CMPE 150.06, Fall'19 Assignment 2 Nurlan Dadashov

1. **Project Description:** The purpose of this program is to take four lines of input from a user, three of which being variable declarations and the fourth one being an arithmetical calculation, and to print the evaluation as the output.
2. **Problem Solution:** The first step was to format the four lines of input in such a way that it will be easier to perform calculations. The method *newstring()* creates a new string where all numbers, parentheses, keywords, and symbols are separated by a space. Then, my program replaces all variable names with their values. After using method *newstring()* again just to make sure the format of the string is right, the method *parenthesis()* is implemented. This method finds all nested parentheses and calculates their values using *calc()* method and then replaces those parentheses with their values. After all expressions with parentheses are calculated and removed from the string, the rest is calculated by the method *calc().* This method first checks if there are * or /. Then, performs all multiplication and division operation from left to right(by checking indexes of * and /). After all * and / are calculated and replaced in the string, the method *calc()* performs addition and subtraction operation in the same manner. To calculate sum/difference /product/ratio, the program calls methods *firstnum()* and *secondnum()* which find the numbers to the left and right of the symbol respectively. By checking if these numbers are either integers or doubles using *check()* method, the program decides whether to call *int_operatins()* or *double_operations()* methods. The method check() returns true if the number is integer(as Integer.parseInt() does not throw NumberFormat Exception) and false if the number is double. As it can be seen from their names, if *firstnum()* and *secondnum()* return integers, we need to call *int_operations()* method; in all other cases *double_operations()* method is used for obvious reasons. Subsequently, using *replaceint()* and *replacedouble()* methods, the program replaces the part of string that was calculated with the value that returns from *int_operations()/ double_operations()* methods. Finally, the program prints out the final answer(without ; symbol).

3. **Implementation:**

```java
import java.util.Scanner;

public class ND2019400300 {
	/*
	 * @author Nurlan Dadashov
	 * @course CMPE 150.06
	 */
	public static void main(String[] args) {
		// taking user input
		Scanner console = new Scanner(System.in);

		String s1 = console.nextLine();
		String s2 = console.nextLine();
		String s3 = console.nextLine();
		String s4 = console.nextLine();

		console.close();
		//modifying strings using newstring method
		//@look at newstring method
		String newsentence = newstring(s4);

		s1 = newstring(s1);
		s2 = newstring(s2);
		s3 = newstring(s3);
		//taking the name of each variable
		String s1s = s1.substring(s1.indexOf(" ") + 1, s1.indexOf("=") - 1);
		String s2s = s2.substring(s2.indexOf(" ") + 1, s2.indexOf("=") - 1);
		String s3s = s3.substring(s3.indexOf(" ") + 1, s3.indexOf("=") - 1);
		//taking value of each variable, finding its name in newsentence
		//and replacing with its value
		String s1n = s1.substring(s1.indexOf("=") + 2, s1.length() - 1);
		if(check(s1n)) {
			newsentence = newsentence.replace(s1s, s1n);
		}
		else {
			newsentence = newsentence.replace(s1s, Double.toString(Double.parseDouble(s1n)));
		}

		String s2n = s2.substring(s2.indexOf("=") + 2, s2.length() - 1);
		if(check(s2n)) {
			newsentence = newsentence.replace(s2s, s2n);
		}
		else {
			newsentence = newsentence.replace(s2s, Double.toString(Double.parseDouble(s2n)));
		}

		String s3n = s3.substring(s3.indexOf("=") + 2, s3.length() - 1);
		if(check(s3n)) {
			newsentence = newsentence.replace(s3s, s3n);
		}
		else {
			newsentence = newsentence.replace(s3s, Double.toString(Double.parseDouble(s3n)));
		}
```

```
        //updating string to the correct format
        newsentence = newstring(newsentence);
        //solving insides of parenthesis
        newsentence = paranthesis(newsentence);
        //calculating the remaining equation
        newsentence = calc(newsentence);
        //printing the final value without ";"
        System.out.println(newsentence.substring(0,newsentence.length() - 1));
    }
    //a method for solving nested paranthesis
    public static String paranthesis(String newsentence) {
        while(newsentence.contains("(") && newsentence.contains(")")) {
            String s4 = newsentence;
            String s = "";

            while(s4.contains("(")) {
                s = s4.substring(s4.indexOf("("));
                s4 = s4.substring(s4.indexOf("(") + 2);

                if(s4.indexOf(")") < s4.indexOf("(")) {
                    break;
                }
            }
            s = s.substring(0 ,s.indexOf(")") + 1);
            s4 = s4.substring(0 ,s4.indexOf(")") - 1);
            //calculating inside of a parathesis
            String s5 =calc(s4);
            //replacing () with s5
            newsentence = replaceparathesis(s, s5, newsentence);
            //formatting string
            newsentence = newstring(newsentence);
        }
        return newsentence;
    }
    //replacing () with result of evaluation of its inside
    public static String replaceparathesis(String s4, String s5, String newsentence) {
        newsentence = newsentence.replace(s4, s5);
        return newsentence;
    }
    //Method used for calculations
    public static String calc(String newsentence) {
        while(newsentence.contains("+") || newsentence.contains("-") ||
newsentence.contains("*") || newsentence.contains("/")) {
            //Multiplication and division
            while(newsentence.contains("*") && ((newsentence.indexOf("*") <
newsentence.indexOf("/")) || !(newsentence.contains("/")))) {
                int result = 0;
                double resultd = 0;
                String symb = "*";
                if(check(firstnum(newsentence,symb)) && check(secondnum(newsentence,symb)))
{
                    int i = Integer.parseInt(firstnum(newsentence,symb));
                    int j = Integer.parseInt(secondnum(newsentence,symb));
                    result = int_operations(newsentence, symb, i, j);
                    newsentence = replaceint(result, newsentence, symb);
```

```
                        }
                        else {
                                double i = Double.parseDouble(firstnum(newsentence,symb));
                                double j = Double.parseDouble(secondnum(newsentence,symb));
                                resultd = double_operations(newsentence, symb, i, j);
                                newsentence = replacedouble(resultd, newsentence, symb);
                        }
                }

                while(newsentence.contains("/") && ((newsentence.indexOf("*") >
newsentence.indexOf("/")) || !(newsentence.contains("*")))) {
                        int result = 0;
                        double resultd = 0;
                        String symb = "/";
                        if(check(firstnum(newsentence,symb)) && check(secondnum(newsentence,symb)))
{
                                int i = Integer.parseInt(firstnum(newsentence,symb));
                                int j = Integer.parseInt(secondnum(newsentence,symb));
                                result = int_operations(newsentence, symb, i, j);
                                newsentence = replaceint(result, newsentence, symb);
                        }
                        else {
                                double i = Double.parseDouble(firstnum(newsentence,symb));
                                double j = Double.parseDouble(secondnum(newsentence,symb));
                                resultd = double_operations(newsentence, symb, i, j);
                                newsentence = replacedouble(resultd, newsentence, symb);
                        }
                }
                //Addition and subtraction
                while(newsentence.contains("+") && ((newsentence.indexOf("+") <
newsentence.indexOf("-")) || !(newsentence.contains("-"))) && !(newsentence.contains("*")) &&
!(newsentence.contains("/"))) {
                        int result = 0;
                        double resultd = 0;
                        String symb = "+";
                        if(check(firstnum(newsentence,symb)) && check(secondnum(newsentence,symb)))
{
                                int i = Integer.parseInt(firstnum(newsentence,symb));
                                int j = Integer.parseInt(secondnum(newsentence,symb));
                                result = int_operations(newsentence, symb, i, j);
                                newsentence = replaceint(result, newsentence, symb);
                        }
                        else {
                                double i = Double.parseDouble(firstnum(newsentence,symb));
                                double j = Double.parseDouble(secondnum(newsentence,symb));
                                resultd = double_operations(newsentence, symb, i, j);
                                newsentence = replacedouble(resultd, newsentence, symb);
                        }
                }

                while(newsentence.contains("-") && ((newsentence.indexOf("+") >
newsentence.indexOf("-")) || !(newsentence.contains("+")))&& !(newsentence.contains("*")) &&
!(newsentence.contains("/"))) {
                        int result = 0;
                        double resultd = 0;
                        String symb = "-";
```

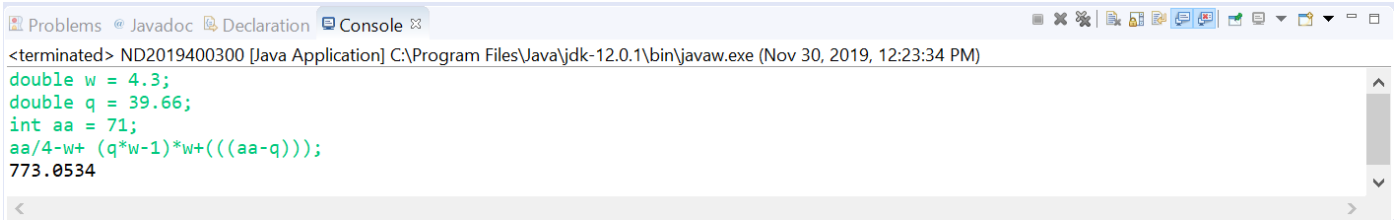
```
                        if(check(firstnum(newsentence,symb)) && check(secondnum(newsentence,symb)))
{
                              int i = Integer.parseInt(firstnum(newsentence,symb));
                              int j = Integer.parseInt(secondnum(newsentence,symb));
                              result = int_operations(newsentence, symb, i, j);
                              newsentence = replaceint(result, newsentence, symb);
                        }
                        else {
                              double i = Double.parseDouble(firstnum(newsentence,symb));
                              double j = Double.parseDouble(secondnum(newsentence,symb));
                              resultd = double_operations(newsentence, symb, i, j);
                              newsentence = replacedouble(resultd, newsentence, symb);
                        }
                  }
            }
            return newsentence;
      }

      public static String replaceint(int result, String newsentence, String symb) {
            String s = newsentence.substring(newsentence.indexOf(symb) -
(firstnum(newsentence,symb).length() + 1), newsentence.indexOf(symb) +
(secondnum(newsentence,symb).length() + 2));
            newsentence = newsentence.replace(s, Integer.toString(result));
            return newsentence;
      }

      public static String replacedouble(double resultd, String newsentence, String symb) {
            String s = newsentence.substring(newsentence.indexOf(symb) -
(firstnum(newsentence,symb).length() + 1), newsentence.indexOf(symb) +
(secondnum(newsentence,symb).length() + 2));
            newsentence = newsentence.replace(s, Double.toString(resultd));
            return newsentence;
      }
      //Finds the number which is left to symbol
      public static String firstnum(String newsentence, String symb) {
            int index = newsentence.indexOf(symb);
            String tmp = newsentence.substring(0, index - 1);
            String result = tmp.substring(tmp.lastIndexOf(" ") + 1);
            return result;
      }
      //Finds the number which is right to symbol
      public static String secondnum(String newsentence, String symb) {
            int index = newsentence.indexOf(symb);
            String tmp = newsentence.substring(index + 2);
            String result = "";
            if(tmp.contains(" ")) {
                  result = tmp.substring(0, tmp.indexOf(" "));
            }
            else {
                  if(tmp.contains(";")) {
                        result = tmp.substring(0, tmp.indexOf(";"));
                  }
                  else {
                        result = tmp.substring(0, tmp.length());
                  }
```

```
        }

        return result;
    }
    //checks whether the number is int or double
    public static boolean check(String str) {
        try
      {
         Integer.parseInt(str);
         return true;
      }
      catch (NumberFormatException e)
      {
        return false;
      }

    }
    //+ - * / operations with integers
    public static int int_operations(String newsentence, String symb, int Inum1, int Inum2) {
        int resultI = 0;
        switch (symb) {
        case "+":
              resultI = Inum1 + Inum2;
              break;
        case "-":
              resultI = Inum1 - Inum2;
              break;
        case "*":
              resultI = Inum1 * Inum2;
              break;
        case "/":
              resultI = Inum1 / Inum2;
              break;
        }
        return resultI;
    }
    //+ - * / operations with doubles
    public static double double_operations(String newsentence, String symb, double Dnum1, double
Dnum2) {

        double resultD = 0;
        switch (symb) {
        case "+":
              resultD = Dnum1 + Dnum2;
              break;
        case "-":
              resultD = Dnum1 - Dnum2;
              break;
        case "*":
              resultD = Dnum1 * Dnum2;
              break;
        case "/":
              resultD = Dnum1 / Dnum2;
              break;
        }
        return resultD;
```

```
    }
    //formatting string(removing spaces so that there is only one space between numbers and
symbols)
    public static String newstring(String s3) {
        String newsentence = "";

        for(int i = 0; i < s3.length(); i++) {
            char c = s3.charAt(i);
            if(c == ' ' && s3.charAt(i + 1) == ' ' ) {
                continue;
            }
            else if(c == ' ' && s3.charAt(i + 1) != ' ') {
                newsentence += ' ';
            }
            else if(c == '+' || c == '-' || c == '/' || c == '*' || c == '(' || c == ')' || c
== '=') {
                if (i >= 1) {
                    if(s3.charAt(i - 1) != ' ') {
                        newsentence += " ";
                    }
                }

                switch (c) {
                case '+':
                    newsentence += "+";
                    break;
                case '-':
                    newsentence += "-";
                    break;
                case '*':
                    newsentence += "*";
                    break;
                case '/':
                    newsentence += "/";
                    break;
                case '(':
                    newsentence += "(";
                    break;
                case ')':
                    newsentence += ")";
                    break;
                case '=':
                    newsentence += "=";
                    break;
                }

                if(s3.charAt(i + 1) != ' ') {
                    newsentence += " ";
                }
            }
            else {
                newsentence += c;
            }
        }
        if(newsentence.indexOf(" ") == 0) {
            newsentence = newsentence.substring(1);
```

```
        }
        return newsentence;
    }
}
```

4. Output of the program:



5. Conclusion:

All in all, the assignment was solved, and the code was written to the best of my ability. I found the assignment quite challenging at first. However, after understanding the logic behind it, I was able to quickly write the code. It was an interesting project, and I look forward to the next challenge. I hope that the solution has been satisfactory, and the reading interesting.

Thank you for reading.