# **Thợ lành nghề #21: Chắp vá (SMCRemote – phần 11)**

“Giờ thì, Alphose thân mến, cậu nghĩ chúng ta nên chạy bản dịch ở đâu?”

Tôi vẫn đang quay cuồng vì vừa mới xóa tất cả mã mà tôi đã viết nên không kịp nghe thấy câu hỏi.

“Tôi xin lỗi, Jean, cô vừa nói gì vậy?”

“Chà, chúng ta không thể chạy trình biên dịch SMC ở những chỗ chỗ cũ được, đúng chứ? Không, chúng ta phải chạy nó trong một thư mục mới. Chúng ta phải sao chép tất cả các tệp sắp có vào trong thư mục đó, chạy trình biên dịch, và sau đó gửi tất cả các tập tin kết quả lại cho người gọi”

“Có tất cả bao nhiêu tệp kết quả ở đó vậy?” Tôi hỏi. “Tôi nghĩ rằng SMC chỉ tạo ra một tập tin đuôi .java”

“Trình tạo mã Java làm như vậy. Nhưng, trình tạo mã C++ tạo ra tệp đuôi .h và .acc. Các trình tạo mã khác cũng có thể tạo ra các loại tệp khác nhau. Tôi e rằng sẽ khó có thể lường trước được. Dù sao thì, cậu bé à, chúng ta muốn trình biên dịch chạy trong môi trường thông thoáng, và điều đó đồng nghĩa với một thư mục rỗng chỉ có tệp đầu vào đuôi .sm hiện tại”

Tôi nghĩ về điều này trong một lúc, và nhận ra rằng bà ấy đã đúng. Tôi bắt đầu thấy rất vui vì đã xóa đoạn mã cũ đó. “Vậy chúng ta nên có một thư mục đặc biệt để chạy các trình biên dịch bên trong?”

“Còn hơn thế cậu bé ạ. Tôi nghĩ chúng ta nên tạo một thư mục làm việc ngay sau khi nhận được CompileFileTransaction. Sau đó chúng ta nên ghi tập tin đầu vào thư mục đó. Nó sẽ là tập tin duy nhất ở đó, và mọi thứ sẽ thuận tiện và gọn gàng hơn, cậu có nghĩ thế không? Và sau đó chúng tôi có thể chạy trình biên dịch, xóa các tập tin đầu vào, và thu thập bất kỳ tập tin nào còn lại và đặt chúng vào ComplierResultsTransaction thành các gói hàng trở lại cho khách hàng. Ồ, điều này nghe thật thú vị. Tại sao cậu không viết một Unit Test nhỏ thôi, để chắc chắn rằng chúng ta có thể tạo được một thư mục mới, được chứ cậu bé?”

Tôi lấy bàn phím và bắt đầu gõ. Trong khi tôi đang gõ, Jean kéo 1 chồng vải vuông màu 6 màu sắc rực rỡ ra khỏi túi và bắt đầu đan chúng lại với nhau.

| 1  2  3  4  5 | public void testMakeWorkingDirectory() throws Exception {  File workingDirectory = SMCRemoteServer.makeWorkingDirectory();  assertTrue(workingDirectory.exists());  assertTrue(workingDirectory.isDirectory());  } |
| --- | --- |

“Ồ, nhìn có vẻ ổn đấy cậu bé. Trước tiên, cậu tạo ra các thư mục và sau đó cậu đảm bảo rằng nó tồn tại, và chỉ ra rằng, nó đó, nó là một thư mục. Bây giờ, hãy làm cho nó có kết quả kiểm thử tốt đi”

Tôi biết là Jean không có ý khiêu khích mình, nhưng từng từ đều làm tôi không thoải mái. Tôi lờ đi và viết mã để qua được lần kiểm thử. Đầu tiên, tôi nhấp vào chức năng còn thiếu và để IDE tự thêm vào cho tôi. Sau đó, tôi điền vào bằng mã sau đây:

| 1  2  3  4  5 | public static File makeWorkingDirectory() {  File workingDirectory = new File("workingDirectory");  workingDirectory.mkdir();  return workingDirectory;  } |
| --- | --- |

Tất nhiên là đoạn mã này đã vượt qua lần kiểm thử ngay trong lần thử đầu tiên, vậy nên tôi ung dung ngồi và chờ các hướng dẫn tiếp theo của Jean.

“Cậu đang đợi gì sao?”

“Tôi đang chờ cô chỉ cho tôi biết phải làm gì tiếp theo”

“Thật sao? Chúa ơi, tôi chỉ là một bà cô vừa ngốc vừa nói nhiều thôi. Chẳng lẽ cậu không có bất kỳ ý tưởng nào của riêng cậu hay sao? Một người đàn ông trẻ tuổi thông minh như cậu phải luôn tràn đầy ý tưởng. Cậu nghĩ chúng ta nên làm gì tiếp theo?”

Lần này thậm chí còn khiêu khích nhiều hơn so với trước. Jasmine và Jerry có thể nghĩ rằng đó là một loại địa vị nghề nghiệp với Jean, nhưng tôi thấy nó khá khó chịu.

“Có lẽ chúng ta nên xóa thư mục khi chúng ta đã xong việc với nó” Tôi gắt lên.

“Tôi nghĩ đó là 1 ý tưởng tuyệt vời. Chúng ta sẽ phải chắc chắn rằng toàn bộ thư mục bị xóa cùng với tất cả tệp bên trong nó”

Vì vậy, tôi đã viết đoạn Unit Test trong khi Jean tiếp tục đan các ô vuông lại với nhau.

| 1  2  3  4  5  6  7  8  9 | public void testDeleteWorkingDirectory() throws Exception {  File workingDirectory = SMCRemoteServer.makeWorkingDirectory();  File someFile = new File(workingDirectory, "someFile");  someFile.createNewFile();  assertTrue(someFile.exists());  SMCRemoteServer.deleteWorkingDirectory(workingDirectory);  assertFalse(someFile.exists());  assertFalse(workingDirectory.exists());  } |
| --- | --- |

Jean nhìn lên từ chiếc may của mình và nói: “Tốt lắm, Alphonse. Đó chính là điều chúng ta muốn”

Ngay sau đó, tôi đã viết đoạn mã sau:

| 1  2  3 | public static void deleteWorkingDirectory(File workingDirectory) {  workingDirectory.delete();  } |
| --- | --- |

Nhưng lần này nó không qua được phép kiểm thử. Phép kiểm thử thông báo rằng một vài file vẫn tồn tại.

“Tôi nghĩ cậu phải xóa tất cả các tệp trong thư mục trước khi cậu có thể xóa chính thư mục đó, Alphonse à. Tôi chưa bao giờ hiểu tại sao chúng lại tạo ra quy tắc đó. Tôi nghĩ sẽ tốt hơn nếu chúng tin tưởng cậu khi cậu nói cậu muốn thư mục bị xóa. Xem nào, tôi đoán cậu sẽ phải kiểm thử qua tất cả tập tin và xóa từng tập tin một, cậu bé à”

Thế nên tôi đã viết như sau:

| 1  2  3  4  5  6  7 | public static void deleteWorkingDirectory(File workingDirectory) {  File[] files = workingDirectory.listFiles();  for (int i = 0; i < files.length; i++) {  files[i].delete();  }  workingDirectory.delete();  } |
| --- | --- |

Đoạn mã này đã qua được lần kiểm thử. Tuy nhiên tôi lại không thích nó. “Jean, cái này có tác dụng, nhưng nếu như còn có các thư mục con thì sao?”

“Chính xác, cậu bé à, chúng ta không thể chắc rằng một trong những máy tạo mã SMC sẽ không tạo ra một hay hai thư mục con. Tôi nghĩ rằng tốt hơn hết, cậu nên tăng đoạn kiểm thử lên để xử lý trường hợp này”

Vậy nên, tôi đã thay đổi đoạn kiểm thử như sau:

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | public void testDeleteWorkingDirectory() throws Exception {  File workingDirectory = SMCRemoteServer.makeWorkingDirectory();  File someFile = new File(workingDirectory, "someFile");  someFile.createNewFile();  assertTrue(someFile.exists());  File subdirectory = new File(workingDirectory, "subdirectory");  subdirectory.mkdir();  File subFile = new File(subdirectory, "subfile");  subFile.createNewFile();  assertTrue(subFile.exists());  SMCRemoteServer.deleteWorkingDirectory(workingDirectory);  assertFalse(someFile.exists());  assertFalse(subFile.exists());  assertFalse(subdirectory.exists());  assertFalse(workingDirectory.exists());  } |
| --- | --- |

Không ngoài dự kiến, lần kiểm thử này đã thất bại ngay lập tức. subFile đã không bị xóa. Tiếp theo, tôi đã thay đổi mã để qua được đoạn kiểm thử

| 1  2  3  4  5  6  7  8  9 | public static void deleteWorkingDirectory(File workingDirectory) {  File[] files = workingDirectory.listFiles();  for (int i = 0; i < files.length; i++) {  if (files[i].isDirectory())  deleteWorkingDirectory(files[i]);  files[i].delete();  }  workingDirectory.delete();  } |
| --- | --- |

“Ồ tốt rồi, cậu bé ạ, rất tốt. Kết quả rất mỹ mãn. Giờ thì, Alphonse, tiếp theo sẽ là gì?”

Có cái gì đó làm tôi lo lắng, tôi liền nói: “Jean, đây là một máy chủ phải không?”

“Đúng rồi cậu bé. Nó đợi trên một ổ cắm cho mã khách hàng để kết nối với nó”

“Có thể có bao nhiêu khách hàng ở đó?”

“Ồ, có thể có hàng tá, có thể nhiều hơn”

“Vậy khi nếu chúng ta có hai khách hàng kết nối cùng một lúc, chúng ta sẽ thử tạo ra thư mục con hai lần, đúng chứ?”

“Tất nhiên rồi, tôi cho rằng đúng là như thế. Điều đó có thể gây ra một chút rắc rối cho chúng ta phải không. Chúng ta có nhiều hơn một phiên sao chép các tập tin vào trong thư mục, và các phiên này chạy các biên dịch trong thư mục, rồi sao chép các tập tin ra khỏi thư mục và cũng xóa luôn thư mục. Điều đó thật khủng khiếp phải không? Cậu nghĩ chúng ta nên giải quyết vấn đề này thế nào hả Alphonse?”

“Chuyện gì sẽ xảy ra nếu như mỗi thư mục làm việc chỉ có một tên duy nhất? Bằng cách đó, mỗi kết nối sắp đến sẽ có thư mục riêng để làm việc”

“Đó là ý tưởng rất hay đấy, Alphonse. Sao cậu không viết các test cases cho điều đó?”

Tôi cũng nghĩ đó là một ý tưởng tuyệt vời, vì vậy tôi đã viết đoạn kiểm thử sau đây

| 1  2  3  4  5 | public void testWorkingDirectoriesAreUnique() throws Exception {  File wd1 = SMCRemoteServer.makeWorkingDirectory();  File wd2 = SMCRemoteServer.makeWorkingDirectory();  assertFalse(wd1.getName().equals(wd2.getName()));  } |
| --- | --- |

Nó thất bại như dự kiến. Để làm cho nó qua được, tôi cần một cách nào đấy để tạo ra những cái tên duy nhất. “Tôi nghĩ rằng tôi có thể mất nguyên ngày để tạo ra một tên duy nhất”. Tôi đang nói với chính tôi hơn là nói với Jean.

“Ừ, sao cậu không thử làm luôn đi?”

Tôi đã viết đoạn mã sau:

| 1  2  3  4  5  6  7  8  9 | public static File makeWorkingDirectory() {  File workingDirectory = new File(makeUniqueWorkingDirectoryName());  workingDirectory.mkdir();  return workingDirectory;  }    private static String makeUniqueWorkingDirectoryName() {  return "workingDirectory" + System.currentTimeMillis();  } |
| --- | --- |

Đoạn kiểm thử đã thành công, nhưng tôi thấy ánh nhìn ẩn ý trên mặt Jean. Vì nên tôi nhấn nút kiểm thử một lần nữa. Bà ấy gật đầu trong khi đoạn kiểm thử tiếp tục thành công. Tôi nhấn nút kiểm thử một vài lần nữa, và chắc chắn có sự thất bại bất chợt.

“Ờ thì”, tôi nói, “nếu hai cuộc gọi đến makeUniqueWorkingDirctionName quá gần nhau, thì currentTimeMillis sẽ trả về cùng một lúc và các tên sẽ không còn là duy nhất nữa.”

“Ừ, cậu bé à.” Jean nói

“Tôi cho rằng tôi chỉ có thể sử dụng một số nguyên tĩnh và tăng dần cho mỗi tên, nhưng sau đó những tên này sẽ bắt đầu lại từ đầu mỗi lần chúng ta khởi động lại máy chủ. Điều đó cũng có thể gây ra xung đột”

“Suy nghĩ rất thấu đáo” Jean nói

“Tại sao chúng ta không sử dụng cả hai kỹ thuật” và tôi ngừng nói chuyện, bắt đầu gõ

| 1  2  3  4  5  6  7  8  9  10  11 | public class SMCRemoteServer {  ...  private static int workingDirectoryIndex = 0;  ...  private static String makeUniqueWorkingDirectoryName() {  return "workingDirectory" +  System.currentTimeMillis() + "\_" +  workingDirectoryIndex++;  }  ...  } |
| --- | --- |

Bây giờ các đoạn kiểm thử được lặp lại và tôi khá hài lòng với bản thân mình. Tôi chắc rằng Jean cũng sẽ muốn được nghỉ sớm, và chúng tôi đã hoàn thành công việc. Mặt khác, những gì chúng tôi đã làm được là một cái gì đó rất khác với mã mà tôi đã xóa trước trước khi Jean bước vào.

“Alphonse”

“Vâng, Jean”

“Cậu có thể xóa tất cả những thư mục còn sót lại trong các thư mục đang làm việc không? Chúng gây ra khá nhiều sự lộn xộn trong thư mục quan trọng của chúng ta. Tôi ghét sự lộn xộn, phải không? Tôi ghét nó trong mã của mình, tôi ghét nó trong phòng của mình và tôi cũng ghét nó trong các thư mục của mình”

Tôi nhìn thư mục và nhận thấy rằng có hàng tá các thư mục đang làm việc còn sót lại. Tôi ngẩng đầu khi nhận ra rằng các phép kiểm thử của tôi đã không tự dọn dẹp. Tôi ngượng nghịu thực hiện những thay đổi cần thiết.

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | public void testMakeWorkingDirectory() throws Exception {  File workingDirectory = SMCRemoteServer.makeWorkingDirectory();  assertTrue(workingDirectory.exists());  assertTrue(workingDirectory.isDirectory());  SMCRemoteServer.deleteWorkingDirectory(workingDirectory);  }    public void testWorkingDirectoriesAreUnique() throws Exception {  File wd1 = SMCRemoteServer.makeWorkingDirectory();  File wd2 = SMCRemoteServer.makeWorkingDirectory();  assertFalse(wd1.getName().equals(wd2.getName()));  SMCRemoteServer.deleteWorkingDirectory(wd1);  SMCRemoteServer.deleteWorkingDirectory(wd2);  } |
| --- | --- |

“Trông tốt hơn rất nhiều rồi đó, cậu bé. Cảm ơn cậu. Bây giờ thì, tôi nghĩ đến giờ giải lao rồi phải không?”

“Thật tốt quá”

Jean đã đan được 16 ô vuông với nhau thành lưới 4×4

Vẫn còn tiếp…….

Các đoạn code của Alphonse và Jean đã viết có thể được lấy từ:

[www.objectmentor.com/resources/articles/CraftsmanCode/Craftsman\_21\_Patchwork.zip](http://www.objectmentor.com/resources/articles/CraftsmanCode/Craftsman_21_Patchwork.zip)

Bài tiếp: [Thợ lành nghề #22: Trừng mắt (SMCRemote – phần 12)](https://codegym.vn/?p=5421&preview=true)

Bài trước: [Thợ lành nghề #20: Chổng gọng -1- (SMCRemote – phần 10)](https://codegym.vn/blog/2018/08/22/tho-lanh-nghe-20-chong-gong-1-smcremote-phan-10/)