# **Thợ lành nghề #48: Túi nâu 5 – State và “xung đột”**

Robert C. Martin

11/3/2006

…Tiếp nối phần trước

*Tháng 9, 1944*

*“Tiến sĩ* [*Oppenheimer*](https://vi.wikipedia.org/wiki/Robert_Oppenheimer)*, Tướng Groves, cảm ơn hai người đã đến đây.” Ánh nắng mặt trời buổi sáng chiếu qua cửa sổ phía đông nam của Phòng Bầu dục, mang đến luồng sáng tươi ấm như hòa hợp với tâm trạng của Tổng thống Wallace.*

*“Hôm qua tôi đã có cuộc nói chuyện rất thú vị với Tiến sĩ Von Braun. Ông ấy cho rằng chúng ta có thể xây dựng một hạm đội tàu không gian lớn liên hành tinh được trang bị năng lượng nguyên tử để giải cứu tàn dư nhân loại khỏi tầm ảnh hưởng của Clyde.”*

*Oppenheimer đang cựa quậy không yên trên ghế, nhưng Groves vẫn dồn hết sự chú ý của mình đặt lên người Tổng thống. “Các quý ông, tôi không giả vờ biết nếu ông ta nói đúng. Cái tôi biết là chúng ta cần thêm nhiều ý tưởng táo bạo như thế. Tôi muốn hai người tìm kiếm dự án Nimbus và đưa ra những ý tưởng, càng lớn, càng táo bạo càng tốt!”*

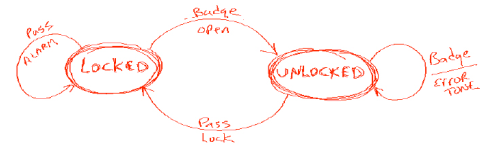
Thứ 6, 1/3/2002, 1102

Tôi chạy vội vì muộn muộn vài phút so với cuộc họp thường ngày lúc 11:00. Khi tôi bước vào phòng đã có vài người đang yên lặng ở trong căn phòng nhỏ đó. Jerry, Avery và Jasper đang quan sát Jasmine vẽ cái gì đó trên tường.

Jasmine làm tôi cứng người với đôi mắt màu xanh thẳm của nàng và noi: “Vừa kịp lúc đấy, Cao thủ. Tôi vừa định trả lời câu hỏi mà cậu đã hỏi tôi hôm qua đấy.”

“Xin lỗi.” Tôi lẩm bẩm khi tôi cố chen đến bên cạnh Avery. Hắn đảo mắt đầy ý tứ trong khi hất đầu về phía Jasmine. Tôi nhìn hắn cười khổ và đưa sự chú ý trở về phía nàng.

Cô ấy nói trong khi vẽ bức hình phía dưới lên tường. “Được rồi, đây là hệ thống trạng thái hữu hạn cho cổng an ninh phía trên cây cầu.”



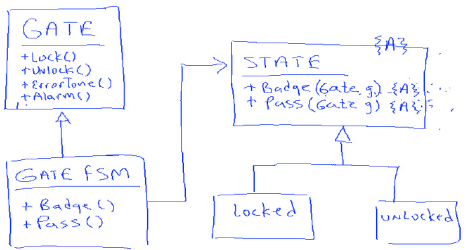
Avery nói: “Thật xin lỗi, nhưng đó không phải là cách nó hoạt động, nó…”

“Ừ, ừ, tôi biết.” Jasmine gắt gỏng ngắt lời, “nhưng tôi không có đủ diện tích tường cũng như là thời gian để hoàn thành FSM.” “Hãy kiên nhẫn giúp tôi. Hệ thống đơn giản hóa này bắt đầu ở trạng thái ĐÃ KHÓA. Nếu bạn đưa ra một dấu hiệu hợp lệ tại đó, cổng sẽ mở và FSM đi vào trạng thái MỞ KHÓA. Một khi bạn đã qua được cổng, nó sẽ tự động đóng và khóa lại, và FSM trở lại trạng thái ĐÃ KHÓA.”

“Sự chuyển đổi hai bên biểu thị cho các trường hợp lỗi. Cái bên phải chỉ ra trường hợp một người nào đó liên tục đưa ra tín hiệu ở một cánh cửa đã mở. Cổng phát ra âm thanh báo lỗi nhằm cố gắng báo cho kẻ ngu ngốc nào đó dừng việc đưa tín hiệu và đi qua cổng. Sự chuyển tiếp bên trái đưa ra trường hợp một người nào đó cố gắng xâm nhập vào cầu nối. Chuông báo động sẽ vang lên.”

Jasmine xoay người ngược lại hướng của tường để nhìn thấy những người trong nhóm. Mái tóc đen dài bung ra và ôm trọn bờ vai của nàng. “Mọi người đều hiểu chứ?” Cô ấy ra vẻ yêu cầu hơn là đang hỏi. Chúng tôi đều ngoan ngoãn gật đầu đáp lại.

“Được rồi, “ cô ấy tiếp tục, “giờ thì chúng ta đều biết rằng SMC1 sẽ tạo ra cái này bằng cách dùng mẫu Trạng thái.” Và nàng vẽ như sau:



Chúng tôi đều đã quen với cái kia nên lập tức gật đầu đồng tình ngay khi cô ấy quay ngoắt lại. Nàng cau mày nửa đùa nửa nghiêm túc và tiếp tục. “Có một vấn đề luôn gặp phải đó là chúng ta bắt buộc phải quản lý các instance của các derivative khóa và chưa khóa trong các biến tĩnh để ngăn chúng ta khỏi việc liên tục tạo thành và phá vỡ chúng. Một vấn đề nữa là đó là chúng ta không thể tạo riêng các phương thức và biến của GateFSM bởi vì chúng được điều khiển bởi các derivatives của State.”

Sau đó, Jasmine đẩy lên vài đoạn mã mà cô ấy đã viết sẵn cho cuộc thảo luận này.

*public class Gate {*

*void lock(){}*

*void unlock(){}*

*void errorTone() {}*

*void alarm() {}*

*}*

*public class GateFSM extends Gate {*

*State state = State.locked;*

*public void badge() {*

*state.badge(this);*

*}*

*public void pass() {*

*state.pass(this);*

*}*

*}*

*public abstract class State {*

*public abstract void badge(GateFSM g);*

*public abstract void pass(GateFSM g);*

*public static State locked = new Locked();*

*public static State unlocked = new Unlocked();*

*}*

*public class Locked extends State {*

*public void badge(GateFSM g) {*

*g.unlock();*

*g.state = State.unlocked;*

*}*

*public void pass(GateFSM g) {*

*g.alarm();*

*}*

*}*

*public class Unlocked extends State {*

*public void badge(GateFSM g) {*

*g.errorTone();*

*}*

*public void pass(GateFSM g) {*

*g.lock();*

*g.state = State.locked;*

*}*

*}*

“Lưu ý các biến tĩnh công khai ở các lớp State, và các phương thức không nội bộ và biến của Gate và GateFSM.”

“Xin lỗi,” Avery ngon ngọt nói. “nhưng cô có thể làm cho các phương thức và biến này nội bộ bằng cách sử dụng các inner class. Đây, để tôi chỉ cho…”

“NGỒI XUỐNG! Vâng, tôi biết, tôi biết chứ. Tôi chỉ là đang cố gắng đặt trọng tâm vào những mã mà SMC tạo ra. Hãy kiên nhẫn đợi tôi được ko!”

Avery ngồi xuống và bắt đầu giận dỗi.

Jasmine đưa cho mọi người một cái cười nhạt, mà đúng hơn là giống như ghê tởm, sau đó tiếp tục. “Đây là cách chúng ta có thể thực thi nó bằng cách dùng cú pháp enum mới mà Alphonse đã cho chúng ta xem hôm qua.”

*public class Gate {*

*private enum State {*

*LOCKED {*

*void badge(Gate g) {*

*g.unlock();*

*g.itsState = UNLOCKED;*

*}*

*void pass(Gate g) {*

*g.lock();*

*g.itsState = LOCKED;*

*}*

*};*

*abstract void badge(Gate g);*

*abstract void pass(Gate g);*

*}*

*private State itsState = State.LOCKED;*

*private void lock() {}*

*private void unlock() {}*

*private void errorTone() {}*

*private void alarm() {}*

*public void badge() {*

*itsState.badge(this);*

*}*

*public void pass() {*

*itsState.pass(this);*

*}*

*}*

*void pass(Gate g) {*

*g.alarm();*

*}*

*},*

*UNLOCKED {*

*void badge(Gate g) {*

*g.errorTone();*

*}*

“Trước tiên, mọi người sẽ nhận thấy rằng tất cả vừa vặn trong một lớp. Đó là nhỏ gọn, súc tích, chúng ta không có bất kỳ biến tĩnh để quản lý, và tất cả các phương pháp của Gate có thể nội bộ. Tôi nghĩ rằng đây là một cách sử dụng khá hay cho enums đa hình, phải không?

Tất cả chúng tôi lại gật đầu tán thành, ngoại trừ Avery, người vẫn tiếp tục hờn dỗi.

Jerry đứng lên và nói: “Ừ, nó rất thú vị. Enumeration cũng giống như một lớp, và mỗi một enumerator cũng giống như một instance tĩnh của một derivative của lớp đó. Nó đủ khó để đầu của các bạn quay cuồng, nhưng nó cũng chắc chắn rất thú vị. “

Sau đó, Jasper bắt đầu hành động. “Vâng thưa sếp, nó khá thú vị Jazzie.” Khuôn mặt của Jasmine đỏ ửng lên, nghiến chặt hàm, nhưng cô vẫn giữ cho mình bình tĩnh khi Jasper mải miết nói . “Nhưng tôi nghĩ những thứ thuộc aviator cũng nên lưu ý. Những vấn đề với mã SMC có thể được giải quyết mà không cần phải dùng tới các enum đa hình. Aviator đề xuất inner class nhưng tôi nghĩ rằng các câu tổ hợp các lệnh switch-case lồng nhau sẽ hoạt động tốt hơn.” Và hắn nhanh chóng viết mã sau lên tường.

*public class Gate {*

*private enum State {LOCKED, UNLOCKED}*

*private enum Event {BADGE, PASS}*

*private State state = State.LOCKED;*

*private void lock() {}*

*private void unlock() {}*

*private void errorTone() {}*

*private void alarm() {}*

*public void badge() {*

*processEvent(Event.BADGE);*

*}*

*public void pass() {*

*processEvent(Event.PASS);*

*}*

*private void processEvent(Event event) {*

*switch (event) {*

*case BADGE:*

*switch (state) {*

*case LOCKED:*

*unlock();*

*state = State.UNLOCKED;*

*break;*

*case UNLOCKED:*

*errorTone();*

*break;*

*}*

*break;*

*case PASS:*

*switch (state) {*

*case LOCKED:*

*alarm();*

*break;*

*case UNLOCKED:*

*lock();*

*state = State.LOCKED;*

*break;*

*}*

*break;*

*}*

*}*

*}*

“Xem nào,” Jasper nói một cách tự mãn, “không có biến tĩnh, không có phương thức nội bộ, gọn gàng và đơn giản.”

“Anh gọi đó là đơn giản à?” Jerry giễu cợt. “Những câu lệnh switch-case lồng nhau làm tôi đau đầu. Ôi quý ông, tôi đã từng tổ bảo trì các FSM lớn được viết theo kiểu này, và chúng hoàn toàn không vui chút nào, để tôi nói cho anh biết. “

“Ừ,” Jasmine xen vào, “và tất cả những logic đấy đang ngồi đó theo một cách nhất định sẽ phát triển và lớn lên, phát triển. Ick!”

“Jazzie … cậu vừa nói Ick chưa?”

“Đừng có mà gọi tôi là Jazzie, Jasper. Tôi đã nói với anh trước đây rồi. Ừ, ick! Mã đó sẽ bị phá hỏng trước khi anh kịp biết đến nó.”

Họ cứ tiếp tục như vậy trong một hoặc hai phút gì đó. Avery rời khỏi phòng trong cơn giận dữ. Rốt cuộc tôi cũng đứng dậy và vỗ vào tường cho đến khi tôi có sự chú ý của mọi người.

“Này mọi người,” tôi nói, nhe răng cười. “Tôi chưa thấy một kiểm thử nào cả. Làm cách nào để chúng ta biết rằng cái FSM nào cũng thực sự hoạt động?”

Ai cũng làu bàu và rời cuộc họp.

Trên đường trở về, Jerry bước tới cạnh tôi và lắc đầu, nói: “Tôi không biết chúng ta sẽ phải như thế nào với Avery đây.”