

커피 사조



목차

- 001 팀원 및 프로젝트 소개
- 002 개발 환경 & 개발 일정
- 003 클래스, 패키지 설계도, Use Case
- 004 기능 설명
- 005 시연 및 Q & A



커피사조

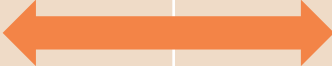



조장 신경섭

조원 김진숙, 송현아, 서정도, 김선인, 문예림

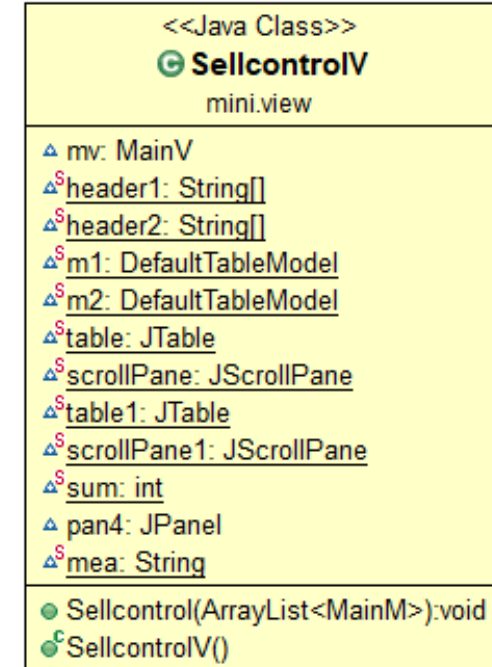
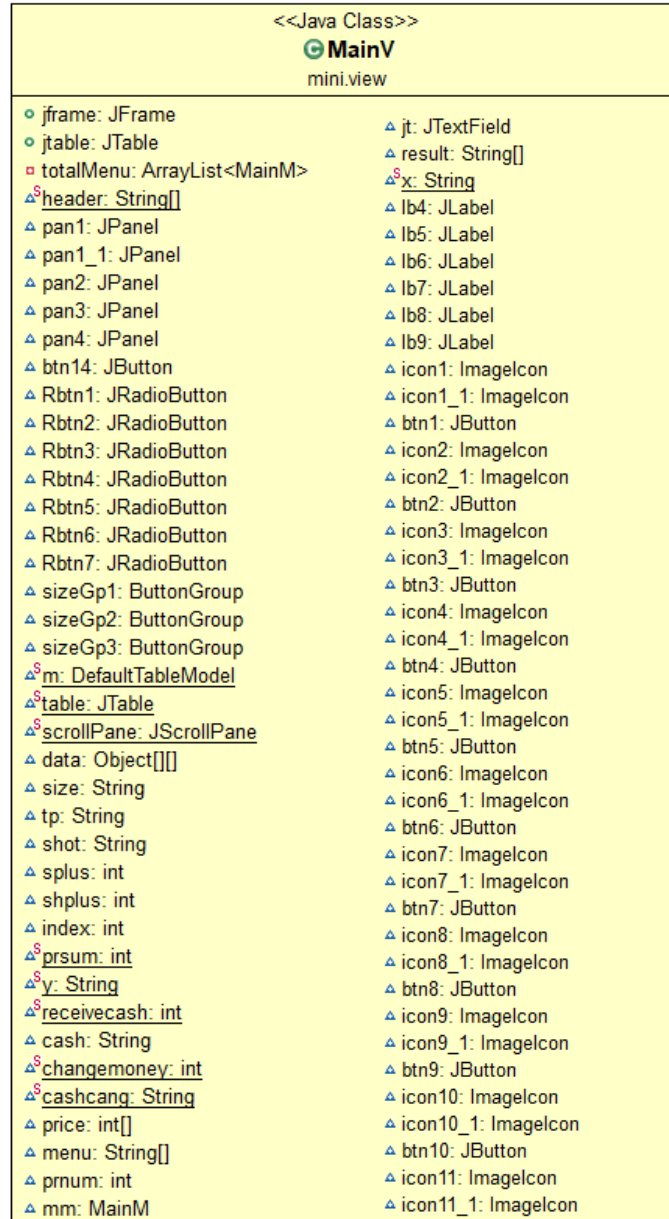
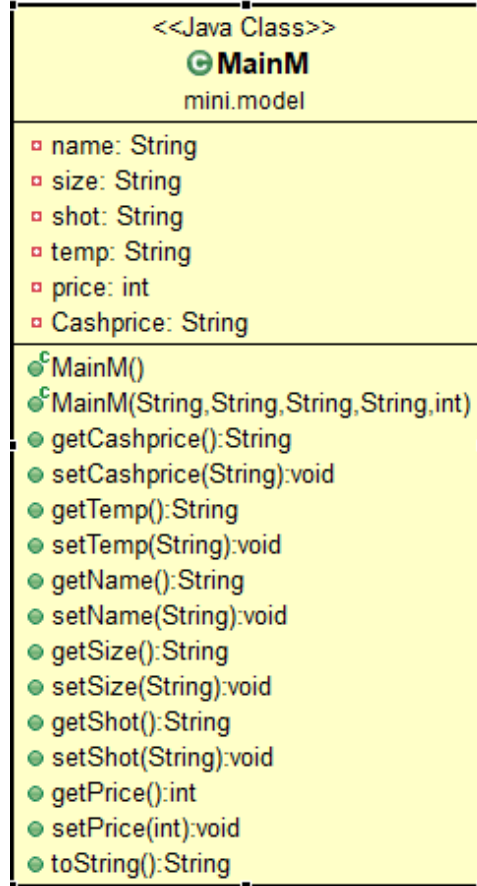
프로젝트 소개

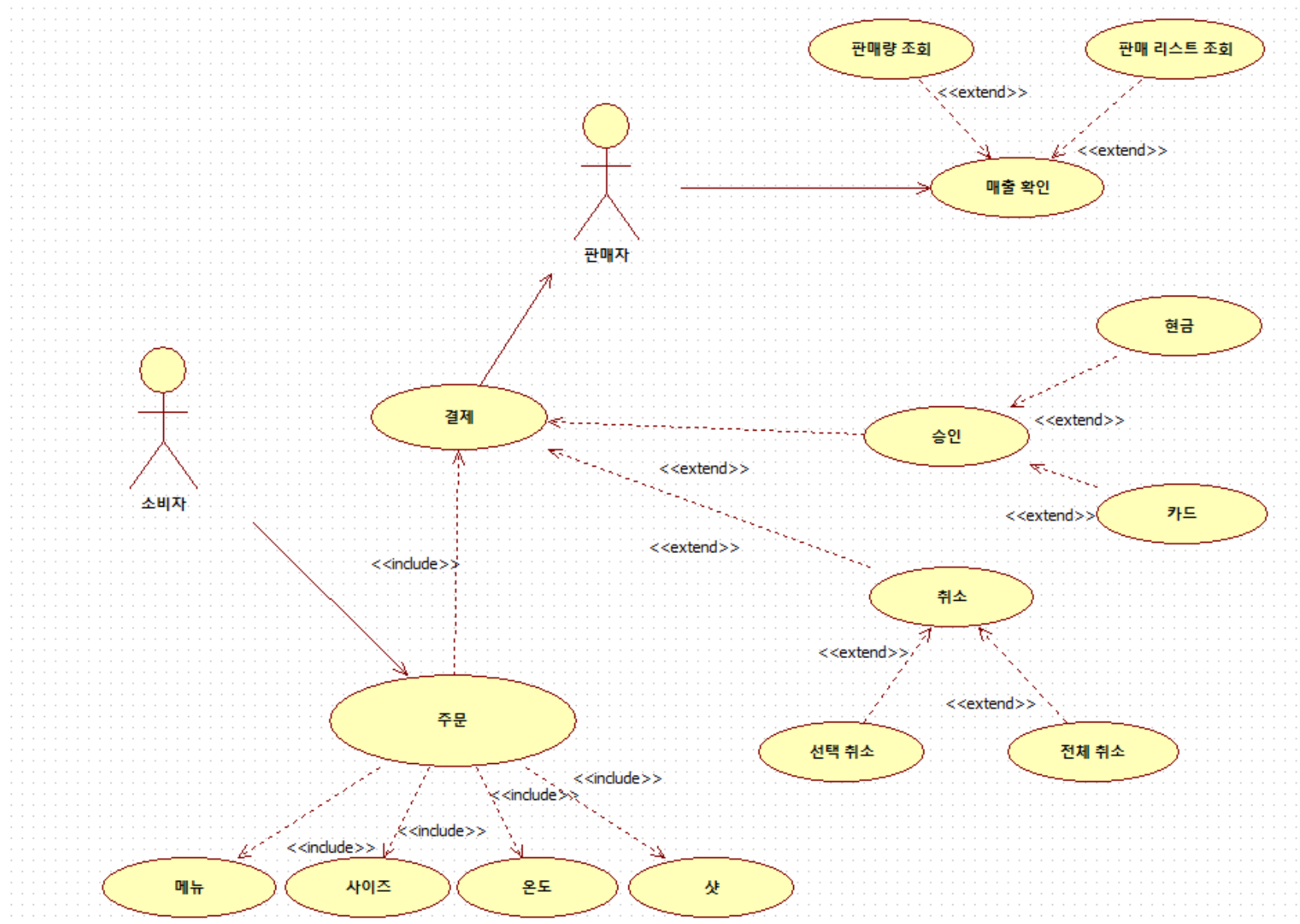
- 커피메뉴를 고를 수 있고, 주문 후 판매, 매출을 조회할 수 있는 카페 관리 시스템
- 커피 POS기와 비슷한 시스템



	9	10	11	12	13	14	15	16	17	18
아이디어 회의										
구현										
UI 디자인										
테스트 디버깅										

003 클래스 다이어그램





004 기능 설명 - 커피 메뉴 / 커피 옵션



```
btn1.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        menu = new String[] {"에스프레소"};  
        price = new int[] {1500};  
    }  
});
```

Coffee Size

- ☐ Small Size
- ☐ Regular Size
- ☐ Large Size

Ice / Hot

- ☐ Ice
- ☐ Hot

Shot Add

- ☐ Yes
- ☐ No

```
Rbtn2.addActionListener(new ActionListener(){  
    @Override  
    public void actionPerformed(ActionEvent e1)  
    {  
        if(Rbtn2.isSelected())  
        {  
            size = Rbtn2.getText(); // 사이즈 칸에 버튼 이름 써짐  
            splus = 500; // 금액 추가 (레귤러) 결제금액에 추가됨  
        }  
    }  
});
```


004 기능 설명 - 화살표 / 추가 / 매출 확인



추 가

매출 확인

```
btn13.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        pan1_1.hide();  
        pan1.setVisible(true);  
    }  
});
```

```
btn13_1.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        pan1.hide();  
        pan1_1.setVisible(true);  
    }  
});
```

```
btn14.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        try{  
            Object[] add = {menu[0],size,tp,shot,(price[0]+splus+shplus)};  
            totalMenu.add(new MainM(menu[0],size,tp,shot,(price[0]+splus+shplus)));  
  
            m.addRow(add);  
  
            prsum=prsum+(int)(price[prnum]+splus+shplus);  
            x=Integer.toString(prsum);  
            lb5.setText(x+y);  
        } catch (NullPointerException e1) {  
            JOptionPane.showMessageDialog(null, "메뉴를 선택해주세요", "Message", JOptionPane.ERROR_MESSAGE);  
            e1.printStackTrace();  
        }  
    }  
});
```

```
btn15.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        SellcontrolV scv = new SellcontrolV();  
        scv.Sellcontrol(totalMenu);  
        jframe.hide();  
    }  
});
```

```

public JFrame jframe = new JFrame();
public JTable jtable = new JTable();
static String[] header = {"메뉴", "사이즈", "온도", "샷 추가", "가격"};

static DefaultTableModel m = new DefaultTableModel(header,0) {
    public boolean isCellEditable(int i, int c) {
        return false;
    }
};

static JTable table = new JTable(m);
static JScrollPane scrollPane = new JScrollPane(table);

```

```

public void MainView() {
    JPanel pan3 = new JPanel();
    pan3.setBackground(Color.white);
    pan3.setBounds(0,600,900,300);
    jframe.getContentPane().add(pan3);
    pan3.setLayout(null);

    Color colorB = new Color(80,60,20); //colorBrown
    Color colorLY = new Color(244,238,206); //colorLightYellow
    Color colorY = new Color(243,226,136); //colorYellow

    table.setRowHeight(30); // 행의 높이
    table.getTableHeader().setReorderingAllowed(false); // 열 이동 불가
    table.getTableHeader().setResizingAllowed(false); // 크기 조절 불가
    table.getTableHeader().setBackground(colorLY); // 배경
    table.getTableHeader().setForeground(colorB); // 글씨
    table.setSelectionBackground(colorY); // 선택된 부분

    DefaultTableCellRenderer dtc = new DefaultTableCellRenderer(); // 가운데 정렬
    dtc.setHorizontalAlignment(JLabel.CENTER);

    DefaultTableCellRenderer dtr = new DefaultTableCellRenderer(); // 오른쪽 정렬
    dtr.setHorizontalAlignment(JLabel.RIGHT);

    table.getColumnModel().getColumn(0).setPreferredWidth(300);
    table.getColumnModel().getColumn(0).setCellRenderer(dtc);
    table.getColumnModel().getColumn(1).setPreferredWidth(150);
    table.getColumnModel().getColumn(1).setCellRenderer(dtc);
    table.getColumnModel().getColumn(2).setPreferredWidth(150);
    table.getColumnModel().getColumn(2).setCellRenderer(dtc);
    table.getColumnModel().getColumn(3).setPreferredWidth(150);
    table.getColumnModel().getColumn(3).setCellRenderer(dtc);
    table.getColumnModel().getColumn(4).setPreferredWidth(150);
    table.getColumnModel().getColumn(4).setCellRenderer(dtr);

    table.getTableHeader().setFont(new Font("맑은고딕", Font.BOLD, 15));
}

```

004 기능 설명 - 테이블

```
btn1.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        menu = new String[] {"에스프레소"};  
        price = new int[] {1500};  
    }  
});
```

```
Rbtn6.addActionListener(new ActionListener(){  
    @Override  
    public void actionPerformed(ActionEvent e1)  
    {  
        if(Rbtn6.isSelected())  
        {  
            shot = Rbtn6.getText();  
            shplus = 500; // 샷추가 금액 결제금액에 추가됨  
        }  
    }  
});
```

메뉴	사이즈	몬도	샷추가	가격
에스프레소	Small Size	Ice	Yes	2000

```
Rbtn1.addActionListener(new ActionListener(){  
    @Override  
    public void actionPerformed(ActionEvent e1)  
    {  
        if(Rbtn1.isSelected())  
        {  
            size = Rbtn1.getText();  
            splus = 0; // 기본 사이즈로써 돈 0원 추가  
        }  
    }  
});
```

```
Rbtn4.addActionListener(new ActionListener(){  
    @Override  
    public void actionPerformed(ActionEvent e1)  
    {  
        if(Rbtn4.isSelected())  
        {  
            tp = Rbtn4.getText();  
        }  
    }  
});
```

004 기능 설명 - 테이블

메뉴	사이즈	온도	샷추가	가격
에스프레소	Small Size	Ice	Yes	2000

```
btn14.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        try{  
            Object[] r_add = {menu[0],size,tp,shot,(price[0]+splus+shplus)};  
            totalMenu.add(new MainM(menu[0],size,tp,shot,(price[0]+splus+shplus)));  
  
            m.addRow(r_add);  
  
            prsum=prsum+(int)(price[prnum]+splus+shplus);  
            x=Integer.toString(prsum);  
            lb5.setText(x+y);  
        } catch (NullPointerException e1) {  
            JOptionPane.showMessageDialog(null, "메뉴를 선택해주세요", "Message", JOptionPane.ERROR_MESSAGE);  
            e1.printStackTrace();  
        }  
    }  
});
```

004 기능 설명 - 테이블 삭제 / 결제

```
btn16.addActionListener(new ActionListener() {
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
try{
```

```
    if(table.getSelectedRow() != -1) {
```

```
        int row = table.getSelectedRow();
```

```
        Object value = table.getValueAt(row, 4);
```

```
        prsum = prsum - (int)value;
```

```
        x=Integer.toString(prsum);
```

```
        lb5.setText(x+y);
```

```
        m.removeRow(table.getSelectedRow()); // 선택 삭제
```

```
    }
```

```
}
```

```
catch(Exception e1){
```

```
    e1.printStackTrace();
```

```
};
```

```
}
```

```
});
```

선택 취소

```
btn17.addActionListener(new ActionListener() {
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
    m.setRowCount(0);
```

```
    prsum = 0;
```

```
    x=Integer.toString(prsum);
```

```
    lb5.setText(x);
```

```
}
```

```
});
```

전체 취소

카드

```
btn18.setBorder(new LineBorder(Color.black));

btn18.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int result = JOptionPane.showConfirmDialog(null, "결제하시겠습니까?",
            "Confirm", JOptionPane.YES_NO_OPTION);
        if(result == JOptionPane.CLOSED_OPTION) {
            jt.setText("Just Closed without Selection");
        } else if(result == JOptionPane.YES_OPTION) {
            jt.setText("네");
            JOptionPane.showMessageDialog(null, "결제가 완료되었습니다.",
                "닫기", JOptionPane.CLOSED_OPTION);
            m.setRowCount(0);
            prsum = 0;
            x=Integer.toString(prsum);
            lb5.setText(x);
        } else {
            jt.setText("아니요");
        }
    }
});
```

현금 결제

```

btn20.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int result = JOptionPane.showConfirmDialog(null, "결제하시겠습니까?",
            "Confirm", JOptionPane.YES_NO_OPTION);
        if(result == JOptionPane.CLOSED_OPTION) {
            jt.setText("Just Closed without Selection");
        } else if(result == JOptionPane.YES_OPTION) {
            jt.setText("네");
            JOptionPane.showMessageDialog(null, "결제가 완료되었습니다.",
                "닫기", JOptionPane.CLOSED_OPTION);
            m.setRowCount(0);
            prsum = 0;
            x=Integer.toString(prsum);
            lb5.setText(x);
        } else {
            jt.setText("아니요");
        }
    }
});

```

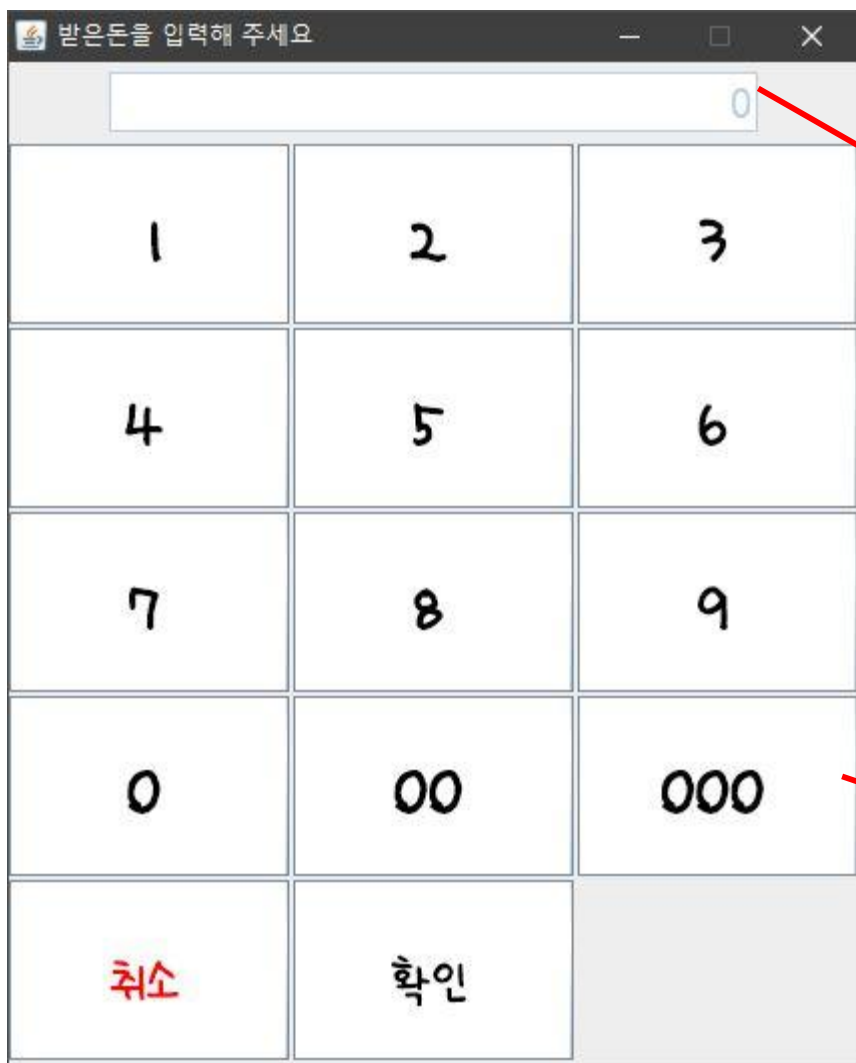
현금 입력

```

btn19.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        cal.calculator(jframe);
    }
});

```

004 기능 설명 - 숫자패드 / 현금 / 거스름



```
jt = new JTextField("0",20); // 초기 0출력, 크기20
```

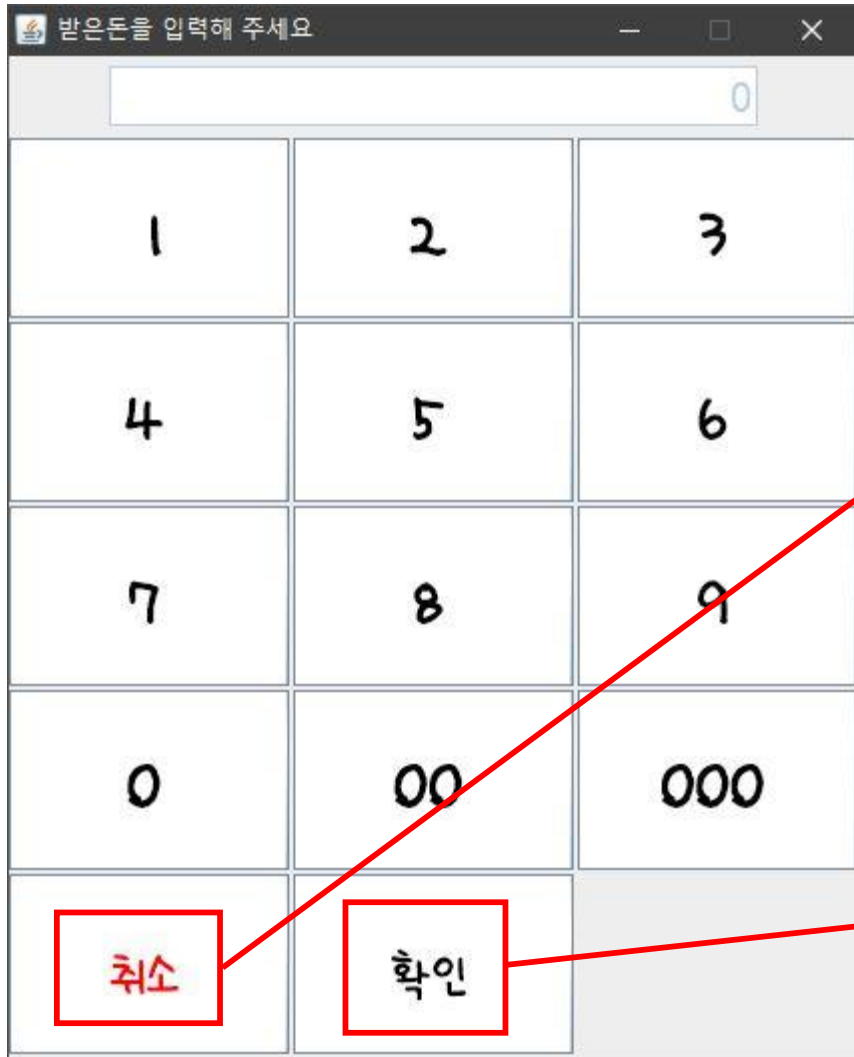
```
jt.setHorizontalAlignment(JTextField.RIGHT); // 초기0을 오른쪽으로 정렬
```

```
JPanel jtpan = new JPanel();  
jtpan.add(jt);  
Font bigFont = jt.getFont().deriveFont(Font.PLAIN, 20f);  
jt.setFont(bigFont);  
frame.add(jtpan);  
jt.disable();  
JCal = new JPanel();  
JCal.setLayout(new GridLayout(5,4,2,2));
```

```
JCal.add(bt000);  
bt000.addMouseListener(new MouseAdapter() {
```

```
    @Override  
    public void mousePressed(MouseEvent e) {  
        num += "000";  
        System.out.println("000");  
        jt.setText(num);  
    }  
});
```


004 기능 설명 - 숫자패드 / 현금 / 거스름



```
JCal.add(cancel);  
cancel.addMouseListener(new MouseAdapter() {
```

```
    @Override  
    public void mousePressed(MouseEvent e) {  
        num = "";  
        System.out.println("cancel");  
        jt.setText(num);  
    }  
});
```

```
JCal.add(ent);  
ent.addMouseListener(new MouseAdapter() {
```

```
    @Override  
    public void mousePressed(MouseEvent e) {  
        MainM mm = new MainM();  
        MainV mv = new MainV();  
  
        mm.setCashprice(jt.getText());  
  
        a = mm.getCashprice();  
  
        mv.lb7.setText(a+b);  
        System.out.println("ent");  
        j.dispose();  
        mv.MainView();  
        frame.hide();  
    }  
});
```

004 기능 설명 - 숫자패드 / 현금 / 거스름

 에스프레소 Espresso 1500원	 아메리카노 Americano 2000원	 아포카토 Affogato 5000원	Coffee Size <input type="radio"/> Small Size <input type="radio"/> Regular Size <input type="radio"/> Large Size Ice / Hot <input type="radio"/> Ice <input type="radio"/> Hot Shot Add <input type="radio"/> Yes <input type="radio"/> No <div style="text-align: center;">← →</div> <div>추 가 매출 확인</div> <div>선택 취소 전체 취소</div> <div>카 드 현금 입력 현금 결제</div>
 카페 콘파냐 Cafe Con Panna 4500원	 카페 크림 Cafe Creme 4500원	 카페 라떼 Cafe Latte 2500원	
 카페 마키야토 Cafe Macchiato 3500원	 카페 모카 Cafe Mocha 3000원	 카푸치노 Cappuccino 3000원	

메뉴	사이즈	온도	샷 추가	가격

결제 금액	0원
현금	0원
거스름 돈	0원

lb7

```
try {
    cash = lb7.getText();
    cash = cash.replace("원", "");
    receivecash = Integer.parseInt(cash); // 손님이 나한테 준돈을 입력해줌
    changemoney ← receivecash - prsum; // 준돈 - 결제금액
    cashcang = Integer.toString(changemoney); // 현금잔에 손님이 준 돈 입력해서 확인 누르면 뜨는 창
    lb9.setText(cashcang + y);
}
catch(Exception e2)
{
    JOptionPane.showMessageDialog(null, "다시 입력해주세요", "Message", JOptionPane.ERROR_MESSAGE);
}
```

004 기능 설명 - 매출 리스트 / 판매량 리스트

< 현재 판매 리스트 >

메뉴	사이즈	온도	샷 수	가격
카푸치노	Small Size	Yes	Ice	3500
카푸치노	Small Size	Yes	Ice	3500
카푸치노	Small Size	Yes	Ice	3500
카푸치노	Small Size	Yes	Ice	3500

조회

< 메뉴별 판매량 >

메뉴	온도	사이즈	가격
카페 크림	Yes	Small Size	5000
카페 크림	Yes	Small Size	5000
카페 크림	Yes	Small Size	5000
카페 크림	Yes	Small Size	5000

조회

004 기능 설명 - 매출 리스트 / 판매량 리스트

```
private ArrayList<MainM> totalMenu = new ArrayList<>();
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
    try{
```

```
        Object[] sivar = {menu[0],size,tp,shot,(price[0]+splus+shplus)};
```

```
        totalMenu.add(new MainM(menu[0],size,tp,shot,(price[0]+splus+shplus)));
```

```
        m.addRow(sivar);
```

```
        prsum=prsum+(int)(price[prnum]+splus+shplus);
```

```
        x=Integer.toString(prsum);
```

```
        lb5.setText(x+y);
```

```
    } catch (NullPointerException e1) {
```

```
        JOptionPane.showMessageDialog(null, "메뉴를 선택해주세요", "Message", JOptionPane.ERROR_MESSAGE);
```

```
        e1.printStackTrace();
```

```
    }
```

```
}
```

```
});
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
    System.out.println(11);
```

```
    for(int i = 0; i < totalMenu.size(); i++ ) {
```

```
        System.out.println(totalMenu.get(i).getName());
```


```
        Object[] data1 = {totalMenu.get(i).getName(),totalMenu.get(i).getSize(),totalMenu.get(i).getTemp(),  
                           totalMenu.get(i).getShot(),totalMenu.get(i).getPrice()};
```

```
        m1.addRow(data1);
```

```
    }
```

```
}
```

```
});
```


A top-down view of four white ceramic coffee cups arranged in a square on a dark, textured wooden tray. Each cup contains a latte with intricate latte art. The cups have dark blue handles. A semi-transparent, light beige circle is centered over the image, containing the text.

**Thank You
&
Q & A**