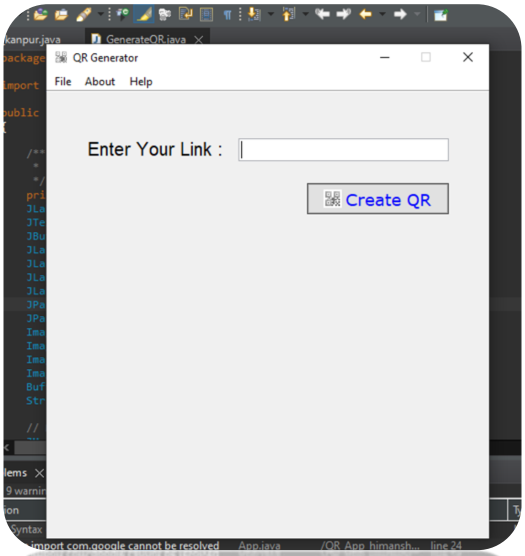
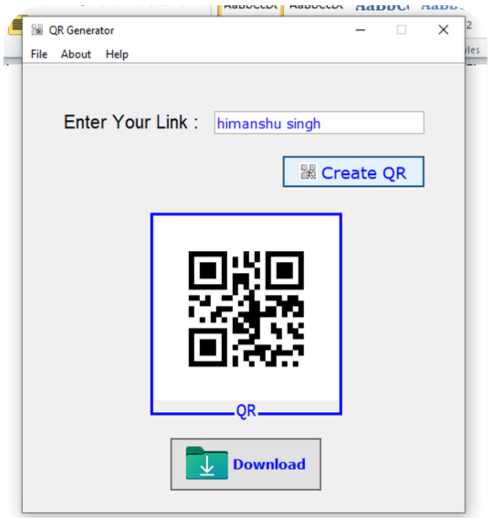
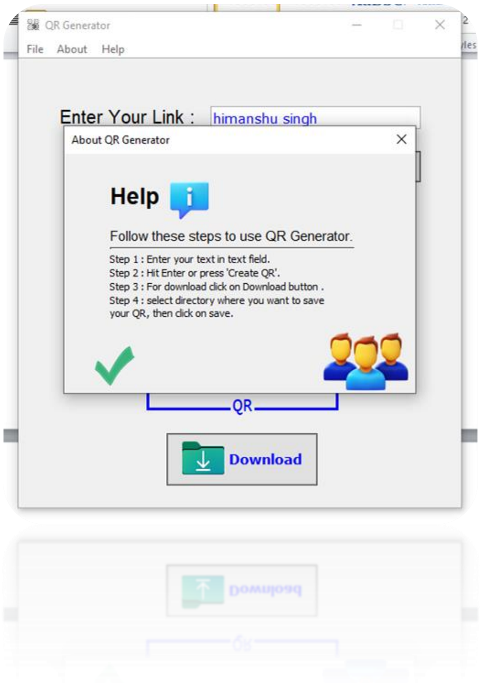
Project: <u>**QR_GENERATOR**</u>

Purpose: “Generate QR CODE FOR ANY TEXT OR LINK AND SAVE IT AS IMAGE.”





Enter Your Link : himanshu singh



Follow these steps to use QR Generator.

Step 1 : Enter your text in text field.
Step 2 : Hit Enter or press 'Create QR'.
Step 3 : For download click on Download button .
Step 4 : select directory where you want to save your QR, then click on save.

This project is developed by SVInfotech for educational purpose.

We have used Eclipse IDE to develop this Project.

Package name: `project.QRproj`

Classes used: GenerateQR.JAVA

ABOUT.JAVA

HELP.JAVA

LIBRARY USED : COM.ZXING LIBRARY

<u>SOURCE CODE: GENERATEQR.JAVA</u>

```
package project.QRproj;


import java.awt.Color;

import java.awt.Desktop;

import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.net.URISyntaxException;
import java.nio.file.Files;
import java.nio.file.StandardCopyOption;
import java.util.HashMap;
import java.util.Map;

import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JDialog;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.KeyStroke;

import javax.swing.UIManager;

import javax.swing.UnsupportedLookAndFeelException;

import javax.swing.border.TitledBorder;


import com.google.zxing.BarcodeFormat;

import com.google.zxing.EncodeHintType;

import com.google.zxing.MultiFormatWriter;

import com.google.zxing.WriterException;

import com.google.zxing.client.j2se.MatrixToImageWriter;

import com.google.zxing.common.BitMatrix;

import com.google.zxing.qrcode.decoder.ErrorCorrectionLevel;


public class GenerateQR extends JFrame

{


	/**

	 *

	 */

	private static final long serialVersionUID = 1L;
```

```
JLabel label;
JTextField text;
JButton qrBtn;
JLabel QRlabel;
JLabel imgLabel;
JLabel inputWarning;
JLabel savedLabel;
JPanel savedPanel;
JPanel qrPanel;
ImageIcon downloadIcon = new ImageIcon("images/download.png");
ImageIcon qrImage = new ImageIcon("images/getQRBtn.png");
ImageIcon savedImage = new ImageIcon("images/saved.png");
ImageIcon qrIcon ;
BufferedImage brImg;
String savedFilePath;

// MENU
JMenuBar menuBar;

JMenu fileMenu;
JMenu aboutMenu;
JMenu helpMenu;

JMenuItem newItem;
JMenuItem saveItem;
JMenuItem exitItem;
public static JMenuItem aboutItem;
public static JMenuItem helpItem;

// MENU DONE


JButton downloadBtn;
boolean downloadable = false;
boolean downloaded = false;
boolean readyToSave = false;

JDialog aboutDialog;



public GenerateQR() {


    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setLayout(null);
    this.setResizable(false);
    this.setTitle("QR Generator");
    ImageIcon logo  = new ImageIcon("images/qrIcon.jpg");
    this.setIconImage(logo.getImage());
    this.setBounds(300,100,500,550);
    // ---------------- ctrl+ s ----------------------------
```

```
        this.getRootPane().registerKeyboardAction(new ActionListener() {

            public void actionPerformed(ActionEvent e) {
                if(downloadable && (inputWarning.getText().isEmpty())) {
                    downloadQR();
                }
            }
        }, KeyStroke.getKeyStroke(KeyEvent.VK_S, KeyEvent.CTRL_DOWN_MASK),
JComponent.WHEN_IN_FOCUSED_WINDOW);

        // ----------------------------------------------------

        // MENU
        menuBar = new JMenuBar();

        fileMenu = new JMenu(" File");
        aboutMenu = new JMenu(" About");
        helpMenu = new JMenu(" Help ");

        newItem = new JMenuItem("   New");
        newItem.setIcon(new ImageIcon("images/newFile.png"));
        newItem.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {
```

```
        text.setText("");
        text.revalidate();
        text.repaint();

        inputWarning.setText("");
        saveItem.setEnabled(false);

        qrPanel.removeAll();
        qrPanel.revalidate();
        qrPanel.repaint();

        savedPanel.removeAll();
        savedPanel.revalidate();
        savedPanel.repaint();

    }
});

saveItem = new JMenuItem("   Save");
saveItem.setIcon(new ImageIcon("images/save.png"));
saveItem.setEnabled(false);
saveItem.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
```

```
                downloadQR();

        }
});


exitItem = new JMenuItem("   Exit");
exitItem.setIcon(new ImageIcon("images/exit.png"));
exitItem.addActionListener(new ActionListener() {

        public void actionPerformed(ActionEvent e) {
                System.exit(0);
        }
});


aboutItem = new JMenuItem("About");
aboutItem.setIcon(new ImageIcon("images/setting.png"));
aboutItem.addActionListener(new ActionListener() {

        public void actionPerformed(ActionEvent e) {

                try {
                        new About();
                }catch(URISyntaxException es) {

                }
        }
});

helpItem = new JMenuItem("Help");
helpItem.setIcon(new ImageIcon("images/help.png"));
helpItem.addActionListener(new ActionListener() {

        public void actionPerformed(ActionEvent e) {

                new Help();

        }
});



fileMenu.add(newItem);
fileMenu.add(saveItem);
fileMenu.add(exitItem);

aboutMenu.add(aboutItem);
helpMenu.add(helpItem);
```

```
menuBar.add(fileMenu);
menuBar.add(aboutMenu);
menuBar.add(helpMenu);

// MENU FINISHED



qrPanel = new JPanel();
qrPanel.setBounds(95, 145,300, 350);
qrPanel.setLayout(new FlowLayout(FlowLayout.CENTER, 1, 15));

label = new JLabel();
label.setText("Enter Your Link : ");
label.setFont(new Font("Arial", Font.PLAIN,20));
label.setBounds(45, 55, 200, 17);

text = new JTextField();
text.setFont(new Font("Verdana", Font.PLAIN,15));
text.setForeground(Color.blue);
text.setBounds(210,51,230,25);
text.setEditable(true);
text.addKeyListener(new KeyListener() {

        public void keyReleased(KeyEvent e) {
                if(e.getKeyCode() == 10) {
```

```
            createAndShowQR();
        }
    }
    public void keyPressed(KeyEvent e) {}
    public void keyTyped(KeyEvent e) {}
});



inputWarning = new JLabel();
inputWarning.setBounds(210,79,230,15);
inputWarning.setForeground(Color.red);
inputWarning.setFont(new Font("Arial",Font.ITALIC, 13));

qrBtn = new JButton("Create QR");
qrBtn.setIcon(qrImage);
qrBtn.setBounds(285,100,155,34);
qrBtn.setFocusable(false);
qrBtn.setToolTipText("Create QR");
qrBtn.setFont(new Font("Verdana",Font.PLAIN, 18));
qrBtn.setBackground(Color.darkGray);
qrBtn.setForeground(Color.blue);
qrBtn.setOpaque(true);
qrBtn.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

                downloaded = false;
                saveItem.setEnabled(true);
                createAndShowQR();
        }
});

downloadBtn = new JButton();
downloadBtn.setText("Download");
downloadBtn.setIcon(downloadIcon);
downloadBtn.setMnemonic('b');
downloadBtn.setToolTipText("Save QR Image");
downloadBtn.setFocusable(false);
downloadBtn.setFont(new Font("Verdana",Font.BOLD, 15));
downloadBtn.setBackground(Color.darkGray);
downloadBtn.setForeground(Color.blue);
downloadBtn.setOpaque(true);
downloadBtn.setBounds(68,245,170,50);
downloadBtn.addActionListener(new ActionListener() {

        public void actionPerformed(ActionEvent e) {
                downloadQR();
        }
});
```

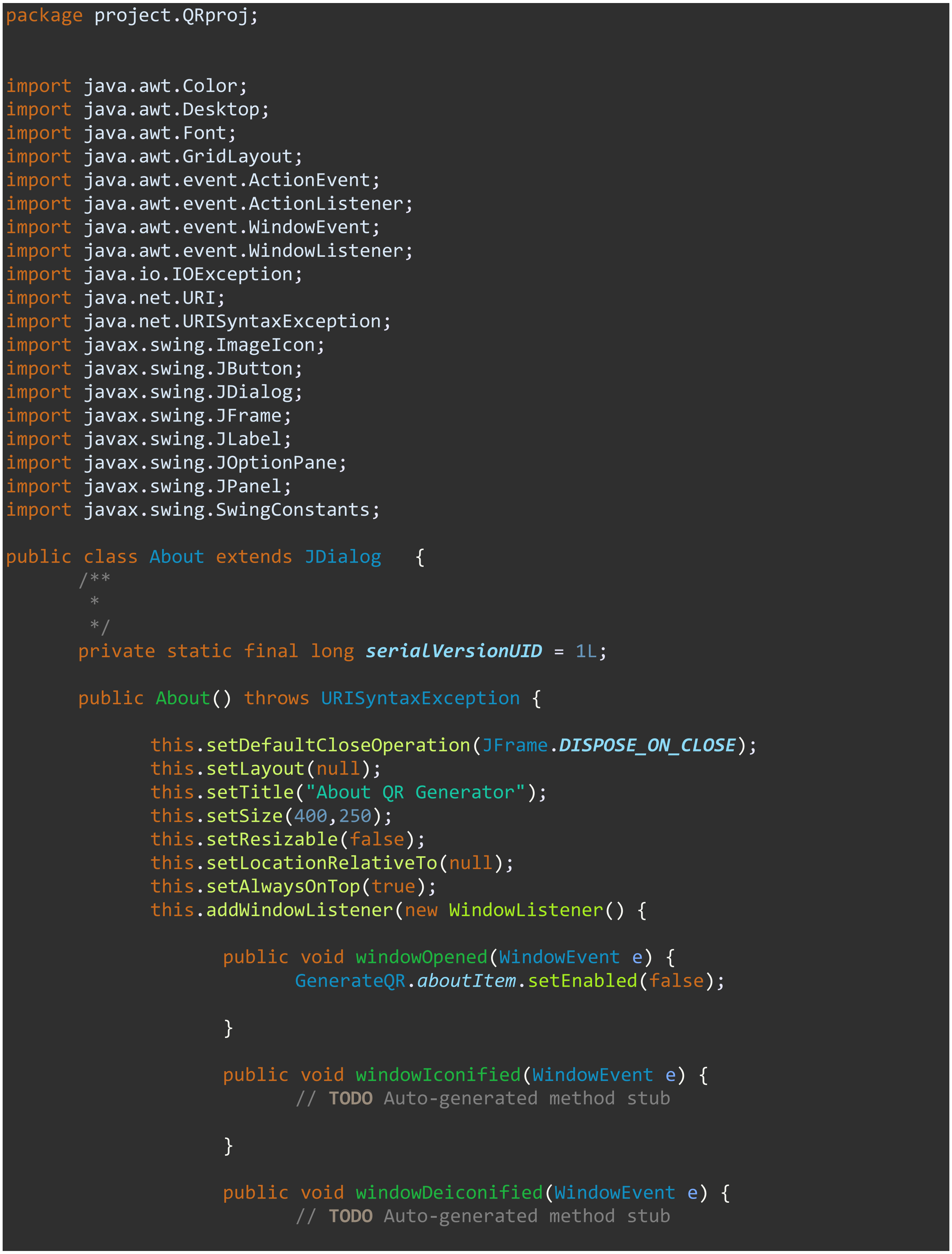
```
            savedLabel = new JLabel("saved");

            savedLabel.setIcon(savedImage);

            savedLabel.setHorizontalTextPosition(JLabel.CENTER);

            savedLabel.setFont(new Font("Verdana",Font.BOLD, 12));

            savedLabel.setVerticalTextPosition(JLabel.BOTTOM);

            savedLabel.setIconTextGap(-2);

            savedLabel.setBounds(0,0,50,70);


            savedPanel = new JPanel();

            savedPanel.add(savedLabel);

            savedPanel.setBounds(390,390,70,90);




            this.add(label);

            this.add(text);

            this.add(qrBtn);

            this.add(inputWarning);

            this.setJMenuBar(menuBar);

            this.setLocationRelativeTo(null);

            this.setVisible(true);


    }


     protected void downloadQR() {
            JFileChooser fileChooser = new JFileChooser("../");

            int response = fileChooser.showSaveDialog(this);


            File source = new File("images/qr.png");
            File dest ;



            if(response == JFileChooser.APPROVE_OPTION) {
                dest = new File(savedFilePath = 
fileChooser.getSelectedFile().getAbsolutePath()+".png");

                try {

    Files.copy(source.toPath(),dest.toPath(),StandardCopyOption.REPLACE_EXISTING);
                    savedPanel.add(savedLabel);
                    this.add(savedPanel);

                    downloaded=true;
                    this.revalidate();
                    this.repaint();
                } catch (IOException e1) {
                    JOptionPane.showMessageDialog(null, "Enter valid 
location","Alert !", JOptionPane.WARNING_MESSAGE);
                }
```

```
        }


}


private void createAndShowQR() {
        String input = text.getText();




        if (input.isEmpty()) {


            savedPanel.removeAll();
            savedPanel.revalidate();
            savedPanel.repaint();


            saveItem.setEnabled(false);
            inputWarning.setText("text field is empty, enter some text.");
            qrPanel.removeAll();
            qrPanel.revalidate();
            qrPanel.repaint();
            return;
        }
        saveItem.setEnabled(true);
        }

        if(!(savedLabel.getText().isEmpty())) {
            this.remove(savedPanel);
            this.revalidate();
            this.repaint();
        }

        if(!(inputWarning.getText().isEmpty())) {
            inputWarning.setText("");
        }



            try {
                brImg = createQR(input);
                showImage(brImg);
                downloadable = true;
            } catch (WriterException e1) {
                e1.printStackTrace();
            } catch (IOException e1) {
                e1.printStackTrace();
            }



}
```

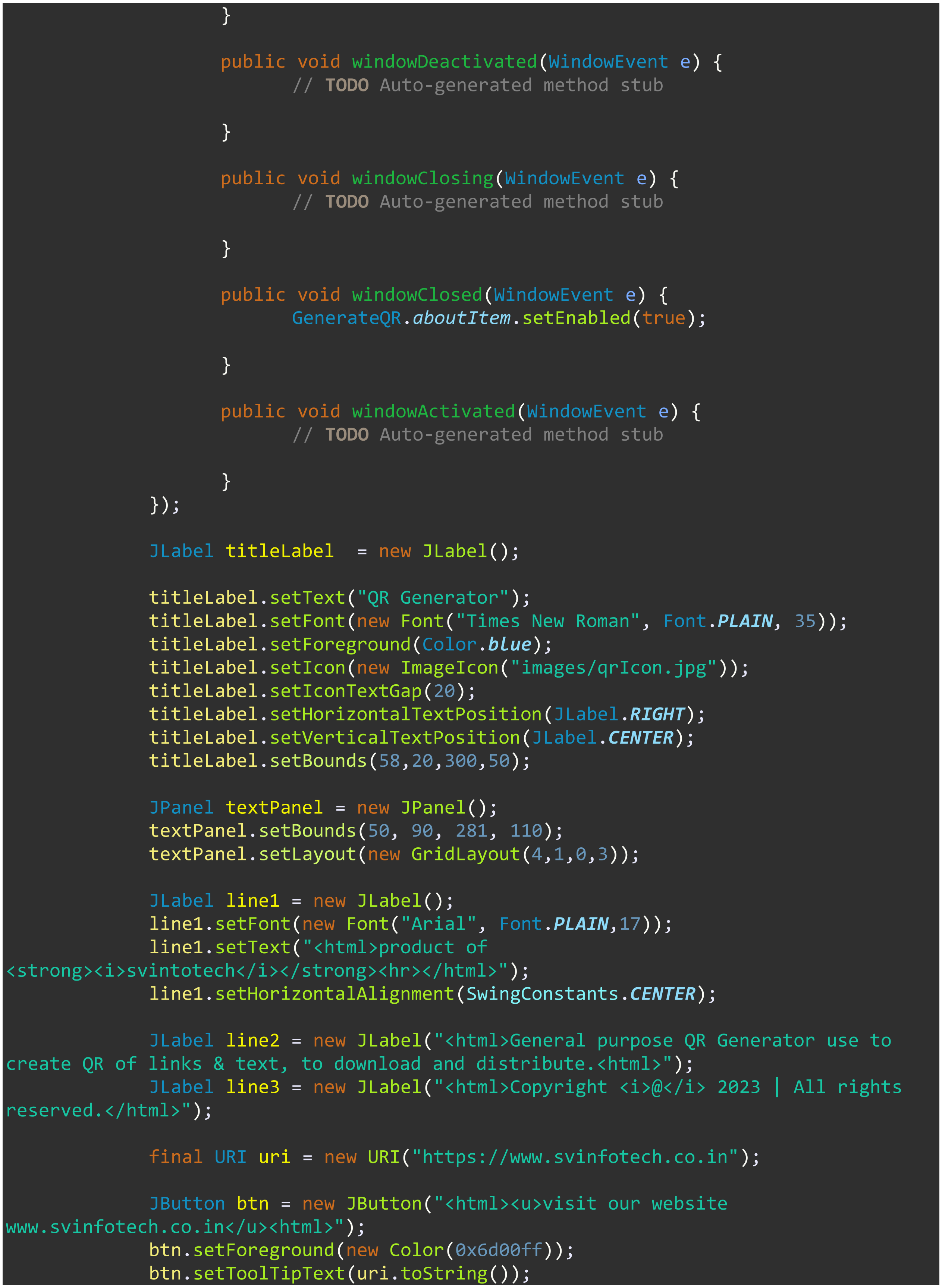
```
        private BufferedImage createQR(String input) throws WriterException, IOException {


                //Encoding charset to be used

                String charset = "UTF-8";


                Map<EncodeHintType, ErrorCorrectionLevel> hashMap = new
HashMap<EncodeHintType, ErrorCorrectionLevel>();


                //generates QR code with Low level(L) error correction capability

                hashMap.put(EncodeHintType.ERROR_CORRECTION, ErrorCorrectionLevel.L);


                BitMatrix matrix = (new MultiFormatWriter()).encode(new
String(input.getBytes(charset), charset), BarcodeFormat.QR_CODE, 200, 200);

                BufferedImage qrImage = MatrixToImageWriter.toBufferedImage(matrix);


                File file = new File("images/qr.png");

                MatrixToImageWriter.writeToPath(matrix, "png", file.toPath());


                return qrImage;

        }


        public void showImage(BufferedImage qrImage) {


                qrIcon = new ImageIcon(qrImage);
                imgLabel = new JLabel();

                qrPanel.removeAll();

                final int qrPositionX = (qrPanel.getWidth()/2)-(qrIcon.getIconWidth()/2);
                final int qrPositionY = (qrPanel.getHeight()/2)-(qrIcon.getIconHeight()/2);

                imgLabel.setIcon(qrIcon);
                imgLabel.setHorizontalAlignment(JLabel.CENTER);
                imgLabel.setVerticalAlignment(JLabel.CENTER);



                imgLabel.setBounds(qrPositionX,qrPositionY - (qrPositionY/2),qrIcon.getIconWidth()+15, qrIcon.getIconHeight()+20);

        imgLabel.setBorder(BorderFactory.createTitledBorder(BorderFactory.createLineBorder(Color.BLUE,3),

                "QR",

                TitledBorder.CENTER ,

                TitledBorder.BOTTOM,

                new Font("consolas", Font.PLAIN, 20),

                Color.BLUE));
                imgLabel.addMouseListener(new MouseListener() {

                    public void mouseReleased(MouseEvent e) {}

                    public void mousePressed(MouseEvent e) {}

                    public void mouseExited(MouseEvent e) {
                        if(downloaded) {


            imgLabel.setBorder(BorderFactory.createTitledBorder(BorderFactory.createLineBorder(Color.BL
UE,3),
                                                    "QR",
                                                    TitledBorder.CENTER ,
                                                    TitledBorder.BOTTOM,
                                                    new Font("consolas", Font.PLAIN, 20),
                                                    Color.BLUE));
                        }
                    }

                    public void mouseEntered(MouseEvent e) {
                        if(downloaded) {

            imgLabel.setBorder(BorderFactory.createTitledBorder(BorderFactory.createLineBorder(Color.G
REEN,3),
                                                    "click to open",
                                    TitledBorder.CENTER ,
                                    TitledBorder.BOTTOM,
                                    new Font("consolas", Font.PLAIN, 20),
                                    Color.BLUE));
                        }
                    }

                    public void mouseClicked(MouseEvent e) {
                        if(e.getButton() == MouseEvent.BUTTON1  && downloaded) {
                            openImage();
                        }

                    }
                });

                qrPanel.add(imgLabel);
                qrPanel.add(downloadBtn);

                this.add(qrPanel);

                qrPanel.revalidate();
                qrPanel.repaint();

            }
```

```
    public void keyTyped(KeyEvent e) {
    }

    public void keyPressed(KeyEvent e) {
    }

    public void keyReleased(KeyEvent e) {
    }

    private void openImage() {

        File file = new File(savedFilePath);
        if(file.exists()) {
            Desktop dt = Desktop.getDesktop();
            try {
                dt.open(file);
            } catch (IOException e) {
                JOptionPane.showMessageDialog(null, "Something went
wrong...!","Error !", JOptionPane.ERROR_MESSAGE);
            }
        }else {

            JOptionPane.showMessageDialog(null, "Image not available at saved
location!","Error !", JOptionPane.ERROR_MESSAGE);
        }
    }

    public static void main(String[] args) {

        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (ClassNotFoundException e) {

        } catch (InstantiationException e) {

        } catch (IllegalAccessException e) {

        } catch (UnsupportedLookAndFeelException e) {

        }

        new GenerateQR();
    }



}
```

SOURCE CODE OF ABOUT.JAVA:

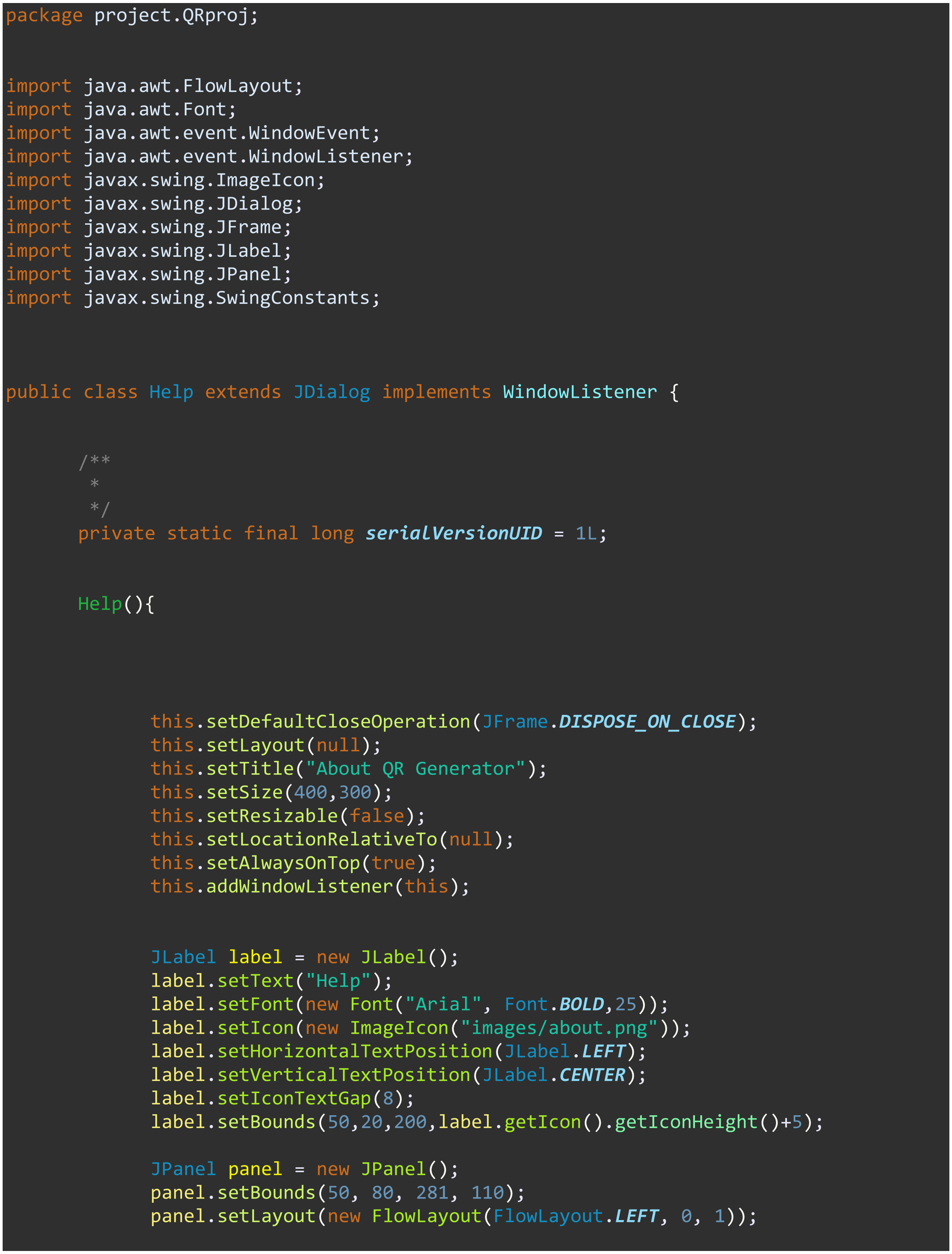


```
package project.QRproj;


import java.awt.Color;
import java.awt.Desktop;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.SwingConstants;

public class About extends JDialog   {
    /**
     *
     */
    private static final long serialVersionUID = 1L;

    public About() throws URISyntaxException {

        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        this.setLayout(null);
        this.setTitle("About QR Generator");
        this.setSize(400,250);
        this.setResizable(false);
        this.setLocationRelativeTo(null);
        this.setAlwaysOnTop(true);
        this.addWindowListener(new WindowListener() {

            public void windowOpened(WindowEvent e) {
                GenerateQR.aboutItem.setEnabled(false);

            }

            public void windowIconified(WindowEvent e) {
                // TODO Auto-generated method stub

            }

            public void windowDeiconified(WindowEvent e) {
                // TODO Auto-generated method stub
```

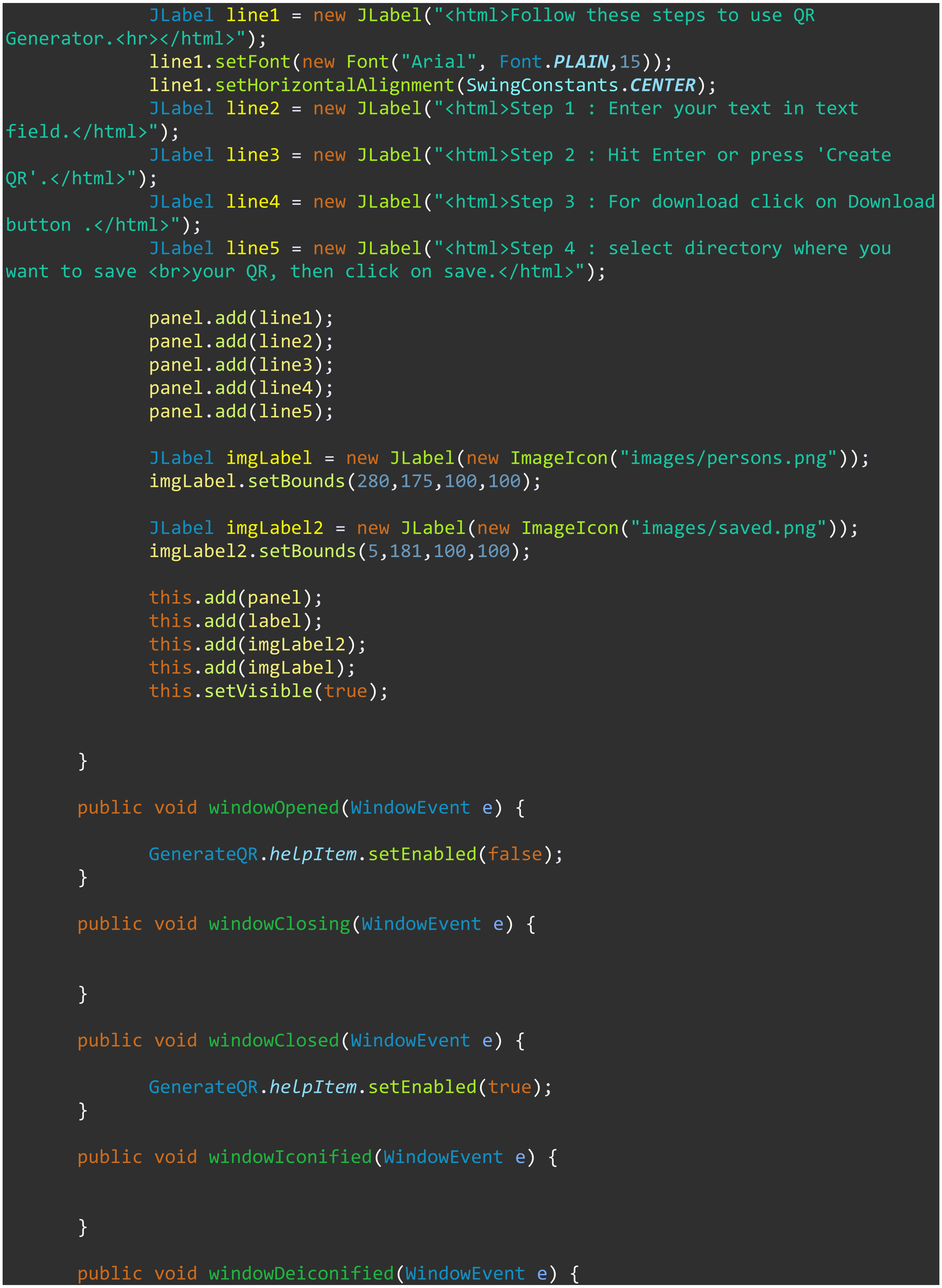


```
            }

            public void windowDeactivated(WindowEvent e) {
                // TODO Auto-generated method stub

            }

            public void windowClosing(WindowEvent e) {
                // TODO Auto-generated method stub

            }

            public void windowClosed(WindowEvent e) {
                GenerateQR.aboutItem.setEnabled(true);

            }

            public void windowActivated(WindowEvent e) {
                // TODO Auto-generated method stub

            }
        });

        JLabel titleLabel  = new JLabel();

        titleLabel.setText("QR Generator");
        titleLabel.setFont(new Font("Times New Roman", Font.PLAIN, 35));
        titleLabel.setForeground(Color.blue);
        titleLabel.setIcon(new ImageIcon("images/qrIcon.jpg"));
        titleLabel.setIconTextGap(20);
        titleLabel.setHorizontalTextPosition(JLabel.RIGHT);
        titleLabel.setVerticalTextPosition(JLabel.CENTER);
        titleLabel.setBounds(58,20,300,50);

        JPanel textPanel = new JPanel();
        textPanel.setBounds(50, 90, 281, 110);
        textPanel.setLayout(new GridLayout(4,1,0,3));

        JLabel line1 = new JLabel();
        line1.setFont(new Font("Arial", Font.PLAIN,17));
        line1.setText("<html>product of
<strong><i>svintotech</i></strong><hr></html>");
        line1.setHorizontalAlignment(SwingConstants.CENTER);

        JLabel line2 = new JLabel("<html>General purpose QR Generator use to
create QR of links & text, to download and distribute.<html>");
        JLabel line3 = new JLabel("<html>Copyright <i>@</i> 2023 | All rights
reserved.</html>");

        final URI uri = new URI("https://www.svinfotech.co.in");

        JButton btn = new JButton("<html><u>visit our website
www.svinfotech.co.in</u><html>");
        btn.setForeground(new Color(0x6d00ff));
        btn.setToolTipText(uri.toString());
        btn.setBorder(null);
        btn.setFocusable(false);
        btn.setOpaque(false);
        btn.setBackground(Color.white);
        btn.setSize(50,50);
        btn.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {
                if(Desktop.isDesktopSupported()) {
                    try {
                        Desktop.getDesktop().browse(uri);
                    } catch (IOException e1) {
                        JOptionPane.showMessageDialog(null, "Unable
to open browser", "Error!", JOptionPane.ERROR_MESSAGE);
                    }
                }

            }
        });

        textPanel.add(line1);
        textPanel.add(line3);
        textPanel.add(line2);
        textPanel.add(btn);

        this.add(textPanel);
        this.add(titleLabel);
        this.setVisible(true);
    }




}
```

SOURCE CODE OF HELP.JAVA

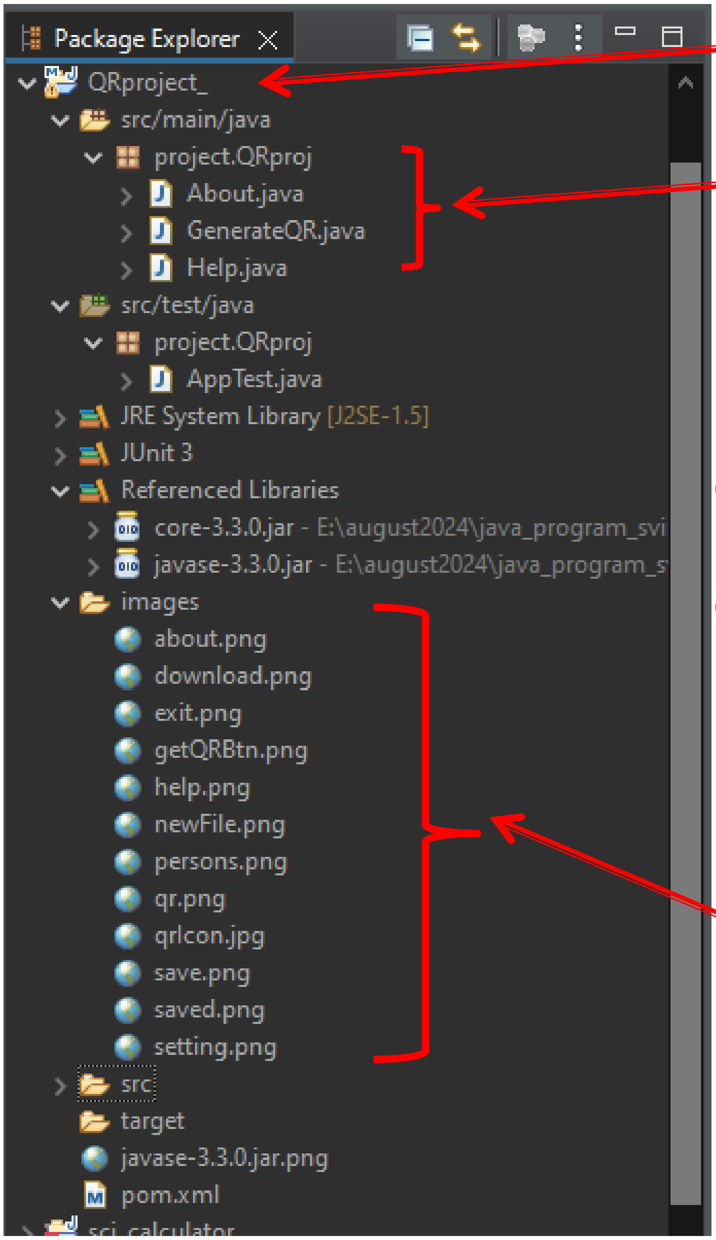


```
package project.QRproj;


import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;
import javax.swing.ImageIcon;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.SwingConstants;



public class Help extends JDialog implements WindowListener {


        /**
         * 
         */
        private static final long serialVersionUID = 1L;


        Help(){




                this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                this.setLayout(null);
                this.setTitle("About QR Generator");
                this.setSize(400,300);
                this.setResizable(false);
                this.setLocationRelativeTo(null);
                this.setAlwaysOnTop(true);
                this.addWindowListener(this);


                JLabel label = new JLabel();
                label.setText("Help");
                label.setFont(new Font("Arial", Font.BOLD,25));
                label.setIcon(new ImageIcon("images/about.png"));
                label.setHorizontalTextPosition(JLabel.LEFT);
                label.setVerticalTextPosition(JLabel.CENTER);
                label.setIconTextGap(8);
                label.setBounds(50,20,200,label.getIcon().getIconHeight()+5);

                JPanel panel = new JPanel();
                panel.setBounds(50, 80, 281, 110);
                panel.setLayout(new FlowLayout(FlowLayout.LEFT, 0, 1));
```



```
                JLabel line1 = new JLabel("<html>Follow these steps to use QR
Generator.<hr></html>");
                line1.setFont(new Font("Arial", Font.PLAIN,15));
                line1.setHorizontalAlignment(SwingConstants.CENTER);
                JLabel line2 = new JLabel("<html>Step 1 : Enter your text in text
field.</html>");
                JLabel line3 = new JLabel("<html>Step 2 : Hit Enter or press 'Create
QR'.</html>");
                JLabel line4 = new JLabel("<html>Step 3 : For download click on Download
button .</html>");
                JLabel line5 = new JLabel("<html>Step 4 : select directory where you
want to save <br>your QR, then click on save.</html>");

                panel.add(line1);
                panel.add(line2);
                panel.add(line3);
                panel.add(line4);
                panel.add(line5);

                JLabel imgLabel = new JLabel(new ImageIcon("images/persons.png"));
                imgLabel.setBounds(280,175,100,100);

                JLabel imgLabel2 = new JLabel(new ImageIcon("images/saved.png"));
                imgLabel2.setBounds(5,181,100,100);

                this.add(panel);
                this.add(label);
                this.add(imgLabel2);
                this.add(imgLabel);
                this.setVisible(true);


        }

        public void windowOpened(WindowEvent e) {

                GenerateQR.helpItem.setEnabled(false);
        }

        public void windowClosing(WindowEvent e) {


        }

        public void windowClosed(WindowEvent e) {

                GenerateQR.helpItem.setEnabled(true);
        }

        public void windowIconified(WindowEvent e) {


        }

        public void windowDeiconified(WindowEvent e) {
        }

        public void windowActivated(WindowEvent e) {


        }

        public void windowDeactivated(WindowEvent e) {

        }




}
```

PROJECT STRUCTURE



Package Explorer

```
QRproject_
  src/main/java
    project.QRproj
      About.java
      GenerateQR.java
      Help.java
  src/test/java
    project.QRproj
      AppTest.java
  JRE System Library [J2SE-1.5]
  JUnit 3
  Referenced Libraries
    core-3.3.0.jar - E:\august2024\java_program_svi
    javase-3.3.0.jar - E:\august2024\java_program_s
  images
    about.png
    download.png
    exit.png
    getQRBtn.png
    help.png
    newFile.png
    persons.png
    qr.png
    qrIcon.jpg
    save.png
    saved.png
    setting.png
  src
  target
  javase-3.3.0.jar.png
  pom.xml
sci_calculator
```

PROJECT PACKAGE NAME

USER DEFINED CLASEES USED IN PROJECT

EXTERNAL JAR FILES USED IN PROJECT

TO ADD GOOGLE ZXING LIBRARY WHEN YOU TEST/RUN THIS PROJECT VIA IDE

YOU HAVE ADD EXTERNAL LIBRARY IN YOUR PROJECT VIA BUILD PROJECT

"YOU CAN FIND IT IN PROJECT MANUAL FOLDER"

EXTERNAL IMAGES USED IN PROJECT