TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

****

**BÁO CÁO ĐỒ ÁN**

**MÔN CÔNG NGHÊ TRI THỨC & MÁY HỌC**

*“XÂY DỰNG BOT KHẢO SÁT ĐỒ THỊ HÀM SỐ”*

Lớp: **I don’t fucking remember omg :D**

GVLT: **abc xyz**

NHÓM THỰC HIỆN:

1. **NGUYỄN LÊ QUỲNH ANH 15520022**
2. **LÊ NGUYỄN NGỌC THẢO 15520818**
3. **NGUYỄN THU HẰNG 155xxx**
4. **HUỲNH VĨ HÀ 155xxx**

Mục lục

[PHÂN CÔNG CÔNG VIỆC 4](#_Toc501752346)

[CHƯƠNG I: GIỚI THIỆU 5](#_Toc501752347)

[1.1. Mục đích đề tài và mô tả 5](#_Toc501752348)

[1.2. Ứng dụng thực tiễn 5](#_Toc501752349)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 6](#_Toc501752350)

[2.1. Tổng quan 6](#_Toc501752351)

[2.2. Về MapleSoft và OpenMaple API 6](#_Toc501752352)

[2.3. Về các thư viện thứ ba 6](#_Toc501752353)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ 7](#_Toc501752354)

[3.1. Phân tích bài toán 7](#_Toc501752355)

[3.1.1. Xây dựng bot đơn giản 7](#_Toc501752356)

[3.1.2. Xây dựng hệ thống khảo sát hàm số 7](#_Toc501752357)

[3.1.3. Các yêu cầu khác 7](#_Toc501752358)

[3.2. Cấu trúc dữ liệu và cách biểu diễn các trạng thái khác của bài toán 7](#_Toc501752359)

[3.2.1. Thực hiện trên Microsoft .NET 7](#_Toc501752360)

[3.2.2. Thực hiện trên package thuộc MapleSoft 7](#_Toc501752361)

[3.2.3. Các yêu cầu thêm khác 7](#_Toc501752362)

[3.3. Các vấn đề và thuật giải 7](#_Toc501752363)

[3.3.1. Vấn đề “Phân tích cú pháp input” 7](#_Toc501752364)

[3.3.2. Xử lí dữ liệu đã phân tích được 7](#_Toc501752365)

[3.3.3. Xây dựng các chương trình con trong package 7](#_Toc501752366)

[3.3.4. Những vấn đề khác 7](#_Toc501752367)

[CHƯƠNG 4: ỨNG DỤNG 8](#_Toc501752368)

[4.1. Giới thiệu chương trình ứng dụng 8](#_Toc501752369)

[4.2. Cài đặt và mô tả 8](#_Toc501752370)

[4.3. Kết quả chạy chương trình 8](#_Toc501752371)

[CHƯƠNG 5: KẾT LUẬN 9](#_Toc501752372)

[5.1. Kết quả đạt được 9](#_Toc501752373)

[5.2. Hạn chế 9](#_Toc501752374)

[5.3. Hướng phát triển 9](#_Toc501752375)

[TÀI LIỆU THAM KHẢO 10](#_Toc501752376)

# PHÂN CÔNG CÔNG VIỆC

Chỗ này dành cho phân công công việc và đánh giá nội bộ

# CHƯƠNG I: GIỚI THIỆU

## 1.1. Mục đích đề tài và mô tả

## 1.2. Ứng dụng thực tiễn

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Tổng quan

## 2.2. Về MapleSoft và OpenMaple API

## 2.3. Về các thư viện thứ ba

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ

## 3.1. Phân tích bài toán

### 3.1.1. Xây dựng bot đơn giản

Bởi mục tiêu của đồ án này là xây dựng bot nên ta cần xây dựng tới một bot đơn giản làm nền tảng để xây dựng đồ án cho sau này. Bot xét theo cách ta khái niệm là phần mềm được cung cấp và xây dựng thuật toán làm sao để có thể hiểu được đầu vào, phân tích đầu vào và cung cấp một đầu ra đáp ứng được những yêu cầu mà đầu vào cung cấp cho phần mềm. Vậy trong bài toán này, ta cần xây dựng được một giao diện mà người dùng có thể giao tiếp được và xây dựng một bộ phân tích đầu vào đơn giản để có thể khởi động được bước tiếp theo “Xây dựng hệ thống khảo sát hàm số”.

### 3.1.2. Xây dựng hệ thống khảo sát hàm số

Sau khi qua bước phân tích đầu vào, ta đưa dữ liệu vừa phân tích được vào hệ thống khảo sát hàm số. Do việc xây dựng khảo sát hàm số trên .NET phức tạp và đòi hỏi nhiều yêu cầu, trong đồ án này sử dụng một API thứ ba tên là OpenMaple cho phép gọi Maple thực hiện bài toán từ dữ liệu mà ta rút trích được từ phần mềm được xây dựng từ ngôn ngữ .NET. Đôi nét về OpenMaple API đã được giới thiệu tại mục 2.2. Cách thức mà API hoạt động như thế nào sẽ được miêu tả chi tiết tại mục 3.3.2.

Nhờ vào MapleSoft, hai mục đích chính cần thực hiện là:

- Xây dựng package thực hiện khảo sát hàm số.

- Làm sao để phần mềm chính đưa dữ liệu đã phân tích được cho Maple xử lý và trả về kết quả?

### 3.1.3. Các yêu cầu khác

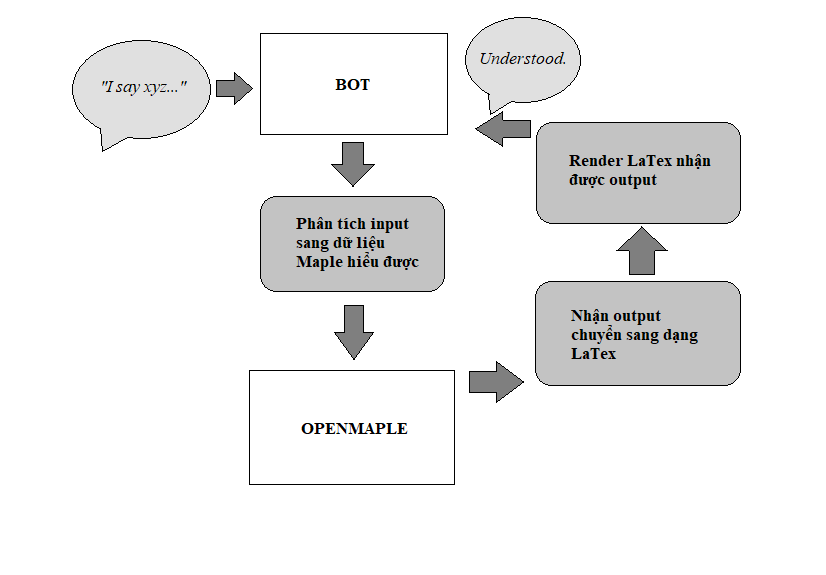
Tuy nhiên, khi Maple trả kết quả về, kết quả sẽ ở dạng thô, ví dụ kết quả:

Dao ham cua y=3x^3-x^2+1 la y’=9x^2-2x

Điều này có thể gây khó chịu cho người dùng, từ đây, ta cần những phương pháp thay đổi kết quả dạng thô sang kết quả đẹp như kết quả được hiển thị trên Maple. Các phương pháp được thử (áp dụng) đến:

- Chuyển đổi kết quả sang dạng LaTex.

- Render kết quả ở dạng LaTex bằng một thư viện thứ ba trợ giúp hiển thị kết quả lên màn hình.

Từ đây, ta có sơ đồ chuẩn thể hiện những gì ta cần thực hiện.

*Hình 3.1.3. Sơ đồ thực hiện bot giải toán*

## 3.2. Cấu trúc dữ liệu và cách biểu diễn các trạng thái khác của bài toán

### 3.2.1. Thực hiện trên Microsoft .NET

Những vấn đề ta cần xử lý trên nền .NET như sau:

- Cho phép người dùng nhập liệu trên WinForm qua một textbox duy nhất có thể thay đổi nhập liệu được và gửi qua một textbox xuất hiện dòng chat giữa bot và người dùng, đồng thời lưu dữ liệu vào một biến string nhằm đưa qua bước phân tích dữ liệu.

- Cần xử lý chuỗi string đầu vào và trả về kết quả trả lời cho Bot.

- Khi tìm được chuỗi string báo cho Bot biết cần phải giải toán, cần đưa phần mềm tới một hàm gọi package của OpenMaple và trả về kết quả string thô.

- Sau đó, đưa qua một hàm khác xử lý chuỗi string thô qua LaTex và thực hiện render LaTex.

Từ những vấn đề cần giải quyết, ta có cấu trúc dữ liệu đơn giản như sau:

string temp; //nhận chuỗi string đầu vào

string CalltheAnswer (string temp) //phân tích chuỗi đầu vào, đồng thời xuất kết quả thích hợp

string TinhHamso(string equa) //nhận giá trị phân tích được từ hàm CalltheAnswer, gọi OpenMaple xử lý và trả về kết quả thô

void renderLatex(string outOld) //xử lý chuỗi thô qua LaTex và thực hiện render LaTex đồng thời xuất trên màn hình

### 3.2.2. Thực hiện trên package thuộc MapleSoft

Từ yêu cầu bài toán của đề tài, những bài toán cần phải tính toán bao gồm: Xét tính đơn điệu của hàm số, Tìm các điểm cực trị của hàm số, Tìm phương trình tiếp tuyết của hàm số và Khảo sát đồ thị hàm số. Ta rút trích được bốn bài toán cần thực hiện, trong đó, bài toán Xét tính đơn điệu của hàm số và Tìm các điểm cực trị của hàm số thuộc bài toán lớn Khảo sát đồ thị hàm số. Cả bốn bài toán đều cần thực hiện đạo hàm.

### 3.2.3. Các yêu cầu thêm khác

## 3.3. Các vấn đề và thuật giải

### 3.3.1. Vấn đề “Phân tích cú pháp input”

Đối với vấn đề phân tích cú pháp, ta chia ra làm ba trường hợp lớn như sau: Các dữ liệu đầu vào xã giao, dữ liệu đầu vào nhằm mục đích giải toán và các dữ liệu khác không xử lý được. Bài toán cần input và output như sau:

INPUT: Chuỗi string đầu vào

OUTPUT: Xuất chuỗi đầu ra cần hiển thị trên màn hình

Đối với đối tượng là các dữ liệu đầu vào xã giao, như những câu đơn giản “chào”, “tạm biệt”, v.v… ta đưa tất cả những chuỗi string có thể xuất hiện từ dữ liệu đầu vào là xã giao vào một danh sách, mỗi lần người dùng nhập liệu, nếu có thể tìm được từ nhập liệu ấy trong danh sách, phần mềm sẽ trả chuỗi string ta định trước để xuất ra màn hình.

Đối với đối tượng là các dữ liệu đầu vào nhằm mục đích giải toán, ta có thể thử tìm trên chuỗi đầu vào những chuỗi có liên quan đến cấu trúc của hàm số. Cấu trúc của hàm số được biểu diễn như sau:

Hướng đi để giải bài toán này là ta duyệt chuỗi, nếu phát hiện ra chuỗi có liên hệ như cấu trúc hàm số, ta đưa chuỗi vào duyệt ra đoạn chuỗi con chứa hàm số. Nhằm mục đích đơn giản, thay vì ta tìm trên đoạn chuỗi nhập liệu thêm các từ khóa để biết cần phải giải bài toán nào, ta có thể điều khiển bot hỏi người dùng cần phải giải bài toán nào.

Đối với đối tượng là các dữ liệu đầu vào ta không lường trước được, ta đơn giản chỉ cần bảo bot trả về một chuỗi cụ thể.

Thuật giải được thể hiện trên mã giả và sơ đồ khối như sau:

int whatMath = -1;

string CallTheAnswer(string inp, int whatMath)

{

List<string> normalTalk = new List<string>(new string[] {//liệt kê các chuỗi đơn giản});

if(inp ∊ normalTalk) //dữ liệu đầu vào xã giao

{

return //xã giao

}

else(//tìm ra inp có dạng hàm số)

{

string formulas;

//tìm hàm số trong chuỗi

whatMath = 0; //đánh dấu đã nhận được hàm số

}

else(//nhận được dấu hiệu chọn bài toán từ whatMath)

{

string output = TinhHamso(formulas, whatMath);

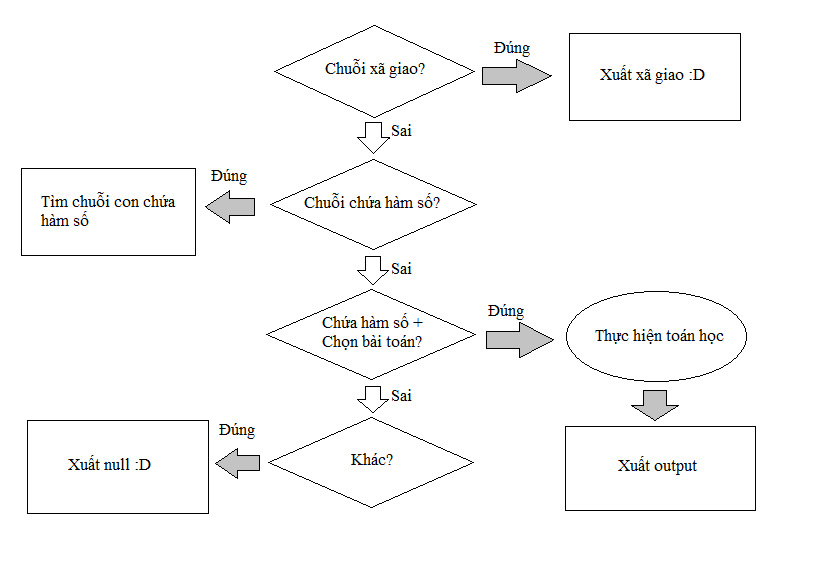
//thực hiện bài toán

return output; //trả về kết quả

}

else return “”; //các kết quả khác trả về chuỗi rỗng

}

*Hình 3.3.1. Sơ đồ thực hiện phân tích cú pháp*

### 3.3.2. Xử lí dữ liệu đã phân tích được

Sau khi đã chọn được bài toán cần giải và tìm được hàm số cần giải, ta xử lí dữ liệu đã phân tích được sao cho Maple có thể hiểu. Như ở mục 2.2 đã nêu, OpenMaple là một nền tảng cho phép người dùng được sử dụng bộ máy tính toán của Maple thông qua dynamic-link library (.dll). Để sử dụng, ta thêm file maplec.dll trong folder installed của Maple thêm vào thư mục .NET, gọi thư viện thông qua Runtime.InteropServices.DllImport. Để sử dụng, ta khai báo một số biến và lớp được miêu tả trong file hướng dẫn của Maple. Ở đây ta cho phép OpenMaple nhận chuỗi hàm số đã được xử lý thông qua hàm: MapleEngine.EvalMapleStatement(IntPtr kv, string input); Hàm này tính toán chuỗi input và xuất output ra một hàm khác là cbText(IntPtr data, int tag, String output), ở hàm này, output ta nhận được là cbText, ta lấy kết quả trả về ở cbText.

Nhằm xử lí dữ liệu để hàm EvalMapleStatement hiểu, ta chuyển chuỗi string hàm số sang câu lệnh mà Maple có thể hiểu, như câu lệnh cho phép Maple truy cập tới package giải toán mà sẽ được đề cập dưới đây.

### 3.3.3. Xây dựng các chương trình con trong package

### 3.3.4. Những vấn đề khác

# CHƯƠNG 4: ỨNG DỤNG

## 4.1. Giới thiệu chương trình ứng dụng

## 4.2. Cài đặt và mô tả

## 4.3. Kết quả chạy chương trình

# CHƯƠNG 5: KẾT LUẬN

## 5.1. Kết quả đạt được

## 5.2. Hạn chế

## 5.3. Hướng phát triển

# TÀI LIỆU THAM KHẢO