## [Bài đọc] Scanner hoạt động như thế nào?

*(Nguồn: Introduction to Java Programming, Comprehensive Version (10th Edition)  - Y. Daniel Liang – Tr 479)*

Phương thức *nextByte(), nextShort(), nextInt(), nextLong(), nextFloat(), nextDouble(),* và *next()* được gọi là phương thức đọc kí tự đại diện (token-reading), vì chúng đọc các kí tự đại diện được phân tách bằng dấu phân cách. Theo mặc định, các dấu phân cách là các ký tự khoảng trắng. Bạn có thể sử dụng phương thức *useDelimiter(String regex)*để đặt một mẫu mới cho dấu phân cách.

Phương thức nhập liệu (input) hoạt động như thế nào? Trước tiên, một phương thức đọc kí tự đại diện sẽ bỏ qua bất kỳ dấu phân cách nào (khoảng trắng theo mặc định), sau đó đọc kí tự đại diện kết thúc bằng dấu phân cách. Kí tự đại diện sau đó sẽ tự động được chuyển đổi thành một giá trị của *byte, short, int, long, float* hoặc kiểu *double* cho *nextByte(), nextShort(), nextInt(), nextLong(), nextFloat(), và nextDouble()*. Đối với phương thức *next(),* không có chuyển đổi nào được thực hiện. Nếu kí tự đại diện không không khớp với kiểu mong đợi, ngoại lệ *java.util.InputMismatchException* sẽ được ném.

Cả hai phương thức *next(*) và *nextLine()* đều đọc một chuỗi. Phương thức *next()* đọc một chuỗi được phân tách bằng dấu phân cách và *nextLine()* đọc một dòng kết thúc bằng dấu phân cách dòng.

Phương thức đọc kí tự đại diện không đọc dấu phân cách sau kí tự đại diện. Nếu phương thức *nextLine()* được gọi sau một phương thức đọc kí tự đại diện, phương thức này đọc các ký tự bắt đầu từ dấu phân cách này và kết thúc bằng dấu phân cách dòng. Dấu phân cách dòng được đọc, nhưng nó không phải là một phần của chuỗi được trả về bởi *nextLine().*

Giả sử một file văn bản có tên test.txt chứa một dòng:

34 567

Sau khi đoạn mã sau được thực thi:

Scanner input = new Scanner(new File("test.txt"));

int intValue = input.nextInt();

String line = input.nextLine();

*intValue* chứa 34 và *line* chứa các ký tự ' ', 5, 6 và 7.