

POS System

김선민
이희성
최윤수

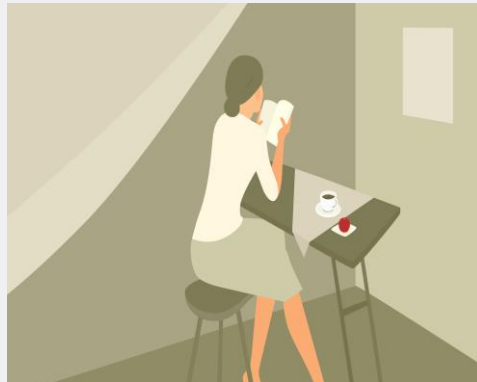


INDEX



01. 프로그램 실행

프로그램 시연



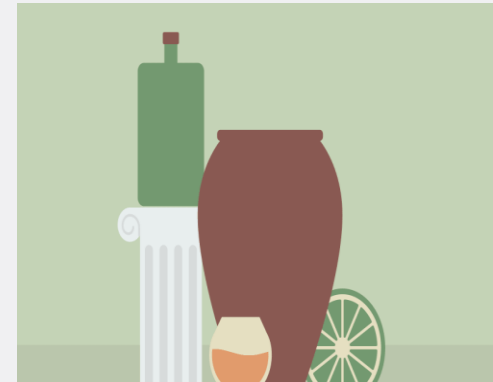
02. 소스코드

소스코드 설명



03. Database

사용 SQL문



04. 제목 입력

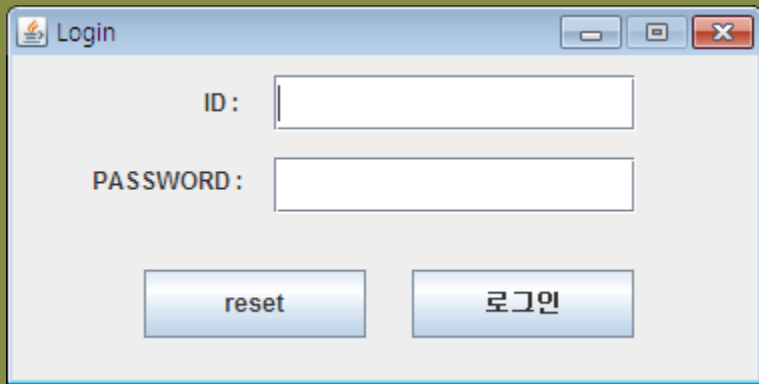
이곳에 제목에 대한
설명을 간략히 적어주세요

01.

프로그램 실행

전체적인 프로그램의 실행 화면

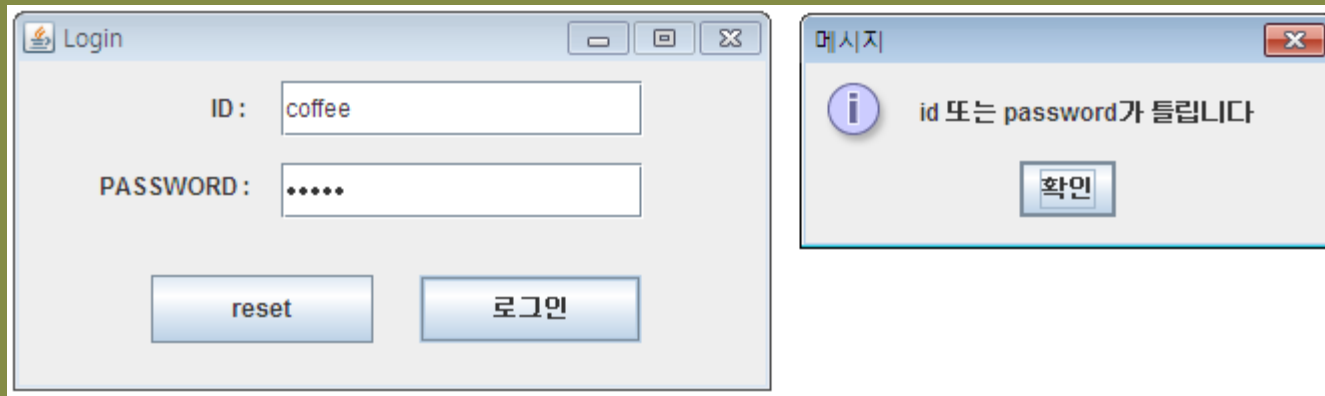
Login 화면



A screenshot of a web browser window titled "Login". It contains two input fields: "ID :" and "PASSWORD :". Below the fields are two buttons: "reset" and "로그인" (Login).

성공

실패



A screenshot showing the "Login" window with "ID :" set to "coffee" and "PASSWORD :" set to ".....". The "로그인" button is highlighted. To the right, a "메시지" (Message) dialog box is displayed with an information icon and the text "id 또는 password가 틀립니다" (ID or password is incorrect). Below the message is a "확인" (OK) button.

Login

프로그램 시작 시
실행되는 class

```
loginBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String id = textField.getText();
        String pw = "";
        // passwordfield는 string형으로 들어가지 않기 때문에 한 단어씩 char배열에 넣어준 후 string으로 형변환 시켜주어야 한다
        char[] secret_pw = passwordField.getPassword();

        CoffeeDAO coffeeDAO = new CoffeeDAO();

        for (char cha : secret_pw ) {
            Character.toString(cha);
            //pw 에 저장하기, pw 에 값이 비어있으면 저장, 값이 있으면 이어서 저장하는 삼항연산자
            pw += (pw.equals("")) ? ""+cha+" " : ""+cha+" ";
        }
        try {
            // coffeeDAO.login(id, pw) 결과값이 1이면
            if(coffeeDAO.login(id, pw) == 1) {
                new Main();
                dispose();
            } else {
                JOptionPane.showMessageDialog(null, "id 또는 password가 틀립니다");
                System.exit(0);
            }
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }
});
panel_2.add(loginBtn);
```

Main 화면

The screenshot displays the 'Ezen Cafeteria' application window. It features a menu table on the left, a beverage selection grid on the right, and a payment section at the bottom left. The interface is designed for a cafeteria management system.

Window Title: Ezen Cafeteria

Menu Table:

메뉴	수량	가격

Payment Section:

결제 방식 ☐ 현금 ☐ 카드

Beverage Selection Grid:

HotCoffee	IceCoffee	Beverage
카라멜마끼아또	에스프레소 콘파냐	카페라떼
카페모카	바닐라라떼	카푸치노
아메리카노		

Action Buttons:

매출	결제	할인	선택취소	전체취소
----	----	----	------	------

Main

로그인 성공 시
실행되는 클래스

```
Vector<Info> rowData = null;  
CoffeeDAO coffeeDAO = null;
```

```
JPanel panel1, panel2, panel3, panel4, panel5, panel6, panel7;  
JTabbedPane menuTab = new JTabbedPane(); //JTabbedPane생성  
JTextField tf = new JTextField(); // 총 금액  
JButton[] HotCoffeeBtn, ICECoffeeBtn, ShakeFlatchinoBtn; // 메뉴 버튼
```

```
// 주문 테이블  
String [] ColName = {"메뉴","수량","가격"};  
String [][] Data ;  
int count =1;  
DefaultTableModel model = new DefaultTableModel(Data,ColName);  
JTable menuTable = new JTable(model);  
JScrollPane menuScroll = new JScrollPane();
```

```
// 메뉴이름, 가격  
String[] HotCoffee;      Integer hotCoffeePrice[];  
String[] ICECoffee;      Integer iceCoffeePrice[];  
String[] ShakeFlatchino; Integer shakeFlatchinoPrice[];
```

```
public Main() {  
    super("Ezen Cafeteria");  
  
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);  
    setBounds(100, 100, 902, 563);  
  
    launched();  
    createMenuBar();  
  
    ImageIcon img = new ImageIcon("C:\\wwwgit!\\wwwjava\\wwwProjects\\wwwCafe_Main\\www장구.jpg");  
    setIconImage(img.getImage());  
    setResizable(false);  
    setVisible(true);  
}
```

메뉴 주문

Ezen Cafeteria

File Edit

메뉴	수량	가격
카라멜마끼아또	1	3900
카페모카	1	3900
바닐라라떼	1	3900
에스프레소 콘파냐	1	3300
카페라떼	1	3700
ICE 바닐라라떼	1	3900

결제 방식 ☐ 현금 ☐ 카드

HotCoffeeIceCoffeeBeverage

카라멜마끼아또

에스프레소 콘파냐

카페라떼

카페모카

바닐라라떼

카푸치노

아메리카노

매출

결제

할인

선택취소

전체취소

Main

메뉴 주문 시 리스트 출력

```
// 주문리스트
menuTable.setRowHeight(38);
menuTable.getColumnModel().setPreferredWidth(140);
menuScroll.setBounds(26, 24, 353, 264);
menuScroll.setViewportView(menuTable);
getContentPane().add(menuScroll);
for(int i=0;i<HotCoffeeBtn.length;i++) {
    final int index = i;
    HotCoffeeBtn[i].addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            DefaultTableModel m = (DefaultTableModel)menuTable.getModel();
            String menu = HotCoffee[index];
            m.addRow(new Object[]{menu,count, hotCoffeePrice[index]});
            if(menu.equals("")) {
                m.removeRow(m.getRowCount()-1);
            }
        }
    });
    ICECoffeeBtn[i].addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            DefaultTableModel m = (DefaultTableModel)menuTable.getModel();
            String menu = ICECoffee[index];
            m.addRow(new Object[]{menu,count, iceCoffeePrice[index]});
            if(menu.equals("")) {
                m.removeRow(m.getRowCount()-1);
            }
        }
    });
    ShakeFlatchinoBtn[i].addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            DefaultTableModel m = (DefaultTableModel)menuTable.getModel();
            String menu = ShakeFlatchino[index];
            m.addRow(new Object[]{menu,count, shakeFlatchinoPrice[index]});
            if(menu.equals("")) {
                m.removeRow(m.getRowCount()-1);
            }
        }
    });
}
```

메뉴 할인

Ezen Cafeteria

File Edit

메뉴	수량	가격
카라멜마끼아또	1	3510
카페모카	1	3120
바닐라라떼	1	0
에스프레소 콘파냐	1	3300
카페라떼	1	3700
ICE 바닐라라떼	1	3900

결제 방식 ☐ 현금 ☐ 카드

매출

결제

할인

선택취소

전체취소

HotCoffee IceCoffee Beverage

카라멜마끼아또

에스프레소 콘파냐

카페라떼

카페모카

바닐라라떼

카푸치노

아메리카노

할인

10%할인

20%할인

스마트멤버십쿠폰

Main

선택 메뉴에 대한 할인

```
// 할인
inventoryManagement[2].addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        Discount discount = new Discount();
        discount.btn10.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                int result = (int) (menuTable.getValueAt(menuTable.getSelectedRow(), 2));
                menuTable.setValueAt((int)(result * 0.9), menuTable.getSelectedRow(), 2);
                discount.dispose();
            }
        });
        discount.btn20.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                int result = (int) (menuTable.getValueAt(menuTable.getSelectedRow(), 2));
                menuTable.setValueAt((int)(result * 0.8), menuTable.getSelectedRow(), 2);
                discount.dispose();
            }
        });
        discount.btnFree.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                menuTable.setValueAt(0, menuTable.getSelectedRow(), 2);
                discount.dispose();
            }
        });
    }
});
}
```

Main

선택 메뉴에 대한
취소 및 전체취소

```
// 선택취소
inventoryManagement[3].addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        DefaultTableModel m = (DefaultTableModel)menuTable.getModel();
        m.removeRow(menuTable.getSelectedRow());
    }
});

// 전체취소
inventoryManagement[4].addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        DefaultTableModel m = (DefaultTableModel)menuTable.getModel();
        m.setRowCount(0);
        tf.setText(String.valueOf(""));
    }
});
```

결제

Ezen Cafeteria

File Edit

메뉴	수량	가격
아메리카노	1	3200
카페모카	1	3900

결제 방식 ☐ 현금 ☒ 카드

총 금액 : 7100원

매출

결제

할인

선택취소

전체취소

HotCoffee IceCoffee Beverage

카라멜마끼아또

에스프레소 콘파나

카페라떼

카페모카

바닐라라떼

카푸치노

아메리카노

결제창

? 카드결제 하시겠습니까?

예(Y)

아니오(N)

결제창

받은 금액 7200

결제

결제창

결제되었습니다

확인

Ezen Cafeteria
File Edit

메뉴	수량	가격
카라멜마끼아또	1	3510
카페모카	1	3120
바닐라라떼	1	0
에스프레소 콘파냐	1	3300
카페라떼	1	3700
ICE 바닐라라떼	1	3900

결제 방식
☒ 현금
☐ 카드

총 금액 : 22330원

HotCoffeeIceCoffeeBeverage

카라멜마끼아또

에스프레소 콘파냐

카페라떼

카페모카

바닐라라떼

카푸치노

아메리카노

매출

결제

할인

선택취소

전체취소

결제창
?
현금결제 하시겠습니까?
예(Y)
아니오(N)

결제창
받은 금액
23000
결제

결제창
결제되었습니다. 거스름돈은 670원 입니다
확인

Main

주문 결제

```
//결제
inventoryManagement[1].addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        int rowCount = menuTable.getRowCount();
        int pay[] = new int[rowCount];
        for(int i=0;i<rowCount;i++) {
            pay[i] = (int)menuTable.getValueAt(i, 2);
        }
        //
        int sum = IntStream.of(pay).sum();
        tf.setText(String.valueOf(" 총 금액 : " + sum + "원"));
        tf.setFont(new Font("고딕", Font.BOLD, 15));

        if(cash.isSelected() != true) {
            int exitOptionCash = JOptionPane.showConfirmDialog(null, "현금결제 하시겠습니까?", "결제창", JOptionPane.YES_OPTION);
            if (exitOptionCash == JOptionPane.YES_OPTION) {
                Payment payment = new Payment();
                payment.payBtn.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent e) {
                        String str = "";
                        str = payment.payMoney.getText();
                        str = str.trim();
                        int paymoney = 0;
                        paymoney = Integer.parseInt(str);
                        if(sum <= paymoney) {
                            CoffeeDAO coffeeDAO = new CoffeeDAO();
                            int price = 0;
                            for (int i = 0; i < menuTable.getRowCount(); i++) {
                                SimpleDateFormat format2 = new SimpleDateFormat ("yyyy년 MM월 dd일 HH시mm분ss초");
                                String format_time2 = format2.format (System.currentTimeMillis());

                                String payway = (String) cash.getText();
                                String menucode = (String) menuTable.getValueAt(i, 0);
                                String menu = (String) menuTable.getValueAt(i, 1);
                                price = (int) menuTable.getValueAt(i, 2);
                                String ordertime = format_time2;

                                coffeeDAO.coffeeadd(payway, menucode, menu, price, ordertime);
                            }
                        }
                    }
                });
            }
        }
    }
});
```

Main

주문 결제

```
JOptionPane.showMessageDialog(null, "결제되었습니다. 거스름돈은 " + (paymoney - sum) + "원 입니다", "결제창", JOptionPane.PLAIN_MESSAGE);
payment.dispose();
// 테이블값 초기화
DefaultTableModel model = (DefaultTableModel)menuTable.getModel();
model.setNumRows(0);
tf.setText("");
} else {
    JOptionPane.showMessageDialog(null, "돈이 부족해요!", "결제창", JOptionPane.INFORMATION_MESSAGE);
    payment.dispose();
}
}
});
}
} else if(card.isSelected() == true) {
    int exitOptionCard = JOptionPane.showConfirmDialog(null, "카드결제 하시겠습니까?", "결제창", JOptionPane.YES_OPTION);
    if (exitOptionCard == JOptionPane.YES_OPTION) {
        Payment payment = new Payment();
        payment.payBtn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String str = "";
                str = payment.payMoney.getText();
                int paymoney = 0;
                paymoney = Integer.parseInt(str);
                if(sum <= paymoney) {
                    CoffeeDAO coffeeDAO = new CoffeeDAO();
                    for (int i = 0; i < menuTable.getRowCount(); i++) {
                        SimpleDateFormat format2 = new SimpleDateFormat ("yyyy년 MM월 dd일 HH시mm분ss초");
                        String format_time2 = format2.format (System.currentTimeMillis());

                        String payway = (String) card.getText();
                        String menucode = (String) menuTable.getValueAt(i, 0);
                        String menu = (String) menuTable.getValueAt(i, 1);
                        int price = (int) menuTable.getValueAt(i, 2);
                        String ordertime = format_time2;

                        coffeeDAO.coffeeadd(payway, menucode, menu, price, ordertime);
                    }
                }
            }
        });
    }
}
```


Main

주문 결제

```

JOptionPane.showMessageDialog(null, "결제되었습니다", "결제창", JOptionPane.PLAIN_MESSAGE);
payment.dispose();
DefaultTableModel model = (DefaultTableModel)menuTable.getModel();
model.setNumRows(0);
tf.setText("");
} else {
    JOptionPane.showMessageDialog(null, "잔액 부족입니다. 다른 결제 수단을 이용해 주세요:", "결제창", JOptionPane.INFORMATION_MESSAGE);
    payment.dispose();
}
}
});
}
}
}
});

```

메뉴 추가

Ezen Cafeteria

File Edit

메뉴	수량	가격

결제 방식 ☐ 현금 ☐ 카드

HotCoffeeIceCoffeeBeverage

카라멜 마끼아토

에스프레소 콘파냐

카페라떼

카페 모카

바닐라라떼

카푸치노

아메리카노

매출

결제

할인

선택취소

전체취소

메뉴생성

메뉴 코드

메뉴 이름

메뉴 가격

생성

메뉴 삭제

Ezen Cafeteria

File Edit

메뉴	수량	가격

결제 방식 ☐ 현금 ☐ 카드

HotCoffeeIceCoffeeBeverage

카라멜마끼아또

에스프레소 콘파냐

카페라떼

카페모카

바닐라라떼

카푸치노

아메리카노

매출

결제

할인

선택취소

전체취소

입력

?

삭제할 메뉴를 적어주세요

확인

취소

Main

메뉴 추가 및 삭제

```
// file 메뉴바 edit 메뉴
class EditMenuActionListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();
        switch (cmd) {
            case "추가" :
                new CreateMenu();
                break;
            case "삭제" :
                // InputDialog에 적은 메뉴를 delete값에 저장
                String delete = (String) JOptionPane.showInputDialog("삭제할 메뉴를 적어주세요");
                CoffeeDAO coffeeDAO = new CoffeeDAO();
                String menu = delete;
                try {
                    coffeeDAO.delete(menu);
                    for (int i = 0; i < HotCoffeeBtn.length; i++) {
                        if(HotCoffeeBtn[i].getText().equals(delete)) {
                            HotCoffeeBtn[i].setText("");
                            hotCoffeePrice[i]=0;
                            HotCoffee[i] = "";
                        }
                    }
                    for (int i = 0; i < ICECoffeeBtn.length; i++) {
                        if(ICECoffeeBtn[i].getText().equals(delete)) {
                            ICECoffeeBtn[i].setText("");
                            iceCoffeePrice[i]=0;
                            ICECoffee[i] = "";
                        }
                    }
                    for (int i = 0; i < ShakeFlatchinoBtn.length; i++) {
                        if(ShakeFlatchinoBtn[i].getText().equals(delete)) {
                            ShakeFlatchinoBtn[i].setText("");
                            shakeFlatchinoPrice[i]=0;
                            ShakeFlatchino[i] = "";
                        }
                    }
                } catch (Exception e1) {
                    e1.printStackTrace();
                }
            }
        }
    }
}
```

Create Menu

메뉴 추가 및 삭제

```

JButton btnNewButton = new JButton("생성");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        CoffeeDAO coffeeDAO = new CoffeeDAO();
        String menuCode = menuCodeJtf.getText();
        String menuName = menuNameJtf.getText();
        // 메뉴가격은 int로 받아야 하기 때문에 textfield에 작성한 값을 integer로 형변환 시켜야 함
        int menuPrice = Integer.parseInt(menuPriceJtf.getText());

        int n = coffeeDAO.menuAdd(menuCode, menuName, menuPrice);
        // 결과값이 1이 나오면
        if(n==1) {
            // JOptionPane.YES_OPTION : 확인버튼이 있는 ConfirmDialog
            int result = JOptionPane.showConfirmDialog(null, "메뉴가 생성되었습니다.", "메뉴생성", JOptionPane.YES_OPTION);
            if(result == JOptionPane.YES_OPTION) {
                dispose();
            }
        }
    }
});

```

매출

결제 방법	MenuCode	메뉴	가격	날짜
카드	HM1	카페모카	3900	2020년 06월 05일 11시35분49초
카드	HC2	카푸치노	3700	2020년 06월 05일 11시35분49초
카드	HC1	카페라테	3700	2020년 06월 05일 11시35분49초
카드	HEC	에스프레소 콘...	3300	2020년 06월 05일 11시35분49초
현금	HCM	카라멜마끼아또	3900	2020년 06월 05일 11시46분16초
현금	HM1	카페모카	3900	2020년 06월 05일 11시46분16초
현금	BD1	딸기 쉐이크	4800	2020년 06월 04일 12시26분24초

총 매출 : 83370 현금 매출 : 50630 카드 매출 : 32740

POS **매출** 금일 매출 메뉴별판매

전체매출

당일매출

결제 방법	MenuCode	메뉴	가격	날짜
카드	HM1	카페모카	3900	2020년 06월 05일 11시35분49초
카드	HC2	카푸치노	3700	2020년 06월 05일 11시35분49초
카드	HC1	카페라테	3700	2020년 06월 05일 11시35분49초
카드	HEC	에스프레소 콘...	3300	2020년 06월 05일 11시35분49초
현금	HCM	카라멜마끼아또	3900	2020년 06월 05일 11시46분16초
현금	HM1	카페모카	3900	2020년 06월 05일 11시46분16초
현금	BD1	딸기 쉐이크	4800	2020년 06월 04일 12시26분24초

총 매출 : 51830 금일 현금 매출 : 30130 금일 카드 매출 : 21700

POS 매출 **금일 매출** 메뉴별판매

매출

매출목록	
메뉴	판매잔수
에스프레소 콘파냐	4
카페모카	4
치즈가쿠키햇대 웨이크	2
바닐라라떼	2
카페라떼	2
카라멜마끼아또	2
카푸치노	1
총 매출 : 51830 금일 현금 매출 : 30130 금일 카드 매출 : 21700	
POS	매출
금일 매출	메뉴별판매

메뉴 별 판매량

Sales

전체 판매 리스트

```
 JButton salesBtn = new JButton("매출");
 salesBtn.addActionListener(new ActionListener() {
     public void actionPerformed(ActionEvent e) { //테이블에 전체 매출 리스트 출력
         rowData = coffeeDAO.GetAllSellList();
         int totalSum=0;
         for(int i=0; i<rowData.size();i++){
             totalSum += rowData.get(i).getPrice();
         }
         rowData = coffeeDAO.GetTotalCash();
         int totalCash = 0;
         for (int i = 0; i < rowData.size(); i++) {
             totalCash += rowData.get(i).getTotalCash();
         }
         rowData = coffeeDAO.GetTotalCard();
         int totalCard = 0;
         for (int i = 0; i < rowData.size(); i++) {
             totalCard += rowData.get(i).getTotalCard();
         }
         totalLabel.setText("총 매출 : " + Integer.toString(totalSum));
         cardLabel.setText("카드 매출 : " + Integer.toString(totalCard));
         cashLabel.setText("현금 매출 : " + Integer.toString(totalCash));
     }
 });
 buttonPanel.add(salesBtn);
```


Coffee Dao

전체 판매 리스트

`JTable(Object[][] rowData, Object[] columnName)`

첫 번째 매개변수는 테이블에 들어갈 데이터이며, 오브젝트 타입의 2차원 배열을 받는다.
두 번째 매개변수는 테이블의 첫 번째 행에 들어가 데이터를 구분 짓는 열의 이름이 들어간다.

```
//벡터를 받아서 매출리스트를 2차원 배열로 만들어주는 메소드
public Object[][] makeArr(Vector<Info> lists){

    Object [][] coffeearr = new Object [lists.size()][5];

    for(int i=0; i<lists.size();i++){
        coffeearr[i][0]=lists.get(i).getPayway();
        coffeearr[i][1]=lists.get(i).getMenuCode();
        coffeearr[i][2]=lists.get(i).getMenu();
        coffeearr[i][3]=lists.get(i).getPrice();
        coffeearr[i][4]=lists.get(i).getDate();
    }

    return coffeearr;
}
```

Sales

당일 판매 리스트

```
JButton todayBtn = new JButton("금일 매출");
todayBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy년 MM월 dd일");
        Date now = new Date();
        String currentTime = simpleDateFormat.format(now);

        rowData = coffeeDAO.currentCellTotal();
        int currentTotal = 0;
        String beforeDate;
        for (int i = 0; i < rowData.size(); i++) {
            beforeDate = rowData.get(i).getDate();
            if (currentTime.compareTo(beforeDate) == 0) {
                currentTotal = rowData.get(i).getPrice();
            }
        }

        rowData = coffeeDAO.currentCellCash();
        int currentCash = 0;
        for (int i = 0; i < rowData.size(); i++) {
            beforeDate = rowData.get(i).getDate();
            if (currentTime.compareTo(beforeDate) == 0) {
                currentCash = rowData.get(i).getTotalCash();
            }
        }

        rowData = coffeeDAO.currentCellCard();
        int currentCard = 0;
        for (int i = 0; i < rowData.size(); i++) {
            beforeDate = rowData.get(i).getDate();
            if (currentTime.compareTo(beforeDate) == 0) {
                currentCard = rowData.get(i).getTotalCard();
            }
        }
        totalLabel.setText("금일총 매출 : " + Integer.toString(currentTotal));
        cashLabel.setText("금일 현금 매출 : " + Integer.toString(currentCash));
        cardLabel.setText("금일 카드 매출 : " + Integer.toString(currentCard));
    }
});
buttonPanel.add(todayBtn);
```

Sales

메뉴별 판매 리스트

```
JButton menuBtn = new JButton("메뉴 별 판매");
menuBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) { //테이블에 메뉴별 판매리스트
        rowData = coffeeDAO.GetAllSellList();
        scrollPane.setVisible(false);
        table2 = new JTable(coffeeDAO.makelistArr(coffeeDAO.Getsellcount()), sellName);
        table2.setRowHeight(38);
        table2.setBounds(1, 27, 450, 288);
        tablePanel.add(table2);

        JScrollPane scrollPane = new JScrollPane(table2);
        scrollPane.setBounds(0, 0, 562, 285);
        scrollPane.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED);
        scrollPane.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED);
        tablePanel.add(scrollPane);
    }
});
buttonPanel.add(menuBtn);
```

Coffee Dao

메뉴별 판매 리스트

//벡터를 받아서 메뉴별 판매량을 2차원 배열로 만들어주는 메소드

```
public Object[][] makelistArr(Vector<Info> lists){
```

```
    Object [][] coffeearr = new Object [lists.size()][2];
```

```
    for(int i=0; i<lists.size();i++){
        coffeearr[i][0]=lists.get(i).getMenu();
        coffeearr[i][1]=lists.get(i).getPrice();
    }
```

```
    return coffeearr;
```

```
}
```

03.

SQL 문법

프로그램에서 사용되어지는
SQL 구문의 이해



1. Database 연동을 위한 Java 문법

[1] 드라이버 정의

```
private String driver = "oracle.jdbc.driver.OracleDriver" ;
```

[2] url 지정

```
private String url = "jdbc:oracle:thin:@localhost:1521:xe" ;
```

[3] database id, password

```
private String username = "scott" ; // db id  
private String password = "tiger" ; // db password
```

[4] 데이터베이스와 연결하는 객체 Connection

```
private Connection conn = null ;
```

[5] JDBC 드라이버 로딩

```
Class.forName(driver) ;
```

[6] Connection 생성

```
conn = DriverManager.getConnection(url, username, password) ;
```

[7] 연결 해제

```
conn.close();
```

[8] commit

```
conn.commit();
```

[9] 구문 수행

```
pstmt.executeQuery(); (by select)
```

```
pstmt.executeUpdate(); (by SELECT 구문을 제외한 다른 구문)
```


2. 프로그램 내 사용 된 SQL 문구

[1] login을 위한 sql 문구

```
select password from login where id = ?
```

[2] 메뉴 추가를 위한 sql 문구

```
insert into coffeemenu (menucode, menu, price) values (?, ?, ?)
```

[3] 메뉴 삭제를 위한 sql 문구

```
delete coffeemenu where menu = ?
```

[4] 결제 후의 매출목록 추가

```
select password from login where id = ?
```

[5] 메뉴 추가를 위한 sql 문구

```
insert into coffee (payway, menucode, menu, price, ordertime) values (?, (select  
menucode from coffeemenu where menu=?), ?, ?, ?)
```

[6] 매출 합계 구하기

```
select sum(price) from coffee where payway like '현금'  
select sum(price) from coffee where payway like '카드'
```

[7] 메뉴코드에 따른 리스트 출력

```
select menu, price from coffeemenu where menucode like 'H%'
```

```
select menu, price from coffeemenu where menucode like 'I%'
```

```
select menu, price from coffeemenu where menucode like 'B%'
```

[8] 메뉴 별 판매량

```
select menu , count(*) from coffee group by menu order by count(*) desc
```

[9] 매출 리스트

```
select * from coffee
```

[10] 당일 매출

```
select sum(price), SUBSTR(ordertime,1,13) from coffee where payway like '카드'  
group by SUBSTR(ordertime,1,13)
```

```
select sum(price), SUBSTR(ordertime,1,13) from coffee where payway like ' 현금'  
group by SUBSTR(ordertime,1,13)
```

```
select sum(price), SUBSTR(ordertime,1,13) from coffee group by  
SUBSTR(ordertime,1,13)
```