**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**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**SVTH : NGUYỄN ANH TUẤN**

**MSSV : 16110**

**SVTH : NGUYỄN HUỲNH VINH**

**MSSV : 16110269**

**TP. Hồ Chí Minh, tháng 12… năm 2017**

**LỜI NÓI ĐẦU**

Đầu tiên nhóm chúng em xin cảm ơn nhà trường các cấp thuộc khoa công nghệ thông tin, đã tổ chức cho chúng em thực hiện bài tập lớn, cho môn “cấu trúc và giải thuật” nhầm giúp chúng em trau dồi những kỹ năng lập trình bằng các ngôn ngữ C,C# và các kỹ năng mềm:làm việc nhóm, phát hiện các vấn đề giải quyết vấn đề,.....

Tiếp theo, nhóm em xin chân thành cảm ơn thầy Trần Công Tú đã trực tiếp giãng dạy và hướng dẫn tận tình cho đề tài nhóm em.

Cuối cùng nhóm em cũng xin cảm ơn sự giúp đỡ các bạn cùng lớp , các bài hướng dẫn trên các diễn đàng .

**MỤC LỤC:**

[I. DANH MỤC HÌNH 4](#_Toc501046838)

[II. MÔ TẢ: MÁY TÍNH 5](#_Toc501046839)

[1.Ứng dụng: 5](#_Toc501046840)

[III. QÚA TRÌNH LÀM: 6](#_Toc501046841)

[1. Thiết kế giao diện: 6](#_Toc501046842)

[2. Mô tả code: 7](#_Toc501046843)

[a) Tạo stack 7](#_Toc501046844)

[b) *Tạo class xử lý hậu tố*( class hauto): 9](#_Toc501046845)

[c) Code giao diện: 15](#_Toc501046846)

[3. Mô tả công việc: 15](#_Toc501046847)

[4. Nhận xét: 15](#_Toc501046848)

# DANH MỤC HÌNH

***Tên hình Mô tả***

Máy tính 1 Hình ảnh máy tính nhóm vẽ

Máy tính 2 Hình ảnh máy tính có trên window

Stack1. Tạo Stack

Stack2. Thêm phần tử vào Stack

Stack3. Lấy phần tử khỏi Stack

Stack4. Lấy phần tử trên cùng của Stack

Stack5. Hàm kiểm tra rỗng

Stack6. Hàm đếm số lượng phần tử trong Stack

Hậu tố 1. Hàm ưu tiên

Hậu tố 2. Tổng quát cả hàm hauto (1)

Hậu tố 3. Tổng quát cả hàm hauto (2)

Hậu tố 4. Giải thích các dòng code trong hậu tố

Hậu tố 5. Giải thích các dòng code trong hậu tố

Hậu tố 6. Giải thích các dòng code trong hậu tố

Hậu tố 7. Giải thích các dòng code trong hậu tố

Hậu tố 8. Giải thích các dòng code trong hậu tố

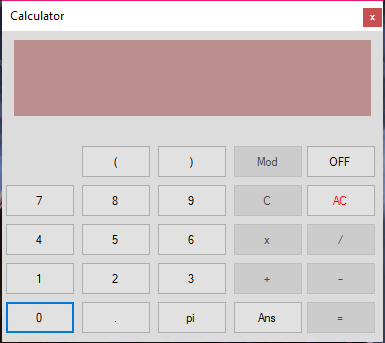
Chuyễn hậu tố 1. Hàm chuyenhauto giải thích các dòng code

Chuyển hậu tố 2. Hàm chuyenhauto giải thích các dòng code

# MÔ TẢ: MÁY TÍNH

## 1.Ứng dụng:

* Phần mềm dùng để tính toán các phép toán (+,-,\*,/ ,mod ) có độ lớn nhất định gặp thường ngày.



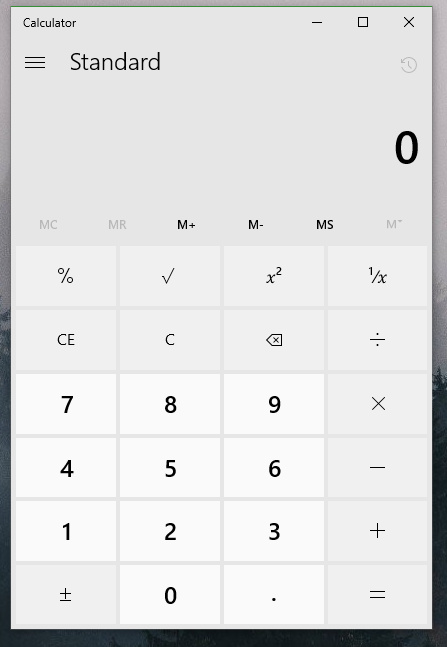
*Máy tính 1*

# QÚA TRÌNH LÀM:

## Thiết kế giao diện:

* Sử dụng winform ngôn ngữ c# (visual studio).
* Sử dụng 24 button mô tả các số ,các phép toán,thao tác xóa .
* Sử dụng 1 lable để mô tả màng hình thao tác nhập ,xuất .

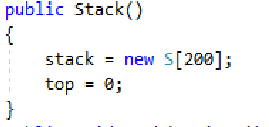
(chèn ảnh máy tính)..

* Thiết kế dựa trên các mô hình máy tính có sẳn trong win 10. 

*Máy tính 2*

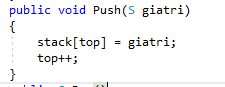
1. Mô tả code:Nhóm chia code ra ba phần
2. Tạo stack (class stack):

* Cài đặt stack bằng mảng một chiều.
* Trong Constructor:
* Khai báo một mảng 200 phần tử.
* Biến top =0.



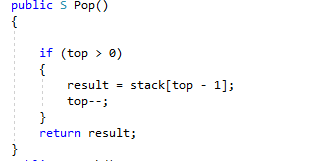
*Stack1.*

* Hàm Push(giatri) (hàm thêm phần tử vào stack):
* Biến giatri là biến được truyền vào khi gọi hàm Push().
* Biến giatri sẻ được lưu vào vị trị trí top trong stack.
* top++ sẽ tạo ra một vị trí mới.



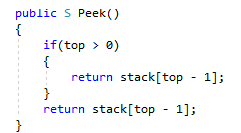
*Stack2.*

* Hàm Pop() (hàm lấy phần tử thứ top ra khỏi stack,sau khi lấy stack không còn phần tử đó):
* Nếu stack không rỗng , trả về giá trị ở top.
* Phần tử đó bị xóa khỏi Stack(top--).



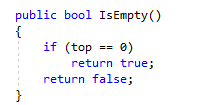
*Stack3.*

* Hàm Peek() (hàm trả về giá trị ở top):
* Nếu stack không rỗng , trả về giá trị ở top.

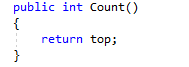


*Stack4.*

* Hàm IsEmpty() (kiểm tra Stack rỗng):
* Nếu top=0 có nghĩa là Stack rỗng.



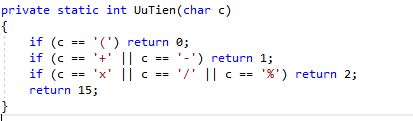
*Stack5.*

* Hàm Count() (hàm đếm số lượng phần tử Stack).

*Stack6.*

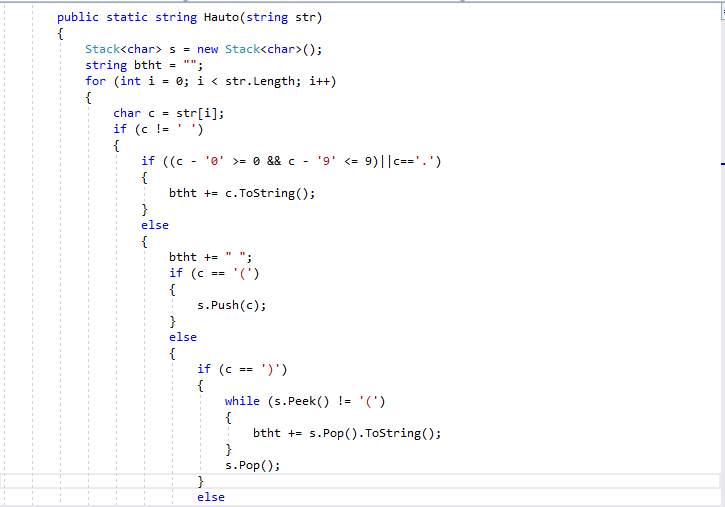
### Tạo class xử lý hậu tố( class hauto):

* Hàm UuTien(char c)
* Nếu là dấu ‘(‘ trả về 0.
* Nếu là dấu ‘+’ hoặc ‘-‘ trả về 1.
* Nếu là dấu ‘x’ hoặc ‘/’ hoặc’%’ trả về 2.
* Còn lại trả về 15.

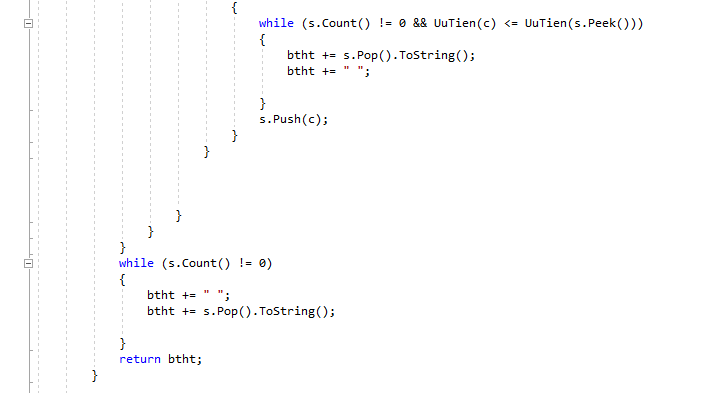


*Hậu tố 1.*

* Hàm ktspace(string str)
* Hàm dùng để kiểm tra khoảng trắng .
* Kiểm tra hết chuỗi truyền vào (string str) Nếu có khoảng trắng thì trả về True còn lại trả về false.
* Hàm Hauto(string str)



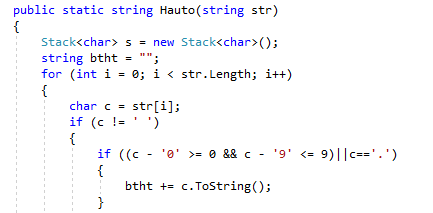
*Hậu tố 2.*



+ Tạo biến

*Hậu tố 3.*

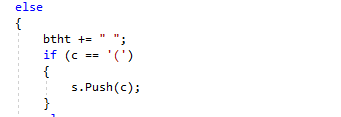
* Stack S, tạo một chuỗi để lưu biểu thức hậu tố(string btht).
* Kiểm tra chuỗi nhập vào,nếu không phải là khoãng trắng và là số từ 0 đến 9 thì lưu vào biểu thức hậu tố.



*Hậu tố 4.*

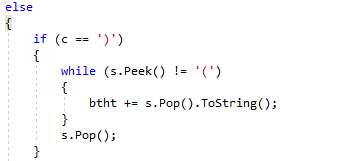
* Nếu không phải số thì thêm khoãng trắng vào biểu thức hậu tố.

Và nếu là dấu’(‘ thì thêm vào Stack.



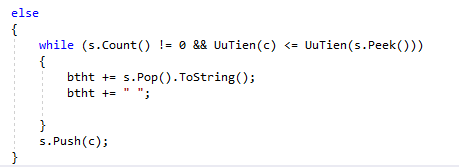
*Hậu tố 5.*

* Nếu không là’(‘ mà là ‘)’ thì cứ thêm hết phần tử trên vào biểu thức hậu tố cho đến khi gặp ‘(’ thì đưa ‘(’ra khỏi Stack.



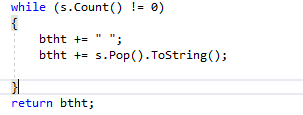
*Hậu tố 6.*

* Còn trường hợp ngược lại(các dấu còn lại) thì nếu còn phần tử trong Stack và độ ưu tiên của dấu bé hơn độ ưu tiên của top trong Stack thì ta thêm tị trí trên cùng Stack vào biểu thức hậu tố và đồng thời thêm khoảng trắng trong biểu thức hậu tố và sau đó là thêm dấu đó vào Stack.



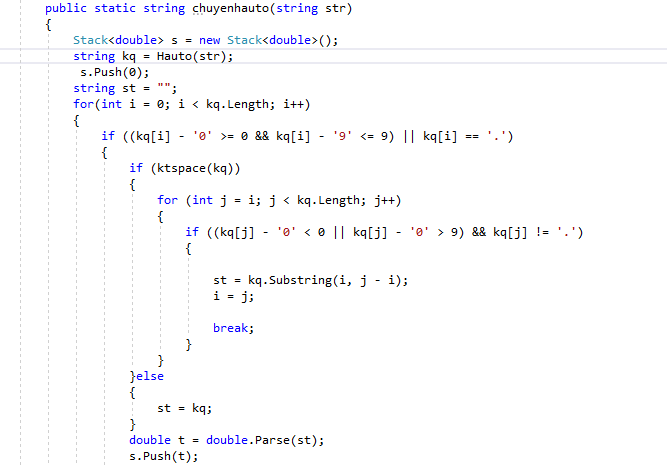
*Hậu tố 7.*

* Khi Stack có phần tử thì thêm vào biểu thức hậu tố khoảng trắng “ “ và thêm vào biểu thức hậu tố phần tử trên cùng , lập lại quá trình đến khi nào không còn phần tử nào trong Stack.



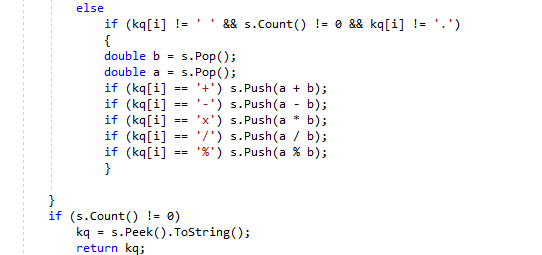
*Hậu tố 8.*

* Cuối cùng trả về biểu thức hậu tố .
* Hàm chuyenhauto(string str):
  + - * Stack S, một chuỗi string kq, bằng biểu thức hậu tố đã chuyển đổi.
      * Xét toàn chuỗi biểu thức hậu tố ,nếu là số và dấu chấm và kiểm tra chuỗi không rỗng. (tức chuỗi có khoảng cách).
      * Xét tại vị trí thõa mãn điều kiện trên chạy từ đó đến hết chuỗi nếu không là số hoặc dấu chấm (tức khoảng cách), thì đưa vào chuỗi kq,vị trí mới sẽ được cập nhật .
      * Còn nếu không chuỗi biểu thức hậu tố không có dâu cách rỗng thì đưa ra chuỗi kết quả (kq).
      * Tạo biến t kiểu double để trả về số và lưu t vào trong Stack.



*Chuyễn hậu tố 1.*

* Trường hợp ngược lại hàm if() đầu tiên trong hình *chuyển hậu tố 1*. Không phải là số hay dấu chấm tức có nghĩa là các dấu tính toán còn lại .
* Tạo thêm hai biến a,b để lưu hai số cùng của trên Stack để tính toán sau đó sẻ lưu vào lại trong Strack.
* Nếu vị trí đang xét là dấu ‘+’ thì lưu vào Stack là a+b.
* Nếu vị trí đang xét là dấu ‘-’ thì lưu vào Stack là a-b.
* Nếu vị trí đang xét là dấu ‘x’ thì lưu vào Stack là a\*b.
* Nếu vị trí đang xét là dấu ‘/’ thì lưu vào Stack là a/b.
* Nếu vị trí đang xét là dấu ‘%’ thì lưu vào Stack là a%b.
* Cuối cùng sau khi xét hết biểu thức hậu tố nếu Stack lưu kết quả không rỗng thì ta trả ra kết quả (string kq) bằng phần tử trên cùng của Stack.



*Chuyển hậu tố 2.*

### Code giao diện:

* Xử lý cơ bản các button.
* Ví dụ: button 1 hiện ra màn hình giá trị 1,...
* Xử lý màu của lable.

## Mô tả công việc:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên công việc | Nguyễn Anh Tuấn | Nguyễn Huỳnh Vinh | Ghi chú |
| 1 | Thiết kế Form xử lý các button | x | X | Chia button ra thiết kế |
| 2 | Code Stack | x |  |  |
| 3 | Code bieuthuchauto | X | X |  |
| 4 | Code chuyenbieuthuchauto | X | X |  |
| 5 | Tìm lỗi | X | X |  |
| 6 | Sữa lỗi | x | x |  |

## Nhận xét:

Mức độ hoàn thiện: khoảng 70% thõa mảng các phép toán cơ bản, các phép toán liên hoàn,...

Khó khăn khi thực hiện: Thực hiện với ngôn ngữ mới C#,cần phải nghiên cứu các thao tác thuật ngữ, có nhiều lỗi phát sinh khi thực hiện lỗi về thuật toán, lỗi về code .

Khắc phục:Nghiên cứu các bài hướng dẫn trên youtube, google, có học nhóm để đưa ra vấn đề,thống nhất đưa ra hương khắc phục hợp lý,kết hợp với sự hỗ trợ của thầy Trần Công Tú.

Ý tưởng phát triển: Phát triển hỗ trợ thêm như chương trình bậc 2,bặc 3 , cos,sin,tan,...Xuất ra thành một chương trình có thể sử dụng rộng rãi, cãi thiện giao diện để dể nhìn dể thao tác hơn.

Khuyết điểm:chưa hỗ trợ nhiều phương thức cos,sin,tan, giải phương trình bặc 2,3.

Ưu điểm: Thõa mảng các phép toán cơ bản, các phép toán liên hoàn, giao diện dể nhìn cơ bản ,xử lý các dấu ngoặc lồng nhau.

TÀI LIỆU THAM KHẢO

Sách:  
“ cấu trúc và giải thuật”

EBOOK\_CTDL

Wed:

“<https://codereview.stackexchange.com>”

“http://vietjack.com”