Object Oriented Programming with Java - Advanced Course

Project Documentation

Robin Müller-Bady

January 19, 2020

Contents

1. Introduction

2. Group Members

Name	Main Responsibilities
SU TZU CHENG	Programing
MAO YU QING	Programing
Matias Nuñez Muñoz	Cleaning
OH YEONSOO	Document Writing

3. Program Description

- Technical Description
  The Description is shown in the comment of each screemshot

(1) Result:

Small Graph:

```
Please Select the ananlizing File
1.small graph
2.med graph
3.Large graph
1

Small Graph has:
number of node=15
node name:
n0, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14
number of edge=28
edge name:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
please enter the first node
n0
please enter the second node
n8
The direct connection length of n0 and n8 is:
9
```

Medium Graph:

```
Please Select the ananlizing File
1.small graph
2.med graph
3.Large graph
2

Medium Graph has:
number of node=50
node name:
n0, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n
number of edge=112
edge name:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 5
please enter the first node
n0
please enter the second node
n42
The direct connection length of n0 and n42 is:
1
```

Large Graph:

```
Please Select the ananlizing File
1.small graph
2.med graph
3.Large graph
3

Large Graph has:
number of node=250
node name:
n0, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n
number of edge=249
edge name:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 5
please enter the first node
n237
please enter the second node
n174
The direct connection length of n237 and n174 is:
4
```

Error Message:

a. Graph Selecting

```
Please Select the ananlizing File
1.small graph
2.med graph
3.Large graph
5
wrong input, please try again
1
Small Graph has:
number of node=15
node name:
n0, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14
number of edge=28
edge name:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
please enter the first node
```

b. Node Entering

```
Small Graph has:
number of node=15
node name:
n0, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14
number of edge=28
edge name:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
please enter the first node
n15
error no such node
 ex: n0, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14
please enter the first node
n0
please enter the second node
n8
The direct connection length of n0 and n8 is:
9
```

(2) Main function

```
package test;

import java.io.BufferedReader;
import java.io.IOException;
import java.util.Scanner;
import test.Counting;
import test.FileReading;


public class Read {


//Main Function
 public static void main(String[] args) {

  BufferedReader objReader = null;
  Scanner selectmod = new Scanner(System.in);
  FileReading FileRead = new FileReading();
  Counting Count = new Counting();

  try {
          String mode = FileRead.address(selectmod);
          String data=FileRead.file(mode);
          System.out.println("number of node="+Count.nodeCount(data));
          System.out.println("node name:\n"+FileRead.nodeNameList(data));
          System.out.println("number of edge="+Count.edgeCount(data));
          System.out.println("edge name:\n"+FileRead.edgeNameList(data));
          System.out.println(Count.edgeLen(selectmod,data));
              }

  catch (IOException e) {

   e.printStackTrace();

  } finally {

   try {
    if (objReader != null)
     objReader.close();
   } catch (IOException ex) {
    ex.printStackTrace();
   }
  }

 }

}
```

(3) Class

1. Class Counting

a. int nodeCount(String)

```
/*Function 1 : int NodeCount(String)>> The Algorithm to count the node
 * This part of function is made for counting the node inside the graphML that selected.
 *
 * The  method is:
 * 1. String a is the graphML file that has already Transfer in the function File.
 * 2. We scanned the file if it contains "node id", if it does counter(nodee) add one and
 *   replace that "node id" to prevent the search next time
 * 3. After we can't find any "node id" in the String a, return the counter(nodee)
 */
int nodeCount(String a){
    int nodee =0;
     while(a.contains("node id")) {
         nodee++;
         a =a.replaceFirst("node id", "HHH");}
     return nodee;
}
```

b. int edgeCount(String)

```
/*Function 2: int EdgeCount(String)>> The Algorithm to count the edge
   * This part of function is made for counting the node inside the graphML that selected.
   *
   * The  method is:
   * 1. String a is the graphML file that has already Transfer in the function File.
   * 2. We scanned the file if it contains "edge source", if it does counter(nodee) add one
   *  and replace that "node id" to prevent the search next time
   * 3. After we can't find any "node id" in the String a, return the counter(nodee)
   */

int edgeCount(String a){
    int edgee =0;
     while(a.contains("edge source")) {
         edgee++;
         a =a.replaceFirst("edge source", "HHH");}
     return edgee;
}
```

c. String edgeLen(Scanner, String)

```
/*Function 3: String EdgeLen(Scanner, String)>> The Algorithm to find the direct length between two node
   * This part of function is made for counting the length of selected two node
   *
   * The  method is:
   * 1. Let the user input the node name node1 and node2( if there's no such node show the error message)
   * 2. Use the forth function in Counting to create the searching object into ex: " source = "n0" target = "n8" "
   * 3. Create the search object into the opposite way ex: " "source = "n8" taget="n0" "
   * 4. Use the searching object to search in the data find out the index of the words
   * 5. Print out the char between the "e_weight\>"and "<" which will be the length
   * 6. If there's no direct connection between two node show "no connection"
   */
```

```
String edgeLen(Scanner S,String a){
    FileReading FileRead = new FileReading();
    StringBuffer stb= new StringBuffer();
    Counting count1= new Counting();
    System.out.print("please enter the first node\n");
    String node1 = S.nextLine();
    String z = FileRead.nodeNameList(a);

    while(!z.contains(node1)) {
        System.out.print("error no such node\n ex: "+z+"\nplease enter the first node\n")
        node1=S.nextLine();
    }
    System.out.print("please enter the second node\n");
    String node2 = S.nextLine();
    while(!z.contains(node2)) {
        System.out.print("error no such node\n ex: "+z+"\nplease enter the second node\n"
        node2=S.nextLine();
    }
```

```
    String n = count1.gen(node1, node2);
    String n2= count1.gen(node2, node1);
    char i = 'a';
    String h =null;
    if(a.contains(n)) {
        int c=a.indexOf(n);
        int c1=0;
        c=a.indexOf("e_weight\">",c);
        c=a.indexOf(">",c)+1;
        c1=a.indexOf("<",c)-1;
        for(i=0; i <= (c1-c); i++)
            stb.append(a.charAt(c+i));
        h=stb.toString();

    }
    if(a.contains(n2)) {
        int c=a.indexOf(n2);
        int c1=0;
        c=a.indexOf("e_weight\">",c);
        c=a.indexOf(">",c)+1;
        c1=a.indexOf("<",c)-1;
        for(i=0; i <= (c1-c); i++)
            stb.append(a.charAt(c+i));
        h=stb.toString();

    }
    if(i=='a') {
        h= "no connection";
    }


    System.out.println("The direct connection length of "+node1+" and "+node2+" is:");
    return h;

}
```

d. String Gen(String, String)

```
/*Function 4: String Gen(String,String)>> Create the searching object
  ------------------------------------------------------------------------------------------------------------------------------------------
   * This part of function is made for Creating the searching object
   *
   * The  method is:
   * Use the BufferString append function to create the object
   ------------------------------------------------------------------------------------------------------------------------------------------*/
String gen(String a, String b) {
    StringBuffer sear1 = new StringBuffer();
    sear1.append("source=\"").append(a).append("\" target=\"").append(b).append("\"");
    return sear1.toString();
}
```

2. Class FileReading
   a. String address(Scanner)

```
/*Function 1 : String Address()>>> Select the Graph that needs to be analyzed
 ------------------------------------------------------------------------------------------------------------------------------------------
  * There's three graph needs to be analyzed. This is the user menu to choose the graph.
  * After the user select, this part of program will return the address to the File reading function.
 ------------------------------------------------------------------------------------------------------------------------------------------*/

String address(Scanner S) {

    System.out.println("Please Select the ananlizing File");
    System.out.println("1.small graph\n2.med graph\n3.Large graph");
    String mode=null;
    String a =S.nextLine();
    while((!a.contains("1"))&(!a.contains("2"))&(!a.contains("3"))) {
        System.out.println("wrong input, please try again");
        a=S.nextLine();
    }
    if(a.contentEquals("2")) {
        System.out.println("\nMedium Graph has:");
        mode = "/Users/denzelsu/Downloads/medium_graph.graphml";
     }
     if(a.contentEquals("1")) {
        System.out.println("\nSmall Graph has:");
        mode = "/Users/denzelsu/Downloads/small_graph.graphml";
     }
     if(a.contentEquals("3")) {
        System.out.println("\nLarge Graph has:");
        mode = "/Users/denzelsu/Downloads/large_graph.graphml";
     }
     return mode;
}
```

b. String file(String)

```
/*Function 2 : String File(String) >> Read the file from the computer
 * Method:
 * 1. input the address of the file returned from the previous function(address)
 * 2. Use the StringBuffer to store all the data into a StringBuffer
 * 3. Transfer the StringBuffer to String
 * 4. Return the String
 */

String file(String a) throws IOException {
     BufferedReader brd= new BufferedReader(new FileReader(a));
     StringBuffer sbf= new StringBuffer();
     String l=null;
     while((l= brd.readLine())!=null) {
          sbf.append(l);
      }
     brd.close();
     l=sbf.toString();
    return l;
 }
```

c. String nodeNameList(String)

```
/*Function 3 : String NodeNameList(String) >> Find out the names of all nodes
 * Method:
 * 1. Find the index of first "node id=" in data
 * 2. Find the first " after that set as n
 * 3. Find the first " after n set as n1
 * 4. Use StringBuffer to put the char into hh and add a comma and a space after that
 * 5. Change the form into String and return
 */
String nodeNameList(String a) {
    StringBuffer hh = new StringBuffer();
    String list= null;
    int n1 =0;
    while (a.contains("node id=")){
        int n = a.indexOf("node id=");
        n = a.indexOf("\"",n)+1;
        n1= a.indexOf("\"",n)-1;

        for (int i=0; i<=(n1-n); i++) {
            hh.append(a.charAt(n+i));
        }
        hh.append(", ");
        a=a.replaceFirst("node id=", "HHHH");
    }
    hh.delete(hh.length()-2, hh.length());
    list = hh.toString();
    return list;
}
```

d. String edgeNameList(String)

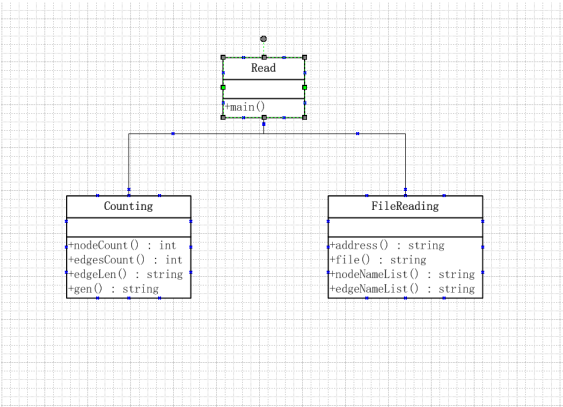
```
/*Function 3 : String EdgeNameList(String) >> Find out the names of all edges
 * Method:
 * 1. Find the index of first " e_id"> " in data
 * 2. Find the first > after that set as n
 * 3. Find the first < after n set as n1
 * 4. Use StringBuffer to put the char into hh and add a comma and a space after that
 * 5. Change the form into String and return
 */
String edgeNameList(String a) {
    StringBuffer hh = new StringBuffer();
    String list= null;
    int n1 =0;
    while (a.contains("e_id\">")){
        int n = a.indexOf("e_id\">");
        n = a.indexOf(">",n)+1;
        n1= a.indexOf("<",n)-1;
        for (int i=0; i<= (n1-n); i++) {
            hh.append(a.charAt(n+i));
        }
        hh.append(", ");
        a=a.replaceFirst("e_id\">", "HHHH");
    }
    hh.delete(hh.length()-2, hh.length());
    list = hh.toString();
    return list;
}
```

- What was implemented?

  We have already implemented the File Reading, Node Counting, Edge Counting, Showing the edge and node names and counting the direct length between two node.
  And an easy user interface to select graph.

- Data flow diagram

- Architecture UML diagram



- Description of individual parts
  1. SU TZU CHENG: Node/Edge Counting Method, OOP Designing. Writing of Class Counting() and Class FileReading().
  2. MAO YU QING: Node/Edge Counting Method,File Reading Method.
  3. OH YEONSOO: Milestone Writing, User Hand Book Writing
  4. Matias Nuñez Muñoz: Program Cleaning/organizing

4. Milestone

4.1 Milestone 1

- Things that we have achieved:
  We already achieve
  1. Read the .graphml file into the Eclipse
  2. Easy user interface to choose the analyzing diagram
  3. Calculates the Node and Edge
  4. Slightly OOP design

- Who did what?
  MAO YU QING and SU TZU CHENG are mainly in charge of the JAVA program writing.
  OH YEONSOO is mainly in charge of Documentation writing.

Matias is mainly in charge of program cleaning.

- What to achieve next?
  We are planning to achieve the algorism of node to node distant and the algorism of the shortest path calculation.
- Problem encountered:
  1. **Problem:** We were thinking to create an object to read the file using the BufferedReader. But we found out that we can only return one line of the file
     Solution: We transform the data from the BufferedReader into StringBuffer using while to sum it up and transform into String. So that we can store the whole data into a String object.

  2. **Problem:** When we are writing the NodeCounter, trying to put the if(xx.contains(String)) in a while loop, the content search index will always point to the start of the string. And we come up with two solutions.
     Solution1: Use indexOf() function to get the current index, and point the last search index at the next seach.

     Solution2: Use replace() function to replace the search target, so the next search won't get the same thing.

4.2 Milestone 2

- Things that we have achieved:
  1. Read the .graphml file into the Eclipse
  2. Easy user interface to choose the analyzing diagram
  3. Calculates the Node and Edge
  4. Calculate the direct distance of two node
  5. Show the names of all nodes and edges
  6. Slightly OOP design

- Who did what?
  MAO YU QING and **SU TZU CHENG** are mainly in charge of the JAVA program writing.
  OH YEONSOO is mainly in charge of Documentation writing.
  Matias is mainly in charge of program cleaning
- What to achieve next?
  We are planning to achieve rout planning and max/min distance counting. And project wrapping.

- Problem encountered:
  (Unsolved)When writing the method about displaying the nodes name, there is no result output. The type of return value is List<String>, and return list in the end, but it doesn't work. There has the screenshot:

```
40
41    public static  void displayVertices(List<String> list) {
42        list = split(list.toString(),"<node id = \"", "\">");
43        for(int i = 0;i<list.size();i++) {
44            System.out.println("/tvertices:" + list);
45        }
46    }
47
48    private static List<String> split(String str, String start, String end) {
49        // TODO Auto-generated method stub
50        List<String> list = new ArrayList<String>();
51        while(str != null) {
52            int a = str.indexOf(start);
53            if(a < 0)
54                break;
55            str = str.substring(a + start.length(),str.length());
56            int b = str.indexOf(end);
57            if(b < 0)
58                break;
59            String ss = str.substring(0,b);
60            list.add(ss);
61            str = str.substring(b+end.length(),str.length());
62        }
63        return list;
64    }
65 }
66
```

Problems Javadoc Declaration Console Coverage
No consoles to display at this time.