



CS 1083

Module 10 Assignment

By Ngoc Phuong Anh Nguyen - 3712361

August 1st 2021

Part A – A Stack Class:

Source codes:

Stack.java:

```
/**
 * This class represents a stack of Node objects
 * @author Ngoc Phuong Anh Nguyen - 3712361
 */
public class Stack extends LinkedList
{
    public Stack()
    {
        super();
    }

    public void push(String temp)
    {
        insertAtTail(temp);
    }

    public String pop()
    {
        String temp;
        if(!isEmpty())
        {
            temp = getTail().getName();
        }
    }
}
```

```
        removeAtTail();

        return temp;
    }
    return null;
}

public String peek()
{
    return getTail().getName();
}
}
```

StackTest.java:

```
/**
 * @author Ngoc Phuong Anh Nguyen - 3712361
 */
public class StackTest
{
    public static void main(String[] args)
    {
        Stack stack = new Stack();

        stack.push("Kelvin");
        stack.push("Amy");
    }
}
```

```
stack.push("Lemon");

System.out.println("List of the contents of the stack:");
stack.displayInReverse();

System.out.println("Size of the stack: "
    + stack.getSize());

stack.pop();
System.out.println("\nList of the contents of the stack after using pop():");
stack.displayInReverse();

System.out.println("Size of the stack: "
    + stack.getSize());

    }
}
```

Output:

```
List of the contents of the stack:
*** Start of list ***
Lemon
Amy
Kelvin
*** End of list ***
```

Size of the stack: 3

List of the contents of the stack after using pop():

*** Start of list ***

Amy

Kelvin

*** End of list ***

Size of the stack: 2

Part B – Prefix to Infix:

Source code:

PrefixToInfix.java:

```
/**
 * @author Ngoc Phuong Anh Nguyen - 3712361
 */

import static java.lang.Character.isDigit;

public class PrefixToInt
{
    public static String convert(String prefix)
    {
        Stack stack = new Stack();

        for(int i = prefix.length() - 1; i >= 0; i--)
        {
            if(isDigit(prefix.charAt(i)))
            {
                String character = "" + prefix.charAt(i);
                stack.push(character);
            }
            else if(prefix.charAt(i) == '+'
                    || prefix.charAt(i) == '-'
                    || prefix.charAt(i) == '*')
```

```

        || prefix.charAt(i) == '/')
    {
        String operand1 = stack.pop();
        String operand2 = stack.pop();

        String input = "(" + operand1 + prefix.charAt(i) + operand2 + ")";
        stack.push(input);
    }
}

Node current = stack.getHead();
String output = "";
while (current != null)
{
    output = current.getName();
    current = current.getNext();
}

return output;
}

public static void main(String[] args)
{
    String[] testExpression = {"+47", "+*+23-85", "*-9/41-/763",
                               "+-*1234", "+3*-/59+821"};

    for(int i = 0; i < testExpression.length; i++)
    {

```

```

        System.out.println("\nPrefix: " + testExpression[i]
            + "\nInfix: " + convert(testExpression[i]));
    }
}

```

Output:

Prefix: +47

Infix: (4+7)

Prefix: *+23-85

Infix: ((2+3)*(8-5))

Prefix: *-9/41-/763

Infix: ((9-(4/1))*(7/6)-3))

Prefix: +-*1234

Infix: (((1*2)-3)+4)

Prefix: +3*-/59+821

Infix: (3+(((5/9)-(8+2))*1))