## [Bài đọc] Các dòng vào/ra trong java API

**Các dòng vào/ra trong java API**

Mục này trình bày lại một cách có hệ thống các kiến thức về thư viện vào ra dữ liệu của Java mà ta đã nói đến rải rác ở các mục trước. Nội dung mục này chỉ ở mức giới thiệu sơ qua về một số dòng vào ra quan trọng. Các chi tiết cần được tra cứu ở tài liệu Java API.

Java coi mỗi file như là một dòng tuần tự các byte. Mỗi dòng như vậy có thể được hiểu là thuộc về một trong hai dạng: dòng kí tự (character-based stream) dành cho vào ra dữ liệu dạng kí tự và dòng byte (byte-based stream) dành cho dữ liệu dạng nhị phân. Ví dụ, nếu 5 được lưu với dòng byte, nó sẽ được lưu trữ ở dạng nhị phân của giá trị số 5, hay chuỗi bit 101. Còn nếu lưu bằng dòng kí tự, nó sẽ được lưu trữ ở dạng nhị phân của kí tự 5, hay chuỗi bit 00000000 00110101 (dạng nhị phân của giá trị 53, là mã Unicode của kí tự 5). File được tạo bằng dòng byte là file nhị phân, còn file được tạo bằng dòng kí tự là file văn bản. Con người có thể đọc nội dung file văn bản bằng các trình soạn thảo văn bản, còn các file nhị phân được đọc bởi các chương trình biến đổi dữ liệu nhị phân ra định dạng con người đọc được.

Để trao đổi dữ liệu với một file hay một thiết bị, chương trình Java tạo một dòng kết nối và nối với file hay thiết bị đó. Ví dụ, ta đã có sẵn ba dòng: System.in là dòng vào chuẩn (thường nối với bàn phím), System.out là dòng ra chuẩn (thường nối với cửa sổ lệnh), và System.err là dòng báo lỗi chuẩn (luôn nối với cửa sổ lệnh).

Các dòng dành cho việc xử lý dữ liệu nhị phân nằm trong hai cây phả hệ: các dòng có tổ tiên là InputStream để đọc dữ liệu, còn các dòng có tổ tiên là OutputStream để ghi dữ liệu. Các dòng cơ sở InputStream/OutputStream chỉ cung cấp các phương thức cho phép đọc/ghi dữ liệu thô ở dạng byte. Các lớp con của chúng cho phép đọc/ghi các giá trị thuộc các kiểu dữ liệu phức tạp hơn hoặc cho phép kết nối với các loại thiết bị cụ thể. Một số dòng quan trọng trong đó gồm có:

1. FileInputStream/FileOutputStream: dòng kết nối để nối trực tiếp với file nhị phân cần đọc/ghi theo dạng tuần tự.
2. ObjectInputStream/ObjectOutputStream: dòng nối tiếp, có thể nối với một InputStream/OutputStream khác. Các dòng này cho phép đọc/ghi từng đối tượng thuộc loại chuỗi hóa được.
3. DataInputStream/DataOutputStream: dòng nối tiếp, có thể nối với một InputStream/OutputStream khác, cho phép đọc/ghi các giá trị thuộc các kiểu cơ bản như int, long, boolean, ... (xem ví dụ như hình dưới)



Các dòng dành cho việc xử lý dữ liệu văn bản nằm trong hai cây phả hệ: các dòng có tổ tiên là Reader đọc dữ liệu, còn các dòng có tổ tiên là Writer ghi dữ liệu. Các dòng cơ sở Reader/Writer chỉ cung cấp các phương thức cho phép đọc/ghi dữ liệu ở dạng char hoặc chuỗi char. Các lớp con của chúng cho phép đọc/ghi với hiệu quả cao hơn và cung cấp các tiện ích bổ sung. Một số dòng quan trọng trong đó gồm có:

1. FileReader/FileWriter: dòng kết nối để nối trực tiếp với file cần đọc/ghi dữ liệu văn bản theo dạng tuần tự. FileReader cho phép đọc String từ file. FileWriter cho phép ghi String ra file.
2. BufferedReader/BufferedWriter: dòng nối tiếp, có thể nối với một Reader/Writer khác để đọc/ghi văn bản với bộ nhớ đệm nhằm tăng tốc độ xử lý.
3. InputStreamReader/OutputStreamWriter : dòng nối tiếp, là cầu nối từ dòng kí tự tới dòng byte, có thể nối với một InputStream/OutputStream. Nó cho phép đọc/ghi dữ liệu dạng kí tự được mã hóa trong một dòng byte theo một bộ mã cho trước.
4. PrintWriter: cho phép ghi dữ liệu có định dạng ra dòng kí tự, có thể kết nối trực tiếp với File, String, hoặc nối tiếp với một Writer hay OutputStream.

Ví dụ về InputStreamReader bên dưới. Kết nối Internet là nguồn dữ liệu dòng byte. Đầu tiên, nguồn vào được nối với một InputStream để có thể đọc dữ liệu byte thô. Sau đó, nó được nối với một InputStreamReader để chuyển từ dữ liệu byte sang dữ liệu văn bản. Cuối dùng, ta nối một BufferReader vào InputStreamReader để có thể đọc văn bản với tốc độ cao hơn.



Ví dụ về sử dụng PrintWriter (hình dưới),  dòng này cung cấp các phương thức ghi dữ liệu ra tương tự như ta quen dùng với dòng System.out

