



INPUT & OUTPUT

ThS. Nguyễn Nghiệm
0913.745.789 - NghiemN@fpt.edu.vn



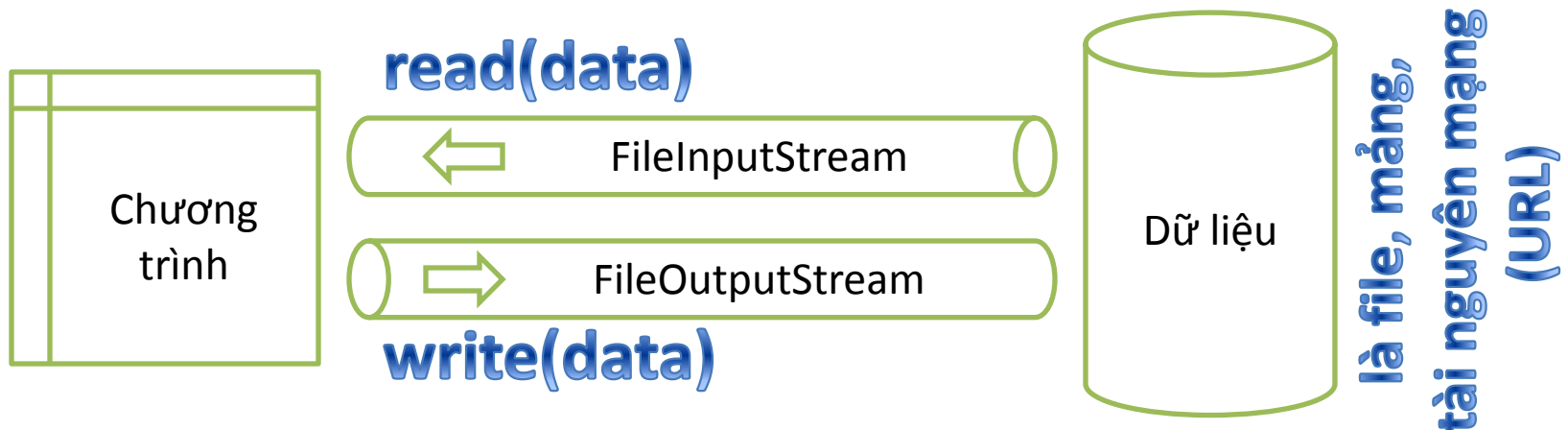
- InputStream & OutputStream
- FileInputStream & FileOutputStream
- ByteArrayInputStream & ByteArrayOutputStream
- Download Internet Resource
- ObjectInputStream & ObjectOutputStream
- File Manager





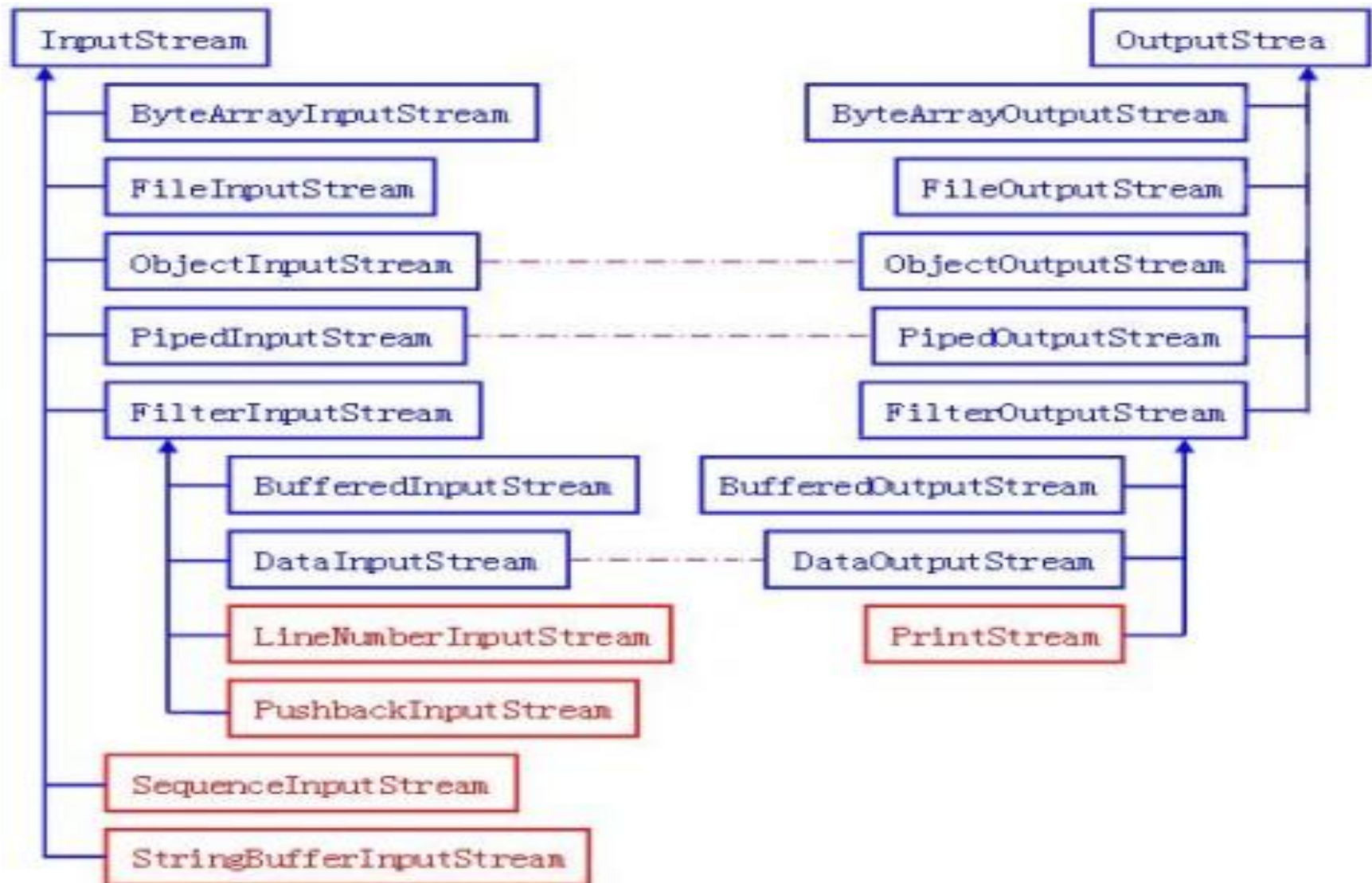
INPUTSTREAM & OUTPUTSTREAM

- **InputStream** là ống dẫn dữ liệu để chương trình có thể đọc dữ liệu từ các nguồn.
- **OutputStream** là ống dẫn dữ liệu để chương trình gửi dữ liệu đến đích.
- Dữ liệu có thể là file, mảng, tài nguyên từ xa, trao đổi với một chương trình khác...





PHÂN CẤP KẾ THỪA IO





INPUTSTREAM & OUTPUTSTREAM API

InputStream API

Phương thức	Mô tả
<code>int read(byte[] bytes)</code>	Đọc các byte tính từ vị trí hiện tại vào mảng @bytes. Trả về là số lượng byte đọc được.
<code>int available()</code>	Số lượng byte có thể đọc được trong luồng (kích thước luồng)
<code>void close ()</code>	Đóng luồng dữ liệu

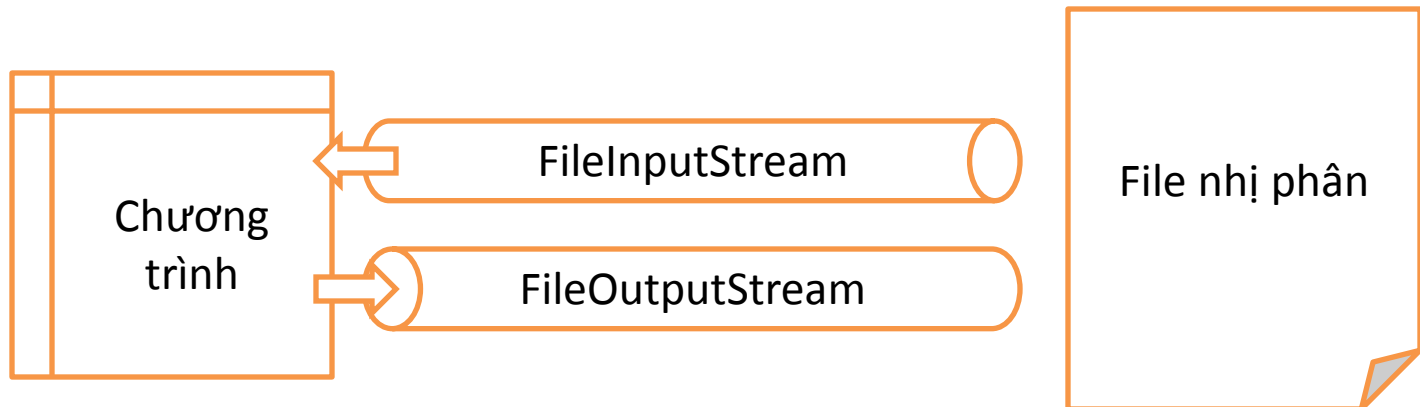
OutputStream API

Phương thức	Mô tả
<code>void write(byte[] bytes)</code>	Ghi mảng @bytes dữ liệu vào luồng
<code>void close()</code>	Đóng luồng dữ liệu



FILE STREAMS

- FileInputStream dùng để đọc dữ liệu file nhị phân
- FileOutputStream dùng để ghi dữ liệu ra file nhị phân





FILEINPUTSTREAM

```
String fileName = "c:/temp/photo.gif";
```

```
FileInputStream fis = new FileInputStream(fileName);
```

```
int n = fis.available();
```

```
byte[] data = new byte[n];
```

```
fis.read(data);
```

```
fis.close();
```

Mở luồng file

Số byte có thể đọc

Đọc dữ liệu từ file
vào mảng data

Đóng luồng file



FILEOUTPUTSTREAM

```
String fileName = "c:/temp/hello.txt";
```

```
FileOutputStream fos = new FileOutputStream(fileName);
```

```
byte[] data = "Chào thế giới Java".getBytes();
```

```
fos.write(data);
```

```
fos.close();
```

Mở luồng file

Dữ liệu nhị phân

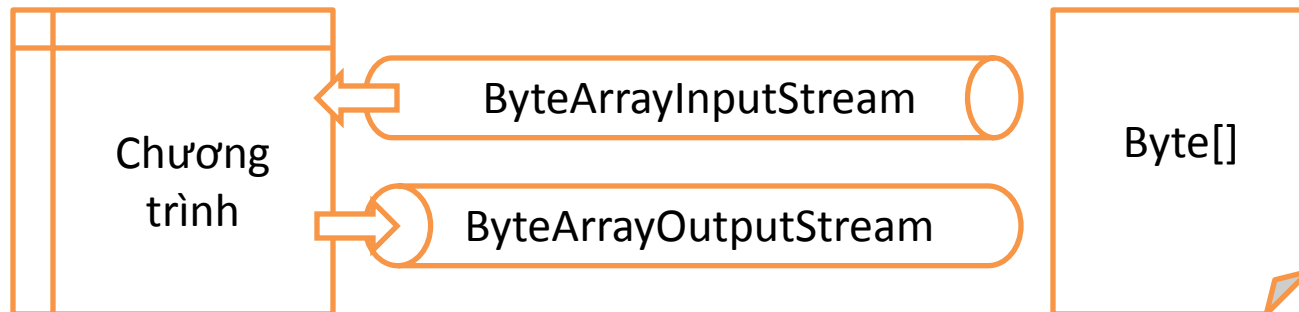
Ghi dữ liệu vào file

Đóng luồng file



BYTEARRAY STREAM

- ByteArrayInputStream và ByteArrayOutputStream được sử dụng để làm việc với mảng byte
- ByteArrayInputStream dùng để đọc dữ liệu nhị phân từ mảng byte[]
- ByteArrayOutputStream dùng để ghi dữ liệu nhị phân mảng byte[]





BYTEARRAYINPUTSTREAM

```
byte[] data = "Chào thế giới Java".getBytes();
```

```
ByteArrayInputStream bais = new ByteArrayInputStream(data);
```

```
int n = bais.available();
```

```
byte[] data2 = new byte[n];
```

```
bais.read(data2);
```

```
bais.close();
```

Cách đọc hoàn toàn tương tự như với file
chỉ khác là nguồn dữ liệu từ mảng byte[]
trong bộ nhớ chứ không phải file trên đĩa



ByteArrayOutputStream

```
ByteArrayOutputStream baos = new ByteArrayOutputStream();  
baos.write("Hello World".getBytes());  
baos.write("Java World".getBytes());
```

```
byte[] data = baos.toByteArray();
```

```
baos.close();
```

Ghi dữ liệu vào mảng

Lấy mảng dữ liệu



DOWNLOAD TỪ INTERNET

```
ByteArrayOutputStream buffer = new ByteArrayOutputStream();
```

Chứa dữ liệu download

```
URL url = new URL("http://www.vnexpress.net");
```

```
InputStream is = url.openStream();
```

Mở luồng dữ liệu đến tài nguyên

```
byte[] block = new byte[4*1024]; // 4KB
```

```
while(true){
```

```
    int n = is.read(block);
```

Chỉ nên đọc 1 lần tối đa 4KB

```
    if(n <= 0){
```

```
        break;
```

Dừng khi đã hết dữ liệu

```
    }
```

```
    buffer.write(block, 0, n);
```

Tích lũy dữ liệu đọc được

```
}
```

```
is.close();
```

```
byte[] data = buffer.toByteArray();
```

Lấy nội dung trang web

```
buffer.close();
```



OBJECT STREAMS

- **ObjectInputStream** và **ObjectOutputStream** là cặp luồng cho phép làm việc với dữ liệu đối tượng.
- Đây là cặp luồng phải ghép nối với **InputStream** hay **OutputStream** khác khi làm việc.
- **ObjectInputStream** là luồng vào dùng để đọc đối tượng
- **ObjectOutputStream** là luồng ra dùng để ghi đối tượng



OBJECT STREAMS API

- **ObjectInputStream API**

- ✱ Object **readObject()**: đọc một đối tượng

- **ObjectOutputStream API**

- ✱ **writeObject**(Object): ghi một đối tượng

- **Chú ý:**

- ✱ Chỉ đọc/ghi với các đối tượng có implements

- Serializable**

- ✱ Không đọc/ghi các trường được khai với **transient** trong đối tượng



LỚP SERIALIZABLE

```
public class SinhVien implements Serializable
{
    public String hoten;
    public double diem;
    public int tuoi;
    public boolean gioitinh;
    public transient String diachi;
}
```

Lớp sinh viên có thể sử dụng trong các thao tác đọc/ghi đối tượng

Trường địa chỉ không thể đọc/ghi cùng với đối tượng



VÍ DỤ OBJECTOUTPUTSTREAM

```
/*----- Mở file objects.dat -----*/  
FileOutputStream fos = new FileOutputStream("objects.dat");  
/*----- Ghép nối luồng cho phép ghi dữ liệu đối tượng -----*/  
ObjectOutputStream oos = new ObjectOutputStream(fos);  
/*----- Ghi thông tin người thứ nhất vào file nhansu.dat -----*/  
ArrayList o1 = new ArrayList();  
o1.add("ABC"); o1.add("123"); o1.add("XYZ");  
oos.writeObject(o1);    // ghi đối tượng ArrayList  
Date o2 = new Date();  
oos.writeObject(o2);    // ghi đối tượng Date  
/*----- Đóng các luồng dữ liệu -----*/  
oos.close();  
fos.close();
```



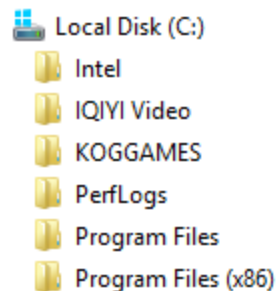

VÍ DỤ OBJECTINPUTSTREAM

```
/*----- Mở file objects.dat để đọc -----*/
FileInputStream fis = new FileInputStream("objects.dat");
/*----- Ghép nối luồng cho phép đọc dữ liệu đối tượng -----*/
ObjectInputStream ois = new ObjectInputStream(fis);
/*----- Đọc ArrayList -----*/
ArrayList list = (ArrayList)ois.readObject();
/*----- Đọc đối tượng thứ 2 -----*/
try{
    Date date = (Date)ois.readObject();
}
catch(ClassNotFoundException ex){
    System.out.println(ex.getMessage());
}
/*----- Đóng các luồng vào -----*/
fis.close();
ois.close();
```



FILE MANAGER

- File là lớp được sử dụng để quản lý tệp, thư mục, ổ đĩa.
- File chỉ quản lý thông tin cấu trúc (không phải dữ liệu)



NgonNguWeb	8/19/2015 7:58 PM	File folder	
slides	8/26/2015 9:31 PM	File folder	
1. Lập trình web với Java.docx	8/14/2015 12:17 AM	Microsoft Word D...	24 KB
2. Hướng dẫn thực hiện dự án java.docx	8/14/2015 12:18 AM	Microsoft Word D...	21 KB
Spring RequestMapping.docx	8/14/2015 8:26 AM	Microsoft Word D...	37 KB



FILE API

Phương thức	Mô tả
File.listRoots()	Lấy danh sách ổ đĩa
getName()	Lấy tên file/thư mục
getParent()	Lấy thư mục cha
getAbsolutePath()	Lấy đường dẫn tuyệt đối
length()	Lấy kích thước file/thư mục
lastModified()	Lấy thời gian cập nhật gần nhất
listFiles()	Lấy file và thư mục con
exists()	Kiểm tra sự tồn tại
isFile()	Kiểm tra file
isDirectory()	Kiểm tra thư mục



FILE API

Phương thức	Mô tả
isHidden()	Kiểm tra ẩn
canExecute()	Kiểm tra thực thi
canRead()	Kiểm tra cho phép đọc
canWrite()	Kiểm tra cho phép ghi
delete()	Xóa file/thư mục rỗng
mkdir()	Tạo 1 thư mục
makedirs()	Tạo nhiều thư mục lồng nhau
renameTo(File)	Di chuyển file
toURI()	Chuyển đổi sang URI
setReadOnly()	Thiết lập chỉ đọc



THÔNG TIN FILE

- Sử dụng lớp File để quản lý hệ thống tập tin và thư mục.
- Thông tin tập tin

```
File file = new File("c:/java/yahoo.com.html");
if (file.exists()) {
    System.out.println(">>Tên file:" + file.getName());
    System.out.println(">>Tên thư mục:" + file.getParent());
    System.out.println(">>Đường dẫn đầy đủ:" + file.getAbsolutePath());
    System.out.println(">>Kích thước file:" + file.length());
    System.out.println(">>Readonly:" + (!file.canWrite()));
    System.out.println(">>Hidden:" + file.isHidden());
    System.out.println(">>Last Modified:" + new Date(file.lastModified()));
}
```



FILE VÀ THƯ MỤC CON

```
File f = new File("c:\\java");
if (f.isDirectory()) {
    // Lấy các file và thư mục con trong thư mục temp
    File[] files = f.listFiles();
    // Xuất các thư mục con ra màn hình
    for (int i = 0; i < files.length; i++) {
        if (files[i].isDirectory()) {
            System.out.println("Thư mục: " + files[i].getName());
        }
    }
    // Xuất các file ra màn hình
    for (int i = 0; i < files.length; i++) {
        if (files[i].isFile()) {
            System.out.println("Tập tin: " + files[i].getName());
        }
    }
}
```



CÁC THAO TÁC KHÁC

● Liệt kê danh sách ổ đĩa

```
File[] disks = File.listRoots();  
for(int i=0;i<disks.length;i++){  
    System.out.println(disks[i].getName());  
}
```

● Tạo thư mục, đổi tên và xóa file

```
File f = new File("c:\\a\\b\\c\\d");  
// Tạo cây thư mục  
f.mkdirs();  
File abc = new File("c:\\abc.gif");  
if(abc.exists()){  
    // Di chuyển file @abc thành file "abc.gif" đặt trong @f  
    abc.renameTo(new File(f, "abc.gif"));  
}  
// xóa tập tin abc.gif  
f.delete();
```



TÓM TẮT

- InputStream & OutputStream
- FileInputStream & FileOutputStream
- ByteArrayInputStream & ByteArrayOutputStream
- Download Internet Resource
- ObjectInputStream & ObjectOutputStream
- File Manager

