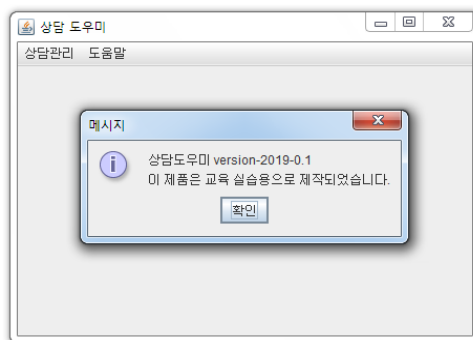
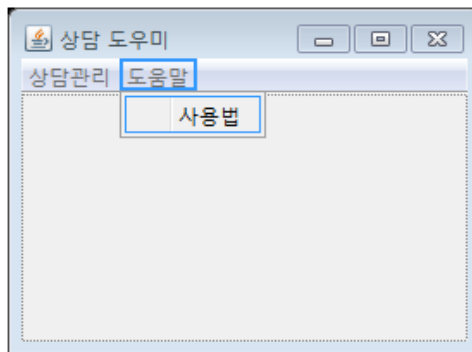
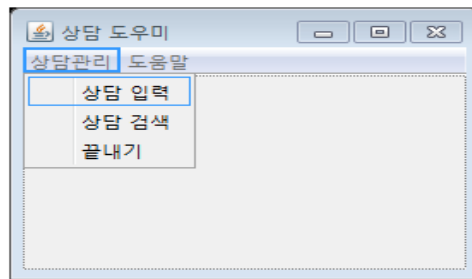


1. (상담관리프로그램 실행 예시) 다음은 상담도우미 프로그램의 실행 화면 예시를 보인 것이다.



상담도우미

상담관리 도움말

상담입력

상담신청자 이름  거주지역  성별 ☐ 남자 ☐ 여자 ☐ 외국인 여부

상담내용

메시지

상담 신청자 이름을 입력하세요.

확인

새 상담 입력 상담 저장

상담도우미

상담관리 도움말

상담입력

상담신청자 이름  거주지역  성별 ☐ 남자 ☐ 여자 ☐ 외국인 여부

상담내용

메시지

상담 신청자 성별을 선택하세요.

확인

새 상담 입력 상담 저장

상담도우미

상담관리 도움말

상담입력

상담신청자 이름  거주지역  성별 ☐ 남자 ☒ 여자 ☐ 외국인 여부

상담내용

메시지

상담 내용을 입력하세요.

확인

새 상담 입력 상담 저장

상담도우미

상담관리 도움말

상담입력

상담신청자 이름  거주지역  성별 ☐ 남자 ☒ 여자 ☐ 외국인 여부

상담내용

전공 학술 동아리 활동 문의

메시지

상담 내용 저장 완료

확인

새 상담 입력 상담 저장

상담도우미

상담관리 도움말

상담검색

상담일시  검색

상담일시	상담자 이름	거주지역	성별	외국인 여부	상담 내용
2019-11-12 01:44:15 (화)	이영희	서울	여자	한국인	전공 학습 동...
2019-11-12 01:46:56 (화)	김철수	부산	남자	한국인	소프트웨어 프...
2019-11-12 01:48:18 (화)	윌리엄 스미스	기타	남자	외국인	한국 문화 소...

Page 1 / 1 이전 다음

상담도우미

상담관리 도움말

상담검색

상담자 이름  이영희 검색

상담일시	상담자 이름	거주지역	성별	외국인 여부	상담 내용
2019-11-12 01:44:15 (화)	이영희	서울	여자	한국인	전공 학습 동...

Page 1 / 1 이전 다음

상담도우미

상담관리 도움말

상담검색

상담자 이름  검색

상담일시	상담자 이름	거주지역	성별	외국인 여부	상담 내용
2019-11-12...	이영희	서울	여자	한국인	전공 학습 ...
2019-11-12...	김철수	부산	남자	한국인	소프트웨어...
2019-11-12...	윌리엄 스...	기타	남자	외국인	한국 문화 ...

상세보기

메시지

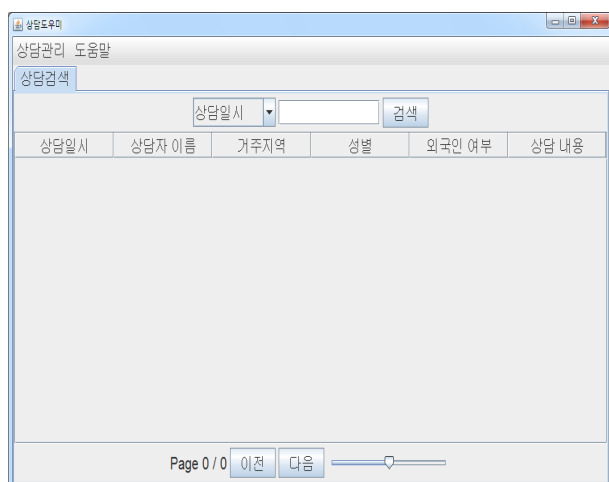
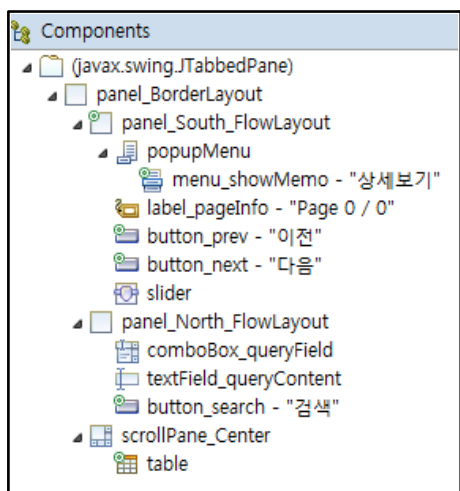
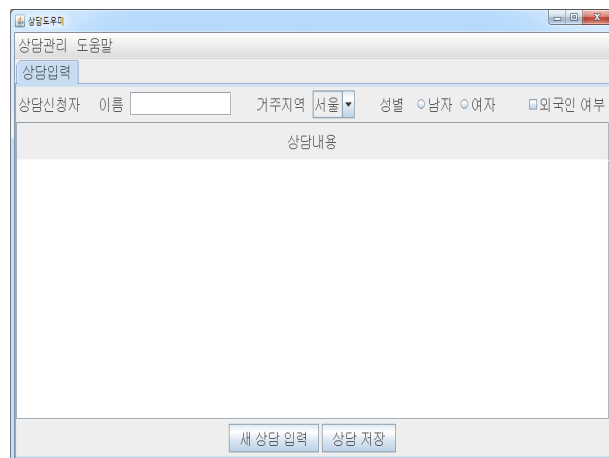
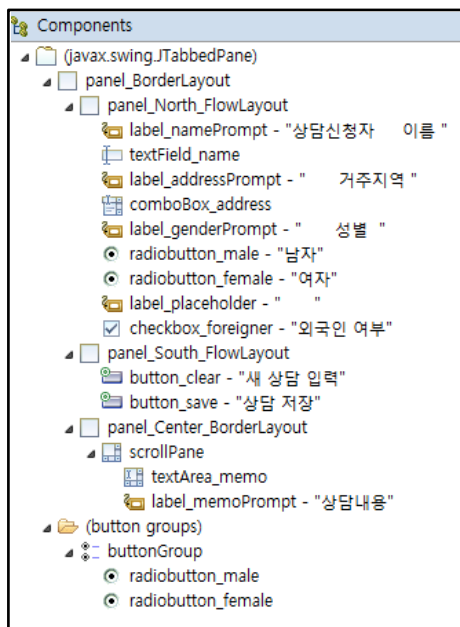
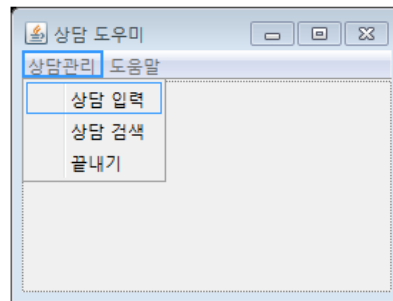
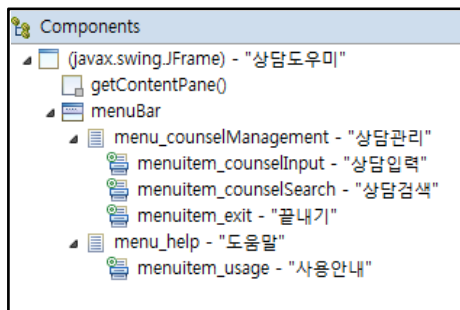
상담 내용

한국 문화 소개 및 체험 프로그램 문의

확인

Page 1 / 1 이전 다음

2. (화면설계) 다음은 1 번에 제시한 상담관리프로그램의 화면 구성을 위한 WindowBuilder 컴포넌트 계층 구조를 보인 것이다. 아래 내용을 참조하여 상담도우미 프로그램의 화면을 구성하시오.



3. (텍스트 파일 출력) 다음은 C:/Temp/data.txt 파일의 끝에 아래 박스 내용의 텍스트를 추가하는 코드이다. C:/Temp/data.txt 파일이 존재하지 않는 경우 새 파일을 만들어진다.

홍길동	부산
이영희	김해
김철수	울산

- 실습 #1: 아래 코드를 여러 번 실행하면서 C:/Temp/data.txt 파일의 내용이 어떻게 변하는지 확인하시오.

- 실습 #2: 아래 코드의 FileWriter 생성자의 두 번째 인자(append 여부) 값 true 를 false 로 변경한 후, 아래 코드를 여러 번 실행하면서 C:/Temp/data.txt 파일의 내용이 어떻게 변하는지 확인하시오.

- 실습 #3: 아래 코드에서 oF.close(); 문장을 주석 처리하고 아래 코드를 실행한 경우, C:/Temp/data.txt 파일의 내용이 어떻게 변하는지 확인하시오.

- 실습 #4: 아래 코드는 C:/Temp 폴더가 존재하지 않는 경우 익셉션이 발생한다. 익셉션 상황을 인위적으로 만들기 위해 C:/Temp/data.txt 를 Z:/Temp/data.txt 로 변경한 후 아래 코드를 실행하고 어떤 익셉션이 발생하는지 확인하시오.

- 실습 #5: 아래 코드의 FileWriter 객체 생성 문장을 주석 처리하고 그 아래 주석처리된 문장의 주석을 해제한 후 실행해 보고 C:/Temp/data.txt 파일의 크기 및 내용을 그 전과 비교해 보시오. Charset.forName("UTF-8")을 인자로 전달하여 생성된 FileWriter 객체는 UTF-8 인코딩을 사용하여 파일에 쓰기 작업을 수행한다.

```
public class Test {  
    public static void main(String[] args) {  
        try {  
            FileWriter oF=new FileWriter("C:/Temp/data.txt", true);  
            //FileWriter oF=new FileWriter("C:/Temp/data.txt", Charset.forName("UTF-8"), true);  
            String v="홍길동"+"\\t"+"부산"+"\\n";  
            oF.write(v);  
            oF.write("이영희"+"\\t"+"김해"+"\\n");  
            oF.write("김철수"+"\\t"+"울산"+"\\n");  
            oF.close();  
        } catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

4. (텍스트 파일 입력) 다음은 이전 문제에서 작성한 C:/Temp/data.txt 파일의 각 행을 출력하는 코드이다. 아래 코드의 `iF.readLine()`은 파일에서 읽은 한 행을 줄바꿈문자들이 제거된 문자열로 반환하며, 파일의 끝에 도달한 경우 `null`을 반환한다. 아래 코드는 C:/Temp/data.txt의 각 행이 다음 박스의 내용과 같이 탭으로 분리된 두 개 필드로 이루어져 있다고 가정한다. 즉 아래 박스 내 첫 행에서 홍길동과 부산 사이에는 하나의 탭(`\t`) 문자가 저장되어 있다.

C:/Temp/data.txt

홍길동	부산
이영희	김해
김철수	울산

- **실습 #1:** 아래 코드는 C:/Temp/data.txt 파일의 모든 행을 출력한다. C:/Temp/data.txt의 각 행에 대해 행의 문자열 내에 "김"이 포함된 행(들)만을 출력하도록 아래 코드를 수정하십시오.

- **실습 #2:** 아래 코드는 C:/Temp/data.txt 파일의 모든 행을 출력한다. C:/Temp/data.txt의 각 행에 대해 행의 첫번째 필드 문자열의 첫 문자가 "김"인 포함된 행(들)만을 출력하도록 아래 코드를 수정하십시오.

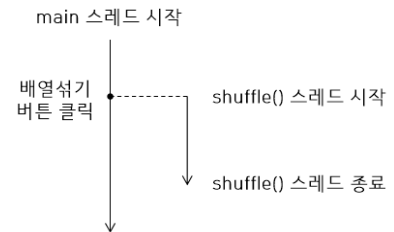
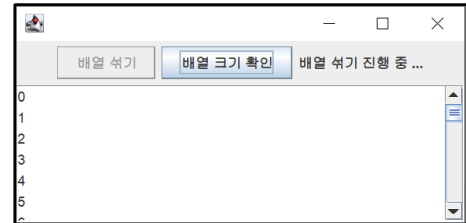
- **실습 #3:** 아래 코드의 `FileReader` 객체 생성 문장을 아래 문장과 같이 `Charset.forName("UTF-8")`을 인자로 사용하도록 변경한 후 실행해 보시오. 이 경우, C:/Temp/data.txt 파일의 문자인코딩이 UTF-8이 아닌 경우 파일에서 읽은 행의 내용이 올바르게 표시되지 않는다.

```
FileReader fr=new FileReader("C:/Temp/data.txt", Charset.forName("UTF-8"));
```

```
public class Test {
    public static void main(String[] args) {
        try {
            FileReader fr=new FileReader("C:/Temp/data.txt");
            BufferedReader iF=new BufferedReader(fr);
            // BufferedReader iF=new BufferedReader(new FileReader("C:/Temp/data.txt"));
            while(true) {
                String line=iF.readLine(); // 줄바꿈문자 제거된 문자열 반환
                if(line==null) break;
                String v[]=line.split("\t");
                System.out.println(v[0]+" "+v[1]);
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

5. (실습: 스레드: 프로세스 내에서의 실행의 흐름) 다음은 멀티스레드 코드의 예로 배열섞기 버튼 클릭 시 수행되는 shuffle() 메소드를 스레드로 수행(아래 파란색코드부분)하므로 배열섞기 작업 중에도 다른 작업(예: 배열크기확인 버튼 클릭)의 동시? 수행이 가능하다. 아래 thread.start();는 Runnable 객체 내 run() 메소드를 호출한다. 아래 파란색 코드 부분을 주석처리하고, 그 아래 주석처리된 shuffle(); 호출문의 주석을 해제한 후 실행(이 경우 다중 작업 동시 수행 불가)해 보고 그 차이를 확인하시오. 인터페이스 Runnable 은 추상메소드 void run();을 포함한다. 스레드 실행의 한 방법은 아래 코드 예와 같이 (1) 스레드로 실행할 코드를 포함하도록 Runnable 인터페이스의 run() 메소드를 구현하고 (2) run() 메소드가 구현된 Runnable 객체를 인자로 갖는 Thread 객체를 생성한 후 (3) 생성된 Thread 객체의 start() 메소드를 호출하는 것이다.

```
public class Test extends JFrame {
    int v[]=new int[10000];
    JLabel label=new JLabel();
    JTextArea textArea=new JTextArea();
    JButton button_shuffle=new JButton("배열 섞기");
    JButton button_size=new JButton("배열 크기 확인");
    public Test() {
        setSize(400,400);
        JPanel panel=new JPanel();
        panel.add(button_shuffle);
        panel.add(button_size);
        panel.add(label);
        for (int i = 0; i < v.length; i++) v[i]=i;
        for (int i = 0; i < v.length; i++) textArea.append(v[i]+"\\n");
        add(panel,BorderLayout.NORTH);
        add(new JScrollPane(textArea),BorderLayout.CENTER);
        button_size.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(null, "배열 크기: "+v.length);
            }
        });
        button_shuffle.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                Thread thread=new Thread(new Runnable() {
                    @Override
                    public void run() { shuffle(); }
                });
                thread.start();
                //shuffle();
            }
        });
    }
    protected void shuffle() {
        button_shuffle.setEnabled(false);
        label.setText("배열 섞기 진행 중 ...");
        Random random=new Random();
        for (int i = 0; i < v.length*v.length; i++) {
            int p1=random.nextInt(v.length), p2=random.nextInt(v.length);
            int temp=v[p1]; v[p1]=v[p2]; v[p2]=temp;
        }
        label.setText("");
        textArea.setText("");
        for (int i = 0; i < v.length; i++) textArea.append(v[i]+"\\n");
        button_shuffle.setEnabled(true);
    }
    public static void main(String[] args) { new Test().setVisible(true); }
}
```



6. (실습) 다음 코드의 두 문장 `do_multiply(n,1);`와 `do_multiply(n,-1);`은 순차 실행된다. 아래 두 문장 `do_multiply(n,1);`와 `do_multiply(n,-1);`이 멀티스레드로 동시 실행되도록 아래 코드를 수정하시오.

```
public class Test {
    public static void main(String[] args) {
        int n[]=new int[100];
        for (int i = 0; i < n.length; i++) n[i]=i+1;
        do_multiply(n,1);
        do_multiply(n,-1);
    }
    private static void do_multiply(int[] v, int m) { // 배열 v의 각 값에 m을 곱한 값을 출력
        for (int i = 0; i < v.length; i++) {
            System.out.println(v[i]*m);
        }
    }
}
```

7. (날짜시간 표현, SimpleDateFormat) 다음은 현재 시각에 해당하는 연월일시를 문자열로 생성하여 출력하는 코드이다.

```
public class Test {
    public static void main(String[] args) {
        SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd hh:mm:ss (E)");
        String date=sdf.format(new Date()); // import java.util.Date;
        System.out.println(date);
    }
}
```

8. 다음은 문자열 memo 내 연속된 공백문자열("\t", "\r", "\n", ' ' 등)을 하나의 공백으로 변환하는 코드이다. 아래 코드에서 정규표현식 `\\s+`는 길이 1 이상의 연속된 공백문자들의 문자열을 의미한다.

```
public class Test {
    public static void main(String[] args) {
        String memo="전공 학습 방법 문의\n프로그래밍 실력\t향상 방법 문의";
        memo=memo.replaceAll("\\s+", " ");
        System.out.println(memo);
    }
}
```

9. (`System.exit()`, `JOptionPane.showConfirmDialog()`) 다음은 `System.exit()` 및 YES/NO 선택창 사용법에 대한 예시 코드이다.

```
public class Test extends JFrame {
    public static void main(String[] args) {
        int result=JOptionPane.showConfirmDialog(null, "계속하시겠습니까?", "실행 확인", JOptionPane.YES_NO_OPTION);
        if(result==JOptionPane.YES_OPTION) new Test().setVisible(true);
        else System.exit(-1);
    }
}
```



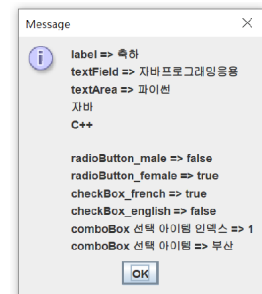
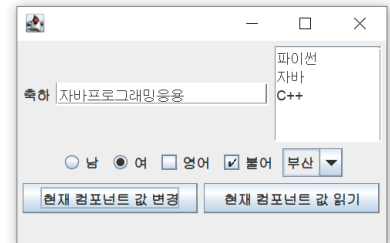
10. 다음은 여러 Swing 컴포넌트들의 객체 생성, 컨테이너 내 배치, 값 변경, 값 읽기 방법에 대한 예시 코드이다.

```
public class Test {
    public static void main(String[] args) {
        JFrame w=new JFrame();
        w.setLayout(new FlowLayout());
        JLabel label=new JLabel("안녕");
        JTextField textField=new JTextField(20);
        JTextArea textArea=new JTextArea(5, 10);
        JRadioButton radioButton_male=new JRadioButton("남");
        JRadioButton radioButton_female=new JRadioButton("여");
        ButtonGroup buttonGroup=new ButtonGroup();
        buttonGroup.add(radioButton_male);
        buttonGroup.add(radioButton_female);
        JCheckBox checkBox_english=new JCheckBox("영어");
        JCheckBox checkBox_french=new JCheckBox("불어");
        String fields[]= {"서울", "부산", "제주"};
        JComboBox<String> comboBox=new JComboBox<>(fields);
        w.add(label);
        w.add(textField);
        w.add(new JScrollPane(textArea));
        w.add(radioButton_male);
        w.add(radioButton_female);
        w.add(checkBox_english);
        w.add(checkBox_french);
        w.add(comboBox);

        JButton button_write=new JButton("현재 컴포넌트 값 변경");
        button_write.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                label.setText("축하");
                textField.setText("자바프로그래밍응용");
                textArea.setText("파이썬\n자바\nC++");
                radioButton_female.setSelected(true);
                checkBox_french.setSelected(true);
                checkBox_english.setSelected(false);
                comboBox.setSelectedIndex(1);
            }
        });

        JButton button_read=new JButton("현재 컴포넌트 값 읽기");
        button_read.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String v="label => "+label.getText()+"\n";
                v+="textField => "+textField.getText()+"\n";
                v+="textArea => "+textArea.getText()+"\n";
                v+="radioButton_male => "+radioButton_male.isSelected()+"\n";
                v+="radioButton_female => "+radioButton_female.isSelected()+"\n";
                v+="checkBox_french => "+checkBox_french.isSelected()+"\n";
                v+="checkBox_english => "+checkBox_english.isSelected()+"\n";
                v+="comboBox 선택 아이템 인덱스 => "+comboBox.getSelectedIndex()+"\n";
                v+="comboBox 선택 아이템 => "+comboBox.getSelectedItem().toString()+"\n";
                JOptionPane.showMessageDialog(null, v);
            }
        });

        w.add(button_write);
        w.add(button_read);
        w.pack();
        w.setVisible(true);
    }
}
```

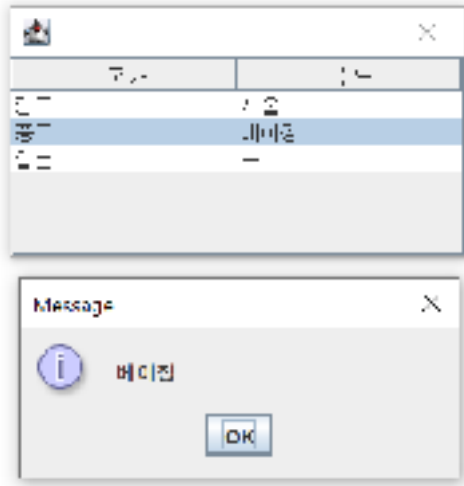


11. (JTable, DefaultTableModel)다음은 **행 추가** 버튼 클릭 시 테이블에 행을 추가하는 코드이다.



```
public class Test {
    public static void main(String[] args) {
        JFrame w=new JFrame();
        String fields[]= {"국가", "수도"};
        DefaultTableModel model=new DefaultTableModel(fields, 0);
        JTable table=new JTable(model);
        w.add(new JScrollPane(table), BorderLayout.CENTER);
        JButton button_insert=new JButton("행 추가");
        JButton button_clear=new JButton("모든 행 삭제");
        JPanel panel=new JPanel();
        panel.add(button_insert);
        panel.add(button_clear);
        w.add(panel, BorderLayout.NORTH);
        button_insert.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                DefaultTableModel model=(DefaultTableModel) table.getModel();
                String s="한국\t서울";
                String v[]=s.split("\t");
                model.addRow(v);
                table.setModel(model);
            }
        });
        button_clear.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                DefaultTableModel model=(DefaultTableModel) table.getModel();
                model.setRowCount(0);
                table.setModel(model);
            }
        });
        w.pack();
        w.setVisible(true);
    }
}
```

12. (JPopupMenu) 다음은 테이블 행 위에서 왼쪽 마우스 클릭 시 팝업 메뉴를 표시하고, 팝업 메뉴 내 **상세내용** 메뉴 아이템 클릭 시 마우스 클릭된 행 위치의 두번째 필드 값(수도 명)을 메시지 창에 출력하는 코드이다.



```
public class Test {
    public static void main(String[] args) {
        JFrame w=new JFrame();
        JTable table=new JTable();
        JPopupMenu popupMenu=new JPopupMenu();
        JMenuItem menuItem=new JMenuItem("상세내용");
        menuItem.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                int row=table.getSelectedRow();
                JOptionPane.showMessageDialog(null, table.getModel().getValueAt(row, 1));
            }
        });
        popupMenu.add(menuItem);
        String fields[]= {"국가", "수도"};
        DefaultTableModel model=new DefaultTableModel(fields, 0);
        model.addRow("한국\t서울".split("\t"));
        model.addRow("중국\t베이징".split("\t"));
        model.addRow("일본\t도쿄".split("\t"));
        table.setModel(model);
        table.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                if(e.getButton()!=MouseEvent.BUTTON1) return;
                popupMenu.show(e.getComponent(), e.getX(), e.getY());
            }
        });
        w.add(new JScrollPane(table), BorderLayout.CENTER);
        w.pack();
        w.setVisible(true);
    }
}
```

13. (실습: 파일입출력 기반 상담도우미 제작) 앞 문제 1 번에서 제시한 상담도우미 프로그램과 같이 동작하는 자바코드를 작성하시오. 상담 기록을 memo.txt 파일에 저장하시오.

- A. 상담관리 메뉴의 **끝내기** 메뉴 클릭 시 프로그램을 종료한다.
- B. 도움말 메뉴의 **사용법** 메뉴 클릭 시 프로그램에 대한 소개 정보를 표시하는 메시지 창을 출력한다.
- C. 상담입력 화면
  - i. **상담저장** 버튼 클릭 시
    - 1. 이름, 거주지역, 상담 내용 등 입력 값 누락된 경우 → 오류 메시지 출력
    - 2. 정상 입력된 경우 → 상담일시, 이름, 거주지역, 성별, 외국인 여부, 상담내용의 6개 필드 값들을 탭 구분자로 연결된 하나의 행으로 만들어 memo.txt 파일의 끝에 추가한다.
  - ii. **새상담입력** 버튼 클릭 시 → 기존 입력된 값들을 초기화한다.
- D. 상담검색 화면
  - i. **검색** 버튼 클릭 시 → 상담자 이름이 입력되지 않은 경우 memo.txt 파일의 각 행을 테이블에 표시한다. 상담자 이름이 입력된 경우 상담자 이름이 포함된 상담 데이터들을 테이블에 표시한다.

14. 다음은 이클립스에서 제작한 자바프로젝트를 jar 파일로 저장한 후 실행하는 예시이다.

A. Runnable jar 파일 만들기: Export 메뉴 → Java → Runnable JAR file → Launch configuration 에서 export 할 프로젝트 및 실행 시작 클래스 선택 → Export destination 에 폴더 및 jar 파일명 선택 → Finish

i. Runnable jar 파일 실행하기: CounselHelper.jar 파일이 C:/Temp 폴더에 저장되었다고 가정하고, 윈도우즈 명령창에서 아래 명령을 실행한다.

```
java -jar CounselHelper.jar
```

B. 일반 jar 파일 만들기: Export 메뉴 → Java → JAR file → export 할 프로젝트 선택 → Export destination 에 저장 폴더 및 jar 파일명 선택 → Finish

i. 일반 jar 파일 실행하기: CounselHelper.jar 파일이 C:/Temp 폴더에 저장되어 있으며 CounselHelper.jar 내 실행 시작 클래스가 CounselMain.class 라고 가정하고, 윈도우즈 명령창에서 아래 명령을 실행한다.

```
java -cp C:\Temp\CounselHelper.jar CounselMain
```

C. 다음 내용을 CounselHelper.bat 파일에 저장한 후 CounselHelper.bat 파일을 더블클릭하여 실행한다.

```
start /B javaw -jar CounselHelper.jar
```

## References

김윤명. (2008). 뇌를 자극하는 Java 프로그래밍. 한빛미디어.

남궁성. 자바의 정석. 도우출판.

김윤명. (2010). 뇌를 자극하는 JSP & Servlet. 한빛미디어.