Đại học Quốc gia thành phố Hồ Chí Minh

Đại học Công Nghệ Thông Tin

**BÁO CÁO SERMINAR**

**ĐỀ TÀI KIỂM THỬ BẰNG PHƯƠNG PHÁP PHÂN LỚP TƯƠNG ĐƯƠNG  
(Equivalence Class Partitioning)**

**GVHD:**

**ThS. Hầu Nguyễn Thành Nam**

**SVTH:**

**11520104 – Nguyễn Quang Hiển**

**11520475 – Nguyễn Trung Việt**

Nội dung

[CHƯƠNG I: TỔNG QUAN VỀ KIỂM THỬ HỘP ĐEN 1](#_Toc439674626)

[1. Kiểm thử hộp đen là gì? 1](#_Toc439674627)

[2. Quy trình kiểm thử hộp đen 1](#_Toc439674628)

[3. Các kỹ thuật kiểm thử hộp đen: 2](#_Toc439674629)

[4. Ưu điểm và nhược điểm của kiểm thử hộp đen 2](#_Toc439674630)

[CHƯƠNG II: KIỂM THỬ BẰNG PHƯƠNG PHÁP PHÂN LỚP TƯƠNG ĐƯƠNG 4](#_Toc439674631)

[1. Giới thiệu: 4](#_Toc439674632)

[2. Ví dụ về kiểm thử dùng phương pháp phân lớp tương đương: 5](#_Toc439674633)

[3. Ưu và nhược điểm của phương pháp phân lớp tương đương: 10](#_Toc439674634)

[CHƯƠNG III: KẾT LUẬN 11](#_Toc439674635)

[1. Nhận xét 11](#_Toc439674636)

[2. Hướng phát triển 11](#_Toc439674637)

[TÀI LIỆU KHAM THẢO 12](#_Toc439674638)

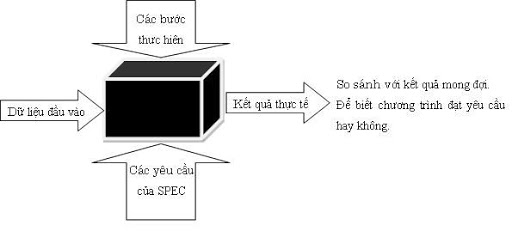
# 

# CHƯƠNG I: TỔNG QUAN VỀ KIỂM THỬ HỘP ĐEN

## Kiểm thử hộp đen là gì?

Đối tượng được kiểm thử là 1 thành phần phần mềm (TPPM). TPPM có thể là 1 hàm chức năng, 1 module chức năng, 1 phân hệ chức năng… Nói chung, chiến lược kiểm thử hộp đen thích hợp cho mọi cấp độ kiểm thử từ kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống, kiếm thử độ chấp nhận của người dùng.

Kiểm thử hộp đen (black-box testing) là chiến lược kiểm thử TPPM dựa vào thông tin duy nhất là các đặc tả về yêu cầu chức năng của TPPM tương ứng.



Đây là chiến lược kiểm thử theo góc nhìn từ ngoài vào, các người tham gia kiểm thử hộp đen không cần có kiến thức nào về thông tin hiện thực TPPM cần kiểm thử (mã nguồn của thành phần phần mềm, thuật giải được dùng, các dữ liệu được xử lý…).

## Quy trình kiểm thử hộp đen

Bước 1: Phân tích đặc tả về các yêu cầu chức năng mà TPPM cần thực hiện.

Bước 2: Dùng 1 kỹ thuật định nghĩa các testcase xác định (sẽ giới thiệu sau) để định nghĩa các testcase. Định nghĩa mỗi testcase là xác định 3 thông tin sau:

* Giá trị dữ liệu nhập để TPPM xử lý (hoặc hợp lệ hoặc không hợp lệ).
* Trạng thái của TPPM cần có để thực hiện testcase.
* Giá trị dữ liệu xuất mà TPPM phải tạo được.

Bước 3: Kiểm thử các testcase đã định nghĩa.

Bước 4: So sánh kết quả thu được với kết quả kỳ vọng trong từng testcase, từ đó lập báo cáo về kết quả kiểm thử.

Lập báo cáo

Kiểm thử

Xác định phương pháp

Phân tích yêu cầu

## Các kỹ thuật kiểm thử hộp đen

Vì chiến lược kiểm thử hộp đen thích hợp cho mọi mức độ kiểm thử nên nhiều người đã nghiên cứu tìm hiểu và đưa ra nhiều kỹ thuật kiểm thử khác nhau, chúng ta sẽ chọn ra 8 kỹ thuật có nhiều ưu điểm nhất và được dùng phổ biến nhất, đó là :

* Kỹ thuật phân lớp tương đương (Equivalence Class Partitioning).
* Kỹ thuật phân tích các giá trị biên (Boundary value analysis).
* Kỹ thuật dùng các bảng quyết định (Decision Tables).
* Kỹ thuật kiểm thử các bộ n thần kỳ (Pairwise).
* Kỹ thuật dùng bảng chuyển trạng thái (State Transition).
* Kỹ thật phân tích vùng miền (domain analysis).
* Kỹ thuật dựa trên đặc tả Use Case (Use case).
* Kỹ thuật dùng lược đồ quan hệ nhân quả (Cause-Effect Diagram).

## Ưu điểm và nhược điểm của kiểm thử hộp đen

Kiểm thử hộp đen nhìn chung có đặc điểm chính là chỉ quan tâm chủ yếu đến kết quả đạt được của từng TPPM nên ta có thể suy ra ưu và nhược điểm nó như sau:

Ưu điểm:

* Tester không cần tìm hiểu thuật giải hay kiến trúc phần mềm quá nhiều nên ở góc độ nào đó có thể giảm thiểu thời gian kiểm thử.
* Giúp tester cũng như những người hỗ trợ testing có một cái nhìn độc lập so với developer nên trong một số tình huống giúp phát hiện ra nhiều lỗi mà developer không nhận ra.
* Vì không cần biết kiến trúc cũng như thuật giải phần mềm nên việc kiểm thử hộp đen có thể nhận được nhiều sự trợ giúp từ nhiều phía kể cả người dùng cuối.

Khuyết điểm:

* Nếu phát sinh ra lỗi thì việc testing không giúp được gì nhiều trong việc tìm nguyên nhân và cách khắc phục lỗi (so với các kỹ thuật kiểm thử khác).
* Trong nhiều trường hợp việc kiểm thử hộp đen sử dụng số lượng testcase nhiều hơn các loại kiểm thử khác (nguyên nhân vẫn do việc không nắm rõ thuật giải).

Tóm lại,kiểm thử hộp đen có ưu điểm là "an unaffiliated opinion" (một quan điểm độc lập), mặt khác, nó có nhược điểm là "blind exploring" (khám phá mù).

# CHƯƠNG II: KIỂM THỬ BẰNG PHƯƠNG PHÁP PHÂN LỚP TƯƠNG ĐƯƠNG

## Giới thiệu

Tinh thần của kỹ thuật này là cố gắng phân các testcase ra thành nhiều nhóm (họ) khác nhau : các testcase trong mỗi họ sẽ kích hoạt TPPM thực hiện cùng 1 hành vi.

Mỗi nhóm testcase thỏa mãn tiêu chuẩn trên được gọi là 1 lớp tương đương, ta chỉ cần xác định 1 testcase đại diện cho nhóm và dùng testcase này để kiểm thử TPPM.



Như vậy ta đã giảm rất nhiều testcase cần định nghĩa và kiểm thử, nhưng chất lượng kiểm thử không bị giảm sút bao nhiêu so với vét cạn. Điều này là dựa vào kỳ vọng rất hợp lý sau đây :

* Nếu 1 testcase trong lớp tương đương nào đó gây lỗi TPPM thì các testcase trong lớp này cũng sẽ gây lỗi như vậy.
* Nếu 1 testcase trong lớp tương đương nào đó không gây lỗi TPPM thì các testcase trong lớp này cũng sẽ không gây lỗi.

Vấn đề kế tiếp là có cần định nghĩa các lớp tương đương đại diện các testcase chứa các giá trị không hợp lệ theo đặc tả hay không? Điều này phụ thuộc vào tinh thần kiểm thử :

* Nếu ta dùng tinh thần kiểm thử theo hợp đồng (Testing by Contract) thì không cần định nghĩa các lớp tương đương đại diện các testcase chứa các giá trị không hợp lệ theo đặc tả vì không cần thiết.
* Còn nếu ta dùng tinh thần kiểm thử phòng vệ (Defensive Testing), nghĩa là kiểm thử hoàn hảo, thì phải định nghĩa các lớp tương đương đại diện các testcase chứa các giá trị không hợp lệ theo đặc tả để xem TPPM phản ứng như thế nào với những testcase này.

## Ví dụ về kiểm thử dùng phương pháp phân lớp tương đương:

Thí dụ ta cần kiểm thử 1 TPPM “quản lý nguồn nhân lực” với đặc tả chức năng như sau: mỗi lần nhận 1 hồ sơ xin việc, TPPM sẽ ra quyết định dựa vào tuổi ứng viên theo bảng sau :

|  |  |
| --- | --- |
| Tuổi ứng viên | Kết quả |
| 0-16 | Không thuê |
| 16-18 | Thuê dạng bán thời gian |
| 18-55 | Thuê toàn thời gian |
| 55-99 | Không thuê |

Lưu ý rằng bảng đặc tả chức năng phía trên có lỗi ở các giá trị đầu và (hoặc) cuối trong từng luật, và giả sử chúng ta chưa phát hiện lỗi này. Chúng ta sẽ thấy bằng cách nào sẽ phát hiện dễ dàng lỗi này.

Phân tích đặc tả chức năng của TPPM cần kiểm thử ở trên, ta thấy có 4 lớp tương đương, mỗi lớp chứa các testcase ứng với 1 chế độ xử lý của TPPM : không thuê vì quá trẻ, thuê dạng bán thời gian, thuê toàn thời gian, không thuê vì quá già.

Ứng với mỗi lớp tương đương, ta định nghĩa 1 testcase đại diện, thí dụ ta chọn 4 testcase sau :

1. Testcase 1 : {Input : 2 tuổi, Output : không thuê}

2. Testcase 2 : {Input : 17 tuổi, Output : thuê bán thời gian}

3. Testcase 3 : {Input : 35 tuổi, Output : thuê toàn thời gian}

4. Testcase 4 : {Input : 90 tuổi, Output : không thuê}

Trong thí dụ trên, thay vì phải kiểm thử vét cạn 100 testcase, ta chỉ kiểm thử 4 testcase → chí phí giảm rất lớn, nhưng chất lượng kiểm thử hy vọng không bị giảm sút là bao.

Tại sao chúng ta hy vọng chất lượng kiểm thử dùng lớp tương đương không giảm sút nhiều ? Hãy xét đoạn code mà những người lập trình bình thường sẽ viết khi xử lý TPPM cần kiểm thử ở trên:

if (applicantAge >= 0 && applicantAge <=16) qd ="NO";

if (applicantAge >= 16 && applicantAge <=18) qd ="PART";

if (applicantAge >= 18 && applicantAge <=55) qd ="FULL";

if (applicantAge >= 55 && applicantAge <=99) qd ="NO";

Ở góc nhìn kiểm thử hộp trắng, nếu dùng 4 testcase đại diện của 4 lớp tương đương, ta sẽ kiểm thử được ở phủ cấp 3, cấp phủ rất tốt vì đã kiểm thử 100% các lệnh mã nguồn, 100% các nhánh quyết định.

Tuy nhiên nếu người lập trình hiện thực như sau (rất cá biệt vì đây là người lập trình rất yếu tay nghề) :

if (applicantAge == 0) qd ="NO";

…

if (applicantAge == 16) qd ="PART";

…

if (applicantAge == 53) qd ="FULL";

…

if (applicantAge == 99) qd ="NO";

Thì nếu dùng 4 testcase đại diện của 4 lớp tương đương, ta mới kiểm thử được 4/100 lệnh mã nguồn của TPPM, mức độ phủ này chưa thể nói lên gì nhiều về TPPM!

Làm sao chọn testcase đại diện cho lớp tương đương ? Điều này phụ thuộc vào kiểu dữ liệu nhập. Ta hãy lần lượt xét 1 số kiểu dữ liệu nhập phổ biến.

Thí dụ ta cần kiểm thử 1 TPPM “xét đơn cầm cố nhà” với đặc tả chức năng như sau : mỗi lần nhận 1 đơn xin cầm cố, TPPM sẽ ra quyết định chấp thuận nếu 4 điều kiện sau đều thỏa mãn :

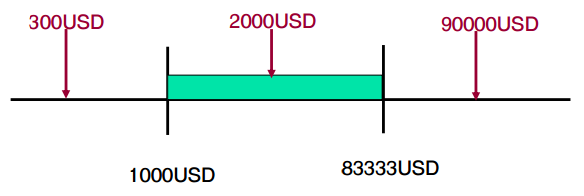
1. Thu nhập hàng tháng của đương đơn nằm trong khoảng từ 1000$ đến 83333$.

2. Số nhà xin cầm cố từ 1 đến 5.

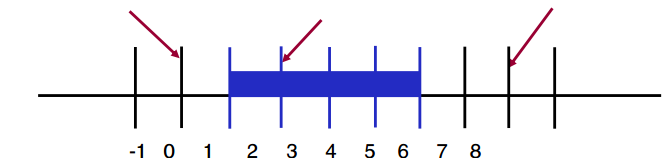
3. Đương đơn phải là cá nhân, không được là hội, công ty hay người được ủy nhiệm (partnership, trust, corporation).

4. Loại nhà cầm cố phải là loại nhà cố định (single family,condo, townhouse), không xét loại nhà di động (treehouse,duplex, mobile home).

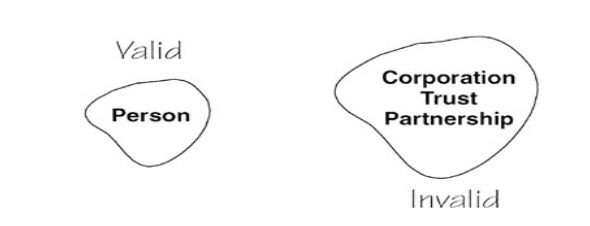
Với điều kiện thứ 1: Nếu lớp tương đương được xác định bởi các dữ liệu nhập là số thực liên tục, thì ta chọn 1 testcase đại diện có giá trị nhập hợp lệ nằm trong khoảng liên tục các giá trị hợp lệ, và nếu muốn, 2 testcase miêu tả giá trị không hợp lệ nằm phía dưới và phía trên khoảng trị hợp lệ (số testcase cho mỗi lớp tương đương là từ 1 tới 3).



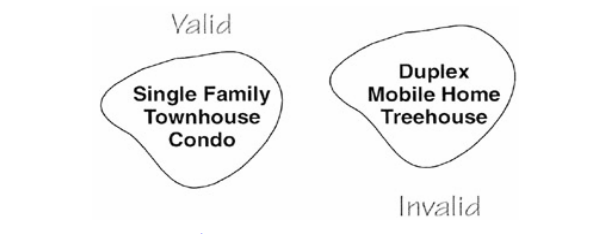
Với điều kiện thứ 2: Nếu lớp tương đương được xác định bởi các dữ liệu nhập là số nguyên liên tục, trong trường hợp này ta chọn 1 testcase đại diện có giá trị nhập hợp lệ nằm trong khoảng liên tục các giá trị hợp lệ, và nếu muốn, 2 testcase miêu tả giá trị không hợp lệ nằm phía dưới và phía trên khoảng trị hợp lệ (số testcase cho mỗi lớp tương đương là từ 1 tới 3).



Với điều kiện thứ 3: Nếu lớp tương đương được xác định bởi các dữ liệu dạng liệt kê rời rạc và không có mối quan hệ lẫn nhau gồm 1 trị hợp lệ và nhiều trị không hợp lệ. Trong trường hợp này ta chọn 1 testcase có giá trị nhập hợp lệ và nếu muốn, 2 testcase miêu tả 2 giá trị không hợp lệ nào đó, nhưng cho dù chọn 2 testcase nào cũng không đại diện tốt cho các trường hợp không hợp lệ còn lại (số testcase cho mỗi lớp tương đương là từ 1 tới 3).



Với điều kiện thứ 4: Nếu lớp tương đương được xác định bởi các dữ liệu dạng liệt kê rời rạc và không có mối quan hệ lẫn nhau gồm n trị hợp lệ và m trị không hợp lệ. Trong trường hợp này ta chọn 1 testcase có giá trị nhập hợp lệ nào đó và nếu muốn, 2 testcase miêu tả 2 giá trị không hợp lệ nào đó, nhưng cho dù chọn các testcase nào cũng -10 1 2 3 4 5 6 7 8 không đại diện tốt cho các trường hợp hợp lệ và không hợp lệ còn lại (số testcase cho mỗi lớp tương đương là từ 1 tới 3).



Khi TPPM cần kiểm thử nhận nhiều dữ liệu nhập (thí dụ TPPM xét đơn cầm cố nhà ở trên trước có 4 loại dữ liệu nhập), ta định nghĩa các testcase độc lập cho các dữ liệu hay testcase dựa trên tổng hợp các dữ liệu nhập?

Nếu định nghĩa các testcase độc lập trên từng loại dữ liệu nhập, số lượng testcase cần kiểm thử sẽ nhiều. Trong TPPM xét đơn cầm cố nhà, ta phải xử lý ít nhất là 3 testcase cho từng loại dữ liệu \* 4 loại dữ liệu = 12 testcase.  
Để giảm thiểu số lượng testcase nhưng vẫn đảm bảo chất lượng kiểm thử, người ta đề nghị chọn tescase như sau:

* 1 testcase cho tổ hợp các giá trị hợp lệ.
* n testcase cho tổ hợp các giá trị trong đó có 1 giá trị không hợp lệ, các giá trị còn lại hợp lệ, nên thay đổi các giá trị hợp lệ trong tổ hợp cho mỗi testcase.

Thí dụ TPPM xét đơn cầm cố nhà ở trên có 4 loại dữ liệu nhập, ta định nghĩa các testcase dựa trên tổ hợp các dữ liệu nhập như sau :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thu nhập/tháng | Số lượng nhà | Đương đơn | Kiểu nhà | Kết quả |
| 5.000 | 2 | Person | Condo | Valid |
| 100 | 1 | Person | Single | Invalid |
| 1.342 | 0 | Person | Condo | Invalid |
| 5.432 | 3 | Corporation | Townhouse | Invalid |
| 10.000 | 2 | Person | Treehouse | Invalid |

Với quy tắc như trên chúng ta có 5 testcase với độ che phủ hàm (theo cách nhìn hộp trắng) khá tốt, đảm bảo không giảm chất lượng kiểm thử nhiều.

## Ưu và nhược điểm của phương pháp phân lớp tương đương:

Vì là một phương pháp kiểm thử theo quy tắc của kiểm thử hộp đen nên phương pháp phân lớp tương đương cũng có những ưu,nhược điểm đã đề cập ở chương trước. Ngoài ra ta còn có thể rút ra một số ưu,nhược điểm khác:

**Ưu điểm:**

Vì mỗi vùng tương đương ta chỉ cần test trên các phần tử đại diện nên số lượng test case được giảm đi khá nhiều nhờ đó mà thời gian thực hiện test cũng giảm đáng kể.

**Nhược điểm:**

Không phải với bất kỳ bài toán nào đều có thể áp dụng kỹ thuật này. Có thể bị lack lỗi ở biên nếu chỉ chọn giá trị ở khoảng giữa của miền tương đương.  
Vì vậy việc kết hợp linh hoạt giữa kỹ thuật phân vùng tương đương và phân tích giá trị biên dưới đây sẽ mang lại hiệu quả cao hơn để vừa tối ưu số lượng test case và vẫn đảm bảo đươc chất lượng phần mềm.

# CHƯƠNG III: KẾT LUẬN

## Nhận xét

Ngoài kỹ thuật thiết kế test case đã nói ở trên, bạn cũng có thể tìm hiểu và sử dụng rất nhiều các kỹ thuật khác nữa như: thiết kế test case dựa trên đồ thị nguyên nhân – kết quả(Cause-Effect Diagram), dựa trên bảng quyết định(Decision Tables)…

Phương pháp kiểm thử này có thể được áp dụng cho tất cả các cấp kiểm thử phần mềm: đơn vị, tích hợp, hệ thống và chấp nhận. Nó không thể thực hiện được tất cả các kiểm thử các cấp độ cao hơn nhưng nó có thể tạo ưu thế tốt khi kiểm thử từng đơn vị.

Để giảm thiểu số case đến mức tối ưu mà vẫn đảm bảo chất lượng phần mềm, mỗi tester cần linh hoạt trong việc lựa chọn các kỹ thuật thiết kế test case.

## Hướng phát triển

Việc sử dụng phương pháp nào phụ thuộc rất nhiều yếu tố, không có phương pháp nào hoàn hảo, chỉ có phương pháp phù hợp nhất với việc phát triển phần mềm của từng công ty.

Vì thế, việc sử dụng kiểm thử đòi hỏi nhiều tâm huyết cũng như kinh nghiệm của tester và các yếu tố khác trong quá trình phát triển phần mềm.

# TÀI LIỆU KHAM THẢO

Wikipedia.

The art of software testing- Glenford J. Myers.

Tham khảo nhiều luận văn của các trường khác.