

**TEXTBANK 文本银行软件 V1.0 代码**

```

package View;
import Model.Encrypt;
import Model.GetCounts;
import UserService.User;
import UserService.UserOperator;
import javax.swing.*;
import javax.swing.border.TitledBorder;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
public class LoginView extends JFrame implements ActionListener {
    JButton register;
    JButton login;
    JButton forgetPasswd;
    JButton jButton;
    JLabel name;
    JLabel password;
    JTextField username;
    JPasswordField ipassword;
    User user;
    public String getp() {
        return String.valueOf(ipassword.getPassword());
    }
    public LoginView() {
        name = new JLabel("用户名");
        name.setBounds(76, 166, 120, 30);
        name.setFont(new Font("", 24));
        getContentPane().add(name);
        getContentPane().setLayout(null);
        username = new JTextField();
        username.setBounds(178, 160, 240, 42);
        username.setFont(new Font("", Font.BOLD, 24));
        getContentPane().add(username);
        password = new JLabel("密码");
        password.setBounds(78, 214, 120, 30);
        password.setFont(new Font("", Font.BOLD, 24));
        getContentPane().add(password);
        ipassword = new JPasswordField();
        ipassword.setBounds(178, 208, 240, 42);
        ipassword.setFont(new Font("", Font.BOLD, 24));
        getContentPane().add(ipassword);
        register = new JButton("注册");
        register.setText("注册");
        register.setBounds(282, 262, 136, 46);
        getContentPane().add(register);
        register.setFocusPainted(false);
        login = new JButton("登录");
        login.setText("登录");
        login.setBounds(134, 262, 136, 46);
        getContentPane().add(login);
        login.setFocusPainted(false);
        jButton = new JButton();
        jButton.setBorder(new TitledBorder(null, "", TitledBorder.DEFAULT_JUSTIFICATION,
TitledBorder.DEFAULT_POSITION, null, null));
        jButton.setForeground(new Color(255, 0, 0));
        jButton.setFont(new Font("", Font.BOLD, 36));

```

```

jButton.setText("登录到我的文本银行");
jButton.setBounds(78, 56, 340, 72);
jButton.setBorderPainted(false);
jButton.setFocusPainted(false);
jButton.setContentAreaFilled(false);
getContentPane().add(jButton);
setTitle("登录窗口");
setSize( 560, 420);
getContentPane().setLayout(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ImageIcon imageIcon=new ImageIcon("./lib/TextBank.png");
this.setIconImage(imageIcon.getImage());
this.setLocationRelativeTo(null);
forgetPasswd = new JButton("忘记密码");
forgetPasswd.setBounds(400, 214, 100, 30);
forgetPasswd.setBorderPainted(false);
forgetPasswd.setFocusPainted(false);
forgetPasswd.setContentAreaFilled(false);
forgetPasswd.addActionListener(this::actionPerformed);
getContentPane().add(forgetPasswd);
login.addActionListener(this);
register.addActionListener(this);
setVisible(true);
}
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == register) {
        dispose();
        new RegisterView();
    } else if (e.getSource() == forgetPasswd) {
        dispose();
        new ForgetPasswdView();
    } else if (e.getSource() == login) {
        UserOperator userOperator = new UserOperator();
        try {
            user = userOperator.FindUser(username.getText());
            Encrypt encrypt = new Encrypt();
            String str = encrypt.encode(getp());
            if (user != null) {
                if (user.getId().equals(username.getText()) && user.getPasswd().equals(str)) {
                    dispose();
                    new BankView();
                } else {
                    JOptionPane.showMessageDialog(null, "用户名或密码错误");
                }
            }
        } else {
            JOptionPane.showMessageDialog(null, "用户未注册");
        }
    } catch (SQLException throwables) {
        throwables.printStackTrace();
    }
}
}
public static void main(String[] args) {
    new LoginView();
}
}

```

```

package DataService;
public class Data {
    private int id;
    private String text;
    private String summary;
    public String getSummary() {
        return summary;
    }
    public void setSummary(String summary) {
        this.summary = summary;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getText() {
        return text;
    }
    public void setText(String text) {
        this.text = text;
    }
}

package DataService;
import Model.DBUtil;
import javax.swing.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class DataOperator {
    public Data Find(int id) throws SQLException {
        String sql = "select * from DataSource where ID=?";
        Connection conn = DBUtil.getConn();
        PreparedStatement ps = conn.prepareStatement(sql);
        ps.setInt(1,id);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            Data data=new Data();
            data.setId(rs.getInt("ID"));
            data.setText(rs.getString("CONTENT"));
            data.setSummary(rs.getString("SUMMARY"));
            return data;
        } else{
            return null;
        }
    }
    public void Insert(Data data) throws SQLException {
        String sql="insert into `DataSource`(`ID`, `CONTENT`, `SUMMARY`) "+ "VALUES (?,?,?)";
        Connection conn=DBUtil.getConn();
        PreparedStatement ps=conn.prepareStatement(sql);
        ps.setInt(1,data.getId());
        ps.setString(2,data.getText());
        ps.setString(3,data.getSummary());
        ps.executeUpdate();
        conn.close();
        JOptionPane.showMessageDialog(null, "添加成功");
    }
}

```

```

    }
    public void Update(Data data) throws SQLException {
        String sql="update DataSource SET CONTENT=? where ID=?"+"VALUES (?,?)";
        Connection conn=DBUtil.getConn();
        PreparedStatement ps=conn.prepareStatement(sql);
        ps.setString(1,data.getText());
        ps.setInt(2,data.getId());
        ps.executeUpdate();
        conn.close();
        JOptionPane.showMessageDialog(null, "更新成功");
    }
    public void delete(Data data) throws SQLException, InterruptedException {
        String sql="delete from `DataSource` where CONTENT=?";
        Connection conn=DBUtil.getConn();
        PreparedStatement ps=conn.prepareStatement(sql);
        if (data != null) {
            ps.setString(1,data.getText());
            ps.executeUpdate();
            conn.close();
            Thread.sleep(10);
            String sqlw="delete from `DataSource` where ID=?";
            Connection connw=DBUtil.getConn();
            PreparedStatement psw=connw.prepareStatement(sqlw);
            psw.setInt(1,data.getId());
            psw.executeUpdate();
            conn.close();
            JOptionPane.showMessageDialog(null, "删除成功");
        }
    }
    public int getCount() {
        int count = -1;
        String sql = "select count(*) from DataSource";
        try {
            Connection conn=DBUtil.getConn();
            if (conn!=null) {
                PreparedStatement ps=conn.prepareStatement(sql);
                ResultSet rs=ps.executeQuery();
                if(rs.next()) {
                    count=rs.getInt(1);
                }
            }else{
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return count;
    }
}

package Model;
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.datatransfer.Clipboard;
import java.awt.datatransfer.StringSelection;
import java.awt.datatransfer.Transferable;
public class ClipBoardModel {
    public void setSysClipboardText(String writeMe) {
        Clipboard clip = Toolkit.getDefaultToolkit().getSystemClipboard();
        Transferable tText = new StringSelection(writeMe);
    }
}

```

```

        clip.setContents(tText, null);
        JOptionPane.showMessageDialog(null, "已复制到剪切板");
    }
}
package Model;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.Properties;
import javax.sql.DataSource;
import org.apache.commons.dbcp.BasicDataSourceFactory;
public class DBUtil {
    private static DataSource ds;
    private static Connection conn;
    static {
        try {
            Properties pro = new Properties();
            InputStream is = DBUtil.class.getResourceAsStream("/sqlite.properties");
            pro.load(is);
            ds = BasicDataSourceFactory.createDataSource(pro);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    public static Connection getConn() throws SQLException {
        if (conn == null || conn.isClosed()) {
            conn = ds.getConnection();
        }
        return conn;
    }
    public static void closeConn() throws SQLException {
        try {
            if (conn != null && !conn.isClosed()) {
                conn.close();
            }
        } finally {
            conn = null;
        }
    }
    private DBUtil() {
    }
}
package Model;
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
public class Encrypt {
    private static final String key = "~!@#$%^&*()<>?/.,\";
    private static final Charset charset = StandardCharsets.UTF_8;
    private static final byte[] bytes = key.getBytes(charset);
    public String encode(String enc) {
        byte[] encBytes = enc.getBytes(charset);
        for (int i = 0, size = encBytes.length; i < size; i++) {
            for (byte keyBytes0 : bytes) {
                encBytes[i] = (byte) (encBytes[i] ^ keyBytes0);
            }
        }
        return new String(encBytes);
    }
}

```

```

}
package Model;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
public class GetCounts {
    public int getCount() {
        int count = 0;
        PreparedStatement ps = null;
        ResultSet rs = null;
        String sql = "select count(*) from DataSource";
        try {
            Connection con=DBUtil.getConn();
            if (con!=null) {
                ps=con.prepareStatement(sql);
                rs=ps.executeQuery();
                if(rs.next()) {
                    count=rs.getInt(1);
                }
            }
        } catch (Exception e) {}
        return count;
    }
}
package Model;
import javax.swing.*.*;
public class ListModel extends AbstractListModel implements ComboBoxModel{
    private String[] choose={"纪念日","重要的一天","旅行过的城市","我的好朋友"};
    private String selection=null;
    @Override
    public int getSize() { return choose.length; }
    @Override
    public Object getElementAt(int index) {
        return choose[index];
    }
    @Override
    public void setSelectedItem(Object anItem) {
        selection=(String)anItem;
    }
    @Override
    public Object getSelectedItem() {
        return selection;
    }
}
package UserService;
public class User {
    private String id;
    private String passwd;
    private String question;
    private String answer;
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getPasswd() {
        return passwd;
    }
}

```

```

    }
    public void setPasswd(String passwd) {
        this.passwd = passwd;
    }
    public String getQuestion() {
        return question;
    }
    public void setQuestion(String question) {
        this.question = question;
    }
    public String getAnswer() {
        return answer;
    }
    public void setAnswer(String answer) {
        this.answer = answer;
    }
}
package UserService;
import Model.DBUtil;
import javax.swing.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class UserOperator {
    public User FindUser(String str) throws SQLException {
        String sql = "select * from userdata where ACCOUNT=?";
        Connection conn = DBUtil.getConn();
        PreparedStatement ps = conn.prepareStatement(sql);
        ps.setString(1, str);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            User user = new User();
            user.setId(rs.getString("ACCOUNT"));
            user.setPasswd(rs.getString("PASSWD"));
            user.setQuestion(rs.getString("QUESTION"));
            user.setAnswer(rs.getString("ANSWER"));
            return user;
        } else {
            return null;
        }
    }
    public void Insert(User user) throws SQLException {
        String sql="insert into `userdata`(`ACCOUNT`,`PASSWD`,`QUESTION`,`ANSWER`) "+"VALUES (?,?,?,?)";
        Connection conn=DBUtil.getConn();
        PreparedStatement ps=conn.prepareStatement(sql);
        ps.setString(1,user.getId());
        ps.setString(2,user.getPasswd());
        ps.setString(3,user.getQuestion());
        ps.setString(4,user.getAnswer());
        ps.executeUpdate();
        conn.close();
        JOptionPane.showMessageDialog(null, "注册成功");
    }
    public void update(User user) throws SQLException {
        String sql="update `userdata` set PASSWD=? where ACCOUNT=?";
        /*String sql="update `userdata` set PASSWD=? where ACCOUNT=? "+"VALUES(?,?,?)";*/
        Connection conn=DBUtil.getConn();

```

```

        PreparedStatement ps=conn.prepareStatement(sql);
        ps.setString(1,user.getPasswd());
        ps.setString(2,user.getId());
        ps.executeUpdate();
        conn.close();
        JOptionPane.showMessageDialog(null, "修改成功");
    }
}
package View;
import DataService.Data;
import DataService.DataOperator;
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.awt.font.ImageGraphicAttribute;
import java.sql.SQLException;
public class AddTextView extends JFrame implements ActionListener {
    JFrame jFrame;
    JPanel jpanel;
    JButton saveButton;
    JButton exitButton;
    JScrollPane jScrollPane;
    Data data=new Data();
    JTextArea jTextArea;
    private String s;
    private int i;
    public AddTextView(int id, String str) {
        super();
        setUndecorated(true);
        jFrame = new JFrame("退出不要点击右上角");
        jFrame.setSize(600, 420);
        Container contentPane = jFrame.getContentPane();
        contentPane.setLayout(new BorderLayout());
        jFrame.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
        ImageIcon imageIcon=new ImageIcon("./lib/TextBank.png");
        jFrame.setIconImage(imageIcon.getImage());
        jFrame.setLocationRelativeTo(null);
        final JPanel panel = new JPanel();
        jTextArea = new JTextArea();
        panel.add(jTextArea);
        getContentPane().add(jTextArea);
        jTextArea = new JTextArea();
        jTextArea.setLineWrap(true);
        jScrollPane = new JScrollPane(jTextArea);
        jpanel = new JPanel();
        jpanel.setLayout(new GridLayout());
        saveButton = new JButton("保存文本");
        saveButton.addActionListener(this::actionPerformed);
        saveButton.setFocusPainted(false);
        jpanel.add(saveButton);
        exitButton = new JButton("退出");
        exitButton.addActionListener(this::actionPerformed);
        exitButton.setFocusPainted(false);
        jpanel.add(exitButton);
    }
}

```



```

        contentPane.add(jscrollPane, BorderLayout.CENTER);
        contentPane.add(jpanel, BorderLayout.SOUTH);
        jFrame.setVisible(true);
        jFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        i=id;
        s=str;
        jFrame.setLocationRelativeTo(null);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == saveButton) {
            DataOperator dataOperator=new DataOperator();
            try {
                data.setld(i);
                data.setSummary(s);
                data.setText(jTextArea.getText());
                dataOperator.Insert(data);
                System.out.println(jTextArea.getText());
                jFrame.dispose();
            } catch (SQLException throwables) {
                throwables.printStackTrace();
                JOptionPane.showMessageDialog(null, "请选中最后一行后进行添加文本");
            }
        } else {
            jFrame.dispose();
        }
    }
}

package View;
import DataService.Data;
import DataService.DataOperator;
import Model.ClipBoardModel;
import Model.GetCounts;
import javax.swing.*.*;
import javax.swing.event.TableModelEvent;
import javax.swing.event.TableModelListener;
import javax.swing.table.DefaultTableModel;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Vector;
public class BankView extends JFrame {
    private static final long serialVersionUID = 1L;
    private JTable table;
    private DefaultTableModel tableModel;
    private JTextField TextField;
    private int rows;
    private String s;
    public BankView() {
        super();
        setTitle("文本银行");
        setSize(964, 643);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

```

```

final JScrollPane scrollPane = new JScrollPane();
getContentPane().add(scrollPane, BorderLayout.CENTER);
String[] rowValues = { "文本内容" };
this.setLocationRelativeTo(null);
ImageIcon imageIcon=new ImageIcon("./lib/TextBank.png");
this.setIconImage(imageIcon.getImage());
GetCounts getCounts=new GetCounts();
int counts=getCounts.getCount();
tableModel = new DefaultTableModel(rowValues,1);
tableModel.addTableModelListener(new TableModelListener() {
    public void tableChanged(TableModelEvent e) {
        int type = e.getType();
        int row = e.getFirstRow();
        rows=row;
        /*if (type == TableModelEvent.INSERT) { rows=row;}
        else if (type == TableModelEvent.UPDATE){ rows=row;}
        else if (type == TableModelEvent.DELETE) { rows=row;}
        else { rows=row;}*/ });
table = new JTable(tableModel);
table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
table.setRowHeight(30);
scrollPane.setViewportView(table);
final JPanel panel = new JPanel();
getContentPane().add(panel, BorderLayout.SOUTH);
final JLabel aLabel = new JLabel("添加文本标题");
panel.add(aLabel);
TextField = new JTextField(30);
panel.add(TextField);
final JButton addButton = new JButton("添加文本");
addButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==addButton){
            String[] rowValues = {TextField.getText()};
            tableModel.addRow(rowValues);
            Data data=new Data();
            data.setld(rows);
            data.setSummary(TextField.getText());
            new AddTextView(data.getld(),data.getSummary());
            TextField.setText(null);
        }
    }
});
panel.add(addButton);
addButton.setFocusPainted(false);
final JButton delButton = new JButton("删除文本");
delButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==delButton){
            int[] selectedRows = table.getSelectedRows();
            for (int row = 0; row < selectedRows.length; row++) {
                tableModel.removeRow(selectedRows[row] - row);
                DataOperator dataOperator=new DataOperator();
                try {
                    Data data=dataOperator.Find(selectedRows[row]);
                    dataOperator.delete(data);
                } catch (SQLException | InterruptedException throwables) {
                    throwables.printStackTrace();
                }
            }
        }
    }
});

```

```

    }
}
});
panel.add(delButton);
delButton.setFocusPainted(false);
final JButton checkButton = new JButton("查看文本");
checkButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==checkButton){
            rows=table.getSelectedRow();
            new CheckTextView(rows);
        }
    }
});
panel.add(checkButton);
checkButton.setFocusPainted(false);
final JButton copyButton = new JButton("复制文本");
copyButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==copyButton){
            int[] selectedRows = table.getSelectedRows();
            for (int row = 0; row < selectedRows.length; row++) {
                DataOperator dataOperator=new DataOperator();
                Data data;
                try {
                    data=dataOperator.Find(row);
                    if (data!=null){
                        ClipBoardModel clipBoardModel=new ClipBoardModel();
                        clipBoardModel.setSysClipboardText(data.getText());
                    }
                } catch (SQLException throwables) {
                    throwables.printStackTrace();
                }
            }
        }
    }
});
panel.add(copyButton);
copyButton.setFocusPainted(false);
setVisible(true);
if (counts>0){
    for (int i = 0; i < counts+2; i++) {
        DataOperator dataOperator=new DataOperator();
        try {
            Data data= dataOperator.Find(i);
            if (data!=null){
                String[] rowValue = {data.getSummary()};
                tableModel.addRow(rowValue);
            }else{ }
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
    }
}
}
}
}

```

```
package View;
import DataService.Data;
import DataService.DataOperator;
import Model.ClipBoardModel;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.sql.SQLException;
public class CheckTextView extends JFrame implements ActionListener {
    JFrame jFrame;
    JPanel jpanel;
    JButton deleteButton;
    JButton copyButton;
    JButton saveButton;
    JButton exitButton;
    JTextArea jTextArea;
    ClipBoardModel clipBoardModel=new ClipBoardModel();
    Data data=new Data();
    public CheckTextView(int id) {
        super();
        jFrame = new JFrame("退出时不要点击右上角");
        jFrame.setSize(600, 420);
        Container contentPane = jFrame.getContentPane();
        contentPane.setLayout(new BorderLayout());
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jFrame.setLocationRelativeTo(null);
        ImageIcon imageIcon=new ImageIcon("./lib/TextBank.png");
        jFrame.setIconImage(imageIcon.getImage());
        final JPanel panel = new JPanel();
        jTextArea = new JTextArea();
        JScrollPane jScrollPane=new JScrollPane();
        panel.add(jTextArea);
        getContentPane().add(jTextArea);
        jTextArea = new JTextArea();
        jTextArea.setLineWrap(true);
        jScrollPane = new JScrollPane(jTextArea);
        jpanel = new JPanel();
        jpanel.setLayout(new GridLayout());
        copyButton = new JButton("复制文本");
        copyButton.addActionListener(this::actionPerformed);
        jpanel.add(copyButton);
        copyButton.setFocusPainted(false);
        saveButton = new JButton("保存文本");
        saveButton.addActionListener(this::actionPerformed);
        jpanel.add(saveButton);
        saveButton.setFocusPainted(false);
        exitButton = new JButton("退出");
        exitButton.addActionListener(this::actionPerformed);
        exitButton.setFocusPainted(false);
        jpanel.add(exitButton);
        contentPane.add(jScrollPane, BorderLayout.CENTER);
        contentPane.add(jpanel, BorderLayout.SOUTH);
        jFrame.setVisible(true);
        jFrame.addWindowListener(new WindowAdapter() {
```

```

        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });
    DataOperator dataOperator=new DataOperator();
    try {
        Data data=dataOperator.Find(id);
        if(data!=null){
            jTextArea.setText(dataOperator.Find(id).getText());
        }
    } catch (SQLException throwables) {
        throwables.printStackTrace();
    }
    data.setId(id);
    data.setText(jTextArea.getText());
}
public void actionPerformed(ActionEvent e) {
    if(e.getSource() == copyButton){
        clipBoardModel.setSysClipboardText(jTextArea.getText());
        jFrame.dispose();
    }
    else if(e.getSource()==exitButton){
        jFrame.dispose();
    }
    else {
        DataOperator dataOperator=new DataOperator();
        try {
            dataOperator.Update(data);
            jFrame.dispose();
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
    }
}
}

package View;
import Model.Encrypt;
import Model.ListModel;
import UserService.User;
import UserService.UserOperator;
import javax.swing.*;
import javax.swing.border.TitledBorder;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
public class ForgetPasswdView extends JFrame implements ActionListener {
    JButton update;
    JButton cancel;
    JButton lab;
    JLabel name;
    JLabel newpasswd;
    JLabel question;
    JLabel ans;
    private final JComboBox jComboBox;
    private final JTextField username;
    private final JTextField answer;
    private final JPasswordField ipassword;

```

```

public String getp() {
    return String.valueOf(ipassword.getPassword());
}

public ForgetPasswdView() {
    name = new JLabel("用户名");
    name.setBounds(76, 90, 120, 30);
    name.setFont(new Font("", Font.BOLD, 24));
    getContentPane().add(name);
    getContentPane().setLayout(null);
    newpasswd = new JLabel("密保问题");
    newpasswd.setBounds(76, 140, 120, 30);
    newpasswd.setFont(new Font("", Font.BOLD, 24));
    getContentPane().add(newpasswd);
    question = new JLabel("密保答案");
    question.setBounds(76, 190, 120, 30);
    question.setFont(new Font("", Font.BOLD, 24));
    getContentPane().add(question);
    ans = new JLabel("新密码");
    ans.setBounds(76, 240, 120, 30);
    ans.setFont(new Font("", Font.BOLD, 24));
    getContentPane().add(ans);
    username = new JTextField();
    username.setBounds(176, 90, 240, 30);
    username.setFont(new Font("", Font.BOLD, 22));
    getContentPane().add(username);
    jComboBox=new JComboBox(new ListModel());
    jComboBox.setBounds(176,140,240,30);
    getContentPane().add(jComboBox);
    answer = new JTextField();
    answer.setBounds(176, 190, 240, 30);
    answer.setFont(new Font("", Font.BOLD, 22));
    getContentPane().add(answer);
    ipassword = new JPasswordField();
    ipassword.setBounds(176, 240, 240, 30);
    ipassword.setFont(new Font("", Font.BOLD, 22));
    getContentPane().add(ipassword);
    update = new JButton("提交");
    update.setText("提交");
    update.setBounds(134, 300, 136, 40);
    getContentPane().add(update);
    update.setFocusPainted(false);
    cancel = new JButton("取消");
    cancel.setText("取消");
    cancel.setBounds(282, 300, 136, 40);
    getContentPane().add(cancel);
    cancel.setFocusPainted(false);
    lab = new JButton();
    lab.setBorder(new TitledBorder(null, "", TitledBorder.DEFAULT_JUSTIFICATION,
TitledBorder.DEFAULT_POSITION, null, null));
    lab.setForeground(new Color(0, 130, 114));
    lab.setFont(new Font("", Font.BOLD, 40));
    lab.setText("忘记密码");
    lab.setSize(200, 50);
    lab.setBorderPainted(false);
    lab.setFocusPainted(false);
    lab.setContentAreaFilled(false);

```

```

        getContentPane().add(lab);
        setSize(560, 420);
        getContentPane().setLayout(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        ImageIcon imagelcon=new ImageIcon("./lib/TextBank.png");
        this.setIconImage(imagelcon.getImage());
        this.setLocationRelativeTo(null);
        cancel.addActionListener(this);
        update.addActionListener(this);
        setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()== update){
            UserOperator userOperator=new UserOperator();
            try {
                User user=userOperator.FindUser(username.getText());
                if(user==null){
                    JOptionPane.showMessageDialog(null, "未查询到用户");
                }else{
                    String ques=(String)jComboBox.getSelectedltem();
                    if
(user.getAnswer().equals(answer.getText())&&user.getQuestion().equals(ques)&&user.getId().equals(username.getText
())){
                        Encrypt encrypt=new Encrypt();
                        user.setPasswd(encrypt.encode(getp()));
                        userOperator.update(user);
                        dispose();
                        new LoginView();
                    }else{
                        JOptionPane.showMessageDialog(null, "更新失败, 请重试");
                    }
                }
            } catch (SQLException throwables) {
                throwables.printStackTrace();
            }
        }else{
            new LoginView();
        }
    }
}

package View;
import Model.Encrypt;
import Model.ListModel;
import UserService.User;
import UserService.UserOperator;
import javax.swing.*;
import javax.swing.border.TitledBorder;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
public class RegisterView extends JFrame implements ActionListener {
    JButton register ;
    JButton cancel;
    JButton lab;
    JLabel regname;
    JLabel regpasswd;

```

```

JLabel question;
JLabel ans;
JComboBox jComboBox;
JTextField username;
JTextField answer;
JPasswordField ipassword;
public String getp(){
return String.valueOf(ipassword.getPassword());
}
public RegisterView() {
    regname = new JLabel("用户名");
    regname.setBounds(76, 90, 120, 30);
    regname.setFont(new Font("", Font.BOLD, 24));
    getContentPane().add(regname);
    getContentPane().setLayout(null);
    regpasswd = new JLabel("密码");
    regpasswd.setBounds(76, 140, 120, 30);
    regpasswd.setFont(new Font("", Font.BOLD, 24));
    getContentPane().add(regpasswd);
    question = new JLabel("密保问题");
    question.setBounds(76, 190, 120, 30);
    question.setFont(new Font("", Font.BOLD, 24));
    getContentPane().add(question);
    ans = new JLabel("密保答案");
    ans.setBounds(76, 240, 120, 30);
    ans.setFont(new Font("", Font.BOLD, 24));
    getContentPane().add(ans);
    username = new JTextField();
    username.setBounds(176, 90, 240, 30);
    username.setFont(new Font("", Font.BOLD, 22));
    getContentPane().add(username);
    ipassword = new JPasswordField();
    ipassword.setBounds(176, 140, 240, 30);
    ipassword.setFont(new Font("", Font.BOLD, 22));
    getContentPane().add(ipassword);
    jComboBox=new JComboBox(new ListModel());
    jComboBox.setBounds(176,190,240,30);
    getContentPane().add(jComboBox);
    answer = new JTextField();
    answer.setBounds(176, 240, 240, 30);
    answer.setFont(new Font("", Font.BOLD, 22));
    getContentPane().add(answer);
    register = new JButton("注册");
    register.setText("注册");
    register.setBounds(134, 300, 136, 40);
    getContentPane().add(register);
    register.setFocusPainted(false);
    cancel = new JButton("取消");
    cancel.setText("取消");
    cancel.setBounds(282, 300, 136, 40);
    getContentPane().add(cancel);
    cancel.setFocusPainted(false);
    lab = new JButton();
    lab.setBorder(new TitledBorder(null, "", TitledBorder.DEFAULT_JUSTIFICATION,
TitledBorder.DEFAULT_POSITION, null, null));
    lab.setForeground(new Color(0, 130, 114));
    lab.setFont(new Font("", Font.BOLD, 40));

```



```

lab.setText("注册账号");
lab.setSize(200, 50);
lab.setBorderPainted(false);
lab.setFocusPainted(false);
lab.setContentAreaFilled(false);
getContentPane().add(lab);
setSize(560, 420);
getContentPane().setLayout(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
ImageIcon imageIcon=new ImageIcon("./lib/TextBank.png");
this.setIconImage(imageIcon.getImage());
this.setLocationRelativeTo(null);
cancel.addActionListener(this);
register.addActionListener(this);
setVisible(true);
}
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource()==cancel){
        dispose();
        new LoginView();
    }else{
        Encrypt encrypt=new Encrypt();
        String id=username.getText();
        String an=answer.getText();
        String pa=encrypt.encode(getp());
        String ques=(String)jComboBox.getSelectedItemAt();
        UserOperator userOperator=new UserOperator();
        try {
            User user;
            user=userOperator.FindUser(id);
            if (user!=null){
                JOptionPane.showMessageDialog(null, "用户已存在");
            }else{
                user=new User();
                user.setId(id);
                user.setAnswer(an);
                user.setPasswd(pa);
                user.setQuestion(ques);
                userOperator.Insert(user);
                dispose();
                new LoginView();
            }
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
    }
}
}
}
}

```