# **Thợ lành nghề #31: Turn Off Force File (Giám sát liều lượng – Phần 8)**

**Robert C. Martin**

**30/10/2004**

*(…Tiếp tục phần trước)*

*Dự án Nimbus bắt đầu từ những tháng đầu tiên năm 1940. Đến giữa năm, khi người Đức tiến quân vào Paris, và trên khắp các vùng trời nước Ạnh bị phủ đen bởi các máy bay ném bom của Đức, General Leslie R. Groves và J. Robert Oppenheimer đã thiết lập một cơ sở lớn ở Los Alamos, New Mexico.*

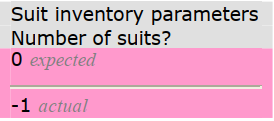
*FDR nhận định chiến tranh ở châu Âu quan trọng hơn rất nhiều so với mối đe dọa của một “hòn đá ngoài không gian”, nhưng cũng nhận thức được món quà mà hội đoàn tặng cho nước Mỹ tuyệt vời như thế nào. Bản tuyên bố chính thức của dự án Nimbus là phát triển các công nghệ nguyên tử, điện tử và tên lửa cần thiết để bảo vệ nước Mỹ chống lại các kẻ thù bên ngoài. Thêm vào đấy, đó là tất cả những gì mà các tướng lĩnh và các nhà hoạch định chính sách thực sự mong đợi. Đó cũng chính là mặt lợi ích mà các công nghệ này chính xác là những công nghệ hội đoàn yêu cầu như một thiết yếu để phòng thủ chống lại Clyde.*

*Trong khi đó elip mục tiêu của Clyde tiếp tục thu hẹp lại với Trái Đất, nơi vẫn duy trì khoảng cách gần với trọng tâm của nó. Khi năm 1940 dần trôi qua, tỉ lệ va chạm vẫn là 1000/1; cho đến tận khi Werner Von Braun phóng thành công tên lửa A4 ở sa mạc New Mexico, mục tiêu của trong đầu anh ta cũng không có trên Trái Đất.*

**21/2/2002 1330**

Jerry và tôi có giờ nghỉ trong chốc lát. Nhạc điệu ngọt ngào của Turn off this Force Filed du dương phát ra từ loa phóng thanh ở phòng nghỉ. Giai điệu buồn làm tôi nghĩ tới Avery trong Phòng trừng phạt với Jean. Tôi tự hỏi không biết trong đó đang diễn ra chuyện gì.

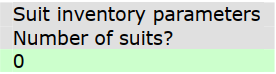
Avery và Jean vẫn còn ở đó khi chúng tôi trở lại sau giờ giải lao. Lúc ngồi xuống, Jerry và tôi nhìn nhau nhưng không hề nói câu nào về điều mà cả hai đang nghĩ trong đầu. Thay vào đó, Jerry nói: “Được rồi, hãy làm kiểm thử chấp nhận này qua được đi. Đây là bảng đầu tiên bị thất bại.”



“Được thôi.” Tôi nói. “Chúng ta có thể làm nó thành công bằng cách thay đổi phương thức thích hợp trong lớp Utilities.” Rồi tôi cầm lấy bàn phím và gõ:

| 1  2  3 | public static int getNumberOfSuitsInInventory() {  return 0;</>  } |
| --- | --- |

Thế là đủ để kiểm thử chuyển thành màu xanh.



Nhưng Jerry lại lắc đầu. “Không, Alphonse, chúng ta không muốn nó thành công theo cách này. Chúng ta muốn nó thực sự qua được cơ.”

Tôi chỉ vào ô màu xanh lá cây trrn màn hình và nói: “Ý anh là sao? Theo như tôi thấy thì kiểm thử thực sự đang thành công rồi mà.”

“Kiểm thử chấp thuận không giống như kiểm thử đơn vị. Chúng ta không dùng chúng cho cùng mục đích. Chúng ta dùng kiểm thử đơn vị để giúp chúng ta thiết kế các lớp và phương thức. Chúng ta dùng kiểm thử chấp nhận để đảm bảo rằng hệ thống chúng ta hoạt động theo như chỉ định. Thay vì chỉ đơn giản là làm cho kiểm thử chấp nhận chuyển thành màu xanh, cái mà chúng ta thực sự muốn làm là phát triển phương thức getNumberofSuiteInInventory để nó hoạt động theo cách nó nên như thế. Và vì điều này, chúng ta sẽ cần một vài kiểm thử đơn vị.”

“Tôi không chắc là tôi có thể hiểu được ý của anh.” Tôi nói. “Không phải chúng ta phải làm điều đơn giản nhất chúng ta có thể nghĩ ra để làm các kiểm thử thành công sao?”

“Vói các kiểm thử đơn vị thì gần như là vậy. Nhưng chúng ta làm các kiểm thử chấp nhận thành công với mã đã được kiểm thử đơn vị rất tốt.”

“Được rồi, tôi nghĩ tôi hiểu ra rồi. Nếu tôi thực sự muốn làm cái gì đó siêu đơn giản để có một kiểm thử chấp nhận thành công, đồng nghĩa với tôi nên viết một kiểm thử đơn vị.”

Jerry cười. “Đó là cách hay để nghĩ về nó.”

“OK, vậy hãy thêm một trường hợp kiểm thử vào UtilitiesTest.” Tôi lấy lại bàn phím và viết:

| 1  2  3 | public void testNoSuitsInInventory() throws Exception {  assertEquals(0, Utilities.getNumberOfSuitsInInventory());  } |
| --- | --- |

Đúng như dự đoán, kiểm thử rất nhanh đã thành công. Sau đó tôi viết

| 1  2  3  4 | public void testOneSuitInInventory() throws Exception {  Utilities.addSuit(new Suit(1, new Date()));  assertEquals(1, Utilities.getNumberOfSuitsInInventory());  } |
| --- | --- |

Đoạn này không được biên dịch vì không có phương thức addSuit. Nên tôi tiếp tục viết:

| 1  2 | public static void addSuit(Suit suit) {  } |
| --- | --- |

Rồi tôi dừng lại. “Chúng ta nên thêm bộ đồ như thế nào?” Tôi hỏi.

“Hỏi rất hay.” Jerry nói. “Mày nghĩ xem nó nên ở chỗ nào thì được?”

“Ờm, tôi đoán đại loại chúng ta cần một cơ sở dữ liệu gì đó. Chúng ta có nên tạo ra một cái luôn không?”

“Không, còn quá sớm để làm nó lúc này.” Jerry đáp.

“Chà vậy tôi không thể hoàn thiện phương thức này được.” Tôi nói.

“Mày chắc chắn có thể.” Jerry thúc giục. “Chỉ cần dùng một interface.”

“Interface cho cái gì cơ?” Tôi không hiểu nổi điều hắn đang nói với tôi.

Jerry thở dài và nói: “Một interface cho một cổng.”

Sự du dương uyển chuyển của Turn off This Force Field bắt đầu vang lên trong đầu tôi lần nữa. Những lời Jerry nói chẳng còn ý nghĩa gì cả, nên tôi chỉ nhìn chằm chằm hắn ta trong khi những giai điệu cứ lởn vởn trong đầu tôi. Sau khoảng 15 giây, Jerry đảo mắt, chộp lấy bàn phím và nói:

“Một cách để giải quyết cơ sở dữ liệu là tạo một đối tượng gọi là gateway. Một gateway đơn giản là một đối tượng biết làm thế nào để vận hành các bản ghi trong một database. Nó biết cách để tạo ra chúng, cập nhất chúng, truy vấn chúng, xóa chúng đi và còn nhiều thứ nữa. Trong trường hợp này, chúng ta sẽ tạo một thứ có tên TABLE DATA GATEWAY1.” Một TDG biết cách vận hành các hàng trong một cơ sở dữ liệu cụ thể.”

Jerry viết như dưới:

| 1  2  3  4  5 | package dtrack.gateways;  import dtrack.dto.Suit;  public interface SuitGateway {  public void add(Suit suit);  } |
| --- | --- |

“Vâng, tôi nghĩ tôi hiểu.” Tôi nói. “Vậy chúng ta sẽ có thể sửa đổi lớp Utilities như thế này”. Và tôi lấy bàn phím gõ:

| 1  2  3  4  5  6  7 | public class Utilities {  ...  private static SuitGateway suitGateway;  public static void addSuit(Suit suit) {  suitGateway.add(suit);  }  } |
| --- | --- |

Mọi thứ đã được biên dịch, nên tôi nhấn nút kiểm tra như một thói quen. Chúng tôi có được một NullPointerException trong addSuit, giống như cái bạn đang trông đợi.

“Giờ thì sao?” Tôi nói.

“Tạo ra một thực thi của SuitGateway giữ các bộ đồ trong RAM.” Jerry nói.

“Chúng ta sẽ bỏ nó theo cách đó, phải chứ? Ý tôi là những bộ đồ đó thực sự cần phải được viết vào một cơ sở dữ liệu có thực, đúng không?”

Jerry nhìn tôi với cái nhìn khó hiểu và nói: “Mày sợ cái gì vậy hả Alphonse? Mày nghĩ chúng ta sẽ quên lưu trữ dữ liệu trên đĩa sao?”

“Không phải, chỉ là điều này có vẻ như… Tôi không biết nữa… sẽ có lúc hoạt động không đúng”

“Chà, tao nói cho mày biết nhá, nó sẽ không lệch quy trình đâu, Một trong những cách tệ nhất khi thiết kế một hệ thống là nghĩ đến cơ sở dữ liệu đầu tiên. Tao biết chứ.” Jerry liếc nhìn vào phòng trừng phạt một lúc. “Tin tao đi, tao biết mà. Cái chúng ta muốn làm là đẩy các decision vào trong cơ sơ sở dữ liệu ngay khi có thể. Rốt cuộc, chúng ta sẽ tìm ra cách để đảm bảo rằng các bộ đồ được lưu trữ trên đĩa. Tôi không biết liệu rằng chúng ta có sử dụng một cơ sở dữ liệu thật (dù đó là bất kì cái gì) hay không. Có lẽ chúng ta chỉ lấy tất cả dữ liệu trong RAM và viết nó vào các tệp phẳng theo định kỳ. Ngay lúc này tao thực sự không biết, và tao cũng không muốn biết. Chúng ta sẽ tiến hành các chi tiết đó ngay khi có tiến triển.”

Tôi không thích những lời nói đó. Điều đó với tôi dường như là bạn có thể rơi vào thế tự dồn mình vào chân tường khá nhanh chóng nếu như bạn không nghĩ về cơ sở dữ liệu trước.

“Đã có Đối số Cơ sở dữ liệu chứ?”

Đó là Carole. Cô ta chắc đã vô tình nghe thấy cuộc nói chuyện của chúng tôi. Jerry gật đầu đồng thuận và nói: “Đang đúng tiến độ.”

Carole mỉm cười và nói: “Alphonse, nhớ nhắc tôi kể cho cậu nghe về Hệ thống giám sát liều lượng đầu tiên mà tôi và Jerry đã viết cách đây ít năm đấy.”

“Tôi thách cô đấy!” Jerry đáp trả với sự tức giận và có thoáng sợ hãi.

Carole lại cười tươi, lắc đầu và đi về phía trạm làm việc của mình.

“Được rồi, Alphonse, hãy viết một bộ nhớ RAM đơn giản dựa trên thực thi của SuitGateway”. Vậy nên tôi lấy bàn phím và gõ:

| 1  2  3  4  5  6 | public class InMemorySuitGateway implements SuitGateway {  private Map suits = new HashMap();  public void add(Suit suit) {  suits.put(new Integer(suit.barCode()), suit);  }  } |
| --- | --- |

Sau đó tôi sửa lại lớp Utilities để tạo một instance thích hợp.

| 1  2  3  4 | public class Utilities {  ...  private static SuitGateway suitGateway <strong>= new InMemorySuitGateway()</strong>;  } |
| --- | --- |

Và giờ thì kiểm thử thất bại với:

| 1 | expected:<1> but was:<0> |
| --- | --- |

“Tốt rồi, Alphonse, mày đã biết cách để làm nó thành công rồi đúng không?”

Tôi gật đầu, và thêm mã còn lại:

| 1  2  3  4  5  6  7  8  9  1 | public class Utilities {  ...  public static int getNumberOfSuitsInInventory() {  return <strong>suitGateway.getNumberOfSuits();  </strong>}  }  public interface SuitGateway {  public void add(Suit suit);  <strong>int getNumberOfSuits();  </strong>}  public class InMemorySuitGateway implements SuitGateway {  private Map suits = new HashMap();  public void add(Suit suit) {  suits.put(new Integer(suit.barCode()), suit);  }  <strong>public int getNumberOfSuits() {  return suits.size();  }  </strong>} |
| --- | --- |

Và giờ tất cả kiểm thử đơn vị đều qua được, kể cả bảng các tham số kho hàng bộ đồ của kiểm thử chấp nhận.

Tôi nhìn đoạn mã này một chốc lát và tôi nói: “Jerry, tôi không chắc lớp Utilities có cái tên thực sự tốt.”

“Thật sao?” Hắn nói – dù rằng tôi có thể nói hắn đang cười thầm.

“Ừ, và tôi cũng không chắc các hàm static như getNumberOfSuitsInInventory cũng phải ở đó.”

“Vậy mày nghĩ nó nên trông như thế nào?”

“Có lẽ chúng ta nên đổi tên Utilities thành Gateways. Có lẽ nó nên chỉ có các biến static giữ tất cả các đối tượng gateway. Và có lẽ mọi người thực hiện các cuộc gọi thông qua các cổng này.” Tôi nói.

“Ý tưởng thú vị đấy.” Jerry nói.

“Chào mọi người.”

Đó là Avery. Jerry và tôi lập tức quay lại khi nghe thấy giống hắn ta. Cả Avery và Jean đều đang đứng đợi ở đó để nói chuyện với chúng tôi. Tôi có thể thoáng nghe thấy Turn Of This Force Field xuất phát từ cánh cửa đang mở ở phòng giải lao.

*Còn tiếp…*