# **Thợ lành nghề #47: Túi nâu 4 – Hằng số enum, “Đồ hãm tài”**

Robert C. Martin

7/2/2006

…Tiếp nối phần trước

Tháng 9, 1944

Tổng thống Henry Wallace ngồi một mình và úp mặt vào tay. “Có lẽ chúng ta đáng bị phá hủy.” Ông ấy nói với một khoảng không vô định. Việc hủy diệt nhân loại dường như đã chắc chắn ở trong vài khía cạnh nào đó. Clyde sẽ đến trong 15 năm tới, và có vẻ như không gì cản được nó. [Third Reich](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%A9c_Qu%E1%BB%91c_X%C3%A3) đã hủy diệt toàn bộ nhân loại một cách có hệ thống. Stalin tiếp tục sự thanh trừng đối với bất kỳ dấu hiệu nào của các tư tưởng đối lập. Sự độc quyền của Mỹ về vũ khí nguyên tử không thể kéo dài lâu hơn – và sau đó sẽ có chiến tranh và sự tàn sát trên quy mô mà chỉ có Clyde có thể làm đối thủ. Quả thật, có lẽ Clyde đơn giản giống như người chôn cất của Chúa; được gửi đến để chôn vùi một Trái đất đã chết sẵn rồi.

Lúc sau có tiếng gõ cửa: “Thưa ngài Tổng thống? Tiến sĩ Von Braun đến đây để gặp ngài.”

Thứ 5, 28/2/2002, 1100

Avery và tôi cùng đi đến một phòng họp nhỏ mà bọn tôi dùng cho các cuộc thảo luận hàng ngày. Tôi cho rằng sự cau có trên mặt hắn ta có liên quan đến việc Jerry và Jasmine đang theo sau. Mọi người ổn định chỗ ngồi và Jerry nói: “Được rồi, hôm nay đến lượt của ai đây?” Avery càng cau có hơn nhưng vẫn im lặng.

“Thường thì Avery và tôi luân phiên nhau.” Tôi nói. “Nhưng Avery hôm qua bỏ lượt vậy nên…”

Avery lạnh lùng liếc nhìn Jerry, nhưng sau đó nói: “Không sao đâu, Alphonse, nay là lượt của mày. Cứ tự nhiên đi. Tao chẳng chuẩn bị thứ gì cả đâu.”

“Được thôi, thật tình tao cũng không chuẩn bị gì nhiều nhưng tao có một câu hỏi. Có ai hiểu được tính năng mới của enum trong Java 5 không? Rõ ràng ta có thể định nghĩa các hằng số enum mà có các biến và phương thức.”

Avery đột nhiên trở nên hoạt bát. “Mày có thể viết một ví dụ lên bảng không Alphonse?”

“Được chứ.”, và rồi tôi lấy tay viết những dòng code sau lên bảng tường.

public class Taco {

enum Color {red, green, burntumber}

}

“Đây.”, tôi nói, “chúng ta đều biết rằng Java 1.5 có các enum trông như vậy. Chúng là những hằng số nguyên bản.”

Avery nhảy lên tường, bắt đầu vừa viết vừa nói cùng lúc. “Ừ, trước 1.5 chúng ta sẽ hoàn thành nó.”

public class Taco {

public static final int RED = 0;

public static final int GREEN = 1;

public static final int BURNT\_UMBER = 2;

}

Cái bảng tường nhận ra cách viết của Avery và đảm bảo mã của hắn nằm ở trình soạn thảo riêng biệt với cái của tôi, vậy nên chúng không có xung đột.

“Đúng là thế.” Tôi nói. “Nhưng đó là nơi sự đồng dạng chấm dứt. Trong Java 1.5 một bảng liệt kê thực sự là một lớp, và mỗi một phần liệt kê là một lớp dẫn xuất. Các lớp này có thể có các biến. Chẳng hạn như: ” Và tôi lấy tay mình sửa đổi mã trên màn hình.

public class Taco {

enum Color {red, green, burntumber}

enum Salsa {

mild(10, Color.red),

medium(1000, Color.green),

hot(100000, Color.burntumber);

public final int scovilles;

public final Color color;

Salsa(int scovilles, Color color) {

this.scovilles = scovilles;

this.color = color;

}

}

}

“Hay lắm!” Jasmine nói. “Trông rất tuyệt đấy. Nhìn này Jerry, enum có một hàm khởi tạo và các biến instance!”

Jerry thoạt nhìn lên tường một chút rồi nói: “Ừ, thực sự hữu ích. Điều này có nghĩa rằng một hằng số có thể có nhiều hơn một giá trị. Vì vậy, hãy xem xét đại diện này cho một màu đỏ / xanh dương / xanh lá: “Jerry cũng bắt đầu viết lên tường.

public enum RGB {

red(0xff, 0x00, 0x00),

green(0x00, 0xff, 0x00),

blue(0x00, 0x00, 0xff),

white(0xff,0xff,0xff);

public final int r;

public final int g;

public final int b;

RGB(int r, int g, int b) {

this.r = r;

this.g = g;

this.b = b;

}

}

Tôi tựa lưng vào tường và nói: “Đúng thế, và điều này cho phép anh làm những điều thú vị giống như thế này.”

public class Taco {

enum Color {red, green, burntumber}

enum Salsa {

mild(10, Color.red),

medium(1000, Color.green),

hot(100000, Color.burntumber);

public final int scovilles;

public final Color color;

Salsa(int scovilles, Color color) {

this.scovilles = scovilles;

this.color = color;

}

}

public static boolean isTooHotForMe(Salsa s) {

return s.scovilles > 30000;

}

}

“Ồ, oa!” Jasmine nói. “Điều đó có nghĩa là chúng ta có thể viết mã để đưa ra những quyết định về enum mà không cần để ý tới bảng liệt kê nào!”

“Hả?” Jerry nói. “Ý cô là gì?”

“Chà, hãy nhìn vào những gì Alphonse đã viết!” Cô thúc giục. “Phương thức isTooHotForMe sẽ làm việc cho bất kỳ điều tra viên nào trong liệt kê Salsa. Ngay cả khi sau này chúng ta thêm các phần liệt kê mới, phương thức isTooHotForMe sẽ tiếp tục hoạt động không thay đổi.”

“Nguyên tắc đóng mở!” Tôi buột miệng.

“Ừ!” Avery nói. “Chúng tôi có thể thêm, thay đổi hoặc xóa các liệt kê mà không ảnh hưởng đến phương pháp isTooHotForMe!”

Jerry nhìn chăm chú vào bức tường trong vài giây, rồi nói: “Hừm, điều đó ám chỉ một cách suy nghĩ hoàn toàn mới về hằng số và enum. Đôi khi chúng ta KHÔNG muốn biết những phần lietj kê có gì, và thay vào đó phụ thuộc vào cấu trúc bên trong. Ví dụ, chúng ta có thể viết một hàm isBlue như sau:

class SomeClass {

public static boolean isBlue(RGB color){

return color.b > 2\*(color.r + color.g);

}

}

Jasmine đảo mắt, dùng khuỷu tay huých vào xương sườn Jerry. “Đầu anh chỉ toàn màu thôi hả anh bạn thân.” Sau đó, họ mỉm cười với nhau lâu hơn một chút quá dài.

Tôi lắc đầu và nói: “Phương thức đó có thể tốt hơn trong bản liệt kê. Nhìn cái này đi.”

public enum RGB {

red(0xff, 0x00, 0x00),

green(0x00, 0xff, 0x00),

blue(0x00, 0x00, 0xff),

white(0xff,0xff,0xff);

public final int r;

public final int g;

public final int b;

RGB(int r, int g, int b) {

this.r = r;

this.g = g;

this.b = b;

}

public boolean isBlue(){

return b > 2\*(r + g);

}

}

“Nó chạy được chứ?” Avery gặng hỏi.

“Chắc rồi.” Tôi nói và viết một kiểm thử để chứng minh.

public class RGBTest extends TestCase {

public void testIsBlue() throws Exception {

assertTrue(RGB.blue.isBlue());

assertFalse(RGB.red.isBlue());

assertFalse(RGB.green.isBlue());

}

“Ồ.” Jerry nói.

“Oa.” Jasmine nói.

“Đồ hãm tài!” Cả hai đều nói, cười khúc khích.

“Vậy chúng ta cũng có thể làm điều tương tự với Taco.Salsa enum, đúng không?”, Avery hỏi. “Để tôi thử.”

Và hắn ta bắt đầu thay đổi mọi thứ trên tường.

enum Color {

red, green, burntumber}

enum Salsa {

mild(10, Color.red),

medium(1000, Color.green),

hot(100000, Color.burntumber);

public final int scovilles;

public final Color color;

Salsa(int scovilles, Color color) {

this.scovilles = scovilles;

this.color = color;

}

public boolean isTooHotForMe() {

return scovilles > 30000;

}

}

“Và đây là các kiểm thử.” Hắn tiếp tục.

public class TacoTest extends TestCase {

public void testIsTooHot() throws Exception {

assertTrue(Taco.Salsa.hot.isTooHotForMe());

assertFalse(Taco.Salsa.medium.isTooHotForMe());

assertFalse(Taco.Salsa.mild.isTooHotForMe());

}

}

Avery ngây người đứng cười khi hắn bắt đầu nhìn chằm chằm vào thanh chỉ báo kiểm tra màu xanh lá cây trên tường. “Điều đó thật tuyệt.” Cuối cùng hắn ta cũng lên tiếng.

“Ừ.” Tôi đồng ý. “Nhưng nó còn tốt hơn thế. Những phương thức đó có thể đa hình. ”

“Cái gì?” Jerry nói.

“Gì cơ?” Jasmine nói.

“Đồ hãm tài!” (Cười khúc khích).

Avery và tôi đảo mắt với nhau.

“Ừ.” Nhìn đây này.

public class Taco {

enum Color {

red, green, burntumber}

enum Salsa {

mild(10, Color.red) {

public int sips() {

return 0;

}

},

medium(1000, Color.green){

public int sips() {

return 1;

}

},

hot(100000, Color.burntumber){

public int sips() {

return 5;

}

};

public final int scovilles;

public final Color color;

public abstract int sips();

Salsa(int scovilles, Color color) {

this.scovilles = scovilles;

this.color = color;

}

public boolean isTooHotForMe() {

return scovilles > 30000;

}

}

}

“Nó chạy được chứ?” Avery lại gặng hỏi lần nữa.

Mọi người nhìn hắn trong một giây, và sau đó bắt đầu cười. Sau một lúc, ngay cả Avery cũng cười.

“Chắc chắn, đây là các kiểm thử.” Tôi nói.

public void testSips() throws Exception {

assertEquals(0,Taco.Salsa.mild.sips());

assertEquals(1, Taco.Salsa.medium.sips());

assertEquals(5, Taco.Salsa.hot.sips());

}

Điều gì đã xảy ra tiếp theo tôi chỉ có thể mô tả trong đoạn mã:

wow() \* 3;

jinx() \* 3;

laugh() \* 4;

“Được rồi.” Jerry nói với một chút dứt khoát. “Việc đã đi quá xa rồi đấy. Tôi nghĩ đã đến lúc chúng ta trở lại làm việc. ”

Đã có cái gật đầu đồng tình, và mọi người hướng về phía cửa.

“Nhưng khoan đã!” Tôi nói. “Tôi vẫn chưa hỏi câu hỏi của tôi.”

“OK, cao thủ.” Jasmine trả lời. “Nhanh lên nào. Câu hỏi của mày là gì vậy?”

Tôi đứng tại chỗ và nhìn ba người họ. Sau đó tôi nói: “Anh sẽ sử dụng cái này để làm gì?”