CMPE 150 PROJECT II

Problem Description :

In this assignment, we were asked to make a calculator which is able to store three variables, and use them. The names and the types (double, integer) of the variables are all changeable by the user. The calculator can perform four operation ; division, substraction, addition and multiplication. It is also able to perform these operation with respect to process priority in Mathematics & with respect to the data types of the variables.

Problem Solution :

I used two methods, sixty-one if statements, and thirty for loops for the solution.

In the main method, there’s a scanner object and it asks a line from the user. This line is the first variable. Then, the name and the value of it is stored in two different strings using the substring methods. The same process is made for all 3 of the variables.

If the keyword double is used while defining the variable and there’s no dot in it, the program adds a dot to it, because in the calculations, only the strings which contain a dot is turned into a double value. If the keyword “int” is used, the program contains it’s value as an Integer value.

After the name and values of the variables are stored, the program checks if they are used in the 4th line, which is the whole calculation. If it is, the name of the variables are replaced with their numeric values.

The first method was removeSpaces(). It takes an string, (in this problem, an input from user) and removes all the spaces in the string. This was needed because the input can contain any number of spaces, and it could be a huge problem while processing the input.

The second input is calculateTheInput(). In this method, the whole input is calculated. In the beginning , there’s a string named “empty”, which will used to store the expression inside parantheses. Then, in a while loop, there’s a for loop to find the parantheses on the far right and store the expression inside them to the string “empty”. Secondly, if there’s “/” or the “\*” in the “empty”, a for loop finds the numbers before and after the operator, and if they contain a dot, it stores them as a double value. Else, it stores them as a integer. Then, the process is calculated and replaced with it’s value. After the multiplication and division operators are gone, the code does the same process for addition and substraction. This process continues until there is no operator left in the string “empty”, and the parantheses are replaced with their value, which is, the string “empty”.

Until there’s no parantheses left, this process is repeated. After all parantheses are gone, the input turns into a calculation with only numeric values. ”, a for loop finds the numbers before and after the operator, and if they contain a dot, it stores them as a double value. Else, it stores them as a integer. Then, the process is calculated and replaced with it’s value. After the multiplication and division operators are gone, the code does the same process for addition and substraction. Every time a process is calculated, the numbers and the operators as a whole replaces with the calculation’s result. At the end, the calculation turns into it’s result.

Implementation :

**public** **class** ES2018400183 {

**public** **static** **void** main(String[] args) {

Scanner x = **new** Scanner(System.***in***);

String variable1=x.nextLine(); //first variable

variable1=*removeSpaces*(variable1); //removes the spaces in the input

variable1=variable1.replace(";", ""); //removes ";" characters in the input

String name1="";

String number1="";

**if**(variable1.contains("double")) {

name1=variable1.substring(variable1.indexOf('e')+1,variable1.indexOf('=')); //name of the double variable

}**else** {

name1=variable1.substring(variable1.indexOf('t')+1,variable1.indexOf('=')); //name of the integer variable

}

number1=(variable1.substring(variable1.indexOf('=')+1)); //value of the variable

**if**(!(number1.contains("."))) { //adds a dot to the double variables if there isn't any

number1=number1+".";

}

String variable2=x.nextLine(); //second variable

variable2=*removeSpaces*(variable2);

variable2=variable2.replace(";", "");

String name2="";

String number2="";

**if**(variable2.contains("double")) {

name2=variable2.substring(variable2.indexOf('e')+1,variable2.indexOf('='));

}**else** {

name2=variable2.substring(variable2.indexOf('t')+1,variable2.indexOf('='));

}

number2=(variable2.substring(variable2.indexOf('=')+1));

**if**(!(number2.contains("."))) {

number2=number2+".";

}

String variable3=x.nextLine(); //third variable

variable3=*removeSpaces*(variable3);

variable3=variable3.replace(";", "");

String name3="";

String number3="";

**if**(variable3.contains("double")) {

name3=variable3.substring(variable3.indexOf('e')+1,variable3.indexOf('='));

}**else** {

name3=variable3.substring(variable3.indexOf('t')+1,variable3.indexOf('='));

}

number3=(variable3.substring(variable3.indexOf('=')+1));

**if**(!(number3.contains("."))) {

number3=number3+".";

}

String cal = x.nextLine(); //the whole calculation input

**if**(cal.contains(name1)) { //replaces variables with their values

cal=cal.replace(name1, number1);

}

**if**(cal.contains(name2)) {

cal=cal.replace(name2, number2);

}

**if**(cal.contains(name3)) {

cal=cal.replace(name3, number3);

}

*calculateTheInput*(*removeSpaces*(cal)); //calls the method which calculates the input

}

**public** **static** String removeSpaces(String cal) { //this method is used to remove spaces in the calculation

String upt = "";

upt = cal.replace(" ", "");

**return** (upt);

} // method

**public** **static** **void** calculateTheInput(String cal) { //this method is used to calculate the input

String empty="";

**int** h=0;

**while**(cal.contains("(")) { //if there's parantheses in the input, it detects their place

**for**(**int** k=cal.lastIndexOf("("); k<cal.length(); k++) { //finds the first ")" after the last "("

**if**(cal.charAt(k) != ')') {

**continue**;

}**else**{

h=k;

**break**;

}}

empty=cal.substring(cal.lastIndexOf("(")+1,h); // inside of the parantheses'

h=0;

// calculation inside the parantheses'

**double** mult=1; //double multiplication beginning value

**int** mult1=1; //integer multiplication beginning value

String number1="";

String number2="";

**for**(**int** a=0; a<empty.length(); a++) { //finds the first "/" or "\*"

**if**(empty.charAt(a)=='/' || empty.charAt(a)=='\*') {

**for**(**int** k=a; k>=0; k--) { //takes the number before the "\*" or "/" character as a string

**if**(empty.charAt(k)<=57 && empty.charAt(k)>=48 || empty.charAt(k)=='.') {

number1=empty.charAt(k)+number1;

**if**(k==0 || (empty.charAt(k-1)=='+' || empty.charAt(k-1)=='-' || empty.charAt(k-1)=='/' || empty.charAt(k-1)=='\*' )) {

k=-1;

}

}

}

**for**(**int** b=a; b<empty.length(); b++) { //takes the number after the "\*" or "/" character as a string

**if**(empty.charAt(b)<=57 && empty.charAt(b)>=48 || empty.charAt(b)=='.' ) {

number2=number2+empty.charAt(b);

**if**(b==empty.length()-1 || (empty.charAt(b+1)=='+' || empty.charAt(b+1)=='-' || empty.charAt(b+1)=='/' || empty.charAt(b+1)=='\*' )) {

b=empty.length();

}

}

}

**if**(empty.charAt(a)=='\*') { //turns the strings to doubles if they contain ".", else to integers, then multiplicates them

**if**(number1.contains(".") || number2.contains(".") ) {

mult= mult\*Double.*parseDouble*(number1)\*Double.*parseDouble*(number2);

}**else** **if**(!(number1.contains(".") && number2.contains("."))){

mult1= mult1\*Integer.*parseInt*(number1)\*Integer.*parseInt*(number2);

}

}**else** **if**(empty.charAt(a)=='/') { //turns the strings to doubles if they contain ".", else to integers, then divides them

**if**(number1.contains(".") || number2.contains(".") ) {

mult= Double.*parseDouble*(number1)/Double.*parseDouble*(number2)/mult;

}**else** **if**(!(number1.contains(".") && number2.contains("."))) {

mult1= (**int**)Integer.*parseInt*(number1)/Integer.*parseInt*(number2)/mult1;

}

}

**if**(number1.contains(".") || number2.contains(".") ) { //replaces the calculation with calculation's result

empty=empty.substring(0,empty.indexOf(empty.charAt(a))-number1.length())+mult+empty.substring(empty.indexOf(empty.charAt(a))+number2.length()+1);

}**else** {

empty=empty.substring(0,empty.indexOf(empty.charAt(a))-number1.length())+mult1+empty.substring(empty.indexOf(empty.charAt(a))+number2.length()+1);

}

a=0; //sets all the values to their initial value

number1="";

number2="";

mult=1;

mult1=1;

}

}

**double** sum=0; //double addition beginning value

**int** sum1=0; //integer addition beginning value

String number3="";

String number4="";

**for**(**int** c=0; c<empty.length(); c++) { //finds the first "-" or "+"

**if**(empty.charAt(c)=='+' || empty.charAt(c)=='-') {

**for**(**int** k=c; k>=0; k--) { //takes the number before the "+" or "-" character as a string

**if**(empty.charAt(k)<=57 && empty.charAt(k)>=48 || empty.charAt(k)=='.') {

number3=empty.charAt(k)+number3;

**if**(k==0 || (empty.charAt(k-1)=='+' || empty.charAt(k-1)=='-' || empty.charAt(k-1)=='/' || empty.charAt(k-1)=='\*' )) {

k=-1;

}

}

}

**for**(**int** b=c; b<empty.length(); b++) { //takes the number after the "+" or "-" character as a string

**if**(empty.charAt(b)<=57 && empty.charAt(b)>=48 || empty.charAt(b)=='.') {

number4=number4+empty.charAt(b);

**if**(b==empty.length()-1 || (empty.charAt(b+1)=='+' || empty.charAt(b+1)=='-' || empty.charAt(b+1)=='/' || empty.charAt(b+1)=='\*' )) {

b=empty.length();

}

}

}

**if**(empty.charAt(c)=='+') { //turns the strings to doubles if they contain ".", else to integers, then adds them

**if**(number3.contains(".") || number4.contains(".") ) {

sum= sum+Double.*parseDouble*(number3)+Double.*parseDouble*(number4);

}**else** **if**(!(number3.contains(".") && number4.contains("."))) {

sum1= sum1+Integer.*parseInt*(number3)+Integer.*parseInt*(number4);

}

}**else** **if**(empty.charAt(c)=='-') { //turns the strings to doubles if they contain ".", else to integers, then extracts them

**if**(number3.contains(".") || number4.contains(".")) {

sum= Double.*parseDouble*(number3)-Double.*parseDouble*(number4)-sum;

}**else** {

sum1= Integer.*parseInt*(number3)-Integer.*parseInt*(number4)-sum1;

}

}

**if**(number3.contains(".") || number4.contains(".")) { //replaces the calculation with calculation's result

empty=empty.substring(0,empty.indexOf(empty.charAt(c))-number3.length())+sum+empty.substring(empty.indexOf(empty.charAt(c))+number4.length()+1);

}**else** {

empty=empty.substring(0,empty.indexOf(empty.charAt(c))-number3.length())+sum1+empty.substring(empty.indexOf(empty.charAt(c))+number4.length()+1);

}

c=0; //sets all the beginning values to their initial values

number3="";

number4="";

sum=0;

sum1=0;

}

}

**int** f=0;

**for**(**int** k=cal.lastIndexOf("("); k<cal.length(); k++) { //finds the first ")" after the last "("

**if**(cal.charAt(k) != ')') {

**continue**;

}**else**{

f=k;

**break**;

}}

cal=cal.substring(0,cal.lastIndexOf("("))+empty+cal.substring(f+1);

f=0;

}

//at this point, all the parantheses have been calculated and replaced with their values in the input, rest does the whole calculations again

**double** mult=1; //double multiplication beginning value

**int** mult1=1; //integer multiplication beginning value

String number1="";

String number2="";

**for**(**int** a=0; a<cal.length(); a++) {

**if**(cal.charAt(a)=='/' || cal.charAt(a)=='\*') { //finds the first "/" or "\*"

**for**(**int** k=a; k>=0; k--) {

**if**(cal.charAt(k)<=57 && cal.charAt(k)>=48 || cal.charAt(k)=='.') {

number1=cal.charAt(k)+number1;

**if**(k==0 || (cal.charAt(k-1)=='+' || cal.charAt(k-1)=='-' || cal.charAt(k-1)=='/' || cal.charAt(k-1)=='\*' )) {

k=-1;

}

}

}

**for**(**int** b=a; b<cal.length(); b++) {

**if**(cal.charAt(b)<=57 && cal.charAt(b)>=48 || cal.charAt(b)=='.') {

number2=number2+cal.charAt(b);

**if**(b==cal.length()-1 || (cal.charAt(b+1)=='+' || cal.charAt(b+1)=='-' || cal.charAt(b+1)=='/' || cal.charAt(b+1)=='\*' )) {

b=cal.length();

}

}

}

**if**(cal.charAt(a)=='\*') { //turns the strings to doubles if they contain ".", else to integers, then multiplicates them

**if**(number1.contains(".") || number2.contains(".") ) {

mult= mult\*Double.*parseDouble*(number1)\*Double.*parseDouble*(number2);

}**else** **if**(!(number1.contains(".") && number2.contains("."))) {

mult1= mult1\*Integer.*parseInt*(number1)\*Integer.*parseInt*(number2);

}

}**else** **if**(cal.charAt(a)=='/') {

**if**(number1.contains(".") || number2.contains(".") ) { //turns the strings to doubles if they contain ".", else to integers, then divides them

mult= Double.*parseDouble*(number1)/Double.*parseDouble*(number2)/(**int**)mult;

}**else** **if**(!(number1.contains(".") && number2.contains("."))) {

mult1= Integer.*parseInt*(number1)/Integer.*parseInt*(number2)/mult1;

}

}

**if**(number1.contains(".") || number2.contains(".")) { //replaces the calculation with calculation's result

cal=cal.substring(0,cal.indexOf(cal.charAt(a))-number1.length())+mult+cal.substring(cal.indexOf(cal.charAt(a))+number2.length()+1);

}**else** {

cal=cal.substring(0,cal.indexOf(cal.charAt(a))-number1.length())+mult1+cal.substring(cal.indexOf(cal.charAt(a))+number2.length()+1);

}

a=0; //sets all the beginning values to their initial values

number1="";

number2="";

mult=1;

mult1=1;

}

}

**double** sum=0;

**int** sum1=0;

String number3="";

String number4="";

**for**(**int** c=0; c<cal.length(); c++) {

**if**(cal.charAt(c)=='+' || cal.charAt(c)=='-') { //finds the first "-" or "+"

**for**(**int** k=c; k>=0; k--) { //takes the number before the "+" or "-" character as a string

**if**(cal.charAt(k)<=57 && cal.charAt(k)>=48 || cal.charAt(k)=='.') {

number3=cal.charAt(k)+number3;

**if**(k==0 || (cal.charAt(k-1)=='+' || cal.charAt(k-1)=='-' || cal.charAt(k-1)=='/' || cal.charAt(k-1)=='\*' )) {

k=-1;

}

}

}

**for**(**int** b=c; b<cal.length(); b++) { //takes the number after the "+" or "-" character as a string

**if**(cal.charAt(b)<=57 && cal.charAt(b)>=48 || cal.charAt(b)=='.' ) {

number4=number4+cal.charAt(b);

**if**(b==cal.length()-1 || (cal.charAt(b+1)=='+' || cal.charAt(b+1)=='-' || cal.charAt(b+1)=='/' || cal.charAt(b+1)=='\*' )) {

b=cal.length();

}

}

}

**if**(cal.charAt(c)=='+') { //turns the strings to doubles if they contain ".", else to integers, then adds them

**if**(number3.contains(".") || number4.contains(".") ) {

sum= sum+Double.*parseDouble*(number3)+Double.*parseDouble*(number4);

}**else** **if**(!(number3.contains(".") && number4.contains("."))) {

sum1= sum1+Integer.*parseInt*(number3)+Integer.*parseInt*(number4);

}

}**else** **if**(cal.charAt(c)=='-') { //turns the strings to doubles if they contain ".", else to integers, then substracts them

**if**(number3.contains(".") || number4.contains(".")) {

sum= Double.*parseDouble*(number3)-Double.*parseDouble*(number4)-sum;

}**else** **if**(!(number3.contains(".") && number4.contains("."))) {

sum1= Integer.*parseInt*(number3)-Integer.*parseInt*(number4)-sum1;

}

}

**if**(number3.contains(".") || number4.contains(".")) { //replaces the calculation with calculation's result

cal=cal.substring(0,cal.indexOf(cal.charAt(c))-number3.length())+sum+cal.substring(cal.indexOf(cal.charAt(c))+number4.length()+1);

}**else** {

cal=cal.substring(0,cal.indexOf(cal.charAt(c))-number3.length())+sum1+cal.substring(cal.indexOf(cal.charAt(c))+number4.length()+1);

}

c=0; //sets all the beginning values to their initial values

number3="";

number4="";

sum=0;

sum1=0;

}

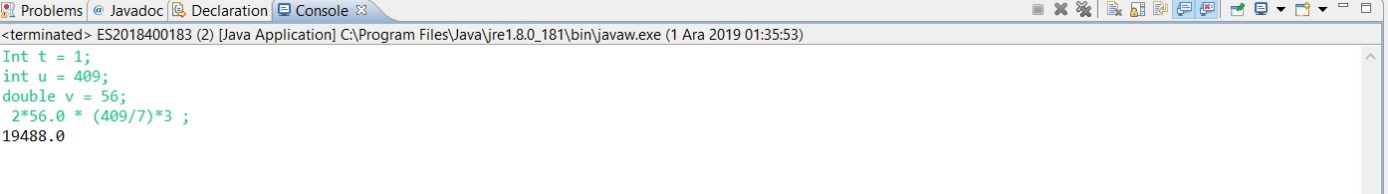
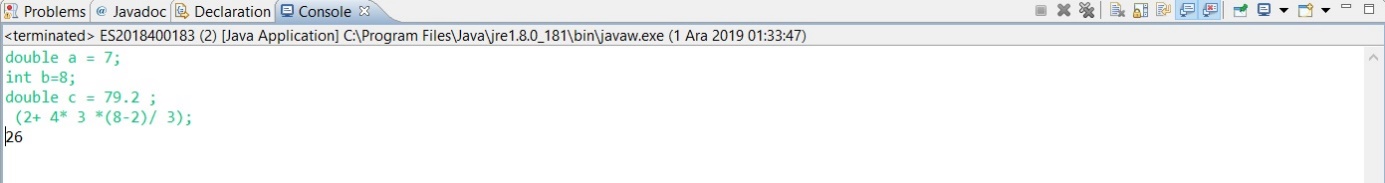
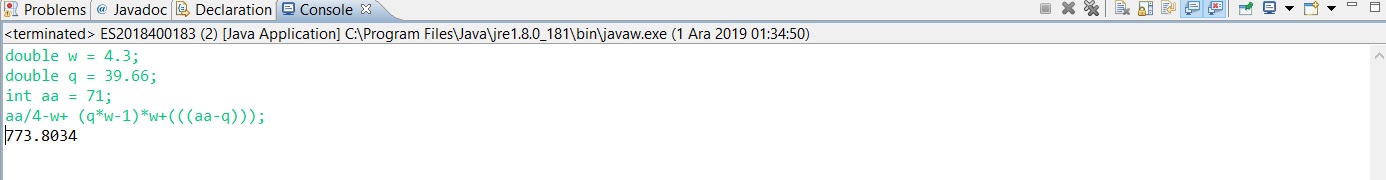
}

System.***out***.println(cal.replace(";", "")); //removes the ";" characters

}

}//end

Output of the Program :



Conclusion :

The problem is solved successfully with all the cases because it can solve complicated expressions with nested parantheses, while respects to the data types and process priority as seen in outputs.

ECENUR SEZER

2018400183