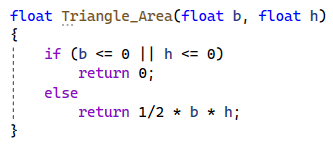
**Nguyễn Trắc Hùng**

**Câu 1: Trình bày và cho một ví dụ về kiểm thử hộp trắng**

Kiểm thử hộp trắng (White-box testing) là phương pháp kiểm thử phần mềm, trong đó các nhà phát triển phân tích và kiểm tra các phần mềm thông qua việc tìm hiểu các mã nguồn, cấu trúc và logic nội bộ của ứng dụng.

Một ví dụ về kiểm thử hộp trắng có thể là kiểm tra tính đúng đắn của một chương trình tính toán diện tích hình tam giác:



Trong ví dụ này, chúng ta kiểm tra tính đúng đắn của chương trình tính toán diện tích hình tam giác. Hàm Triangle\_Area sẽ trả về diện tích hình tam giác dựa trên độ dài đáy và chiều cao của nó. Tuy nhiên, hàm này cũng kiểm tra xem đáy và chiều cao có lớn hơn không hay không. Nếu bất kỳ tham số nào nhỏ hơn hoặc bằng không, hàm sẽ trả về giá trị 0.

Để kiểm tra tính đúng đắn của chương trình, các nhà phát triển có thể sử dụng kiểm thử hộp trắng bằng cách thiết kế các ca kiểm thử kiểm tra các trường hợp khác nhau của hàm, bao gồm các giá trị đầu vào hợp lệ và không hợp lệ. Ví dụ:

* Kiểm tra với độ dài đáy và chiều cao dương: Triangle\_Area **(3, 4)** trả về giá trị 6.
* Kiểm tra với độ dài đáy hoặc chiều cao âm: Triangle\_Area **(-3, 4)** và Triangle\_Area **(3, -4)** đều trả về giá trị 0.
* Kiểm tra với độ dài đáy hoặc chiều cao bằng 0: Triangle\_Area **(0, 4)** Triangle\_Area**(3, 0)** đều trả về giá trị 0.

Từ các kết quả kiểm thử này, các nhà phát triển có thể đảm bảo rằng chương trình sẽ hoạt động chính xác với tất cả các trường hợp đầu vào hợp lệ và không hợp lệ.

**Câu 2: Trình bày kiểm thử đường dẫn căn bản, ví dụ minh họa**

Kiểm thử đường dẫn căn bản (Basic Path Testing) là một phương pháp kiểm thử phần mềm tập trung vào việc kiểm tra các kịch bản (scenario) cơ bản, điểm qua tất cả các nhánh (branch) trong mã nguồn của phần mềm. Phương pháp này nhằm đảm bảo rằng phần mềm hoạt động đúng theo thiết kế và không bị ảnh hưởng bởi các lỗi phát sinh từ những hành vi không mong muốn của người dùng.

Ví dụ, khi kiểm thử một trang web đăng ký tài khoản, kiểm thử đường dẫn căn bản sẽ yêu cầu kiểm tra tất cả các trường hợp cơ bản trong quá trình đăng ký, bao gồm:

1. Trường hợp đăng ký thành công: Kiểm tra xem việc nhập thông tin tài khoản (tên đăng nhập, mật khẩu, email) đầy đủ và đúng cú pháp sẽ đăng ký thành công và trả về thông báo đăng ký thành công.
2. Trường hợp đăng ký thất bại do tên đăng nhập đã tồn tại: Kiểm tra xem việc nhập tên đăng nhập đã có trong hệ thống sẽ trả về thông báo lỗi tên đăng nhập đã tồn tại.
3. Trường hợp đăng ký thất bại do email đã tồn tại: Kiểm tra xem việc nhập email đã có trong hệ thống sẽ trả về thông báo lỗi email đã tồn tại.
4. Trường hợp đăng ký thất bại do mật khẩu không đủ mạnh: Kiểm tra xem việc nhập mật khẩu không đủ mạnh sẽ trả về thông báo lỗi mật khẩu không đủ mạnh.
5. Trường hợp đăng ký thất bại do không nhập đầy đủ thông tin: Kiểm tra xem việc không nhập đủ thông tin (tên đăng nhập, mật khẩu, email) sẽ trả về thông báo lỗi nhập đủ thông tin.
6. Trường hợp hủy đăng ký tài khoản: Kiểm tra xem việc hủy đăng ký tài khoản sẽ trả về thông báo xác nhận hủy đăng ký tài khoản.

Những trường hợp này đại diện cho các kịch bản đăng ký cơ bản và đầy đủ nhất, đảm bảo rằng các kịch bản này sẽ hoạt động đúng cách và không có lỗi phát sinh. Tuy nhiên, các kịch bản này chỉ đảm bảo được tính năng cơ bản vàcó thể bỏ qua những tình huống phức tạp hoặc không quan trọng, nên kiểm thử đường dẫn căn bản thường được sử dụng như một phương pháp kiểm thử ban đầu, trước khi chuyển sang các phương pháp kiểm thử chi tiết hơn như kiểm thử đường dẫn biên, kiểm thử giá trị biên, kiểm thử lỗi xác nhận, kiểm thử tương tác giữa các chức năng khác nhau,...

**Câu 3: Kiểm thử Frontend và ví dụ**

Kiểm thử frontend (hay còn gọi là kiểm thử giao diện người dùng) là một trong những phương pháp kiểm thử phần mềm quan trọng nhằm đảm bảo tính năng và độ tin cậy của giao diện người dùng của một ứng dụng hoặc trang web. Kiểm thử frontend thường được sử dụng để đảm bảo rằng giao diện người dùng của một sản phẩm hoạt động đúng và đáp ứng được các yêu cầu của người dùng.

Phương pháp kiểm thử frontend có thể bao gồm kiểm thử chức năng (đảm bảo các nút bấm, link, tính năng,... hoạt động đúng), kiểm thử giao diện (kiểm tra hình ảnh, vị trí và màu sắc của các phần tử trên giao diện), kiểm thử tương tác (kiểm tra sự tương tác của người dùng với giao diện), kiểm thử tính năng hợp lệ (kiểm tra tính hợp lệ của dữ liệu đầu vào) và kiểm thử hiệu suất (kiểm tra tốc độ tải trang, thời gian đáp ứng, thời gian phản hồi).

Ví dụ, khi kiểm thử giao diện của một trang web bán hàng, chúng ta có thể chia các trường hợp kiểm thử thành các tình huống sau:

1. Trường hợp xem trang chủ: Kiểm tra việc tải trang chủ một cách nhanh chóng và hiển thị đầy đủ các thông tin sản phẩm và các mục điều hướng.
2. Trường hợp tìm kiếm sản phẩm: Kiểm tra việc tìm kiếm sản phẩm trên trang web theo từ khóa được nhập vào và hiển thị kết quả tìm kiếm chính xác.
3. Trường hợp xem sản phẩm chi tiết: Kiểm tra việc hiển thị đầy đủ thông tin sản phẩm, các hình ảnh, đánh giá, và các tính năng khác của sản phẩm.
4. Trường hợp thêm sản phẩm vào giỏ hàng: Kiểm tra việc thêm sản phẩm vào giỏ hàng và kiểm tra số lượng sản phẩm hiển thị trong giỏ hàng.
5. Trường hợp thanh toán và đặt hàng: Kiểm tra việc điền đầy đủ thông tin thanh toán và đặt hàng, sau đó kiểm tra việc hiển thị thông báo xác nhận đặt hàng.

Với các trường hợp kiểm thử trên, chúng ta có thể đảm bảo rằng trang web bán hàng hoạt đ t động đúng và đáp ứng được các yêu cầu của người dùng, đồng thời giúp tăng tính năng và độ tin cậy của giao diện người dùng của trang web.

Ngoài ra, kiểm thử frontend còn có thể áp dụng các kỹ thuật như kiểm thử tự động và kiểm thử thủ công. Kiểm thử tự động giúp tiết kiệm thời gian và tăng độ chính xác của kiểm thử, trong khi kiểm thử thủ công đảm bảo việc kiểm tra thực tế và đảm bảo sự phù hợp với nhu cầu thực tế của người dùng.

Ví dụ, khi kiểm thử một trang web bán hàng bằng kiểm thử tự động, chúng ta có thể sử dụng các công cụ như Selenium hoặc Cypress để tự động hóa việc kiểm thử các trường hợp như đã đề cập ở trên. Các công cụ này cho phép tạo ra các kịch bản kiểm thử tự động và chạy chúng một cách tự động trên trình duyệt web, giúp giảm thiểu sự phụ thuộc vào con người và tiết kiệm thời gian.

Tuy nhiên, kiểm thử frontend bằng phương pháp tự động không thể thay thế hoàn toàn kiểm thử thủ công, vì nó không thể kiểm tra được các tình huống phức tạp và không đầy đủ. Do đó, kiểm thử frontend cần phải sử dụng cả kiểm thử tự động và kiểm thử thủ công để đảm bảo tính chính xác và độ tin cậy của giao diện người dùng.

Trong tổng quát, kiểm thử frontend là một phần quan trọng trong quá trình phát triển phần mềm, giúp đảm bảo tính năng và độ tin cậy của giao diện người dùng, cải thiện trải nghiệm người dùng và tăng sự hài lòng của người dùng.

**Câu 4: Tìm hiểu và sử dụng các kỹ thuật kiểm thử được hỗ trợ bởi công cụ kiểm thử Testpad.**

Ví dụ về sử dụng các kỹ thuật kiểm thử được hỗ trợ bởi Testpad:

Giả sử chúng ta có một ứng dụng web e-commerce đang được phát triển và chúng ta muốn kiểm tra tính năng thanh toán trên trang thanh toán. Để thực hiện kiểm thử, chúng ta có thể sử dụng Testpad và các kỹ thuật kiểm thử như sau:

1. Kiểm thử hộp đen: Chúng ta có thể tạo ra các ca kiểm thử dựa trên các yêu cầu và chức năng của trang thanh toán, ví dụ như kiểm tra tính năng thanh toán bằng thẻ tín dụng, kiểm tra tính năng thanh toán bằng ví điện tử, kiểm tra tính năng nhập mã giảm giá, v.v.
2. Kiểm thử hộp trắng: Chúng ta có thể kiểm tra cấu trúc bên trong của trang thanh toán để tìm lỗi bên trong. Ví dụ, chúng ta có thể kiểm tra tính đúng đắn của các hàm xử lý thanh toán hoặc kiểm tra các điều kiện biên của các trường nhập liệu trên trang thanh toán.
3. Kiểm thử hướng dịch vụ: Nếu trang thanh toán có các API liên quan đến thanh toán, chúng ta có thể sử dụng tính năng kiểm thử API của Testpad để kiểm tra tính đúng đắn của các API đó.
4. Kiểm thử tự động: Chúng ta có thể tạo ra các kịch bản kiểm thử tự động để thực hiện kiểm thử trên trang thanh toán. Ví dụ, chúng ta có thể tạo ra một kịch bản kiểm thử tự động để kiểm tra tính năng thanh toán bằng thẻ tín dụng bằng cách nhập các thông tin thẻ và thực hiện thanh toán.
5. Kiểm thử tích hợp: Nếu trang thanh toán tích hợp với các phần mềm khác như hệ thống thanh toán của ngân hàng, chúng ta có thểsử dụng tính năng kiểm thử tích hợp của Testpad để kiểm tra tính tương thích của trang thanh toán với các hệ thống đó.
6. Kiểm thử giao diện người dùng: Chúng ta có thể sử dụng tính năng kiểm thử giao diện người dùng của Testpad để kiểm tra tính đúng đắn và tương thích của giao diện trang thanh toán trên các trình duyệt khác nhau và các thiết bị khác nhau.

Tất cả các kỹ thuật kiểm thử này có thể được thực hiện và quản lý thông qua Testpad, giúp đảm bảo chất lượng của phần mềm và tiết kiệm thời gian và công sức của nhóm kiểm thử.