Contents

[Mobile Phone 1](#_Toc405547061)

[Requirements 1](#_Toc405547062)

[Class Diagrams 1](#_Toc405547063)

[Mobile.java 3](#_Toc405547064)

[Contact.java 18](#_Toc405547065)

[Javadoc 20](#_Toc405547066)

# 

# Mobile Phone

## Requirements

The objective of this project is to create a Mobile phone emulator in java. The phone is to include several functions such as a screen, keypad, exit button etc...

The phone will includes several JComponents such as JTextArea, JButton, JLabel and JOptionPane. The device will also the user to add contacts and save them to memory using a LinkedList. The contacts can then be read back onto the screen. The screen can then be cleared easily with the clear button. The contact class is an instantiable class where contact objects can be created

## Class Diagrams



## Mobile.java

//Mobile.java

/\*\*Colin Curran Multimedia T00058011\*\*/

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.util.\*;

public class Mobile extends JFrame implements ActionListener

{

private LinkedList <Contact> contacts = new LinkedList<Contact>();

private LinkedList <Messages> messages = new LinkedList<Messages>();

private JTextArea screen;

private JPanel jp,jp1,jp2,jp3,jp4;

private final String Main\_Title = "Mobile";

//adding contacts

private String name,number,text;

private int choice;

private JMenuBar menuBar;

JMenuItem menuItem;

private JButton Button1 = new JButton("1");

private JButton Button2 = new JButton("2");

private JButton Button3 = new JButton("3");

private JButton Button4 = new JButton("4");

private JButton Button5 = new JButton("5");

private JButton Button6 = new JButton("6");

private JButton Button7 = new JButton("7");

private JButton Button8 = new JButton("8");

private JButton Button9 = new JButton("9");

private JButton Button0 = new JButton("0");

private JButton ButtonStar = new JButton("\*");

private JButton ButtonHash = new JButton("#");

private JButton clearButton = new JButton("C");

private JButton callButton = new JButton(new ImageIcon("callB.jpg"));

private JButton exitButton = new JButton("Exit");

Font font = new Font("Verdana", Font.BOLD, 12);

public Mobile()

{

/\*Constructor

JFrame Attributes\*/

setTitle(Main\_Title);

setSize(180,420);

setLocationRelativeTo(null);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setResizable(false);

setVisible(true);

/\*\*http://java-demos.blogspot.ie/2013/10/3-ways-to-set-icon-image-for-jframe.html\*\*/

setIconImage(new ImageIcon("images.jpeg").getImage());

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

menuBar = new JMenuBar();

//associates JMenu to JFrame

setJMenuBar(menuBar);

//FlowLayout/TextArea

setLayout(new FlowLayout());

createMenus();

screen = new JTextArea(10,12);

screen.setLineWrap(true);//Ref John Brosnan :-)

add(screen);

//font etc..

screen.setFont(font);

screen.setForeground(Color.BLUE);

jPanels();

keyPad();

Container contentPane = getContentPane();

contentPane.setBackground(Color.lightGray);

}//end contructor

public static void main(String args[])

{

//JFrame

Mobile frame = new Mobile();

frame.setVisible(true);

}//end main

public void keyPad()

{

//Buttons 1-9

Button1.addActionListener(this);

jp.add(Button1);

Button2.addActionListener(this);

jp.add(Button2);

Button3.addActionListener(this);

jp.add(Button3);

Button4.addActionListener(this);

jp1.add(Button4);

Button5.addActionListener(this);

jp1.add(Button5);

Button6.addActionListener(this);

jp1.add(Button6);

Button7.addActionListener(this);

jp2.add(Button7);

Button8.addActionListener(this);

jp2.add(Button8);

Button9.addActionListener(this);

jp2.add(Button9);

ButtonStar.addActionListener(this);

jp3.add(ButtonStar);

Button0.addActionListener(this);

jp3.add(Button0);

ButtonHash.addActionListener(this);

jp3.add(ButtonHash);

//Clear

clearButton.addActionListener(new ClearButtonListener());

clearButton.setBackground(Color.RED);

jp4.add(clearButton);

//Call

/\*\*\*\*https://docs.oracle.com/javase/7/docs/api/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)\*\*/

callButton.addActionListener(new CallButtonListener());

callButton.setPreferredSize(new Dimension(30,30));

jp4.add(callButton);

///\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Exit

exitButton.addActionListener(new ExitEventHandler());

exitButton.setBackground(Color.ORANGE);

jp4.add(exitButton);

}

public void jPanels()

{

//JPanels

jp = new JPanel();

jp.setBackground(Color.lightGray);

add(jp);

//Row 1

jp1 = new JPanel();

jp1.setBackground(Color.lightGray);

add(jp1);

//Row2

jp2 = new JPanel();

jp2.setBackground(Color.lightGray);

add(jp2);

//Row3

jp3 = new JPanel();

jp3.setBackground(Color.lightGray);

add(jp3);

//Row4

jp4 = new JPanel();

jp4.setBackground(Color.lightGray);

add(jp4);

add(jp4);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==Button1)

{

screen.append("1");

}

else if(e.getSource()==Button2)

{

screen.append("2");

}

else if(e.getSource()==Button3)

{

screen.append("3");

}

else if(e.getSource()==Button4)

{

screen.append("4");

}

else if(e.getSource()==Button5)

{

screen.append("5");

}

else if(e.getSource()==Button6)

{

screen.append("6");

}

else if(e.getSource()==Button7)

{

screen.append("7");

}

else if(e.getSource()==Button8)

{

screen.append("8");

}

else if(e.getSource()==Button9)

{

screen.append("9");

}

else if(e.getSource()==Button0)

{

screen.append("0");

}

else if(e.getSource()==ButtonStar)

{

screen.append("\*");

}

else if(e.getSource()==ButtonHash)

{

screen.append("#");

}

}//end actionPerformed

//ClearButton Listener

public void clearDisplay()

{

//sets screen to blank!

screen.setText(" ");

}

private class ClearButtonListener implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

clearDisplay();

}

}

//Calling Listener

public void calling()

{

//says out of credit

screen.setText("Out of Credit " +

"\nYou need to top up!");

//sets color to red

screen.setForeground(Color.RED);

}

private class CallButtonListener implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

calling();

}

}

//Exit Listener

private class ExitEventHandler implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

exit();

}

}

//Exit Button

public void exit()

{

choice = JOptionPane.showConfirmDialog(null,"Do you wish to exit? ","Power Off",JOptionPane.YES\_OPTION);

if(choice == JOptionPane.YES\_OPTION){

System.exit(0);

}

}

//Creating Navigation Menus

public void createMenus()

{

//JMenu Objects are created

JMenu file = new JMenu("Contacts");

JMenu messages = new JMenu("Messages");

JMenu info = new JMenu("Info");

//adds the option to menu bar

menuBar.add(file);

menuBar.add(messages);

menuBar.add(info);

//adds menu items to file in menu bar

JMenuItem add1 = new JMenuItem("Add");

//adds option under file option

file.add(add1);

add1.addActionListener(new NavButtonListener());

//adds serarator line

file.addSeparator();

//view contacts option

JMenuItem view = new JMenuItem("View");

file.add(view);

view.addActionListener(new NavButtonListener());

//Messages

JMenuItem create = new JMenuItem("Create");

messages.add(create);

create.addActionListener(new NavButtonListener());

messages.addSeparator();

JMenuItem send = new JMenuItem("Send");

messages.add(send);

send.addActionListener(new NavButtonListener());

JMenuItem about = new JMenuItem("About");

info.add(about);

about.addActionListener(new NavButtonListener());

}

private class NavButtonListener implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

if(e.getActionCommand().equals("Add"))

{

name = JOptionPane.showInputDialog(null,"Please enter the name ","Add Contacts!",JOptionPane.INFORMATION\_MESSAGE);

number = JOptionPane.showInputDialog(null,"Please enter the number ","Add Contacts!",JOptionPane.INFORMATION\_MESSAGE);

try{

//creates contact object

Contact c = new Contact(name,number);

contacts.add(c);

//saving file

FileOutputStream f = new FileOutputStream("contacts.dat");

ObjectOutputStream o = new ObjectOutputStream(f);

o.writeObject(contacts);

o.close();

}

catch(Exception ex){

System.out.println(ex.getMessage());

}//end catch

}

else if(e.getActionCommand().equals("View"))

{

try{

//reading file

File f = new File("contacts.dat");

FileInputStream fis = new FileInputStream(f);

ObjectInputStream ois = new ObjectInputStream(fis);

contacts = (LinkedList <Contact>)ois.readObject();

}//end try

catch(Exception ex){

System.out.println(ex.getMessage());

}//end catch

for(Contact c: contacts)

{

//Displays contact on screen of phone

screen.setText(c.toString());

}

}

//writing a message

else if(e.getActionCommand().equals("Create"))

{

text = JOptionPane.showInputDialog(null,"Please write your message ","Create Message",JOptionPane.INFORMATION\_MESSAGE);

try{

//creating message object

Messages m = new Messages(text);

messages.add(m);

FileOutputStream f = new FileOutputStream("messageFile.dat");

ObjectOutputStream o = new ObjectOutputStream(f);

o.writeObject(messages);

o.close();

}

catch(Exception ex){

System.out.println(ex.getMessage());

}//end catch

}

//send message

else if(e.getActionCommand().equals("Send"))

{

JOptionPane.showMessageDialog(null,"Insufficient Credit","Top Up Required",JOptionPane.WARNING\_MESSAGE,new ImageIcon("callB.jpg"));

}

else if(e.getActionCommand().equals("About"))

{

JOptionPane.showMessageDialog(null,"Copyright Colin Curran © 2014","About",JOptionPane.INFORMATION\_MESSAGE,new ImageIcon("info.jpg"));

}

} //End ActionPerformed Method

}//end NavButtonListener

}

//end Mobile class

## Contact.java

/\*\*This is an instantiable Contact class.

@author Colin Curran

@version 1.0 \*/

import java.io.\*;

import java.util.\*;

public class Contact implements Serializable

{

//attributes

private String name;

private String number;

/\*\* accessor method to return the student name

\*@return the name of the student \*/

public String getName() {return name;}

public String getNumber() {return number;}

/\*\* mutator method to set the student name

\*@param name the name of the student \*/

public void setName(String name)

{

this.name=name;

}

public void setNumber(String number)

{

this.number=number;

}

/\*\* Two-args constructor, to create a contact

@param name and number of contact \*/

public Contact(String name,String number)

{

setName(name);

setNumber(number);

}

/\*\* no argument constructor method

@param name the default details of the contact \*/

public Contact()

{

this("Not Givin","0");

}

/\*\* toString method to return the contact details

\*@return the name and number of the contact as a string \*/

public String toString()

{

return getName() + "\n" + getNumber();

}

}//end class

## Javadoc



