Assignment 1 Parentheses Checker

Ertuğrul Yılmaz Student ID: 041701030

Date: 16.11.2019

COMP 201 Data Structures and Algorithms

1. Algorithm Explanation

In this assignment, I had made an algorithm which displays the parentheses stack for every push and finally shows that the given input code's parentheses are correct or not. For that I took the .java code which is going to check from user as input. Then I converted that .java code to .txt file. I read that .txt file line by line and I stored it in a String variable. Then I read that variable char by char for every open parentheses it pushes to the stack and for every close parentheses it compares with the last element of the stack to see the equality. If it is equal it pops the last element from the stack. End of the given code it looks the stacks inside if it is empty parentheses are correct, if it is not parentheses are not correct for most cases except one and that is when the stack is empty and also there is one close parentheses. I figured that out with checked the stacks emptiness after checking current char.

2. Sample Outputs

In this algorithm, I checked four different java codes

- 1.Two .java files which has correct parentheses. (Figure 1) (Figure 2).
- 2.Two .java files which has wrong parentheses. (Figure 3) (Figure 4).

```
correct1
                                            import java.io.IOException;
( {
{ {
                                           /** Ertuğrul Yılmaz, ID: 041701030, 16 November 2019
                                                 class with correct parentheses.*
                                            public class CreateServlet extends HttpServlet {
                                                public void doGet(<u>HttpServletRequest</u> request, <u>HttpServletResponse</u> response)
    throws <u>ServletException</u>, <u>IOException</u> {
 << stack doubled >>
                                                     response.setContentType("text/html"
                                                     PrintWriter out=response.getWriter();
  {
  String username=request.getParameter("username");
                                                     String password=request.getParameter("password");
                                                     String repassword=request.getParameter("repassword");
                                                     String amoun=request.getParameter("amount"):
                                                     double amount=Double.parseDouble(amoun);
                                                     String adderess=request.getParameter("adderess"):
                                                     String ph=request.getParameter("phone");
                                                     double phone=Double.parseDouble(ph);
//double mname=Double.parseDouble(num)
                                                     //String country=request.getParameter("country");
(({(((()
                                                     int status=RegisterUser.register(username, password, repassword, amount, adderess,phone);
                                                     if(status>0){
                                                         out.print("WELCOME! YOUR ACCOUNT HAS OPENED");
                                                         RequestDispatcher rd=request.getRequestDispatcher("index.jsp");
                                                         rd.include(request, response):
                                                     else{
                                                         out.print("Sorry,Registration failed. please try later");
                                                         RequestDispatcher rd=request.getRequestDispatcher("MyHtml.html");
  { {
{ {
        {
                                                         rd.include(request, response);
                                                out.close();
                                                                                                                                Figure 1
Parentheses are correct
                                            }
```

```
correct2
                              package g;
{ {
{ {
                              import java.sql.*;[]
                              public class RegisterUser {
                              static int status=0;
                              /** Ertuğrul Yılmaz, ID: 041701030, 16 November 2019
                               * A class with correct parentheses.*/
                              public static int register(String username,String password,String repassword,
 << stack doubled >>
                                      double amount,String adderess,double phone){
( { {
{ { { { { { }
                                  Connection con=GetCon.getCon();
                                  PreparedStatement ps;
                                  try {
                                      ps = con.prepareStatement("Insert into NEWACCOUNT values(?,?,?,?,?,?)");
                                      int nextvalue1=GetCon.getPrimaryKey();
 << stack doubled >>
                                      ps.setInt(1,nextvalue1);
                                      ps.setString(2,username);
(({{{
ps.setString(3,password);
                                      ps.setString(4, repassword);
                                      ps.setDouble(5,amount);
                                      ps.setString(6,adderess);
                                      ps.setDouble(7,phone);
( { { { {
                                      status=ps.executeUpdate();
                                  } catch (SQLException e) {
  { { {
                                      e.printStackTrace();
( { {
return status;
                                                                                                   Figure 2
Parentheses are correct
```

```
notCorrect1
К
( {
{ {
 << stack doubled >>
          ⊝/** <u>Ertuğrul Yılmaz</u>, ID: 041701030, 16 November 2019
( { {
            * A class with wrong parentheses.*/
  {
           public class InputShort {
  { {

⇒ public static void main(String[] args) {
            /* Prompt the input */
           java.util.Scanner input = new java.util.Scanner(System.in); System.out.print(" Enter a number in pounds: ");
           double pounds = input.nextDouble();
double kilograms = pounds * 0.454;
( {
           } /* this is a wrong parenthesis */
           System.out.println(pounds + " pounds is " + kilograms + " kilograms"); /* Too many parentheses */
           int a = (((((8)))));
 << stack doubled >>
((((
 << stack doubled >>
(((({
((((({
                                                                                                            Figure 3
Parentheses do not match!
```

```
notCorrect2
                                    /** Ertuirul Yılmaz, ID: 041701030, 16 November 2019
* A class with wrong parentheses.*/
import java.io.File;[]
{
{ {
                                    public class readingFiles (
                                        public static void main(String[] args) {
                                            String file= "data2.txt";
                                            int[] numbers = loadWumbers("data2.txt");
 << stack doubled >>
                                            System.out.println(Arrays.toString(numbers)); //This line is printing array output.
( { {
                                            int[] frequency = computeFrequency(numbers);
( { {
( ( {
         {
                                        public static int[] computeFrequency(int[] numbers) { // it has to take int[] numbers array as an input.
                                            int x=0; // use x and w to find the other frequent number.
( { {
                                            int w=0;
                                            int number =0;
                                            int count=0; // This will show the frequency of our most frequent number.
{ {
                                            for(int i=0; i<numbers.length; i++) {
                                                int numberHolder= numbers[1]; //numberHolder will hold the number from array and it will count with countTwo.
                                                int countTwo=0;
                                                for(int j=0; j<numbers.length; j++) {
   if(numbers[j]==numberHolder)</pre>
 << stack doubled >>
                                                        countTwo++;
                                                    if(countTwo>count) { //if it's greater than the previous number, it means I found most frequent number.
                                                        number=numberHolder;
( { {
                                                        count=countTwo;
for(int k=0; k<numbers.length; k++) {
                                                    if((countTwo==count) & (numberHolder != number)) {
                                                        x=numberHolder;
                                                        w=countTwo;
                                                    } /*else {
 << stack doubled >>
                                                        x=0;
                                                        w=0;
                                                    }*/
   { { { { {
                                               }
   { { { {
                                           }
  System.out.println("Most Frequent Elements: ");
System.out.println("Number: " + number + "," + " Frequency: " + count);
System.out.println("Number: " + x + "," + " Frequency: " + w);
{
                                            return null:
       ł
          {
   (
             {
       {
          {
             {
   (
(
                                        public static int[] loadNumbers(String file) {
   {
       {
          {
                                            try (
                                                File myFile= new File (file);
   {
      {
         {
                                                Scanner input = new Scanner(myFile);
                                                int counter=0; // I have to know how big my file is.
                                                while (input.hasNextInt()) { // while loop will access the file and count the integers with s.nextInt() one by one.
                                                    counter++;
                                                    input.nextInt();
                                                System.out.println("Amount of integers in " + file + " file: " + counter);
                                                int[] myArray = new int[counter];
                                                Scanner input2= new Scanner (myFile);
                                                for (int i=0; i<myArray.length; i++
                                                    myArray[i]=input2.nextInt();
  << stack doubled >>
                                                return myArray;
                                            catch(Exception e) {
{ { {
                                                return null;
( { { { {
                                        }
                                    }
  << stack doubled >>
    ( { { {
   {
    { { {
(
   { { {
    { { {
   ( { { { {
Parentheses do not match!
                                                                                                                                               Figure 4
```