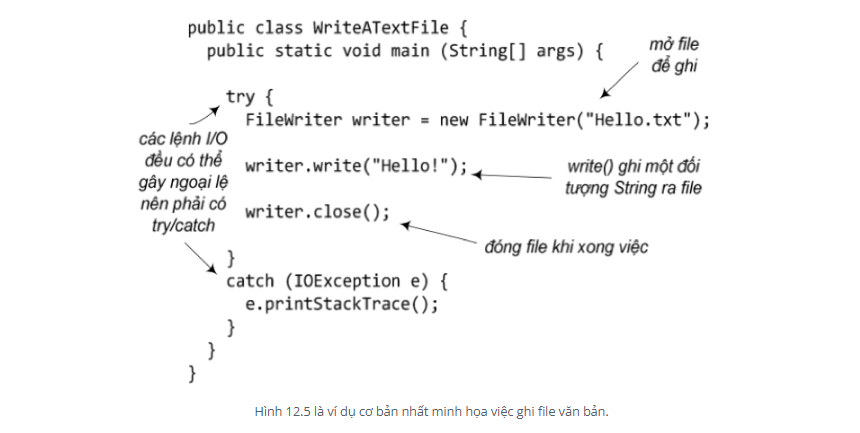
## [Bài đọc] Đọc/Ghi file văn bản

**Ghi chuỗi ký tự ra tệp văn bản**

Sử dụng cơ chế chuỗi hóa cho việc lưu trữ đối tượng là cách dễ dàng nhất để lưu trữ và khôi phục dữ liệu giữa các lần chạy của một chương trình Java. Nhưng đôi khi, ta cũng cần lưu dữ liệu vào một file văn bản, chẳng hạn khi file đó để cho một chương trình khác (có thể không viết bằng Java) đọc.

Việc ghi một chuỗi kí tự ra file văn bản tương tự với việc ghi một đối tượng, chỉ khác ở chỗ ta ghi một đối tượng String thay vì một đối tượng chung chung, và ta dùng các dòng khác thay cho FileOutputStream và ObjectOutputStream.



Java cho ta nhiều cách để tinh chỉnh chuỗi các dòng ra dùng cho việc ghi file.

**Bộ nhớ đệm**

Bộ nhớ đệm (buffer) cho ta một nơi lưu trữ tạm thời để tăng hiệu quả của thao tác đọc/ghi dữ liệu. Cách sử dụng BufferWriter như sau:

BufferWriter writer = new BufferWriter(new FileWriter(aFile);

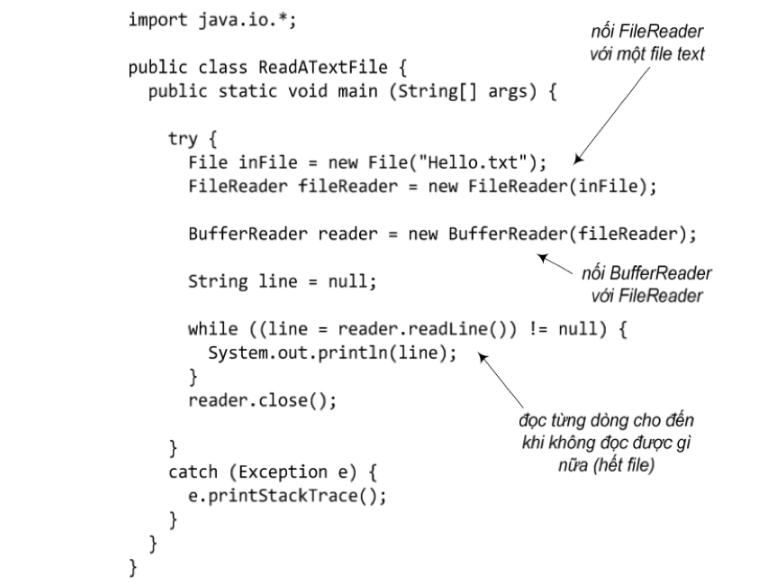
Sau lệnh trên thì ta chỉ cần làm việc với BufferWriter mà không cần để ý đến đối tượng FileWriter vừa tạo nữa.

Lợi ích của việc sử dụng BufferWriter được giải thích như sau: Nếu chỉ dùng FileWriter, mỗi lần ta yêu cầu FileWriter ghi một chuỗi dữ liệu nào đó, chuỗi đó lập tức được đổ vào file. Chi phí về thời gian xử lý cho mỗi lần ghi file là rất lớn so với chi phí cho các thao tác trong bộ nhớ. Khi nối một dòng BufferWriter với một FileWriter, BufferWriter sẽ giữ những gì ta ghi vào nó cho đến khi đầy. Chỉ khi bộ nhớ đệm BufferWriter đầy thì FileWriter mới được lệnh ghi dữ liệu ra đĩa. Như vậy, ta tăng được hiệu quả về mặt thời gian của việc ghi dữ liệu do giảm số lần ghi đĩa cứng. Nếu ta muốn đổ dữ liệu ra đĩa trước khi bộ nhớ đệm đầy, ta có thể gọi writer.flush() để lập tức xả toàn bộ nội dung trong bộ nhớ đệm.

**Đọc tệp văn bản**

Đọc từ file văn bản là công việc có quy trình tương tự ghi file, chỉ khác là giờ ta dùng một đối tượng FileReader để trực tiếp thực hiện công việc đọc file và một đối tượng BufferReader nối với nó để tăng hiệu quả đọc.

Hình 12.6 là ví dụ đơn giản về việc đọc một file văn bản. Trong đó, một đối tượng FileReader – một dòng kết nối cho dạng kí tự – được nối với một file để đọc trực tiếp. Tiếp theo là một đối tượng BufferReader được nối với FileReader để tăng hiệu quả đọc. Vòng while lặp đi lặp lại việc đọc một dòng từ BufferReader cho đến khi dòng đọc được là rỗng (tham chiếu null), đó là khi không còn gì để đọc nữa - đã chạm đến cuối file.



Như vậy với cách đọc này, ta đọc được dữ liệu dưới dạng các dòng văn bản. Để tách các giá trị dữ liệu tại mỗi dòng, ta cần xử lý chuỗi theo định dạng mà dữ liệu gốc đã được ghi. Chẳng hạn, nếu dữ liệu là các chuỗi kí tự cách nhau bởi dấu phảy thì ta sẽ phải tìm vị trí của các dấu phảy để tách các giá trị dữ liệu ra. Phương thức split của lớp String cho phép ta làm điều này. Ví dụ sử dụng phương thức split được cho trong Hình 12.7. Có thể tra cứu chi tiết về phương thức này tại tài liệu Java API.

