exception{
    try{
        Scanner in = new Scanner(System.in); //klavyeden okuma nesnesi
        int x = in.nextInt();
    }catch (Exception e) {
        System.out.println("Girilen sayı integer değil");
    }
}
```

Java'da dosya işlemler için file sınıfından yararlanılır. File sınıfına parametre olarak dosya uzantısı ve adı verilmelidir. Uzantısı verilmeyen dosyalar projenin dizininde aranır. Okuma işlemi için Scanner nesnesi üretilmelidir. Scanner nesnesi ile veri okuma işlemi aşağıda verilmiştir.

```
File dosya = new File("dosya.txt");
Scanner in = new Scanner(dosya);

while(in.hasNextLine()) {
    System.out.println(in.nextLine());
}
```

Yazma işlemleri için PrintWriter nesnesi üretilmelidir. PrintWriter nesnesi parametre olarak file nesnesi almaktadır. Yeni bir txt dosyası oluşturan ve 0 ile 9 arasındaki sayıları bu dosyaya satır satır yazdıran kod aşağıda verilmiştir. Burada pw nesnesinin işlem sonunda kapatılması verilerin düzgün bir şekilde kaydedilmesi açısından önemlidir.

```
File dosya = new File("dosya.txt");
dosya.createNewFile();
PrintWriter pw = new PrintWriter(dosya);

for (int i = 0; i < 10; i++) {
    pw.println(i);
}
pw.close();
```

Bir dosyanın silinmesi için file sınıfına ait delete fonksiyonu kullanılır. Dosya silinmeden önce var mı yok mu diye kontrol edilmesi programın sağlıklı çalışması adına önemlidir. Dosya silme kodu aşağıda verilmiştir.

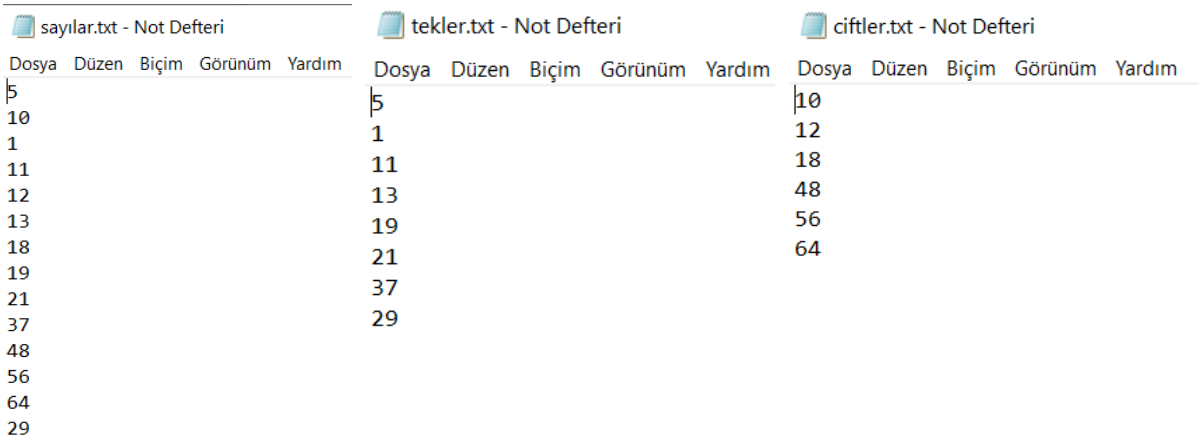
```
File dosya = new File("dosya.txt");

if(dosya.exists()) {
    dosya.delete();
    System.out.println("Dosya silindi");
}else
    System.out.println("Dosya zaten yok");
```

Bir dizin altında yer alan alt dizinleri ve dosyaları sıralamak için list komutundan yararlanılır. Bir dizin içindeki dosyaları görüntülemede önce dizinin varlığı ve dizin olduğu kontrolü gerçekleştirilmelidir. Aşağıda bir dizinde yer alan alt dizinler ve dosyaların isimlerini boyutları ile birlikte yazdıran kod parçacığı verilmiştir.

```
File dosya = new File("C:\\");

if(dosya.exists() && dosya.isDirectory()) {
    String dosyalar [] = dosya.list();

    for (int i = 0; i < dosyalar.length; i++) {
        File dosya2 = new File("C:\\"+dosyalar[i]);

        System.out.println(dosyalar[i] + " " + dosya2.length());
    }
}
```

Örnek:

Sayılar içerisinde tek ve çift sayılardan oluşan bir takım sayı yer almaktadır. Sizden bu sayılardan tekleri tekler.txt, çiftleri çiftler.txt dosyasına yazmanız istenmektedir. Gerekli Java kodunu yazınız. Örnek txt dosyaları aşağıda verilmiştir:



Çözüm:

```
public static void main(String[] args) throws FileNotFoundException{
    File f1 = new File("sayılar.txt");
    File ftek = new File("tekler.txt");
    File fcift = new File("ciftler.txt");

    Scanner in = new Scanner(f1);
    PrintWriter pwTek = new PrintWriter(ftek);
    PrintWriter pwCift = new PrintWriter(fcift);

    while(in.hasNext()) {
        int s = in.nextInt();
        if(s%2==0) {
            pwCift.println(s);
        }else {
            pwTek.println(s);
        }
    }

    pwCift.close();
    pwTek.close();

}
```

Örnek:

Vizeler.txt ve finallar.txt olmak üzere iki adet dosyada öğrencilere ait vize ve final notları sıralı olarak verilmiştir. Vizeler dosyasındaki ilk öğrencinin final notu finaller dosyasının ilk sırasında yer almaktadır. Ortalama notlar vizenin 0.4 ü ile finalin 0.6'sının toplamı ile hesaplanmaktadır. Sizden öğrencilerin vize ve final notlarını okuyarak sene sonu ortalamalarını bulmanız istenmektedir. Bulunan sonuçların ortalamalar.txt dosyasına aynı sırada yazılması isteniyor. Gerekli Java kodunu yazınız.

Çözüm:

```
public static void main(String[] args) throws Exception{
    File vizeler = new File("vizeler.txt");
    File finaller = new File("finaller.txt");
    File ortalamalar = new File("ortalamalar.txt");
    ortalamalar.createNewFile();

    Scanner voku = new Scanner(vizeler);
    Scanner foku = new Scanner(finaller);
    PrintWriter yaz = new PrintWriter(ortalamalar);

    while(voku.hasNext()) {
        int vize = voku.nextInt();
        int finaln = foku.nextInt();

        double ortalama = vize * 0.4 + finaln * 0.6;

        yaz.println(ortalama);
    }

    yaz.close();
}
```