MANUEL, Jan Louie S.

🡨 **INFIX** 🡪

**Operators are written in-between their operands.**

public class low {

public static void main(String[] args) {

int zxc = 3;

int cxz = 4;

int xzc = 5;

int val = (zxc + cxz) \* xzc;

System.out.println("value = " + val);

}

}

🡨**POSTIFX**🡪

**the notational system ,that arguments come first, then the operation or function.**

import java.util.Stack;

public class lowPostfix {

public static int valuation(String expression) {

Stack<Integer> elepant = new Stack<>();

for (char giraffe : expression.toCharArray()) {

if (Character.isDigit(giraffe)) {

elepant.push(Character.getNumericValue(giraffe));

} else {

int lion = elepant.pop();

int zebra = elepant.pop();

int resulta = performOperation(giraffe, zebra, lion);

elepant.push(resulta);

}

}

return elepant.pop();

}

private static int performOperation(char op, int opand1, int opand2) {

switch (op) {

case '+':

return opand1 + opand2;

case '-':

return opand1 - opand2;

case '\*':

return opand1 \* opand2;

case '/':

return opand1 / opand2;

default:

throw new IllegalArgumentException("error: " + op);

}

}

public static void main(String[] args) {

String ekspres = "34+5\*";

int resulta = valuation(ekspres);

System.out.println("result: " + resulta);

}

}

🡨**PREFIX**🡪

**operators are written before their operands. Like postfix notation, it also doesn't require parentheses or priority rules.**

import java.util.Stack;

public class lowPrefix {

public static int ebalweyshon(String ekspresyon) {

Stack<Integer> aso = new Stack<>();

for (int tigre = ekspresyon.length() - 1; tigre >= 0; tigre--) {

char daga = ekspresyon.charAt(tigre);

if (Character.isDigit(daga)) {

aso.push(Character.getNumericValue(daga));

} else {

int liyon = aso.pop();

int kabayo = aso.pop();

int value = ferpormOP(daga, kabayo, liyon);

aso.push(value);

}

}

return aso.pop();

}

private static int ferpormOP(char math, int una, int sekand) {

switch (math) {

case '+':

return una + sekand;

case '-':

return una - sekand;

case '\*':

return una \* sekand;

case '/':

return una / sekand;

default:

throw new IllegalArgumentException("ur math is error: " + math);

}

}

public static void main(String[] args) {

String prefekspresyon = "\*+345";

int value = ebalweyshon(prefekspresyon);

System.out.println("value: " + value);

}

}