

EE3206 Java Programming and Applications
Sem A, 2018/19
Midterm test

Three Java files are given to you, MidTermTest.java, TelDirectory.java, and Pair.java. Read the documents carefully before you start to answer the questions.

In your answers, you need not copy the program codes given to you.

Question 1 [10 marks]

Complete the design of the main() method in MidTermTest.java. The main() method is required to do the following:

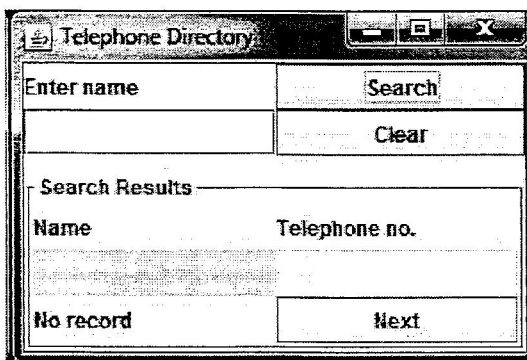
- Create an object instance of TelDirectory.
- Use JOptionPane.showInputDialog to prompt the user to enter the file name that stores a list of telephone records. The API of showInputDialog is shown below.

```
static String    showInputDialog(Object message)
                Shows a question-message dialog requesting input from the user.
```

- Call the method readDataFile() in class TelDirectory to read in the list of telephone records. You need to design the interface of the readDataFile() method.
- If the data file is not found, the program prompts the user to try again.
- After successfully reading in the list of telephone records, the frame window is made visible.

Question 2

The GUI layout is defined for you in the constructor.



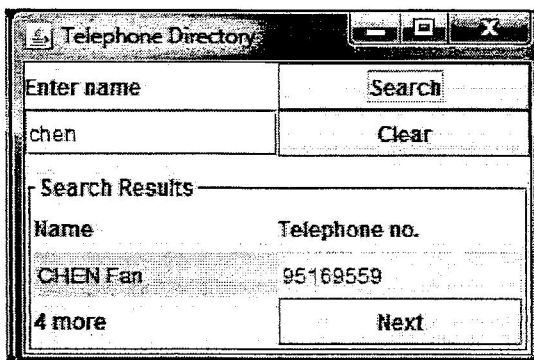
- (a) [10 marks] Design the readDataFile() method to read in the list of telephone records from the given data file to an ArrayList<Pair<String, String>>. The data file stores one telephone record per line. Each telephone record contains a name and a telephone number separated by a comma “,”. After reading in the telephone records, sort the ArrayList by name.
- (b) [5 marks] Complete the implementation of the constructor of the class TelDirectory.
- (c) [25 marks] Implement the actionPerformed() method of the class ButtonListener.

Interactions of the program with the user are as follow.

- The user enters a name (or part of a name) in the `searchKey` of the `inputPanel`.
- When the user clicks on the `Search` button or hit the enter key in the keyboard, the program searches the list of telephone records that contain the given name. The search is case-insensitive. You can use the method `contains` in class `String` (API of the method `contains` is shown below).

`boolean` `contains(CharSequence s)`
Returns true if and only if this string contains the specified sequence of char values.

- If no matching record is found, the program displays "No record" in the `moreLabel` of the `resultPanel` (as shown in the above diagram). You can use the `setText()` method to modify the text shown in the `JLabel`.
- If 1 or more matching records are found, results of the search will be displayed in the `resultPanel` one record at a time.
- The number of remaining records in the result list is displayed in the `moreLabel` of the `resultPanel`.



- When the user clicks on the `Next` button, the next result record (if available) will be displayed. If there is no more result record to show, the `nextLabel` will show "No record" and the `nameField` and `telField` are cleared.
- When the user clicks on the `Clear` button, the GUI is reset to the initial state.

```
// MidTermTest.java
```

```
import java.io.FileNotFoundException;
import javax.swing.JOptionPane;
```

```
public class MidTermTest
{
    public static void main(String[] args)
    {
        // Question 1. Implement the main() method
    }
}
```

```
-----
// Pair.java
```

```
public class Pair<S, T> // DO NOT modify this class
{
    private S first;
    private T second;

    public Pair(S n1, T n2)
    {
        first = n1;
        second = n2;
    }

    public S getFirst()
    {
        return first;
    }

    public T getSecond()
    {
        return second;
    }

    public void setFirst(S v)
    {
        first = v;
    }

    public void setSecond(T v)
    {
        second = v;
    }

    @Override
    public String toString()
    {
        return "(" + first + ", " + second + ")";
    }
}
```

```
// TelDirectory.java

import java.awt.Color;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Comparator;
import java.util.Scanner;
import javax.swing.*;
import javax.swing.border.*;

public class TelDirectory extends JFrame
{
    private final JTextField searchKey, nameField, telField;
    private final JLabel searchLabel, nameLabel, telLabel, moreLabel;
    private final JButton search, clear, next;

    // list of telephone records
    private ArrayList<Pair<String,String>> list;

    // ArrayList to store the search results
    private ArrayList<Pair<String,String>> resList;

    public TelDirectory(String filename)
    {
        // Need NOT copy the program codes given to you in your answer.

        final int DEFAULT_FRAME_WIDTH = 300;
        final int DEFAULT_FRAME_HEIGHT = 200;

        setTitle("Telephone Directory");
        setSize(DEFAULT_FRAME_WIDTH, DEFAULT_FRAME_HEIGHT);

        searchKey = new JTextField(20);
        nameField = new JTextField(20);
        telField = new JTextField(8);
        searchKey.setEditable(true);
        nameField.setEditable(false);
        nameField.setBackground(Color.lightGray);
        telField.setEditable(false);
        telField.setBackground(Color.yellow);

        searchLabel = new JLabel("Enter name");
        nameLabel = new JLabel("Name");
        telLabel = new JLabel("Telephone no.");
        moreLabel = new JLabel("No record");
        moreLabel.setForeground(Color.blue);

        search = new JButton("Search");
        clear = new JButton("Clear");
        next = new JButton("Next");

        JPanel inputPanel = new JPanel();
        inputPanel.setLayout(new GridLayout(2, 2));
        inputPanel.add(searchLabel);
        inputPanel.add(search);
        inputPanel.add(searchKey);
        inputPanel.add(clear);

        Border redLine = BorderFactory.createLineBorder(Color.red);
```

```
TitledBorder border = BorderFactory.  
    createTitledBorder(redLine, "Search Results");
```

```
border.setTitleColor(Color.red);  
JPanel resultPanel = new JPanel();  
resultPanel.setBorder(border);  
resultPanel.setLayout(new GridLayout(3, 2));  
resultPanel.add(nameLabel);  
resultPanel.add(telLabel);  
resultPanel.add(nameField);  
resultPanel.add(telField);  
resultPanel.add(moreLabel);  
resultPanel.add(next);
```

```
add(inputPanel, "North");  
add(resultPanel, "South");
```

```
// DO NOT copy the above program codes in your answer.
```

```
// Question 2(b). Add your codes to complete the implementation  
// of the constructor.
```

```
}
```

```
// Question 2(a). Design a method readDataFile() to read in the  
// telephone records from a file.
```

```
private class ButtonListener implements ActionListener  
{
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent event)
```

```
    {
```

```
        // Question 2(c). Implement the actionPerformed() method.
```

```
    }
```

```
}
```

```
}
```