**Thợ lành nghề #18: Chậm mà chắc (SMCRemote – phần 8)**

Nhịp chậm chạp của người hướng dẫn mới làm cho tiến độ là gì đó của quá khứ, Alphonse muốn bay lên và học được tầm quan trọng của sự chậm rãi

“Được rồi cậu bé thân mến, tôi nghĩ đã đến lúc chúng ta bắt đầu làm việc với phần server của chương trình này, phải không cậu bé? Đến lúc này phần client làm việc có vẻ ngon lành như chúng ta thấy được, và một client mà không có server thì trông cô đơn quá. Cậu đồng ý không nào?”

Jean đã an toạ trong chiếc ghế bành của bà. Trong khi nói, bà vói tay vào trong giỏ và lôi ra bộ đồ đan. Chiếc sô (hay cái gì đó) đã dài ra một cách đáng kể sau buổi giải lao (khoảng thời gian tôi biết được về con và cháu của Jean).

“È, server?” Tôi chỉ có thể rặn ra hai từ này để trả lời một tràng phát biểu của Jean.

“Đúng vậy, cậu bé thân mến, server. Server sắp sửa nhận tệp tin mà cậu đã chăm chú gởi, lưu trữ vào đĩa và biên dịch nó với SMC. Nên nhớ cậu bé, chúng ta đang xây dựng một trình dịch từ xa cho SMC, State Machine Compiler. Bây giờ, cậu bé thân mến ơi, chúng ta nên viết test case đầu tiên ra sao?”

Test case gì đây hỡi? tôi gắng vắt óc nghĩ ngợi về chuyện này. Phía server của chương trình lắng nghe trên một socket và nhận các CompileFileTransaction objects. Những object này chứa các gói -1- tệp tin trong các sub-objects -2- FileCarrier. Server cần viết những tệp tin được gói này đến một nơi an toàn trên đĩa, biên dịch chúng và gởi kết quả ngược lại client trong một object gọi là CompilerResultsTransaction.

Điều đầu tiên tôi lo ngại là cách biên dịch các tệp tin này. Bằng cách nào đó, chúng ta cần gọi trình dịch và đòi hỏi nó biên dịch tệp tin đã lưu.

“Ừm…. mình có thể viết cái test case cho phân đoạn gọi trình dịch không nhỉ?”

Jean mỉm cười theo lối bà ngoại mỉm cười với đứa cháu chập chững bước đi đầu tiên. “Sao lại không, tất nhiên rồi cậu bé thân mến, khởi đầu ở điểm này thú vị đây. Cậu biết cách gọi trình dịch không? Hãy xem, tôi nghĩ chúng ta chỉ dựng đúng dòng lệnh và gọi nó thôi. Nào, cú pháp dòng lệnh ấy sao nhỉ? Đã lâu tôi không mó đến SMC, tôi chẳng nhớ rõ. Nghe đây cậu bé, gõ thử lệnh này: java smc.Smc. Đúng rồi đó cậu, bây giờ xem thử nó báo lỗi gì? Class def not found? Ồ vâng, chúng ta quên cái classpath. Đừng bị lão hoá nha cậu bé, cậu sẽ bắt đầu quên đủ thứ. Gõ cái này: java -cp C:/SMC/smc.jar smc.Smc -3-. Tốt rồi, vậy thông điệp ấy nói gì vậy?”

Thông điệp trên màn hình báo lỗi cách dùng và mô tả mọi thông số cần thiết để dùng SMC.

“È.. đó là một thông điệp chỉ dẫn cách dùng.”

“Đúng rồi cậu bé, đúng rồi. Và hình như muốn gọi SMC, chúng ta dùng lệnh sau: java -cp C:/SMC/smc.jar smc.Smc -f myFile.sm. Đồng ý chớ cậu bé? Sao cậu không thử viết một cái test case dùng để tạo dòng lệnh ấy?”

Thế nên tôi tạo một unit test class tên là SMCRemoteServerTest.

public class SMCRemoteServerTest extends TestCase {

public void testBuildCommandLine() throws Exception {

assertEquals(“java -cp C:/SMC/smc.jar -f myFile.sm”,

SMCRemoteServer.buildCommandLine(“myFile.sm”)) ;

}

}

Và tiện thể tôi viết luôn SMCRemoteServer.

public class SMCRemoteServer {

public static String buildCommandLine(String file) {

return null;

}

}

“Tốt lắm, cậu bé thân mến. Cái test bị hỏng như dự làm cho nó đạt là chuyện đơn giản, phải không nào? ”

“Èm…. hẳn nhiên rồi.”

public static String buildCommandLine(String file) {

return “java -cp C:/SMC/smc.jar smc.Smc -f ” + file;

}

“Tốt rồi cậu bé, chạy thử tôi xem nào? Tốt. Ôi! Lạ nhỉ, cái test bị hỏng rồi cậu bé thân mến, chuyện gì đã xảy ra nhỉ?”

Thông điệp trên màn hình là: expected:<……> but was: <…smc.Smc …>. Tôi cẩn thận xem xét đoạn mã và nhận ra tôi quên phần smc.Smc trong test case. “È… tôi đoán là tôi viết sai đoạn test.”

“Đúng rồi, đúng rồi. Vâng, đôi khi chúng ta viết test sai. Bugs -4- có thể xảy ra bất cứ nơi nào. Đó là lý do tại sao viết cả test lẫn mã nguồn là việc nên làm. Bằng cách này, cậu viết mọi thứ hai lần. Tôi có cậu cháu làm kế toán cho tàu vận tải, và cậu ấy luôn luôn lưu các chi tiết chuyển ngân vào hai tệp tin kế toán riêng biệt. Cậu ấy gọi nó là gì nhỉ? Ồ, vâng, sổ sách đôi. -5- Cậu ấy nói rằng cách này giúp cậu ngăn ngừa và lục tìm các sai sót. Và đây chính là chuyện chúng ta đang làm, phải không cậu bé thân mến. Chúng ta viết tất cả các hàm hai lần. Một lần trong phần test, và một lần trong mã nguồn. Và nó thật sự giúp chúng ta ngăn ngừa và tìm các sai sót, phải không nào?”

Trong khi bà ta tiếp tục với âm thanh của giọng nói đều đều, đơn tẻ, tôi đã sửa xong test case. Jean là một bà già dễ thương và rất giỏi nữa nhưng lỗ nhĩ của tôi lùng bùng với lối nói không ngừng nghỉ của bà.

public void testBuildCommandLine() throws Exception {

assertEquals(“java -cp C:/SMC/smc.jar smc.Smc -f myFile.sm”,

SMCRemoteServer.buildCommandLine(“myFile.sm”)) ;

}

Thay đổi này làm đoạn test đạt.

“Tuyệt vời, cậu bé, tuyệt vời; nhưng đoạn mã hơi rối, cậu nghĩ thế không? Hãy dọn dẹp nó trước khi tiếp tục. Dọn đống bừa bộn trước khi chúng bắt đầu (trở nên bừa bộn), tôi luôn nói như thế.”

Tôi xem đoạn mã và chẳng thấy có gì bừa bộn cả. Thế nên tôi ngoái lại (về hướng Jean ngồi) và nói: “Èm, bà thích khởi đầu từ đâu vậy?”

“Trời phật ơi, sao hỏi vậy cậu bé, nó đơn giản như chiếc mũi trên khuôn mặt cậu vậy thôi! hàm buildCommandLine đó cần chỉnh đốn một chút. Chuỗi string trong dòng trả về đầy tính giả định và nhất định sẽ làm ai đó đọc đoạn mã này sẽ bối rối. Chúng ta không muốn làm cho ai bối rối cả, phải không nào? Tôi muốn nói là chúng ta không muốn làm như thế. Đây cậu, đưa cho tôi cái bàn đánh.”

Bà đặt bồ đồ đan xuống (hình như mảnh đan bắt đầu phần cổ tay của cái khăn choàng) và với sự cố gắng rõ rệt để điều khiển bàn đánh, bà gõ chậm rãi nhưng đầy chủ định như thể mỗi cú gõ làm bà đau đớn. Rốt cuộc bà thay đổi đoạn trả lại như sau:

return “java -cp ” + “C:/SMC/smc.jar” + ” ” + “smc.Smc” + ” -f ” + file;

Rồi bà chạy thử đoạn test và nó đạt.

“Rồi đó cậu bé thân mến, cậu thấy chưa? Chúng ta cần thay thế đoạn string ấy bằng các giá trị bất biến để giải thích dòng lệnh đã được hình thành bằng cách nào. Cậu muốn làm thế không, hay để tôi?”

Vẻ đau đớn của bà đã đủ trả lời. Tôi vớ lấy bàn đánh và thêm vào các giá trị bất biến tôi nghĩ là bà muốn đưa vào.

public class SMCRemoteServer {

private static final String SMC\_CLASSPATH = “C:/SMC/smc.jar”;

private static final String SMC\_CLASSNAME = “smc.Smc”;

public static String buildCommandLine(String file) {

return "java -cp "

+ SMC\_CLASSPATH

+ " " +SMC\_CLASSNAME

+ " -f " + file;

}

}

“Đúng rồi cậu bé thân mến, đích thị điều tôi muốn. Cậu không nghĩ rằng như thế sẽ giúp những người khác sẽ dễ hiểu hơn sao? -6- Tôi biết chắc nếu tôi là họ, tôi sẽ cảm thấy dễ hiểu hơn. Bây giờ, cậu bé thân mến, sao cậu và tôi không đi đến phòng giải lao để cho đầu óc nghỉ ngơi đôi chút nhỉ?”

Điều gì đó nảy ra trong đầu tôi.

“Jean, chúng ta chưa làm được tí gì cả!”

“Sao cơ, ý cậu thế nào, chúng ta đã hoàn thành khá nhiều rồi.”

Tôi chẳng nghĩ đến chuyện tôi đang nói với ai. Tôi vẫn một mực lảm nhảm một cách đần độn. “Chúng ta chỉ mới xong một cái hàm ngu xuẩn mà thôi. Nếu mình giữ cái đà “con sên” này -7-, ông C sẽ thuyên chuyển chúng ta đến bộ phận canh tác để dọn chuồng thôi!” -8-

“Thật thế sao cậu bé, tại sao cậu lại nghĩ ngớ ngẩn đến như vậy nhỉ? Tôi làm việc với văn phòng này hơn ba mươi năm, cậu bé thân mến, và chưa bao giờ có ai đề cập đến chuyện dọn chuồng.”

Bà ta ngưng chốc lát như thể bà đang tổng hợp các ý nghĩ. Rồi bà nhìn tôi với cái nhìn đôn hậu nhưng nghiêm khắc hiện rõ trên khuôn mặt.

“Alphonse thân mến, cách duy nhất để hoàn thành chương trình này nhanh chóng là làm tối đa những gì cậu có thể. Nếu cậu vội vã, hay nếu cậu đi đường tắt, cậu phải trả giá xứng đáng trong giai đoạn tìm lỗi. Ông C trả công cho thời gian của cậu, cậu bé, và ông ấy muốn kết quả tốt nhất cậu có thể làm được. Mã nguồn chính là sản phẩm của cậu, cậu bé thân mến, và ông ấy không muốn trả cho thứ sản phẩm kém chất lượng. Ông ta không muốn nghe rằng cậu hoàn tất được ngày làm do vấy vá cho xong chuyện bởi vì ông ta biết cậu sẽ trả một giá rất đắt cho tính vấy vá. Ông ấy chỉ muốn thứ mã nguồn tốt nhất mà cậu có thể viết được. Ông ta muốn mã nguồn sạch, có nghĩa lý và đã được thử nghiệm cẩn thận ở mức tối đa cậu có thể tạo ra. Và rồi, cậu bé thân mến, ông C thừa biết rằng nếu cậu làm như vậy, cậu sẽ hoàn tất nhanh chóng hơn những cậu ngốc non choẹt kia cứ ngỡ chúng có thể xong chuyện nhanh hơn với thói nhếch nhác. Bây giờ, mình hãy đi làm một chén trà nhỉ?”

-1- encapsulated: một thể trạng được gói bên trong một thể trạng, phương tiện nào khác. Ví dụ, một Vector object “encapsulate” các object bên trong và các object này có thể khác nhau về tính chất và thể loại. Hay nói ngược lại, các object bên trong một Vector được encapsulated.

-2- sub-objects: các thể trạng (object) ngầm. Ví dụ, một Vector chứa các object và các object này chứa những object khác bên trong.

-3- Trong nguyên bản tác giả cho phép tải smc.jar từ: http://www.objectmentor.com/resource…/smc\_java

-4- Bugs: lỗi và vấn đề trục trặc trong lập trình. Từ “bug” này hết sức thông dụng và đặc thù cho nên tôi dùng từ nguyên thuỷ thay vì cố dịch sang một từ tương đương tiếng Việt (như một số từ khác đã được dùng trong suốt series Craftsman này).

-5- dual entry bookkeeping: tệp tin kế toán được ghi nhận và lưu giữ hai nơi khác nhau. Trên thực tế, tôi không rõ có phương pháp kế toán như thế này không nhưng đây là chi tiết dùng để liên hệ đến vấn đề viết test và viết code nên được ghi nhận là một chi tiết quan trọng.

-6- Câu nói ở dạng negative (negative form) đặc biệt được nhân vật Jean dùng thường xuyên. Có lẽ tác giả muốn tạo nhân vật Jean này với cá tính rất mềm mỏng và nhẹ nhàng trong khi trao đổi. Tuy nhiên, những điều Jean đưa ra rất xác thực và cần thiết cho dù lối nói của bà ta rất dông dài.

-7- snail’s pace: mức độ, tốc độ của con sên. Ý nói sự việc tiến triển rất chậm chạp.

-8- Clean out the Dribin cages: tôi không tìm được nguồn gốc của cụm từ này. Độc giả có ai rõ, xin góp ý.

**Bài tiếp:** [Thợ lành nghề #19: Kiên nhẫn (SMCRemote Phần 9)](https://codegym.vn/blog/2018/08/21/tho-lanh-nghe-19-kien-nhan-smcremote-phan-9/)

**Bài trước:** [Thợ lành nghề #17: Gọi bảo kê (SMCRemote – phần 7)](https://codegym.vn/blog/2018/08/14/tho-lanh-nghe-17-goi-bao-ke-smcremote-phan-7/)

**Tác giả:** Robert C. Martin

**Người dịch:** Hoàng Ngọc Diêu