

Görsel Programlama

DERS 06

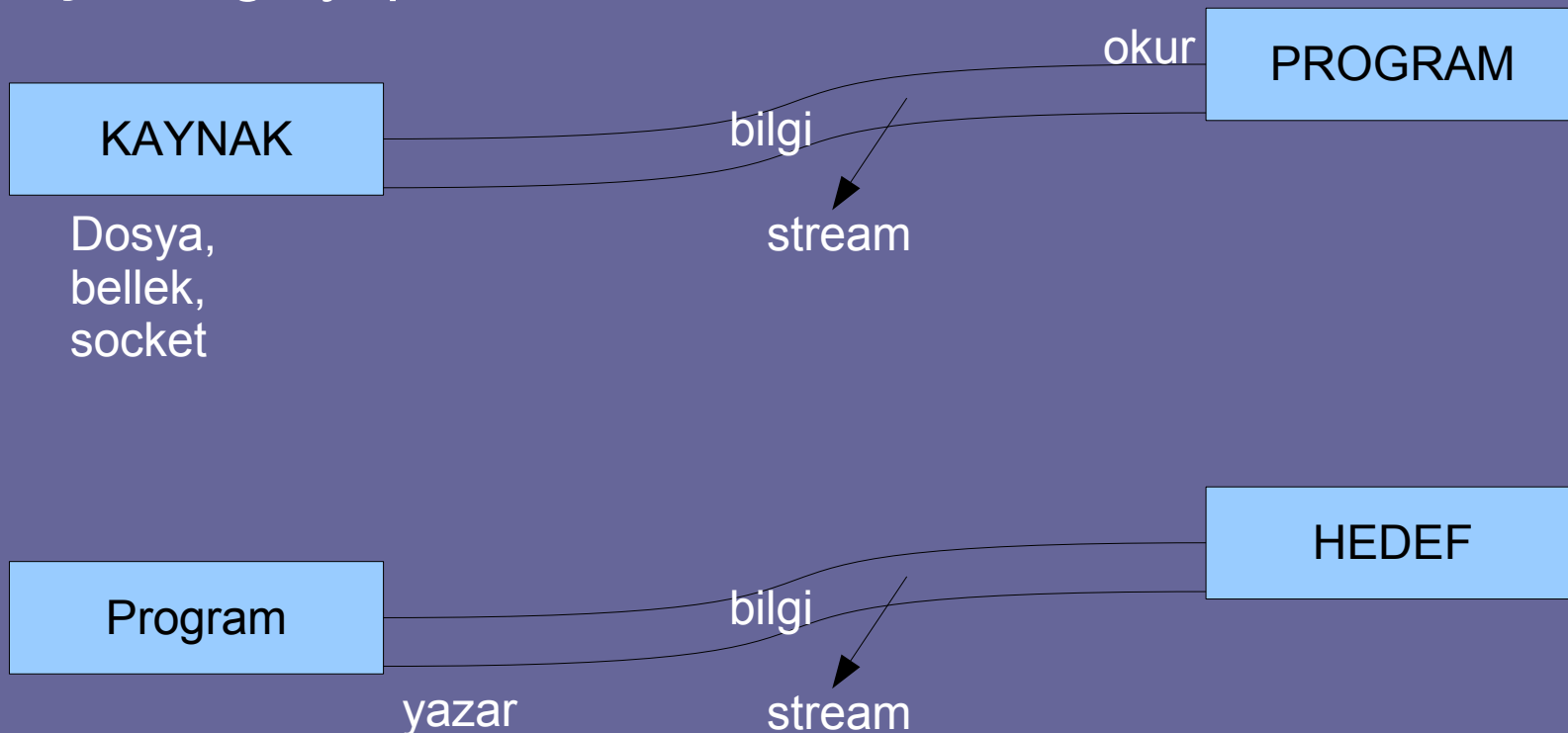
Girdi/Çıktı (I/O) Dosyalar

Programın çalışması sırasında oluşturulan değişkenlerde saklanan veriler program sonlandığında silinirler. Verilerin değerlerinin saklamak amacıyla dosyalar kullanılır.

Girdi/Çıktı ile ilgili sınıflar “**java.io**” paketinde bulunmaktadır.

Streams (Akımlar)

Stream okuma-yazma sırasında belirsiz sayıda byte ların oluşturduğu yapılardır.



Streams (Akımlar)

Veri okuma ve yazma algoritması şu şekildedir.

Okuma

1. Bir stream aç
2. Bilgi olduğu sürece
 - 2.1 Bilgi oku
3. Stream'i kapat

Yazma

1. Bir stream aç
2. Bilgi olduğu sürece
 - 2.1 Bilgiyi yaz
3. Stream'i kapat

Streams (Akımlar)

Bir java programı çalışmaya başladığında üç adet stream nesnesi hemen oluşturulur.

- System.in : Klavyeden girdi için
- System.out : Ekranı çıktı için
- System.err : Ekranı hata görüntülemek için

Streams (Akımlar)

```
StreamOrnek1.java X
package com.comu.gorsel_programlama.ders06;

public class StreamOrnek1 {
    public static void main(String[] args) {

        byte[] tampon = new byte[60];

        System.out.println("Yazı giriniz:");
        try {
            System.in.read(tampon);
        } catch (Exception e) {
            System.err.println("Hata:" + e.toString());
        }
        String s = new String(tampon, 0, 60);
        System.out.println("Yazı = " + s);

    }
}
```

```
package com.comu.gorsel_programlama.ders06;

import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class Kopyala {
    public static void main(String[] args) {
        File girdiDosyasi = new File("s.txt");
        File ciktiDosyasi = new File("c.txt");
        FileReader in=null;
        FileWriter out=null;
        try {
            in = new FileReader(girdiDosyasi);
            out = new FileWriter(ciktiDosyasi);
            int c;
            while ((c=in.read())!=-1){
                out.write(c);
            }
        } catch (Exception e) {
            System.err.println(e.getMessage());
        } finally{
            if (in!=null){
                try { in.close();} catch (IOException e)
                { e.printStackTrace();}
            }
            if (out!=null){
                try {out.close();} catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

```
package com.comu.gorsel_programlama.ders06;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

public class SatirKopyala {
    public static void main(String[] args) {
        BufferedReader girdi = null;
        PrintWriter cikti = null;
        try {
            girdi = new BufferedReader(new FileReader("g.txt"));
            cikti = new PrintWriter(new FileWriter("c.txt"));
            String satir ;
            while ((satir = girdi.readLine()) != null) {
                cikti.println(satir);
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                girdi.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
            cikti.close();
        }
    }
}
```


Çeşitli Dosya Sınıfları

InputStream : Temel okuma işlemlerinin yapıldığı streamdir. Okuma amaçlı tüm sınıflar bu sınıftan türetilmiştir. Abstract'tır.

OutputStream: Temel yazma işlemlerinin yapıldığı streamdir. Yazma amaçlı tüm sınıflar bu sınıftan türetilmiştir.

FileInputStream: Dosyadan okumayı gerçekleştirir.

FileOutputStream: Dosyaya yazmayı gerçekleştirir.

BufferedInputStream: Okuma işleminde tamponlama desteği verir ve bu işlemin performansını artırır.

BufferedOutputStream: Yazma işleminde tamponlama desteği verir.

Çeşitli Dosya Sınıfları

DataInputStream : Verilerin tiplere göre okunmasını sağlar.
(Örn:readInt(),readUTF(),readLine(),readDouble())

DataOutputStream: Verilerin tiplere göre yazılmasını sağlar.
(writeInt(), writeUTF(),writeLine(), writeDouble())

Buradaki sınıflar byte stream tabanlıdır yani verileri byte lar olarak okuyup yazarlar.

Çeşitli Dosya Sınıfları

Character tabanlı stream ler de bulunmaktadır. Bunlarda verileri int boyutlu olarak character stream leri olarak okuyup yazarlar. Character stream ler okunan karakterleri yerel karakter kümesine dönüşüm işlemini yaparlar.

Tüm karakter stream sınıfları **Reader** ve **Writer** sınıflarından türetilmiştir.

FileReader, **FileWriter** sınıfları **FileInputStream** ve **FileOutputStream** sınıflarının character stream karşılıklarıdır.

Çeşitli Dosya Sınıfları

`InputStreamReader` ve `OutputStreamReader` sınıfları byte streamlerden character streamlere geçiş için kullanılan köprü sınıflardır.

```
package com.comu.gorsel_programlama.ders06;

import java.io.DataOutputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class SiraliYazma {
    public static void main(String[] args) {
        String ad = "deneme";
        int a = 12;
        boolean b = true;
        double c = 12.99;
        FileOutputStream dosya=null;
        DataOutputStream ddosya=null;
        try {
            dosya = new FileOutputStream("dosya.txt");
            ddosya = new DataOutputStream(dosya);
            ddosya.writeUTF(ad);
            ddosya.writeInt(a);
            ddosya.writeBoolean(b);
            ddosya.writeDouble(c);
        } catch (Exception e) {
            e.printStackTrace();
        } finally{
            try {
                ddosya.close();
                dosya.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Sıralı Erişimli Dosyalar

```
package com.comu.gorsel_programlama.ders06;

import java.io.DataInputStream;

public class SiraliOkuma {
    public static void main(String[] args) {
        String ad;
        int a;
        boolean b;
        double c;
        FileInputStream dosya=null;
        DataInputStream ddosya=null;
        try {
            dosya = new FileInputStream("dosya.txt");
            ddosya = new DataInputStream(dosya);
            ad = ddosya.readUTF();
            a = ddosya.readInt();
            b = ddosya.readBoolean();
            c = ddosya.readDouble();
            if (b) {
                System.out.println("Ad="+ad+", a="+a+", c="+c);
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                ddosya.close();
                dosya.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Sıralı Erişimli Dosyalar

```
package com.comu.gorsel_programlama.ders06;

import java.io.IOException;
import java.io.RandomAccessFile;

public class DogrudanErisimOrnek {
    public static void main(String[] args) {
        RandomAccessFile dosya = null;
        try {
            dosya = new RandomAccessFile("dosya.txt", "rw");
            dosya.writeBoolean(true);
            dosya.writeInt(123456);
            dosya.writeChar('a');
            dosya.writeChars("yücel");
            dosya.seek(0); // 1. byte a konumlanır.
            System.out.println(dosya.readBoolean());
            dosya.seek(1);
            System.out.println(dosya.readInt());
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                dosya.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Doğrudan Erişimli Dosyalar

Örnekler

```
CreateNewFile.java x
package com.comu.gorsel_programlama.ders06;

import java.io.File;

public class CreateNewFile {
    public static void main(String[] args) {
        try {
            File dosya = new File("dosyaadi.txt");

            if (dosya.createNewFile()){
                System.out.println("başarıyla oluşturuldu");
            }else{
                System.out.println("dosya var");
            }
        } catch (Exception e) {
            // TODO: handle exception
        }
    }
}
```


Örnekler

```
DeleteFile.java x
package com.comu.gorsel_programlama.ders06;

import java.io.File;

public class DeleteFile {
    public static void main(String[] args) {
        File dosya=null;
        try {
            dosya = new File("dosyaadi.txt");
            if (dosya.delete()){
                System.out.println("başarıyla silindi");
            }else{
                System.out.println("silinemedi");
            }
        } catch (Exception e) {
            // TODO: handle exception
        }
    }
}
```

Örnekler

```
YenidenAdlandirma.java x
package com.comu.gorsel_programlama.ders06;

import java.io.File;

public class YenidenAdlandirma {
    public static void main(String[] args) {
        File eski = null;
        File yeni = null;
        try {
            eski = new File("dosya_yada_klasoradi");
            yeni = new File("yeni_dosya_yada_klasoradi");

            if (eski.renameTo(yeni)) {
                System.out.println("başarılı");
            } else {
                System.out.println("başarısız");
            }
        } catch (Exception e) {
            // TODO: handle exception
        }
    }
}
```

Örnekler

```
KlasorOlusturma.java X
package com.comu.gorsel_programlama.ders06;

import java.io.File;

public class KlasorOlusturma {
    public static void main(String[] args) {
        File klasor = null;
        try {
            klasor = new File("klasor_adi");
            if (klasor.mkdir()) {
                System.out.println("başarıyla oluşturuldu");
            } else {
                System.out.println("başarısız");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
package com.comu.gorsel_programlama.ders06;

import java.io.IOException;
import java.io.RandomAccessFile;

public class DogrudanErisimOrnek {
    public static void main(String[] args) {
        RandomAccessFile dosya = null;
        try {
            dosya = new RandomAccessFile("dosya.txt", "rw");
            dosya.writeBoolean(true);
            dosya.writeInt(123456);
            dosya.writeChar('a');
            dosya.writeChars("yücel");
            dosya.seek(0); // 1. byte a konumlanır.
            System.out.println(dosya.readBoolean());
            dosya.seek(1);
            System.out.println(dosya.readInt());
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                dosya.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Doğrudan Erişimli Dosyalar