

**Kiểm thử CƠ BẢN**

**LAB 3**



Giảng viên hướng dẫn: **Huỳnh Khắc Duy**

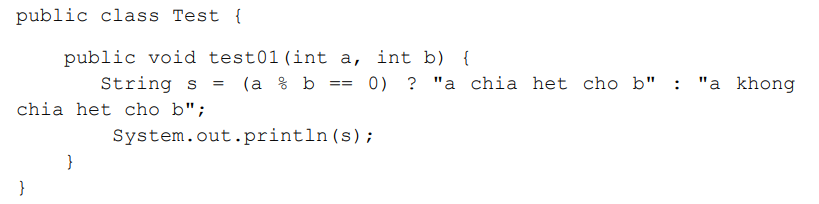
Lớp/Kỳ: SD18303/SPRING2024

Thành viên nhóm :

|  |  |  |
| --- | --- | --- |
| **Chức vụ** | **MSSV** | **Họ và Tên** |
| Trưởng nhóm | PS33069 | Nguyễn Quốc Khánh |
| Thành viên | PS28808 | Lê Đình Mẫn |
| Thành viên | PS28752 | Cao Ngọc Thái |
| Thành viên | PS26461 | Phạm Thế Tình |
| Thành viên | PS28794 | Phạm Văn Thiện |
| Thành viên | PS21469 | Nguyễn Thị Kim Yến |

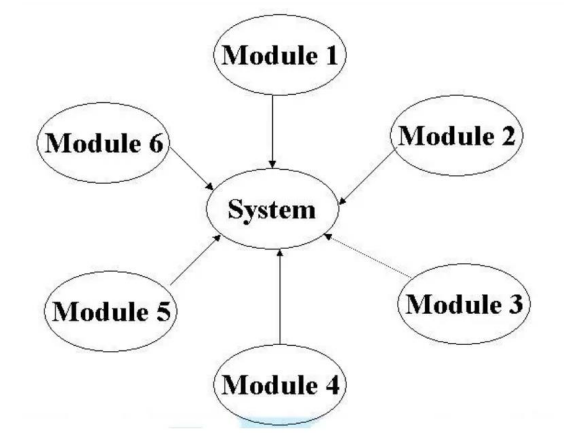
February 7, 2022

**Câu 1: Hãy liệt kê các trường hợp cần kiểm thử trong function test01?**

****

1. Trường hợp a chia hết cho b: Trường hợp khi phép chia a % b cho ra kết quả là 0.
   * 1. Giá trị của a là một số nguyên chia hết cho b.
     2. Giá trị của b khác 0.
2. Trường hợp a không chia hết cho b: Trường hợp khi phép chia a % b cho ra kết quả khác 0.
   * 1. Giá trị của a là một số nguyên không chia hết cho b.
     2. Giá trị của b khác 0.

**Câu 2: Hình bên dưới là phương pháp kiểm thử gì?**

  
- Là phương pháp kiểm thử bigbang trong mức độ kiểm thử intergation testing

- Đây là một phương pháp đơn giản và không cần lên lịch trước, trong đó các thành phần của hệ thống được kết hợp và kiểm thử một cách tổng thể mà không cần xác định trước thứ tự nào.

**Câu 3: null**

**Câu 4: Giả định khách hàng thay đổi requirement ở bất cứ thời điểm nào. Vì thế để thỏa mãn sự thay đổi của khách hàng, developer cần thay đổi logic và code của họ. Sau khi developer thay đổi code, chúng ta cần thực thi mức kiểm thử nào ?**

1. **Kiểm thử Đơn vị (Unit Testing):** Trước hết, các đoạn mã mới hoặc được sửa đổi nên được kiểm thử ở mức đơn vị. Điều này giúp đảm bảo rằng các hàm và phương thức hoạt động đúng như mong đợi và không gây ra các lỗi cơ bản.
2. **Kiểm thử Tích hợp (Integration Testing):** Sau khi đảm bảo rằng các đơn vị hoạt động đúng, bạn cần kiểm thử tích hợp để đảm bảo rằng các thành phần tương tác với nhau một cách đúng đắn. Điều này có thể giúp phát hiện ra các vấn đề tích hợp sớm hơn.
3. **Kiểm thử Hệ thống (System Testing):** Sau khi tích hợp các thành phần, bạn cần kiểm tra toàn bộ hệ thống để đảm bảo rằng nó đáp ứng đúng các yêu cầu của khách hàng. Điều này bao gồm kiểm thử chức năng, hiệu suất, bảo mật và các khía cạnh khác của hệ thống.
4. **Kiểm thử Chấp nhận (Acceptance Testing):** Trước khi triển khai sản phẩm, bạn nên thực hiện kiểm thử chấp nhận để đảm bảo rằng sản phẩm đáp ứng tất cả các yêu cầu của khách hàng. Các kịch bản kiểm thử chấp nhận thường được xác định dựa trên các yêu cầu từ người dùng cuối.
5. **Kiểm thử Tự động (Automated Testing):** Đối với các dự án lớn và có tính liên tục triển khai, kiểm thử tự động có thể giúp giảm thời gian kiểm thử và đảm bảo tính ổn định của hệ thống khi có sự thay đổi.
6. **Kiểm thử Hồi quy (Regression Testing):** Khi có thay đổi trong mã nguồn, bạn cần thực hiện kiểm thử hồi quy để đảm bảo rằng các thay đổi không ảnh hưởng đến các tính năng đã tồn tại và đã được kiểm thử trước đó.

**Câu 5: Nếu bạn có phần mềm máy tính có khả năng phép cộng, trừ, nhân, chia thì khi ra phiên bản mới bạn cần thực thi mức kiểm thử nào ? Và nêu một số trường hợp bạn có thể kiểm thử dựa trên mức kiểm thử bạn chọn ?**

Trong kiểm thử cơ bản của một phần mềm máy tính với khả năng thực hiện các phép toán cơ bản như phép cộng, trừ, nhân, chia, bạn cần thực hiện kiểm thử theo các mức độ khác nhau để đảm bảo rằng phần mềm hoạt động đúng và ổn định. Dưới đây là một số mức kiểm thử và trường hợp bạn có thể xem xét:

1. **Kiểm thử chức năng cơ bản:**

* Mục tiêu: Đảm bảo rằng các phép toán cơ bản như cộng, trừ, nhân, chia được thực hiện đúng.
* Trường hợp kiểm thử:
* Thực hiện phép cộng với các số nguyên dương và âm.
* Thực hiện phép trừ với các số nguyên dương và âm.
* Thực hiện phép nhân với các số nguyên dương và âm.
* Thực hiện phép chia với các số nguyên dương và âm (đảm bảo xử lý chia cho 0).

**2. Kiểm thử giới hạn:**

* Mục tiêu: Đảm bảo rằng phần mềm xử lý đúng ở các giới hạn của kiểu dữ liệu và biên của dữ liệu đầu vào.
* Trường hợp kiểm thử:
* Thử nghiệm với giá trị lớn nhất và nhỏ nhất của kiểu dữ liệu (ví dụ: INT\_MAX, INT\_MIN).
* Kiểm tra xem phần mềm có xử lý đúng khi dữ liệu vượt qua giới hạn của kiểu dữ liệu hay không.

**3. Kiểm thử hiệu suất:**

* Mục tiêu: Đảm bảo rằng phần mềm hoạt động hiệu quả với các bộ dữ liệu lớn.
* Trường hợp kiểm thử: Thực hiện các phép toán trên một loạt các số lớn để đảm bảo rằng hiệu suất không giảm đi đáng kể.

**4. Kiểm thử giao diện người dùng (UI):**

* Mục tiêu: Đảm bảo rằng giao diện người dùng hiển thị kết quả đúng và dễ hiểu.
* Trường hợp kiểm thử: Kiểm tra giao diện khi thực hiện các phép toán và hiển thị kết quả.

**5. Kiểm thử tương thích:**

* Mục tiêu: Đảm bảo rằng phần mềm hoạt động đúng trên nhiều hệ điều hành và môi trường khác nhau (nếu có).
* Trường hợp kiểm thử: Kiểm tra tính tương thích trên các hệ điều hành phổ biến như Windows, Linux, MacOS.

**6. Kiểm thử bảo mật cơ bản:**

* Mục tiêu: Đảm bảo rằng phần mềm không bị dễ tấn công thông qua các lỗ hổng cơ bản.
* Trường hợp kiểm thử:
* Kiểm tra xem phần mềm có xử lý đúng khi có dữ liệu đầu vào độc hại hay không.
* Mức kiểm thử cần phải thực hiện phụ thuộc vào yêu cầu cụ thể của ứng dụng và ngữ cảnh triển khai của nó.