COL216

Explanation of the Assignment-2

By: Ankit Shubham 2014CS10158

The assignment is written in Java(Project “col216\_2\_1”). The project contains four classes:

1. Node.java
2. NodeStack.java
3. VariableArray.java
4. Press.java

Node.java

It creates nodes for the linked list. I shall implement stack data structure using linked list. Node.java contains functions:

setElement(): sets element of Object type

setNext():points to next node(passed as parameter)

getElement():returns the element stored

getNext():returns the node to which the currently node is pointing to

NodeStack.java

This is the stack implemented using Node. It is instantiated with a node called “top” which denotes the top of the stack. It contains the following functions:

getSize(): returns the size of the stack

isEmpty(): returns true if the stack is empty; else false

top():returns the top element of the stack

pop(): pops the stack

push(): pushes element(type Object) into the stack

VariableArray.java

It stores the array(type Object) of the 10 variables viz. v0,v1,…,v9. There is an indicator “current” which stores the index of the array to which the black pointer currently is pointing.

enterValue(a,b): stores a into the bth index of the array.

**Press.java**

This is the main file. It is instantiated with three stacks, prim\_stack(p), sec\_stack(s) and tert\_stack(t) along with a VariableArray array. It has the following functions:

* int priority(char c): takes in the operator and return its priority.
* manageDigits(): multi-digit cannot be entered by user. Instead they are stored as single digits stacked over in an order. For example 354 is stored as 3 above 5 above 4. To get back the number from the aggregate of sinle digits, this function is used. The detailed functioning is commented in the file itself where this function is written.
* manageUnary(): it manages the unary operator. For example, if 3,\*,-,5 is present in the prim\_stack, then it puts 3,\*,-5 in the sec\_stack. The detailed functioning is commented in the file itself where this function is written.
* Standardize():It calls manageDigits() and then manageUnary() to convert the raw input from user into readable form into sec\_stack. Input from user enters into prim\_stack.
* conv\_postfix(): the readable form is in infix notation which I converted using this function into postfix form so that expressions could be evaluated easily. The detailed functioning is commented in the file itself where this function is written. The postfix expression is contained in the tert\_stack.
* calculate(): calculates the value of the postfix expression contained in the tert\_stack. The detailed functioning is commented in the file itself where this function is written.
* evaluateExpression(operand1,operand2,operator): returns (operand1.operator.operand2)
* nullRecallError()
* overflowError()
* segmentDisplay(i): displays i on 8-segment display
* pressBlue(a): mimics the blue button box where the parameter is any digit or any of the four operators or “assign” or “recall”.
* pressBlack(a):mimics the black buttons. Two values ‘l’ and ‘r’ are allowed where ‘l’ moves the pointer to the index just lower than the current index and ‘r’ moves the pointer to the index just higher than the current index.
* showVariableChart(): prints the variable array, the current pointer and the entered expression.