# **Thợ lành nghề 23: SMCRemoteServer (Phần 13 – Lộn xộn)**

#### 20/2/2002, 14:00

Avery nhìn tôi với nụ cười tự mãn trên cái mặt đầy mụn của hắn. Lông mày màu đỏ rậm của hắn ta tăng thêm thái độ cho cái nhìn ấy. Hắn đã có được sự tự tin trong một vài giờ qua, và sự lắp bắp của hắn đã biến mất. “Được rồi, vậy bây giờ mấy chủ thực thi các biên dịch trong một thư mục trống. Bây giờ tao nghĩ rằng đã đến lúc kiểm tra toàn bộ máy chủ từ đầu đến cuối”

“Ý mày là mày muốn mở một socket với máy chủ, gửi trong ComplierFileTransaction và xem xem liệu có một ComplierResponseTransaction tương thích đi ra không?”

“Chính xác, Alphonse, chính xác là thế.” Khi có sự tự tin, hắn ta bắt đầu nói chuyện một cách kì quặc. Có chút lạ lẫm, nhưng cũng có chút thú vị. Ra vẻ hợp tác, tôi nói: “Tao hiểu rồi, Avery. Tao nghĩ mày nên tạo một trường hợp kiểm thử thích hợp, sau đó tao sẽ cố gắng thỏa mãn nó.”

“Đề xuất rất hay, Alphonse. Tao sẽ tiến hành luôn chứ?”

“Làm đi thôi.”

Avery cắm mặt vào màn hình, xoay các khớp tay và vươn vai trước khi chạm vào bàn phím. Hắn ta gõ:

public void testServerEndToEnd() throws Exception {

}

Hắn dừng lại lưng chừng và nói: “Mày thích cái tên vậy không?” “Tất nhiên”, tôi đáp lại. Tôi cố tình nhấn vào chữ đó. Hắn ta tiếp tục gõ trong khi nói

“Mục tiêu đầu tiên của chúng ta là khởi tạo SMCRemoteServer và liên kết nó với cổng thuận tiện cho các trường hợp kiểm thử”

public void testServerEndToEnd() throws Exception {

**SMCRemoteServer server = new SMCRemoteServer(999);**

}

“Tao đánh giá cao đoạn mã đó, nhưng hãy nhớ rằng nó không thể được biên dịch. Dường như không có constructor nào có một int.”

“Phát hiện tốt lắm, Alphonse, phát hiện rất tốt. Thế nên chúng ta phải tạo một constructor như thế.”

public class SMCRemoteServer {

…

**public SMCRemoteServer(int port) {**

**}**

…

}

“Tốt lắm, Avery. Tao nghĩ lần kiểm tra này sẽ được với cấu trúc hiện tại của nó.” “Tao tin là mày sẽ đúng, Alphonse. Chúng ta thử chứ?”

“Ừ, tao cũng nghĩ thế.”

Avery nhấn nút kiểm thử và thanh màu xanh lóe lên trên mà hình thông báo rằng lần kiểm thử đã thành công.

“Đúng như dự đoán”. Avery nói với giọng tự mãn. “Tiếp theo chúng ta sẽ kiểm tra kết nối với [socket](https://en.wikipedia.org/wiki/Network_socket) như sau.”

public void testServerEndToEnd() throws Exception { SMCRemoteServer server = new SMCRemoteServer(999);

**Socket client = new Socket(“localhost”, 999);**

}

“Nhưng mà tao lại mong nó thất bại, Avery.”

“Tao cũng muốn thế. Chúng ta sẽ không bao giờ tạo ra được server socket.” Nhấn lại nút kiểm thử thì rất nhanh chóng, trên màn hình hiện lên thanh màu đỏ. Chúng tôi nhìn nhau và gật đầu, hài lòng với sự tâm đầu ý hợp này.

Tôi đưa tay ra phía trước, như một cử chỉ lịch sự để yêu cầu Avery đưa bàn phím theo hướng của tôi chỉ. Hắn ta làm theo một cách dứt khoát và cái nhìn thấu hiểu.

“Cảm ơn Avery. Để vượt qua lần kiểm thử này, chúng ta sẽ dùng một tiện ích tên là SocketService mà Jerry và tao đã viết tuần trước.”

public SMCRemoteServer(int port) throws Exception {

**SocketService service = new SocketService(port, new SocketServer() { public void serve(Socket theSocket) {**

**try {**

**theSocket.close();**

**} catch (IOException e) {**

**}**

**}**

**});**

}

“Tốt lắm, Alphonse. Tao có nên chạy đoạn kiểm thử không?”

“Chắc chắn rồi!” Tôi trượt bàn phím trở lại phía hắn ta.

Chúng tôi lại mỉm cười và gật đầu – giống như đang trình diễn vậy – khi thanh màu xanh lóe lên màn hình.

“Giờ thì,” Avery hắng giọng, “máy chủ nên đáp trả lại bằng một tin nhắn kết nối.”

“Nó nên như thế, Avery.” Tôi nói, nhớ lại hôm thứ năm tuần trước. “Jerry và tao quyết định rằng tin nhắn sẽ là một chuỗi bắt đầu với SMCR.”

“Vậy thì, rất dễ để kiểm tra.”

public void testServerEndToEnd() throws Exception { SMCRemoteServer server = new SMCRemoteServer(999); Socket client = new Socket(“localhost”, 999);

**ObjectInputStream is = new ObjectInputStream(client.getInputStream()); String header = (String) is.readObject(); assertTrue(header.startsWith(“SMCR”));**

}

“Ừm, trông khá ổn đấy. Khá thích hợp.” Tôi nói và cầm lấy bàn phím khi Avery đưa đẩy nó theo hướng tôi chỉ. Tôi chạy đoạn kiểm thử và chú ý với một sự thỏa mãn khi nó thất bại vì EOFException. Giờ thì, để nó thành công, chúng tôi chỉ tạo ra một chuỗi thích hợp và gửi nó ra khỏi socket – như thế này!”

public SMCRemoteServer(int port) throws Exception {

SocketService service = new SocketService(port, new SocketServer() { public void serve(Socket theSocket) {

try {

**ObjectOutputStream os =**

**new ObjectOutputStream(theSocket.getOutputStream()); os.writeObject(“SMCR”);**

theSocket.close();

} catch (IOException e) {

}

}

});

}

Đoạn kiểm thử báo về một thanh màu xanh, và chúng tôi lại lặp lại nghi thức điệu cười và cả cái gật đầu. Tôi trả bàn phím lại cho Avery giữ như thể tôi đang vượt qua rào chắn để đến chỗ địch thủ lúc bắt đầu trận đấu. “Thưa ngài, vũ khí của ngài đây.”

“Tuân lệnh, thưa ngài”. Hắn ta đáp lại. “Tao nghĩ bây giờ chúng ta bắt buộc phải chuẩn bị gửi ComplierFileTransaction. Điều này sẽ yêu cầu chúng ta trước hết phải viết một tệp nguồn có chứa mã mà trình biên dịch SMC có thể biên dịch được.”

“À, quả thật là vậy, Jean và tao đã tạo ra một tệp như thế vào sáng nay. Mã nằm trong hàm testExecuteCommand(). Chúng ta sẽ có thể nén nó thành một hàm của riêng nó tên là writeSourceFile().”

private File writeSourceFile(String theSourceFileName) throws IOException { File sourceFile = new File(theSourceFileName);

PrintWriter pw = new PrintWriter(new FileWriter(sourceFile)); pw.println(“Context C”);

pw.println(“FSMName F”);

pw.println(“Initial I”);

pw.println(“{I{E I A}}”);

pw.close();

return sourceFile;

}

“Xuất sắc! Cảm ơn mày, Alphonse. Vậy giờ tao có thể tạo ComplierFileTransaction, gửi nó đến máy chủ, và chờ ComplierResultsTransaction trở lại – đúng chứ?”

“Tao phải nói là đúng như vậy đấy, Avery”

public void testServerEndToEnd() throws Exception { SMCRemoteServer server = new SMCRemoteServer(999); Socket client = new Socket(“localhost”, 999);

**ObjectInputStream is = new ObjectInputStream(client.getInputStream());** ObjectOutputStream os = new ObjectOutputStream(client.getOutputStream()); String header = (String) is.readObject(); assertTrue(header.startsWith(“SMCR”));

**File sourceFile = writeSourceFile(“mySourceFile.sm”);**

**CompileFileTransaction cft = new CompileFileTransaction(“mySourceFile.sm”);**

**os.writeObject(cft);**

**os.flush();**

**CompilerResultsTransaction crt =**

**(CompilerResultsTransaction) is.readObject(); assertNotNull(crt);**

}

Với thanh màu đỏ từ lần kiểm thử thất bại, Avery nói: “Alphonse, mày có thấy đoạn mã này diễn tả đúng ý của tớ chứ.”

“Ừ, tao nghĩ nó diễn đạt rất rõ ràng. Và giờ thì nếu mày đưa tao bàn phím, tao sẽ làm cho nó thành công”

public void serve(Socket theSocket) {

try {

ObjectOutputStream os =

new ObjectOutputStream(theSocket.getOutputStream());

**ObjectInputStream is =**

**new ObjectInputStream(theSocket.getInputStream());** os.writeObject(“SMCR”);

**os.flush();**

**CompileFileTransaction cft =**

**(CompileFileTransaction) is.readObject(); CompilerResultsTransaction crt = new CompilerResultsTransaction(); os.writeObject(crt);**

**os.flush();**

theSocket.close();

} catch (Exception e) {

}

}

“Thế này chắc đủ để nó qua lần kiểm thử rồi”

“Làm tốt lắm Alphonse!”

“Cảm ơn Avery. Thật ra tao chẳng làm gì với nó cả”

“Trong trường hợp này, tao sẽ đưa cho mày một thử thách.” Hắn ta lấy bàn phím và bắt đầu gõ lại lần nữa

public void testServerEndToEnd() throws Exception { SMCRemoteServer server = new SMCRemoteServer(999); Socket client = new Socket(“localhost”, 999);

ObjectInputStream is = new ObjectInputStream(client.getInputStream()); ObjectOutputStream os = new ObjectOutputStream(client.getOutputStream()); String header = (String) is.readObject(); assertTrue(header.startsWith(“SMCR”));

File sourceFile = writeSourceFile(“mySourceFile.sm”); CompileFileTransaction cft = new CompileFileTransaction(“mySourceFile.sm”); os.writeObject(cft);

os.flush();

CompilerResultsTransaction crt =

(CompilerResultsTransaction) is.readObject(); assertNotNull(crt);

**File resultFile = new File(“F.java”);**

**assertFalse(resultFile.exists());**

**crt.write();**

**assertTrue(resultFile.exists());**

}

“À ha!” Tôi nói. “Mày muốn tao gọi trình biên dịch à, hử?” “Đúng vậy, Aphonse thân mến của tao, Đó là điều tao muốn làm.”

“Vậy hãy tự chuẩn bị cho sự biên tập đi, nhóc Avery.”

public void serve(Socket theSocket) {

try {

ObjectOutputStream os =

new ObjectOutputStream(theSocket.getOutputStream());

ObjectInputStream is =

new ObjectInputStream(theSocket.getInputStream()); os.writeObject(“SMCR”);

os.flush();

CompileFileTransaction cft =

(CompileFileTransaction) is.readObject();

**String command = buildCommandLine(cft.getFilename()); CompilerResultsTransaction crt = compile(cft, command);** os.writeObject(crt);

os.flush();

theSocket.close();

} catch (Exception e) {

}

}

Chúng tôi đều dựa vào màn hình khi tôi nhấn nút kiểm tra.

Thanh màu xanh. Điệu cười. Cái gật đầu.

“Chúng ta đã làm rất tốt, Avery. Nhưng tao nghĩ chúng ta nên dọn dẹp đoạn mã này một chút.” “Tao hoàn toàn đồng ý, Alphonse, nó có hơi một chút, nên nói thế nào nhỉ, lộn xộn?”

Chúng tôi cùng nhau làm việc thông qua một mã máy chủ mới và chia nó thành một loạt phương thức nhỏ hơn kết nối với nhau thành để nói về một câu chuyện. Khi chúng tôi hoàn thành, nó trông thế này, và vẫn qua được tất cả các lần kiểm thử.

public SMCRemoteServer(int port) throws Exception { SocketService service =

new SocketService(port, new **SMCRemoteServerThread()**);

}

**private static class SMCRemoteServerThread implements SocketServer {**

**private ObjectOutputStream os;**

**private ObjectInputStream is;**

**private CompileFileTransaction cft;**

**private CompilerResultsTransaction crt;**

**public void serve(Socket theSocket) {**

**try {**

**initializeStreams(theSocket);**

**sayHello();**

**readTransaction();**

**doCompile();**

**writeResponse();**

**theSocket.close();**

**} catch (Exception e) {**

**}**

**}**

**private void readTransaction() throws IOException, ClassNotFoundException { cft = (CompileFileTransaction) is.readObject();**

**}**

**private void initializeStreams(Socket theSocket) throws IOException { os = new ObjectOutputStream(theSocket.getOutputStream());**

**is = new ObjectInputStream(theSocket.getInputStream());**

**}**

**private void writeResponse() throws IOException { os.writeObject(crt);**

**os.flush();**

**}**

**private void sayHello() throws IOException {**

**os.writeObject(“SMCR”);**

**os.flush();**

**}**

**private void doCompile() throws Exception {**

**String command = buildCommandLine(cft.getFilename()); crt = compile(cft, command);**

**}**

**}**

“Được rồi đó, trông tốt hơn nhiều rồi. Tao nghĩ đến lúc thư giãn rồi, phải không?” Tôi nói với Avery.

“Yeah, tao nghĩ chúng ta đã ở đây mấy tiếng đồng hồ rồi. Đến phòng trò chơi và chơi LandCraft thôi.”

….Vẫn còn tiếp