



# HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÀI GIẢNG MÔN

## Lập trình mạng

Giảng viên:

TS. Nguyễn Trọng Khánh

Điện thoại/E-mail:

khanhnt@ptit.edu.vn

Bộ môn:

CNPM- Khoa CNTT1

Học kỳ/Năm biên soạn: 23/08/2021

# Luồng vào ra với Java





# Nội dung



- ❖ Input Streams
- ❖ Output Streams
- ❖ Filter Streams
- ❖ Reader & Writer



# Luồng vào ra



## ❖ Luồng ?

- Chuỗi dữ liệu với chiều dài chưa xác định.
- **Luồng vào** di chuyển dữ liệu từ nguồn bên ngoài (keyboard, file, network client) → chương trình Java.
- **Luồng ra** di chuyển dữ liệu từ chương trình Java → đích bên ngoài (console, file, network server).

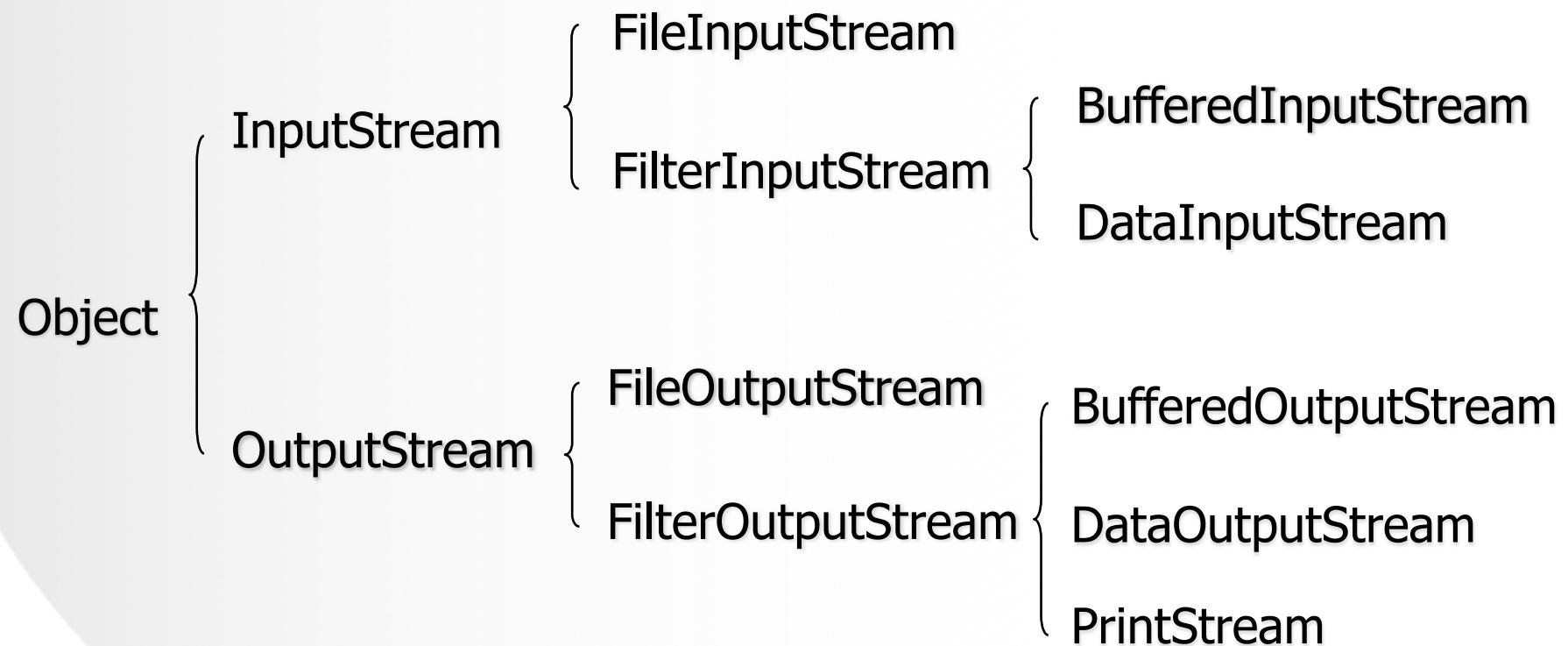
## ❖ Các luồng Java

- các byte hoặc ký tự dữ liệu rác.



# Các luồng byte

- ❖ Các lớp InputStream và OutputStream đọc và ghi byte





# InputStream (1)

- ❖ **java.io.InputStream** lớp trừu tượng cho tất cả luồng vào
- ❖ Phương thức để đọc các byte dữ liệu từ luồng vào (key board, files, network client)
  - public abstract int read()
  - public int read (byte[] buf)
  - public int read(byte[] buf, int offset, int length)



# InputStream (2)

Vào từ bàn phím:

```
try{
    InputStream input = new InputStream(System.in);
    while((input.read()) != -1) {
        //do something with data...
    }
    input.close();
} catch(IOException e){
    System.out.println(e);
}
```



# InputStream (3)

Vào từ file:

```
try{
    InputStream input = new FileInputStream("input.txt");
    while((input.read()) != -1) {
        //do something with data...
    }
    input.close();
} catch(IOException e) {
    System.out.println(e);
}
```



# InputStream (4)

Vào từ socket:

```
try{
    ServerSocket myServer = new ServerSocket(số cổng);
    Socket clientSocket = myServer.accept();
    InputStream input = new
    DataInputStream(clientSocket.getInputStream());

    while((input.read()) != -1) {
        //do something with data...
    }
    input.close();
} catch(IOException e){
    System.out.println(e);
}
```



# OutputStream (1)

- ❖ Lớp trừu tượng cho các luồng ra
- ❖ Phương thức quan trọng
  - public abstract void write()
  - public abstract void write(byte b)
  - public void write(byte b[])
  - public void write(byte b[], int offset, int length)
  - public void flush( ) throws IOException
  - public void close()



## Ra màn hình:

```
try{
    OutputStream output = new
    BufferedOutputStream(System.out);
    output.write(1111111);

    output.close();
} catch(IOException e) {
    System.out.println(e);
}
```



# OutputStream(2)



## Ra file:

```
try{
    OutputStream output = new
FileOutputStream("output.txt");
output.write(1111111);

    output.close();
} catch(IOException e) {
    System.out.println(e);
}
```



# OutputStream(3)



## Ra socket:

```
try{  
    ServerSocket myServer = new ServerSocket(155);  
    Socket clientSocket = myServer.accept();  
    OutputStream output = new  
    DataOutputStream(clientSocket.getOutputStream());  
  
    output.write(1111111);  
  
    output.close();  
} catch(IOException e){  
    System.out.println(e);  
}
```



# OutputStream(4)

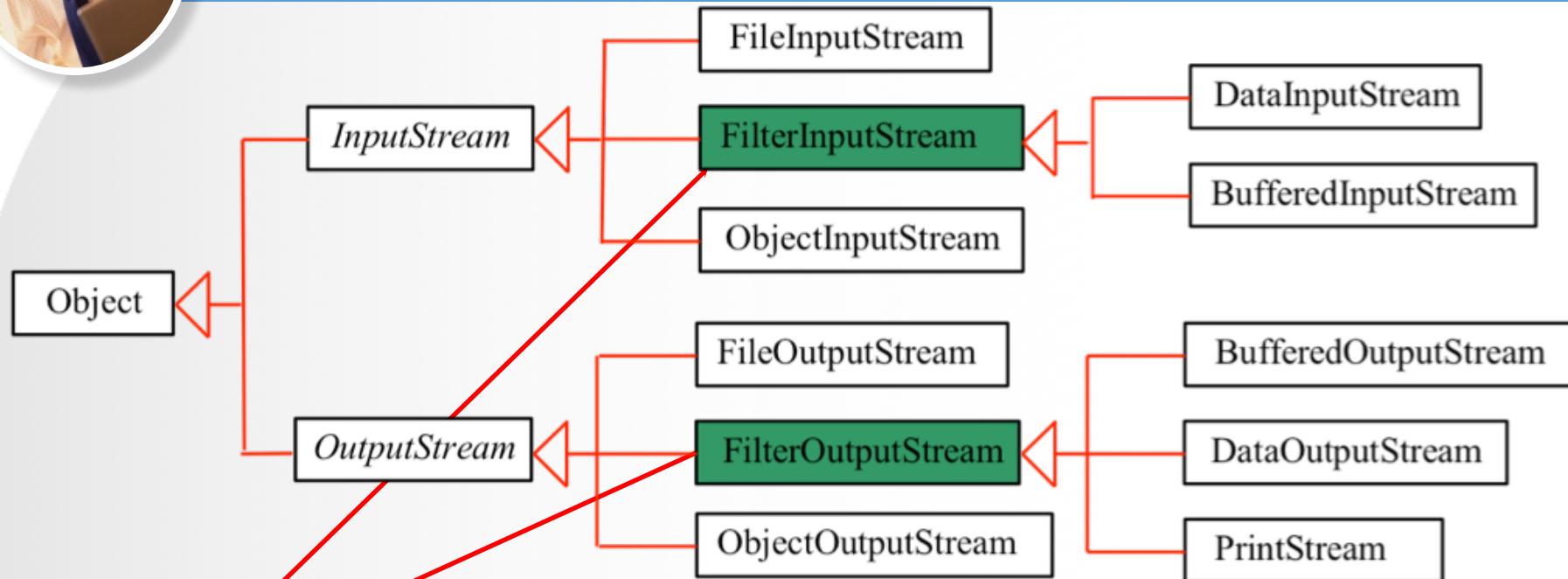


# Filter Stream

- ❖ `java.io.FilterInputStream` và `java.io.FilterOutputStream` là lớp con của `InputStream` and `OutputStream`
- ❖ Bản thân 2 lớp này ít được sử dụng hơn so với lớp con của chúng



# FilterInputStream/FilterOutputStream



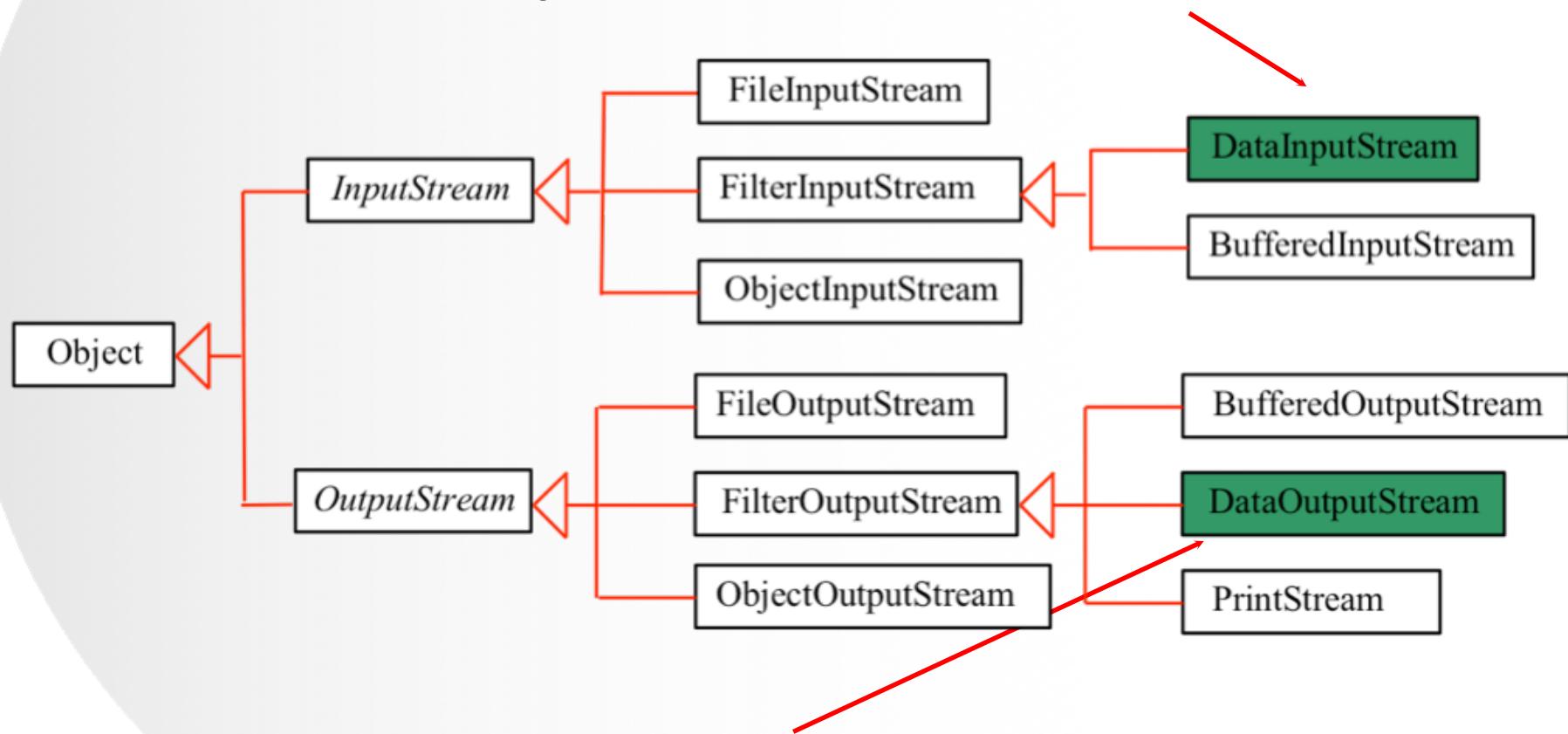
~~Filter streams~~: những luồng lọc byte

- Việc đọc file theo byte cơ bản chỉ cung cấp phương thức để đọc các bytes.
- Nếu muốn đọc các số nguyên, double, string .. , cần sử dụng các lớp lọc.
- FilterInputStream và FilterOutputStream là lớp cơ sở để lọc dữ liệu.
- Để xử lý các kiểu dữ liệu gốc kiểu số : DataInputStream và DataOutputStream.



# DataInputStream/DataOutputStream

DataInputStream đọc bytes từ luồng và chuyển thành kiểu dữ liệu gốc phù hợp hoặc string



DataOutputStream chuyển các giá trị kiểu dữ liệu gốc hoặc string thành các byte và xuất ra luồng.



# DataInputStream (1)

Vào từ bàn phím:

```
try{
    DataInputStream input = new
        DataInputStream(System.in);

    String in = input.readUTF();
    //do something with data...

    input.close();
} catch(IOException e){
    System.out.println(e);
}
```



# DataInputStream (2)

Vào từ file:

```
try{
    DataInputStream input = new
        DataInputStream(new FileInputStream("input.txt"));

    String in = input.readUTF();
    //do something with data...

    input.close();
} catch(IOException e) {
    System.out.println(e);
}
```



# DataInputStream (3)

Vào từ socket:

```
try{
    ServerSocket myServer = new ServerSocket(số cổng)
    Socket clientSocket = myServer.accept();
    DataInputStream input = new
        DataInputStream(clientSocket.getInputStream());

    String in = input.readUTF();
    //do something with data...

    input.close();
} catch(IOException e){
    System.out.println(e);
}
```



# DataOutputStream (1)

Ra màn hình:

```
try{
    DataOutputStream output = new
        DataOutputStream(System.out);

    output.writeUTF("some thing to write");

    output.close();
} catch(IOException e){
    System.out.println(e);
}
```



# DataOutputStream (2)

Ra file:

```
try{
    DataOutputStream output = new
        DataOutputStream(new FileOutputStream("output.txt"));

    output.writeUTF("some thing to write");

    output.close();
} catch(IOException e){
    System.out.println(e);
}
```



# DataOutputStream (3)

Ra socket:

```
try{
    ServerSocket myServer = new ServerSocket(155);
    Socket clientSocket = myServer.accept();
    DataOutputStream output = new
        DataOutputStream(clientSocket.getOutputStream());

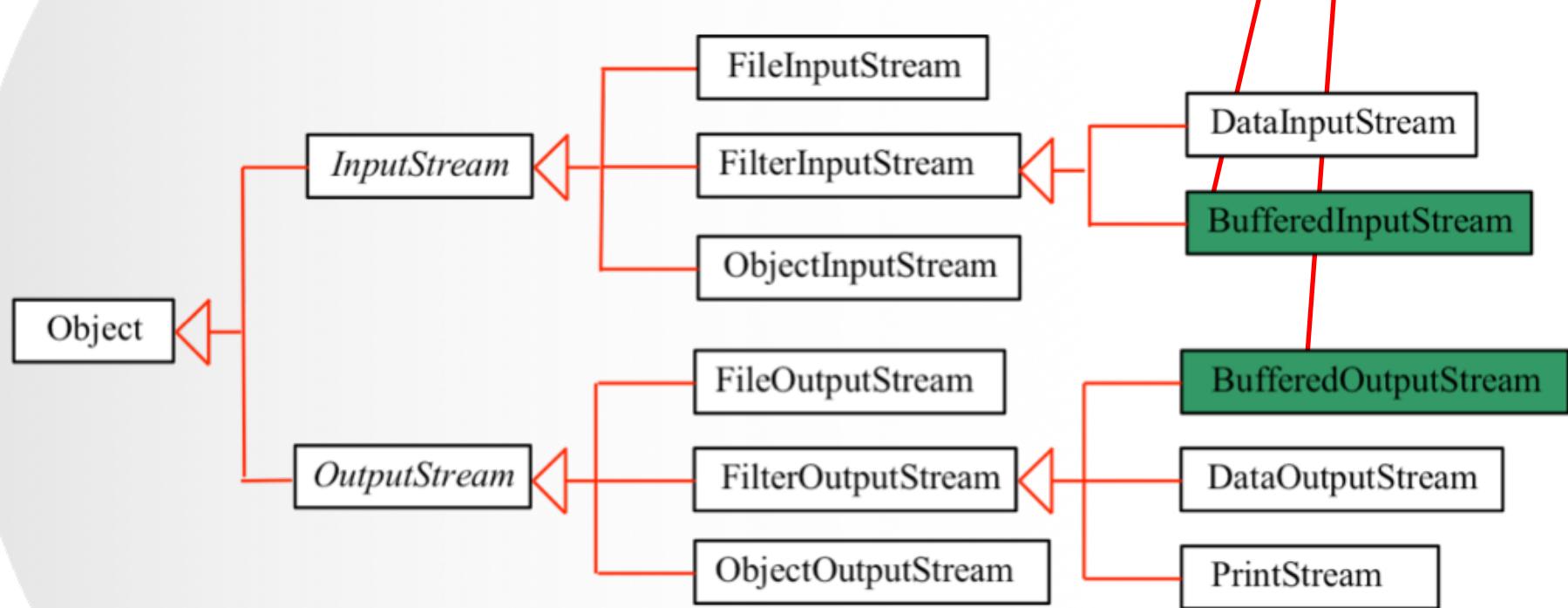
    output.writeUTF("some thing to write");

    output.close();
} catch(IOException e) {
    System.out.println(e);
}
```



# BufferedInputStream/BufferedOutputStream

Sử dụng bộ nhớ đệm để tăng tốc vào ra



- BufferedInputStream/BufferedOutputStream : không có các phương thức mới; đều thừa kế từ InputStream/OutputStream.



# BufferedInputStream (1)

Vào từ bàn phím:

```
try{
    BufferedInputStream input = new
        BufferedInputStream(System.in);

    byte[] in = new byte[1024];
    while((input.read(in)) != -1) {
        //do something with data...
    }
    input.close();
} catch(IOException e){
    System.out.println(e);
}
```



# BufferedInputStream (2)

Vào từ file:

```
try{
    BufferedInputStream input = new
        BufferedInputStream(new FileInputStream("input.txt"));

    byte[] in = new byte[1024];
    while((input.read(in)) != -1) {
        //do something with data...
    }
    input.close();
} catch(IOException e) {
    System.out.println(e);
}
```



# BufferedInputStream (3)

Vào từ socket:

```
try{
    ServerSocket myServer = new ServerSocket(số cổng);
    Socket clientSocket = myServer.accept();
    BufferedInputStream input = new
        BufferedInputStream(clientSocket.getInputStream());

    byte[] in = new byte[1024];
    while((input.read(in)) != -1) {
        //do something with data...
    }
    input.close();
} catch(IOException e){
    System.out.println(e);
}
```



# BufferedOutputStream (1)

Ra màn hình:

```
try{
    BufferedOutputStream output = new
        BufferedOutputStream(System.out);

    output.write(1111111);
    output.close();
} catch(IOException e){
    System.out.println(e);
}
```



# BufferedOutputStream (2)

Ra file:

```
try{
    BufferedOutputStream output = new
        BufferedOutputStream(new FileOutputStream("output.txt"));

    output.write(1111111);

    output.close();
} catch(IOException e){
    System.out.println(e);
}
```



# BufferedOutputStream (3)

Ra socket:

```
try{
    ServerSocket myServer = new ServerSocket(155);
    Socket clientSocket = myServer.accept();
    BufferedOutputStream output = new
        BufferedOutputStream(clientSocket.getOutputStream());

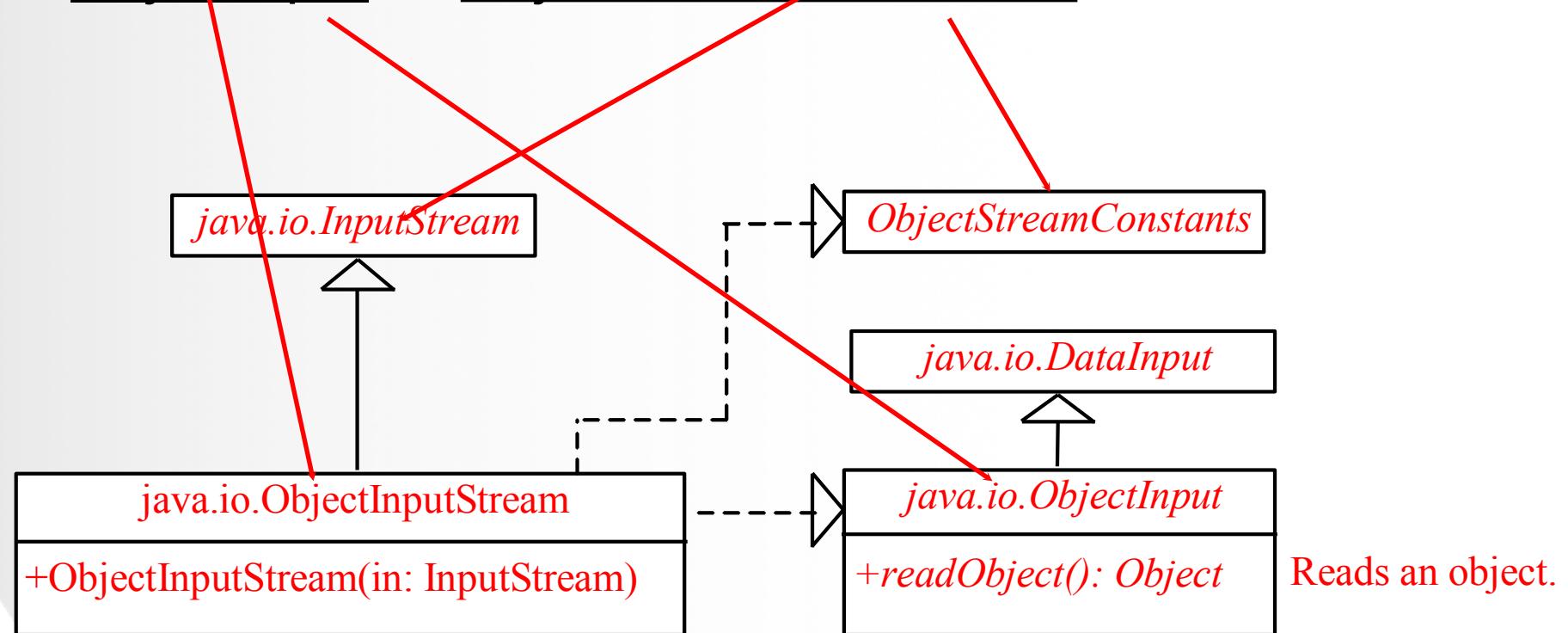
    output.write(1111111);

    output.close();
} catch(IOException e) {
    System.out.println(e);
}
```



# ObjectInputStream

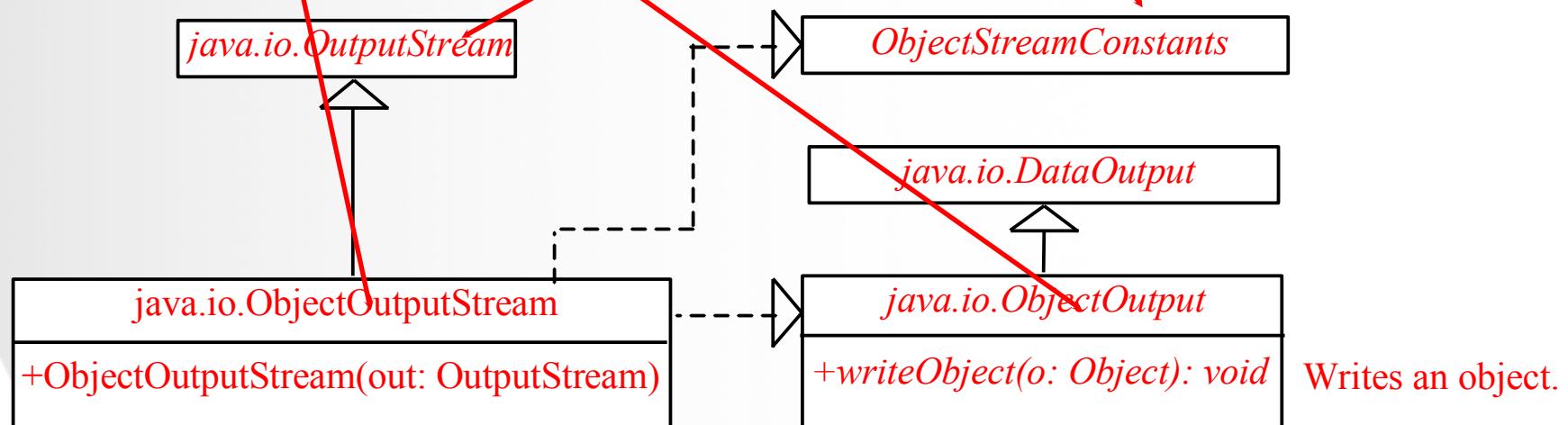
ObjectInputStream kế thừa InputStream và thực thi  
ObjectInput và ObjectStreamConstants.





# ObjectOutputStream

ObjectOutputStream kế thừa OutputStream và thực thi ObjectOutput và ObjectStreamConstants.





# Giao tiếp Serializable

Không phải tất cả các đối tượng đều có thể được ghi tới luồng ra.

- Những đối tượng có thể ghi được tới một luồng ra thì được gọi là tuần tự hoá được (*Serializable*). Một đối tượng *Serializable* là một thực thể của giao tiếp java.io.Serializable. -> để *Serializable* lớp cần thực thi giao tiếp Serializable.

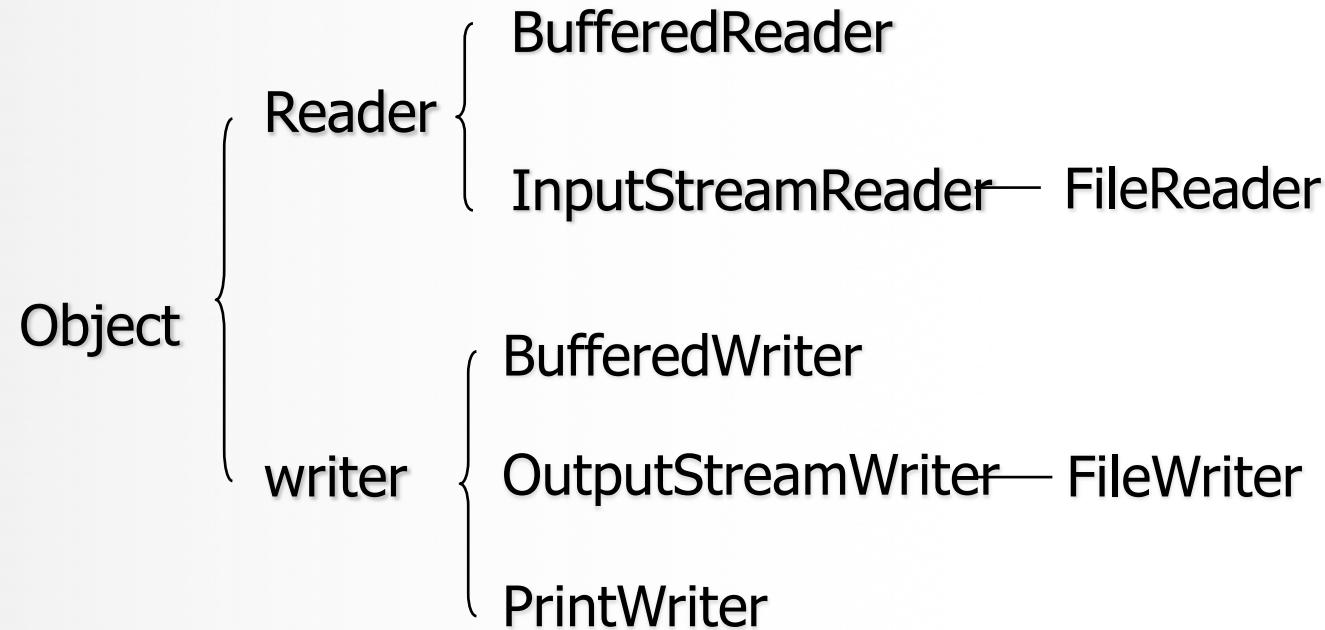
Giao tiếp Serializable là giao tiếp đánh dấu.

- Không có phương thức -> không cần viết thêm code để thực thi giao tiếp.



# Luồng ký tự

- ❖ Các Reader và Writer đọc và ghi dữ liệu kiểu ký tự.





# BufferedReader

Vào từ bàn phím:

```
BufferedReader input = new BufferedReader(new  
InputStreamReader(System.in));
```

Vào từ file:

```
FileReader reder = new FileReader("input.txt");  
BufferedReader input = new  
BufferedReader(new FileInputStream("input.txt"));
```

Vào từ socket:

```
ServerSocket myServer = new ServerSocket(số cổng);  
Socket clientSocket = myServer.accept();  
BufferedReader input = new  
BufferedReader(clientSocket.getInputStream());
```

```
int in = input.read();  
//do something with35data...
```



# BufferedWriter

Ra màn hình:

```
BufferedWriter output = new BufferedWriter(new  
OutputStreamReader(System.out));
```

Ra file:

```
FileWriter fileWriter = new FileWriter("output.txt")  
BufferedWriter output = new  
BufferedWriter(new FileOutputStream("output.txt"));
```

Ra socket:

```
ServerSocket myServer = new ServerSocket(155);  
Socket clientSocket = myServer.accept();  
BufferedWriter output = new  
BufferedWriter(clientSocket.getOutputStream());  
  
output.write(1111111);
```



# InputStreamReader

Vào từ bàn phím:

```
InputStreamReader input = new  
InputStreamReader(System.in);
```

Vào từ file:

```
InputStreamReader input = new  
InputStreamReader(new FileInputStream("input.txt"));
```

Vào từ socket:

```
ServerSocket myServer = new ServerSocket(số cổng);  
Socket clientSocket = myServer.accept();  
InputStreamReader input = new  
InputStreamReader(clientSocket.getInputStream());
```

```
String in = input.readUTF();  
//do something with data...
```



# OutputStreamWriter

Ra màn hình:

```
OutputStreamWriter output = new OutputStreamWriter(System.out);
```

Ra file:

```
OutputStreamWriter output = new  
OutputStreamWriter(new FileOutputStream("output.txt"));
```

Ra socket:

```
ServerSocket myServer = new ServerSocket(155);  
Socket clientSocket = myServer.accept();  
OutputStreamWriter output = new  
OutputStreamWriter(clientSocket.getOutputStream());  
  
output.writeUTF("some thing to write");
```



## Bài tập tại lớp

❖ Sử dụng luồng byte vào ra với đối tượng, viết ứng dụng sau:

- Khởi tạo dữ liệu danh sách sinh viên SV(maSV - String, hoten - String, nhom - Int) từ file đã được lưu theo danh sách ArrayList<SV> đối tượng kiểu sinh viên từ trước
- Cho phép người dùng thêm mới, cập nhập dữ liệu từ bàn phím, hiển thị ra màn hình và lưu xuống file



❖ N9, 15/8