**Thợ lành nghề #4 - Bài kiểm tra tính kiên nhẫn**

*Ngày 12 tháng 7 năm 2002*

Nhật ký thân yêu,

Tối qua tôi ngồi tựa vào cửa sổ hàng giờ, nhìn các vì sao mờ dần trong bầu trời đêm. Tôi thấy việc làm của tôi và Jerry hôm qua có nhiều xung đột. Tôi học hỏi rất nhiều trong khi làm việc với Jerry với vấn đề tạo số nguyên tố, nhưng tôi không tin tôi gây ấn tượng gì với gã. Và, thật tình mà nói, tôi cũng không nể gã cho lắm. Thật ra, gã tốn khá nhiều thời gian mài dũa các đoạn mã cho dù những đoạn này làm việc ngon lành.

Hôm nay Jerry đến gặp tôi với một bài tập mới. Gã yêu cầu tôi viết một chương trình tính thừa số nguyên tố của số nguyên. Gã cho biết gã sẽ làm việc với tôi ngay từ đầu nên hai chúng tôi ngồi xuống và bắt đầu lập trình.

Tôi tin chắc tôi biết cách làm. Hôm qua chúng tôi đã viết chương trình tạo số nguyên tố. Dò tìm các thừa số nguyên tố chỉ là vấn đề đi xuyên qua danh sách các số nguyên tố và xét thử có thừa số nào từ các số nguyên đã định. Thế nên tôi vớ lấy bàn phím và bắt đầu viết mã. Khoảng nửa giờ sau khi viết và kiểm tra, tôi làm được như sau:

import java.util.Iterator;

import java.util.LinkedList;

import java.util.List;

public class PrimeFactorizer {

public static void main(String[] args) {

int[] factors = findFactors(Integer.parseInt(args[0]));

for (int i = 0; i &amp;amp;lt; factors.length; i++) {

System.out.println(factors[i]);

}

}

public static int[] findFactors(int multiple) {

List factors = new LinkedList();

int[] primes = PrimeGenerator.generatePrimes((int) Math.sqrt(multiple));

for (int i = 0; i &amp;amp;lt; primes.length; i++) {

for (; multiple % primes[i] == 0; multiple /= primes[i]) {

factors.add(new Integer(primes[i]));

}

}

return createFactorArray(factors);

}

private static int[] createFactorArray(List factors) {

int factorArray[] = new int[factors.size()];

int j = 0;

for (Iterator fi = factors.iterator(); fi.hasNext();) {

Integer factor = (Integer) fi.next();

factorArray[j++] = factor.intValue();

}

return factorArray;

}

}

Tôi kiểm tra chương trình bằng cách chạy nó với nhiều tham số khác nhau. Mọi thứ dường như ổn thoả. Chạy chương trình với giá trị tham số là 100 cho tôi kết quả 2, 2, 5 và 5. Chạy nó với 32767 cho tôi 7, 31 và 151. Chạy với 32768 cho tôi 15 số 2.

Jerry ngồi nhìn tôi. Gã chẳng nói nửa lời. Ðiều này làm tôi hơi hoảng nhưng tôi tiếp tục nắn bóp và thử nghiệm mã nguồn cho đến lúc tôi hài lòng. Sau đó, tôi bắt đầu viết phần kiểm thử đơn vị.

Jerry hỏi: “Mày làm gì vậy?”

“Chương trình chạy nên tôi đang viết các kiểm thử đơn vị.” Tôi đáp lại.

“Nếu chương trình đã chạy thì việc gì mày cần kiểm thử đơn vị?” Gã hỏi tiếp.

Tôi không nghĩ đến điểm này. Tôi chỉ biết theo thông lệ cần phải viết kiểm thử đơn vị. Tôi liều lĩnh đoán mò: “Ðể các lập trình viên khác biết được là chương trình đó chạy?”

Jerry nhìn tôi khoảng 30 giây rồi gã lắc đầu và nói: “Thời buổi này họ dạy dỗ tụi mày cái gì ở trường vậy?”

Tôi cứng lưỡi không trả lời được nhưng gã ngăn tôi lại bằng một cái nhìn.

“OK”, gã nói, “xoá hết những thứ mày đã làm đi. Tao chỉ cho mày cách tụi tao làm ở đây.”

Tôi quả không chuẩn bị cho tình thế như vậy. Gã muốn tôi xoá những gì tôi đã tạo ra trong ba mươi phút qua. Tôi chỉ ngồi yên, không tưởng tượng nổi.

Cuối cùng Jerry nói: “Xoá đi.”

Tôi trả lời: “Nhưng chương trình đó chạy mà.”

“Thì sao?” Jerry đáp lại.

Tôi bắt đầu nổi cáu. Tôi nói cứng: “Chương trình này chẳng có gì sai hết!”

“Thực vậy hở?” gã lầm bầm và vớ lấy bàn phím, rồi xoá hết mã nguồn của tôi.

Tôi điếng người. Không phải, tôi điên tiết lên. Gã mới vừa chồm qua và xoá hết đồ của tôi. Trong phút chốc ấy tôi chẳng còn thiết gì đến ưu thế được làm một tay học việc cho ông C nữa. Học việc mà phải đụng đến những kẻ tàn bạo như Jerry thì còn hay ho gì nữa? Với ý nghĩ như thế và những ý nghĩ còn kém phần tưởng thưởng khác diễn ra trong đầu khi tôi nhìn gã chằm chặp.

“À, tao thấy mày nổi đoá rồi đó.” Jerry nói một cách điềm tĩnh.

Tôi lắp bắp nhưng chẳng thốt được gì cho hay ho.

“Này.” Jerry nói, rõ ràng đang cố làm dịu tôi xuống. “Ðừng có bám rịt mã nguồn của mày quá như vậy. Chỉ có ba mươi phút làm việc mà thôi, chẳng phải là cái gì ghê gớm đâu. Mày phải chuẩn bị tinh thần vứt bỏ thêm cả đống mã nguồn nữa nếu mày muốn trở thành một thứ lập trình viên gì đó. Vứt bỏ được hàng đống mã nguồn thường là điều tốt nhất mà mày nên làm.”

Tôi buột miệng: “Nhưng làm như thế thì rất phí!”

Gã hỏi lại: “Bộ mày nghĩ giá trị của chương trình nằm trong mã nguồn sao? Không phải vậy. Giá trị của một chương trình nằm trong cái đầu của mày đó.”

Gã nhìn tôi chừng một giây rồi tiếp tục. “Có bao giờ mày lỡ tay xoá cái gì đó đang làm chưa? Cái gì đó mà mày phải mất vài ngày để làm đó?”

“Có một lần, ở trường”. Tôi nói “Cái đĩa bị hỏng và bản lưu trữ thì cách đó hai ngày rồi.”

Gã cau mày gật đầu biểu lộ sự thông cảm rồi hỏi: “Mày mất bao lâu để làm lại những cái đã bị mất?”

“Tôi nắm khá rõ những cái bị mất nên chỉ mất có nửa ngày để làm lại.”

“Ra thế mày thật sự chẳng mất khối lượng công việc của hai ngày.”

Tôi chẳng màng gì đến cái lý luận của gã. Tôi không bắt bẻ được nhưng tôi không khoái cái lý luận đó. Chỉ đơn giản là tôi cảm thấy bị mất một khối lượng hai ngày làm việc!

Gã hỏi tiếp: “Mày có nhận thấy phần mã làm lại tốt hơn hay tệ hơn phần mã mày bị mất không?”

“Ồ, tốt hơn nhiều.” Tôi nói, ngay lập tức hối tiếc vì đã phát biểu như thế. “Lần thứ hai tôi có thể dùng một cấu trúc tốt hơn nhiều.”

Gã cười. “Thế thì cố thêm 25%, mày đưa ra được một giải pháp tốt hơn.”

Lý luận của gã làm tôi bực mình. Tôi lắc đầu và gần như thét lên: “Có phải ông giả định là chúng ta luôn luôn vứt bỏ mã nguồn sau khi làm xong?”

Trả lời cho sự ngạc nhiên của tôi, gã gật đầu và nói: “Gần như là như vậy. Tao giả định chuyện vứt bỏ mã nguồn là một việc đáng giá và hữu dụng. Tao giả định mày không nên xem đó là chuyện hoang phí. Tao giả định mày không nên ôm khư khư cái mã nguồn của mày.”

**Bài tiếp:** [Thợ lành nghề #5: Bước nhỏ](https://codegym.vn/blog/2018/08/03/tho-lanh-nghe-5-buoc-nho/)

**Bài trước:** [Thợ lành nghề #3 – Tính rõ ràng và sự cộng tác](https://codegym.vn/blog/2018/08/01/tho-lanh-nghe-3-tinh-ro-rang-va-su-cong-tac/)

**Tác giả:** Robert C. Martin

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