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

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

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



**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG CALCULATOR**

**SVTH : LÊ MẠNH HÙNG**

**MSSV : 15110058**

**SVTH : NGUYỄN BÁ LÊ AN**

**MSSV : 15110001**

**Tp. Hồ Chí Minh, tháng 11 năm 2018**

**MỤC LỤC**

[**MỤC LỤC** 3](#_Toc531069876)

[**Danh mục các hình** 4](#_Toc531069877)

[**Danh mục các bảng** 4](#_Toc531069878)

[**1.** **Mô tả** 5](#_Toc531069879)

[**2.** **Mục tiêu** 5](#_Toc531069880)

[**3.** **Thuật toán** 6](#_Toc531069881)

[**4.** **Thiết kế giao diện** 8](#_Toc531069882)

[**5.** **Các hàm chính** 8](#_Toc531069883)

[**6.** **Cài đặt và kiểm thử** 9](#_Toc531069884)

[**6.1.** **Cài đặt** 9](#_Toc531069885)

[**6.2.** **Kiểm thử** 9](#_Toc531069886)

[**7.** **Kết luận** 10](#_Toc531069887)

[**7.1.** **Ưu điểm và nhược điểm** 10](#_Toc531069888)

[**7.2.** **Hướng phát triển** 10](#_Toc531069889)

[**8.** **Tài liệu tham khảo** 10](#_Toc531069890)

**Danh mục các hình**

Hình 1: Ví dụ về thuật toán.

Hình 2: Form giao diện Calculator.

**Danh mục các bảng**

Bảng 1: Mức độ ưu tiên toán tử

**1. Mô tả**

Ứng dụng Calculator trong thực tế giúp mọi người tính toán các biểu thức đại số nhanh hơn, tiết kiệm thời gian trong việc tính toán.

Ứng dụng Calculator trong lập trình, các biểu thức đại số được sử dụng hằng ngày đều được biểu diễn dưới dạng trung tố (infix). Cách biểu diễn này rất dễ hiểu với con người vì hầu hết các toán tử (+, -, \*, /) đều là toán tử hai ngôi và chúng phân cách giữa hai toán hạng với nhau. Tuy nhiên đối với máy tính, để tính được giá trị của một biểu thức đại số theo dạng này không đơn giản như ta vẫn làm. Để khắc phục điều đó, máy tính cần chuyển cách biểu diễn các biểu thức đại số từ trung tố sang một dạng khác là tiền tố (prefix) hoặc hậu tố (postfix). Trong đồ án này nhóm em sẽ ứng dụng Stack để chuyển biểu thức trung tố sang hậu tố để tính giá trị biểu thức thông qua ngôn ngữ minh họa C#. Chương trình này thực hiện được các phép tính số nguyên và các phép tính số thực, bao gồm các phép toán sau: phép (+), (-), (\*), (/), (sin), (cos), (tan), (cotg), căn bậc 2, phép chia lấy dư (%). Chương trình sẽ tính giá trị biểu thức đúng như các quy tắc tính giá trị biểu thức thông thường, ví dụ như biểu thức: (3+2)\*(5-3) thì chương trình sẽ thực hiện các phép toán trong ngoặc trước và thực hiện phép nhân sau.

**2. Mục tiêu**

Mục tiêu cần đạt được trong đề tài xây dựng ứng dụng Calculator là:

* Về giao diện:
* Sử dụng các Toolbox của C# Winform: button, textbox để tạo giao diện cho ứng dụng.
* Về thuật toán:
* Ứng dụng được Stack để chuyển một biểu thức trung tố sang biểu thức hậu tố và sau đó tính giá trị biểu thức hậu tố đó.
* Thực hiện được biểu thức có chứa các toán tử như sau: +, -, \* , / , sin, cos, tag, cotg, sqrt, %, (, )
* Thực hiện đúng cách tính giá trị biểu thức: ưu tiên thực hiện biểu thức trong dấu ngoặc trước, sau đó mới thực hiện phép nhân, chia rồi đến cộng, trừ.

**3. Thuật toán**

Trước khi đi vào thuật toán, ta phải hiểu thế nào là toán tử, toán hạng, biểu thức trung tố và hậu tố là gì:

* Toán tử là các ký hiệu như: +, -, \*, / , %, (, ),…
* Toán hạng là các phần tử như: a,b,1,2,…
* Biểu thức trung tố (Infix): được biểu diễn bằng cách đặt toán tử giữa hai toán hạng. Ví dụ: 1+3
* Biểu thức hậu tố (Postfix): được biểu diễn bằng cách đặt các toán tử ra sau các toán hạng. Ví dụ: 13+

Thuật toán chuyển một biểu thức trung tố (Infix) sang dạng hậu tố (Postfix):

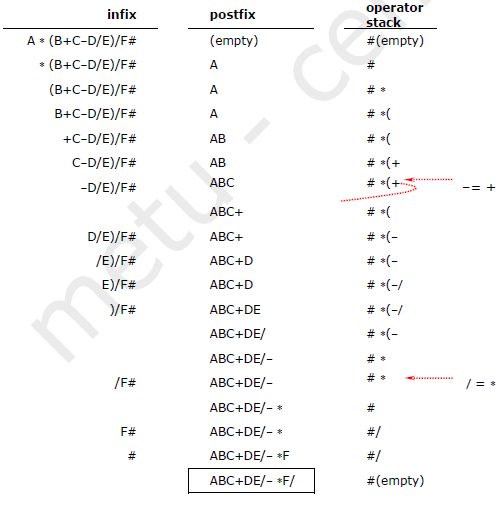
* Giả sử S là một biểu thức được cho ở dạng trung tố. Khởi tạo 2 Stack: Sh, St.
* Stack Sh dùng để lưu trữ toán hạng, Stack St dùng để lưu trữ toán tử.
* Duyệt S từ trái sang phải:
* Nếu S[i] là toán hạng, Push vào Sh.
* Nếu S[i] = ’(’, Push vào St.
* Nếu S[i] là toán tử có độ ưu tiên cao hơn toán tử hiện có trên đỉnh St thì Push vào St
* Nếu S[i] là toán tử có độ ưu tiên thấp hơn hoặc bằng toán tử hiện có trên đỉnh St thì Pop toán tử có trên đỉnh St và Pop 2 toán hạng trên đỉnh Sh. Sau đó, thực hiện phép tính, kết quả Push vào Sh.Tiếp theo, Push S[i] vào St.
* Nếu S[i] = ’)’ thì Pop toán tử có trên đỉnh St và Pop 2 toán hạng trên đỉnh Sh. Sau đó, thực hiện phép tính, kết quả Push vào Sh. Tiếp theo, bỏ dấu ‘(‘ gặp phải đầu tiên ra khỏi St.
* Thực hiện đến khi nào St rỗng và Sh còn 1 toán hạng duy nhất thì đó là kết quả.

Mức độ ưu tiên toán tử:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Toán tử | + | - | \* | / | % | sin | cos | tan | cotg | sqrt |
| Mức ưu tiên | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 |

Bảng 1: Mức độ ưu tiên toán tử

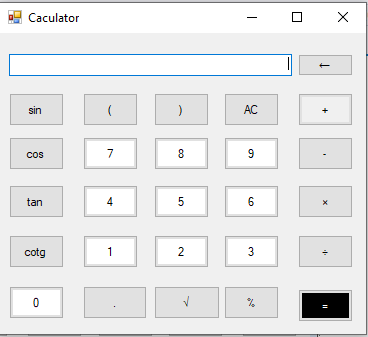
Xét ví dụ đơn giản sau:

****

Hình 1: Ví dụ về thuật toán.

**4. Thiết kế giao diện**

Giao diện gồm 1 form duy nhất:



Hình 2: Form giao diện Caculator.

**5. Các hàm chính**

Trong đề tài sẽ có những hàm chính sau đây:

* + - ThemKiTu()
    - XoaKiTu()
    - ThucHienPhepTinh()

Trong hàm ThucHienPhepTinh() sẽ có 2 vòng lặp:

* + - while (i < manhinh.Text.Length-1)
    - while (st.Count > 0)

Vòng lặp while (i < manhinh.Text.Length-1):

**6. Cài đặt và kiểm thử**

***6.1. Cài đặt***

Ứng dụng Caculator được cài đặt trên Visual Studio 2017, chạy trên Window10 x64 bit.

***6.2. Kiểm thử***

Các trường hợp đã thử:

TH1: 9-2\*3

TH2: 9-2+3

TH3: 11-5+22

TH4: 16 - - 15

TH5: (3+5) \* 3

TH6:

**7. Kết luận**

***7.1. Ưu điểm và nhược điểm***

**7.1.1. Ưu điểm**

**7.1.2. Nhược điểm**

***7.2. Hướng phát triển***

Hướng phát triển của đồ án Caculator là:

* Cần bổ sung thêm chức năng nhập dữ liệu từ bán phím.
* Thêm chức năng quay lại phép tính trước.
* Thêm chức năng tính toán như: giải phương trình bậc 1, bậc 2, phân số, lũy thừa, hàm mũ.
* Cải tiến lại giao diện để dễ dùng, đẹp hơn.

**8. Tài liệu tham khảo**

[1] https://yinyangit.wordpress.com/2011/01/26/algorithm-chuyển-biểu-thức-trung-tố-sang-tiền-tố-va-hậu-tố-bằng-stack/

[2] https://www.coursehero.com/file/11745204/12ARITHMETIC-EXPRESSIONS/