

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * GUI.java
 *
 * Created on 2009/10/13, 10:40:04
 */

package pacman;

import javax.swing.JOptionPane;

/**
 *
 * @author cbradley
 */
public class GUI extends javax.swing.JFrame {
    db dbMan = new db();
    ClientArray manyClient = new ClientArray();
    /** Creates new form GUI */
    public GUI() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jtaMain = new javax.swing.JTextArea();
        barMain = new javax.swing.JMenuBar();
        menDisplay = new javax.swing.JMenu();
        jmiDisplayAll = new javax.swing.JMenuItem();
        jmiNoStyle = new javax.swing.JMenuItem();
        jmiDispAppo = new javax.swing.JMenuItem();
        jmiDispAreas = new javax.swing.JMenuItem();
        jmiDispTints = new javax.swing.JMenuItem();
        menOther = new javax.swing.JMenu();
        jmiMakeApp = new javax.swing.JMenuItem();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jtaMain.setColumns(20);
        jtaMain.setRows(5);
        jScrollPane1.setViewportView(jtaMain);

```

```
menDisplay.setText("Display");

jmiDisplayAll.setText("Display All Clients");
jmiDisplayAll.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jmiDisplayAllActionPerformed(evt);
    }
});
menDisplay.add(jmiDisplayAll);

jmiNoStyle.setText("Number of Style");
jmiNoStyle.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jmiNoStyleActionPerformed(evt);
    }
});
menDisplay.add(jmiNoStyle);

jmiDispAppo.setText("Display appointments");
jmiDispAppo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jmiDispAppoActionPerformed(evt);
    }
});
menDisplay.add(jmiDispAppo);

jmiDispAreas.setText("Display Areas");
jmiDispAreas.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jmiDispAreasActionPerformed(evt);
    }
});
menDisplay.add(jmiDispAreas);

jmiDispTints.setText("Display Tints Due");
jmiDispTints.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jmiDispTintsActionPerformed(evt);
    }
});
menDisplay.add(jmiDispTints);

barMain.add(menDisplay);

menOther.setText("Other");

jmiMakeApp.setText("Make appointment");
jmiMakeApp.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jmiMakeAppActionPerformed(evt);
    }
});
menOther.add(jmiMakeApp);
```

```

barMain.add(menOther);

setJMenuBar(barMain);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 380,
Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 257,
Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
);

pack();
} // </editor-fold>

private void jmiDisplayAllActionPerformed(java.awt.event.ActionEvent evt) {
    jtaMain.setText(dbMan.displayAll());
}

private void jmiNoStyleActionPerformed(java.awt.event.ActionEvent evt) {
    //this is used if there is a parent GUI
    String style=JOptionPane.showInputDialog(this,"Enter the stlye");
    int iNumStyle=dbMan.countStyle(style);
    jtaMain.setText(style+" : "+iNumStyle+" People");
}

private void jmiDispAppoActionPerformed(java.awt.event.ActionEvent evt) {
    String sClient=JOptionPane.showInputDialog(this,"Enter name of client");
    String sMess=dbMan.getAppointment(sClient);
    //To display an appropriate message
    if(sMess.equals("")){
        JOptionPane.showMessageDialog(this, "Sorry there is no client by this name");
    }else{
        JOptionPane.showMessageDialog(this, sMess);
    }
}

private void jmiMakeAppActionPerformed(java.awt.event.ActionEvent evt) {
    String sClient=JOptionPane.showInputDialog("Please enter Name of client");
    String sDate=JOptionPane.showInputDialog("Please enter date of appointment in
YYYY/MM/DD");
    int cID=dbMan.getID(sClient);
    //Checks if client is here

```

```

if(cID!=0) {
    dbMan.addAppointment(cID, sDate);
}else{
    int iStyle=Integer.parseInt(JOptionPane.showInputDialog(dbMan.getMenuStyle()));
    String sTint=JOptionPane.showInputDialog("Enter tint");
    String sPhone=JOptionPane.showInputDialog("Enter phone number");
    dbMan.addClient(sClient, sPhone, sTint, iStyle);
    cID=dbMan.getID(sClient);
    dbMan.addAppointment(cID, sDate);
}
}

private void jmiDispAreasActionPerformed(java.awt.event.ActionEvent evt) {
    jtaMain.setText(manyClient.toString());
}

private void jmiDispTintsActionPerformed(java.awt.event.ActionEvent evt) {
    jtaMain.setText(manyClient.getTintDue());
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new GUI().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JMenuBar barMain;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JMenuItem jmiDispAppo;
private javax.swing.JMenuItem jmiDispAreas;
private javax.swing.JMenuItem jmiDispTints;
private javax.swing.JMenuItem jmiDisplayAll;
private javax.swing.JMenuItem jmiMakeApp;
private javax.swing.JMenuItem jmiNoStyle;
private javax.swing.JTextArea jtaMain;
private javax.swing.JMenu menDisplay;
private javax.swing.JMenu menOther;
// End of variables declaration
}

```

```
package pacman;

import java.sql.*;
import javax.swing.JOptionPane;

/** @author Craig Bradley */
public class db {

    private Connection conn;

    public db() {
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            conn = DriverManager.getConnection("jdbc:odbc:DRIVER={Microsoft Access Driver
(*.mdb)};DBQ=HairDresser.mdb");
        } catch (Exception ex) {
            System.out.println("Connection error: " + ex.getMessage());
        }
    }

    public void update(String SQL) {
        try {
            Statement stmt = conn.createStatement();
            stmt.execute(SQL);
        } catch (Exception e) {
            //Error message for user
            JOptionPane.showMessageDialog(null, "Sorry I was unable to update the database");
            //Error message for programmer
            System.err.println("Update error: " + e.getMessage() + "\nSQL: " + SQL);
        }
    }

    public ResultSet query(String SQL) {
        ResultSet result = null;
        try {
            Statement stmt = conn.createStatement();
            result = stmt.executeQuery(SQL);
        } catch (Exception e) {
            //Error message for user
            JOptionPane.showMessageDialog(null, "Sorry error in running query");
            //Error message for programmer
            System.err.println("Query error: " + e.getMessage() + "\nSQL: " + SQL);
        }
        return result;
    }

    public String displayAll() {
        String sAll = "Name\tPhone Number\n";
        String SQL = "SELECT Name,[Phone Number] FROM tblClients ORDER BY Name";
        try {
            ResultSet result = query(SQL);
            while (result.next()) {
                sAll += result.getString("Name") + "\t";
                sAll += result.getString("Phone Number") + "\n";
            }
        }
    }
}
```

```

    }
} catch (Exception e) {
    //Error message for user
    JOptionPane.showMessageDialog(null, "Sorry I was unable to display all");
    //Error message for programmer
    System.err.println("Display all error: " + e.getMessage() + "\nSQL: " + SQL);
}
return sAll;
}

//Two different ways of solving this problem is included. Second way is suggested. Or
print out from database the style table allowing the user to enter in number
public int countStyle(String Style) {
    //Solution 1
    int iNumStyle = 0;
    String SQL = "SELECT StyleID FROM tblStyles WHERE StyleDesc='" + Style + "'";
    System.out.println(SQL);
    try {
        ResultSet result = query(SQL);
        int iStyle = 0;
        if (result.next()) {
            iStyle = result.getInt("StyleID");
        } else {
            System.out.println("No records in get StyleID: " + SQL);
        }
        System.out.println("Style No=" + iStyle);
        SQL = "SELECT COUNT(*) AS numOfStyle FROM tblClients WHERE StyleID=" + iStyle;
        result = query(SQL);
        if (result.next()) {
            iNumStyle = result.getInt("numOfStyle");
        } else {
            System.out.println("No records in count style: " + SQL);
        }
    } catch (Exception e) {
        //Error message for user
        JOptionPane.showMessageDialog(null, "Sorry I was unable to count clients using " +
Style + " Style");
        //Error message for programmer
        System.err.println("Display all error: " + e.getMessage() + "\nSQL: " + SQL);
    }
    //Solution 2
    SQL = "SELECT COUNT(*) AS numStyle FROM tblClients INNER JOIN tblStyles ON
tblClients.styleID=tblStyles.styleID WHERE StyleDesc='" + Style + "'";
    try {
        ResultSet result = query(SQL);
        int iNumStyleV2 = 0;
        if (result.next()) {
            iNumStyleV2 = result.getInt("numStyle");
        } else {
            System.out.println("No records in count style: " + SQL);
        }
        System.out.println("Second method: " + iNumStyleV2);
    } catch (Exception e) {
        //Error message for user
        JOptionPane.showMessageDialog(null, "Sorry I was unable to count clients using " +

```

```

Style + " Style");
        //Error message for programmer
        System.err.println("second solution to number of style error: " + e.getMessage() +
"\nSQL: " + SQL);
    }
    return iNumStyle;
}
//Method to get appointment of client
public String getAppointment(String sClient) {
    //variables are used to keep information on client
    String sBooking = "";
    String sTint = "";
    String sStyle = "";
    int cID = 0;
    //this SQL is to get information on client
    String SQL = "SELECT Tint,[Client ID] AS clientID,StyleDesc FROM tblClients INNER JOIN
tblStyles ON tblClients.StyleID=tblStyles.StyleID WHERE Name =' " + sClient + "'";
    try {
        ResultSet result = query(SQL);
        if (result.next()) {
            cID = result.getInt("ClientID");
            sTint = result.getString("Tint");
            System.out.println(sTint);
            if (sTint == null || sTint.equals("") || sTint.equals("null")) {
                sTint = "None";
            }
            sStyle = result.getString("StyleDesc");
        }
    } catch (Exception e) {
        //Error message for user
        JOptionPane.showMessageDialog(null, "Error in finding client: " + sClient);
        //Error message for programmer
        System.err.println("get Client ID error in getAppointment: " + e.getMessage() +
"\nSQL: " + SQL);
    }
    //if is to check that the client does exist
    if (cID != 0) {
        //Date is a reserved word, this is why I put the SQL in [] brackets. A field name
is extremely unlikely to be given with this name.
        //This SQL is going to find the booking/s this client has made
        SQL = "SELECT [Date] AS appDate FROM tblAppmts WHERE ClientID=" + cID;
        try {
            //adding information of client
            sBooking = "Client: " + sClient + " Tint: " + sTint + " Style: " + sStyle +
"\nAppointment/s:\n";
            ResultSet result = query(SQL);
            //adding bookings client has made
            while (result.next()) {
                sBooking += result.getDate("appDate") + "\n";
            }
        } catch (Exception e) {
            //Error message for user
            JOptionPane.showMessageDialog(null, "Error in getting appointment date");
            //Error message for programmer

```

```

        System.err.println("get dates error in get appointments: " + e.getMessage() +
"\nSQL: " + SQL);
    }
}
return sBooking;
}
//Display styles so user can choose the style ID
public String getMenuStyle(){
    String sMen="";
    String SQL="SELECT * FROM tblStyles";
    try {
        ResultSet result = query(SQL);
        while(result.next()){
            sMen+=result.getInt("StyleID")+" "+result.getString("StyleDesc");
        }
    } catch (Exception e) {
        //Error message for user
        JOptionPane.showMessageDialog(null, "Error in getting Client ID");
        //Error message for programmer
        System.err.println("Error in getting client ID: " + e.getMessage() + "\nSQL: " +
SQL);
    }
    return sMen;
}
//Code that will get the clients ID from the name, used in add appointment
public int getID(String sClient) {
    String SQL = "SELECT [client ID] AS clientID FROM tblClients WHERE Name='" + sClient +
""";
    int cID = 0;
    try {
        ResultSet result = query(SQL);
        if (result.next()) {
            cID = result.getInt("clientID");
        }
    } catch (Exception e) {
        //Error message for user
        JOptionPane.showMessageDialog(null, "Error in getting Client ID");
        //Error message for programmer
        System.err.println("Error in getting client ID: " + e.getMessage() + "\nSQL: " +
SQL);
    }
    return cID;
}
//Code for add appointment, will be called if client does not exist.
public void addClient(String sClient, String sPhone, String sTint,int iStyle) {
    //If you where running out of time you could display this as a menu e.g. 1 - Cut, 2 -
etc...
    String SQL = "";
    SQL = "INSERT INTO tblClients(Name,[Phone Number],Tint,StyleID) VALUES('" + sClient +
"', '" + sPhone + "', '" + sTint + "', '" + iStyle + ")";
    update(SQL);
}

public void addAppointment(int cID, String sDate) {

```

```
String SQL = "";
try {
    SQL = "INSERT INTO tblAppmts([Date],ClientID) VALUES("# + sDate + "#," + cID +
    ")";

    update(SQL);
    JOptionPane.showMessageDialog(null, "Message displayed successfully");
} catch (Exception e) {
    //Error message for user
    JOptionPane.showMessageDialog(null, "Error in adding appointment");
    //Error message for programmer
    System.err.println("Error in adding appointments: " + e.getMessage() + "\nSQL: " +
SQL);
}
}
```

```
package pacman;

/** @author Craig Bradley */
public class Style {
    protected String sName;
    protected String sPhone;
    protected String sArea;

    public Style(String sName, String sPhone) {
        this.sName = sName;
        this.sPhone = sPhone;
        sArea=getArea(sPhone);
    }
    public String getArea(String Phone){
        String Area="Unknown";
        String[][] arrAreaCodes= new String [3][2];
        arrAreaCodes[0][0]="edenvale";
        arrAreaCodes[0][1]="452,453,609";
        arrAreaCodes[1][0]="BedfordView";
        arrAreaCodes[1][1]="455";
        arrAreaCodes[2][0]="Kensington";
        arrAreaCodes[2][1]="616";

        String areaCode=Phone.substring(3,6);
        System.out.println("AreaCode: "+areaCode);

        for(int i=0;i<arrAreaCodes.length;i++){
            if(arrAreaCodes[i][1].contains(areaCode)){
                Area=arrAreaCodes[i][0];
                break;
            }
        }
        return Area;
    }
    @Override
    public String toString(){
        return sName+"\t"+sPhone+"\t"+sArea;
    }

    public String getSArea() {
        return sArea;
    }

    public void setSArea(String sArea) {
        this.sArea = sArea;
    }

    public String getSName() {
        return sName;
    }

    public void setSName(String sName) {
        this.sName = sName;
    }
}
```

```
public String getSPhone() {  
    return sPhone;  
}  
  
public void setSPhone(String sPhone) {  
    this.sPhone = sPhone;  
}  
}
```

```
package pacman;

/** @author Craig Bradley */
public class tint extends Style{
    private String sColour;
    public tint(String sName, String sPhone,String sColour) {
        super(sName, sPhone);
        this.sColour=sColour;
    }

    @Override
    public String toString() {
        return super.toString()+"\t"+sColour;
    }

    public String getSColour() {
        return sColour;
    }

    public void setSColour(String sColour) {
        this.sColour = sColour;
    }
}
```

```

package pacman;

import java.sql.ResultSet;
import javax.swing.JOptionPane;

/** @author Craig Bradley */
public class ClientArray {

    public Style[] arrStyles = new Style[50];
    int size;

    public ClientArray() {
        size = 0;
        db dbMan = new db();
        String SQL = "SELECT * FROM tblClients";
        try {
            ResultSet result = dbMan.query(SQL);
            while (result.next()) {
                String sName = result.getString("Name");
                String sPhone = result.getString("Phone Number");
                String sTint = result.getString("Tint");
                if (sTint == null || sTint.equals("") || sTint.equalsIgnoreCase("null")) {
                    arrStyles[size] = new Style(sName, sPhone);
                } else {
                    arrStyles[size] = new tint(sName, sPhone, sTint);
                }
                size++;
            }
        } catch (Exception e) {
            //Error message for user
            JOptionPane.showMessageDialog(null, "Error in adding clients to array from db");
            //Error message for programmer
            System.err.println("Error in adding clients to array from db: " + e.getMessage() +
"\nSQL: " + SQL);
        }
    }

    public String getTintDue(){
        String sDue="";
        String SQL="SELECT ClientID FROM tblAppmts WHERE [Date]<(#2008/06/01#-6*7)";
        db dbMan=new db();
        try {
            ResultSet result = dbMan.query(SQL);
            while(result.next()){
                int cID=result.getInt("ClientID");
                SQL="SELECT Tint,Name,[Phone Number] AS phone FROM tblClients WHERE [Client
ID]="+cID;

                ResultSet result2 = dbMan.query(SQL);
                if(result2.next()){
                    String Tint=result2.getString("");
                    if(!(Tint==null||Tint.equals("")||Tint.equalsIgnoreCase("null"))){
                        sDue+=result.getString("Name")+"\t"+result.getString("phone")+"\n";
                    }
                }
            }
        }
    }
}

```

```
    } catch (Exception e) {
        //Error message for user
        JOptionPane.showMessageDialog(null, "Error in getting tints due");
        //Error message for programmer
        System.err.println("Error in getting tints due: " + e.getMessage() + "\nSQL: " +
SQL);
    }
    return sDue;
}
public String toString(){
    String sAll="";
    for(int i=0;i<size;i++){
        sAll+=arrStyles[i].toString()+"\n";
    }
    return sAll;
}
}
```