

**Kiểm thử CƠ BẢN**

**LAB 5**



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 |

**Câu 1: Sử dụng kỹ thuật phân vùng tương đương để liệt kê các trường hợp kiểm thử**

1. **Phân vùng cho xếp loại Kém:**
   * Giá trị dưới 5 (ngoài phạm vi).
   * Giá trị từ 5 đến 14 (xếp loại Kém).
   * Giá trị từ 15 đến 40 (ngoài phạm vi).
   * Giá trị từ 41 đến 60 (ngoài phạm vi).
   * Giá trị từ 61 trở lên (ngoài phạm vi).
2. **Phân vùng cho xếp loại Trung bình:**
   * Giá trị dưới 5 (ngoài phạm vi).
   * Giá trị từ 5 đến 14 (ngoài phạm vi).
   * Giá trị từ 15 đến 40 (xếp loại Trung bình).
   * Giá trị từ 41 đến 60 (ngoài phạm vi).
   * Giá trị từ 61 trở lên (ngoài phạm vi).
3. **Phân vùng cho xếp loại Khá:**
   * Giá trị dưới 5 (ngoài phạm vi).
   * Giá trị từ 5 đến 14 (ngoài phạm vi).
   * Giá trị từ 15 đến 40 (ngoài phạm vi).
   * Giá trị từ 41 đến 60 (xếp loại Khá).
   * Giá trị từ 61 trở lên (ngoài phạm vi).
4. **Phân vùng cho xếp loại Giỏi:**
   * Giá trị dưới 5 (ngoài phạm vi).
   * Giá trị từ 5 đến 14 (ngoài phạm vi).
   * Giá trị từ 15 đến 40 (ngoài phạm vi).
   * Giá trị từ 41 đến 60 (ngoài phạm vi).
   * Giá trị từ 61 trở lên (xếp loại Giỏi).

Các giá trị được chọn từ mỗi phân vùng trên sẽ được sử dụng để thực hiện các bài kiểm thử để đảm bảo rằng hệ thống xử lý đúng các xếp loại theo yêu cầu.

**Câu 2: Sử dụng kỹ thuật phân tích giá trị biên để liệt kê các trường hợp kiểm thử**

1. **Xếp loại Kém (5-14):**
   * Kiểm thử với giá trị 5 (giá trị thấp nhất).
   * Kiểm thử với giá trị 14 (giá trị cao nhất).
   * Kiểm thử với giá trị ngẫu nhiên trong khoảng từ 5 đến 14.
2. **Xếp loại Trung bình (15-40):**
   * Kiểm thử với giá trị 15 (giá trị thấp nhất).
   * Kiểm thử với giá trị 40 (giá trị cao nhất).
   * Kiểm thử với giá trị ngẫu nhiên trong khoảng từ 15 đến 40.
3. **Xếp loại Khá (41-60):**
   * Kiểm thử với giá trị 41 (giá trị thấp nhất).
   * Kiểm thử với giá trị 60 (giá trị cao nhất).
   * Kiểm thử với giá trị ngẫu nhiên trong khoảng từ 41 đến 60.
4. **Xếp loại Giỏi (61-100):**
   * Kiểm thử với giá trị 61 (giá trị thấp nhất).
   * Kiểm thử với giá trị 100 (giá trị cao nhất).
   * Kiểm thử với giá trị ngẫu nhiên trong khoảng từ 61 đến 100.
5. **Kiểm thử giá trị ngoài phạm vi:**
   * Kiểm thử với giá trị dưới 5.
   * Kiểm thử với giá trị trên 100.

Bằng cách này, bạn đảm bảo rằng các giá trị biên của mỗi phân vùng đã được kiểm thử, đồng thời cũng kiểm tra các giá trị ngẫu nhiên trong khoảng để đảm bảo tính toàn vẹn của hệ thống.

Câu 3: Để áp dụng kỹ thuật bảng quyết định cho bài toán này, chúng ta cần xác định các biến đầu vào và giá trị của chúng, sau đó tạo ra bảng quyết định chứa các trường hợp kiểm thử.

Các biến đầu vào:

1. **Loại khách hàng:**
   * Khách hàng mới.
   * Khách hàng cũ (với thẻ Vip).
2. **Có Coupon hay không:**
   * Có Coupon.
   * Không có Coupon.

Các biến đầu vào khác có thể thêm vào bảng quyết định nếu có yêu cầu cụ thể khác.

Bảng quyết định:

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại khách hàng** | **Có Coupon** | **Giảm giá** | **Ghi chú** |
| Khách hàng mới | Không | 15% | Giảm giá 15% cho khách hàng mới |
| Khách hàng mới | Có | 20% | Giảm giá 20% với Coupon (Không có áp dụng với khách hàng mới) |
| Khách hàng cũ (Vip) | Không | 10% | Giảm giá 10% cho khách hàng cũ có thẻ Vip |
| Khách hàng cũ (Vip) | Có | 20% | Giảm giá 20% cho khách hàng cũ có thẻ Vip |

Bảng quyết định này bao gồm các trường hợp kiểm thử dựa trên các biến đầu vào và các điều kiện giảm giá. Mỗi ô trong bảng đại diện cho một trường hợp kiểm thử cụ thể.

Lưu ý rằng có thể cần thêm vào bảng quyết định nếu có các biến khác hoặc điều kiện phức tạp hơn trong thực tế.

1. **Xếp loại Kém (5-14):**
   * Kiểm thử với giá trị 5 (giá trị thấp nhất).
   * Kiểm thử với giá trị 14 (giá trị cao nhất).
   * Kiểm thử với giá trị ngẫu nhiên trong khoảng từ 5 đến 14.
2. **Xếp loại Trung bình (15-40):**
   * Kiểm thử với giá trị 15 (giá trị thấp nhất).
   * Kiểm thử với giá trị 40 (giá trị cao nhất).
   * Kiểm thử với giá trị ngẫu nhiên trong khoảng từ 15 đến 40.
3. **Xếp loại Khá (41-60):**
   * Kiểm thử với giá trị 41 (giá trị thấp nhất).
   * Kiểm thử với giá trị 60 (giá trị cao nhất).
   * Kiểm thử với giá trị ngẫu nhiên trong khoảng từ 41 đến 60.
4. **Xếp loại Giỏi (61-100):**
   * Kiểm thử với giá trị 61 (giá trị thấp nhất).
   * Kiểm thử với giá trị 100 (giá trị cao nhất).
   * Kiểm thử với giá trị ngẫu nhiên trong khoảng từ 61 đến 100.
5. **Kiểm thử giá trị ngoài phạm vi:**
   * Kiểm thử với giá trị dưới 5.
   * Kiểm thử với giá trị trên 100.

Bằng cách này, bạn đảm bảo rằng các giá trị biên của mỗi phân vùng đã được kiểm thử, đồng thời cũng kiểm tra các giá trị ngẫu nhiên trong khoảng để đảm bảo tính toàn vẹn của hệ thống.

Câu 3: Để áp dụng kỹ thuật bảng quyết định cho bài toán này, chúng ta cần xác định các biến đầu vào và giá trị của chúng, sau đó tạo ra bảng quyết định chứa các trường hợp kiểm thử.

Các biến đầu vào:

1. **Loại khách hàng:**
   * Khách hàng mới.
   * Khách hàng cũ (với thẻ Vip).
2. **Có Coupon hay không:**
   * Có Coupon.
   * Không có Coupon.

Các biến đầu vào khác có thể thêm vào bảng quyết định nếu có yêu cầu cụ thể khác.

Bảng quyết định:

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại khách hàng** | **Có Coupon** | **Giảm giá** | **Ghi chú** |
| Khách hàng mới | Không | 15% | Giảm giá 15% cho khách hàng mới |
| Khách hàng mới | Có | 20% | Giảm giá 20% với Coupon (Không có áp dụng với khách hàng mới) |
| Khách hàng cũ (Vip) | Không | 10% | Giảm giá 10% cho khách hàng cũ có thẻ Vip |
| Khách hàng cũ (Vip) | Có | 20% | Giảm giá 20% cho khách hàng cũ có thẻ Vip |

Bảng quyết định này bao gồm các trường hợp kiểm thử dựa trên các biến đầu vào và các điều kiện giảm giá. Mỗi ô trong bảng đại diện cho một trường hợp kiểm thử cụ thể.

Lưu ý rằng có thể cần thêm vào bảng quyết định nếu có các biến khác hoặc điều kiện phức tạp hơn trong thực tế.

**Câu 3: Sử dụng kỹ thuật bảng quyết định xác định test case cho bài toán khách hàng đến mở thẻ tín dụng với các điều kiện sau:**

**+ Nếu bạn là một khách hàng mới, đến mở thẻ tín dụng, bạn sẽ được giảm giá 15%.**

**+ Nếu bạn là khách hàng cũ, và có thẻ Vip, bạn sẽ được giảm giá 10%.**

**+ Nếu bạn có Coupon, bạn sẽ được giảm giá 20% (nhưng nó không được sử dụng giảm giá cùng với khách hàng mới.**

**+ Việc giảm giá có thể được cộng nếu như phù hợp.**

Để áp dụng kỹ thuật bảng quyết định cho bài toán này, chúng ta cần xác định các biến đầu vào và giá trị của chúng, sau đó tạo ra bảng quyết định chứa các trường hợp kiểm thử.

**Các biến đầu vào:**

1. **Loại khách hàng:**
   * Khách hàng mới.
   * Khách hàng cũ (với thẻ Vip).
2. **Có Coupon hay không:**
   * Có Coupon.
   * Không có Coupon.

Các biến đầu vào khác có thể thêm vào bảng quyết định nếu có yêu cầu cụ thể khác.

Bảng quyết định:

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại khách hàng** | **Có Coupon** | **Giảm giá** | **Ghi chú** |
| Khách hàng mới | Không | 15% | Giảm giá 15% cho khách hàng mới |
| Khách hàng mới | Có | 20% | Giảm giá 20% với Coupon (Không có áp dụng với khách hàng mới) |
| Khách hàng cũ (Vip) | Không | 10% | Giảm giá 10% cho khách hàng cũ có thẻ Vip |
| Khách hàng cũ (Vip) | Có | 20% | Giảm giá 20% cho khách hàng cũ có thẻ Vip |

Bảng quyết định này bao gồm các trường hợp kiểm thử dựa trên các biến đầu vào và các điều kiện giảm giá. Mỗi ô trong bảng đại diện cho một trường hợp kiểm thử cụ thể.

Lưu ý rằng có thể cần thêm vào bảng quyết định nếu có các biến khác hoặc điều kiện phức tạp hơn trong thực tế.

**Câu 4: Với hàm code bên dưới hãy thực hiện xác định các nút, vẽ đường cơ bản và liệt kê các trường hợp kiểm thử bằng đường đi:**

**void soNT(int n) {**

**boolean co = true;**

**for (int i = 2; i < n; i++) {**

**if (n % i == 0) {**

**co = false; break;**

**}**

**}**

**if (n != 1 && co == true) {**

**System.out.print("là số nguyên tố ");**

**} else System.out.print("không phải là nguyên tố");**

**}**

Để xác định các đường đi và liệt kê các trường hợp kiểm thử bằng đường đi, chúng ta cần phân tích các điều kiện trong hàm. Dưới đây là phân tích chi tiết cho hàm soNT:

1. Đường đi cơ bản:

· Không thực hiện vòng lặp (n = 1)

2. Các đường đi có vòng lặp:

· Vòng lặp không được thực hiện (n < 2)

· Vòng lặp thực hiện ít nhất một lần (n >= 2)

· Lặp qua các giá trị của i từ 2 đến n-1

3. Các điều kiện kiểm tra trong vòng lặp:

· n % i == 0 (nếu điều kiện đúng thì thoát khỏi vòng lặp)

4. Điều kiện cuối cùng:

· n != 1 && co == true

Dựa trên phân tích này, chúng ta có thể liệt kê các trường hợp kiểm thử bằng đường đi như sau:

1. Trường hợp: n = 1

· Đầu vào: n = 1

· Đầu ra mong đợi: "không phải là nguyên tố"

2. Trường hợp: n < 2

· Đầu vào: n = 0 hoặc n = 1

· Đầu ra mong đợi: "không phải là nguyên tố"

3. Trường hợp: n >= 2 và n là số nguyên tố

· Đầu vào: n là số nguyên tố (ví dụ: n = 2, n = 3, ...)

· Đầu ra mong đợi: "là số nguyên tố"

4. Trường hợp: n >= 2 và n không phải là số nguyên tố

· Đầu vào: n không phải là số nguyên tố (ví dụ: n = 4, n = 6, ...)

· Đầu ra mong đợi: "không phải là nguyên tố"

Với các trường hợp kiểm thử trên, bạn có thể tạo các bộ kiểm thử để đảm bảo rằng hàm soNT hoạt động đúng cho tất cả các trường hợp có thể xảy ra.

**Câu 5: Xác định số test case tối thiểu bao phủ 100% Câu lệnh - Statement coverage**

**if ((temperature < 0) or (temperature > 100)) { alert ("DANGER"); if ((speed > 100) and (load <= 50)) { speed = 50; } else { check = false; }**

Để xác định số lượng test case tối thiểu bao phủ 100% Câu lệnh (Statement coverage) cho đoạn mã này, chúng ta cần xem xét từng điều kiện và các nhánh điều kiện.

Đoạn mã có hai điều kiện if-else, vậy nên chúng ta cần thiết kế test case để kiểm thử cả hai nhánh của mỗi điều kiện.

1. Điều kiện thứ nhất:

· Test Case 1: temperature < 0

· Test Case 2: temperature > 100

· Test Case 3: 0 <= temperature <= 100 (để kiểm tra nhánh else)

2. Điều kiện thứ hai:

· Test Case 4: (speed > 100) and (load <= 50)

· Test Case 5: (speed <= 100) or (load > 50) (để kiểm tra nhánh else)

Các test case trên đã bao phủ tất cả các nhánh của đoạn mã. Bạn có thể kiểm thử từng điều kiện và nhánh của chúng để đảm bảo rằng mọi trường hợp đều được xử lý đúng.

Lưu ý: Các test case trên chỉ đảm bảo 100% Câu lệnh coverage, điều này không nhất thiết đồng nghĩa với việc bao phủ đầy đủ mọi trường hợp có thể xảy ra trong chương trình. Đối với kiểm thử tổng thể tốt hơn, bạn cũng có thể xem xét kiểm thử các biên của dải giá trị cho các biến khác nhau (temperature, speed, load) để đảm bảo tính chặt chẽ của chương trình.