

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN** **1**

**ĐỀ TÀI**: ỨNG DỤNG PHÂN TÍCH VÀ TÍNH GIÁ TRỊ CỦA BIỂU THỨC SỬ DỤNG THUẬT TOÁN RPN

**SVTH 1 : Phạm Huỳnh Thanh Lâm**

**MSSV: 17110168**

**SVTH 2 : Chu Minh Hoàng**

**MSSV:17110**

**TP. Hồ Chí Minh, tháng năm 2019**

**MỤC LỤC**

[PHẦN MỞ ĐẦU 1](#_Toc22372561)

[1. Lý do chọn đề tài: 1](#_Toc22372562)

[2. Mục đích nghiên cứu đế tài: 1](#_Toc22372563)

[3. Yêu cầu : 1](#_Toc22372564)

[4. Phương pháp nghiên cứu: 1](#_Toc22372565)

[PHẦN NỘI DUNG 2](#_Toc22372566)

[I. Đặc tả 2](#_Toc22372567)

[II. Phân công 3](#_Toc22372568)

[III. Thiết kế 3](#_Toc22372569)

[1) Thuật toán 3](#_Toc22372570)

[2) Thiết kế lớp 4](#_Toc22372571)

[3) Thiết kế giao diện 6](#_Toc22372572)

[PHẦN KẾT LUẬN 8](#_Toc22372573)

[1. Kết quả đạt được 8](#_Toc22372574)

[2. Kết luận 8](#_Toc22372575)

[TÀI LIỆU THAM KHẢO 9](#_Toc22372576)

**DANH MỤC CÁC HÌNH**

[Hình 1: Ví dụ về thuật toán 3](#_Toc22372771)

[Hình 2: Giao diện màn hình 6](#_Toc22372772)

[Hình 3: Giao diện nhập biễu thức 7](#_Toc22372773)

[Hình 4:Kết quả 7](#_Toc22372774)

[Hình 5:Kết quả sai 7](#_Toc22372775)

# PHẦN MỞ ĐẦU

## Lý do chọn đề tài:

Vì việc tính toán toán một biểu thức một cách nhanh chóng và hiệu quả, đễ có được một kết quả chính xác với biểu thức cần tính. Đồng thời giup ôn lại những phương thức về các đối tượng và class trong các môn học học trước đó.

## Mục đích nghiên cứu đế tài:

Mục đích của đề tài là tạo ra một chương trình tính toán đơn giản thông qua các ngôn ngữ Java và các tính chất của đối tượng trong class.Từ đó tạo ra một chương trình Java tính toán một biểu thức.

## Yêu cầu :

Yêu cầu được đưa ra trong đồ án là áp dụng phương áp chuyển từ trung tố sang hậu tố bằng thuật toán RPN.

Thực hiện được các phép tính đơn giản của một biểu thức bình thường ở dạng infix với các kiến thức đã học của những môn học trước.

## Phương pháp nghiên cứu:

Nghiên cứu tham khảo tra cứu thông tin và tài liệu thông qua sự cho phép của giáo viên.

Hoàn thiện đề tài với sự hướng dẫn của giáo viên.

# PHẦN NỘI DUNG

## Đặc tả

* Chương trình dùng để phân tích biểu thức và tính giá trị biểu thức đó dựa trên

chuỗi postfix khi ta muốn tính giá trị của một biểu thức toán học.

* Dữ liệu đầu vào là một biểu thức hay còn gọi là Infix (trung tố). Sau đó được chuyển sang chuỗi Postfix (hậu tố). Các số và phép tính nhập vào được gọi lần lượt là toán hạng và toán tử.
* Đặc tả các chức năng: Dùng để tính một biểu thức bất kì để có giá trị phép tính +, -, \*, / số thực.Trong đó các số thực : số tự nhiên( 1,2,….,362 ), số âm( -1,-2,…), số hữu tỷ (1/2,0.12,1.233 ở đây số thập phân được viết có dấu ‘.’ Thay cho dấu ‘,’) .Sử dụng để tính các phép tính trong ngoặc.

+Chuyển các biểu thức infix thành postfix, từ biểu thức postfix tiến hành tính toán kết quả.

+Sử dụng LIFO.

+Chia làm hai giai đoạn:

                    a.Chuyển trung tố thành hậu tố: Có 1 stack(chỉ chứa toán tử) và 1 danh sách liên kết đơn

* Ở đây ta dùng danh sách liên kết đơn vì khi dùng mảng ta có các nhược điểm .Nhưng một danh sách với chiều dài cố định ( trong bài toán chưa xác định được độ dài). Mảng đặc biệt thích hợp cho các tình huống mà ta biết trước được số lượng phần tử hoặc có thể xác định được khi chạy chương trình( chưa biết được kích thước bao nhiêu). Mảng là sự lựa chọn tuyệt vời để giữ các giá trị bởi yêu cầu bài toán không đòi hỏi một thứ tự lưu trữ cụ thể và các phép tính toán cũng không đòi hỏi gì khác ngoài việc duyệt qua toàn bộ các phần tử.( trong bài toán đòi hỏi lưu trữ 1 cách cụ thể). Sử dụng mảng không tối ưu được bộ nhớ, có thể thừa hoặc thiếu bộ nhớ khi xóa hoặc chèn phần tử vào mảng.
* Cho nên ở đây chúng em sử dụng linked list
* Một số lượng phần tử tùy ý và dễ dàng thay đổi kích thước, cũng như dễ dàng bỏ đi các phần tử bên trong nó.
* Sử dụng danh sách liên kết tối ưu được bộ nhớ. Vùng nhớ được cấp phát thêm khi cần chèn thêm phần tử mới, vùng nhớ được free khi xóa phần tử.Trong bài toán ta chỉ cần 1 nơi lưu trữ cũng như là duỵet mảng để làm nên việc sử dụng là tối ưu hơn sử dụng mảng)

b.Tính trung tố: Có 1 stack (chỉ chứa toán hạng)

* Xuất kết quả
* Bổ xung các công thức tính căn, mũ,lượng giác,..
* Chức năng kiểm tra sai sót trong khi nhập
* Giao diện: Chương trình được chạy trên Console đơn giản với dòng đầu là nhập biểu thức, dòng hai sẽ là biểu thức với kết quả được xuất ra

## Phân công

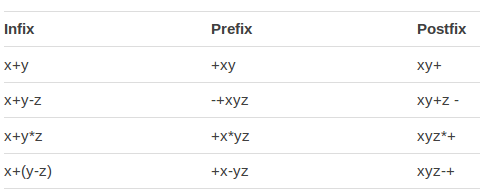
|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên SV | Công việc | Phẩn trăm đóng góp |
| 1 | Phạm Huỳnh Thanh Lâm | Tìm hiểu thuật toán,xác định toán tử tạo đối tượng Operater,hàm chuyển postfix,xử lý số âm,dương | 100% |
| 2 | Chu Minh Hoàng | Tìm hiểu thuật toán,phương thức chứa biểu thức linklist cho infix và postfix,tính kết quả,kiểm tra phần tử có là toán tử hay không | 100% |

## Thiết kế

### Thuật toán

Thuật toán được gọi là “ký pháp nghịch đảo Ba Lan” hoặc được viết tắt là RPN

(Reverse Polish notation), được phát minh vào khoảng giữa thập kỷ 1950 bởi một triết học gia và nhà khoa học máy tính Charles Hamblin người Úc.



Hình 1: Ví dụ về thuật toán

Thuật toán sẽ đọc từng token trong biểu thức infix từ trái qua phải, với mỗi token ta thực hiện các bước sau:– Nếu là toán hạng: cho ra output.– Nếu là dấu mở ngoặc “(“: cho vào stack– Nếu là dấu đóng ngoặc “)”: lấy các toán tử trong stack ra và cho vào output cho đến khi gặp dấu mở ngoặc “(“. (Dấu mở ngoặc cũng phải được đưa ra khỏi stack)– Nếu là toán tử: +Chừng nào ở đỉnh stack là toán tử và toán tử đó có độ ưu tiên lớn hơn hoặc bằng toán tử hiện tại thì lấy toán tử đó ra khỏi stack và cho ra output. +Đưa toán tử hiện tại vào stackSau khi duyệt hết biểu thức infix, nếu trong stack còn phần tử thì lấy các token trong đó ra và cho lần lượt vào output.

### Thiết kế lớp

#### Danh mục các lớp

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên lớp | Mục đích | SV phụ trách |
| 1 | Tinhtoan | Các phương thức để tính toán bài toán |  |
| 2 | Operator | Đối tượng chứa các toán tử,độ ưa tiên và sự so sánh chúng với nhau |  |
| 3 | Stack : st,sh  Giải thích: Stack được hỗ trợ trong java | Ngăn xếp dùng để chứa toán tử và toán hạng |  |

#### Phương thức trong mỗi lớp

1. TinhToan.java

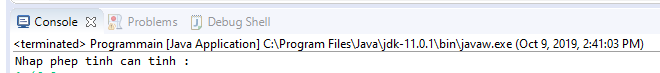
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file số thức tự dòng khai báo | SV phụ trách |
| 1 | **public** **boolean** isOperator(String c)  Input:Kí tự c  Ouput:True hoặc False  Giải thích: Sắp xếp chuỗi tăng dần với phương thức arry.sort và array.binarySearch xuất vị trí của toán tử trong mảng | Kiểm tra xem kí tự c có phải là các toán tử “+”,”-“,”\*”,”/”,”(“,”)” | Dòng 151 |  |
| 2 | **public** String Chuanam(String bieuthuc)  Input:String  Ouput:String  Giải thích: sử lý vấn đề về số âm  Vd-1+2,1--2=>1+2,--1+2=>+1+2 | Chuẩn hoá số âm hoặc việc nhập nhiều số âm trước khi làm bài | Dòng 158 |  |
| 3 | **public** LinkedList<String> ChuanHoa(String bieuthuc)  Input:Chuỗi biểu thức <String>  Ouput:Danh sách liên kết chứa các phần tử của chuỗi trên | Chuẩn hoá đưa chuỗi lưu thành 1 danh sách liên kết đơn | Dòng 118 Class Tinhtoan |  |
| 4 | **public** Linked List<String> chuyendoi(LinkedList<String> elementMath)  Input: LinkedList infix  Ouput: LinkedList postfix | Chuyển từ infix sang fostfix | Dòng 55 Class Tinhtoan |  |
| 5 | **public** Double Xuatkq(String[] doi)  Input:Chuỗi postfix  Ouput:Một số | Tính toán các phép tính đưa ra kết quả cuối cùng | Dòng 27 Class Tinhtoan |  |
| 6 | **public** String Xuatkq(LinkedList<String> doi)  Input: LinkedList  Ouput:String | Kết quả | Dòng 7 Class Toantu |  |

1. Operate.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file số thức tự dòng khai báo | SV phụ trách |
| 1 | **public** **char** getkitu()  **public** **int** getdouutien()  **public** Operator(**char** kitu, **int** douutien)  **public** Operator() | Khai báo operate gồm 2 thành phần toán tử và độ ưa tiên | Dòng 9 tới 29 |  |
| 2 | **public** **void** Chuoi(**char** elementMath)  Input: toán tử | Xét độ ưa tiên vaq kí tự cho đúng Operate cần xét | Dòng 31 |  |
| 3 | **public** Character[] xetdouutien(**char** elementMath)  Input: toán tử  Ouput: một mảng 2 phần tử toán tử và độ ưu tiên | Đưa toán tử và độ ưu tiên vào mảng để đưa vào stack | Dòng 38 |  |
| 4 | **public** **boolean** sosanh(Operator a, Operator b)  Input: Operate a và b  Ouput: True hay False | So sánh 2 toán tử với nhau coi toán tử a có <= toán tử b hay không | Dòng 47 |  |
| 5 | **public** **static** **int** GetPriority(**char** op)  Input: toán tử char  Ouput: độ ưu tiên int | Gán độ ưu tiên cho toán tử | Dòng 54 |  |

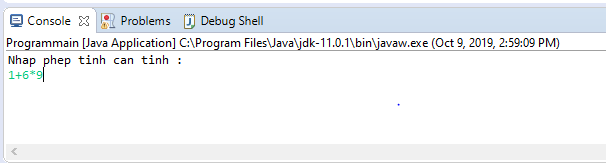
### Thiết kế giao diện

* Dòng in thông báo nhập biểu thức



Hình 2: Giao diện màn hình

* Dòng nhập biểu thức



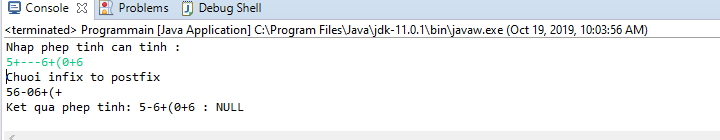
Hình 3: Giao diện nhập biễu thức

* Dòng in kết quả



Hình 4:Kết quả

* Với kết quả sai



Hình 5:Kết quả sai

# 

# PHẦN KẾT LUẬN

## Kết quả đạt được

## Ta có một chương trình tính toán biểu đơn giản với các phép tính toán học. Qua bài báo cáo trên ta được biết rõ hơn các phép tính của toán học và phương pháp chuyển trung tố sang hậu tố để máy tính thực hiện và từ hậu tố chúng ta tính được kết quả của một phép toán .Chúng ta còn biết được các ứng dụng của stack và linklist trong cấu trúc dữ liệu mà ta sử dùng ở đây.

## Kết luận

1. **Ưu điểm**

Sau khi hoàn thành đồ án này thì chúng ta tự nhận thấy đã hoàn thành tốt phần nào những yêu cầu cầu của đề tài đã đặt ra,và cả những mục tiêu mà nhóm chúng tôi đã tự đặt ra cho bản thân. Nhưng bên cạnh đó trong quá trình làm thì còn những mục tiêu mà nhóm chúng ta tự nhận thấy là mình chưa thực hiện tốt lắm và còn thiếu sót trong quá trình thực hiện.Và chúng em đã sữa lại cho đúng tính chất của các môn chúng em học trước đây điển hình là hướng đối tượng và các phương thức lưa trữ dữ liệu.

1. **Nhược điểm**

Tuy nhiên đồ án này của chúng em tuy hoàn thành nhưng chưa hoàn thiên vì vẫn còn một số sai sót trong đồ án trong việc thực hiện các phép tính và chưa tính được các hàm mũ. Từ đó rút ra được những sai xót của bản thân mà chúng em cần khắc phục.

# TÀI LIỆU THAM KHẢO