

모바일 프로그래밍

프로그래밍 과제 #3

제출일 : 2018. 11. 05.

학과 : 컴퓨터정보공학부

학번 : 2014722075

이름 : 이동준

과제결과요약

	결과	비고	자체평가
1번	검증완료		3점
2번	검증완료		3점
3번	검증완료		3점
4번	검증완료		3점
5번	검증완료		3점
6번	검증완료		3점

* 결과 - "검증완료, 디버깅필요, 미구현" 중 하나를 기재할 것.

* 비고 - 구체적인 사유를 요약해서 기재할 것. (위의 예시 참고)

* 자체평가 기준 - 3점: 검증 완료, 2점: 일부 버그 존재, 0점: 컴파일에러나 미구현

1번	결과: 검증 완료	자체평가 점수: 3점
----	-----------	-------------

- 코드

```
package Assignment3;

import java.util.Scanner;

public class A3_1 {
    public static void main(String[] args) {
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이동준"); //print Name

        Scanner keyboard = new Scanner(System.in);
        //enter the size of array A
        System.out.print("Enter size of array A: ");
        int n = keyboard.nextInt();
        //make array
        int[] arr = new int[n];
        //enter numbers in array
        System.out.print("Enter numbers: ");
        for(int i = 0; i < n; i++)
            arr[i] = keyboard.nextInt();
        //make object
        Solution S = new Solution();
        //call solution method in Solution Class
        int result = S.solution(arr);
        //print result
        System.out.print("Result: "+result);
        keyboard.close();
    }
}
```

```
class Solution
{//Class of Solution
    public int solution(int[] A)
    {//solution method
        int count = 0; //count
        for(int i = 0 ; i < A.length ; i++)
        {//if already exist value in before index i, not count
            //if not exist value in before index i, count
            boolean al = false;
            for(int j = i-1; j >= 0 ; j--)
            {
                if(A[j] == A[i])
                {
                    al = true;
                }
            }
            if( al == false )
                count++;
        }
        //return count
        return count;
    }
}
```

- 실행 결과

Student ID : 2014722075 Name : 이동준 Enter size of array A: 6 Enter numbers: 5 1 1 5 7 1 Result: 3	Student ID : 2014722075 Name : 이동준 Enter size of array A: 7 Enter numbers: 1 2 3 1 2 3 10 Result: 4
--	---

* 입력을 통해 array의 size를 지정하여 array를 생성한 뒤, 입력한 수 만큼의 number를 입력한다. 입력한 number를 array에 적재 후 solution method의 인자로 넘겨 결과를 도출한다.

Solution Class를 생성하고 Class 내부에 solution method를 생성하였다. solution method는 인자로 넘겨받은 int형의 array A를 탐색한다. for문을 이용해 각 index의 value를 검사 하는데, 현재 진행 중인 index보다 하위 index의 value들과 비교하여 이미 값이 존재를 한다면 not count, 하위 index의 value중 중복된 값이 없으면 count를 하여 중복값을 제외한 개별 값의 개수를 찾아낸다.

2번

결과: 검증 완료

자체평가 점수: 3점

- 코드

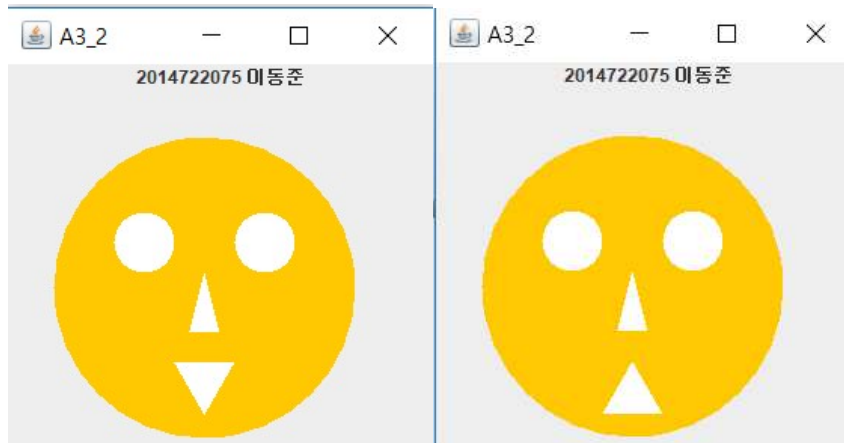
```
package Assignment3;

import java.awt.*;

class MyDrawPanel extends JPanel
{
    //Class of MyDrawPanel, it is extended by JPanel
    public void paintComponent(Graphics g)
    {
        //paintComponent method
        //make face Circle. it's color is orange
        g.setColor(Color.orange);
        g.fillOval(30,30,200,200);
        //make eyes 2 Circles. it's color is white
        g.setColor(Color.WHITE);
        g.fillOval(70,80,40,40);
        g.fillOval(150,80,40,40);
        //make nose triangle. it's color is white
        int x[] = { 120, 130, 140 };
        int y[] = { 160, 120, 160 };
        g.fillPolygon(x,y,3);
        //make mouse inverted triangle. it's color is white
        int x2[] = { 110, 130, 150 };
        int y2[] = { 180, 215, 180 };
        g.fillPolygon(x2,y2,3);
    }
}

public class A3_2 {
    public static void main(String[] args) {
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이동준"); //print Name
        //make object
        A3_2 GUI = new A3_2();
        //call go method
        GUI.go();
    }
    public void go()
    {
        //go method
        //make Frame
        JFrame frame = new JFrame();
        //if close frame, exit the program
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //set title of frame
        frame.setTitle("A3_2");
        //make label
        JLabel label = new JLabel("2014722075 이동준",JLabel.CENTER);
        //make MyDrawPanel object
        MyDrawPanel S = new MyDrawPanel();
        //insert S(face) in Center of frame
        frame.getContentPane().add(BorderLayout.CENTER, S);
        //insert Label in North of frame
        frame.getContentPane().add(BorderLayout.NORTH,label);
        //set size of frame
        frame.setSize(300, 300);
        frame.setVisible(true);
    }
}
```

- 실행 결과



<Happy>

<Sad>

* 위의 모양의 띄는 face를 생성하여 java frame을 통해 출력한다. JAVA GUI를 통해 Frame을 생성하고 Frame에 text와 그림을 적재하여 출력을 한다. main에서는 main class의 object를 생성하여 method go를 실행한다. go method는 frame을 생성 및 설정하며 text를 포함하는 label과 그림을 포함하는 Panel을 생성하고 frame에 적재한다. MyDrawPanel Class는 paintComponent method를 통해 원하는 모양의 도형을 만들고 위치와 색을 결정하여 frame에 적재한다.

3번	결과: 검증 완료	자체평가 점수: 3점
----	-----------	-------------

- 코드

<Client>

```
package Assignment3;

import java.io.*;
import java.net.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class A3_3 {
    //chat client
    JTextArea incoming;
    JTextField outgoing;
    BufferedReader reader; //declare reader
    PrintWriter writer; //declare writer
    Socket sock; //socket for connection

    public static void main(String[] args) {
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이동준"); //print Name
        new A3_3().go();
    }

    public void go()
    {
        //make frame
        JFrame frame = new JFrame("Chat Client");
        JPanel namePanel = new JPanel();
        JLabel l = new JLabel("2014722075 이동준");
        namePanel.add(l);
        JPanel mainPanel = new JPanel();
        //TextArea of chat
        incoming = new JTextArea(15,30);
        incoming.setLineWrap(true);
        incoming.setWrapStyleWord(true);
        incoming.setEditable(false);
        JScrollPane qScroller = new JScrollPane(incoming);
        qScroller.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
        qScroller.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //textfield of input chat
        outgoing = new JTextField(20);
        //make button
        JButton sendButton = new JButton("Send");
        sendButton.addActionListener(new SendButtonListener());
        mainPanel.add(qScroller);
        mainPanel.add(outgoing);
        mainPanel.add(sendButton);
        setUpNetworking(); //network

        int id = (int)(Math.random()*10 + 10); //id
        Thread readerThread = new Thread(new IncomingReader());
        readerThread.start();
        //set of frame
        frame.getContentPane().add(BorderLayout.CENTER,mainPanel);
        frame.getContentPane().add(BorderLayout.SOUTH,namePanel);
        frame.setSize(400, 500);
        frame.setVisible(true);
    }
}
```

```

private void setUpNetworking()
{
    //networking method
    try {
        sock = new Socket("127.0.0.1", 5000);
        InputStreamReader streamReader = new InputStreamReader(sock.getInputStream());
        reader = new BufferedReader(streamReader);
        writer = new PrintWriter(sock.getOutputStream());
        System.out.println("networking established");
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}

public class SendButtonListener implements ActionListener
{
    //Listener of "send" Button
    public void actionPerformed(ActionEvent ev) {
        try {
            writer.println(outgoing.getText());
            writer.flush();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        outgoing.setText("");
        outgoing.requestFocus();
    }
}

public class IncomingReader implements Runnable
{
    //read message
    public void run()
    {
        String message;
        try
        {
            //print message
            while((message = reader.readLine())!=null)
            {
                System.out.println("read " + message);
                incoming.append(message + "\n");
            }
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
        }
    }
}
}

```


<Server>

```
package Assignment3;

import java.io.*;
import java.net.*;
import java.util.*;

public class A3_3_2 {
    //chat server
    ArrayList clientOutputStreams;

    public class ClientHandler implements Runnable
    {
        BufferedReader reader;
        Socket sock;
        long id;//id

        public ClientHandler(Socket clientSocket,long i)
        {
            id = i;
            try
            {
                //client connect server
                sock = clientSocket;
                InputStreamReader isReader = new InputStreamReader(sock.getInputStream());
                reader = new BufferedReader(isReader);
            }
            catch(Exception ex)
            {
                ex.printStackTrace();
            }
        }

        public void run() {
            String message;
            try
            {
                //manage message
                while((message = reader.readLine())!=null)
                {
                    System.out.println("read " + message);
                    tellEveryone(message,id);
                }
            }
            catch(Exception ex)
            {
                ex.printStackTrace();
            }
        }
    }

    public static void main(String[] args)
    {
        new A3_3_2().go();
    }
}
```



```

public void go()
{
    clientOutputStreams = new ArrayList();
    try
    {
        ServerSocket serverSock = new ServerSocket(5000);
        //make socket
        while(true)
        {
            //connection
            Socket clientSocket = serverSock.accept();
            PrintWriter writer = new PrintWriter(clientSocket.getOutputStream());
            clientOutputStreams.add(writer);
            long r = (long)(Math.random()*10);
            Thread t = new Thread(new ClientHandler(clientSocket,r));

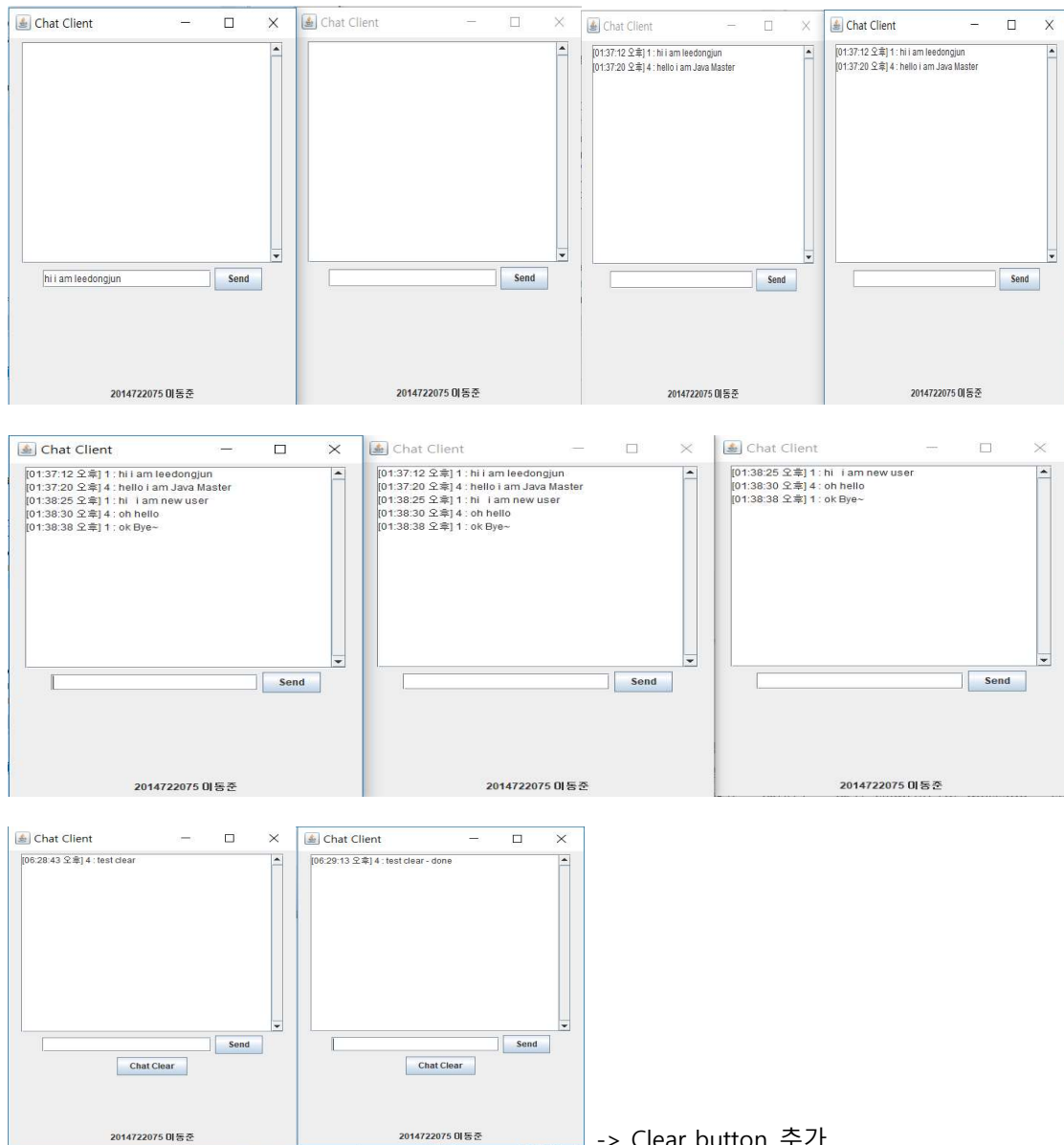
            t.start();
            System.out.println("got a connection");
        }
    }
    catch(Exception ex)
    {
        ex.printStackTrace();
    }
}

public void tellEveryone(String message,long id)
{
    //manage message
    Iterator it = clientOutputStreams.iterator();
    while(it.hasNext())
    {
        //send message
        try {
            PrintWriter writer = (PrintWriter) it.next();
            String ids = Long.toString(id);
            Date t = new Date();
            SimpleDateFormat time = new SimpleDateFormat("hh:mm:ss a");

            writer.println "["+time.format(t)+"] "+ids+" : "+message);
            writer.flush();
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
        }
    }
}

```

- 실행 결과



* 총 2개의 java file을 통해 프로그램이 동작한다. client와 server로 나누어 진행을 하며, 각 client는 server를 통해 message를 주고 받는 형식이다. Client에서는 frame을 형성하고 message를 주고 받을 수 있는 기본 환경을 설정하고 실행한다. 실행된 client끼리의 message는 server를 통해 전달이 되고 출력이 된다. thread를 이용하기 때문에 각기 다른 client를 보여주기 위해 id를 난수로 생성하여 채팅을 출력할 때 좌측에 각자의 id가 출력되게 하였다. 또한 send의 시간을 기록하는 기능을 추가 하였다. socket을 이용하여 연결 및 message를 주고 받으며, send button을 통해 입력한 message가 전달된다. 추가적으로 Chat Clear Button을 추가하여 하나의 클라이언트에서만 chat 이력을 제거하는 button을 추가하였다.

4번

결과: 검증 완료

자체평가 점수: 3점

- 코드

```
package Assignment3;

import java.awt.BorderLayout;

class Checkbox
{
    //Class of Checkbox
    int sum = 0 ; //sum of price
    JLabel b; //print price
    JFrame frame = new JFrame(); // make frame
    JCheckBox[] c = new JCheckBox[5]; //make 5 checkboxes

    Checkbox()
    {
        //constructor method of Checkbox Class
        //set title of frame
        frame.setTitle("A3_4");
        //if close frame, exit the program
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //make Panel
        JPanel panel = new JPanel();
        //set Layout of panel -> BoxLayout
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

        //make Label : print total price
        b = new JLabel("Total is "+sum, JLabel.CENTER);
        //set label border's color
        b.setBorder(BorderFactory.createLineBorder(Color.BLUE, 2));
        String[] name = new String[5];
        //price list
        name[0] = "Pizza 10,000";
        name[1] = "Oil pasta 7,500";
        name[2] = "Seafood pasta 8,000";
        name[3] = "Fried rice 6,000";
        name[4] = "Noodle 4,000";

        for( int i = 0 ; i<5 ; i++)
        {
            //make checkbox.
            c[i] = new JCheckBox(name[i]);
            //set listener
            c[i].addItemListener(new MyItemListener());
            //insert checkbox in panel
            panel.add(c[i]);
        }
        //make label
        JLabel field = new JLabel("2014722075 이동준", JLabel.CENTER);
        field.setBorder(BorderFactory.createLineBorder(Color.BLUE, 2));
        //insert label(2014722075 이동준) in North of frame
        frame.getContentPane().add(BorderLayout.NORTH, field);
        //insert panel(checkboxes) in Center of frame
        frame.getContentPane().add(BorderLayout.CENTER, panel);
        //insert label(total price) in South of frame
        frame.getContentPane().add(BorderLayout.SOUTH, b);
        //set size of frame
        frame.setSize(200,250);
        frame.setVisible(true);
    }
}
```

```

class MyItemListener implements ItemListener
{
    //Listener of checkbox
    public void itemStateChanged(ItemEvent e) {
        //if occurred Event
        int s = 1;
        if(e.getStateChange() == ItemEvent.SELECTED)
            s = 1; //if check
        else
            s = -1; //if not check
        //which checkbox event occurred
        if( e.getItem() == c[0])//1st check box
            sum += s * 10000; //set price
        else if( e.getItem() == c[1])//2nd
            sum += s * 7500;
        else if( e.getItem() == c[2])//3rd
            sum += s * 8000;
        else if( e.getItem() == c[3])//4th
            sum += s * 6000;
        else if( e.getItem() == c[4])//5th
            sum += s * 4000;

        String r = String.format("%,d",sum);
        //set text in label using new total price
        b.setText("Total is "+r);
    }
}

}

public class A3_4 {
    public static void main(String[] args) {
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이동준"); //print Name
        //call Checkbox()
        new Checkbox();
    }
}

```

- 실행 결과

2014722075 이동준	2014722075 이동준	2014722075 이동준
<input type="checkbox"/> Pizza 10,000	<input checked="" type="checkbox"/> Pizza 10,000	<input checked="" type="checkbox"/> Pizza 10,000
<input type="checkbox"/> Oil pasta 7,500	<input type="checkbox"/> Oil pasta 7,500	<input checked="" type="checkbox"/> Oil pasta 7,500
<input type="checkbox"/> Seafood pasta 8,000	<input type="checkbox"/> Seafood pasta 8,000	<input checked="" type="checkbox"/> Seafood pasta 8,000
<input type="checkbox"/> Fried rice 6,000	<input checked="" type="checkbox"/> Fried rice 6,000	<input checked="" type="checkbox"/> Fried rice 6,000
<input type="checkbox"/> Noodle 4,000	<input checked="" type="checkbox"/> Noodle 4,000	<input checked="" type="checkbox"/> Noodle 4,000
Total is 0	Total is 20,000	Total is 35,500

* 5개의 음식의 price를 지정하고 checkbox를 생성하여 만약 check가 되었을 시 check가 된 box에 한해서 total price를 측정하고 출력을 한다. 각 checkbox의 변화에 따라 total price는 유동적이다. Checkbox Class를 생성하여 checkbox의 상태와 price를 관리한다. 총 5개의 checkbox를 생성하고 각 checkbox는 listener를 setting하여 MyItemListener method를 통해 관리된다. 어떠한 checkbox에서 event가 발생하였는 지를 구분하여 check인지 not check인지를 판단하여 total price를 결정하고 재출력을 한다.

5번	결과: 검증 완료	자체평가 점수: 3점
----	-----------	-------------

- 코드

```
package Assignment3;

import java.awt.*;

public class A3_5 implements ActionListener {
    int line = 0; //num of lines
    int word = 0; //num of words
    int chars = 0; //num of chars
    //each labels show number of each values
    JLabel l1 = new JLabel("Number of lines: "+line, JLabel.CENTER);
    JLabel l2 = new JLabel("Number of words: "+word, JLabel.CENTER);
    JLabel l3 = new JLabel("Number of chars: "+chars, JLabel.CENTER);
    //make text area
    JTextArea text = new JTextArea(5, 23);
    //make 2 buttons
    JButton button = new JButton("Process the Text");
    JButton button2 = new JButton("Clear the Text");

    public static void main(String[] args) {
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이종준"); //print Name
        //make object
        A3_5 gui = new A3_5();
        gui.go(); //call go method
    }

    public void go()
    { //go method
        //make Frame
        JFrame frame = new JFrame();
        //set title of frame
        frame.setTitle("A3_5");

        //setting buttons
        button.setSize(300, 30);
        button2.setSize(300, 30);
        //set listener of process button
        button.addActionListener(this);
        //set listener of clear button
        button2.addActionListener(new ActionListener()
        { //listener of clear button
            public void actionPerformed(ActionEvent e)
            { //clear text area and reset each values
                text.setText("");
                line = 0; word = 0; chars = 0;
                l1.setText("Number of lines: "+line);
                l2.setText("Number of words: "+word);
                l3.setText("Number of chars: "+chars);
            }
        });
        //if close frame, exit the program
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //make Panel
        JPanel panel = new JPanel();
        //insert buttons in panel
        panel.add(button);
        panel.add(button2);
        //set Layout of panel -> BorderLayout
        panel.setLayout(new BorderLayout(panel, BorderLayout.Y_AXIS));
    }
}
```

```

//setting each labels
l1.setSize(300, 30);
l1.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2));

l2.setSize(300, 30);
l2.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2));

l3.setSize(300, 30);
l3.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2));
//insert each labels in panel
panel.add(l1);
panel.add(l2);
panel.add(l3);

JLabel name = new JLabel("2014722075 이동준",JLabel.CENTER);
//make scrollbar including text area
JScrollPane scrollbar = new JScrollPane(text);
text.setLineWrap(true);
//set
scrollbar.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
scrollbar.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
//make panel2
JPanel panel2 = new JPanel();
//insert scrollbar in panel2
panel2.add(scrollbar);

//insert panels and label in frame
frame.getContentPane().add(BorderLayout.CENTER, panel2);
frame.getContentPane().add(BorderLayout.SOUTH, panel);
frame.getContentPane().add(BorderLayout.NORTH, name);
//set size of frame
frame.setSize(300, 300);
frame.setVisible(true);
}

```

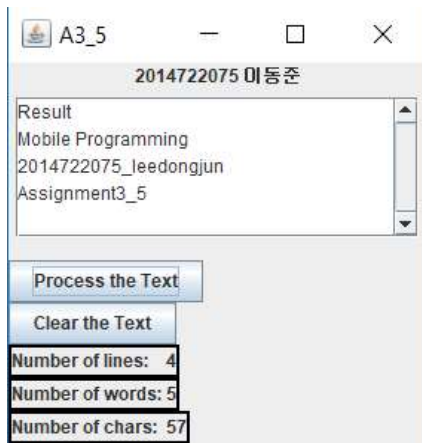
```

public void actionPerformed(ActionEvent ev) {
    //listener of process button
    line = 0; word = 0; chars = 0;

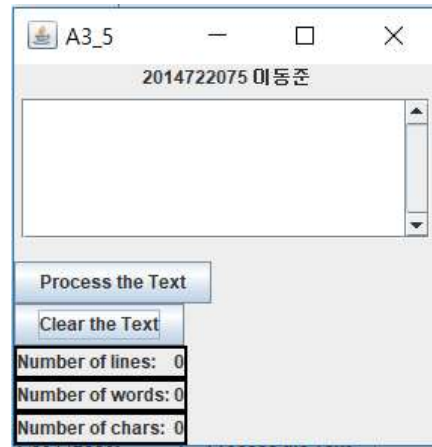
    String t = text.getText();
    char[] c = new char[t.length()];
    c = t.toCharArray();
    //find lines,words,chars
    for(int i = 0; i<t.length(); i++)
    {
        if(c[i] == ' ')
            word++;
        else if(c[i] == '\n' || i == t.length()-1)
        {
            line++;
            if(c[i-1] != '\n' && c[i-1] != ' ')
            {
                word++;
            }
        }
        if(c[i] != '\n' && c[i] != ' ')
            chars++;
    }
    //setting label using new each values
    l1.setText("Number of lines: "+line);
    l2.setText("Number of words: "+word);
    l3.setText("Number of chars: "+chars);
}
}

```

- 실행 결과



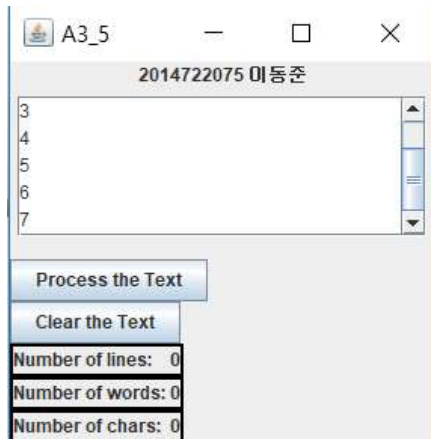
->



<Enter text and press Process button>

<press Clear button>

* Frame에 textArea와 button을 추가하여 입력한 text에 대한 line, word, chars의 수를 계산하여 출력을 한다. text를 입력 후 "Process the Text" Button을 누르면 계산이 실행되어 결과를 출력하고, "Clear the Text" Button을 누르면 textArea를 empty상태로 만들고 결과를 초기화 한다. 우선 Frame을 생성 및 설정하고 그 안에 각 요소들을 추가한다. Button 두 개에 대한 event listener를 따로 설정하여 두 button에 대한 동작을 달리 설정한다. main에서 실행하는 go method를 통해 동작이 수행되고, actionPerformed method를 통해 "Process the Text" Button에 대한 결과 계산과 동작을 제어한다. 또한 scrollbar를 설치하여 눈에 보이는 area 이상의 text가 입력되었을 경우 우측 스크롤을 통하여 text를 관리한다.



<스크롤 예제>

6번	결과: 검증 완료	자체평가 점수: 3점
----	-----------	-------------

- 코드

```
package Assignment3;

import java.io.File;

public class A3_6 {
    public static void main(String[] args) {
        System.out.println("Student ID : 2014722075"); //print ID
        System.out.println("Name : 이동준"); //print Name
        Scanner keyboard = new Scanner(System.in);
        while(true)
        { //loop
            System.out.println("\n\nNow Path is : 'c:/Users/'");
            System.out.print("Enter the Directory :");
            //insert path
            String path = keyboard.nextLine();
            if(path.equals("end"))
                break; //if enter the "end", exit
            String source = "c:/Users/";
            source = source + path; //make path
            System.out.println("< RESULT in "+source+">");
            subDirList(source); //call subDirList method
        }
        //end
        System.out.println("< END >");
        keyboard.close();
    }

    public static void subDirList(String p){
        File dir = new File(p); //make file object
        File[] fileList = dir.listFiles(); //assign list
        try{
            //visit all of files
            for(int i = 0 ; i < fileList.length ; i++){
                File file = fileList[i];

                if(file.isFile()){ //if file
                    //print file name
                    if(!file.getParent().equals(p))
                        System.out.println(i+1+": File name : " + file.getName());
                    else
                        System.out.println("\t - File name : " + file.getName());
                }
                else if(file.isDirectory()){ //if directory
                    //print directory name
                    if(!file.getParent().equals(p))
                        System.out.println(i+1+": Directory name : " + file.getName());
                    else
                        System.out.println("\t - Directory name : " + file.getName());
                    //recursive call if directory
                    subDirList(file.getCanonicalPath().toString());
                }
            }
        } catch(IOException e) { }
    }
}
```

- 실행 결과

Student ID : 2014722075

Name : 이동준

Now Path is : 'c:/Users/'

Enter the Directory : leedongjun/A3_6

< RESULT in c:/Users/leedongjun/A3_6>

1: Directory name : Directory1

- Directory name : Directory1_1

- File name : File1_1_1.txt

- File name : File1_1.txt

- File name : File1_2.txt

- File name : File1_3.txt

2: Directory name : Directory2

- File name : File2_1.txt

- File name : File2_2.txt

- File name : File2_3.txt

3: Directory name : Directory3

4: File name : File_A3_6.txt

Now Path is : 'c:/Users/'

Enter the Directory : leedongjun/A3_6/Directory1

< RESULT in c:/Users/leedongjun/A3_6/Directory1>

1: Directory name : Directory1_1

- File name : File1_1_1.txt

2: File name : File1_1.txt

3: File name : File1_2.txt

4: File name : File1_3.txt

Now Path is : 'c:/Users/'

Enter the Directory : end

< END >

* Now Path의 상황에서 입력한 path의 directory와 file의 list를 출력한다. 만약 directory의 경우 해당 directory의 subdirectory 및 내부 file까지 출력을 한다. loop문을 통해 입력을 받아 해당되는 directory를 탐색하고 'end'를 입력했을 경우 프로그램이 종료한다. subDirList method를 통해 탐색 동작이 진행되며, 입력한 directory의 모든 file과 directory의 list를 받아와 출력을 하며 만약 directory의 경우에는 recursive call을 통해 내부의 file과 directory까지 출력을 한다.