

Due date: 6 November 2019 (Thu)

Assignment 4

Full mark: 100

Expected normal time spent: 6 hours

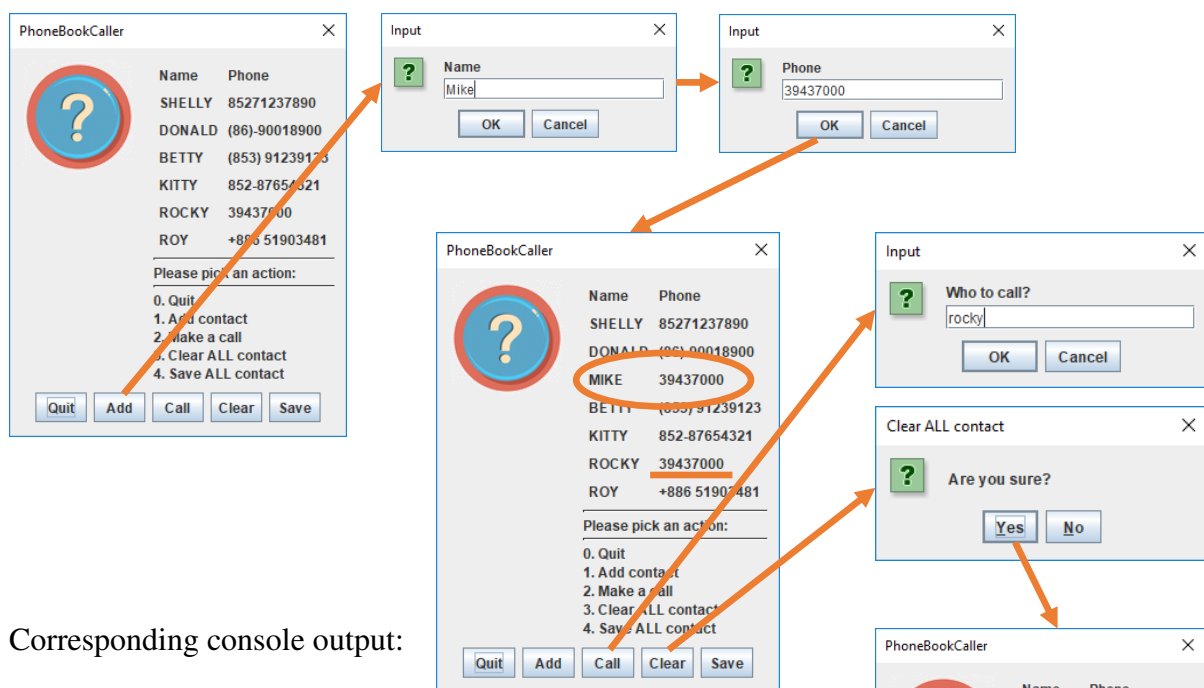
Phone Book Caller

- Aim:
1. build a practical phone book application using Java.
 2. practise using existing classes and objects.
 3. practise creating our own methods and sending messages

Task: Create a Java application for phone book management and making calls with MP3 recordings!

More information (for enthusiastic ones): http://en.wikipedia.org/wiki/Dual-tone_multi-frequency

The application starts with playing a dial tone (MP3), as a "welcome message." It maintains a phone book and allows the user to add (or change) a contact, make a call, clear and save all contacts.



Corresponding console output:

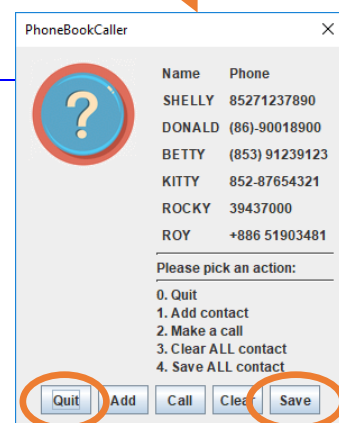
Name	Phone
SHELLY	85271237890
DONALD	(86)-90018900
BETTY	(853) 91239123
KITTY	852-87654321
ROCKY	39437000
ROY	+886 51903481

Choice: 1

Name	Phone
SHELLY	85271237890
DONALD	(86)-90018900
MIKE	39437000
BETTY	(853) 91239123
KITTY	852-87654321
ROCKY	39437000
ROY	+886 51903481

Choice: 2

Calling rocky: 3 9 4 3 7 0 0 0



Name	Phone	Phone Book properties:
-----	-----	
SHELLY	85271237890	
DONALD	(86)-90018900	
MIKE	39437000	
BETTY	(853) 91239123	
KITTY	852-87654321	
ROCKY	39437000	
ROY	+886 51903481	
Choice: 3		
Name	Phone	
-----	-----	
SHELLY	85271237890	
DONALD	(86)-90018900	
BETTY	(853) 91239123	
KITTY	852-87654321	
ROCKY	39437000	
ROY	+886 51903481	
Choice: 4		
Name	Phone	
-----	-----	
SHELLY	85271237890	
DONALD	(86)-90018900	
BETTY	(853) 91239123	
KITTY	852-87654321	
ROCKY	39437000	
ROY	+886 51903481	
Choice: 0		

Phone Book properties:

Names are all stored in UPPER CASE, and may contain non-alphabet symbols.

Phone numbers may contain non-digit symbols.

Allow entries with same phone number, like MIKE and ROCKY.

When user adds a "new" contact with name colliding with an existing one, it means updating the existing one with a new phone number.

If user clicks Cancel, No or Close dialogue buttons, it means "regret", so do nothing and return to main menu.

If any name or phone text input is **null** or **isBlank()**, do nothing and return to main menu.

Every time when showing the main menu dialog, output also the current phone book on console output. Echo also the choice on console output.

On calling a number, play the digit tone sound and echo the digit on console output one-by-one.

When it calls a number, it does not *only* print the phone digits one-by-one, it also plays the DTMF number tone (MP3).

Procedure (Step-by-Step):

1. Plug a pair of speakers or headset to the computer you use. Make sure it sounds!
2. Create a new NetBeans Java Application project **PhoneBookCaller**. Apply suggested package name **phonebookcaller** and main class **PhoneBookCaller**.
3. Download a third party MP3 processing package **JLayer 1.0.1** and extract **j11.0.1.jar** from <http://www.javazoom.net/javayer/sources/jlayer1.0.1.zip>
4. Download given **PhoneBookCallerResources.zip** from Blackboard.
5. Under OS file manager, copy the Java library file **j11.0.1.jar**, the given MP3 folder **DTMF_DialTone_MP3** and the given GIF file **question-mark...gif** to your NetBeans project folder **PhoneBookCaller**. You need NOT create a lib\ folder. Save them to somewhere like **C:\Users\your_name\Documents\NetBeansProjects\PhoneBookCaller\.**
6. Under NetBeans, Add JAR/Folder **j11.0.1.jar** to your project library.
7. Your new Java main class **PhoneBookCaller** imports/ makes use of a few packages and classes:

```
package phonebookcaller;
```

```
import java.util.HashMap;           // for phone book data management
import javax.swing.*;               // for GUI dialogues and image handling
import java.io.*;                   // for file opening and stream handling
import javazoom.jl.player.advanced.AdvancedPlayer; // for MP3 stream playing
```

8. Clear ALL contact means resetting the phone book to the initial state, with these six entries:

SHELLY	85271237890
DONALD	(86)-90018900
BETTY	(853) 91239123
KITTY	852-87654321
ROCKY	39437000
ROY	+886 51903481

9. Save ALL contact means saving the current phone book to a plain text file "**phonebook.txt**" which should usually reside in your NetBeans application project folder by default, in the following format:

```
SHELLY
85271237890
Name2
Phone2
Name3
Phone3
...
...
ROY
+886 51903481
```

10. Your class **PhoneBookCaller** should include at least the following fields and methods in addition to some other given methods:

```
private static String dialogQuestionMarkIconImageFilename = "question-mark-1451232961b8W_from_publicdomainpictures.net_en_images=142577.gif";

private HashMap<String, String> phoneBook; // a String-String mapping type

private void clearAndSetupDefaultPhoneBook()
{
    // drops, if any, existing phone book and creating a new one
    phoneBook = new HashMap<>();
    addContact("Rocky", "39437000");
    addContact("Kitty", "852-87654321");
    addContact("Betty", "(853) 91239123");
    addContact("Donald", "(86)-90018900");
    addContact("Roy", "+886 51903481");
    addContact("Shelly", "85271237890");
}

/**
 * Show all contacts in the phone book on System.out in some order AND
 * @return a String representation of the phone book in a HTML table
 */
public String showAllContacts()
{
    String table = "<table>";
    table += "<tr>";
    table += "<td>Name<td>Phone";
    table += "</tr>";
```

```

// for-each: iterating all records in the phoneBook which is a HashMap
for (String name : phoneBook.keySet()) {
    // String name will loop through all names (keys)
    String phone = phoneBook.get(name);
    table += "<tr>";
    table += "<td>" + name + "<td>" + phone;
    table += "</tr>";
}
table += "</table><hr>";

System.out.printf("%-20s %s\n", "Name", "Phone");
System.out.printf("-----\n");

/** student's work here to print the phone book on System.out **/

return table; // a String representation of the phone book in HTML
}

/**
 * Show a menu of choices and get user's input
 * @return an integer value: 0 means Quit, and options 1, 2, ...
 */
public int showMenu()
{
    String menuHTML = "<html>";
    menuHTML += showAllContacts();
    menuHTML += "Please pick an action:<hr>";
    menuHTML += "0. Quit<br>";
    menuHTML += "1. Add contact<br>";
    menuHTML += "2. Make a call<br>";
    menuHTML += "3. Clear ALL contact<br>";
    menuHTML += "4. Save ALL contact<br>";
    menuHTML += "</html>";
    String[] options = {"Quit", "Add", "Call", "Clear", "Save"};

    ImageIcon icon = new ImageIcon(dialogQuestionMarkIconImageFilename);
    int choice = JOptionPane.showOptionDialog(null, menuHTML,
this.getClass().getSimpleName(), 0, 0, icon, options, null);
    if (choice == JOptionPane.CLOSED_OPTION) // CLOSED_OPTION = -1
        choice = 0;
    System.out.println("Choice: " + choice);
    return choice;
}

public void addContact(String name, String phone) {...}

public void call(String name) {
    // When calling a number, "dial" each digit by playing the corresponding given MP3 recording file:
    "DTMF_DialTone_MP3/DTMF-n.mp3" where n is a digit in 0 – 9.
}

public void savePhoneBook(String filename) {...}

public static void playMP3File(String filename) {
    // try-catch...
    FileInputStream mp3Stream = new FileInputStream(filename);
    AdvancedPlayer mp3Player = new AdvancedPlayer(mp3Stream);
    mp3Player.play();
}

```

```
public static void main(String[] args)
{
    PhoneBookCaller pbc = new PhoneBookCaller();
    // ...
}
```

11. Your **main()** method should do the followings:

- a) Play the given sound file “**DialTone.mp3**” using a static method **playMP3File()**.
- b) Create a new **PhoneBookCaller** object.
- c) Repeatedly **showMenu()** and process user requests.

Note that file operations and MP3 playing may fail. Thus methods dealing with such operations should try-catch and silent all related Exceptions (i.e., to avoid run-time failure and do not propagate.)

Your Task:

1. Locate your NetBeans project folder, e.g.,

C:\Users\your_name\Documents\NetBeansProjects\PhoneBookCaller\.

Your project folder should contain also Java library **jl1.0.1.jar**, a given image file **question-mark...gif** and a given sub-folder of MP3 files in **DTMF_DialTone_MP3**.

2. ZIP the whole NetBeans project folder **PhoneBookCaller** and Submit the file **PhoneBookCaller.zip** via our online Assignment Collection Box on Blackboard.

Marking Scheme and Notes:

1. The submitted program should be free of any typing mistakes, compilation errors and warnings.
2. Comment/remark, indentation, style are under assessment in every programming assignments unless specified otherwise. Variable naming, proper indentation for code blocks and adequate comments are important. Include also your personal particulars and your academic honesty declaration statement in a header comment block of each source file.
3. Remember to upload your submission and **click Submit** before 18:00 p.m. of the due date. No late submission would be accepted.
3. If you submit multiple times, ONLY the content and time-stamp of the latest one would be counted. You may delete (i.e. take back) your attached file and re-submit. We ONLY take into account the last submission.

University Guideline for Plagiarism:

Attention is drawn to University policy and regulations on honesty in academic work, and to the disciplinary guidelines and procedures applicable to breaches of such policy and regulations.

Details may be found at <http://www.cuhk.edu.hk/policy/academichonesty/>.

With each assignment, students are required to submit a statement that they are aware of these policies, regulations, guidelines and procedures, in a header comment block.

Faculty of Engineering Guidelines to Academic Honesty:

MUST read: <https://www.erg.cuhk.edu.hk/erg/AcademicHonesty>

(you may need to access via CUHK campus network/ CUHK1x/ CUHK VPN)

Further Possibilities:

1. Can you create a computer digital piano using the techniques learnt in this assignment?
2. The JLayer library is not just a MP3 player. It also provides audio format conversion facilities! Find its documentation on the web.
3. Use the NetBeans Navigator window wisely. Point to some methods in the list and leave your mouse pointer there. Wait and see what appears.