

Universidad Nacional Autónoma de México

Facultad de estudios Superiores Aragón

Ingeniería en Computación

Área: Programación

Materia: Estructura de Datos

Profesor: Roberto Blanco Bautista

Título: Insertar un Dato

Alumno: Hernández Gorgonio Henry

Grupo: 2307

Fecha: 28/02/2022

Insertar un Dato

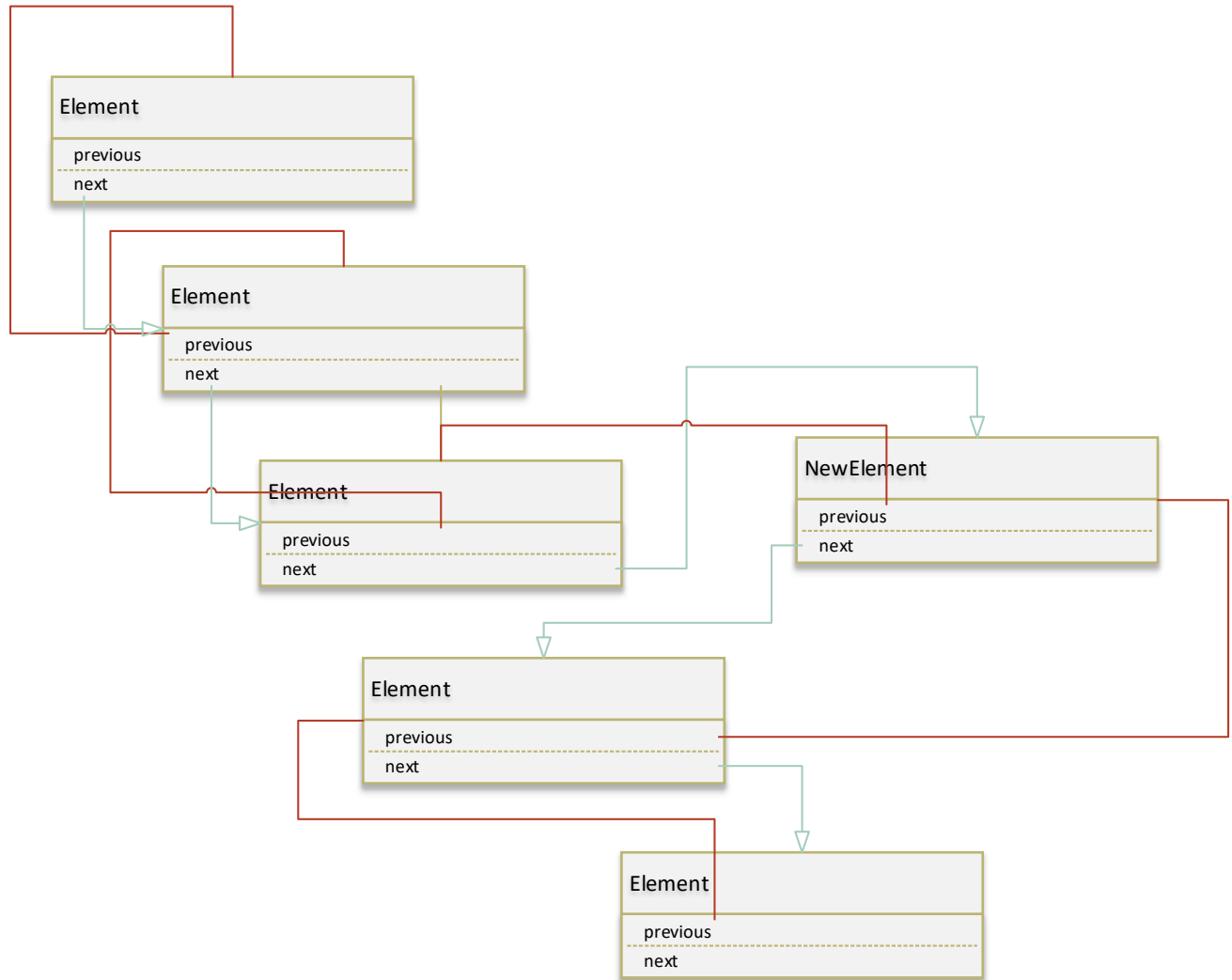
- a) Explicar el procedimiento y argumentar por qué es conveniente efectuar este proceso

Procedimiento.

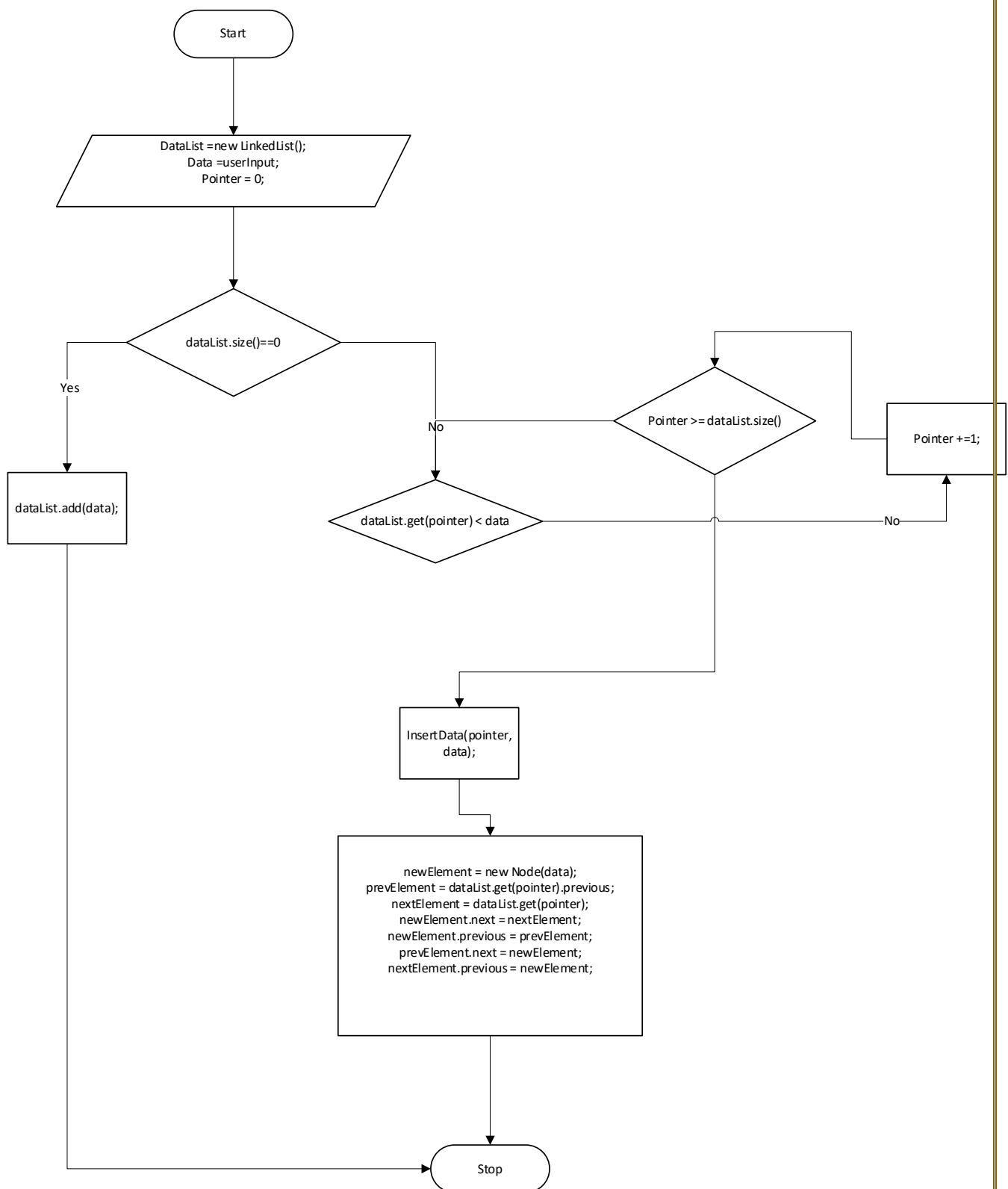
El algoritmo del programa hace uso de la estructura de datos conocida como LinkedList o Lista Enlazada, en la que cada uno de los elementos consiste en un objeto que tiene en sus atributos el "link" tanto a su predecesor como a su sucesor, esta estructura de datos es conveniente cuando no es necesario que los elementos contenidos tengan una posición continua en la memoria y se requiera hacer uso de una estructura de datos capaz de modificar su tamaño de manera continua.

El procedimiento por el cual se insertan nuevos datos, una vez obtenido el input del usuario, es comparar la palabra de entrada, con los elementos contenidos en la estructura de datos, partiendo desde el primero de ellos. Si el elemento contenido en la estructura de datos es lexicográficamente menor al introducido por el usuario, se pasa al siguiente. Cuando una palabra contenida en la lista enlazada resulta ser lexicográficamente mayor que la introducida por el usuario la variable "pointer" almacena el número de elemento en el que la palabra mayor se encuentra. Posteriormente se hace uso de la función perteneciente a la lista enlazada llamada "add(int index, E element)" que agrega el elemento "element" en el lugar "index".

Esta inserción se realiza de la siguiente forma:







- b) Hacer el diagrama de flujo para este procedimiento indicando el significado de las variables usadas en el diagrama



- c) Codificar el diagrama en el lenguaje que dominen y anexar una impresión del código fuente

Main

```
1   /*
2  |   * Click nbfs://nbhost/SystemFileSystem/Templates/I
3  |   * Click nbfs://nbhost/SystemFileSystem/Templates/C
4  |   */
5  |   package insertdata2;
6  |
7   /**
8  |   *
9  |   * @author Hoeco
10 |   */
11 |   public class InsertData2 {
12 |
13 |        /**
14 |       |   * @param args the command line arguments
15 |       |   */
16 |        public static void main(String[] args) {
17 |       |   // TODO code application logic here
18 |       |   View app = new View();
19 |       |   }
20 |
21 |   }
22 |
```

MainPanel

```
5  package insertdata2;
6  import java.awt.BorderLayout;
7  import java.awt.Color;
8  import javax.swing.JPanel;
9
10 /**
11  *
12  * @author Hoeco
13  */
14 public final class MainPanel extends JPanel{
15
16     public MainPanel() {
17         this.setLayout(new BorderLayout());
18         this.setBackground(Color.blue);
19         this.loadComponents();
20     }
21
22     public void loadComponents() {
23
24     }
25 }
26
```

HeadPanel

```
5  package insertdata2;
6  import insertdata2.headpanelfolder.TitlePanel;
7  import insertdata2.headpanelfolder.ClockPanel;
8  import insertdata2.headpanelfolder.DatePanel;
9  import java.awt.Color;
10 import java.awt.GridLayout;
11 import javax.swing.JLabel;
12 import javax.swing.JPanel;
13 import javax.swing.SwingConstants;
14
15 /**
16  *
17  * @author Hoeco
18  */
19 public final class HeadPanel extends JPanel{
20     private ClockPanel clockPanel;
21     private TitlePanel titlePanel;
22     private DatePanel datePanel;
23
24     public HeadPanel() {
25         this.setLayout(new GridLayout(1,3));
26         this.setBackground(Color.gray);
27         this.loadComponents();
28     }
29
30     public void loadComponents() {
31
32         clockPanel = new ClockPanel();
33         this.add(clockPanel);
34         titlePanel = new TitlePanel();
35         this.add(titlePanel);
36         datePanel = new DatePanel();
37         this.add(datePanel);
38     }
39 }
40
```

ClockPanel

```
5  package insertdata2.headpanelfolder;
6
7  import java.awt.Color;
8  import java.awt.Font;
9  import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11 import java.text.SimpleDateFormat;
12 import java.util.Calendar;
13 import java.util.TimerTask;
14 import java.util.logging.Level;
15 import java.util.logging.Logger;
16 import javax.swing.JLabel;
17 import javax.swing.JPanel;
18 import javax.swing.Timer;
19
20 /**
21  *
22  * @author Hoeco
23  */
24 public class ClockPanel extends JPanel implements ActionListener{
25     JLabel timeLabel;
26     SimpleDateFormat timeFormat;
27     Calendar calendar;
28     String time;
29     Timer tick;
30
31
32     public ClockPanel() {
33         timeFormat = new SimpleDateFormat(" HH : mm : ss ");
34         this.timeLabel = new JLabel();
35
36         timeLabel.setFont(new Font("SansSerif", Font.PLAIN, 25));
37         timeLabel.setForeground(new Color(0x505050));
38         this.add(timeLabel);
39         //this.setBackground(Color.black);
40
41         this.tick = new Timer(1000, this);
42         tick.start();
43     }
44
45     @Override
46     public void actionPerformed(ActionEvent e) {
47         time = timeFormat.format(Calendar.getInstance().getTime());
48         timeLabel.setText(time);
49     }
50 }
51
52
```


DatePanel

```
5  package insertdata2.headpanelfolder;
6
7  import java.awt.Color;
8  import java.awt.Font;
9  import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11 import java.text.SimpleDateFormat;
12 import java.util.Calendar;
13 import javax.swing.JButton;
14 import javax.swing.JLabel;
15 import javax.swing.JPanel;
16 import javax.swing.Timer;
17
18 /**
19  *
20  * @author Hoeco
21  */
22 public class DatePanel extends JPanel implements ActionListener{
23     Calendar calendar;
24     SimpleDateFormat dateFormat;
25     JLabel dateLabel;
26     Timer hourChange;
27     String date;
28
29     public DatePanel() {
30         this.dateFormat = new SimpleDateFormat("dd / MMM / Y");
31         this.dateLabel = new JLabel ();
32         setDate();
33         this.dateLabel.setFont(new Font("SansSerif", Font.PLAIN, 25));
34         this.dateLabel.setForeground(new Color(0x505050));
35         //this.setBackground(Color.BLACK);
36         this.add(dateLabel);
37         this.hourChange = new Timer(3600000,this);
38         this.date="";
39     }
40
41     @Override
42     public void actionPerformed(ActionEvent e) {
43         setDate();
44     }
45
46     public void setDate() {
47         date = dateFormat.format(Calendar.getInstance().getTime());
48         dateLabel.setText(date);
49     }
50 }
51
```

TitlePanel

```
4  -  ~/  
5  package insertdata2.headpanelfolder;  
6  
7  import java.awt.Font;  
8  import javax.swing.JLabel;  
9  import javax.swing.JPanel;  
10  
11  /**  
12   *  
13   * @author Hoeco  
14   */  
15  public class TitlePanel extends JPanel(  
16      JLabel title;  
17  
18  public TitlePanel () {  
19      loadComponents();  
20  }  
21  
22  public void loadComponents() {  
23      this.title = new JLabel("    Data Insertion    ");  
24      title.setFont(new Font("SansSerif", Font.PLAIN, 30));  
25      this.add(title);  
26  }  
27  }  
28
```

CentralPanel

```
5  package insertdata2;
6  import insertdata2.centralpanelfolder.FormPanel;
7  import insertdata2.centralpanelfolder.*;
8  import insertdata2.headpanelfolder.ClockPanel;
9  import java.awt.Color;
10 import java.awt.GridLayout;
11 import javax.swing.BorderFactory;
12 import javax.swing.JPanel;
13 import javax.swing.border.Border;
14
15 /**
16  *
17  * @author Hoeco
18  */
19 public final class CentralPanel extends JPanel{
20     FormPanel formPanel;
21     DataListPanel dataListPanel;
22     Model appModel;
23
24     public CentralPanel() {
25         this.setLayout(new GridLayout(1,2,50,50));
26         this.setBorder(BorderFactory.createEmptyBorder(100,20,30,40));
27         this.setBackground(Color.LIGHT_GRAY);
28         this.appModel = new Model();
29         this.dataListPanel = new DataListPanel(appModel);
30         loadComponents();
31     }
32
33     public void loadComponents() {
34
35         formPanel = new FormPanel(appModel, dataListPanel);
36         this.add(formPanel);
37         this.add(dataListPanel);
38     }
39 }
40
```

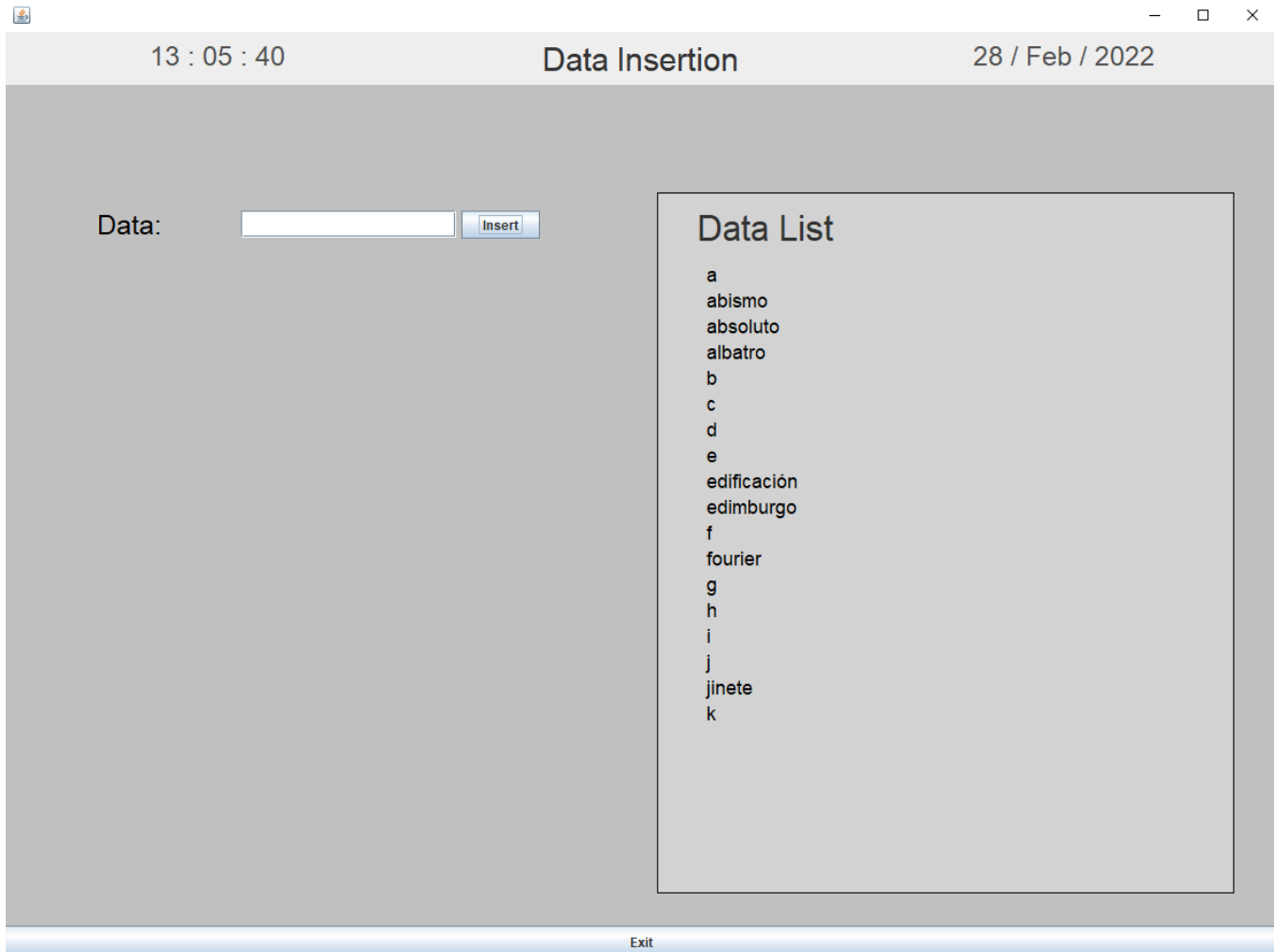
FormPanel

```
--
22 public final class FormPanel extends JPanel implements ActionListener{
23     JLabel dataLabel;
24     JPanel voidPanel;
25     JTextField dataInput;
26     JButton submitButton;
27
28     Model appModel;
29     DataListPanel dataListPanel;
30
31     public FormPanel ( Model appModel, DataListPanel dataListPanel) {
32         this.dataLabel = new JLabel(" Data: ");
33         this.dataLabel.setFont(new Font("SansSerif", Font.PLAIN, 25));
34         this.dataLabel.setForeground(new Color(0x000000));
35         this.add(dataLabel);
36
37         this.voidPanel = new JPanel();
38         this.voidPanel.setPreferredSize(new Dimension(50,50));
39         this.voidPanel.setBackground(Color.LIGHT_GRAY);
40         this.add(voidPanel);
41
42         this.dataInput = new JTextField();
43         this.dataInput.setPreferredSize(new Dimension(200,25));
44         this.add(dataInput);
45
46         this.submitButton = new JButton(" Insert ");
47         this.submitButton.addActionListener(this);
48         this.add(submitButton);
49         this.setBackground(Color.LIGHT_GRAY);
50
51         this.appModel = appModel;
52         this.dataListPanel = dataListPanel;
53     }
54
55     @Override
56     public void actionPerformed(ActionEvent e) {
57         System.out.println(dataInput.getText());
58         appModel.insertData(dataInput.getText());
59         dataListPanel.removeAll();
60         dataListPanel.refreshData(appModel.returnData());
61         dataListPanel.updateUI();
62         this.dataInput.setText("");
63     }
64 }
65
66
```

DataListPanel

```
33 public final class DataListPanel extends JPanel implements ActionListener{
34     ArrayList<JLabel> dataListLabels;
35     JLabel tableTitle;
36     Border blackLine;
37
38
39     Model appModel;
40
41     public DataListPanel( Model appModel) {
42         this.setLayout(new BorderLayout(this, BorderLayout.Y_AXIS));
43         this.setBackground(new Color(210,210,210));
44         //this.data = new String[10];
45         this.dataListLabels = new ArrayList<>();
46         //for(int i =0; i<data.length; i++) {data[i]="e";}
47         this.appModel = appModel;
48         this.blackLine = BorderFactory.createLineBorder(Color.black);
49         this.setBorder(blackLine);
50         loadComponents();
51     }
52
53     public void loadComponents() {
54
55         this.tableTitle = new JLabel("    Data List");
56         this.tableTitle.setFont(new Font("SansSerif", Font.PLAIN, 33));
57
58         tableTitle.setBorder(BorderFactory.createEmptyBorder(10,0,10,10));
59         this.add(tableTitle);
60
61     }
62
63
64     public void refreshData(String[] dataArray){
65         this.add(tableTitle);
66         for (int i = 0; i<dataArray.length; i+=1) {
67             dataListLabels.add(i, new JLabel("    " + dataArray[i]));
68             dataListLabels.get(i).setFont(new Font("SansSerif", Font.PLAIN, 18));
69             this.dataListLabels.get(i).setForeground(new Color(0x000000));
70             this.add(dataListLabels.get(i));
71
72         }
73     }
74
75     @Override
76     public void actionPerformed(ActionEvent e) {
77         this.refreshData(appModel.returnData());
78     }
79 }
```

d) Anexar una copia de la corrida del programa ejecutable



e) Interpretar los resultados obtenidos.

No hay mucho qué decir, creo que el programa es funcional y se adecúa a los requerimientos establecidos. Se utiliza una Lista enlazada pero pudo haberse hecho con Array, con ArrayList, u otras estructuras de datos similares.