

Chương 5

PHƯƠNG PHÁP KIỂM THỬ HỘP ĐEN

5.1. Khái niệm về kiểm thử hộp đen

5.1.1. Khái niệm

Kiểm thử hộp đen là xem chương trình như là một “hộp đen”. Mục đích của ta là hoàn toàn không quan tâm đến cách xử lý và cấu trúc bên trong của chương trình. Thay vào đó là tập trung vào tìm các trường hợp mà chương trình không thực hiện theo các đặc tả của nó.

Kiểm tra hộp đen, coi xử lý kết quả như là minh chứng bởi dữ liệu. Khái niệm kiểm tra hộp đen không quan tâm đến logic. Khuynh hướng này hiệu quả đối với các module chức năng đơn và các hệ thống cấp cao. Ba phương pháp chính là:

- *Phân hoạch cân bằng*: Mục đích của phương pháp này là tối thiểu các trường hợp kiểm tra cho trước, các mục dữ liệu vào được chia thành các nhóm dữ liệu có vai trò cân bằng nhau, mỗi nhóm đại diện cho một tập dữ liệu. Nguyên tắc là bằng cách kiểm tra triệt để một mục của mỗi tập hợp, chúng ta có thể chấp nhận tất cả các mục tương đương khác của tập hợp đó cũng sẽ được kiểm tra một cách kỹ càng.
- *Phân tích cực biên*: Phân tích giá trị cực biên là một dạng nghiêm ngặt của phân hoạch cân bằng có sử dụng các giá trị biên hơn là bất cứ giá trị nào trong tập cân bằng. Phân tích cực biên thường xuyên được dùng tại các mức module để xác định các mục dữ liệu đặc trưng cho testing.
- *Đoán lỗi*: Dựa trên cơ sở trực giác và kinh nghiệm, các chuyên gia có thể dễ dàng kiểm tra các điều kiện lỗi bằng cách đoán cái nào dễ xảy ra nhất. Và vì dựa trên trực giác, nên phép thử này khó tìm hết các lỗi.

Một nhược điểm của phân hoạch cân bằng và phân tích cực biên là tổ hợp của các miền hợp không được xác định. Để bổ sung, người ta dùng sơ đồ nguyên nhân – kết quả (Cause – Effect Graphing). Sơ đồ nguyên nhân – kết quả chỉ ra các đầu ra và thông tin di chuyển như là

hệ quả và các đầu vào gây ra hệ quả trên. Các ký hiệu graphic xác định tương tác, lựa chọn, logic và các điều kiện tương đương. Một vòng đại diện một dãy các thao tác không có điểm quyết định hoặc điều khiển. Mỗi dòng biểu diễn một lớp cân bằng của dữ liệu và các điều kiện sử dụng nó. Mỗi lần graph được thực hiện thì ít nhất một giá trị có nghĩa và một không có nghĩa cho mỗi tập được thử.

Theo hướng tiếp cận này, dữ liệu kiểm tra được lấy chỉ từ hồ sơ đặc tả của bài toán.

Các đặc trưng chính:

- Thực hiện các phép thử qua giao diện.
- Ít chú ý tới cấu trúc nội tại bên trong của chương trình.
- Nhằm thuyết minh các chức năng của phần mềm là đủ và vận hành đúng.

5.1.2. Mục đích

Kỹ thuật kiểm thử hộp đen nhằm tìm ra các lỗi sau:

- Chức năng thiếu hoặc không đúng đắn.
- Sai về giao diện.
- Sai thực thi chức năng.
- Sai khởi đầu hoặc kết thúc modul.

5.1.3. Ưu nhược điểm

Kiểm thử hộp đen không có mối liên quan nào tới mã lệnh, và kiểm thử viên chỉ rất đơn giản tâm niệm là: một mã lệnh phải có lỗi. Sử dụng nguyên tắc “Hãy đòi hỏi và bạn sẽ được nhận”, những kiểm thử viên hộp đen tìm ra lỗi mà những lập trình viên đã không tìm ra. Nhưng mặt khác, người ta cũng nói kiểm thử hộp đen “giống như là đi trong bóng tối mà không có đèn vậy”, bởi vì kiểm thử viên không biết các phần mềm được kiểm tra thực sự được xây dựng như thế nào. Đó là lý do mà có nhiều trường hợp mà một kiểm thử viên hộp đen viết rất nhiều ca kiểm thử để kiểm tra một thứ gì đó mà đáng lẽ có thể chỉ cần kiểm tra bằng 1 ca kiểm thử duy nhất, và/hoặc một số phần của chương trình không được kiểm tra chút nào.

Vì vậy, kiểm thử hộp đen có ưu điểm của “ một sự đánh giá khách quan”, mặt khác nó lại có nhược điểm là mang tính “thăm dò mù”.

5.2. Các phương pháp kiểm thử hộp đen

5.2.1. Phương pháp phân lớp tương đương

Phân lớp tương đương là 1 phương pháp kiểm thử hộp đen dựa trên nguyên tắc chia miền đầu vào của một chương trình thành các lớp dữ liệu, để từ đó lập ra các ca kiểm thử theo mỗi lớp đó.

Mục tiêu của phương pháp này là tìm ra 1 ca kiểm thử mà làm lộ ra 1 lớp lỗi. Vì vậy sẽ giảm tổng số các trường hợp kiểm thử phải được xây dựng.

Thiết kế ca kiểm thử cho phân lớp tương đương dựa trên sự đánh giá về các lớp tương đương với 1 điều kiện nào đó. Lớp tương đương biểu thị cho tập trạng thái hợp lệ (tập trạng thái mô tả các đầu vào hợp lệ của chương trình) hay không hợp lệ (tập trạng thái mô tả tất cả các trạng thái có thể khác của điều kiện) đối với các điều kiện vào. Chẳng hạn như thông tin về tên của một người Việt, lớp tương đương hợp lệ bao gồm tập các tên riêng tiếng việt hợp lệ có độ dài từ 1 đến 7 kí tự. Lớp tương đương không hợp lệ là tập các tên riêng không có trong tiếng Việt hoặc được viết bằng số hay kí tự đặc biệt.

Thiết kế trường hợp thử bằng lớp tương đương được tiến hành theo 2 bước:

- Xác định các lớp tương đương
- Xác định các ca kiểm thử

1. Xác định các lớp tương đương

Các lớp tương đương được xác định bằng cách lấy mỗi trạng thái đầu vào và phân chia nó thành 2 hay nhiều nhóm. Các nhóm này có thể là các lớp tương đương hợp lệ, các lớp tương đương không hợp lệ,... Với 1 đầu vào hay 1 điều kiện bên ngoài đã cho, việc xác định các lớp tương đương hầu như là một quy trình mang tính kinh nghiệm. Để xác định các lớp tương đương có thể áp dụng theo các nguyên tắc sau:

1. Nếu 1 trạng thái đầu vào định rõ giới hạn của các giá trị, xác định 1 lớp tương đương hợp lệ và 2 lớp tương đương không hợp lệ.

2. Nếu 1 trạng thái đầu vào xác định số giá trị, xác định 1 lớp tương đương hợp lệ và 2 lớp tương đương bất hợp lệ.
3. Nếu 1 trạng thái đầu vào chỉ định một tình huống “chắc chắn”, xác định 1 lớp tương đương hợp lệ và 1 lớp tương đương không hợp lệ.

Nếu chương trình xử lý các phần tử trong cùng một lớp là khác nhau thì phải chia lớp tương đương đó thành các lớp tương đương nhỏ hơn.

2. Xác định các ca kiểm thử

Với các lớp tương đương được xác định ở bước trên, bước tiếp theo là dựa vào các lớp tương đương đó để xác định các ca kiểm thử. Quá trình này được thực hiện như sau:

1. Gán 1 số duy nhất cho mỗi lớp tương đương.
2. Cho đến khi tất cả các lớp tương đương hợp lệ được bao phủ bởi các ca kiểm thử, viết 1 ca kiểm thử mới bao phủ càng nhiều các lớp tương đương đó chưa được bao phủ càng tốt.
3. Cho đến khi các ca kiểm thử của bạn đã bao phủ tất cả các lớp tương đương không hợp lệ, viết 1 ca kiểm thử bao gồm 1 và chỉ 1 trong các lớp tương đương không hợp lệ được bao phủ.

3. Ví dụ

Viết chương trình đọc vào 3 giá trị nguyên từ bàn phím. Ba giá trị này tương ứng với độ dài 3 cạnh của một tam giác. Chương trình hiển thị thông báo cho biết tam giác đó là tam giác đều, tam giác thường hay tam giác cân.

Phân tích bài toán: Ba giá trị nhập vào thỏa mãn là 3 cạnh của tam giác khi và chỉ khi cả 3 số đều là số nguyên dương, và tổng 2 số bất kì phải luôn lớn hơn số còn lại. Khi đó một tam giác đều là tam giác có 3 cạnh bằng nhau, tam giác cân là tam giác có 2 cạnh bằng nhau, tam giác thường là tam giác có 3 cạnh khác nhau.

Ta có mã lệnh của chương trình như sau:

Static void Main()

```
{
    short x,y,z;
    try
    {
        Console.WriteLine("nhap canh x= ");
        x = short.Parse(Console.ReadLine());
        Console.WriteLine("nhap canh y= ");
        y = short.Parse(Console.ReadLine());
        Console.WriteLine("nhap canh z= ");
        z = short.Parse(Console.ReadLine());
        ketqua(x,y,z)
    }
    catch
    {
        Console.WriteLine("Sai lỗi định dạng");
    }
}
```

Static void ketqua(short x, short y, short z)

```
{
    If (x+y<=z || x+z<=y || y+z<=x || x<=0 || y<=0 || z<=0)
        Console.WriteLine("Đây không phải là 3 cạnh của tam giác");
    Else
    {
        If (x==y && y==z)
            Console.WriteLine("Đây là tam giác đều");
        Else
        {
            If (x==y || x==z || y==z)
                Console.WriteLine("Đây là tam giác cân");
            Else
                Console.WriteLine("Đây là tam giác thường");
        }
    }
}
```

1. Xác định các lớp tương đương

Bảng 1: Bảng xác định các lớp tương đương

| Các giá trị đầu vào | Lớp tương đương hợp lệ | Lớp tương đương không hợp lệ |
|----------------------------------|---|---|
| Các giá trị dương | Cả 3 giá trị đều dương (1) | Tồn tại 1 trong 3 giá trị ≤ 0 (2) |
| Các giá trị là hằng số | Các giá trị nằm trong: -32768:32767 (3) | Các giá trị < -32768 (4), > 32767 (5) |
| Tổng 2 số bất kì so với số thứ 3 | Lớn hơn (6) | Nhỏ hơn hoặc bằng (7) |
| Các giá trị là nguyên | Cả 3 giá trị đều là số nguyên (8) | Tồn tại 1 trong 3 giá trị không phải là số nguyên (9) |
| Các giá trị nhập vào là số | Cả 3 giá trị đều là số (10) | Tồn tại 1 trong 3 giá trị không phải là số (11) |

2. Xác định các ca kiểm thử

- Các ca kiểm thử bao phủ các lớp tương đương hợp lệ là: các lớp (1), (3), (6), (8), (10).

Ví dụ: 3, 4, 5 và các hoán vị bao phủ các lớp trên.

- Các ca kiểm thử tương ứng với từng ca kiểm thử không hợp lệ:

(2) 2, 3, -1 và 2 hoán vị

(4) -32769, 1, 1 và 2 hoán vị

(5) 32768, 1, 1 và 2 hoán vị

(7) 1, 2, 5 và 5 hoán vị

(9) 1.2, 1, 2 và 2 hoán vị

(11) N, 1, 1 và 2 hoán vị

5.2.2. Phương pháp phân tích giá trị biên

Trong thực tế các lỗi thường có xu hướng xuất hiện tại biên của vùng hơn là tại các giá trị ở vùng trung tâm, có thể cả trong và ngoài biên. Do đó kỹ thuật kiểm thử giá trị biên được phát triển. Kỹ thuật này sẽ lựa chọn một số trường hợp kiểm thử tại các giá trị biên của dữ liệu. Nó được thực hiện theo phương châm:

- Nếu điều kiện đầu vào là một miền giá trị giới hạn bởi a và b thì cần thiết kế các ca kiểm thử cho cả a và b , và cả trên a , dưới b .
- Nếu điều kiện đầu vào xác định một phạm vi được chỉ định bởi tập hợp nhiều giá trị, lúc này các trường hợp kiểm thử sẽ được thiết kế tại các giá trị biên *min* và *max* của tập hợp đó. Ngoài ra các giá trị lớn hơn *max*, nhỏ hơn *min* cũng được kiểm thử.
- Hai phương châm trên cũng được áp dụng cho giá trị trả về.
- Nếu đầu vào hay đầu ra của chương trình là một tập hợp được sắp xếp có thứ tự (chẳng hạn như 1 file tuần tự) tập trung chú ý vào các phần tử đầu tiên và cuối cùng trong tập hợp đó.

Ví dụ: Xét ví dụ đã cho ở trên thì quá trình kiểm thử được thực hiện như sau:

Bảng 2: Các trường hợp kiểm thử theo phương pháp phân tích giá trị biên.

| Giá trị đầu vào kiểm thử | Giá trị mong muốn |
|------------------------------|---------------------------------------|
| 32767, 32767, 32767 | Đây là tam giác đều. |
| -32768, -32768, -32768 | Đây không phải là 3 cạnh của tam giác |
| 32768, 32768, 32768 | Sai lỗi định dạng. |
| -32769, -32769, -32769 | Sai lỗi định dạng. |
| 32767, 32767, 1 và 2 hoán vị | Đây là tam giác cân. |
| 32768, 32768, 1 và 2 hoán vị | Sai lỗi định dạng. |
| A, 1, 1 và 2 hoán vị | Sai lỗi định dạng. |
| 32768, 1, 1 và 2 hoán vị | Sai lỗi định dạng. |

| | |
|----------------------------------|--|
| -32768, 1, 1 và 2 hoán vị | Đây không phải là 3 cạnh của tam giác. |
| 1.1, 1, 1 và 2 hoán vị | Sai lỗi định dạng. |
| 32767, 32766, 32765 và 5 hoán vị | Đây là tam giác thường. |

5.2.3. Phương pháp sử dụng đồ thị nguyên nhân-kết quả

Một yếu điểm của phân tích giá trị biên và phân lớp tương đương là chúng không khảo sát sự kết hợp của các trường hợp đầu vào. Việc kiểm tra sự kết hợp đầu vào không phải là một nhiệm vụ đơn giản bởi vì nếu bạn phân lớp tương đương các trạng thái đầu vào, thì số lượng sự kết hợp thường là rất lớn. Nếu bạn không có cách lựa chọn có hệ thống một tập con các trạng thái đầu vào, có lẽ bạn sẽ chọn ra một tập tùy hứng các điều kiện, điều này có thể dẫn tới việc kiểm thử không có hiệu quả.

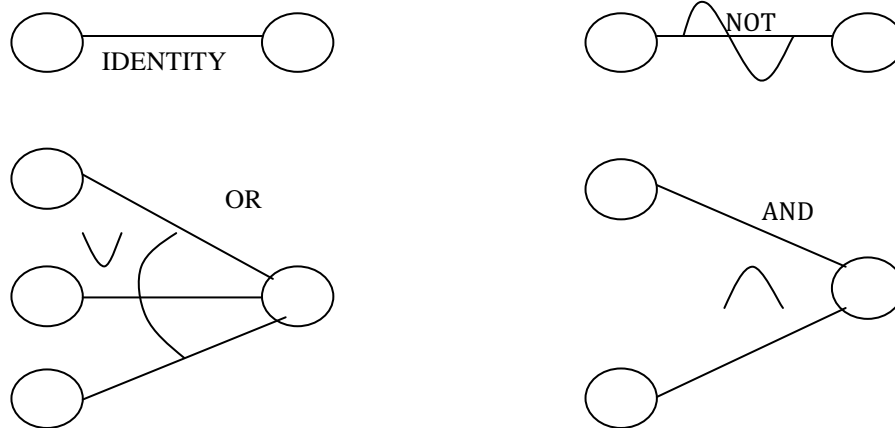
Đồ thị nguyên nhân-kết quả hỗ trợ cho việc lựa chọn một cách có hệ thống tập các ca kiểm thử có hiệu quả cao. Nó tác động tới việc chỉ ra các tình trạng chưa đầy đủ hoặc nhập nhằng trong đặc tả. Ngoài ra nó còn cung cấp cách biểu diễn chính xác cho các điều kiện logic và hành động tương ứng.

1. Các bước được sử dụng để xây dựng các test-case sử dụng đồ thị nguyên nhân – kết quả:

1. Đặc tả được chia thành các phần có thể thực hiện được. Điều này là cần thiết vì kỹ thuật này sẽ khó sử dụng khi được sử dụng trên những đặc tả quá lớn
2. Nguyên nhân và kết quả trong đặc tả được nhận biết. Một nguyên nhân là một trạng thái đầu vào nhất định hoặc cũng có thể là một lớp tương đương của các trạng thái đầu vào. Một kết quả là một trạng thái đầu ra hoặc một sự biến đổi hệ thống nào đó. Để nhận biết các nguyên nhân và kết quả bạn phải đọc phần đặc tả, sau đó gạch chân các từ hoặc cụm từ mô tả nguyên nhân và kết quả. Khi đó mỗi nguyên nhân và kết quả này sẽ được gán bởi một số duy nhất.
3. Xây dựng đồ thị nguyên nhân-kết quả bằng cách phát triển và biến đổi nội dung ngữ nghĩa của đặc tả thành đồ thị Boolean nối giữa nguyên nhân và kết quả.

4. Chuyển đồ thị thành một bảng quyết định mục vào giới hạn. Mỗi cột trong bảng mô tả một ca kiểm thử.
5. Các cột trong bảng quyết định được chuyển thành các ca kiểm thử.

2. Các ký hiệu cơ bản trong đồ thị nguyên nhân – kết quả



Hình 1. Các kí hiệu cơ bản trong đồ thị nguyên nhân-kết quả

Mỗi nút sẽ có giá trị 0 hoặc 1, trong đó 0 mô tả trạng thái vắng mặt còn 1 mô tả trạng thái có mặt.

Hàm **IDENTITY** (đồng nhất) nói:

- Nếu a là 1 thì b là 1
- Nếu a là 0 thì b là 0

Hàm **NOT** nói:

- Nếu a là 1 thì b là 0
- Nếu a là 0 thì b là 1

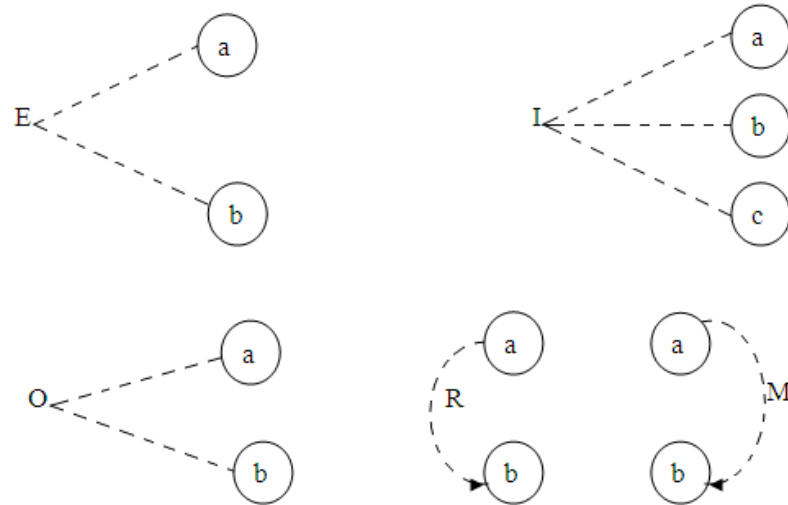
Hàm **OR** nói:

- Nếu a hoặc b hoặc c là 1 thì d là 1
- Nếu a hoặc b hoặc c là 0 thì d là 0

Hàm **AND** nói:

- Nếu cả a và b là 1 thì c là 1
- Nếu cả a và b là 0 thì c là 0

Trong hầu hết các chương trình sự kết hợp của một vài nguyên nhân nào đó là không thể vì lí do ngữ nghĩa và môi trường (chẳng hạn một kí tự thì không thể vừa là kí tự A vừa là kí tự B). Lúc này ta sử dụng các kí hiệu ràng buộc ở các hình sau:



Hình 2. Các kí hiệu ràng buộc trong đồ thị nguyên nhân-kết quả

Ràng buộc E (Exclude- ràng buộc loại trừ): khẳng định rằng tối đa chỉ có hoặc a hoặc b là 1 (a và b không đồng thời là 1).

Ràng buộc I (Include- ràng buộc bao hàm): khẳng định ít nhất 1 trong a,b hoặc c phải luôn luôn là 1 (a,b hoặc c không đồng thời là 0).

Ràng buộc O (Only- ràng buộc chỉ một): khẳng định rằng 1 và chỉ 1 a hoặc b là 1.

Ràng buộc R (Request- ràng buộc yêu cầu): khẳng định rằng khi a là 1 thì b phải là 1.

Ràng buộc M (Masks- ràng buộc mặt nạ): khẳng định nếu kết quả a là 1 thì kết quả b bắt buộc phải là 0.

Bước tiếp theo là tạo bảng quyết định mục vào giới hạn. Trong bảng này nguyên nhân chính là các điều kiện và kết quả chính là các hành động. Quá trình được thực hiện như sau:

1. Chọn 1 kết quả để làm trạng thái có mặt.
2. Lăn ngược trở lại đồ thị tìm tất cả các sự kết hợp của các nguyên nhân mà sẽ thiết lập kết quả này thành 1.
3. Tạo 1 cột trong bảng quyết định cho mỗi sự kết hợp nguyên nhân.

4. Với mỗi sự kết hợp quy định trạng thái của tất cả các kết quả khác và đặt chúng vào mỗi cột.

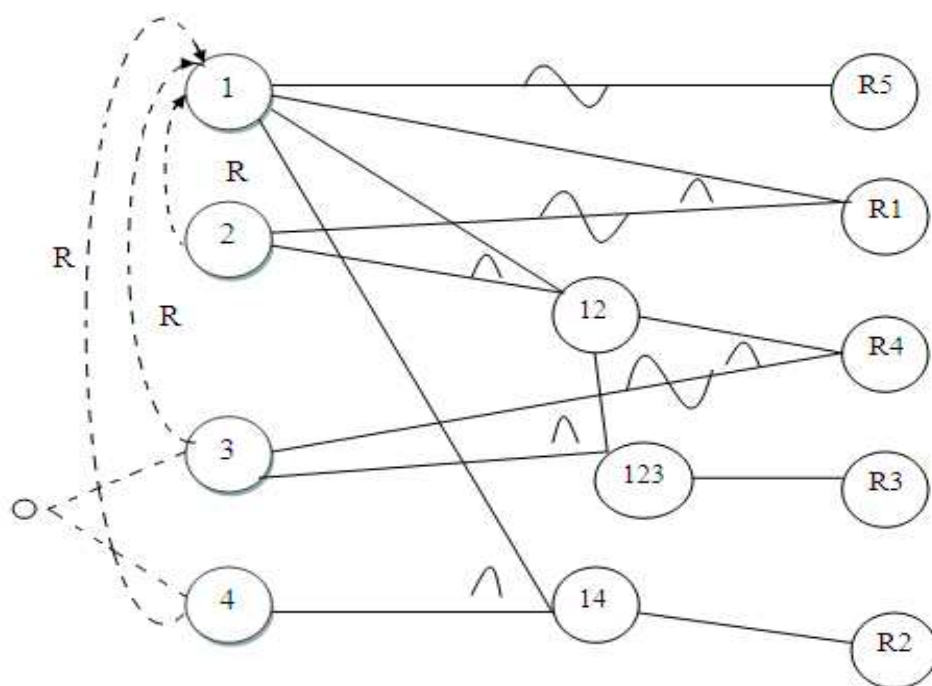
3. Ví dụ

Tạo các trường hợp thử cho ví dụ đã cho ở trên theo phương pháp này.

Từ đặc tả chương trình ta xác định được các nguyên nhân và kết quả sau:

- Nguyên nhân là:
 1. Cả 3 số nhập vào đều là số nguyên dương.
 2. Tổng 2 số bất kì trong 3 số luôn lớn hơn số còn lại.
 3. Hai trong 3 số có giá trị bằng nhau.
 4. Cả 3 số có giá trị bằng nhau.
- Kết quả là:
 - R1. Thông báo 3 giá trị nhập vào không tạo thành 1 tam giác.
 - R2. Thông báo 3 giá trị nhập vào tạo thành 1 tam giác đều.
 - R3. Thông báo 3 giá trị nhập vào tạo thành 1 tam giác cân.
 - R4. Thông báo 3 giá trị nhập vào tạo thành 1 tam giác thường.
 - R5. Thông báo lỗi nhập dữ liệu

Đồ thị nguyên nhân-kết quả được thể hiện như sau:



Bảng quyết định:

Ta chọn R1 là trạng thái có mặt đầu tiên. Khi đó:

- R1 có mặt nếu: Nút 1 = 1 và nút 2 = 0.
- R2 có mặt nếu: Nút 14 = 1 ,mà nút 14 = 1 khi nút 1 = 1 và nút 4 = 1.
- R3 có mặt nếu: Nút 123 = 1,mà nút 123 = 1 khi nút 12 = 1 và nút 3 = 1.Nút 12= 1 khi nút 1 = 1 và nút 2 = 1.
- R4 có mặt nếu: Nút 12 = 1 và nút 3 = 0. Nút 12 = 1 khi nút 1 =1 và nút 2 =1.
- R5 có mặt nếu: Nút 1 = 0.

Bảng 3: Bảng quyết định

| | 1 | 2 | 3 | 4 | 5 |
|----|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 |
| 2 | 0 | | 1 | 1 | |
| 3 | | | 1 | 0 | |
| 4 | | 1 | | | |
| R1 | 1 | 0 | 0 | 0 | 0 |
| R2 | 0 | 1 | 0 | 0 | 0 |
| R3 | 0 | 0 | 1 | 0 | 0 |
| R4 | 0 | 0 | 0 | 1 | 0 |
| R5 | 0 | 0 | 0 | 0 | 1 |

Các ca kiểm thử:

Bảng 4: Các trường hợp kiểm thử theo phương pháp đồ thị nguyên nhân- kết quả

| STT | Các điều kiện | Ca kiểm thử | Hành động |
|-----|---|--|-----------|
| 1 | Cả 3 giá trị nhập vào đều là 3 số nguyên dương và tồn tại 2 trong 3 số có tổng nhỏ hơn hoặc bằng số còn lại. | 1,2,5 và 5 hoán vị | R1 |
| 2 | Cả 3 số nhập vào đều là 3 số nguyên dương và cả 3 số đó có giá trị bằng nhau. | 2,2,2 | R2 |
| 3 | Cả 3 số nhập vào đều là số nguyên dương ,tổng 2 số bất kì trong 3 số luôn lớn hơn số còn lại và tồn tại 1 cặp 2 trong 3 số đó có giá trị bằng nhau. | 4,4,5 và 2 hoán vị | R3 |
| 4 | Cả 3 số nhập vào đều là số nguyên dương ,tổng 2 số bất kì trong 3 số luôn lớn hơn số còn lại và không có cặp 2 số bất kì nào trong 3 số có giá trị bằng nhau. | 3,4,5 và 5 hoán vị | R4 |
| 5 | Tồn tại 1 giá trị nhập vào không phải là số nguyên dương. | -1,2,3 N,1,2 1.2,3,4 và 5 hoán vị cho mỗi trường hợp. | R5 |

5.3. Bài toán áp dụng: “Kiểm thử cho hệ thống quản lý điểm sinh viên”

5.3.1. Hồ sơ đặc tả bài toán

Công tác quản lý điểm sinh viên của Khoa Công nghệ Thông tin được phát biểu như sau:

Khoa gồm 4 lớp cho mỗi khóa: A, B, C, D. Mỗi lớp có các thông tin: tên lớp, mỗi lớp có 1 mã lớp duy nhất để phân biệt với các lớp khác trong trường.

Mỗi sinh viên cần quản lý các thông tin như: họ tên, số CMND với họ tên là 1 chuỗi kí tự dạng chữ có độ dài ≤ 30 . Số CMND là 1 chuỗi kí tự dạng số có độ dài $= 9$. Mỗi sinh viên được cấp 1 mã số duy nhất để phân biệt với các sinh viên khác. Mỗi sinh viên chỉ được phân vào duy nhất 1 trong 4 lớp của khoa.

1. Chức năng phân lớp cho sinh viên và quản lý điểm cho sinh viên theo lớp học được thực hiện như sau:

- Nhập đầy đủ thông tin cho sinh viên cần phân lớp. Trong đó mã lớp không cần nhập vào mà được tạo ra theo quy tắc sau:
 - Đầu tiên ta sẽ lấy mã lớp của sinh viên cuối cùng trong danh sách chứa toàn bộ sinh viên. Sau đó lấy mã lớp này tăng thêm 1, kết quả này chính là mã lớp của sinh viên vừa nhập. Nếu kết quả trên lớn hơn mã lớp của lớp cuối cùng trong khoa thì lúc này mã lớp của sinh viên đó chính là mã lớp của lớp đầu tiên.
- Lưu các thông tin trên của sinh viên vào cơ sở dữ liệu.
- Quá trình trên cứ diễn ra liên tục cho đến khi hết sinh viên cần phân lớp.

Mỗi môn học có 1 tên gọi cụ thể, được học trong 1 số đơn vị học trình cụ thể. Ứng với mỗi môn học là 1 mã số duy nhất để phân biệt với các môn học khác.

Một sinh viên có thể học nhiều môn học khác nhau. Đối với mỗi môn học mỗi sinh viên sẽ có 2 cột điểm: điểm lần 1 và điểm lần 2. Nếu điểm lần 1 ≥ 5 thì điểm lần 2 sẽ có giá trị NULL. Ngược lại sẽ có cả 2 cột điểm.

Điểm của sinh viên cần quản lý các thông tin: điểm của môn học nào, của sinh viên nào, điểm lần 1, điểm lần 2.

Sau mỗi học kì dựa vào điểm của sinh viên cán bộ phòng đào tạo sẽ tính điểm trung bình cho các sinh viên. Công thức tính điểm trung bình cho mỗi học kì:

$$\text{ĐTB} = \text{tổng điểm} / \text{tổng số đvht}$$

Trong đó: tổng điểm = điểm lần 1 / điểm lần 2 * số đvht

Nếu điểm lần 1 ≥ 5 hoặc điểm lần 2 = Null thì sẽ tính theo điểm lần 1. Ngược lại sẽ tính theo điểm lần 2.

Sau khi tính điểm trung bình sẽ thực hiện xếp loại cho sinh viên với quy định:

- ĐTB ≥ 8 -> Xếp loại giỏi.
- ĐTB ≥ 7 -> Xếp loại khá.
- ĐTB ≥ 5 -> Xếp loại trung bình.
- Còn lại -> Xếp loại yếu.

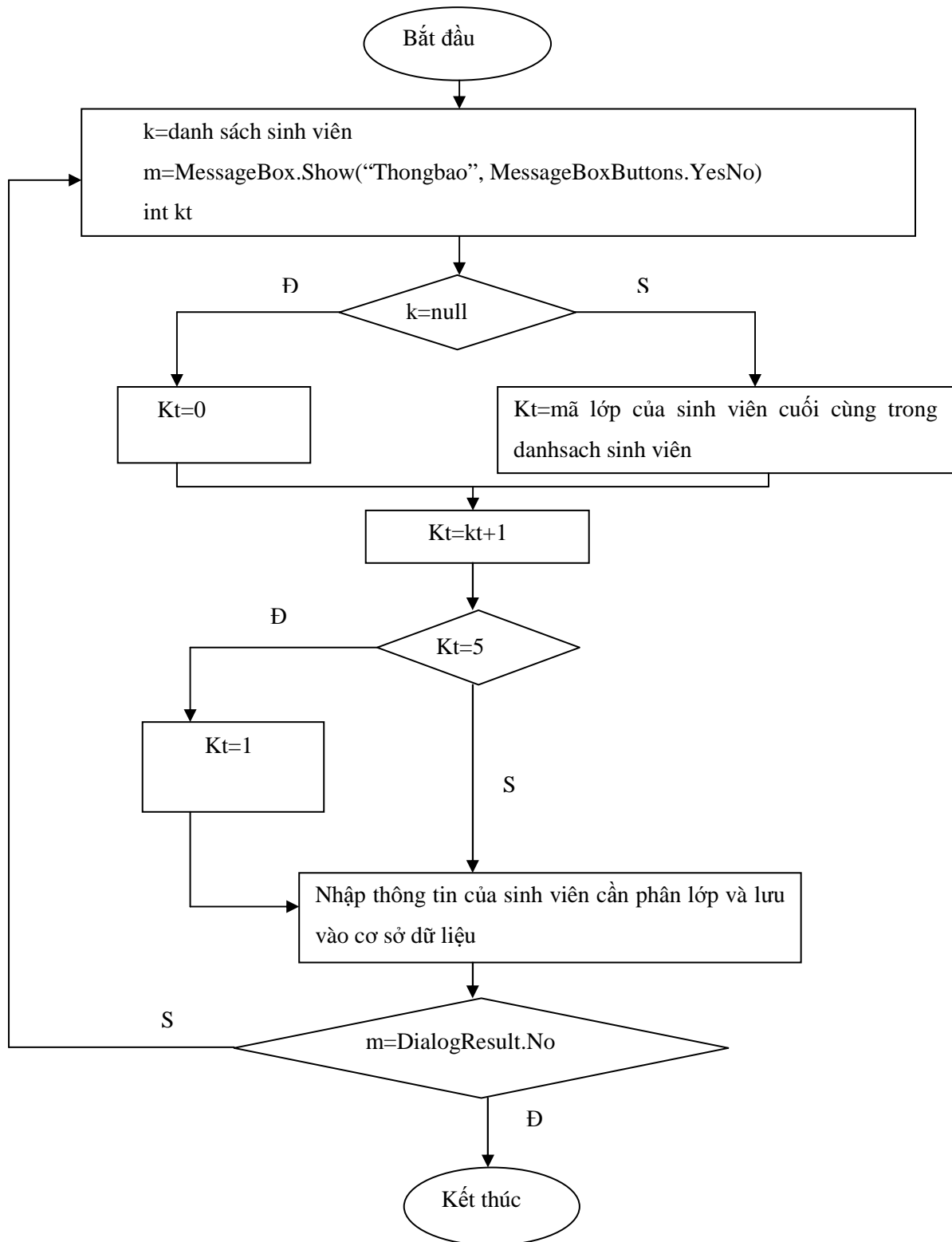
Với phát biểu trên ta cần xây dựng hệ thống có các chức năng sau:

- Phân lớp cho sinh viên.
 - Tính điểm trung bình.
 - Xếp loại cho sinh viên.
- Thiết kế chức năng: Phân lớp cho sinh viên

Đầu vào: thông tin của sinh viên muốn phân lớp. Trong đó chỉ cần nhập họ tên, số CMND.

Đầu ra: một danh sách các sinh viên đã được phân lớp.

Thuật toán thiết kế như sau:



Hình 3. Lưu đồ thuật toán chức năng phân lớp cho sinh viên

2. Chức năng tính điểm trung bình và xếp loại cho sinh viên

Đầu vào: một danh sách sinh viên cần tính điểm trung bình và xếp loại.

Đầu ra: một danh sách sinh viên đã được tính điểm trung bình và xếp loại.

Thuật toán:

Bước 1: đầu tiên khởi tạo 2 danh sách: ds và DanhSachDiemSV

Trong đó:

ds: chứa điểm của tất cả các sinh viên muốn xuất ra. Thông tin trong danh sách này gồm: hoten, điểm trung bình, xếp loại, mã sinh viên.

DanhSachDiemSV: danh sách này sẽ chứa điểm của tất cả các môn học của các sinh viên. Trong đó mỗi phần tử là điểm của tất cả các môn học của một sinh viên bất kì.

Bước 2: đối với mỗi phần tử trong “DanhSachDiemSV”:

- Tạo 1 đối tượng có kiểu tương ứng với kiểu của mỗi phần tử trong “ds”.
- Lưu: hoten, điểm trung bình, xếp loại, mã sinh viên vào đối tượng trên.
- Sau đó Add nó vào “ds”.

Bước 3: cứ lặp lại bước 2 cho đến khi hết “DanhSachDiemSV”.

Bước 4: dừng nếu “DanhSachDiemSV” đã hết phần tử. Lúc này “ds” chính là kết quả cần tìm.

5.3.2. Cài đặt bài toán

Chức năng phân lớp cho sinh viên

```
public static int kt;
private DialogResult m = new DialogResult();
private void btnnhapsinhvien_Click(object sender, EventArgs e)
{
    if (txttensinhvien.Text == "" || txtsocmnd.Text=="")
        MessageBox.Show("Phai nhap day du thong tin cho sinh vien ", "thong bao");
    else
    {
        if ((txttensinhvien.Text).Length > 30 || (txtsocmnd.Text).Length != 9)
```

```

        MessageBox.Show("Sai dinh dang","thong bao");
    else
    {
        do
        {
            Frmchinh.kt++;
            if (Frmchinh.kt == 5)
                Frmchinh.kt = 1;

                //sinh mã lớp cho sinh viên
            STUDENT std = new STUDENT();
            std.ClassId = Frmchinh.kt;

            //nhập thông tin cho sinh viên
            std.StudentName = txttensinhvien.Text;
            std.SoCMND = txtsocmnd.Text;

            // lưu sinh viên vào CSDL
            Frmchinh.db.STUDENTs.InsertOnSubmit(std);
            Frmchinh.db.SubmitChanges();

            m = MessageBox.Show("Nhap xong sinh vien " +
                                std.StudentName, "Nhap tiep
                                khong",MessageBoxButtons.YesNo);

            if (m == DialogResult.No)
                break;
        }

        while (m == DialogResult.No);
    }
}

```

Chức năng tính điểm trung bình và xếp loại cho sinh viên

```
private StudentDataContext db = new StudentDataContext();
private void Frmbangdiem_Load(object sender, EventArgs e)
{
    //danh sách bảng điểm của sinh viên
    List<BangDiemModel> ds = new List<BangDiemModel>();
    var danhSachDiem = db.MARKs.ToList();
    //danh sách tất cả điểm các môn học của 1 sinh viên
    List<IGrouping<int,MARK>>danhSachDiemSv=newList<IGrouping<int, MARK>>();
    //nhóm trong danhSachDiem theo mã sinh viên
    danhSachDiemSv=danhSachDiem.GroupBy(m=> m.StudentId).ToList();
    foreach (var item in danhSachDiemSv)
    {
        BangDiemModel model = new BangDiemModel();
        //tìm họ tên của sinh viên có mã SV là item.key
        model.Hoten = db.STUDENTS.Where(m =>m.Id==
            item.Key).FirstOrDefault().StudentName;
        float s = 0, Nums = 0;
        foreach (var diem in item)
        {
            if (diem.Mark1 >= 5 || diem.Mark2 == null)
            {
                //tính theo điểm lần 1 của môn đó
                s += diem.Mark1 * diem.SUBJECT.Num;
            }
            else
            {
                //tính theo điểm lần 2
                s += diem.Mark2.Value * diem.SUBJECT.Num;
```

```

    }
    //lưu số dvht của tất cả các môn học
        Nums += diem.SUBJECT.Num;
    }
    model.DiemTrungBinh = s / Nums;
    //xếp loại cho sinh viên
    if (model.DiemTrungBinh >= 8)
        model.XepLoai = "Gioi";
    else
        if (model.DiemTrungBinh >= 7)
            model.XepLoai = "Kha";
        else
            if (model.DiemTrungBinh >= 5)
                model.XepLoai = "Trung binh";
            else
                {
                    model.XepLoai = "Yeu";
                }
    model.MaSv = item.Key;
    ds.Add(model);
}
dataGridView1.DataSource = ds;
}

```

5.3.3. Thực hiện kiểm thử theo phương pháp hộp đen

5.3.3.1. Phân lớp tương đương

Thực hiện các ca kiểm thử cho chức năng: Phân lớp cho sinh viên.

1. Xác định các lớp tương đương.

Bảng 6: Bảng xác định các lớp tương đương.

| Giá trị kiểm thử | Lớp tương đương hợp lệ | Lớp tương đương không hợp lệ |
|------------------|---|---|
| Kí tự nhập vào | Họ tên sinh viên là 1 chuỗi các kí tự dạng chữ (1), và có độ dài thuộc [1,30]. (2) Số CMND là 1 chuỗi các kí tự dạng số (3), và có độ dài= 9.(4) | -Một trong các kí tự nhập vào trong chuỗi họ tên là kí tự dạng số. (5) -Họ tên là chuỗi có độ dài <1. (6) -Họ tên là chuỗi có độ dài > 30. (7) -Một trong các kí tự nhập vào trong chuỗi số CMND là kí tự dạng chữ. (8) -Số CMND là chuỗi có độ dài khác 9. (9) |

2. Các ca kiểm thử.

- Các ca kiểm thử hợp lệ là các ca kiểm thử bao phủ các lớp tương đương hợp lệ (1), (2), (3), (4).

Ví dụ:

Họ tên: Nguyễn Thị Ngân

Số CMND: 191655340

- Các ca kiểm thử không hợp lệ:
 - (5) Họ tên: Ngan9
 - (6) Họ tên là 1 chuỗi rỗng
 - (7) Họ tên: Nguyễn Nguyễn Tôn Tăng Tôn Nữ Ngọc Ngà.
 - (8) Số CMND: 19265534a
 - (9) Số CMND: 1234567899

5.3.3.2. Phân tích giá trị biên

Thực hiện các ca kiểm thử cho chức năng: Phân lớp cho sinh viên.

Bảng 7: Các trường hợp kiểm thử theo phương pháp phân tích giá trị biên

| Giá trị đầu vào kiểm thử | Kết quả mong muốn |
|--|---|
| Nhập vào chuỗi họ tên có đúng 30 kí tự dạng chữ. Số CMND là chuỗi có đúng 9 kí tự dạng số. | Lưu các thông tin đó vào cơ sở dữ liệu. Hiển thị thông báo: Nhập xong sinh viên. |
| Nhập vào chuỗi họ tên có đúng 30 kí tự dạng chữ. Số CMND là chuỗi các kí tự dạng số, và có độ dài khác 9. | Hiển thị thông báo: Sai định dạng |
| Nhập vào chuỗi họ tên có đúng 1 kí tự dạng chữ. Số CMND là chuỗi có đúng 9 kí tự dạng số. | Lưu các thông tin đó vào cơ sở dữ liệu. Hiển thị thông báo: Nhập xong sinh viên. |
| Nhập vào chuỗi họ tên có đúng 1 kí tự dạng chữ. Số CMND là chuỗi các kí tự dạng số, và có độ dài khác 9. | Hiển thị thông báo: Sai định dạng |
| Nhập vào chuỗi họ tên là chuỗi các kí tự dạng chữ, và có độ dài >30. Số CMND là chuỗi có đúng 9 kí tự dạng số. | Hiển thị thông báo: Sai định dạng |

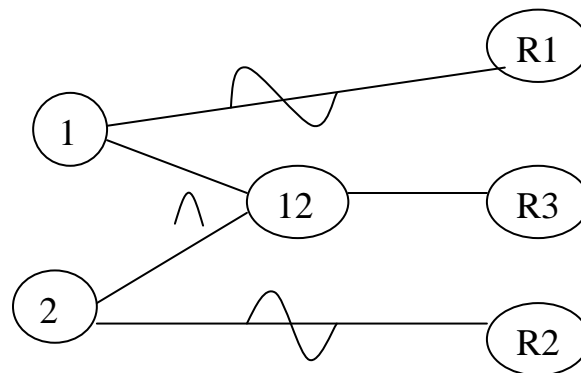
| | |
|---|---|
| Nhập vào chuỗi họ tên là chuỗi các kí tự dạng chữ, và có độ dài >30. Số CMND là chuỗi các kí tự dạng số, và có độ dài khác 9. | Hiển thị thông báo: Sai định dạng |
| Nhập vào chuỗi họ tên hoặc số CMND là 1 chuỗi rỗng. | Hiển thị thông báo: bạn cần nhập đầy đủ thông tin cho sinh viên này |

5.3.3.3. Đồ thị nguyên nhân –kết quả

Tạo các TeseCase cho chức năng: Phân lớp cho sinh viên.

Từ đặt tả chương trình ta xác định được các nguyên nhân và kết quả sau:

- Nguyên nhân là:
 1. Họ tên là chuỗi kí tự dạng chữ có độ dài khác rỗng. Số CMND là chuỗi kí tự dạng số có độ dài khác rỗng.
 2. Họ tên là chuỗi kí tự có độ dài ≤ 30 . Số CMND là chuỗi kí tự có độ dài $= 9$.
- Kết quả là:
 - R1. Thông báo: bạn cần nhập đầy đủ thông tin cho sinh viên này.
 - R2. Thông báo: Sai định dạng
 - R3. Thông tin nhập vào sẽ được lưu vào cơ sở dữ liệu và hiển thị thông báo: Nhập xong sinh viên.



Đồ thị nguyên nhân-kết quả

Bảng 8: Bảng quyết định

| | 1 | 2 | 3 |
|----|---|---|---|
| 1 | 0 | | 1 |
| 2 | | 0 | 1 |
| R1 | 1 | 0 | 0 |
| R2 | 0 | 1 | 0 |
| R3 | 0 | 0 | 1 |

Bảng 9: Các trường hợp kiểm thử theo phương pháp đồ thị nguyên nhân-kết quả

| STT | Các điều kiện | Cá kiểm thử | Hành động |
|-----|--|---|-----------|
| 1 | Họ tên là chuỗi kí tự dạng chữ có độ dài bằng rỗng. Hoặc số CMND là chuỗi kí tự dạng số có độ dài bằng rỗng. | Họ tên = "" Hoặc số CMND = "" | R1 |
| 2 | Họ tên là chuỗi kí tự có độ dài >30. Số CMND là chuỗi kí tự có độ dài khác 9. | Họ tên: Nguyễn Nguyễn Tôn Tăng Tôn Nữ Ngọc Ngà. Số CMND:1234 | R2 |
| 3 | Họ tên là chuỗi kí tự dạng chữ khác rỗng và có độ dài <=30. Số CMND là chuỗi kí tự dạng số và có độ dài= 9. | Họ tên: Nguyễn Thị Ngân. Số CMND: 191655340 | R3 |