

CÁC LUỒNG VÀO RA - stream

NỘI DUNG

- Làm việc với Stream
- Lớp `java.io.InputStream`
- Lớp `java.io.OutputStream`
- Nhập chuỗi từ một `InputStream`
- Xuất chuỗi ra một `OutputStream`

Làm việc với Stream (1)

- Stream là một dòng liên tục, có thứ tự các bytes dữ liệu chảy giữa chương trình và các thiết bị ngoại vi.
- Nó là khái niệm trừu tượng hóa – giảm bớt các thao tác vào/ra phức tạp đối với người lập trình – có thể kết nối nhiều loại thiết bị ngoại vi với chương trình.
- Thao tác viết (write): chương trình gửi dữ liệu vào Stream.
- Thao tác đọc (read): chương trình nhận dữ liệu từ Stream.

Làm việc với Stream (2)

- Java hỗ trợ hai lớp Stream cơ bản trong gói java.
 - + `Java.io.InputStream`; Stream dùng nhận dữ liệu (Stream nhập) – dòng dữ liệu trong stream có hướng chảy từ ngoại vi vào chương trình.
 - + `Java.io.OutputStream`; Stream dùng gửi dữ liệu (Stream xuất) - dòng dữ liệu trong stream có hướng chảy từ chương trình ra thiết bị ngoại vi.
 - + Dữ liệu không được định dạng: Stream bao gồm từng ký tự (char) hoặc từng byte rời rạc (`byte[]`).
- Các lớp con của 2 lớp trên định nghĩa dạng dữ liệu nhận và dữ liệu gửi: `DataInputStream`, `DataOutputStream`, `FileInputStream`, `FileOutputStream`, ...

Lớp java.io.InputStream (1)

- Là loại Stream cho phép chương trình nhận dữ liệu từ ngoại vi.

- * Một số phương thức

- int read() throws IOException: Đọc một byte từ Stream.

- + return: 0-255: giá trị byte nhận được từ ngoại vi.

- 1 : Stream đã kết thúc.

- int read (byte b[]) throws IOException: Đọc tất cả các byte hiện có trong Stream đặt vào mảng b.

- + return: Số lượng byte đọc được.

- 1: Stream đã kết thúc.

- int read (byte b[], int offset, int len) : Đọc len byte từ Stream hiện tại, lưu vào trong mảng b bắt đầu từ vị trí offset.

- + return: Số lượng byte đọc được.

- 1 : Stream đã kết thúc.

Lớp java.io.InputStream (2)

- Các phương thức trên ngừng cho đến khi có dữ liệu hoặc kết thúc Stream hay một ngoại lệ xuất hiện.
- `int available()` : Trả về số lượng byte hiện có trong Stream mà không ngừng.
- `System.in` là một `InputStream` kết nối với bàn phím được tạo ra sẵn bởi hệ thống – để nhận các ký tự vào từ bàn phím.

Ví dụ 1: Hãy viết chương trình để nhập giá trị vào từ bàn phím InStream.java

Đọc một mảng các byte Stream:
VD.java

Lớp java.io.OutputStream

- Là loại Stream cho phép chương trình xuất dữ liệu ra thiết bị ngoại vi.
- * Một số phương thức:
 - void write (int b) throws Exception
 - + viết byte b vào Stream hiện tại.
 - + return: void
 - void write (byte[] b) throws Exception
 - + Viết tất cả các phần tử của mảng b vào Stream hiện tại.
 - + return: void
 - void write (int b[], int offset, int len) throws Exception
 - + Viết len byte trong mảng b vào Stream hiện tại, bắt đầu từ phần tử có chỉ số offset trong mảng.
 - + return: void
 - Trong đó: System.out là một OutputStream kết nối với màn hình được tạo ra sẵn bởi hệ thống – để gửi các ký tự ra màn hình.

Ví dụ

Nhập chuỗi từ một InputStream (1)

- InputStream là một Stream nhập gồm mảng các byte – nó đọc từng byte và lưu mảng các byte.
- InputStreamReader chuyển Stream nhập dạng byte sang stream nhập dạng ký tự.
- BufferedReader: hỗ trợ việc đọc văn bản từ một stream nhập dạng ký tự, lớp này thích hợp cho việc đọc các dòng văn bản.
- Java hỗ trợ 2 lớp cơ bản trong gói java.io.
 - + java.io.InputStreamReader: Là cầu nối để chuyển InputStream dạng byte sang InputStream dạng ký tự.
 - + java.io.BufferedReader: Hỗ trợ việc đọc văn bản.

Nhập chuỗi từ một InputStream (2)

- Phương thức `String readLine()` throws `IOException` của `BufferedReader`:

- + Đọc dòng văn bản kế tiếp trong stream nhập: một dòng kết thúc bởi cặp ký tự `'\n'` `'\n'` hoặc kết thúc stream.

- + return: một chuỗi ký tự hoặc null.

- Cách đọc một chuỗi từ một `InputStream`

- 1) `InputStreamReader isr = new InputStreamReader(System.in);`

- 2) `BufferedReader br = new BufferedReader(isr);`

- 3) `String str = br.readLine();`

- Ví dụ: Viết chương trình Cong.java để thực hiện phép cộng 2 số a, b (a,b được nhập từ bàn phím).

Xuất chuỗi ra một OutputStream (1)

- Lớp `java.io.PrintWriter`: để gửi chuỗi ra một `OutputStream`.

- * Một số phương thức:

- Khởi tạo:

`PrintWriter(OutputStream.out)`

- Gửi một chuỗi: `void write(String str);`

- Gửi một dòng: `void println(String line);`

- Đẩy dữ liệu từ bộ đệm ra Stream:

`void flush();`

Xuất chuỗi ra một OutputStream (2)

- Cách thức sử dụng:

```
OutputStream os = new OutputStream();
```

```
PrintWriter pw = new PrintWriter(os);
```

```
String str = “Đây là ví dụ về chuỗi \n \r”;
```

```
pw.write(str);
```

```
String line = “Đây là ví dụ về dòng”;
```

```
pw.println(line);
```

```
pw.flush();
```

Viết chuỗi ra màn hình Lưu chương trình sau vào tập tin
PrintString.java

- HẾT