7   YÊU CẦU KHÔNG CHỨC NĂNG

1. Mã ứng dụng tài liệu

Ghi lại tất cả mã ứng dụng một cách chính xác để có thể dễ dàng bảo trì và nâng cao.

Nhận xét tài liệu

Nhận xét tài liệu nên được đặt trước bất kỳ lớp, giao diện, phương thức, hàm tạo hoặc khai báo trường nào. Các  
tiện ích Javadoc sử dụng những ý kiến để tạo ra định dạng HTML tham chiếu lớp hoặc Application Programming Interface  
(API) tài liệu. Mô tả tất cả các lớp công khai và lớp bảo vệ, lớp bên trong, giao diện, hàm tạo, phương thức và  
trường trong mã.

Viết giao diện lập trình trước mã vì tiện ích Javadoc có thể chạy trên các tệp nguồn chứa các lớp có phương thức là sơ khai không có phần thân.

Lập tài liệu cho các thành viên công khai, được bảo vệ, gói và riêng tư để cho phép tạo tài liệu cụ thể bằng tiện ích Javadoc.

· Bao gồm một phiên bản và dòng tác giả cho mỗi lần thay đổi mã

· Bao gồm mô tả chi tiết hơn về chức năng được cung cấp bởi lớp và phương thức thích hợp

Kiểu bình luận

Một nhận xét tài liệu tiêu chuẩn chứa một mô tả theo sau là một thẻ.

Ví dụ

/ \*\*

\* Thông tin chi tiết về tác giả

\*

\* @author Michael Smith

\* /

Việc sắp xếp thẻ Javadoc sau đây đã được Sun Microsystems khuyến nghị.

Trong các lớp và mô tả giao diện (Một thẻ @author và @version phải được bao gồm trong mọi mô tả lớp hoặc giao diện.):

/ \*\*

\* Sự miêu tả.

\*

\* @author

\* @phiên bản

\*

\* @xem

\* @từ

\* @deprecated

\* /

Trong mô tả phương thức

\*

\* Sự miêu tả.

\*

\* @param

\* @trở về

\* @ngoại lệ

\*

\* @xem

\* @từ

\* @deprecated

\* /

Một thẻ @param phải được bao gồm cho mọi thông số. Một thẻ @return nên được bao gồm nếu phương thức trả về một kiểu khác với void. Một thẻ @exception phải được bao gồm cho mọi ngoại lệ được kiểm tra trong mệnh đề ném và cho mọi ngoại lệ không được kiểm tra được ném.

Nhận xét  nội bộ bổ sung các nhận xét nội bộ đặc biệt về lý do tại sao mã đang làm những gì nó làm.

2. Các yêu cầu phi chức năng khác

•    trả tiền mặt hoặc dùng thẻ.