Chương 5

PHƯƠNG PHÁP KIỂM THỬ HỘP TRẮNG

6.1. Khái niệm về kiểm thử hộp trắng

Kiểm thử hộp trắng là kĩ thuật kiểm thử trái ngược hoàn toàn với kĩ thuật kiểm thử hộp đen. Đây là phương pháp kiểm thử cho phép bạn khảo sát cấu trúc bên trong của chương trình. Kĩ thuật này xuất phát từ dữ liệu kiểm thử bằng sự kiểm thử tính logic của chương trình. Kiểm thử viên sẽ truy cập vào cấu trúc dữ liệu và giải thuật bên trong chương trình.

Mục tiêu kiểm thử hộp trắng

Các mục tiêu đặt ra đối với kĩ thuật kiểm thử này:

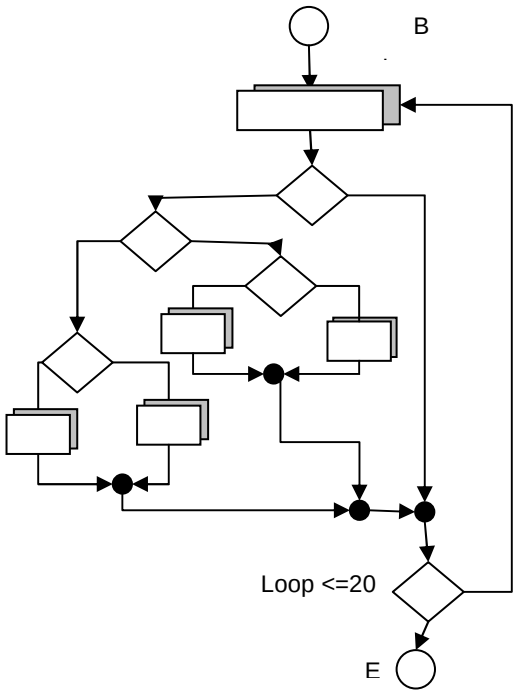
- Mọi đường độc lập trong một modul cần được thực hiện ít nhất một lần.
- Mọi ràng buộc logic cần được kiểm thử cả hai phía đúng và sai.
- Tất cả các vòng lặp ở biên của nó cũng được thực hiện.
- Mọi cấu trúc dữ liệu nội tại phải bảo đảm hiệu lực thi hành của nó.

Ưu và nhược điểm

Phương pháp này thường tốn nhiều thời gian và công sức nếu như mức độ kiểm thử được nâng lên ở cấp kiểm thử chức năng hay kiểm thử hệ thống.Vì vậy mà phương pháp này chủ yếu được dùng để kiểm thử đơn vị.

Kĩ thuật này kiểm tra trạng thái của chương trình tại nhiều điểm trong chương trình. Chính vì vậy đã giúp cho tính đúng đắn của chương trình được nâng cao.

Tuy nhiên việc kiểm định một cách triệt để tất cả các trường hợp là một bài toán quá lớn và tốn rất nhiều chi phi. Chúng ta hay xem xét ví dụ sau



Bên trái là thuật toán giả sử cho một chương trình được viết bằng khoản 100 dòng mã với một vòng lặp chính thực thi đoạn mã bên trong và lăp lại không quá 20 lần. Tuy nhiên khi tính toán cho thấy đối với chương trình này có đến khoảng 10^{14} đường có thể được thực hiện.

Chúng ta làm tiếp một phép tính nhanh để thấy được chi phí dùng để kiểm thử đoạn chương trình nay một cách thấu đáo và chi tiết. Ta giả sử rằng để kiểm định một trường hợp cần chạy trung bình tồn một giây. Và chương trình kiểm thử sẽ được chạy 24 giờ một ngày và chạy suốt 365 ngày một năm. Vây thì để chạy kiểm thử cho tất cả các trường hợp này cũng cần phải tốn khoản 3170 năm.

Do đó kiểm thử một cách triệt để như trên là một việc bất khả thi cho những hệ thống lớn. Mặc dù kỹ thuật này không thể hiện thực được trong thực tế với lượng tài nguyên có hạn, tuy nhiên với một số lượng có giới hạn các đường diễn tiến logic quan trọng có chọn lựa trước để kiểm thử. Phương pháp này có thể là rất khả thi

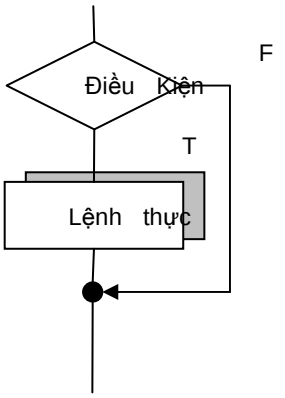
Ngoài ra các trường hợp kiểm thử còn có thể là sự kết hợp của cả hai kỹ thuật “đen và trắng” nhằm đạt được các mục tiêu của việc kiểm thử.

6.2. Các phương pháp kiểm thử hộp trắng

6.2.1. Kiểm thử đường thi hành cơ bản

1. Khái niệm

Khái niệm một đường thi hành cơ bản (đường diễn tiến) của chương trình là một tập hợp lệnh được thực thi có thứ tự.trong chương trình. Để đơn giản hơn có thể hiểu một đoạn chương trình hay một chương trình chứa rất nhiều các đường diễn tiến tại một lênh điều kiện rẽ nhánh tạo ra một tập đường mới



Đường thi hành cơ bản

Vi dụ trên ta sẽ có 2 đường một đường khi điều kiện A nhận giá trị đúng và một đường khi điều kiện A mang giá trị sai.

=> Kiểm thử đường thi hành cơ bản là quá trình thiết kế các trường hợp kiểm thử trên từng câu lệnh của chương trình, sao cho từng câu lệnh được thực hiện ít nhất một lần.

Mục tiêu

Bảo đảm mọi đường thi hành của đơn vị phần mềm cần kiểm thử đều chạy đúng. Tuy nhiên trên thực tế để đạt được mục tiêu trên thì rất khó, vì công sức và thời gian phải bỏ ra là rất lớn, ngay cả đối với các đơn vị phần mềm nhỏ.

2. Các bước để xây dựng tập hợp các trường hợp kiểm thử

Bước 1: Xác định các nút. Nút là các câu lệnh tuần tự, điểm kết thúc của một vòng lặp, điểm kết thúc của một hàm.

Bước 2: Vẽ đồ thị thể hiện đường diễn tiến của chương trình. Từ các nút đã xác định ở bước 1, xây dựng đồ thị dòng điều khiển tương ứng.

Bước 3: Xác định số đường kiểm thử V(G) và chỉ rõ các đường tương ứng

Số đường kiểm thử V(G)= E - N + 2

E: số cạnh

N: Số nút

=> Xác định các đường kiểm thử: tập các đường độc lập tuyến tính

Bước 4: Dựa vào các trường hợp kiểm thử xác định các TestCase tương ứng

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.

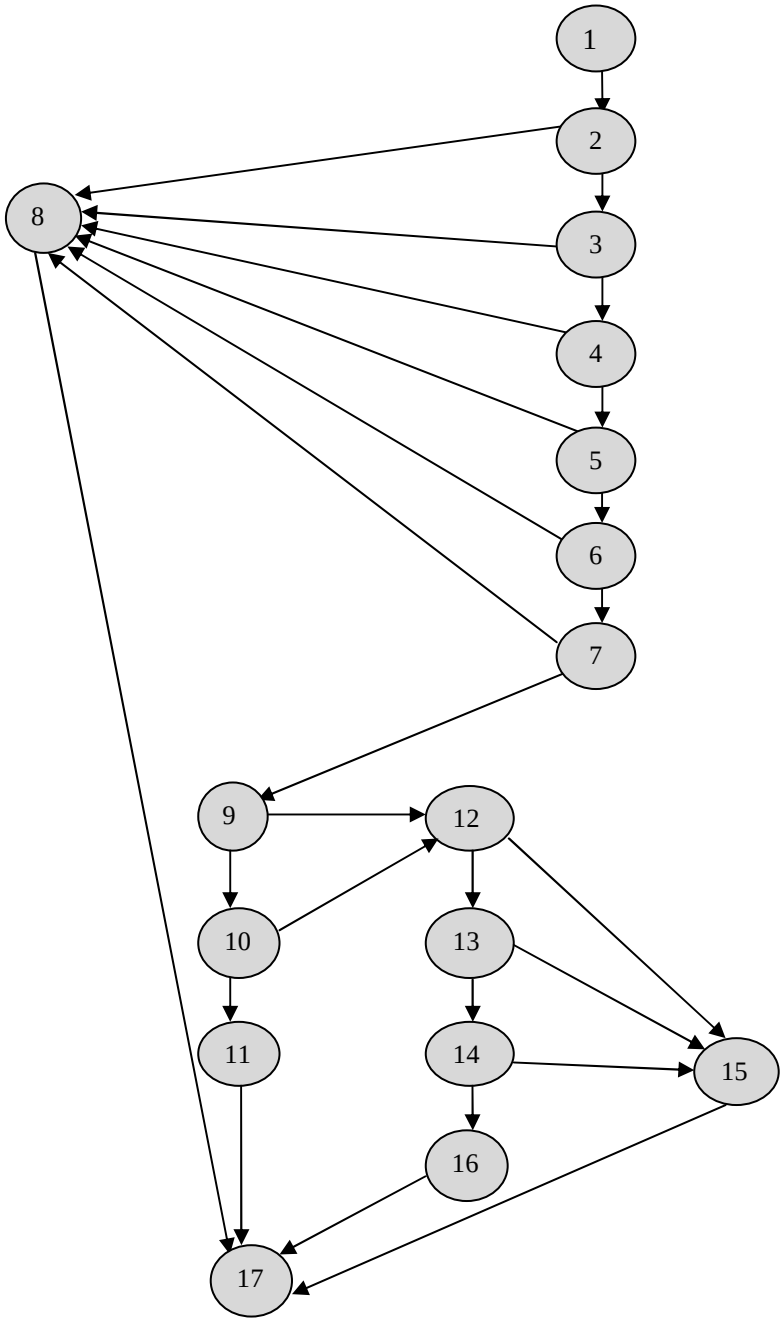
```
Static void ketqua(short  x, short  y, short z)
{
    If (x+y<=z || x+z<=y || y+z<=x || x<=0 || y<=0 ||   z<=0)
        Console.WriteLine("Đây không phải là 3 cạnh của tam giác");
    Else
     {
        If (x==y && y==z)
            Console .WriteLine("Đây là tam giác đều");
        Else
      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

Ta thiết kế các trường hợp kiểm thử cho hàm ketqua:

- Xác định các nút

```
Static void ketqua(short  x, short  y, short z)
{ 1
    2     3     4     5     6     7
  If (x+y<=z || x+z<=y || y+z<=x || x<=0 || y<=0 ||   z<=0)
        8
      Console.WriteLine("Đây không phải là 3 cạnh của tam giác");
  Else
  {
        9     10
      If (x==y && y==z)
          Console .WriteLine("Đây là tam giác đ 11
       Else
       {      12    13    14
          If (x==y || x==z || y==z)
              Console .WriteLine("Đây là tam giác cân"); 15
          Else
              Console .WriteLine("Đây là tam giác thường"); 16
       }
  }
}   17
```

- Vẽ đồ thị thể hiện đường diễn tiến của chương trình



<u>Hình 1.</u> Đồ thị đường diễn tiến

- Xác định số đường kiểm thử V(G): V(G)= 27-17+2= 12

Các đường kiểm thử là:

1. 1. 2. 8. 17
2. 1. 2. 3. 8. 17
3. 1. 2. 3. 4. 8. 17
4. 1. 2. 3. 4. 5. 8. 17
5. 1. 2. 3. 4. 5. 6. 8. 17
6. 1. 2. 3. 4. 5. 6. 7. 8. 17
7. 1. 2. 3. 4. 5. 6. 7. 9. 10. 11. 17
8. 1. 2. 3. 4. 5. 6. 7. 9. 10. 12. 15. 17
9. 1. 2. 3. 4. 5. 6. 7. 9. 12. 15. 17
10. 1. 2. 3. 4. 5. 6. 7. 9. 12. 13. 15. 17
11. 1.2.3.4.5.6.7.9.12.13.14.15.17
12. 1. 2. 3. 4. 5. 6. 7. 9. 12. 13. 14. 16. 17

- Xác định các Test Case:

Bảng 5: Các trường hợp kiểm thử theo phương pháp kiểm thử đường thi hành cơ bản

Đường kiểm thử	Giá trị đầu vào	Kết quả mong đợi
1	x=1, y=1,z=3	Đây không phải là 3 cạnh của tam giác
2	x=1, y=4, x=2	Đây không phải là 3 cạnh của tam giác
3	x=5, y=1, z=3	Đây không phải là 3 cạnh của tam giác
4	x=-1, y=1, z=1	Đây không phải là 3 cạnh của tam giác
5	x=1, y=-1, z=1	Đây không phải là 3 cạnh của tam giác
6	x=1, y=1, z=-1	Đây không phải là 3 cạnh của tam giác
7	x=4, y=4, z=4	Đây là tam giác đều
8	x=3, y=3, z=4	Đây là tam giác cân
9	x= 3, y=4, z=4	Đây là tam giác cân
10	x= 3, y=4, z=3	Đây là tam giác cân
11	x= 3, y=4, z=4	Đây là tam giác cân
12	x=2, y = 3, z=4	Đây là tam giác thường

6.2.2. Kiểm thử vòng lặp

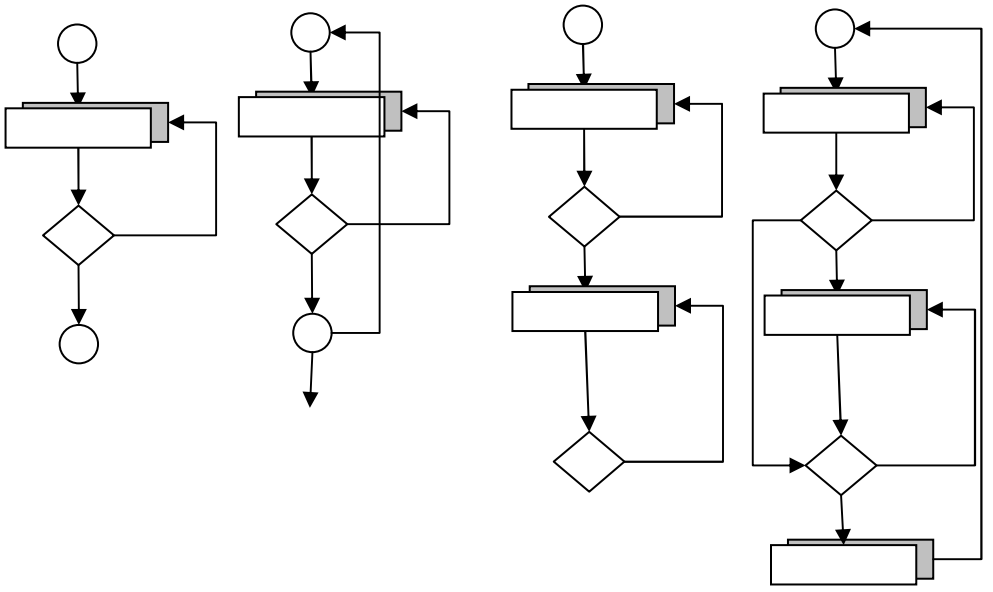
1. Khái niệm

Là kĩ thuật kiểm thử tập trung vào tính hợp lệ của các cấu trúc vòng lặp.

2. Các loại vòng lặp cơ bản

Có 4 loại vòng lặp cơ bản sau:

Các Cấu Trúc Lặp



vòng lặp đơn giản | vòng lặp lồng nhau | vòng lặp móc nối liền kề | vòng lặp không cầu trúc

1. Vòng lặp đơn: là loại vòng lặp mà thân của nó chỉ chứa các lệnh khác. Khi kiểm thử cho loại vòng lặp này ta nên chọn các test-case sau đây để kiểm thử lệnh lặp n lần (n là maximum số lần lặp). Ta bỏ tình toàn vẹn của vòng lặp và xét duyệt:

- Chạy 0 lần
- Chạy 1 lần
- Chạy k lần với k là một giá trị bất kì thỏa 2<k<n-1

- Chạy n-1 lần
- Chạy n lần
- Chạy n +1 lần

2. Kiểm thử vòng lặp lồng nhau: là vòng lặp mà thân của nó chứa thêm lệnh lặp khác. Đối với vòng lặp này ta kiểm thử tuần tự từng vòng lặp từ trong ra ngoài, với cách thực hiện như sau:

- Bắt đầu kiểm thử với vòng lặp trong cùng, trong khi đó cho các vòng lặp ngoài chạy với giá trị min.
- Lặp lại bước trên để tiến dần ra các vòng lặp bên ngoài, cho đến khi tất cả các vòng lặp đều được kiểm thử.

3. Kiểm thử vòng lặp liền kề: là vòng lặp có 2 hay nhiều lệnh lặp kế tiếp nhau. Đối với vòng lặp này có thể kiểm thử như khi kiểm thử vòng lặp đơn nếu các biến lặp độc lập với nhau. Tuy nhiên khi các biến lặp không độc lập với nhau (biến lặp của vòng lặp thứ nhất được sử dụng như là biến khởi tạo trong vòng lặp thứ hai) thì ta nên tiến hành kiểm thử như khi kiểm thử vòng lặp lồng nhau.

4. Kiểm thử vòng lặp không có cấu trúc: vì đây là loại vòng lặp phức tạp nên khi gặp vòng lặp này ta nên thiết kế lại đoạn code sao cho nó không chứa các vòng lặp dạng này.

6.2.2. Kiểm thử cấu trúc điều kiện

1. Khái niệm

Kiểm thử cấu trúc điều kiện là phương pháp kiểm thử dựa trên những điều kiện logic của hàm hay module.

Một điều kiện đơn giản là một biến Boolean hoặc là một biểu thức quan hệ:

- X hay Not X một điều kiện logic đơn giản.
- Biểu thức quan hệ thường có dạng : E1 <phép toán quan hệ> E2

Trong đó, E1, E2 là các biểu thức số học và phép toán quan hệ là một trong các phép toán sau : <, <=, ==, != , > hay >=. Một điều kiện kết hợp của 2 hay nhiều điều kiện đơn giản, các phép toán boolean : OR (| |, AND (&) and NOT (!)

Các loại lỗi của điều kiện bao gồm

Lỗi trong các thao tác luận lý (lỗi tồn tại một biểu thức không đúng, thiếu hoặc thừa các thao tác luận lý)

- Lỗi do giá trị của biến luận lý
- Lỗi do dấu ngoặc
- Lỗi do phép toán quan hệ
- Lỗi trong biểu thức toán học

Mục đích của kiểm thử cấu trúc điều kiển là phát hiện không chỉ lỗi trong điều kiện mà còn những lỗi khác trong chương trình. Nếu một tập kiểm thử cho một chương trình P là hiệu quả cho việc phát hiện lỗi trong điều kiện của P,thì bộ kiểm thử đò cũng có thể phát hiện các lỗi khác trong P.

E1 <phép toán quan hệ> E2

Ba trường hợp kiểm thử được yêu cầu để kiểm tra là giá trị E1 lớn hơn, nhỏ hơn và bằng giá trị của E2. Nếu <phép toán quan hệ> là không đúng và E1, E2 là đúng thì 3 loại kiểm thử trên có đảm bảo có thể xác định được lỗi trong phép toán quan hệ. Để phát hiện lỗi trong E1và E2 thì các trường hợp kiểm thử E1 lớn hơn, nhỏ hơn E2 có thể phát hiện ra được lỗi.

Một biểu thức có n biến, thì có 2^n khả năng kiểm thử xãy ra khi (n>0)

2. Các tiêu chuẩn trong loại hình kiểm thử này

1. Bao phủ câu lệnh

Quá trình kiểm thử được thiết kế sao cho mọi câu lệnh trong chương trình được thực hiện ít nhất 1 lần. Phương pháp này xuất phát từ ý tưởng:

- Trừ phi 1 câu lệnh được thực hiện, nếu không ta không thể biết được câu lệnh đó có lỗi hay không.
- Nhưng với việc kiểm tra 1 giá trị đầu vào thì sẽ không đảm bảo câu lệnh đó sẽ đúng cho mọi trường hợp.

2. Bao phủ quyết định

Là quá trình kiểm thử sao cho mỗi hướng phân nhánh phải được kiểm thử ít nhất 1 lần.

Bao phủ quyết định thường thỏa mãn bao phủ câu lệnh.Vì mỗi câu lệnh thường bắt nguồn từ 1 đường đi phụ thuộc nào đó hoặc là từ 1 câu lệnh rẽ nhánh hoặc từ điểm vào của chương trình. Mỗi câu lệnh được thực hiện nếu mỗi câu lệnh rẽ nhánh được thực hiện. Do đó mà bao phủ quyết định mỗi quyết định phải có kết luận đúng hoặc sai, và mỗi câu lệnh phải được thực hiện ít nhất 1 lần.

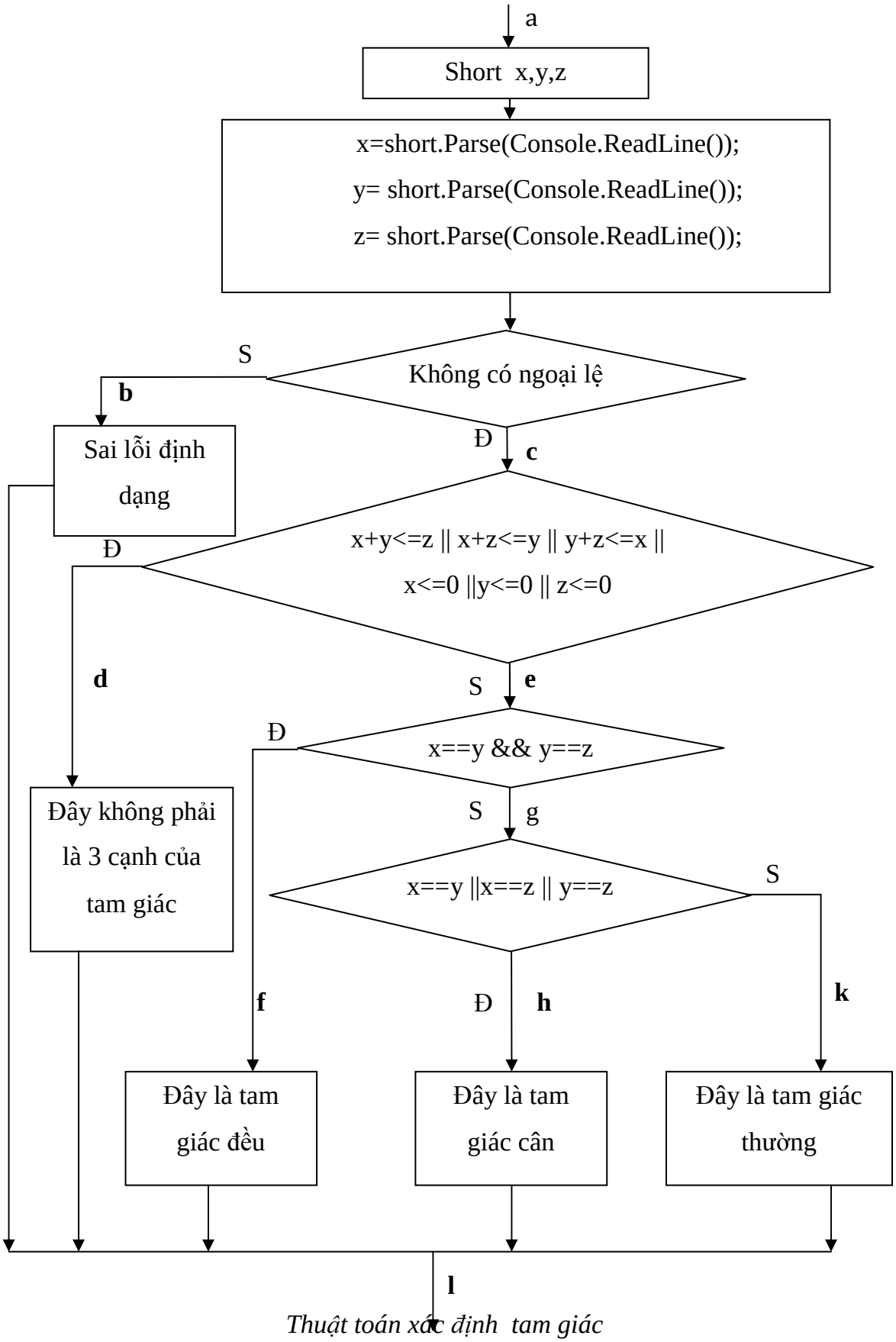
3. Bao phủ điều kiện

Là quá trình kiểm thử các biểu thức điều kiện trên 2 giá trị True và False. Một điểm cần chú ý khi kiểm thử điều kiện là cần xem xét kết hợp các điều kiện với nhau.

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.

Lưu đồ thuật toán cho chương trình trên như sau:



Thuật toán xác định tam giác

1. Bao phủ quyết định: ta có các trường hợp kiểm thử sau:
    1. 1.1, 1, 2 và 5 hoán vị của nó (abl)
    2. -1, 2, 3 và 5 hoán vị của nó (acdl)
    3. 32767, 32767, 32767 (acefl)
    4. 32767, 32767, 32766 và 2 hoán vị của nó (aceghl)
    5. 32767, 32765, 32766 và 5 hoán vị của nó (acegkl)
2. Bao phủ điều kiện: ta có các trường hợp kiểm thử sau:
    1. 1.2, 2, 3 và 5 hoán vị của nó (abl)
    2. 1, 2, 9 và 5 hoán vị của nó (acdl)
    3. 9, 9, 9 (acefl)
    4. 5, 5, 8 và 2 hoán vị của nó (aceghl)
    5. 6, 7, 8 và 5 hoán vị của nó (acegkl)

6.3. Kiểm thử cho "Bài toán quản lí điểm sinh viên"

Đặc tả bài toán

Công tác quản lí điểm sinh viên của khoa công nghệ thông tin được phát biểu như sau:

Khoa gồm 4 lớp : 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 lí 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 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 lí 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:

ĐTB= tổng điểm / 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:

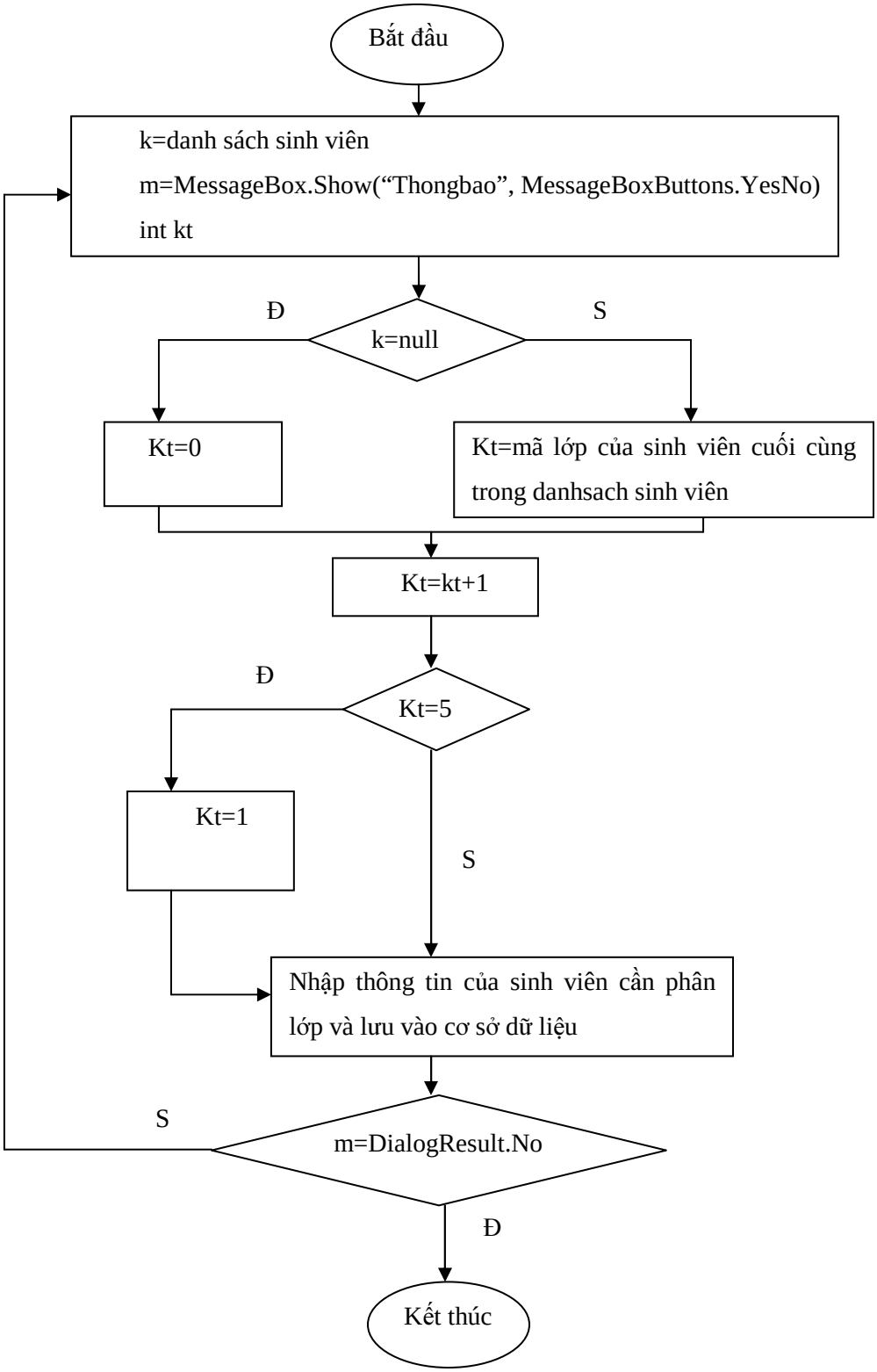
- Phân lớp cho sinh viên.
- Tính điểm trung bình.
- Xếp loại cho sinh viên.

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:



<u>Hình 2.</u> 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.

Cài đặt

Cài đặt 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("ban can 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);

              }
          }
   }
```

Cài đặt cho 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=new          List<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)
                      {
                //tinh 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;
    }
```

Phương pháp kiểm thử hộp trắng

1. Kiểm thử đường thi hành cơ bản

Thiết kế các trường hợp kiểm thử cho chức năng: Phân lớp cho sinh viên.

- Xác định các nút:

public static int kt;

private DialogResult m= new DialogResult();

private void btnnhapsinhvien_Click(object sender, EventArgs e)

{ (1)

(2) (3)

if (txttensinhvien.Text == "" || txtsocmnd.Text == "")

MessageBox.Show("bạn cần nhập đầy đủ thông tin cho sinh viên này","thong bao");

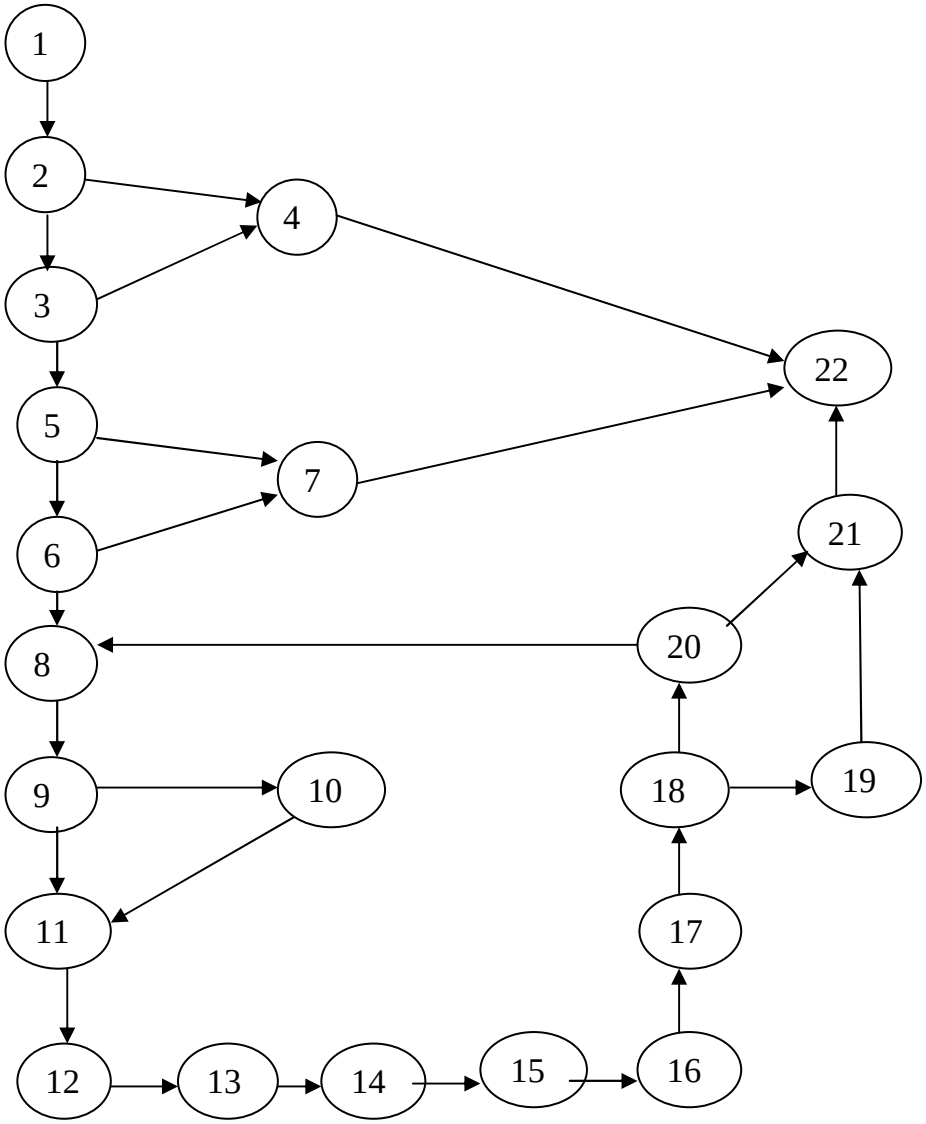
(4)

```
else
 {                    5                              6
    if ((txttensinhvien.Text).Length > 30 || (txtsocmnd.Text).Length != 9)
                  7  MessageBox.Show(" Sai định dạng ", "thong bao");
    else
        {
          do
          {
         //Sinh mã lớp cho sinh viên
            Frmchinh.kt++;       8
                        9
            if (Frmchinh.kt == 5)
                    Frmchinh.kt = 1;       10
            STUDENT std = new STUDENT();            11
         //nhập thông tin cho sinh viên
              std.ClassId = Frmchinh.kt;          12
            std.StudentName = txttensinhvien.Text;          13
            std.SoCMND=txtsocmnd.Text;          14
         // lưu sv vào csdl
            Frmchinh.db.STUDENTs.InsertOnSubmit(std);          15
            Frmchinh.db.SubmitChanges();          16
              m=MessageBox.Show("Nhập xong sinh  viên"+std.StudentName,"Nhap tiep
                                  17          khong", MessageBoxButtons.YesNo);
                          18
             if (m == DialogResult.No)
                        break;          19
            }
                    20
          while (m == DialogResult.No);
         }     21
```

}

} 22

- Vẽ đồ thị thể hiện đường diễn tiến của chương trình:



<u>Hình 3.</u> Đồ thị đường diễn tiến

- Xác định số đường kiểm thử V(G):

V(G)=28-22+2=8

Các đường kiểm thử là:

1. 1. 2. 4. 22

2. 1. 2. 3. 4. 22
3. 1. 2. 3. 5. 7. 22
4. 1. 2. 3. 5. 6. 7. 22
5. 1. 2. 3. 5. 6. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 21. 22
6. 1. 2. 3. 5. 6. 8. 9. 11. 12. 13. 14. 15. 16. 17. 18. 19. 21. 22
7. 1. 2. 3. 5. 6. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 20. 8. 9…
8. 1. 2. 3. 5. 6. 8. 9. 11. 12. 13. 14. 15. 16. 17. 18. 20. 8. 9…

Ba chấm sau những đường 7, 8 cho biết rằng 1 đường đi bất kì qua phần còn lại của cấu trúc điều khiển đều chấp nhận được.

- Xác định các Test Case:

Bảng 10: Các trường hợp kiểm thử theo phương pháp kiểm thử đường thi hành cơ bản

Đường kiểm thử	Giá trị đầu vào	Kết quả mong đợi
1	-họ tên là 1 chuỗi rỗng. -số CMND là chuỗi kí tự có độ dài =9.	Hiển thị thông báo: bạn cần nhập đầy đủ thông tin cho sinh viên này
2	-họ tên là 1 chuỗi có độ dài<=30. -số CMND là 1 chuỗi rỗng	Hiển thị thông báo: bạn cần nhập đầy đủ thông tin cho sinh viên này
3	-họ tên là 1 chuỗi khác rỗng, có độ dài >30. -số CMND là chuỗi có 9 kí tự.	Hiển thị thông báo: Sai định dạng
4	- họ tên là 1 chuỗi khác rỗng và có độ dài <= 30. -số CMND là chuỗi có độ dài khác 9	Hiển thị thông báo: Sai định dạng
5	-kt=5 -họ tên là 1 chuỗi khác rỗng và có độ dài <=30. -số CMND là chuỗi kí tự dạng số, có độ dài =9. -m= DialogResult.No	Lưu sinh viên vừa nhập vào cơ sở dữ liệu. Hiển thị thông báo: Nhập xong sinh viên.

6	-kt!=5 -họ tên là 1 chuỗi khác rỗng và có độ dài <=30. -số CMND là chuỗi kí tự dạng số, có độ dài =9. -m= DialogResult.No	Lưu sinh viên vừa nhập vào cơ sở dữ liệu. Hiển thị thông báo: Nhập xong sinh viên.
7	-kt=5 -họ tên là 1 chuỗi khác rỗng và có độ dài <= 30. -số CMND là chuỗi kí tự dạng số, có độ dài =9. -m= DialogResult.Yes	Lưu sinh viên vừa nhập vào cơ sở dữ liệu. Hiển thị thông báo: Nhập xong sinh viên.
8	-kt!=5 -họ tên là 1 chuỗi khác rỗng và có độ dài <= 30. -số CMND là chuỗi kí tự dạng số, có độ dài = 9. -m= DialogResult.Yes	Lưu sinh viên vừa nhập vào cơ sở dữ liệu. Hiển thị thông báo: Nhập xong sinh viên.

2. Kiểm thử vòng lặp

Các trường hợp kiểm thử cho chức năng : tính điểm trung bình và xếp loại cho sinh viên.

Xét vòng lặp ngoài ta có các trường hợp kiểm thử sau:

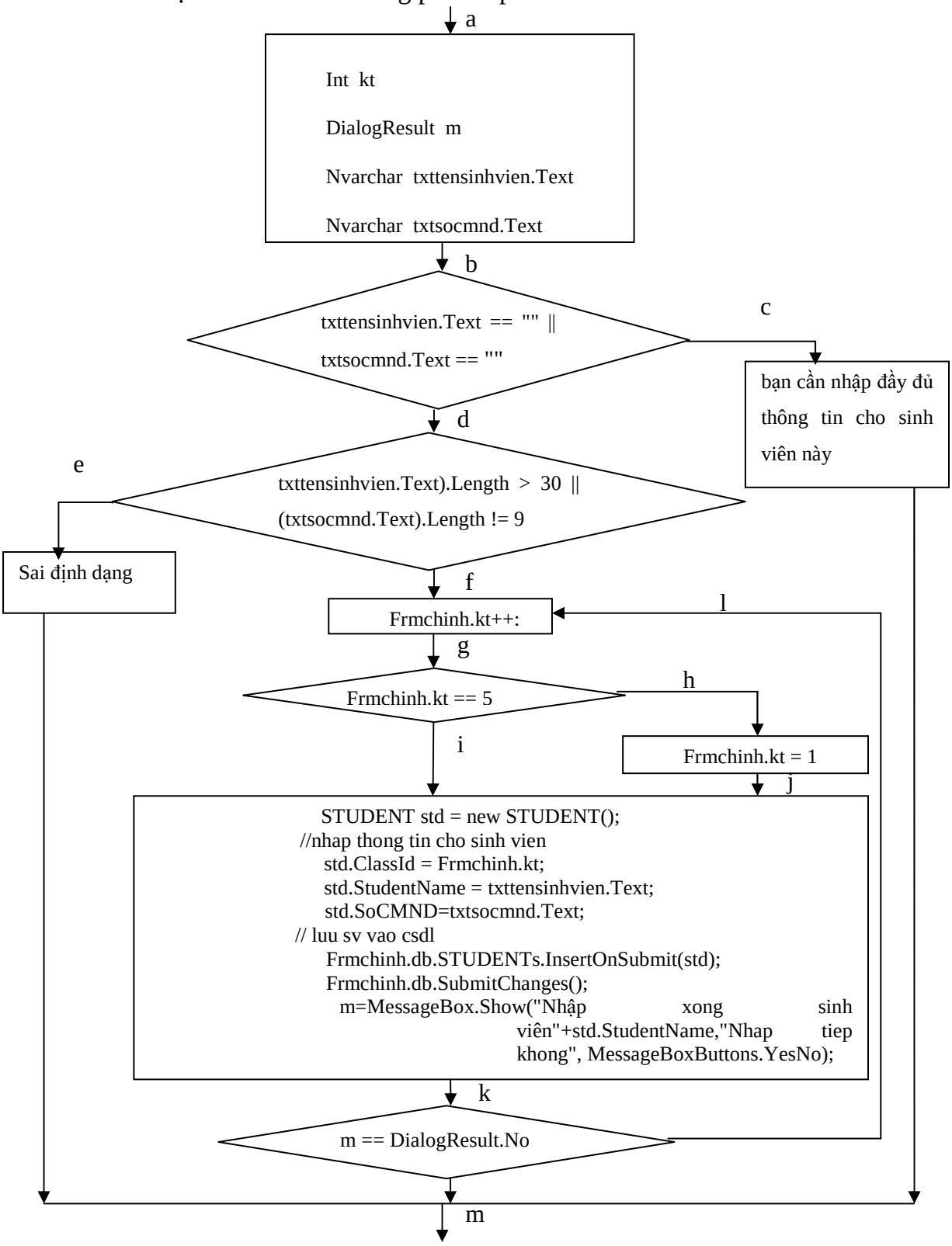
danhSachDiemSv: rỗng.

danhSachDiemSv có số phần tử > 0.

Ở đây “Item” là 1 phần tử trong “danhSachDiemSv”, do đó nếu “danhSachDiemSv ” là 1 danh sách có số phần tử >0 thì “Item” luôn khác rỗng,ngược lại “danhSachDiemSv” là 1 danh sách rỗng thì “Item” sẽ rỗng .Vì vậy đối với chương trình này ta chỉ kiểm thử đối với danh sách “danhSachDiemSv”.

3. Kiểm thử cấu trúc điều khiển

Ta có lưu đồ thuật toán cho chức năng phân lớp cho sinh viên như sau:



1. Bao phủ câu lệnh

Xét chức năng Phân lớp cho sinh viên ta có các trường hợp kiểm thử:

1. Họ tên hoặc số CMND là một chuỗi rỗng. (abcm)
2. Họ tên là một chuỗi kí tự dạng chữ khác rỗng, nhưng lại có độ dài>30. Hoặc Số CMND là 1 chuỗi kí tự dạng số có độ dài khác 9. (abdm)
3. Nếu giá trị kt= 5, họ tên là 1 chuỗi kí tự dạng chữ có dộ dài <= 30, số CMND là chuỗi kí tự dạng số có độ dài= 9, m= DialogResult.Yes/No. (abdfghjklg…m)
4. Nếu giá trị kt khác 5, họ tên là 1 chuỗi kí tự dạng chữ có độ dài <=30, số CMND là chuỗi kí tự dạng số có độ dài= 9, m= DialogResult.Yes/No. (abdfgiklg…m)

Trong đó: Ba chấm sau những đường 3, 4 cho biết rằng 1 đường đi bất kì qua phần còn lại của cấu trúc điều khiển đều chấp nhận được.

2. Bao phủ quyết định

Xét chức năng Phân lớp cho sinh viên ta có các trường hợp kiểm thử:

1. Họ tên hoặc số CMND là một chuỗi rỗng. (abcm)
2. Họ tên là một chuỗi kí tự dạng chữ khác rỗng, nhưng lại có độ dài>30. Hoặc Số CMND là 1 chuỗi kí tự dạng số có độ dài khác 9.(abdm)
3. Nếu giá trị kt=5, họ tên là 1 chuỗi kí tự dạng chữ có dộ dài <= 30, số CMND là chuỗi kí tự dạng số có độ dài= 9, m= DialogResult.Yes/No. (abdfghjklg…m)
4. Nếu giá trị kt khác 5, họ tên là 1 chuỗi kí tự dạng chữ có độ dài <= 30,số CMND là chuỗi kí tự dạng số có độ dài= 9, m= DialogResult.Yes/No. (abdfgiklg…m)

3. Bao phủ điều kiện

Xét chức năng Phân lớp cho sinh viên ta có các trường hợp kiểm thử:

1. Họ tên hoặc số CMND là một chuỗi rỗng. (abcm)
2. Họ tên là một chuỗi kí tự dạng chữ khác rỗng, nhưng lại có độ dài>30. Hoặc Số CMND là 1 chuỗi kí tự dạng số có độ dài khác 9. (abdm)
3. Nếu giá trị kt= 5, họ tên là 1 chuỗi kí tự dạng chữ có dộ dài <=30, số CMND là chuỗi kí tự dạng số có độ dài =9, m= DialogResult.Yes/No. (abdfghjklg…m)

4. Nếu giá trị kt khác 5, họ tên là 1 chuỗi kí tự dạng chữ có độ dài <=30, số CMND là chuỗi kí tự dạng số có độ dài =9, m= DialogResult.Yes/No. (abdfgiklg…m)