BỘ GIÁO DỤC VÀ ĐÀO TAO

**TRƯỜNG ĐẠI HỌC QUY NHƠN**

**-----🙠**🕮**🙢-----**

**ĐỖ ĐĂNG KHOA**

**ĐỀ CƯƠNG LUẬN VĂN CAO HỌC**

**ĐỘ TƯƠNG TỰ VỀ HÀNH VI CỦA CHƯƠNG TRÌNH   
VÀ THỰC NGHIỆM.**

Chuyên ngành: **Khoa Học Máy Tính**

Mã số: **60 48 01 01**

Khóa: **K19**

Giảng viên hướng dẫn: **Ts. Phạm Văn Việt**

Thời gian thực hiện: **Từ 12/2017 đến 6/2018**

**Bình Định – 2017**

**MỤC LỤC**

[**MỞ ĐẦU** 3](#_Toc501055816)

[1. Lý do chọn đề tài 3](#_Toc501055817)

[2. Mục tiêu nghiên cứu 4](#_Toc501055818)

[2.1. Mục tiêu nghiên cứu chính 4](#_Toc501055819)

[2.2. Mục tiêu nghiên cứu cụ thể 4](#_Toc501055820)

[3. Đối tượng, phạm vi nghiên cứu 4](#_Toc501055821)

[3.1. Đối tượng nghiên cứu 4](#_Toc501055822)

[3.2. Phạm vi nghiên cứu 4](#_Toc501055823)

[4. Phương pháp nghiên cứu, thực nghiệm 5](#_Toc501055824)

[4.1. Nghiên cứu lý thuyết 5](#_Toc501055825)

[4.2. Thực nghiệm 5](#_Toc501055826)

[5. Nội dung dự kiến 5](#_Toc501055827)

[6. Dự kiến kế hoạch thực hiện 6](#_Toc501055828)

# **MỞ ĐẦU**

## **1. Lý do chọn đề tài**

Trong những năm gần đây, xu hướng đào tạo lập trình viên nói riêng và công nghệ phần mềm nói chung đang ngày càng trở nên phổ biến. Các trường đại học và một số trung tâm đào tạo đã cho ra đời nhiều chương trình đào tạo phong phú về nội dung, đa dạng về hình thức và thu hút được nhiều sự quan tâm.

Trong mỗi khóa học như vậy thường có hằng trăm, hằng ngàn người tham gia nhưng số giảng viên giảng dạy chỉ có một vài người, việc đảm bảo chất lượng của các khóa đào tạo trở thành một thách thức. Trong quá trình dạy học, yêu cầu người dạy phải đọc và hiểu tất cả mã lập trình của sinh viên, nhưng công việc này lại tốn quá nhiều thời gian. Nếu bỏ qua hoặc trì hoãn các công việc như vậy thì người dạy không thể theo dõi được quá trình học tập của người học. Về phía người học, yêu cầu đặt ra là phải tiến bộ theo thời gian, nắm vững lý thuyết và thành thạo kỹ năng lập trình. Những không phải lúc nào gặp khó khăn người học đều có sự hỗ trợ kịp thời từ giảng viên. Họ có thể nhờ sự giúp đỡ từ đồng nghiệp, bạn bè, nhưng tất cả mọi người chưa chắc đã đủ trình độ, kinh nghiệm hoặc thời gian để ngồi bên cạnh giúp đỡ người học khi cần.

Để giảm bớt những khó khăn nêu trên, một giải pháp đễ hỗ trợ quá trình đào tạo cho sinh viên được hiệu quả và tiết kiệm thời gian hơn đó là một công cụ giúp tự động hóa (có thể một phần) việc đánh giá kết quả lập trình của sinh viên, cũng như hỗ trợ theo giõi sự tiến bộ của người học. Công cụ tự động hóa này sẽ tính toán, định lượng tỷ lệ chính xác sự tương tự về hành vi giữa chương trình của người học và chương trình của người dạy đưa ra trước đó. Dựa trên kết quả, giảng viên sẽ đánh giá được kỹ năng lập trình của sinh viên, sự giống nhau giữa hai chương trình càng cao thì tỷ lệ độ tương tự càng cao, điểm số càng cao. Nếu điểm số thấp, sinh viên có thể quay lại kiểm tra để viết mã chương trình đúng hơn, hạn chế được nguy cơ tìm ẩn trong cách viết chương trình của sinh viên.

Những đánh giá này có thể thực hiện được nếu ta đo được độ tương tự giữa các chương trình có độ chính xác cao. Đề tài ***“Độ tương tự về hành vi của các chương trình và làm thực nghiệm”*** với mục đích sẽ giải quyết các vấn đề nếu trên.

## **2. Mục tiêu nghiên cứu**

### **2.1. Mục tiêu nghiên cứu chính**

- Đánh giá độ tương tự về hành vi của các chương trình.

### **2.2. Mục tiêu nghiên cứu cụ thể**

- Tìm hiểu sự tương tự ngữ nghĩa của chương trình.

- Tìm hiểu kỹ thuật, công cụ sinh testcase tự động.

- Phân tích các độ đo và áp dụng kỹ thuật sinh testcase tự động trên các độ đo.

- Tìm cách kết hợp các độ đo với nhau.

- Tìm một số ứng dụng của độ đo, chọn một ứng dụng để làm thực nghiệm, đánh giá kết quả thực nghiệm.

## **3. Đối tượng, phạm vi nghiên cứu**

### **3.1. Đối tượng nghiên cứu**

- Kỹ thuật sinh testcase

- Độ đo tương tự hành vi

- Một số ứng dụng của độ đo

### **3.2. Phạm vi nghiên cứu**

- Đo độ tương tự hành vi dựa vào test case

- Thực nghiệm, đánh giá trên các chương trình C#

**4. Phương pháp nghiên cứu, thực nghiệm**

### **4.1. Nghiên cứu lý thuyết**

- Độ tương tự hành vi

- Một số kỹ thuật sinh test cases tự động

- Độ đo tương tự hành vi dựa trên test cases

- So sánh, kết hợp các độ đo

### **4.2. Thực nghiệm**

- Tiến hành cài kỹ thuật đo độ tương tự hành vi

- Thực nghiệm trên dữ liệu thực của *Pex4fun*

- Phân tích, đánh giá dựa trên kết quả thực nghiệm

## **5. Nội dung dự kiến**

Đề tài dự kiến được tổ chức thành 4 chương:

***Chương 1: Giới thiệu***

- Lý do chọn đề tài, ngữ cảnh bài toán

- Những nghiên cứu có liên quan

***Chương 2: Kiến thức cơ sở***

- Một số khái niệm, định nghĩa

- Kỹ thuật lập trình trên C#

- Một số kỹ thuật sinh testcase

***Chương 3: Độ tương tự hành vi của chương trình***

- Một số khái niệm, định nghĩa

- Một số phép đo độ tương tự hành vi dựa trên testcase

- So sánh các phép đo và phương pháp kết hợp

***Chương 4: Cài đặt các phép đo và kết quả thực nghiệm***

- Cài đặt các phép đo độ tương tự hành vi

- Thực nghiệm và đánh giá kết quả các phép đo

**Phần kết luận**

Trình bày những kết quả đạt được của đề tài, những hạn chế và hướng phát triển.

**Tài liệu tham khảo**

## **6. Dự kiến kế hoạch thực hiện**

|  |  |  |
| --- | --- | --- |
| **STT** | **Thời gian** | **Công việc** |
| 1 | Từ 01/12/2017 đến 31/01/2018  (02 tháng) | - Kiến thức cơ sở  - Một số nghiên cứu liên quan |
| 2 | Từ 01/02/2018 đến 30/4/2018  (03 tháng) | - Kỹ thuật sinh test case tự động  - Lý thuyết về một số độ đo tương tự  - So sánh các độ đo, kết hợp |
| 3 | Từ 01/5/2017 đến 15/6/2018  (45 ngày) | - Một số ứng dụng thực tế  - Thực nghiệm, đánh giá |
| 4 | Từ 16/6/2017 đến 30/6/2018  (15 ngày) | - Hoàn thiện |

**Tài liệu tham khảo**

1. [Kaur, 2015] – A novel approach for syntactic similarity between two short text
2. [Gamallo, 2001] – Syntactic-based methods for measuring word similarity
3. [Baster, 1998] – Clone detection using abstract syntax trees
4. [Sihan, 2016] - Measuring code behavioral similarity for programming and software engineering education
5. [Jiang, 2009] - Automatic mining of functionally equivalent code fragments via random testing
6. [Bollegala, 2007] – Measuring semantic similarity between words using web search engines
7. [Roover, 2007] - Behavioral similarity matching using concrete source code templates in logic queries
8. Pex4Fun. http://www.pex4fun.com/.

|  |  |
| --- | --- |
| **CHỮ KÝ CỦA GVHD**  **Ts. Phạm Văn Việt** | **CHỮ KÝ CỦA HỌC VIÊN**  **Đỗ Đăng Khoa** |
| **DUYỆT CỦA TRƯỞNG KHOA** | |