**MỤC TIÊU:**

Kết thúc bài thực hành này bạn có khả năng

* Sử dụng kỹ thuật hộp đen để phân tích các trường hợp kiểm thử
* Sử dụng kỹ thuật kinh nghiệm để phân tích các trường hợp kiểm thử
* Sử dụng kỹ thuật hộp trắng để phân tích các trường hợp kiểm thử

# PHẦN I

Giả sử một ô textbox cho phép nhập số điểm với các ràng buộc như sau:

+ Số điểm từ 5-14: xếp loại kém

+ Số điểm từ 15-40: xếp loại Trung bình

+ Số điểm từ 41-60: xếp loại khá

+ Số điểm từ 61-100: xếp loại Giỏi

**Bài 1 (2 điểm)**

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

Kỹ thuật phân vùng tương đương là kỹ thuật kiểm thử phần mềm chia miền đầu vào của một chương trình thành nhiều lớp dữ liệu, từ đó suy dẫn ra các ca kiểm thử. Các lớp dữ liệu này được gọi là các lớp tương đương.

Để sử dụng kỹ thuật phân vùng tương đương để liệt kê các trường hợp kiểm thử, cần thực hiện các bước sau:

1. Xác định các miền đầu vào.
2. Xác định các lớp tương đương.
3. Xác định các trường hợp kiểm thử.

Bước 1: Xác định các miền đầu vào

Miền đầu vào là tập hợp tất cả các giá trị có thể được nhập vào chương trình. Trong ví dụ sau, miền đầu vào là các giá trị số nguyên.

int input;

Bước 2: Xác định các lớp tương đương

Các lớp tương đương là các tập hợp các giá trị có cùng hành vi đối với chương trình. Trong ví dụ sau, có thể chia miền đầu vào thành các lớp tương đương sau:

* Lớp tương đương 1: Giá trị nhỏ hơn hoặc bằng 0.
* Lớp tương đương 2: Giá trị từ 1 đến 10.
* Lớp tương đương 3: Giá trị lớn hơn hoặc bằng 11.

Bước 3: Xác định các trường hợp kiểm thử

Mỗi lớp tương đương sẽ được đại diện bởi một trường hợp kiểm thử. Do đó, trong ví dụ sau, có thể xác định các trường hợp kiểm thử sau:

* Trường hợp kiểm thử 1: Nhập giá trị 0.
* Trường hợp kiểm thử 2: Nhập giá trị 5.
* Trường hợp kiểm thử 3: Nhập giá trị 12.

Với kỹ thuật phân vùng tương đương, có thể đảm bảo rằng tất cả các giá trị trong miền đầu vào đều được kiểm tra. Tuy nhiên, kỹ thuật này có thể không phát hiện được các lỗi biên. Để phát hiện các lỗi biên, có thể sử dụng các kỹ thuật kiểm thử khác như phân tích giá trị biên.

Dưới đây là một số ví dụ về kỹ thuật phân vùng tương đương trong thực tế:

* Kiểm tra tính hợp lệ của dữ liệu đầu vào. Ví dụ: kiểm tra xem số điện thoại có phải là số điện thoại hợp lệ hay không.
* Kiểm tra các trường hợp ngoại lệ. Ví dụ: kiểm tra xem chương trình có xử lý đúng các trường hợp lỗi hay không.
* Kiểm tra các tính năng mới. Ví dụ: kiểm tra xem tính năng mới có hoạt động đúng như mong đợi hay không.

**Bài 2 (2 điểm)**

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

Kỹ thuật phân tích giá trị biên là kỹ thuật kiểm thử phần mềm tập trung vào các giá trị biên của miền đầu vào. Các giá trị biên là các giá trị nằm ở ranh giới giữa các lớp tương đương.

Để sử dụng kỹ thuật phân tích giá trị biên để liệt kê các trường hợp kiểm thử, cần thực hiện các bước sau:

1. Xác định các miền đầu vào.
2. Xác định các lớp tương đương.
3. Xác định các giá trị biên.
4. Liệt kê các trường hợp kiểm thử.

Bước 1: Xác định các miền đầu vào

Miền đầu vào là tập hợp tất cả các giá trị có thể được nhập vào chương trình. Trong ví dụ sau, miền đầu vào là các giá trị số nguyên.

int input;

Bước 2: Xác định các lớp tương đương

Các lớp tương đương là các tập hợp các giá trị có cùng hành vi đối với chương trình. Trong ví dụ sau, có thể chia miền đầu vào thành các lớp tương đương sau:

* Lớp tương đương 1: Giá trị nhỏ hơn hoặc bằng 0.
* Lớp tương đương 2: Giá trị từ 1 đến 10.
* Lớp tương đương 3: Giá trị lớn hơn hoặc bằng 11.

Bước 3: Xác định các giá trị biên

Các giá trị biên là các giá trị nằm ở ranh giới giữa các lớp tương đương. Trong ví dụ sau, các giá trị biên là:

* Giá trị biên 1: Giá trị lớn nhất của lớp tương đương 1, tức là 0.
* Giá trị biên 2: Giá trị nhỏ nhất của lớp tương đương 2, tức là 1.
* Giá trị biên 3: Giá trị lớn nhất của lớp tương đương 2, tức là 10.
* Giá trị biên 4: Giá trị nhỏ nhất của lớp tương đương 3, tức là 11.

Bước 4: Liệt kê các trường hợp kiểm thử

Mỗi giá trị biên sẽ được đại diện bởi một trường hợp kiểm thử. Do đó, trong ví dụ sau, có thể xác định các trường hợp kiểm thử sau:

* Trường hợp kiểm thử 1: Nhập giá trị -1.
* Trường hợp kiểm thử 2: Nhập giá trị 0.
* Trường hợp kiểm thử 3: Nhập giá trị 10.
* Trường hợp kiểm thử 4: Nhập giá trị 11.

Với kỹ thuật phân tích giá trị biên, có thể đảm bảo rằng tất cả các giá trị biên của miền đầu vào đều được kiểm tra. Các giá trị biên thường là những giá trị dễ xảy ra lỗi, do đó việc kiểm tra các giá trị biên có thể giúp phát hiện được nhiều lỗi hơn.

Dưới đây là một số ví dụ về kỹ thuật phân tích giá trị biên trong thực tế:

* Kiểm tra tính hợp lệ của dữ liệu đầu vào. Ví dụ: kiểm tra xem số điện thoại có phải là số điện thoại hợp lệ hay không.
* Kiểm tra các trường hợp ngoại lệ. Ví dụ: kiểm tra xem chương trình có xử lý đúng các trường hợp lỗi hay không.
* Kiểm tra các tính năng mới. Ví dụ: kiểm tra xem tính năng mới có hoạt động đúng như mong đợi hay không.

Một số lưu ý khi sử dụng kỹ thuật phân tích giá trị biên:

* Cần xác định các miền đầu vào và các lớp tương đương chính xác.
* Cần xác định các giá trị biên một cách đầy đủ.
* Ngoài các giá trị biên, cần kiểm tra thêm các giá trị đặc biệt khác, chẳng hạn như các giá trị bằng không, các giá trị rỗng, v.v.

## Bài 3 (1 điểm)

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.

Bảng quyết định

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Điều kiện | Khách hàng mới | Khách hàng cũ | Có thẻ Vip | Có Coupon |
| Khách hàng mới | 15% | - | - | - |
| Khách hàng cũ | - | 10% | - | - |
| Có thẻ Vip | - | 10% | - | - |
| Có Coupon | - | - | - | 20% |

Liệt kê các trường hợp kiểm thử

Từ bảng quyết định, có thể liệt kê các trường hợp kiểm thử như sau:

|  |  |  |
| --- | --- | --- |
| Trường hợp kiểm thử | Điều kiện | Kết quả mong đợi |
| 1 | Khách hàng mới, có Coupon | 15% + 20% = 35% |
| 2 | Khách hàng mới, không có Coupon | 15% |
| 3 | Khách hàng cũ, có thẻ Vip, không có Coupon | 10% |
| 4 | Khách hàng cũ, có thẻ Vip, có Coupon | 10% + 20% = 30% |
| 5 | Khách hàng cũ, không có thẻ Vip, không có Coupon | Không được giảm giá |
| 6 | Khách hàng cũ, không có thẻ Vip, có Coupon | 20% |

Lưu ý

* Trường hợp 1 và 2 kiểm tra trường hợp khách hàng mới được giảm giá 15% hoặc 35% tùy thuộc vào việc có Coupon hay không.
* Trường hợp 3 và 4 kiểm tra trường hợp khách hàng cũ được giảm giá 10% hoặc 30% tùy thuộc vào việc có thẻ Vip hay không.
* Trường hợp 5 và 6 kiểm tra trường hợp khách hàng cũ không được giảm giá.

Ngoài các trường hợp kiểm thử trên, có thể bổ sung thêm các trường hợp kiểm thử sau để kiểm tra các trường hợp đặc biệt:

* Khách hàng có thể nhập giá trị không hợp lệ cho các điều kiện. Ví dụ, khách hàng nhập "Không" cho điều kiện "Khách hàng mới".
* Khách hàng có thể nhập giá trị rỗng cho các điều kiện. Ví dụ, khách hàng không nhập giá trị cho điều kiện "Có thẻ Vip".

# PHẦN II

## Bài 4 (2 điểm)

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");

}

Xác định các nút

Dựa trên cấu trúc của hàm, có thể xác định các nút sau:

* Nút đầu vào: int n
* Nút quyết định: if (n % i == 0)
* Nút kết thúc: if (n != 1 && co == true)

Vẽ đường cơ bản

Đường cơ bản là đường đi qua tất cả các nút của hàm, bao gồm cả các nút quyết định. Đường cơ bản của hàm soNT() như sau:

Nút đầu vào --> Nút quyết định --> Nút kết thúc

Liệt kê các trường hợp kiểm thử bằng đường đi

Từ đường cơ bản, có thể liệt kê các trường hợp kiểm thử bằng đường đi như sau:

|  |  |  |
| --- | --- | --- |
| Trường hợp kiểm thử | Giá trị của n | Kết quả mong đợi |
| 1 | 2 | Là số nguyên tố |
| 2 | 3 | Là số nguyên tố |
| 3 | 4 | Không phải là số nguyên tố |
| 4 | 5 | Là số nguyên tố |
| 5 | 6 | Không phải là số nguyên tố |

Lưu ý:

* Các trường hợp kiểm thử 1 và 2 kiểm tra trường hợp n là số nguyên tố.
* Các trường hợp kiểm thử 3 và 4 kiểm tra trường hợp n không phải là số nguyên tố.

Ngoài các trường hợp kiểm thử trên, có thể bổ sung thêm các trường hợp kiểm thử sau để kiểm tra các trường hợp đặc biệt:

* n là số âm.
* n bằng 0.
* n là số nguyên tố lớn.

## Bài 5 (2 điểm)

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 đạt được 100% Statement Coverage (bao phủ câu lệnh), cần tối thiểu 3 trường hợp kiểm thử.

Giải thích:

* Câu lệnh 1: if ((temperature < 0) or (temperature > 100)) {
  + Để bao phủ câu lệnh này, cần một trường hợp thỏa mãn điều kiện (temperature < 0 hoặc temperature > 100) và một trường hợp không thỏa mãn điều kiện.
* Câu lệnh 2: alert ("DANGER");
  + Câu lệnh này nằm bên trong khối if đầu tiên, nên nó chỉ được thực thi khi điều kiện câu lệnh 1 đúng. Vì vậy, nó sẽ tự động được bao phủ nếu câu lệnh 1 được bao phủ.
* Câu lệnh 3: if ((speed > 100) and (load <= 50)) {
  + Câu lệnh này nằm bên trong khối if đầu tiên và có điều kiện riêng. Cần một trường hợp thỏa mãn điều kiện (speed > 100 và load <= 50) và một trường hợp không thỏa mãn điều kiện để bao phủ câu lệnh này.
* Câu lệnh 4: speed = 50;
  + Câu lệnh này nằm bên trong khối if thứ hai, nên nó chỉ được thực thi khi cả điều kiện câu lệnh 1 và điều kiện câu lệnh 3 đúng. Nó sẽ được bao phủ nếu các câu lệnh 1 và 3 được bao phủ.
* Câu lệnh 5: check = false;
  + Câu lệnh này nằm trong khối else của câu lệnh 3, nên nó chỉ được thực thi khi điều kiện câu lệnh 3 sai. Nó sẽ được bao phủ nếu câu lệnh 3 được bao phủ.

Các trường hợp kiểm thử:

1. Nhập giá trị cho temperature thỏa mãn (temperature < 0 hoặc temperature > 100), speed > 100 và load <= 50.
   * Điều này sẽ thực thi các câu lệnh 1, 2, 3 và 4.
2. Nhập giá trị cho temperature không thỏa mãn điều kiện (0 <= temperature <= 100), speed > 100 và load <= 50.
   * Điều này sẽ thực thi các câu lệnh 1, 3 và 5.
3. Nhập giá trị cho temperature thỏa mãn (temperature < 0 hoặc temperature > 100), speed <= 100 hoặc load > 50.
   * Điều này sẽ thực thi các câu lệnh 1 và 2.

Với 3 trường hợp kiểm thử này, tất cả các câu lệnh trong đoạn mã đều được thực thi ít nhất một lần, đảm bảo 100% Statement Coverage.

## Bài 6 (1 điểm)

Giảng viên cho thêm