**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**

**Đề tài: Xây dựng ứng dụng Calculator**

**SVTH : Ngô Thanh Tài**

**MSSV : 16110201**

**SVTH : Trần Tuấn Kiệt**

**MSSV : 16110131**

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

**Mục lục**

[**NỘI DUNG** 3](#_Toc500836081)

[***1.*** ***Mô tả project*** 3](#_Toc500836082)

[**1.1.** **Công dụng của chương trình** 3](#_Toc500836083)

[**1.2.** **Sơ đồ nội dung phần mềm** 4](#_Toc500836084)

[4](#_Toc500836085)

[***2.*** ***Mô tả quá trình làm*** 4](#_Toc500836086)

[**2.1.** **Thiết kế các Class** 4](#_Toc500836087)

[**2.2.** **Thiết kế giao diện** 5](#_Toc500836088)

[**a.** **Màn hình chính** 5](#_Toc500836089)

[**2.3.** **Thiết kế các Code chính** 6](#_Toc500836090)

[***Cái đặt và kiểm thử*** 10](#_Toc500836091)

[**2.4.** **Cài đặt** 10](#_Toc500836092)

[**2.5.** **Thử nghiệm** 11](#_Toc500836093)

[***3.*** ***Phân công công việc*** 11](#_Toc500836094)

[***4.*** ***Kết luận*** 11](#_Toc500836095)

[**a.** **Kết quả** 11](#_Toc500836096)

[**b.** **Ưu điểm** 12](#_Toc500836097)

[**c.** **Khuyết điểm** 12](#_Toc500836098)

# **NỘI DUNG**

1. ***Mô tả project***
   1. **Công dụng của chương trình**

* Giúp người dùng tính toán và cho biết dạng hậu tố của các dãy phép tính.
  1. **Sơ đồ nội dung phần mềm**

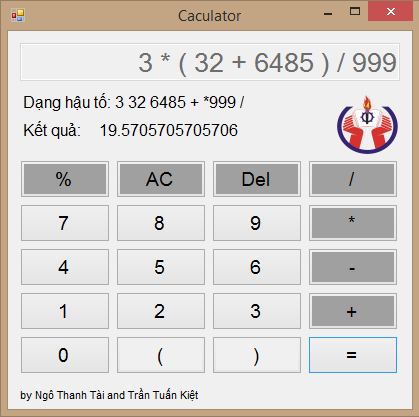
* 1. **Sơ đồ nội dung phần mềm**

1. ***Mô tả quá trình làm***
   1. **Thiết kế các Class**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Class** | **Công dụng** |
| 1 | Node | Dùng để khai báo, khởi tạo và chứa các hàm liên quan của Node |
| 2 | Stack | Dùng để khai báo, khởi tạo và chứa các hàm liên quan của Stack như Pop(), Push(), … |
| 3 | Calculator | Thực hiện các giải thuật liên quan đến Node, Stack để tính hậu tố và kết quả của chuỗi phép tính. |

* 1. **Thiết kế giao diện**

1. **Màn hình chính**



Màn hình chính gồm:

* Một textbox dùng để chứa dãy phép tính.
* 2 label phía dưới textbox dùng để hiển thị dạng hậu tố và kết quả.
  1. **Thiết kế các Code chính**

#### Lớp Node.cs:

class Node

{

String info;

Node pNext;

public Node()

{

info = "";

pNext = null;

}

public void createNode(string s)

{

info = s;

pNext = null;

}

public void Next(Node x)//dung trong push

{

pNext = x;

}

public string getInfo()

{

return info;

}

public Node getNext()

{

return pNext;

}

}

#### 

#### Lớp Stack.cs:

class Stack

{

Node pTop;

public Stack()

{

pTop = new Node();

pTop.createNode("#");

}

public void Push(string s)

{

Node p = new Node();

p.createNode(s);

if (pTop == null)

pTop = p;

else

{

p.Next(pTop);//pNext cua p tro den pTop

pTop = p;

}

}

public string Pop()

{

if (pTop.getInfo() != "#")

{

string s;

s = pTop.getInfo();

pTop = pTop.getNext();

return s;

}

else return "#";

}

public string getTop()

{

return pTop.getInfo();

}

}

#### Lớp Calculator.cs:

private void hauTo(List<string> IN,List<String>OUT)

{

string x;

for (int i = 0; IN[i] != "\0"; i++)

{

if (isNumb(IN[i]) == 1)

{

OUT.Add(IN[i]);

}

else

if (IN[i] == "(")

S.Push("(");

else

if (IN[i] == ")")

{

while (isEmpty() == 0)//

{

x = S.Pop();

if (x != "(")

{

OUT.Add(x);

}

else break;

}

}

else

{

while (isEmpty() == 0 && level(IN[i]) <= level(S.getTop()))

{

OUT.Add(S.Pop());

}

S.Push(IN[i]);

}

}

while (isEmpty() == 0)

{

OUT.Add(S.Pop());

}

OUT.Add("\0");

}

string calculate(List<string> OUT)

{

double x1, x2;

double kq;

int i = 0;

while (i < OUT.Count && OUT[i] != "\0")

{

while (OUT[i] == " ")

{

i++;

}

if (isNumb(OUT[i]) == 1)

{

double num = 0;

while (isNumb(OUT[i]) == 1)

{

num = num \* 10 + Convert.ToDouble(OUT[i]) ;

i++;

}

S.Push(Convert.ToString((num)));

}

else

{

x1 = Convert.ToDouble(Convert.ToDouble(S.Pop()));

x2 = Convert.ToDouble(Convert.ToDouble(S.Pop()));

switch (OUT[i])

{

case "+":

kq = x1 + x2;

break;

case "-":

kq = x2 - x1;

break;

case "\*":

kq = x1 \* x2;

break;

case "/":

kq = x2 / x1;

break;

case "%":

kq = x2 % x1;

break;

default:

return "0";

}

S.Push(Convert.ToString(kq));

}

i++;

}

return S.Pop();

}

* 1. **Cài đặt**
* Chương trình được cái đặt trên nền Visual Studio 2017, sử dụng ngôn ngữ C# và Project Window Form để thực hiện
* Hàm tính hậu tố được thực hiện theo công thức: Lấy chuỗi phép tính trên textbox và kiểm tra tính đúng đắn, sau đó đưa từng toán tử cũng như toán hạng vào trong List IN. Tiến hành dò từng phần tử của list IN, nếu là toán tử thì Add vào list OUT, nếu là toán tử thì ta Push vào Stack, nếu stack không rỗng thì xét mức độ ưu tiên của toán tử đang xét với toán tử nằm trên cùng của Stack rồi
* Hàm Calculate thực hiện bằng cách áp dụng thuật toán tính toán từ dãy hậu tố ra kết quả.
  1. **Thử nghiệm**
* Nhập dãy phép tính vào textbox, sau đó nhấn dấu “=” thì phía dưới sẽ hiện ra dạng hậu tố của chuỗi phép tính và kết quả.

1. ***Phân công công việc***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Mô tả công việc** | **Ngô Thanh Tài** | **Trần Tuấn Kiệt** | **Chi tiết** |
| 1 | Thiết kế Class Node | x | x | 50 - 50 |
| 2 | Thiết kế Class Stack | x | x | 50 - 50 |
| 3 | Thiết kế Class calculator | x | x | 80 - 20 |
| 4 | Thiết kế báo cáo | x | x | 50 - 50 |

1. ***Kết luận***
2. **Kết quả**

Sau khi trải qua nữa học kì nghiên cứu cũng như thực hiện với sự giúp đỡ của thầy Trần Công Tú, nhóm tôi đã hoàn thành được một chương trình Calculator đơn giản.

1. **Ưu điểm**

* Phần mềm có giao diện trực quan, dễ sử dụng, có thể nhập phép toán trực tiếp bằng tay hoặc click vào các button tương ứng có sẵn trên ứng dụng.
* Tính toán được với các toán tử, số âm, phần trăm,…

1. **Khuyết điểm**

* Ứng dụng chưa thể tính toán được số thực và các số lớn hơn giới hạn của Double.
* Vị trí hiển thị kết quả chưa được đẹp.
* Khi nhập trực tiếp bằng tay phải nhập khoảng trắng giữa các toán tử, toán hạng.