01.12.2019

FALL 2019 – Assignment 2

CMPE 150.06

HATİCE ERK – 2018400090

PROJECT REPORT

PROBLEM DESCRIPTION

Writing a java program which takes four lines input from user and evaluate arithmetic expressions. In first three lines input will be like ' int a =2 ;' and the last line will contain just some calculations. We should perform summation, subtraction, multiplication and division with operator precedence.Solving statements with nested parenthesis is also expected.

PROBLEM SOLUTION

In solving problem, I used 7 methods, two of them to calculation, one of them to evaluate first three lines, four of them to help finding numbers and limits. Firstly, I evaluated first three lines with using 'lines' method.This method takes a line and returns a string which contains the name and value of the variable.Then if the last line contains variables, I replaced variables with their values. Then, if the last line contains parenthesis, I went inside them. I took each operation into two methods in turn.

In my first method ' calculation' , I did multiplication and division and then in my second method , ı did summation and subtraction to save precedence. My other methods 'sayıAlt' and 'sayıUst' took a string and an integer as a parameter and returned a string which contains a number which ı used in calculation. My last methods 'sınırAlt' and 'sınırUst' again took a string and an integer as a parameter and returned an integer which ı used to determine where my operation begins and ends.

Finally ı printed the result of calculation.

IMPLEMENTATION

```
package haticeCmpe;

import java.util.*;

public class HE2018400090 {
```

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

                Scanner input = new Scanner(System.in);

                String firstLine = input.nextLine();

                //I took the first line

                Scanner newFl =new Scanner(lines(firstLine));

                //to get name and value more easily, ı changed to scanner

                String name1 = newFl.next();

                //name1 is the name of my first variable

                //because ı didn't mention before,program don't know that is this an integer or a double.So ı used to containing '.'

                int ilines1 = -1;

                double dlines1 = -1;

                // because ı don't know which one will occur, ı gave some values and then ı will use which one changed

                if(newFl.hasNextInt()){

                        ilines1 = newFl.nextInt();

                }if(newFl.hasNext()){

                        String e = newFl.next();

                        dlines1 = Double.parseDouble(e);

                }


                String secondLine = input.nextLine();

                //I took the second line

                Scanner newSl =new Scanner(lines(secondLine));

                String name2 = newSl.next();

                int ilines2 = -1;

                double dlines2 = -1;

                if(newSl.hasNextInt()){

                        ilines2 = newSl.nextInt();

                }else{
```

```
                String e = newSl.next();
                dlines2 = Double.parseDouble(e);
            }
            // program works same as what ı did first line
            String thirdLine = input.nextLine();
            //ı took the third line
            Scanner newTl =new Scanner(lines(thirdLine));
            String name3 = newTl.next();
            int ilines3 = -1;
            double dlines3 = -1;
            if(newTl.hasNextInt()){
                ilines3 = newTl.nextInt();
            }else{
                String e = newTl.next();
                dlines3 = Double.parseDouble(e);
            }
            //again, program works same as what ı did first line
            String lastLine = input.nextLine();
            lastLine.toLowerCase();
            //because ı take names of variables in lower case, ı did my last
line in lower case
            //now ı replace the name of variable with the value of variable
            while(lastLine.contains(name1)){
                String s1 = name1.substring(0,1);
                int i1 = lastLine.indexOf(s1);
                int i2 = i1 + name1.length()-1;
                //this part is necessary to understand if my variable is
integer or double
                if(ilines1!=-1){
```

```
                    lastLine = lastLine.substring(0,i1) + ilines1 +
lastLine.substring(i2+1);
                }if(dlines1!=-1){
                    lastLine = lastLine.substring(0,i1) + dlines1 +
lastLine.substring(i2+1);
                }
            }while(lastLine.contains(name2)){
                String s1 = name2.substring(0,1);
                int i1 = lastLine.indexOf(s1);
                int i2 = i1 + name2.length()-1;
                if(ilines2!=-1){
                    lastLine = lastLine.substring(0,i1) + ilines2 +
lastLine.substring(i2+1);
                }if(dlines2!=-1){
                    lastLine = lastLine.substring(0,i1) + dlines2 +
lastLine.substring(i2+1);
                }
            }while(lastLine.contains(name3)){
                String s1 = name3.substring(0,1);
                int i1 = lastLine.indexOf(s1);
                int i2 = i1 + name3.length() -1 ;
                if(ilines3!=-1){
                    lastLine = lastLine.substring(0,i1) + ilines3 +
lastLine.substring(i2+1);
                }if(dlines3!=-1){
                    lastLine = lastLine.substring(0,i1) + dlines3 +
lastLine.substring(i2+1);
                }
            }
            while(lastLine.contains(" ")){
                int space = lastLine.indexOf(" ");
```

```
                lastLine = lastLine.substring(0,space) +
lastLine.substring(space+1);
            }
            //again, ı get rid of spaces to work more easily
            lastLine = lastLine.substring(0,lastLine.length()-1);
            //and ı get rid of ';'
            //in this part ı aim to work with the parenthesis
            while(lastLine.contains(")")){
                int iparantesis = lastLine.indexOf(")");
                //to work more easily,  ı used the index of ')'
                String forward = lastLine.substring(iparantesis+1);
                //to save the rest of line
                int max=0;
                for(int
a=0;a<lastLine.substring(0,lastLine.indexOf(")")).length();a++){
                    char c = lastLine.charAt(a);
                    if(c=='('){
                        max= a;
                    }
                }
                //to get the first '(' before the ')'

                String back = lastLine.substring(0,max);
                //to save the beginning of line
                String newLine = lastLine.substring(max+1,iparantesis);
                //now the variable newLine equals to operation which ı
need to do
                lastLine = back + cal2(calculation(newLine))+ forward ;
                //this cal2(calculation) will do my operation with
operator precedence
                //and now lastLine equals to line without this parenthesis
```

```
                }
                String finalresult = cal2(calculation(lastLine));
                System.out.println(finalresult);
                //and the result
        }
    // my first method do multiplication and division
        public static String calculation(String newLine){
            // first,ı controlled if the line contains '*' or '/' or both of them
            while(newLine.contains("*") && newLine.contains("/")){
                //when the line contains both of them, ı looked which one occurs first
                int carpi = newLine.indexOf("*");
                int bolu = newLine.indexOf("/");
                int operation =0;
                if(carpi<bolu){
                    operation = carpi;
                }else{
                    operation = bolu;
                }
            // ı described a double and an integer to each of my number which will be used because ı dont know if it is an integer or not
                double d1 = -1, d2 = -1;
                int i1 = -1, i2 = -1;

                if(sayıAlt(newLine,operation).contains(".")){
                    d1 = Double.parseDouble(sayıAlt(newLine,operation));
                }else{
                    i1 = Integer.parseInt(sayıAlt(newLine,operation));
                }
```

```
                    if(sayıUst(newLine,operation).contains(".")){
                        d2 =
Double.parseDouble(sayıUst(newLine,operation));
                    }else{
                        i2 = Integer.parseInt(sayıUst(newLine,operation));
                    }
                // and again ı described a double and an integer to my result because
ı dont know what my result is
                    int resulti = -1;
                    double resultd = -1;
                //then ı did operation
                    if(operation == carpi){
                        if(!(i1==-1)){
                            if(!(i2==-1)){
                                resulti = i1*i2;
                            }else{
                                resultd = i1*d2;
                            }
                        }else{
                            if(!(i2==-1)){
                                resultd = d1*i2;
                            }else{
                                resultd = d1*d2;
                            }
                        }
                    }else{
                        if(!(i1==-1)){
                            if(!(i2==-1)){
                                resulti = i1/i2;
```

```
                    }else{
                        resultd = i1/d2;
                    }
                }else{
                    if(!(i2==-1)){
                        resultd = d1/i2;
                    }else{
                        resultd = d1/d2;
                    }
                }
            }
        // finally ı replaced the operation with my result
            if(!(resultd==-1)){
                newLine =
newLine.substring(0,sınırAlt(newLine,operation)) + resultd +
newLine.substring(sınırUst(newLine,operation));
            }if(!(resulti ==-1)){
                newLine =
newLine.substring(0,sınırAlt(newLine,operation)) + resulti +
newLine.substring(sınırUst(newLine,operation));
            }
        }
        //the content is same as the upper one, this part is only the lines which
contains only multiplication
        while(newLine.contains("*") && !(newLine.contains("/"))){
            int operation = newLine.indexOf("*");

            double d1 = -1, d2 = -1;
            int i1 = -1, i2 = -1;
            if(sayıAlt(newLine,operation).contains(".")){
                d1 =
Double.parseDouble(sayıAlt(newLine,operation));
```

```
                }else{
                        i1 = Integer.parseInt(sayıAlt(newLine,operation));
                }
                if(sayıUst(newLine,operation).contains(".")){
                        d2 =
Double.parseDouble(sayıUst(newLine,operation));
                }else{
                        i2 = Integer.parseInt(sayıUst(newLine,operation));
                }

                int resulti = -1;
                double resultd = -1;
                if(!(i1==-1)){
                        if(!(i2==-1)){
                                resulti = i1*i2;
                        }else{
                                resultd = i1*d2;
                        }
                }else{
                        if(!(i2==-1)){
                                resultd = d1*i2;
                        }else{
                                resultd = d1*d2;
                        }
                }
                if(!(resultd==-1)){
                                newLine =
newLine.substring(0,sınırAlt(newLine,operation)) + resultd +
newLine.substring(sınırUst(newLine,operation));
                }if(!(resulti ==-1)){
```

```
                                                newLine =
newLine.substring(0,sınırAlt(newLine,operation)) + resulti +
newLine.substring(sınırUst(newLine,operation));

                                        }

                                }

                        //the content is same as the upper one, this part is only the lines which
contains only division

                                while(newLine.contains("/") && !(newLine.contains("*"))){

                                        int operation = newLine.indexOf("/");



                                        double d1 = -1, d2 = -1;

                                        int i1 = -1, i2 = -1;

                                        if(sayıAlt(newLine,operation).contains(".")){

                                                d1 =
Double.parseDouble(sayıAlt(newLine,operation));

                                        }else{

                                                i1 = Integer.parseInt(sayıAlt(newLine,operation));

                                        }

                                        if(sayıUst(newLine,operation).contains(".")){

                                                d2 =
Double.parseDouble(sayıUst(newLine,operation));

                                        }else{

                                                i2 = Integer.parseInt(sayıUst(newLine,operation));

                                        }



                                        int resulti = -1;

                                        double resultd = -1;

                                        if(!(i1==-1)){

                                                if(!(i2==-1)){

                                                        resulti = i1/i2;

                                                }else{
```

```
                                        resultd = i1/d2;
                                }
                        }else{
                                if(!(i2==-1)){
                                        resultd = d1/i2;
                                }else{
                                        resultd = d1/d2;

                                }
                        }

                        if(!(resultd==-1)){
                                        newLine =
newLine.substring(0,sınırAlt(newLine,operation)) + resultd +
newLine.substring(sınırUst(newLine,operation));
                        }if(!(resulti ==-1)){
                                        newLine =
newLine.substring(0,sınırAlt(newLine,operation)) + resulti +
newLine.substring(sınırUst(newLine,operation));
                        }

                }
                // and this method returns a line without any multiplication and
division operation
                return newLine;

        }
        // my other calculation method is cal2. this method does addition and
subtraction
        public static String cal2(String line){
        // first ı controlled that the line contains '+' or '-' or both of them
                while(line.contains("+")&&line.contains("-")){
```

```
// in this part, ı calculated the line which contains both '+' and'-'
// ı looked which one occurs first
int plus = line.indexOf("+");
int minus = line.indexOf("-");
int operation =0;
if(plus<minus){
    operation = plus;
}else{
    operation = minus;
}
// ı described a double and an integer to each of my number which will be used because ı dont know if it is an integer or not
double d1 = -1, d2 = -1;
int i1 = -1, i2 = -1;

if(sayıAlt(line,operation).contains(".")){
    d1 = Double.parseDouble(sayıAlt(line,operation));
}else{
    i1 = Integer.parseInt(sayıAlt(line,operation));
}
if(sayıUst(line,operation).contains(".")){
    d2 = Double.parseDouble(sayıUst(line,operation));
}else{
    i2 = Integer.parseInt(sayıUst(line,operation));
}
// ı described a double and an integer to my result because ı dont know if it is an integer or not
int resulti = -1;
double resultd = -1;
if(operation == plus){
```

```
            if(!(i1==-1)){
                        if(!(i2==-1)){
                                    resulti = i1+i2;
                        }else{
                                    resultd = i1+d2;
                        }
            }else{
                        if(!(i2==-1)){
                                    resultd = d1+i2;
                        }else{
                                    resultd = d1+d2;
                        }
            }
}else{
            if(!(i1==-1)){
                        if(!(i2==-1)){
                                    resulti = i1-i2;
                        }else{
                                    resultd = i1-d2;
                        }
            }else{
                        if(!(i2==-1)){
                                    resultd = d1-i2;
                        }else{
                                    resultd = d1-d2;
                        }
            }
}
// finally ı replaced the operation with my result
```

```
                    if(!(resultd==-1)){
                        line = line.substring(0,sınırAlt(line,operation)) +
resultd + line.substring(sınırUst(line,operation));
                    }if(!(resulti ==-1)){
                        line = line.substring(0,sınırAlt(line,operation)) +
resulti + line.substring(sınırUst(line,operation));
                    }
                }
                while(line.contains("+")&&!(line.contains("-"))){
                    //the content is same as the upper one, this part is only
the lines which contains only addition
                    int operation = line.indexOf("+");

                    double d1 = -1;
                    double d2 = -1;
                    int i1 = -1;
                    int i2 = -1;

                    if(sayıAlt(line,operation).contains(".")){
                        d1 = Double.parseDouble(sayıAlt(line,operation));
                    }else{
                        i1 = Integer.parseInt(sayıAlt(line,operation));
                    }

                    if(sayıUst(line,operation).contains(".")){
                        d2 = Double.parseDouble(sayıUst(line,operation));
                    }else{
                        i2 = Integer.parseInt(sayıUst(line,operation));
                    }
```

```
                int resulti = -1;
                double resultd = -1;
                if(!(i1==-1)){
                        if(!(i2==-1)){
                                resulti = i1+i2;
                        }else{
                                resultd = i1+d2;
                        }
                }else{
                        if(i2==-1){
                                resultd = d1+d2;
                        }else{
                                resultd = d1+i2;
                        }
                }
                if(!(resultd==-1)){
                        line = line.substring(0,sınırAlt(line,operation)) +
resultd + line.substring(sınırUst(line,operation));
                }if(!(resulti == -1 )){
                        line = line.substring(0,sınırAlt(line,operation)) +
resulti + line.substring(sınırUst(line,operation));
                }

        }
        while(line.contains("-")&&!(line.contains("+"))){
                //the content is same as the upper one, this part is only
the lines which contains only subtraction
                int operation = line.indexOf("-");

                double d1 = -1, d2 = -1;
```

```
int i1 = -1, i2 = -1;
if(sayıAlt(line,operation).contains(".")){
        d1 = Double.parseDouble(sayıAlt(line,operation));
}else{
        i1 = Integer.parseInt(sayıAlt(line,operation));
}
if(sayıUst(line,operation).contains(".")){
        d2 = Double.parseDouble(sayıUst(line,operation));
}else{
        i2 = Integer.parseInt(sayıUst(line,operation));
}

int resulti = -1;
double resultd = -1;

if(!(i1==-1)){
        if(!(i2==-1)){
                resulti = i1-i2;
        }else{
                resultd = i1-d2;
        }
}else{
        if(!(i2==-1)){
                resultd=d1-i2;
        }else{
                resultd = d1-d2;
        }
}
if(!(resultd==-1)){
```

```
                                        line = line.substring(0,sınırAlt(line,operation)) +
resultd + line.substring(sınırUst(line,operation));
                                    }if(!(resulti ==-1)){
                                        line = line.substring(0,sınırAlt(line,operation)) +
resulti + line.substring(sınırUst(line,operation));
                                    }
                                }
                                // and finally it returns only a number
                                return line;
                            }
                            // ı used this method to find the number which comes just before the
operator
                            public static String sayıAlt(String line, int operation){
                                line = line.substring(0,operation);
                                // ı used the variable helpingline to find last operator just before
the number
                                String helpingline = line;
                                int plus = 0;
                                int minus = 0;
                                int first = 0;
                                // ı controlled if it contains '+' or '-'. ı didnt control '*' or '/'
because my code will done this operations
                                if(line.contains("+") || line.contains("-")){
                                    while(helpingline.contains("+")){
                                        plus = plus + helpingline.indexOf("+")+1;
                                        int anotherplus = helpingline.indexOf("+");
                                        helpingline = helpingline.substring(anotherplus+1);
                                    }while(helpingline.contains("-")){
                                        minus = minus+ helpingline.indexOf("-")+1;
                                        int anotherminus = helpingline.indexOf("-");
                                        helpingline =
helpingline.substring(anotherminus+1);
```

```
                }
                if(plus<minus){
                        first = minus;
                }else{
                        first = plus;
                }
            }
        line = line.substring(first);
        //there line equals to number
        return line;
    }
    //// ı used this method to find the number which comes just after the operator
    public static String sayıUst(String line, int operation){
        line = line.substring(operation+1);
        int plus = 10000000;
        int minus = 10000000;
        int carpi = 10000000;
        int bolu = 10000000;
        int last = line.length();
        // now ı need to control '*' and '/' because there might be
        if(line.contains("+") ||line.contains("-") ||line.contains("*") ||line.contains("/")){
            if(line.contains("*")){
                carpi = line.indexOf("*");
            }if(line.contains("/")){
                bolu = line.indexOf("/");
            }if(line.contains("+")){
                plus = line.indexOf("+");
            }if(line.contains("-")){
```

```
                        minus = line.indexOf("-");
                    }
                    int last1;
                if(plus<minus){
                    last1 = plus;
                }else{
                    last1= minus;
                }
                int last2;
                if(carpi<bolu){
                    last2 = carpi;
                }else{
                    last2 = bolu;
                }
                if(last1<last2){
                    last=last1;
                }else{
                    last=last2;
                }
            }
            line = line.substring(0,last);
            //line equals to a number
            return line;
    }
    // ı used this methods to find where my operation ends
    // sınırAlt gives me the lower one
    public static int sınırAlt(String line, int operation){
            line = line.substring(0,operation);
            String helpingline = line;
```

```
                int plus = 0;
                int minus = 0;
                int first = 0;
                if(line.contains("+") || line.contains("-")){
                        while(helpingline.contains("+")){
                                plus = plus + helpingline.indexOf("+")+1;
                                int anotherplus = helpingline.indexOf("+");
                                helpingline = helpingline.substring(anotherplus+1);
                        }while(helpingline.contains("-")){
                                minus = minus + helpingline.indexOf("-")+1;
                                int anotherminus = helpingline.indexOf("-");
                                helpingline =
helpingline.substring(anotherminus+1);
                        }
                        if(plus<minus){
                                first = minus;
                        }else{
                                first = plus ;
                        }
                }
                return first;

        }
        // sısnırUst gives me the upper one
        public static int sınırUst(String line,int operation){
                int a = operation + 1;
                line = line.substring(operation+1);
                int plus = 10000000;
                int minus = 10000000;
```

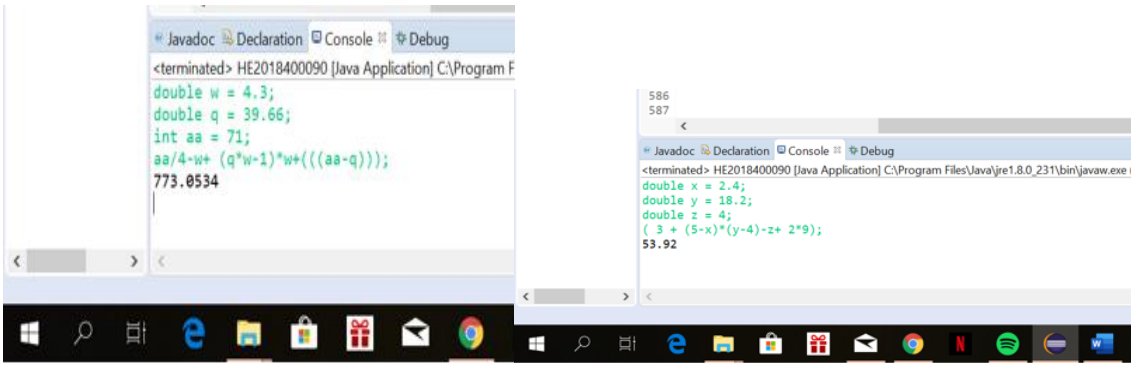
```
                int carpi = 10000000;
                int bolu = 10000000;
                int last = line.length();
                if(line.contains("+") ||line.contains("-
") ||line.contains("*") ||line.contains("/")){
                        if(line.contains("+")){
                                plus = line.indexOf("+");
                        }if(line.contains("-")){
                                minus = line.indexOf("-");
                        }if(line.contains("*")){
                                carpi= line.indexOf("*");
                        }if(line.contains("/")){
                                bolu= line.indexOf("/");
                        }
                        int last1;
                    if(plus<minus){
                        last1 = plus;
                    }else{
                        last1= minus;
                    }
                    int last2;
                    if(carpi<bolu){
                        last2 = carpi;
                    }else{
                        last2 = bolu;
                    }
                    if(last1<last2){
                        last=last1;
                    }else{
```

```
                last=last2;
            }
        }
        return last+a;
    }
    //This method helps me to take the name and the value of my variable without any character like '=' or ';'.
    public static String lines(String line){
        line.toLowerCase();
        while(line.contains(" ")){
            int space = line.indexOf(" ");
            line = line.substring(0,space) + line.substring(space+1);
        }
        //ı did this to ease my job
        String name = "";//the variable "name" will give me the name of my variable
        String number = "";//the variable "number" will give me the value of my variable

        if(line.startsWith("i")){
            line = line.substring(line.indexOf("t")+1);
            name = name + line.substring(0,line.indexOf("="));
            number = number + line.substring(line.indexOf("=")+1,line.indexOf(";"));
        }if(line.startsWith("d")){
            line = line.substring(line.indexOf("e")+1);
            name = name + line.substring(0,line.indexOf("="));
            number = number + line.substring(line.indexOf("=")+1,line.indexOf(";"));
            if(!(number.contains("."))){
                number = number + ".0";
```

```
                                        }// because the value of a double might be '2' or like this, I need to be sure that it will be perceived like a double
                                }
                                String end = name + " " + number;
                                //now ı have only the name and the value;
                                return end;
                        }
                }
```

OUTPUT OF THE PROGRAM



CONCLUSION

In conclusıon, in my opinion it was very educational, ı used most of my knowledge about java. I think ı did correctly and maybe in some cases my code might give error. İt was challenging to me. I hope ı will do other assingments more easily.