Web-based programming with Java



# INPUT & OUTPUT

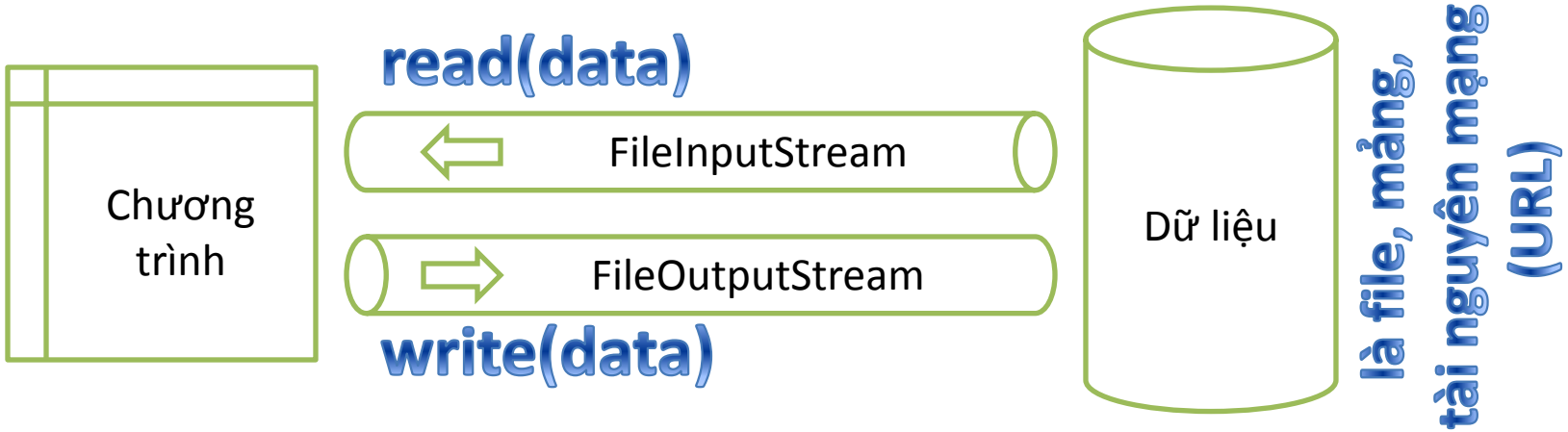
ThS. Nguyễn Nghiệm
0913.745.789 - NghiemN@fpt.edu.vn

# NỘI DUNG

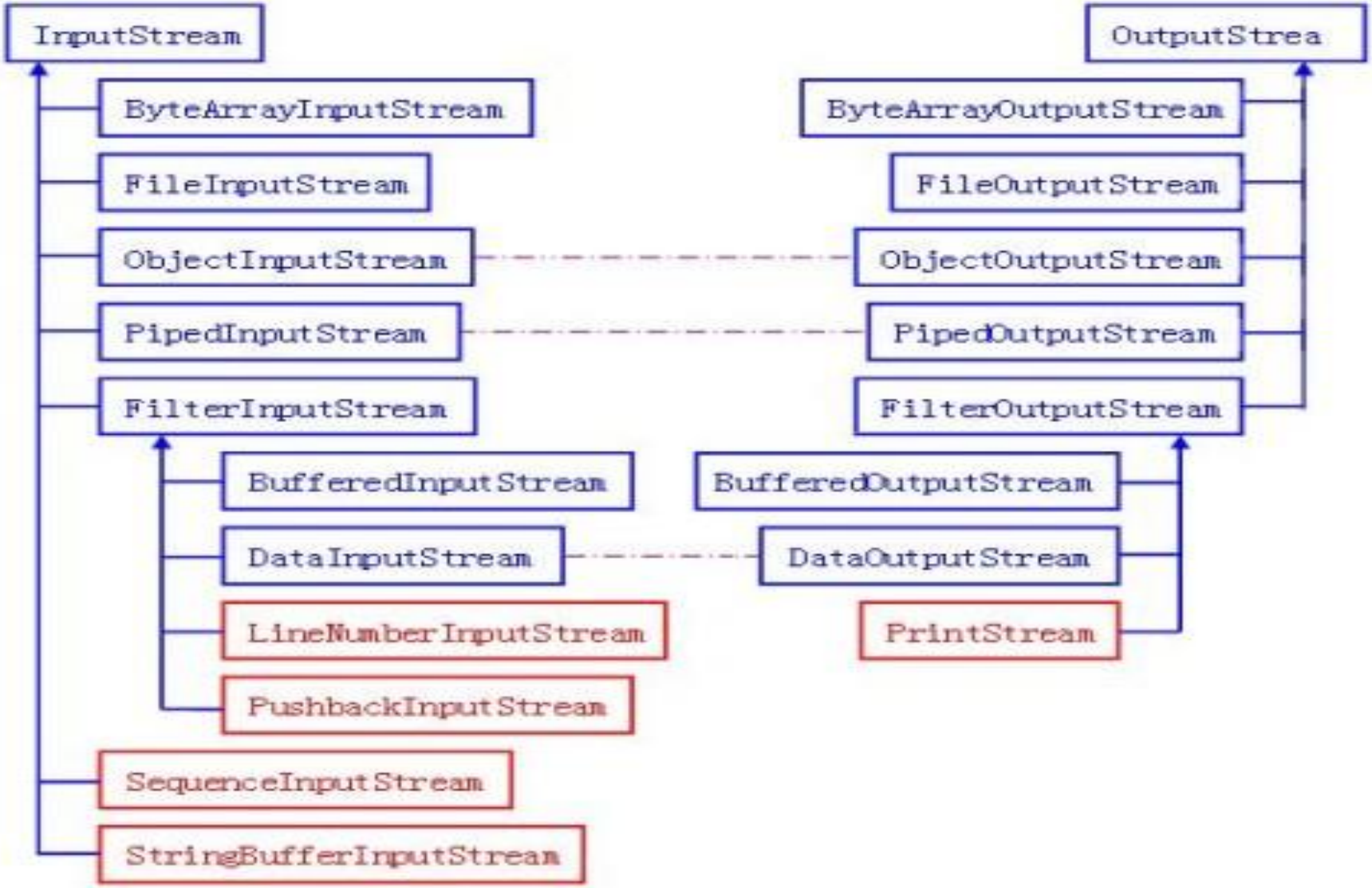
- InputStream & OutputStream
- FileInputStream & FileOutputStream
- ByteArrayInputStream & ByteArrayOutputStream
- Download Internet Resource
- ObjectInputStream & ObjectOutputStream
- File Manager

# 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

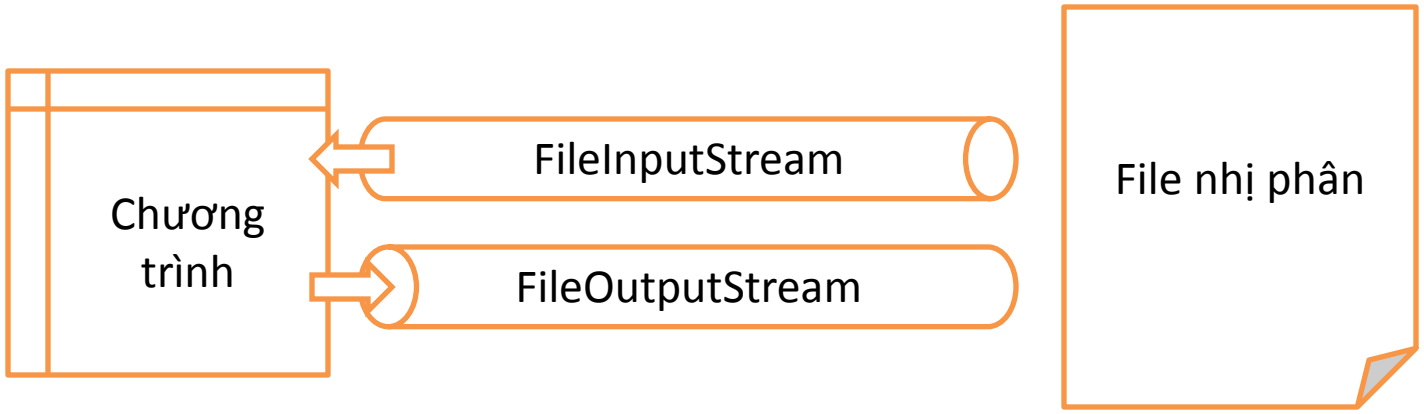
| Phương thức | Mô tả |
|---|---|
| int read(byte[] bytes) | Đọc các byte tính từ vị trí hiện tại vào mảng @bytes. Trả về là số lượng byte đọc được. |
| int available() | Số lượng byte có thể đọc được trong luồng (kích thước luồng) |
| void close () | Đóng luồng dữ liệu |

## OutputStream API

| Phương thức | Mô tả |
|---|---|
| void write(byte[] bytes) | Ghi mảng @bytes dữ liệu vào luồng |
| void close() | 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

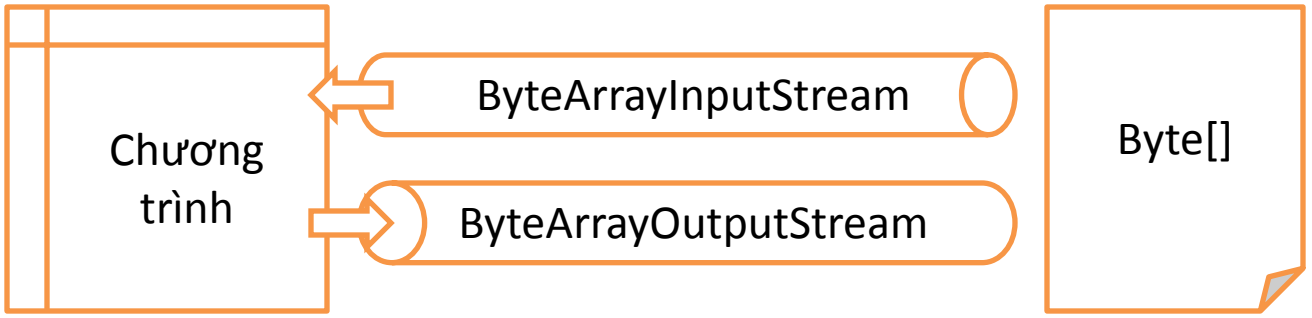
Dữ liệu nhị phân

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();

URL url = new URL("http://www.vnexpress.net");
InputStream is = url.openStream();
byte[] block = new byte[4*1024]; // 4KB
while(true){
   int n = is.read(block);
   if(n <= 0){
      break;
   }
   buffer.write(block, 0, n);
}
is.close();

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

Chứa dữ liệu download

Mở luồng dữ liệu đến tài nguyên

Chỉ nên đọc 1 lần tối đa 4KB

Dừng khi đã hết dữ liệu

Tích lũy dữ liệu đọc được

Lấy nội dung trang web

# OBJECT STREAMS

- ObjectInputStream và ObjectOutputStream là cập 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

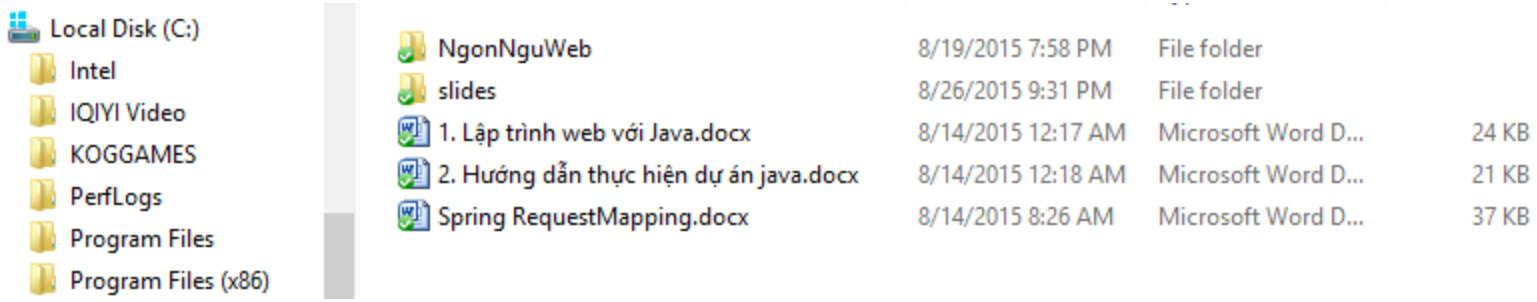
```
/*----- Mở file objects.dat -----*/
FileOutputStream fos = new FileOutputStream("objects.dat");
/*----- Ghép nối luồng cho phép ghi dữ liệu đối tượng -----*/
ObjectOutputStream oos = new ObjectOutputStream(fos);
/*----- Ghi thông tin người thứ nhất vào file nhansu.dat -----*/
ArrayList o1 = new ArrayList();
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)



# FILE API

| Phường thức | Mô tả |
|---|---|
| File.listRoots() | Lấy danh sách ở 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 |
| mkdirs() | 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 thông 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 thu 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