

Like



Share



Subscribe

Anh Nguyen Ngoc

ProIT4All



Giới thiệu

# Luồng nhập/xuất byte



Fan page: <http://facebook.com/Proit4All>

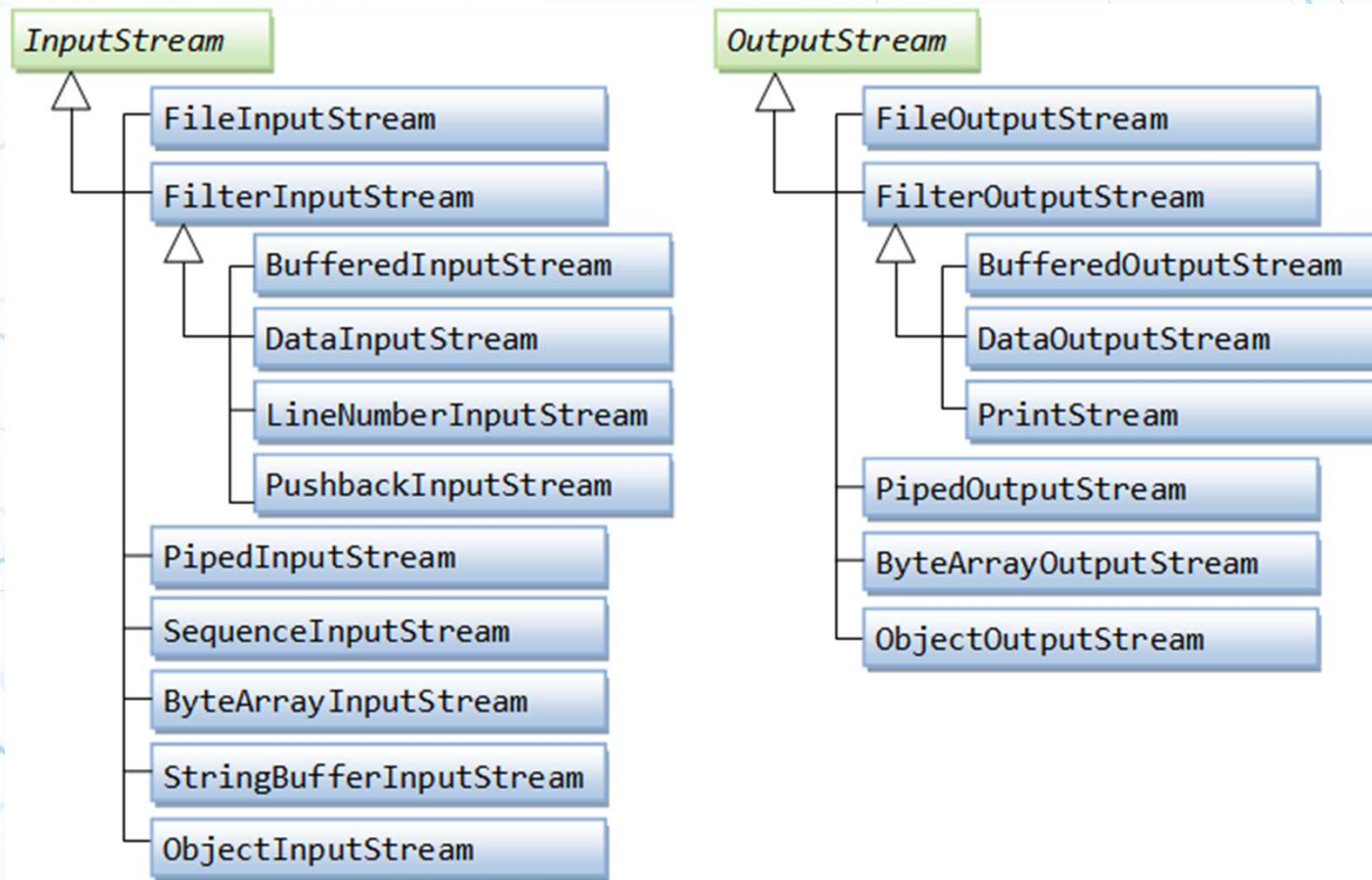


# Luồng nhập xuất byte

- ❑ Java cung cấp các lớp Stream hỗ trợ đọc/ghi byte (8 bits)
  - ❖ Có thể dùng lưu ký tự, videos, âm thanh, hình ảnh,...
- ❑ Lớp cha của các luồng nhập/xuất byte: The
  - ❖ InputStream
  - ❖ OutputStream



# Tổ chức phân cấp các lớp nhập xuất byte





# Danh sách một số lớp con của InputStream

Class	Description
<a href="#"><u>BufferedInputStream</u></a>	Đọc các byte từ bộ đệm (buffer)
<a href="#"><u>ByteArrayInputStream</u></a>	Đọc byte từ mảng byte
<a href="#"><u>DataInputStream</u></a>	Đọc kiểu dữ liệu nguyên thủy
<a href="#"><u>FileInputStream</u></a>	Đọc các byte từ File
<a href="#"><u>FilterInputStream</u></a>	Đọc các byte từ các Stream khác như nguồn dữ liệu chính
<a href="#"><u>ObjectInputStream</u></a>	Đọc các đối tượng
<a href="#"><u>PipedInputStream</u></a>	Đọc từ luồng đầu ra theo đường ống mà luồng đầu (piped output stream) vào theo đường ống (piped input stream) phải được kết nối.
<a href="#"><u>SequenceInputStream</u></a>	Kết nối nhiều Input stream và đọc dữ liệu từ chúng



# Phương thức thường dùng của InputStream

□ `int read():`

- ❖ Trả về một số nguyên 0-255,
- ❖ Trả về -1 khi kết thúc luồng nhập.

□ `int read(byte buffer [])`

- ❖ Đọc mảng các byte và lưu vào buffer
- ❖ Trả về -1 khi kết thúc luồng nhập.

□ `int read (byte buffer [], int loc, int nBytes)`

- ❖ Đọc mảng các byte bắt đầu từ loc và với nBytes cần đọc.
- ❖ Trả về tổng số byte đọc được. Và Trả về -1 nếu kết thúc luồng.





# Phương thức thường dùng của InputStream

❑ `int available()`

❖ Trả về số byte có sẵn để đọc.

❑ `void mark(int nBytes)`

❖ Đánh dấu vị trí hiện tại trong luồng nhập cho đến khi `nBytes` được chỉ định được đọc.

❑ `void reset()`

❖ Reset con trỏ nhập về đánh dấu (Mark) đã thiết lập trước đó.



# Phương thức thường dùng của InputStream

## ☐ long skip (long nBytes)

- ❖ Bỏ qua nByte của luồng nhập và trả về tổng số byte bị bỏ qua.

## ☐ void close()

- ❖ Đóng luồng nhập
- ❖ Ném IOException nếu đóng không thành công.



# Danh sách một số lớp con của OutputStream

Class	Description
<a href="#"><u>BufferedOutputStream</u></a>	Ghi các byte ra bộ đệm (buffer)
<a href="#"><u>ByteArrayOutputStream</u></a>	Ghi các byte ra mảng byte
<a href="#"><u>DataOutputStream</u></a>	Ghi các kiểu dữ liệu nguyên thủy (primitive data type)
<a href="#"><u>FileOutputStream</u></a>	Ghi các byte ra file
<a href="#"><u>FilterOutputStream</u></a>	Ghi ra luồng xuất khác
<a href="#"><u>ObjectOutputStream</u></a>	Ghi đối tượng
<a href="#"><u>PipedOutputStream</u></a>	Ghi các byte ra luồng (piped output stream)
<a href="#"><u>PrintStream</u></a>	In kiểu dữ liệu nguyên thủy (primitive data types)





# Phương thức thường dùng của OutputStream

- ❑ `void write(int i)`

- ❖ Ghi 1 byte vào luồng xuất

- ❑ `void write (byte buffer [] )`

- ❖ Ghi mảng byte vào luồng xuất (Output stream)

- ❑ `void write(bytes buffer[],int loc, int nBytes)`

- ❖ Ghi nByte byte vào luồng xuất từ buffer bắt đầu từ vị trí loc



# Phương thức thường dùng của OutputStream

❑ void flush()

❖ Đổ dữ liệu đệm của luồng xuất và ghi ra nguồn đích

❑ void close()

❖ Đóng luồng xuất

❖ IOException sẽ bị ném nếu không thành công