**Code Busters Four Barber Shop Algorithm Code**

**Interface:**

**public** **interface** BarberFunctions {

**public** **void** addAppointment(**long** phone,String n,String d,String t, String s, **int** p) **throws** Exception;

**public** **void** generateSchedule() **throws** Exception; // complete

**public** **void** generateRevenueReport() **throws** Exception;

**public** **void** deleteAppointment(**long** n) **throws** Exception;

}

**Backend:**

**import** java.io.\*;

**import** java.awt.event.\*;

**import** java.awt.\*;

**import** java.sql.\*;

**import** java.io.\*;

**import** java.awt.event.\*;

**import** java.awt.\*;

**import** java.sql.\*;

**import** java.util.\*;

**public** **class** BarberShopApp **implements** BarberFunctions {

Connection dbConnection = **null**;

Statement dbStatement = **null**;

ResultSet dbResults = **null**;

PreparedStatement insertIntoDatabase;

PreparedStatement removeFromDatabase;

**private** String database = "jdbc:mysql://localhost:3306/barber\_shop";

**private** String user = "root";

**private** String password = "";

PrintWriter schedule = **null**; // file for schedule

PrintWriter revenue = **null**; // file for revenue report

PrintWriter databaseLog = **null**; // file for status of the database (Ex. added, deleted, nothing changed etc)

**private** **int** total = 0; // total for revenue

Calendar currDate;

**long** phoneNum;

String name;

String day;

String time;

String service;

**int** price;

**public** **void** connectToDatabase() **throws** SQLException, ClassNotFoundException {

**try** {

Class.*forName*("com.mysql.jdbc.Driver");

// 1. Connect To database

dbConnection = DriverManager.*getConnection*(database,user,password);

//System.out.println("Test");

// 2. Create the statement

dbStatement = dbConnection.createStatement();

// 3. Generate results

dbResults = dbStatement.executeQuery("SELECT \* FROM Barber\_Shop"); // client phone

}

**catch**(SQLException e) {

e.printStackTrace();

}

**finally** {

}

}

**public** **void** generateSchedule() **throws** SQLException, IOException, ClassNotFoundException{

**try** {

currDate = Calendar.*getInstance*();

schedule = **new** PrintWriter(**new** FileWriter("Schedule.txt"));

schedule.println("\t\t\t\tCODE BUSTERS FOUR BARBERSHOP SCHEDULE");

schedule.println("\t\t\t\t\t" + currDate.getTime());

schedule.println(" ");

schedule.println("PHONE NUMBER\t| \tNAME | \t\t\tDAY | \tTIME | \tSERVICE | \tPRICE");

schedule.println("----------------------------------------------------------------------------");

connectToDatabase();

**while**(dbResults.next()) {

phoneNum = dbResults.getLong(1);

name = dbResults.getString(2);

day = dbResults.getString(3);

time = dbResults.getString(4);

service = dbResults.getString(5);

price = dbResults.getInt(6);

schedule.println(phoneNum + "\t|\t" + name + "\t|\t" + day + "|\t" + time + "|\t" + service + "|\t $ " + price);

}

}

**catch**(FileNotFoundException e) {

System.***out***.println("File Not Found");

}

**finally** {

schedule.close();

}

}

**public** **void** generateRevenueReport() **throws** SQLException, IOException, ClassNotFoundException {

**try** {

total = 0;

currDate = Calendar.*getInstance*();

revenue = **new** PrintWriter(**new** FileWriter("Revenue.txt"));

revenue.println("\t\t\t\tCODE BUSTERS FOUR BARBERSHOP WEEKLY REVENUE REPORT");

revenue.println("\t\t\t\t\t" + currDate.getTime());

revenue.println(" ");

revenue.println("\t\t\t\tDAY\tTIME\tSERVICE\tPRICE");

revenue.println("\t\t\t----------------------------------------------------------");

connectToDatabase();

**while**(dbResults.next()) {

revenue.println("\t\t\t\t" + dbResults.getString(3) + "\t " + dbResults.getString(4) + "\t " + dbResults.getString(5) + " $ "

+ dbResults.getInt(6));

**int** cost = dbResults.getInt(6);

total = total + cost;

}

revenue.println(" ");

revenue.println("\t\t\t\t\t ---------------------------------");

revenue.println("\t\t\t\t\t Total Weekly Revenue: $ " + total);

}

**catch**(FileNotFoundException fe) {

fe.printStackTrace();

}

**finally** {

revenue.close();

}

}

**public** **void** addAppointment(**long** phone,String n,String d,String t, String s, **int** p) **throws** ClassNotFoundException, SQLException, IOException {

// method to add to database

**try** {

connectToDatabase(); // get connection to database

String intoData = " insert into barber\_shop (client\_phone, client\_name, app\_day, app\_time, service,price)"

+ " values (?, ?, ?, ?, ?,?)";

insertIntoDatabase = dbConnection.prepareStatement(intoData);

**try** {

databaseLog = **new** PrintWriter(**new** FileWriter("Database Logs.txt"),**true**);

}

**catch**(FileNotFoundException e) {

System.***out***.println("File Not Found");

}

**if**(phone < 1000000000) {

databaseLog.println(phone + " is an Invalid Phone Number" +

" Database Unaffected");

}

**while**(dbResults.next()) {

**if**(phone == dbResults.getLong(1)) {

databaseLog.println(phone + " Is Already Scheduled For The Week");

}

}

**if**(n.equals("") || n.equals(" ")) {

databaseLog.println("Invalid Name Entry Database Unaffected");

}

**if**(d.equals(" ")) {

databaseLog.print("No Day Selected Database Unaffected");

}

**if**(t.equals(" ")) {

databaseLog.print("No Time Selected Database Unaffected");

}

**if**(s.equals(" ")) {

databaseLog.print("No Service Selected Database Unaffected");

}

**else** {

insertIntoDatabase.setLong(1, phone);

insertIntoDatabase.setString(2, n);

insertIntoDatabase.setString(3, d);

insertIntoDatabase.setString(4, t);

insertIntoDatabase.setString(5, s);

insertIntoDatabase.setInt(6, p);

insertIntoDatabase.execute();

}

}

**catch**(SQLException e) {

e.printStackTrace();

}

}

**public** **void** deleteAppointment(**long** phone) **throws** ClassNotFoundException, SQLException {

**try** {

connectToDatabase();

String removeFrom = "delete from barber\_shop where client\_phone = ?";

removeFromDatabase = dbConnection.prepareStatement(removeFrom);

**while**(dbResults.next()) {

**if**(dbResults.getLong(1) != phone) {

System.***out***.println("Not In Database");

}

**else** {

removeFromDatabase.setLong(1, phone);

removeFromDatabase.executeUpdate();

}

}

}

**catch**(SQLException e) {

e.printStackTrace();

}

}

}

**GUI:**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JRadioButton;

import java.awt.Color;

import java.awt.Font;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.sql.SQLException;

import java.util.InputMismatchException;

import java.awt.event.ActionEvent;

import javax.swing.UIManager;

public class Barbershop extends JFrame {

BarberShopApp newApp = new BarberShopApp();

//BarberAppExceptions appExcep = new BarberAppExceptions();

private JPanel contentPane;

private JTextField clientName;

private JTextField clientPhone;

private JTextField deletePhone;

public JButton btnCancelAppointment;

JButton btnWorkSchedule;

JButton btnRevenueReport;

JButton btnClearForm;

JRadioButton rdbtnMo;

JRadioButton rdbtnTu;

JRadioButton rdbtnWed;

JRadioButton rdbtnTh;

JRadioButton rdbtnFri;

JRadioButton rdbtnSat;

JRadioButton rdbtn5pm;

JRadioButton rdbtn6pm;

JRadioButton rdbtn7pm;

JRadioButton rdbtn8pm;

JRadioButton rdbtn9pm;

JRadioButton rdbtnCut;

JRadioButton rdbtnCutWash;

JRadioButton rdbtnCutWashDye;

/\*

\* Options selected by user

\*/

String selectedDay;

String selectedTime;

String selectedService;

int selectedPrice;

Long numberToDelete;

Long clientInsertPhone; // value to be stored into database

String clientInsertName; // value to be stored into database

String choosenDay; // value to be stored into database

String choosenTime; // value to be stored into database

String choosenService; // value to be stored into database

int totalPrice; // value to be stored into database

/\*\*

\* Launch the application.

\*/

/\*\*

\* Create the frame.

\*/

public Barbershop() {

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(250, 250, 550, 500);

contentPane = new JPanel();

contentPane.setBackground(Color.GRAY);

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JPanel panel = new JPanel();

panel.setBackground(Color.LIGHT\_GRAY);

panel.setBounds(31, 49, 429, 248);

contentPane.add(panel);

panel.setLayout(null);

JLabel lblFullName = new JLabel("Client Full Name : ");

lblFullName.setFont(new Font("Lucida Grande", Font.BOLD | Font.ITALIC, 13));

lblFullName.setBounds(6, 6, 141, 21);

panel.add(lblFullName);

clientName = new JTextField();

clientName.setBounds(159, 1, 149, 26);

panel.add(clientName);

clientName.setColumns(10);

JLabel lblClientPhoneNumber = new JLabel("Client Phone Number :");

lblClientPhoneNumber.setFont(new Font("Lucida Grande", Font.BOLD | Font.ITALIC, 13));

lblClientPhoneNumber.setBounds(6, 39, 160, 16);

panel.add(lblClientPhoneNumber);

clientPhone = new JTextField();

clientPhone.setColumns(10);

clientPhone.setBounds(159, 34, 149, 26);

panel.add(clientPhone);

rdbtnMo = new JRadioButton("Mo");

rdbtnMo.setBounds(179, 79, 55, 20);

rdbtnMo.addActionListener(new SelectDay());

panel.add(rdbtnMo);

JLabel lblClientAppointmentDay = new JLabel("Client Appointment Day :");

lblClientAppointmentDay.setFont(new Font("Lucida Grande", Font.BOLD | Font.ITALIC, 13));

lblClientAppointmentDay.setBounds(6, 83, 177, 16);

panel.add(lblClientAppointmentDay);

rdbtnWed = new JRadioButton("Wed");

rdbtnWed.setBounds(290, 76, 68, 26);

rdbtnWed.addActionListener(new SelectDay());

panel.add(rdbtnWed);

rdbtnTu = new JRadioButton("Tu");

rdbtnTu.setBounds(232, 79, 62, 20);

rdbtnTu.addActionListener(new SelectDay());

panel.add(rdbtnTu);

rdbtnTh = new JRadioButton("Th");

rdbtnTh.setBounds(361, 76, 62, 26);

rdbtnTh.addActionListener(new SelectDay());

panel.add(rdbtnTh);

rdbtnFri = new JRadioButton("Fri");

rdbtnFri.setBounds(179, 111, 62, 23);

rdbtnFri.addActionListener(new SelectDay());

panel.add(rdbtnFri);

rdbtnSat = new JRadioButton("Sat");

rdbtnSat.setBounds(232, 112, 62, 21);

rdbtnSat.addActionListener(new SelectDay());

panel.add(rdbtnSat);

JLabel lblClientAppointmentTime = new JLabel("Client Appointment Time :");

lblClientAppointmentTime.setFont(new Font("Lucida Grande", Font.BOLD | Font.ITALIC, 13));

lblClientAppointmentTime.setBounds(6, 146, 200, 16);

panel.add(lblClientAppointmentTime);

rdbtn5pm = new JRadioButton("5 pm");

rdbtn5pm.setBounds(179, 146, 74, 20);

rdbtn5pm.addActionListener(new ChooseTime());

panel.add(rdbtn5pm);

rdbtn7pm = new JRadioButton("7 pm");

rdbtn7pm.setBounds(302, 144, 68, 20);

rdbtn7pm.addActionListener(new ChooseTime());

panel.add(rdbtn7pm);

JLabel lblSelectService = new JLabel("Select Service : ");

lblSelectService.setFont(new Font("Lucida Grande", Font.BOLD | Font.ITALIC, 13));

lblSelectService.setBounds(6, 195, 121, 16);

panel.add(lblSelectService);

rdbtn6pm = new JRadioButton("6 pm");

rdbtn6pm.setBounds(242, 144, 74, 20);

rdbtn6pm.addActionListener(new ChooseTime());

panel.add(rdbtn6pm);

rdbtn8pm = new JRadioButton("8 pm");

rdbtn8pm.setBounds(361, 144, 74, 20);

rdbtn8pm.addActionListener(new ChooseTime());

panel.add(rdbtn8pm);

rdbtn9pm = new JRadioButton("9 pm");

rdbtn9pm.setBounds(179, 174, 74, 16);

rdbtn9pm.addActionListener(new ChooseTime());

panel.add(rdbtn9pm);

rdbtnCut = new JRadioButton("Cut $10");

rdbtnCut.setBounds(109, 195, 94, 16);

rdbtnCut.addActionListener(new SelectService());

panel.add(rdbtnCut);

rdbtnCutWash = new JRadioButton("Cut & Wash $15");

rdbtnCutWash.setBounds(199, 195, 133, 16);

rdbtnCutWash.addActionListener(new SelectService());

panel.add(rdbtnCutWash);

rdbtnCutWashDye = new JRadioButton("Cut &Wash & Dye $20");

rdbtnCutWashDye.setBounds(109, 223, 193, 16);

rdbtnCutWashDye.addActionListener(new SelectService());

panel.add(rdbtnCutWashDye);

JLabel lblNewLabel = new JLabel("BARBER SHOP");

lblNewLabel.setFont(new Font("Lucida Grande", Font.BOLD | Font.ITALIC, 18));

lblNewLabel.setForeground(Color.WHITE);

lblNewLabel.setBounds(193, 6, 159, 30);

contentPane.add(lblNewLabel);

JButton btnNewButton = new JButton("Add Appointment ");

btnNewButton.setBounds(0, 309, 143, 29);

btnNewButton.addActionListener(new AddAppointment());

contentPane.add(btnNewButton);

btnClearForm = new JButton("Clear Form");

btnClearForm.addActionListener(new ClearForm());

btnClearForm.setBounds(374, 309, 122, 29);

contentPane.add(btnClearForm);

btnCancelAppointment = new JButton("Cancel Appointment ");

btnCancelAppointment.addActionListener(new CancelAppointment());

btnCancelAppointment.setBounds(181, 309, 159, 29);

contentPane.add(btnCancelAppointment);

deletePhone = new JTextField();

deletePhone.setText("Identify Client By Phone Number ");

deletePhone.setBounds(0, 350, 226, 29);

contentPane.add(deletePhone);

deletePhone.setColumns(10);

btnWorkSchedule = new JButton("Work Schedule");

btnWorkSchedule.setBounds(0, 422, 127, 29);

contentPane.add(btnWorkSchedule);

btnWorkSchedule.addActionListener(new ButtonListener());

btnRevenueReport = new JButton("Revenue Report");

btnRevenueReport.addActionListener(new ButtonListener() {

});

btnRevenueReport.setBounds(181, 422, 143, 29);

contentPane.add(btnRevenueReport);

}

public class ButtonListener implements ActionListener {

public void actionPerformed(ActionEvent e){

if(e.getSource().equals(btnWorkSchedule)) {

try {

newApp.generateSchedule();

} catch (ClassNotFoundException | SQLException | IOException e1) {

e1.printStackTrace();

}

}

if(e.getSource().equals(btnRevenueReport)) {

try {

newApp.generateRevenueReport();

} catch (ClassNotFoundException | SQLException | IOException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

}

}

public class ClearForm implements ActionListener {

public void actionPerformed(ActionEvent e) {

if(e.getSource().equals(btnClearForm)) {

clientName.setText("");

clientPhone.setText("");

deletePhone.setText("");

rdbtnMo.setSelected(true);

rdbtnTu.setSelected(false);

rdbtnWed.setSelected(false);

rdbtnTh.setSelected(false);

rdbtnFri.setSelected(false);

rdbtnSat.setSelected(false);

rdbtn5pm.setSelected(true);

rdbtn6pm.setSelected(false);

rdbtn7pm.setSelected(false);

rdbtn8pm.setSelected(false);

rdbtn9pm.setSelected(false);

rdbtnCut.setSelected(true);

rdbtnCutWash.setSelected(false);

rdbtnCutWashDye.setSelected(false);

}

}

}

public class SelectDay implements ActionListener {

public void actionPerformed(ActionEvent e) {

if(e.getSource().equals(rdbtnMo)) {

rdbtnMo.setSelected(true);

rdbtnTu.setSelected(false);

rdbtnWed.setSelected(false);

rdbtnTh.setSelected(false);

rdbtnFri.setSelected(false);

rdbtnSat.setSelected(false);

selectedDay = rdbtnMo.getText();

}

if(e.getSource().equals(rdbtnTu)) {

rdbtnMo.setSelected(false);

rdbtnTu.setSelected(true);

rdbtnWed.setSelected(false);

rdbtnTh.setSelected(false);

rdbtnFri.setSelected(false);

rdbtnSat.setSelected(false);

selectedDay = rdbtnTu.getText();

}

if(e.getSource().equals(rdbtnWed)) {

rdbtnMo.setSelected(false);

rdbtnTu.setSelected(false);

rdbtnWed.setSelected(true);

rdbtnTh.setSelected(false);

rdbtnFri.setSelected(false);

rdbtnSat.setSelected(false);

selectedDay = rdbtnWed.getText();

}

if(e.getSource().equals(rdbtnTh)) {

rdbtnMo.setSelected(false);

rdbtnTu.setSelected(false);

rdbtnWed.setSelected(false);

rdbtnTh.setSelected(true);

rdbtnFri.setSelected(false);

rdbtnSat.setSelected(false);

selectedDay = rdbtnTh.getText();

}

if(e.getSource().equals(rdbtnFri)) {

rdbtnMo.setSelected(false);

rdbtnTu.setSelected(false);

rdbtnWed.setSelected(false);

rdbtnTh.setSelected(false);

rdbtnFri.setSelected(true);

rdbtnSat.setSelected(false);

selectedDay = rdbtnFri.getText();

}

if(e.getSource().equals(rdbtnSat)) {

rdbtnMo.setSelected(false);

rdbtnTu.setSelected(false);

rdbtnWed.setSelected(false);

rdbtnTh.setSelected(false);

rdbtnFri.setSelected(false);

rdbtnSat.setSelected(true);

selectedDay = rdbtnSat.getText();

}

}

}

public class ChooseTime implements ActionListener {

public void actionPerformed(ActionEvent e) {

if(e.getSource().equals(rdbtn5pm)) {

rdbtn5pm.setSelected(true);

rdbtn6pm.setSelected(false);

rdbtn7pm.setSelected(false);

rdbtn8pm.setSelected(false);

rdbtn9pm.setSelected(false);

selectedTime = rdbtn5pm.getText();

}

if(e.getSource().equals(rdbtn6pm)) {

rdbtn5pm.setSelected(false);

rdbtn6pm.setSelected(true);

rdbtn7pm.setSelected(false);

rdbtn8pm.setSelected(false);

rdbtn9pm.setSelected(false);

selectedTime = rdbtn6pm.getText();

}

if(e.getSource().equals(rdbtn7pm)) {

rdbtn5pm.setSelected(false);

rdbtn6pm.setSelected(false);

rdbtn7pm.setSelected(true);

rdbtn8pm.setSelected(false);

rdbtn9pm.setSelected(false);

selectedTime = rdbtn7pm.getText();

}

if(e.getSource().equals(rdbtn8pm)) {

rdbtn5pm.setSelected(false);

rdbtn6pm.setSelected(false);

rdbtn7pm.setSelected(false);

rdbtn8pm.setSelected(true);

rdbtn9pm.setSelected(false);

selectedTime = rdbtn8pm.getText();

}

if(e.getSource().equals(rdbtn9pm)) {

rdbtn5pm.setSelected(false);

rdbtn6pm.setSelected(false);

rdbtn7pm.setSelected(false);

rdbtn8pm.setSelected(false);

rdbtn9pm.setSelected(true);

selectedTime = rdbtn9pm.getText();

}

}

}

public class SelectService implements ActionListener {

public void actionPerformed(ActionEvent e) {

if(e.getSource().equals(rdbtnCut)) {

rdbtnCut.setSelected(true);

rdbtnCutWash.setSelected(false);

rdbtnCutWashDye.setSelected(false);

selectedPrice = 10;

selectedService = "Cut";

}

if(e.getSource().equals(rdbtnCutWash)) {

rdbtnCut.setSelected(false);

rdbtnCutWash.setSelected(true);

rdbtnCutWashDye.setSelected(false);

selectedPrice = 15;

selectedService = "Cut & Wash";

}

if(e.getSource().equals(rdbtnCutWashDye)) {

rdbtnCut.setSelected(false);

rdbtnCutWash.setSelected(false);

rdbtnCutWashDye.setSelected(true);

selectedPrice = 20;

selectedService = "Cut & Wash & Dye";

}

}

}

public class AddAppointment implements ActionListener {

public void actionPerformed(ActionEvent e) {

try {

clientInsertPhone = Long.parseLong(clientPhone.getText());

clientInsertName = clientName.getText();

choosenDay = selectedDay;

choosenTime = selectedTime;

choosenService = selectedService;

totalPrice = selectedPrice;

newApp.addAppointment(clientInsertPhone,clientInsertName,choosenDay,choosenTime, choosenService, totalPrice);

}

catch(ClassNotFoundException ex) {

ex.printStackTrace();

}

catch (SQLException e1) {

clientName.setText("Invalid Field, Please Try Again");

}

catch(InputMismatchException exe) {

clientPhone.setText("Invalid Phone Number");

}

catch (IOException e1) {

e1.printStackTrace();

}

}

}

public class CancelAppointment implements ActionListener {

public void actionPerformed(ActionEvent e) {

try {

numberToDelete = Long.parseLong(deletePhone.getText());

newApp.deleteAppointment(numberToDelete);

}

catch(ClassNotFoundException ex) {

ex.printStackTrace();

}

catch (SQLException exc) {

clientName.setText("Invalid Field, Please Try Again");

}

}

}

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Barbershop frame = new Barbershop();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

}