



โครงการการจัดทำโปรแกรมย่อยครั้งที่ 3
“โปรแกรมแปลงสมการในรูปแบบ Infix ให้เป็นรูปแบบ Postfix”

นายณัฐสิทธิ์ แก้วทอง
รหัสนิสิต 59160651 กลุ่มที่ 3

เสนอ
ผู้ช่วยศาสตราจารย์นวลศรี เด่นวัฒนา
อาจารย์พจน์สพร แซ่ลิ้ม

โครงการนี้เป็นส่วนหนึ่งของรายวิชา
88823359 โครงสร้างข้อมูลและขั้นตอนวิธีสำหรับวิศวกรรมซอฟต์แวร์
สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
ภาคการศึกษาที่ 1 ปีการศึกษา 2560

Pseudocode

อัลกอริทึม : โปรแกรมสำหรับแปลงสมการในรูปแบบ Infix ให้เป็นรูปแบบ Postfix

ข้อมูลรับเข้า : รับค่าสมการรูปแบบ infix

ผลลัพธ์ : แสดงสมการเป็นรูปแบบ Postfix

1. เริ่มต้น

1.1 กำหนดค่าตัวแปร SIZE เท่ากับ 100

2. ประกาศ Class ชื่อ StackPointer

2.1 ประกาศ Class ชื่อ Node แบบ private

2.1.1 ประกาศตัวแปร data ประเภท char

2.1.2 ประกาศตัวแปร Node *link

2.1.3 Method Node(int data)

2.1.3.1 กำหนด this ชื่อที่ data เท่ากับ data

2.1.3.2 กำหนด link เท่ากับ NULL

2.2 สิ้นสุดคลาส Node

2.3 ประกาศตัวแปร Node *top

2.4 ประกาศตัวแปร count ประเภท int

2.5 Method StackPointer()

2.5.1 กำหนด top เท่ากับ NULL

2.5.2 กำหนด count เท่ากับ 0

2.6 Method ~StackPointer()

2.6.1 ทำซ้ำ ตั้งแต่ *temp เท่ากับ top จนกระทั่ง *temp เท่ากับ NULL โดย top เท่ากับ top ชื่อที่ link

2.6.1.1 ลบตัวแปร temp

2.6.1.2 กำหนด temp เท่ากับ NULL

2.6.2 กำหนด top เท่ากับ NULL

2.7 Method push(char value) ประเภท void

2.7.1 ประกาศตัวแปร Node *n เท่ากับ new Node(value)

2.7.2 ถ้าเท่ากับ method isEmpty()

2.7.2.1 กำหนด top เท่ากับ n

2.7.3 กำหนด n ชื่อที่ link เท่ากับ top

2.7.4 กำหนด top เท่ากับ n

2.7.5 ตัวนับ count เพิ่มค่าขึ้น 1

2.8 Method pop() ประเภท char

2.8.1 ประกาศตัวแปร Node *temp

2.8.2 ประกาศตัวแปร x ประเภท char

2.8.3 กำหนด x เท่ากับ top ซึ่งที่ data

2.8.4 ถ้าเท่ากับ method isEmpty()

2.8.4.1 กำหนด Top เท่ากับ NULL

2.8.5 กำหนด temp เท่ากับ top

2.8.6 กำหนด top เท่ากับ top ซึ่งที่ link

2.8.7 ตัวนับ count ลดค่าลง 1

2.8.8 ลบตัวแปร temp

2.8.9 คืนค่า x

2.9 Method size() ประเภท int

2.8.1 คืนค่า count

2.10 Method isEmpty() ประเภท boolean

2.10.1 ถ้า count เท่ากับ 0

2.10.1.1 คืนค่า true

2.10.2 คืนค่า false

2.11 Method isOperator(char a) ประเภท boolean

2.11.1 ถ้าข้อมูล a เท่ากับ '+' หรือ a เท่ากับ '-' หรือ a เท่ากับ '*' หรือ a เท่ากับ '/' หรือ a เท่ากับ '^' หรือ a เท่ากับ '(' หรือ a เท่ากับ ')' หรือ a เท่ากับ ' '

2.11.1.1 คืนค่า true

2.11.2 คืนค่า false

2.12 Method priority(char a) ประเภท int

2.12.1 ประกาศตัวแปร temp ประเภท int มีค่าเท่ากับ 0

2.12.2 ถ้าข้อมูล a เท่ากับ '(' หรือ a เท่ากับ ')' หรือ a เท่ากับ ' '

2.12.2.1 กำหนด temp เท่ากับ 4

2.12.3 ถ้าข้อมูล a เท่ากับ '^'

2.12.3.1 กำหนด temp เท่ากับ 3

2.12.4 ถ้าข้อมูล a เท่ากับ '*' หรือ a เท่ากับ '/'

2.12.4.1 กำหนด temp เท่ากับ 2

2.12.5 ถ้าข้อมูล a เท่ากับ '+' หรือ a เท่ากับ '-'

2.12.5.1 กำหนด temp เท่ากับ 1

2.12.6 คืนค่า temp

2.13 Method precedence(char a, char b) ประเภท boolean

2.13.1 ประกาศตัวแปร weight1 ประเภท int มีค่าเท่ากับ method priority(a)

2.13.2 ประกาศตัวแปร weight2 ประเภท int มีค่าเท่ากับ method priority(b)

2.13.3 ถ้า weight1 มีค่ามากกว่าหรือเท่ากับ weight2

2.13.3.1 คืนค่า true

2.13.4 คืนค่า false

2.14 Method infixToPostfix() ประเภท string

2.14.1 ประกาศตัวแปร infix[SIZE] ประเภท char

2.14.2 ประกาศตัวแปร postfix ประเภท string

2.14.3 รับค่าสมการ infix

2.14.4 ทำซ้ำ ตั้งแต่ i เท่ากับ 0 จนกระทั่ง i น้อยกว่า ที่ infix[i] ไม่เท่ากับ 0 โดย i เพิ่มขึ้นทีละ 1

2.14.4.1 ถ้าไม่เท่ากับ method isOperator(infix[i])

- postfix + ตัวแปร infix[i]

2.14.4.2 ถ้าตัวแปร infix[i] เท่ากับ '('

- เรียกใช้ method push(infix[i])

2.14.4.3 ถ้าตัวแปร infix[i] เท่ากับ ')'

- ทำซ้ำขณะที่ ไม่เท่ากับ isEmpty() และ top ซี่ที่ data ไม่เท่ากับ '('

- postfix + method pop()

- เรียกใช้ method pop()

2.14.4.4 ถ้าเท่ากับ method isOperator(infix[i])

- ถ้าเท่ากับ method isEmpty()
- เรียกใช้ method push(infix[i])

- ถ้าเท่ากับ method isOperator(infix[i])
 - ทำซ้ำขณะที่ ไม่เท่ากับ isEmpty() และ top ซี่ที่ data ไม่เท่ากับ '('
 - postfix + method pop()
- เรียกใช้ method push(infix[i])

2.14.5 ทำซ้ำ ตั้งแต่ i เท่ากับ 0 จนกระทั่ง i น้อยกว่า ที่ infix[i] ไม่เท่ากับ 0 โดย i เพิ่มขึ้นทีละ 1

2.13.5.1 ถ้าเท่ากับ method isEmpty()

- หยุด (break)

2.13.5.2 postfix + method pop()

2.14.6 คำนวณค่า postfix

End Class StackPointer

4. Main program

4.1 กำหนด object คลาส StackPointer ชื่อ s

4.2 ประกาศตัวแปร num ประเภท int

4.3 แสดงเมนูเพื่อรับค่าทางแป้นพิมพ์

4.3.1 ถ้า num เท่ากับ 1

4.3.1.1 เรียกใช้ method infixToPostfix()

4.3.1.2 แสดงผลการแบบ Postfix

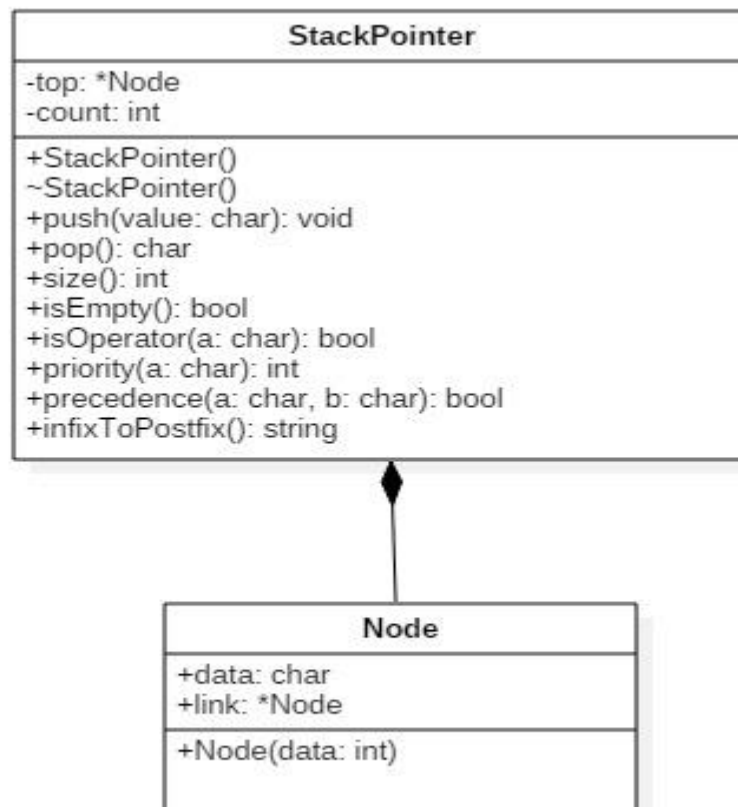
4.3.2 ถ้า num เท่ากับ 0

4.3.2.1 ออกจากโปรแกรม

END Main

5. จบโปรแกรม

Class Diagram



ตารางที่ 1 คลาส StackPointer

สมาชิก	ประเภท	รายละเอียด
top	Attribute	ตัวแปรสำหรับเก็บตำแหน่งบนสุดของสแตก
count	Attribute	ตัวแปรสำหรับเก็บตัวนับ
StackPointer()	Method	Constructor สำหรับกำหนดค่าเริ่มต้น
~StackPointer()	Method	Deconstructor สำหรับลบข้อมูลที่กำหนดขึ้นออก จากหน่วยความจำ
push(char value)	Method	เพิ่มข้อมูลโดยนำไปเก็บไว้ที่บนสุดของสแตก
pop()	Method	นำข้อมูลที่เก็บไว้บนสุดของสแตกออกไปใช้งาน
size()	Method	คืนค่าจำนวนข้อมูลที่มีอยู่ในสแตก
isEmpty()	Method	ตรวจสอบข้อมูลในสแตก โดยถ้าไม่มีข้อมูลจะคืนค่า true หากมีข้อมูล จะคืนค่า false
isOperator(char a)	Method	ตรวจสอบว่าข้อมูลเป็นสัญลักษณ์หรือไม่ ถ้าเป็นจะคืน ค่า true ถ้าไม่เป็น จะคืนค่า false
priority(char a)	Method	กำหนดค่าความสำคัญของสัญลักษณ์ แล้วคืนค่า
precedence(char a, char b)	Method	เปรียบเทียบค่าของสัญลักษณ์ แล้วคืนค่า
infixToPostfix()	Method	รับค่าสมการแบบ infix แล้วแปลงสมการเป็นแบบ postfix

ตารางที่ 2 คลาส Node

สมาชิก	ประเภท	รายละเอียด
data	Attribute	ตัวแปรที่เป็นข้อมูลที่อยู่ใน node
*link	Attribute	เป็นตัวเชื่อม node โดยการเก็บข้อมูลที่อยู่ของ node ถัดไป
Node(int data)	Method	Constructor สำหรับกำหนดค่าเริ่มต้น

เอกสารผลลัพธ์ของโปรแกรม

1. เลือกรายการที่ 1

```
===== Menu =====  
1. Infix expression  
0. Exit  
=====  
>> 1
```

ภาพที่ 1 เลือกรายการ Infix expression

2. ใส่สมการแบบ infix

```
===== Menu =====  
1. Infix expression  
0. Exit  
=====  
>> 1  
Enter an arithmetic expression in Infix : 4^2*3-3+8/4/(1+1)
```

ภาพที่ 2 รับค่าสมการแบบ infix

3. แสดงผลลัพธ์สมการแบบ postfix

```
===== Menu =====  
1. Infix expression  
0. Exit  
=====  
>> 1  
Enter an arithmetic expression in Infix : 4^2*3-3+8/4/(1+1)  
Arithmetic expression in Postfix : 42^3*3-84/11+/  

```

ภาพที่ 3 แสดงผลลัพธ์ของสมการแบบ postfix

สรุปผลการวิเคราะห์ผลลัพธ์ของโปรแกรม

โปรแกรมแปลงสมการในรูปแบบ Infix ให้เป็นรูปแบบ Postfix จัดทำโดยใช้โครงสร้างข้อมูลแบบ Stack เพื่อใช้ในการแปลงสมการที่เป็น infix ให้เป็นแบบ postfix ซึ่ง Stack เป็นโครงสร้างข้อมูลแบบลิเนียร์ลิสต์ ที่มีคุณสมบัติที่ว่าการเพิ่มหรือลบข้อมูลในสแตกจะกระทำที่ปลายข้างเดียวกัน (Top of stack) คือ ข้อมูลที่ใส่หลังสุดจะถูกนำออกมา จากสแตกเป็นลำดับแรกสุด (LIFO : Last in first out)

นอกจากนี้การจัดทำโปรแกรมนี้นี้ขึ้นก็เพื่อศึกษาโครงสร้างข้อมูลแบบสแตก การทำงาน การดำเนินการบนพื้นฐานของ สแตก การแทนที่ของข้อมูลแบบสแตก และการประยุกต์ใช้สแตก ข้าพเจ้าได้นำความรู้ที่เรื่องของโครงสร้างข้อมูลและขั้นตอนวิธีสำหรับวิศวกรรมซอฟต์แวร์ มาประยุกต์ใช้ให้เกิดประโยชน์

ในการจัดทำโปรแกรมจัดการทะเบียนนิติได้มีการเชื่อมโยงกับเรื่อง Big -O Notation ดังนี้

1. Method infixToPostfix() มีการใช้ Big -O แบบ $O(n^2) + O(n)$ ในการอ่านข้อมูล (Operand) และข้อมูล (Operator) ตัวดำเนินการจากซ้ายไปขวา