**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**

**XÂY DỰNG ỨNG DỤNG CALCULATOR**

**GVHD : ThS. TRẦN CÔNG TÚ**

**SVTH MSSV**

**ĐOÀN HUỲNH THIỆN 15110129**

**NGUYỄN QUỐC ANH 15110008**

**LỚP: 15110CL1**

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

**Mục lục**

[1. Giới thiệu 3](#_Toc501300711)

[2. Thiết kế giao diện 3](#_Toc501300712)

[3. Phân tích thuật toán 3](#_Toc501300713)

[3.1 Độ ưu tiên toán tử 3](#_Toc501300714)

[3.2 Các phương thức kiểm tra toán tử 4](#_Toc501300715)

[3.3 Chuyển biểu thức trung tố sang hậu tố 4](#_Toc501300716)

[3.4 Tính kết quả từ biểu thức hậu tố 5](#_Toc501300717)

[4. Cài đặt và kiểm thử 6](#_Toc501300718)

[4.1 Cài đặt 6](#_Toc501300719)

[4.1.1 Cài đặt code 6](#_Toc501300720)

[4.1.2 Cài đặt phần mềm 11](#_Toc501300721)

[4.2 Kiểm thử 13](#_Toc501300722)

[5. Phân công công việc 14](#_Toc501300723)

[6. Kết luận 14](#_Toc501300724)

[Tài liệu tham khảo 16](#_Toc501300725)

# **1. Giới thiệu**

Xây dựng một chương trình mô phỏng máy tính bỏ túi đơn giản dễ sử dụng với các phép tính cơ bản bằng cách sử dụng thuật toán ký pháp Balan để giải quyết các phép tính.

# **2. Thiết kế giao diện**

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Màn hình/Cửa sổ/Dialog | Giải thích thiết kế | Mục đích chính của màn hình |
| 1 | Màn hình chính | Đoàn Huỳnh Thiện  Thiết kế dựa trên ứng dụng Calculator của Microsof, được tùy chỉnh để phù hợp với yêu cầu. | Nhập biểu thức cần tính, cho biết biểu thức trung tố và kết quả. |

# **3. Phân tích thuật toán**

Đồ án sử dụng thuật toán “chuyển biểu thức trung tố sang hậu tố” hay còn được gọi là “ký pháp nghịch đảo Ba Lan”. Để chuyển một biểu thức từ trung tố sang hậu tố là dùng Stack. Các công việc cần làm là:

## ***3.1 Độ ưu tiên toán tử***

Xét độ ưu tiên của các toán tử. Các toán tử “\*, /” có cùng độ ưu tiên và cao hơn hai toán tử “+, -”.

## ***3.2 Các phương thức kiểm tra toán tử***

Trong thuật toán chuyển đổi này ta cần có các phương thức kiểm tra xem một thành phần của chuỗi có phải là toán tử không.

## ***3.3 Chuyển biểu thức trung tố sang hậu tố***

Trước khi chuyển đổi thì ta sẽ tách chuỗi trung tố thành các token.

Thuật toán để chuyển đổi biểu thức từ trung tố sang hậu tố: đọc từng token trong biểu thức trung tố từ trái qua phải, với mỗi token ta thực hiện các bước sau:

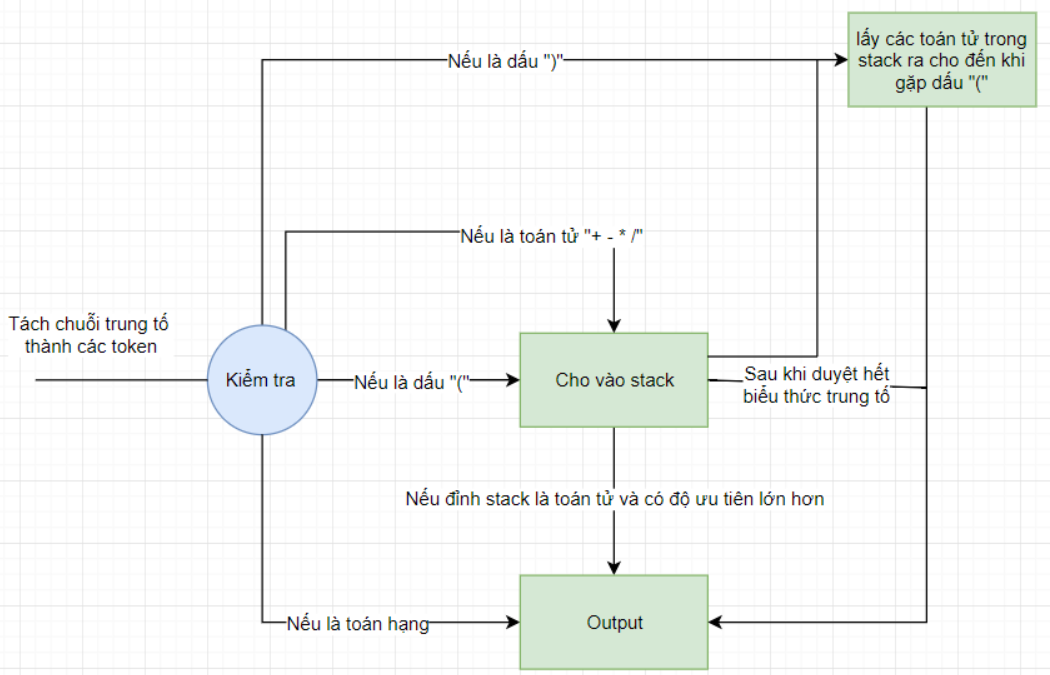
* Nếu là toán hạng: cho ra output.
* Nếu là dấu mở ngoặc “(“: cho vào stack
* Nếu là dấu đóng ngoặc “)”: lấy các toán tử trong stack ra và cho vào output cho đến khi gặp dấu mở ngoặc “(“. (Dấu mở ngoặc “(“ cũng phải được đưa ra khỏi stack)
* Nếu là toán tử:

• 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 stack

* Sau khi duyệt hết biểu thức trung tố, nếu trong stack còn phần tử thì lấy các token trong đó ra và cho lần lượt vào output.

Sơ đồ khối chuyển biểu thức trung tố sang hậu tố



## ***3.4 Tính kết quả từ biểu thức hậu tố***

Để tính kết quả từ biểu thức hậu tố ta thực hiện theo quy tắc sau:

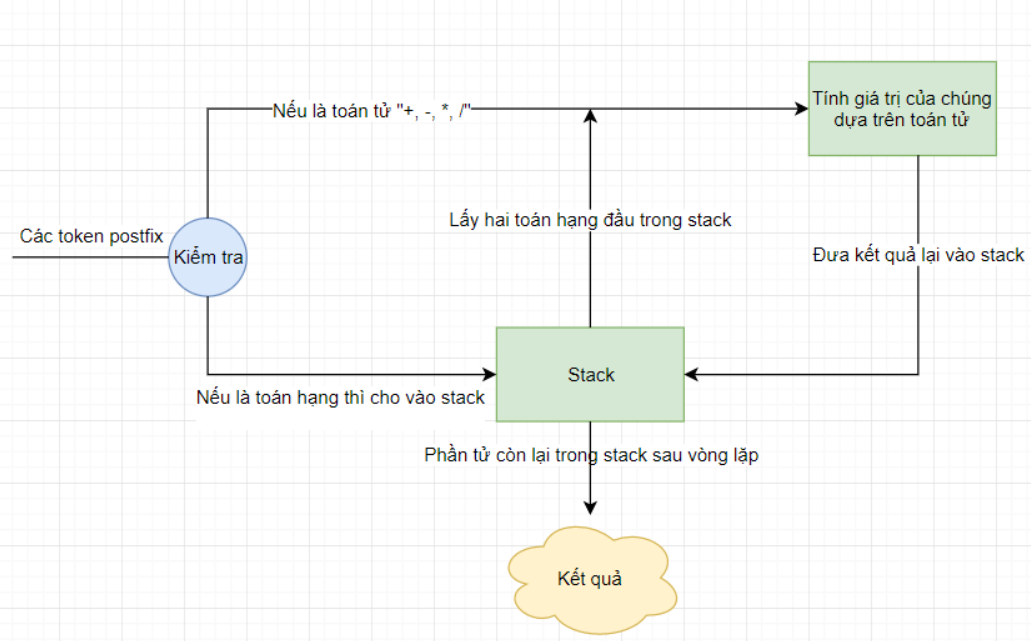
* Lặp qua các token của của biểu thức postfix từ trái qua phải:

• Nếu là toán hạng: push vào stack

• Nếu là toán tử: pop hai toán hạng trong stack ra và tính giá trị của chúng dựa vào toán tử này. Push kết quả đó lại vào stack.

* Phần tử còn sót lại trong stack sau vòng lặp chính là kết quả của biểu thức.

Sơ đồ khối tính kết quả từ biểu thức hậu tố



# **4. Cài đặt và kiểm thử**

## ***4.1 Cài đặt***

### **4.1.1 Cài đặt code**

#### **a. Xét độ ưu tiên toán tử**

|  |
| --- |
| static int Priority(string c)  {  if (c == "\*" || c == "/")  {  return 2;  }  else if (c == "+" || c == "-")  {  return 1;  }  else  {  return 0;  }  } |

**b. Các phương thức kiểm tra toán tử**

|  |
| --- |
| static bool isOperator(string c)  {  if (c == "+" || c == "-" || c == "\*" || c == "/")  {  return true;  }  else  {  return false;  }  } |

#### **c. Thuật toán chuyển đổi trung tố sang hậu tố**

|  |
| --- |
| public void InfixToPostfix()  {  string infix = txtDisplay.Text.Trim();  string[] tokens = infix.Split(' ');  Stack<string> s = new Stack<string>();  List<string> outputList = new List<string>();  double n;  foreach (string c in tokens)  {  if (double.TryParse(c.ToString(), out n)) //Nếu là số thì đưa vào outputList  {  outputList.Add(c);  }  if (c == "(")  {  s.Push(c); //Nếu gặp "(" đưa vào stack  }  if (c == ")")  {  while (s.Count != 0 && s.Peek() != "(")  {  outputList.Add(s.Pop());  }  s.Pop();  }  if (isOperator(c) == true)  {  while (s.Count != 0 && Priority(s.Peek()) >= Priority(c))  {  outputList.Add(s.Pop());  }  s.Push(c);  }  }  while (s.Count != 0)  {  outputList.Add(s.Pop());  }  postfix = outputList.ToArray();  for (int i = 0; i < outputList.Count; i++)  {  txtHT.Text += outputList[i];  }  } |

#### **d. Tính kết quả từ thuật toán hậu tố**

|  |
| --- |
| public void Result()  {  Stack polishNotationStack = new Stack();  double n;  foreach (string c in postfix)  {  if (double.TryParse(c, out n)) //Kiểm tra có phải là toán hạng không  {  polishNotationStack.Push(n);//Đưa vào stack  }  if (c == "+")  {  double x = (double)polishNotationStack.Pop();  double y = (double)polishNotationStack.Pop();  result = x + y;  polishNotationStack.Push(result); //Đưa kết quả vào stack  }  if (c == "-")  {  double x = (double)polishNotationStack.Pop();  double y = (double)polishNotationStack.Pop();  result = y - x;  polishNotationStack.Push(result);  }  if (c == "\*")  {  double x = (double)polishNotationStack.Pop();  double y = (double)polishNotationStack.Pop();  result = x \* y;  polishNotationStack.Push(result);  }  if (c == "/")  {  double x = (double)polishNotationStack.Pop();  double y = (double)polishNotationStack.Pop();  result = y / x;  polishNotationStack.Push(result);  }  }  } |

#### **e. Tính giai thừa**

|  |
| --- |
| private void GiaiThua()  {  try  {  string temp = "";  int l = txtDisplay.Text.ToString().IndexOf("!");  if (txtDisplay.TextLength > l + 1)  {  MessageBox.Show("Lỗi cú pháp", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  else  {  temp = txtDisplay.Text.Replace("!", "");  int k = int.Parse(temp);  kq = k;  for (int i = k - 1; i > 0; i--)  kq = kq \* i;  txtResult.Text = kq.ToString();  }  }  catch (Exception)  {  MessageBox.Show("Lỗi cú pháp", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

#### **f. Tính căn bậc 2**

|  |
| --- |
| private void Sqrt()  {  try  {  string temp = "";  int l = txtDisplay.Text.ToString().IndexOf("√");  if (l == 0)  {  temp = txtDisplay.Text.Replace("√", "");  kq = Math.Sqrt(double.Parse(temp));  txtResult.Text = kq.ToString();  }  else  {  MessageBox.Show("Lỗi cú pháp", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } catch (Exception)  {  MessageBox.Show("Lỗi cú pháp", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

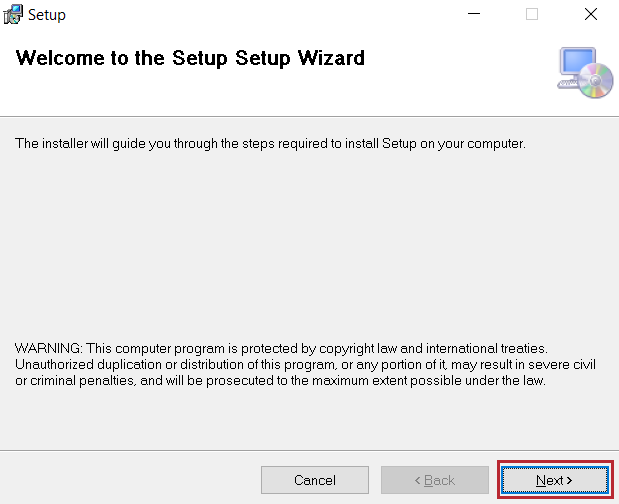
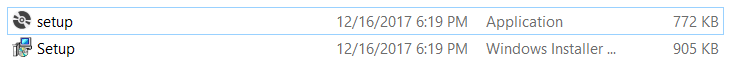
#### **g. Tính mũ hai**

|  |
| --- |
| private void Square()  {  try  {  string temp = "";  int l = txtDisplay.Text.ToString().IndexOf("^");  if (txtDisplay.TextLength > l + 2)  {  MessageBox.Show("Lỗi cú pháp", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  else  {  temp = txtDisplay.Text.Replace("^2", "");  kq = double.Parse(temp) \* double.Parse(temp);  txtResult.Text = kq.ToString();  }  } catch (Exception)  {  MessageBox.Show("Lỗi cú pháp", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

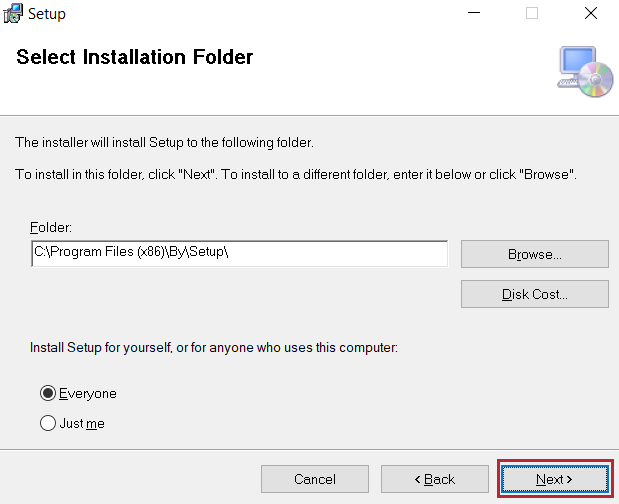
#### **h. Tính mũ ba**

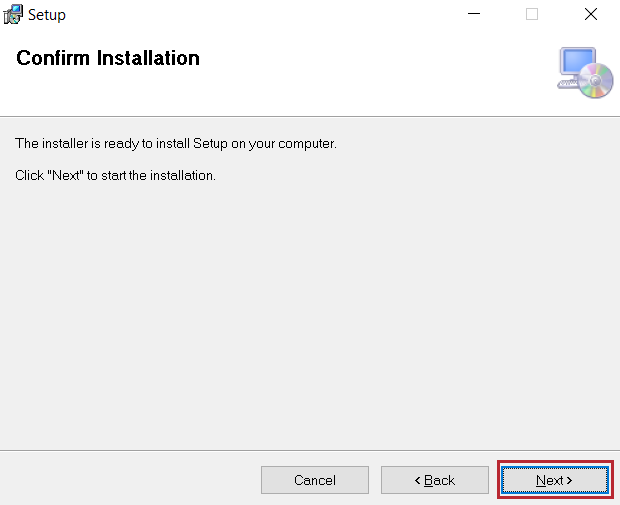
|  |
| --- |
| private void Cube()  {  try  {  string temp = "";  int l = txtDisplay.Text.ToString().IndexOf("^");  if (txtDisplay.TextLength > l + 2)  {  MessageBox.Show("Lỗi cú pháp", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  else  {  temp = txtDisplay.Text.Replace("^3", "");  kq = double.Parse(temp) \* double.Parse(temp) \* double.Parse(temp);  txtResult.Text = kq.ToString();  }  } catch (Exception)  {  MessageBox.Show("Lỗi cú pháp", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

### **4.1.2 Cài đặt phần mềm**

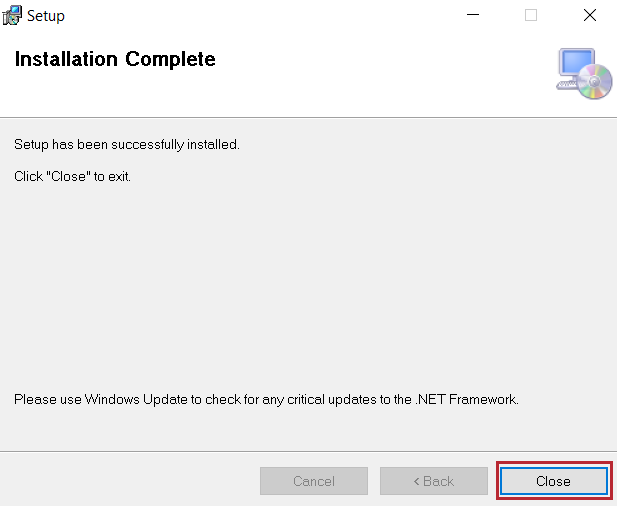
 Mở file setup.exe.  
 Chọn Next.

Bấm vào Browse để chọn nơi muốn cài đặt phần mềm. Sau đó chọn Next.





Sau đó chờ cho phần mềm được cài đặt xong. Chọn Close.



## *4.2 Kiểm thử*

Nhóm đã kiểm thử nhiểu trường hợp và sau đây là các trường hợp tiêu biểu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Biểu thức trung tố | Hậu tố | Đáp án |
| 1 | ( 1 + 2 ) \* 3 - 4 / 5 | 1 2 + 3 \* 4 5 / - | 8.2 |
| 2 | 2 \* 3 + 4 \* ( ( 5 - 6 ) + 4 ) / 3 | 2 3 \* 4 5 6 – 4 + \* 3 / + | 10 |
| 3 | 5.5 \* ( 7 + 3 ) / ( -5 ) | 5.5 7 3 + \* -5 / | -11 |
| 4 | 5.5 / 2 + 3 \* ( 4 / 8 – 2 ) | 5.5 2 / 3 4 8 / 2 - \* + | -1.75 |
| 5 | 2 \* ( 3 / 2 – 1 ) + 5 \* ( 4 - 1 ) \* 7 + 1 \* ( 9 - 4 ) | 2 3 2 / 1 - \* 5 4 1 - \* 7 \* + 1 9 4 - \* + | 111 |

# **5. Phân công công việc**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Công việc | Đoàn Huỳnh Thiện | Nguyễn Quốc Anh |
| 1 | Xử lý các phím số | x |  |
| 2 | Xử lý các phím + - \* / | x |  |
| 3 | Xử lý phím | x |  |
| 4 | Xử lý phím giai thừa | x |  |
| 5 | Xử lý phím căn | x |  |
| 6 | Đổi biểu thức trung tố sang hậu tố | x |  |
| 7 | Xử lý phím mũ 2 |  | x |
| 8 | Xử lý phím mũ 3 |  | x |
| 9 | Tính kết quả từ biểu thức hậu tố |  | x |
| 10 | Xử lý phím Clear |  | x |

Đánh giá đóng góp

|  |  |
| --- | --- |
| Tên sinh viên | Đánh giá chung phần trăm đóng góp |
| Đoàn Huỳnh Thiện | 55% |
| Nguyễn Quốc Anh | 45% |

# **6. Kết luận**

Nhóm đã thực hiện đúng yêu cầu đặt ra là sử dụng thuật toán Ký pháp Balan để thực hiện các phép tính đơn giản, tuy nhiên có một vài khuyết điểm, đôi khi sẽ có những lỗi ngoài mong muốn nên nhóm đánh giá mức độ hoàn thành mục tiệu là 95%

Ưu điểm: Giao diện quen thuộc, dễ sử dụng, cho biết cả biểu thức hậu tố lẫn kết quả.

Khuyết điểm: Khi xóa chỉ được xóa từ vị trí cuối cùng trở về trước, không được xóa tại vị trí đặt con trỏ; các phép tính giai thừa, căn, mũ 2, mũ 3 chỉ thực hiện được đơn lẻ.

Khó khăn gặp phải là khi tìm hiểu cách xây dựng thuật toán để thực hiện chuyển đổi biểu thức trung tố sang hậu tố và thực hiện tính kết quá từ biểu thức trung tố.

Khắc phục khó khăn bằng cách tìm hiểu các tài liệu có liên quan đến 2 vấn đề nêu trên ở trên mạng.

Qua đồ án này, nhóm em được ôn lại kiến thức cấu trúc dữ liệu và giải thuật đồng thời hiểu thêm về thuật toán “Ký pháp Balan”.

# **Tài liệu tham khảo**

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

https://yinyangit.wordpress.com/2011/01/27/algorithm-tinh-gia-tri-bieu-thuc-tien-to-va-hau-to/