

THUYẾT MINH ĐỀ TÀI
NGHIÊN CỨU KHOA HỌC CẤP ĐẠI HỌC HUẾ

1. TÊN ĐỀ TÀI PHÁT HIỆN VÀ CẢI THIẾN MÃ NGUỒN XỬ LÝ NGOẠI LỆ XẤU TRONG JAVASCRIPT			2. MÃ SỐ
3. LĨNH VỰC NGHIÊN CỨU Tự nhiên: <input type="checkbox"/> Kinh tế, XHNV: <input type="checkbox"/> Giáo dục: <input type="checkbox"/> Kỹ thuật: <input checked="" type="checkbox"/> Nông lâm: <input type="checkbox"/> Y Dược: <input type="checkbox"/> Môi trường: <input type="checkbox"/> ATLĐ: <input type="checkbox"/> Sở hữu trí tuệ: <input type="checkbox"/>		4. LOẠI HÌNH NGHIÊN CỨU Cơ bản <input checked="" type="checkbox"/> Ứng dụng <input type="checkbox"/> Triển khai <input type="checkbox"/>	
5. THỜI GIAN THỰC HIỆN: 24 tháng Từ tháng 01 năm 2017 đến tháng 12 năm 2018			
6. CƠ QUAN CHỦ TRÌ Tên cơ quan: TRƯỜNG ĐẠI HỌC KHOA HỌC Địa chỉ: 77 Nguyễn Huệ, Thành phố Huế, Tỉnh Thừa Thiên Huế. Điện thoại: 054 3823290 Fax: 054 3824901 Email: doingoaidhkh@gmail.com			
7. CHỦ NHIỆM ĐỀ TÀI Họ và tên: TRẦN THANH LƯƠNG Học vị, chức danh KH: TS - GV Chức vụ: P. Trưởng phòng Địa chỉ CQ: 77 Nguyễn Huệ, Tp. Huế Địa chỉ NR: M15, Đông Nam Thủy An, Hương Thủy Điện thoại CQ: 054 3846490 Điện thoại NR: Điện thoại di động: 091 4145414 Fax: 054 3824901 E-mail: tluong@hueuni.edu.vn; tluong@gmail.com			
8. NHỮNG NGƯỜI THAM GIA THỰC HIỆN ĐỀ TÀI			
Họ và tên	Đơn vị công tác và lĩnh vực chuyên môn	Nội dung cụ thể được giao	Chữ ký
TS. Trần Thanh Lương	Phòng Đào tạo Đại học, Trường Đại học Khoa học	Chủ nhiệm đề tài, Nghiên cứu các cơ chế ngoại lệ	
ThS. Lê Mỹ Cảnh	Khoa Công nghệ Thông tin, Trường Đại học Khoa học	Thư ký đề tài, Đánh giá các phương pháp cải thiện mã nguồn.	
TS. Hoàng Thị Lan Giao	Khoa Công nghệ Thông tin, Trường Đại học Khoa học	Thành viên, Xác định các ngoại lệ.	
ThS. Hồ Thị Kim Thoa	Khoa Tin học, Trường Đại học Sư phạm	Thành viên, Nghiên cứu tìm các mẫu mã nguồn xấu trong JavaScript.	

9. ĐƠN VỊ PHỐI HỢP CHÍNH		
Tên đơn vị trong và ngoài nước	Nội dung phối hợp nghiên cứu	Họ tên người đại diện
Khoa Khoa học máy tính và Kỹ nghệ thông tin, Đại học Công nghệ Quốc gia Đài Loan	Kiểm chứng và đánh giá các phương án cải tiến mã nguồn	GS. Chin-Yun Hsieh GS. Yu Chin Cheng
<p>10. TÌNH HÌNH NGHIÊN CỨU THUỘC LĨNH VỰC CỦA ĐỀ TÀI TRONG VÀ NGOÀI NƯỚC</p> <p>10.1. Ngoài nước (<i>phân tích, đánh giá tình hình nghiên cứu thuộc lĩnh vực của đề tài trên thế giới, liệt kê danh mục các công trình nghiên cứu, tài liệu có liên quan đến đề tài được đánh trích dẫn khi đánh giá tổng quan</i>).</p> <p>Để nâng cao chất lượng mã nguồn, một số nhà khoa học đã nghiên cứu cách thức phát hiện các mẫu mã xấu và đề xuất các phương pháp cải tiến để loại bỏ các đoạn mã xấu. Tuy nhiên, hầu hết các nghiên cứu đều tập trung vào các mẫu mã xấu trong việc xử lý logic chính của chương trình (M. Fowler, K. Beck, J. Brant, W. Opdyke và D. Roberts, 1999; R. C. Martin, 2008). Những công trình nghiên cứu về mã xấu trong xử lý ngoại lệ còn rất ít. Trong số ít các nghiên cứu này, R. C. Martin đã chỉ ra một số kỹ thuật để viết mã tốt cho việc xử lý ngoại lệ như sử dụng ngoại lệ (exception) thay vì trả về mã lỗi (return code), dùng ngoại lệ không kiểm tra (unchecked exception), không trả về null. Những nghiên cứu hiện tại này đều tập trung vào xử lý ngoại lệ trong các ngôn ngữ hướng đối tượng tĩnh (static object oriented languages) điển hình như Java hoặc C#. Việc phát hiện mã xử lý ngoại lệ xấu và cải tiến chúng trong các ngôn ngữ động (dynamic language) như JavaScript là một lĩnh vực mới ở thời điểm hiện tại.</p> <p>Danh mục các công trình liên quan:</p> <p>[1] M. Fowler, K. Beck, J. Brant, W. Opdyke and D. Roberts, <i>Refactoring: Improving the Design of Existing Code</i>, Addison-Wesley Longman, 1999.</p> <p>[2] R. C. Martin, <i>Clean Code: A Handbook of Agile Software Craftsmanship</i>, Prentice Hall, 2008.</p> <p>[3] C.-T. Chen, Y. C. Cheng, C.-Y. Hsieh and I.-L. Wu, <i>Exception handling refactorings: Directed by goals and driven by bug fixing</i>, The Journal of Systems and Software, vol. 82, no. 2, p. 333–345, 2009.</p> <p>[4] A. F. Garcia, C. M. Rubira, A. Romanovsky and J. Xu, <i>A comparative study of exception handling mechanisms for building dependable object-oriented software</i>, Journal of Systems and Software, vol. 59, no. 2, p. 197–222, 2001.</p> <p>[5] M. Robert and T. Anand, <i>Issues with Exception Handling in Object-Oriented</i>, in ECOOP '97 - Object-Oriented Programming - 11th European Conference, Springer, p. 85-103, 1997.</p> <p>[6] P. A. Buhr and W. Y. Russell Mok, <i>Advanced Exception Handling Mechanisms</i>, IEEE Transactions on Software Engineering, vol. 26, no. 9, pp. 820-836, September 2000.</p> <p>[7] M. E. Daggett, <i>Expert JavaScript</i>, Apress, 2013, pp. 81-86.</p> <p>[8] J. Shore, <i>Fail Fast</i>, IEEE Software, vol. 21, no. 5, pp. 21-25, 2004.</p> <p>[9] C.-Y. Hsieh, L. M. Canh, H. Kim Thoa and Y. C. Cheng, <i>Identification and Refactoring of Exception Handling Code Smells in JavaScript</i>, Taiwan Academic Network Conference, 2015.</p> <p>10.2. Trong nước (<i>phân tích, đánh giá tình hình nghiên cứu thuộc lĩnh vực của đề tài ở Việt Nam, liệt kê danh mục các công trình nghiên cứu, tài liệu có liên quan đến đề tài được đánh trích dẫn khi đánh giá tổng quan</i>).</p> <p>Tại Việt Nam, việc nghiên cứu các mã nguồn xấu nói chung và mã nguồn xử lý ngoại lệ xấu nói riêng chưa được đề cập đến một cách chính thức. Hiện nay, một số tài liệu đề cập đến mẫu thiết kế (design</p>		

patterns), một hướng nghiên cứu về mã nguồn lập trình, để cải thiện chất lượng mã nguồn phục vụ cho các lập trình viên trong việc thiết kế và xây dựng chương trình. Các tài liệu này chủ yếu là các tài liệu dịch từ các sách, tạp chí tiếng nước ngoài chưa phải là những công trình nghiên cứu của các tác giả trong nước.

10.3. Danh mục các công trình đã công bố thuộc lĩnh vực của đề tài của chủ nhiệm và những thành viên tham gia nghiên cứu (*Họ và tên tác giả, bài báo, ấn phẩm, các yếu tố về xuất bản*).

[1] Thanh-Luong Tran, Linh Anh Nguyen and Thi-Lan-Giao Hoang, *A Domain Partitioning Method for Bisimulation-Based Concept Learning in Description Logics*, ICCSAMA'14, Advances in Intelligent Systems and Computing 282, p. 297-312, Springer, 2014.

[2] Thanh-Luong Tran, Linh Anh Nguyen and Thi-Lan-Giao Hoang, *Bisimulation-based concept learning for information systems in description logics*, Vietnam Journal of Computer Science, vol. 2, no. 3, p. 149-167, 2015.

[3] C.-Y. Hsieh, L. M. Canh, H. Kim Thoa and Y. C. Cheng, *Identification and Refactoring of Exception Handling Code Smells in JavaScript*, Taiwan Academic Network Conference, Taiwan, 2015.

11. TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Hiện nay, các sản phẩm của công nghệ phần mềm đã và đang được ứng dụng vào việc quản lý, điều hành và tác nghiệp trong hầu hết các lĩnh vực của đời sống và đem lại hiệu quả kinh tế cao. Do sự tiện lợi của Internet trong xu thế toàn cầu, tốc độ đường truyền đã được cải thiện cũng như sự phát triển của công nghệ nên những sản phẩm ứng dụng được xây dựng trên nền tảng World Wide Web ngày càng phổ biến. Cùng với sự phát triển của ngôn ngữ đánh dấu siêu văn bản thế hệ mới - HTML5, JavaScript đã tạo ra sức mạnh cho các ứng dụng chạy trên nền Web. JavaScript là một ngôn ngữ lập trình kịch bản với đặc tính quan trọng là khả năng tạo và sử dụng các đối tượng cho phép người sử dụng tạo ra các trang Web có tính tương tác cao. Thêm vào đó, hiện tại JavaScript cũng còn được sử dụng để lập trình phía server (Node.js). Ngoài những ưu điểm nổi bật như tính linh hoạt trong lập trình, khả năng chạy và tương thích trên nhiều trình duyệt, khả năng tạo ra các giao diện người dùng tinh tế và phức tạp, cho phép kiểm tra tính hợp lệ của dữ liệu trên máy khách, kích thước các script nhỏ, tốc độ tải xuống nhanh, được hỗ trợ bởi số lượng lớn cộng đồng các nhà phát triển, ... JavaScript cũng có những nhược điểm như khả năng bảo mật kém và hiệu ứng kịch bản phụ thuộc vào trình duyệt và khả năng xảy ra lỗi không mong muốn khi thực thi các script rất cao. Chính tính linh hoạt của JavaScript là một trong những nguyên nhân làm cho lập trình viên dễ tạo ra những đoạn mã tiềm ẩn nhiều nguy cơ phát sinh lỗi, các đoạn mã này được gọi là các mẫu mã xấu (code smells).

Bất cứ một lập trình viên nào cũng muốn ứng dụng của mình thực thi một cách tốt đẹp theo các yêu cầu đã được đặt ra, hạn chế tối đa những lỗi xảy ra trong lúc thực thi chương trình. Nghĩa là phải xây dựng được một hệ thống mã nguồn tốt, ổn định, tránh những mẫu mã xấu trong lúc viết chương trình, đặc biệt là các mẫu mã xấu trong xử lý ngoại lệ. Đây là một vấn đề rất khó thực hiện ngay cả với những lập trình viên có nhiều kinh nghiệm khi xây dựng ứng dụng bằng ngôn ngữ lập trình hướng đối tượng tĩnh. Điều này đặc biệt đúng trong việc xây dựng các ứng dụng với ngôn ngữ lập trình JavaScript - một ngôn ngữ lập trình động, thông dịch, hỗ trợ hướng đối tượng.

Để giúp cho lập trình viên tránh hoặc phát hiện ra các mẫu mã xấu trong xử lý ngoại lệ nhằm giảm thiểu tối đa những lỗi xảy ra trong lúc thực thi chương trình - nâng cao chất lượng mã nguồn, phương pháp phát hiện các mẫu mã xấu trong xử lý ngoại lệ và cách thức cải thiện mã nguồn nhằm loại bỏ chúng là một vấn đề hết sức cần thiết.

12. MỤC TIÊU ĐỀ TÀI

12.1. Mục tiêu chung

Xác định các mẫu mã xấu trong xử lý ngoại lệ đối với ngôn ngữ JavaScript và những ảnh hưởng của chúng tới chất lượng mã nguồn, chất lượng sản phẩm. Từ đó đề xuất các phương thức cải tiến cho các mẫu nhằm loại bỏ chúng.

12.2. Mục tiêu cụ thể

- Xác định được cơ chế xử lý ngoại lệ trong ngôn ngữ JavaScript
- Nghiên cứu các phương pháp xử lý ngoại lệ thông thường trong các dự án sử dụng JavaScript
- Tìm được các mẫu mã xử lý ngoại lệ xấu trong JavaScript
- Xây dựng các cách thức cải tiến mẫu mã xấu trong JavaScript

13. ĐỐI TƯỢNG, PHẠM VI NGHIÊN CỨU

13.1. Đối tượng nghiên cứu

- Các thuộc tính của ngôn ngữ lập trình JavaScript
- Các dự án JavaScript mã nguồn mở
- Các mẫu mã xấu trong JavaScript
- Các cách thức cải tiến mã xấu trong JavaScript

13.2. Phạm vi nghiên cứu

- Nghiên cứu các thuộc tính của ngôn ngữ lập trình JavaScript trong phạm vi liên quan đến xử lý ngoại lệ.
- Nghiên cứu các cơ chế, cách thức xử lý ngoại lệ trong các ngôn ngữ thuần hướng đối tượng.
- Nghiên cứu cụ thể cách thức xử lý ngoại lệ trong JavaScript ở phạm vi lập trình cả đồng bộ (synchronous) và bất đồng bộ (asynchronous), ở cả phía client và phía server.

14. CÁCH TIẾP CẬN, PHƯƠNG PHÁP NGHIÊN CỨU

14.1. Cách tiếp cận

- Bắt đầu với các cơ chế xử lý ngoại lệ trong các ngôn ngữ hướng đối tượng, tìm hiểu các khía cạnh, đặc tính của mỗi cơ chế xử lý ngoại lệ trong các ngôn ngữ này.
- Tìm hiểu, nghiên cứu cơ chế xử lý ngoại lệ trong JavaScript.
- Tìm hiểu các đặc tính chuyên biệt của ngôn ngữ JavaScript có liên quan đến việc xử lý ngoại lệ.
- Thu thập, phát hiện các mẫu mã xử lý ngoại lệ xấu có thể xảy ra trong ngôn ngữ JavaScript. Tìm hiểu các ảnh hưởng của chúng tới chất lượng sản phẩm phần mềm. Từ đó tìm cách loại bỏ các mẫu mã xấu này, đề xuất thành các phương thức cải tiến cụ thể, phân tích các kết quả tích cực có thể thu được khi áp dụng các phương thức cải tiến đề xuất.

14.2. Phương pháp nghiên cứu

- Nghiên cứu thiết kế của một cơ chế xử lý ngoại lệ, các mẫu mã xấu đã được liệt kê và phương thức cải tiến tương ứng.
- Tìm hiểu các dự án JavaScript mã nguồn mở để hiểu cách thức xử lý ngoại lệ hiện tại mà các lập trình viên thường hay sử dụng.
- Đánh giá mức độ rủi ro và những ảnh hưởng của các mẫu mã xấu đến chất lượng phần mềm.

- Cải tiến các mẫu mã xử lý ngoại lệ xấu để xử lý hiệu quả hơn và đề xuất thành phương pháp cụ thể.

15. NỘI DUNG NGHIÊN CỨU VÀ TIẾN ĐỘ THỰC HIỆN

15.1. Nội dung nghiên cứu (trình bày dưới dạng đề cương nghiên cứu chi tiết)

Nội dung 1: Kiến thức cơ sở

- 1.1. Các cơ chế xử lý ngoại lệ trong ngôn ngữ lập trình
- 1.2. Các đặc tính ngôn ngữ chuyên biệt của JavaScript
- 1.3. Cơ chế xử lý ngoại lệ trong ngôn ngữ JavaScript

Nội dung 2: Các mẫu mã xử lý ngoại lệ xấu trong JavaScript

- 2.1. Xác định các mẫu mã xấu trong xử lý ngoại lệ
- 2.2. Ảnh hưởng của các mẫu mã xấu đến chất lượng phần mềm và mã nguồn

Nội dung 3: Các thức cải tiến mã nguồn

- 3.1. Mô hình hóa cách thức cải tiến mã nguồn cho từng mẫu mã xấu
- 3.2. Lợi ích mang lại khi áp dụng các cách thức cải tiến
- 3.3. Ví dụ minh chứng cho mỗi mẫu mã xấu

Nội dung 4: Phân tích, kết luận và mở rộng

- 4.1. Phân tích đánh giá mức độ tương đồng của các mẫu mã xử lý ngoại lệ xấu
- 4.2. Kết luận và các hướng nghiên cứu mở rộng

15.2. Tiến độ thực hiện

STT	Các nội dung và công việc thực hiện chủ yếu	Sản phẩm	Thời gian (bắt đầu-kết thúc)	Người thực hiện
1	Tìm hiểu các cơ chế xử lý ngoại lệ trong các ngôn ngữ lập trình hướng đối tượng.	Báo cáo về các cơ chế xử lý ngoại lệ	Từ 01/2017 đến 03/2017	Trần Thanh Lương, Lê Mỹ Cảnh, Hồ Thị Kim Thoa
2	Tìm hiểu các đặc tính chuyên biệt của JavaScript trong xử lý ngoại lệ	Báo cáo chi tiết về các đặc tính chuyên biệt	Từ 04/2017 đến 09/2017	Trần Thanh Lương, Lê Mỹ Cảnh, Hoàng Thị Lan Giao
3	Đọc các dự án mã nguồn mở để tìm hiểu cách thức xử lý ngoại lệ hiện tại trong JavaScript ở phía client cũng như server (trong Node.js)	01 học viên bảo vệ thành công luận văn thạc sĩ	Từ 10/2017 đến 12/2017	Trần Thanh Lương, Lê Mỹ Cảnh, Hoàng Thị Lan Giao
4	Phân tích, đưa ra các mẫu mã xử lý ngoại lệ xấu.	Bài báo 1	Từ 01/2018 đến 05/2018	Lê Mỹ Cảnh, Hồ Thị Kim Thoa
5	Đánh giá các khả năng cải tiến, đưa ra phương pháp cải tiến cho từng mẫu mã xấu.	Bài báo 2	Từ 06/2018 đến 10/2018	Trần Thanh Lương, Lê Mỹ Cảnh, Hoàng Thị Lan Giao
6	Tập hợp kết quả, viết báo cáo tổng kết đề tài.	Báo cáo tổng kết	Từ 11/2018 đến 12/2018	Trần Thanh Lương, Lê Mỹ Cảnh

16. SẢN PHẨM

16.1. Sản phẩm khoa học

Sách chuyên khảo: ☐

Bài báo đăng Tạp chí nước ngoài: ☐

Sách tham khảo: ☐

Bài báo đăng Tạp chí trong nước: ☒

Giáo trình: ☐

Bài báo đăng HN/HT quốc tế: ☒

16.2. Sản phẩm đào tạo

Nghiên cứu sinh: ☐

Thạc sĩ: ☒

Cử nhân: ☐

16.3. Sản phẩm ứng dụng

Mẫu ☐ Tài liệu dự báo ☐ Quy phạm ☐ Luận chứng kinh tế ☐

Vật liệu ☐ Bản kiến nghị ☐ Giống cây trồng ☐ Quy trình công nghệ ☐

Thiết bị máy móc ☐ Đề án ☐ Báo cáo phân tích ... ☐ Dây chuyền công nghệ .. ☐

Tiêu chuẩn ☐ Phương pháp ☐ Bản quy hoạch ☐ Chương trình máy tính .. ☐

Sơ đồ, bản thiết kế .. ☐ Giống vật nuôi ☐

16.4. Các sản phẩm khác (ghi rõ sản phẩm gì): Không

16.5. Tên sản phẩm, số lượng và yêu cầu khoa học đối với sản phẩm

STT	Tên sản phẩm	Số lượng	Yêu cầu khoa học
1	Bài báo khoa học	02	Đăng ở tạp chí khoa học Đại học Huế hoặc tạp chí của Hội đồng chức danh Giáo sư nhà nước hoặc hội thảo quốc tế.
2	Đào tạo thạc sĩ	01	

17. HIỆU QUẢ (giáo dục và đào tạo, kinh tế-xã hội)

Làm tài liệu tham khảo cho học viên cao học, sinh viên đại học, những người nghiên cứu chuyên ngành công nghệ phần mềm nói chung và cho các lập trình viên JavaScript nói riêng.

18. PHƯƠNG THỨC CHUYỂN GIAO KẾT QUẢ NGHIÊN CỨU VÀ ĐỊA CHỈ ỨNG DỤNG

- Báo cáo tổng kết đề tài làm tài liệu tham khảo cho học viên cao học, sinh viên đại học chuyên ngành công nghệ phần mềm.
- Công bố các kết quả trên một số tạp chí chuyên ngành, báo cáo tại các hội nghị, hội thảo khoa học chuyên ngành trong nước hoặc quốc tế.
- Địa chỉ ứng dụng: Khoa Công nghệ Thông tin của các trường đại học trong cả nước.

19. KINH PHÍ THỰC HIỆN ĐỀ TÀI VÀ NGUỒN KINH PHÍ

Tổng kinh phí (triệu đồng): **64.000.000 đồng** (Sáu mươi tư triệu đồng chẵn)

Trong đó:

Nghân sách Nhà nước: **64.000.000 đồng** (Sáu mươi tư triệu đồng chẵn)

Các nguồn kinh phí khác: **0 đồng** (Không đồng)

Nhu cầu kinh phí từng năm:

- Năm 2017: **32.000.000 đồng** (Ba mươi hai triệu đồng chẵn)

- Năm 2018: **32.000.000 đồng** (Ba mươi hai triệu đồng chẵn)

Dự trù kinh phí theo các mục chi phù hợp với nội dung nghiên cứu:						
STT	Khoản chi, nội dung chi	Thời gian thực hiện	Tổng kinh phí	Nguồn kinh phí		Ghi chú
				Kinh phí từ NSNN	Nguồn khác	
I	Chi công lao động tham gia trực tiếp thực hiện đề tài		44.407.000	44.407.000	0	
	<i>Chủ nhiệm đề tài</i>					
1	Trần Thanh Lương	40 công	20.328.000	20.328.000	0	
	<i>Thư ký, thành viên chính</i>					
2	Lê Mỹ Cảnh	35 công	11.011.000	11.011.000	0	
3	Hoàng Thị Lan Giao	35 công	11.011.000	11.011.000	0	
	<i>Thành viên</i>					
4	Hồ Thị Kim Thoa	20 công	2.057.000	2.057.000	0	
II	Chi khác		19.593.000	19.593.000	0	
1	Tham dự hội nghị, hội thảo		6.000.000	6.000.000	0	
2	Seminar chuyên môn		5.000.000	5.000.000	0	
3	Văn phòng phẩm		1.500.000	1.500.000	0	
4	Photo tài liệu		1.393.000	1.393.000	0	
5	Quản lý của cơ quan chủ trì		3.200.000	3.200.000	0	
6	Nghiệm thu cấp cơ sở		2.500.000	2.500.000	0	
Tổng cộng			64.000.000	64.000.000	0	
(Sáu mươi tư triệu đồng chẵn)						

Ngày tháng năm 2016

Cơ quan chủ trì
(Ký tên, đóng dấu)

Ngày 07 tháng 9 năm 2016

Chủ nhiệm đề tài
(Họ và tên, ký)

TRẦN THANH LƯƠNG

Cơ quan chủ quản duyệt
GIÁM ĐỐC ĐẠI HỌC HUẾ