**Thợ lành nghề #15: Ếch là Bê (SMCRemote – phần 5)**

Hình ảnh cuối tôi thấy trước khi cánh cửa khép lại là mái tóc dài óng mượt của Jasmine, vung vẩy theo nhịp bước đầy chủ ý của nàng. Khi căn phòng tan lắng bóng dáng nàng, tôi cảm thấy lồng ngực của mình nhẹ nhõm trút đi một luồng khí nén chặt. Đôi mắt tôi mất đi tiêu điểm, và suốt nhiều phút gần như tỉnh táo, tôi ngồi thừ, nhìn về cánh cửa mờ nhoà đi trong tầm mắt.

Vẫn còn đờ người ra, tôi nhận ra mình phải trở lại làm việc – nhưng làm thế nào đây? Không biết nếu Jasmine làm, nàng sẽ viết tệp tin và dọn dẹp đoạn mã ra sao nhỉ? Tôi biết chắc là nàng sẽ nói là tôi chẳng có gì xuất sắc để cho nàng xem khi nàng trở lại: “Cao thủ mà hoá ra như vậy sao! Nãy giờ cậu chỉ ngồi thừ ra đó vọc khuấy mấy ngón tay cái phải không?”

“OK, Jasmine, OK,” Tôi nói. “Mình làm gì trước đây?”

“*Nào, cao thủ, chúng mình phải làm sao cho CompilerResultsTransaction chuyên chở nội dung của một tệp tin từ server đến client, và rồi viết tệp tin ấy xuống client.*”

“Ồ, đúng rồi”, tôi đáp. “Trình dịch sẽ tạo ra một tệp tin xuất trên server, và chúng ta phải dời nó đến client – y như thể chúng ta vừa thực hiện trong *CompileFileTransaction*.”

public class CompileFileTransaction implements Serializable {

private String filename;

private char contents[];

public CompileFileTransaction(String filename, char buffer[]) {

this.filename = filename;

this.contents = buffer;

}

public String getFilename() {

return filename;

}

public char[] getContents() {

return contents;

}

}

“Chúng ta mở và đọc tệp tin trong phương thức compileFile, rồi tạo và chuyển tên tệp tin và chuỗi ký tự vào phương thức khởi tạo của CompileFileTransaction.” Tôi tiếp tục.

public boolean compileFile() {

char buffer[] = new char[(int) itsFileLength];

try {

fileReader.read(buffer);

CompileFileTransaction cft = new CompileFileTransaction(itsFilename, buffer);

}

}

“*Rồi, cao thủ, mình vừa làm đúng y như vậy. Có điểm nào cậu không vừa lòng chăng?*”

Tôi không muốn viết đoạn mã y hệt như nhau hai lần. Jerry phản đối cực lực chuyện lặp lại mã nguồn.

“*Jerry không phải là con dao bén nhất trong tủ đâu -2-, cao thủ.”*

“Tôi không rõ chuyện đó nhưng tôi nghĩ, với quan điểm này thì anh ta đúng. Bởi thế, tôi nghĩ là tôi muốn viết một lớp dùng để gửi tệp tin qua socket.”

“*Cái gì đó tương tự như FileCarrier?*”

“Ừa, cái tên đó hay á!”

*“Rồi, cao nhân -3-, viết một cái kiểm thử cho nó đi.*”

“Cao nhân? – èm, OK. Thế này nhé?”

public class FileCarrierTest extends TestCase {

public void testAFile() throws Exception {

final String TESTFILE = “testFile.txt”;

final String TESTSTRING = “test”;

createFile(TESTFILE, TESTSTRING);

FileCarrier fc = new FileCarrier(TESTFILE);

fc.write();

assertTrue(new File(TESTFILE).exists());

String contents = readFile(TESTFILE);

assertEquals(TESTSTRING, contents);

}

}

“*Hay lắm, ông mãnh. Tiếp tục đi.*”

“Được rồi, thưa cô J. Sau đây là hai function tiện ích…”

private String readFile(final String TESTFILE) throws IOException {

BufferedReader reader = new BufferedReader(new FileReader(TESTFILE));

String line = reader.readLine();

return line;

}

private void createFile(final String TESTFILE, final String TESTSTRING) throws IOException {

PrintWriter writer = new PrintWriter(new FileWriter(TESTFILE));

writer.println(TESTSTRING);

writer.close();

}

“… và đây là phần ứng dụng rút gọn của FileCarrier sẽ làm cho phần kiểm thử biên dịch được và không chạy khi kiểm thử.”

public class FileCarrier {

public FileCarrier(String fileName) {

}

public void write() {

}

}

“Rồi, quá đã -4-, mình chỉ cần làm cho cái kiểm thử đạt bằng cách đọc tệp tin trong hàm khởi tạo và viết nó trong hàm write.”

“*Chưa đâu, ông tướng – đầu tiên là chạy cái kiểm thử để biết chắc nó hỏng cái đã.*”

“Ơ, nàng J, chưa có ứng dụng FileCarrier. Tất nhiên là nó sẽ hỏng thôi.”

“*Hả, hoả quân cậu và tôi biết như vậy. Nhưng liệu chương trình có biết thế không?*”

“Tôi thật là khoái những khi cô muốn tôi chạy kiểm thử. Được rồi, này.” [Phần kiểm thử đạt] “Hở? làm sao có thể như vậy được?”

“*Quỷ tha, tôi không rõ nữa. Làm sao kiểm thử có thể đạt trong khi chẳng có gì trong FileCarrier đến —*”

“Egad, Auriculatum! -5- mình chưa hề xoá tệp tin kiểm thử.”

“*À, ông kẹ. Thế sao cậu không chữa nó đi?*”

“OK, đây.”

public void testAFile() throws Exception {

final String TESTFILE = “testFile.txt”;

final String TESTSTRING = “test”;

createFile(TESTFILE, TESTSTRING);

FileCarrier fc = new FileCarrier(TESTFILE);

new File(TESTFILE).delete();

fc.write();

assertTrue(new File(TESTFILE).exists());

String contents = readFile(TESTFILE);

assertEquals(TESTSTRING, contents);

}

“*Ngon lành, bây giờ nó hỏng rồi! nhưng mà ông thần, cậu làm cho nó đạt được không?*

“Hiển nhiên rồi, JJ – xem đây!”

public class FileCarrier implements Serializable {

private String fileName;

private char[] contents;

public FileCarrier(String fileName) throws Exception {

File f = new File(fileName);

this.fileName = fileName;

int fileSize = (int) f.length();

contents = new char[fileSize];

FileReader reader = new FileReader(f);

reader.read(contents);

reader.close();

}

public void write() throws Exception {

FileWriter writer = new FileWriter(fileName);

writer.write(contents);

writer.close();

}

}

“Yee Hah! đại nhân, bây giờ cậu mới nên cơm nên cháo đây -6-!”

“Không có chi, cám ơn đại nương -7-, nhưng cô cũng chưa thấy gì mà. Hãy xem lối tôi tích hợp FileCarrier vào CompileFileTransaction — cái này chắc sẽ làm cô chú ý.”

public class CompileFileTransaction implements Serializable {

FileCarrier sourceFile;

public CompileFileTransaction(String filename) throws Exception {

sourceFile = new FileCarrier(filename);

}

public String getFilename() {

return sourceFile.getFileName();

}

public char[] getContents() {

return sourceFile.getContents();

}

}

“Và bây giờ tôi sẽ đổi hàm compileFile để dùng cái CompileFileTransaction mới đây!”

CompileFileTransaction cft = new CompileFileTransaction(itsFilename);

os.writeObject(cft);

os.flush();

Object response = is.readObject();

CompilerResultsTransaction crt = (CompilerResultsTransaction) response;

crt.write();

“Và bây giờ tôi chạy mấy cái kiểm thử và…. thấy chưa? Chúng đạt hết!”

“*Ô, đại cao thủ, tuyệt! Cậu đã xuất ra dăm ba tuyệt chiêu!*”

“Tính trước hết rồi mà, Dạ Hương -8-, tính hết rồi. Bây giờ hãy xem tôi đặt cái FileCarrier và trong CompilerResultsTransaction!”

public class CompilerResultsTransaction implements Serializable {

private FileCarrier resultFile;

public CompilerResultsTransaction(String filename) throws Exception {

resultFile = new FileCarrier(filename);

}

public void write() throws Exception {

resultFile.write();

}

}

“*Ôi chao!*”

“Và hãy xem cách tôi đổi cái test dùng trong transaction mới một cách nhà nghề đây!”

private void parse(Object cmd) throws Exception {

if (cmd != null) {

if (cmd instanceof CompileFileTransaction) {

CompileFileTransaction cft = (CompileFileTransaction) cmd;

filename = cft.getFilename();

content = cft.getContents();

fileLength = content.length;

fileReceived = true;

TestSMCRemoteClient.createTestFile(“resultFile.java”, “Some content.”);

CompilerResultsTransaction crt = new CompilerResultsTransaction(“resultFile.java”);

os.writeObject(crt);

os.flush();

}

}

}

“Làm sao cậu biết được hết mấy thứ này vậy, Alphonse?”

“Cô thấy đó, Jasmine, tôi biết hết mọi chuyện thuộc về đối tượng. FileCarrier là một đối tượng đó Jasmine. Cô có thấy nó có thể được sử dụng nhiều hơn chỉ một nơi không? Cô có thấy một đối tượng chỉ hàm chứa một trách nhiệm? Cô thấy không? Cô có biết Nguyên Lý Trách Nhiệm Đơn -9- không Jasmine? Có khi nào cô nghe đến nó chưa? đã nghiên cứu nó chưa? Tôi nghiên cứu nó rồi đó. Cô biết nó nói sao không, Jasmine? **Nó nói rằng một lớp chỉ nên có một và chỉ một lý do để thay đổi. Nó nói rằng mọi phương thức và thuộc tính của một lớp chỉ nên làm việc để cùng đưa đến một mục đích. Một lớp không nên cố gắng hoàn thành nhiều hơn một mục đích**…. Cô có đang lắng nghe đó không, Jasmine?”

“*Vâng, Alphonse. Tôi đang chăm chú mà.*”

“Tiếp tục lắng nghe, Jasmine. Trong đám bọn mình ai mà biết nguyên lý này gọi nó là SRP -10-. Đó là ESS ARE PEE Jasmine.”

“*ESS ARE PEE, Alphonse. Ess are pee*.”

“Còn nhớ hàm compileFile không? Còn nhớ cách nó dùng để đọc tệp tin và chuyển chuỗi chữ cái vào trong CompileFileTransaction không? Cô hỏi tôi chuyện tôi không thích cái gì trong hàm đó – tôi sẽ nói cho cô hay tôi không thích cái gì: nó vi phạm nguyên lý SRP! Nó có hai lý do để thay đổi thay vì một. Nó phụ thuộc vào cả chi tiết cách đọc tệp tin lẫn chế độ xây dựng và gởi transactions. Làm như vậy quá nhiều trách nhiệm, Jasmine hỡi – quá sức nhiều.”

“*Cậu làm tôi hoảng lên đây, Alphonse – Ôi! Alphonse!*”

Ngay lúc ấy, cùng một lúc hàng loạt sự kiện xảy ra. Tôi thấy cánh cửa đã mở ra. Tôi nhận ra chiếc ghế bên cạnh trống rỗng. Tôi nghe tiếng vọng của giọng tôi nhại Jasmine còn dội lại từ mấy bức tường. Và, hơn hết, tôi thấy Jasmine đang đứng ngay cửa ra vào.

Đôi mắt nàng lạnh như tiền.

còn tiếp…. không biết chừng.

-1- Ếch là Bê: Nguyên bản tiếng Anh tác giả chơi chữ SRP thành ESS ARE PEE. Cụm này hnd không biết phải dịch ra sao cho ổn nên dịch trại thành “ếch là bê” cho dí dỏm. Nếu có bạn nào có ý kiến nào hay, xin đóng ý.

-2- “the sharpest knife is the drawer”, một ngạn ngữ ám chỉ cho một cá nhân nào đó tài giỏi nhất trong nhóm hoặc một việc gì đó tốt nhất trong hoàn cảnh cho phép.

-3- Các từ lóng “hottie”, “hot stuff”, “over-temp”, exotherm”, “boiling point”, “tepid breath”, “latent heat”, “electron volt”, “fever man” dùng trong bài có chủ ý gia tăng cường độ. Những từ này đều dùng để đề cao một cá nhân một cách dí dỏm, thân mật và chút gì đó chế giễu. hnd tạm dịch những từ này là “cao nhân”, “ông mãnh”, “ông tướng”, “hỏa quân”, “ông kẹ”, “ông thần”, “đại nhân”, “đại cao thủ” và biết chắc là không thể tìm các từ hóm hỉnh tiếng Việt tương tự để chuyển dịch cho mỗi chữ lóng tiếng Anh này.

-4- “Jazzy-wazzy” cũng là một cụm từ lóng chỉ cho sự ngon lành, nhuần nhuyễn, vừa ý. hnd tạm dịch là “quá đã” để bình dân hoá từ lóng này.

-5- “Egad, Auriculatum”: Egad là một cách gọi cảm thán tương tự như “oh God!” và Auriculatum có nguồn gốc từ tiếng Latin, chỉ cho bộ “nghe” hoặc miêu tả hình dạng giống như chiếc lá, hoặc vành tai. Ngoài ra, “auricula” còn một số nghĩa bóng khác. Cụm “Egad, Auriculatum” này có thể dịch nôm na là “ôi trời, nghe đây” nhưng hnd để nguyên văn cho thêm phần…. bùa chú

-6- Cụm “you are cooking” là một idiom rất phổ biến, chỉ cho sự tiến triển đúng hướng và tốt đẹp.

-7- “Grandiflorum” từ grandiflora tiếng La tinh chỉ cho giống hoa hồng mọc theo dạng bụi, khóm. Từ này ám chỉ cho nữ giới. Ở trên JJ gọi Alphonse là đại nhân nên bên dưới hnd tạm dịch theo là đại nương cho khớp với tinh thần.

-8- “Night Bloomer” chỉ cho các giống hoa nở ban đêm nên tạm dịch là Dạ Hương.

-9- “Single Responsibility Principle” tạm dịch là nguyên lý trách nhiệm đơn.

-10- “SRP” hay Single Responsibility Principle, xin đọc thêm tài liệu tiếng Anh ở: <http://www.objectmentor.com/resources/articles/srp.pdf>

**Bài tiếp:** [Thợ lành nghề #16: Lịch thiệp quá mức (SMCRemote – phần 6)](https://codegym.vn/blog/2018/08/13/tho-lanh-nghe-16-lich-thiep-qua-muc-smcremote-phan-6/)

**Bài trước:** [Thợ lành nghề #14: Giao dịch (SMCRemote – phần 4)](https://codegym.vn/blog/2018/08/12/tho-lanh-nghe-14-giao-dich-smcremote-phan-4/)

**Tác giả:** Robert C. Martin

**Người dịch:** Hoàng Ngọc Diêu (conmale)