

# Chapter 15

---

입출력

I/O

## [ 연습문제 ]

**[15-1]** 커맨드라인으로 부터 파일명과 숫자를 입력받아서, 입력받은 파일의 내용의 처음  
부터 입력받은 숫자만큼의 라인을 출력하는 프로그램(FileHead.java)을 작성하라.

[Hint] BufferedReader의 readLine()을 사용하라.

### [실행결과]

```
C:\jdk1.8\work\ch15>java FileHead 10
USAGE: java FileHead 10 FILENAME

C:\jdk1.8\work\ch15>java FileHead 10 aaa
aaa은/는 디렉토리이거나, 존재하지 않는 파일입니다.

C:\jdk1.8\work\ch15>java FileHead 10 FileHead.java
1:import java.io.*;
2:
3:class FileHead
4:{
5:    public static void main(String[] args)
6:    {
7:        try {
8:            int line = Integer.parseInt(args[0]);
9:            String fileName = args[1];
10:

C:\jdk1.8\work\ch15>
```

**[15-2]** 지정된 이진파일의 내용을 실행결과와 같이 16진수로 보여주는 프로그램  
(HexaViewer.java)을 작성하라.

[Hint] PrintStream과 printf()를 사용하라.

### [실행결과]

```
C:\jdk1.8\work\ch15>java HexaViewer HexaViewer.class
CA FE BA BE 00 00 00 31 00 44 0A 00 0C 00 1E 09
00 1F 00 20 08 00 21 0A 00 08 00 22 0A 00 1F 00
23 07 00 24 0A 00 06 00 25 07 00 26 0A 00 08 00
27 0A 00 06 00 28 08 00 29 07 00 2A 0A 00 2B 00
2C 0A 00 08 00 2D 0A 00 08 00 2E 0A 00 06 00 2F
0A 00 08 00 2F 07 00 30 0A 00 12 00 31 07 00 32
01 00 06 3C 69 6E 69 74 3E 01 00 03 28 29 56 01
00 04 43 6F 64 65 01 00 0F 4C 69 6E 65 4E 75 6D
62 65 72 54 61 62 6C 65 01 00 04 6D 61 69 6E 01
00 16 28 5B 4C 6A 61 76 61 2F 6C 61 6E 67 2F 53
... 중간생략

C:\jdk1.8\work\ch15>
```

**[15-3]** 다음은 디렉토리의 요약정보를 보여주는 프로그램이다. 파일의 개수, 디렉토리의 개수, 파일의 총 크기를 계산하는 `countFiles()`를 완성하시오.

#### 【연습문제】/ch15/DirectoryInfoTest.java

```
import java.io.*;

class DirectoryInfoTest {
    static int totalFiles = 0;
    static int totalDirs = 0;
    static int totalSize = 0;

    public static void main(String[] args) {
        if(args.length != 1) {
            System.out.println("USAGE : java DirectoryInfoTest DIRECTORY");
            System.exit(0);
        }

        File dir = new File(args[0]);

        if(!dir.exists() || !dir.isDirectory()) {
            System.out.println("유효하지 않은 디렉토리입니다.");
            System.exit(0);
        }

        countFiles(dir);

        System.out.println();
        System.out.println("총 " + totalFiles + "개의 파일");
        System.out.println("총 " + totalDirs + "개의 디렉토리");
        System.out.println("크기 " + totalSize + " bytes");
    } // main

    public static void countFiles(File dir) {
        /*
         * (1) 아래의 로직에 맞게 코드를 작성하시오.
         * 1. dir의 파일목록(File[])을 얻어온다.
         * 2. 얻어온 파일목록의 파일 중에서...
         *     디렉토리이면, totalDirs의 값을 증가시키고 countFiles()를 재귀호출한다.
         * 3. 파일이면, totalFiles를 증가시키고 파일의 크기를 totalSize에 더한다.
         */
    } // countFiles
}
```

#### 【실행결과】

```
C:\jdk1.8\work\ch15>java DirectoryInfoTest .
```

```
총 786개의 파일
총 27개의 디렉토리
크기 2566228 bytes
```

```
C:\jdk1.8\work\ch15>
```

**[15-4]** 커맨드라인으로 부터 여러 파일의 이름을 입력받고, 이 파일들을 순서대로 합쳐서 새로운 파일을 만들어 내는 프로그램(FileMergeTest.java)을 작성하시오. 단, 합칠 파일의 개수에는 제한을 두지 않는다.

**【실행결과】**

```
C:\jdk1.8\work\ch15>java FileMergeTest  
USAGE: java FileMergeTest MERGE_FILENAME FILENAME1 FILENAME2 ...  
  
C:\jdk1.8\work\ch15>java FileMergeTest result.txt 1.txt 2.txt 3.txt  
  
C:\jdk1.8\work\ch15>type result.txt  
1111111111  
2222222222  
33333333333333  
  
C:\jdk1.8\work\ch15>java FileMergeTest result.txt 1.txt 2.txt  
  
C:\jdk1.8\work\ch15>type result.txt  
1111111111  
2222222222  
  
C:\jdk1.8\work\ch15>type 1.txt  
1111111111  
  
C:\jdk1.8\work\ch15>type 2.txt  
2222222222  
  
C:\jdk1.8\work\ch15>type 3.txt  
33333333333333  
  
C:\jdk1.8\work\ch15>
```

**[15-5]** 다음은 `FilterWriter`를 상속받아서 직접 구현한 `HtmlTagFilterWriter`를 사용해서 주어진 파일에 있는 태그를 모두 제거하는 프로그램이다. `HtmlTagFilterWriter`의 `write()`가 태그를 제거하도록 코드를 완성하시오.

```
[연습문제]/ch15/Exercise15_5.java
import java.io.*;

class Exercise15_5
{
    public static void main(String[] args)
    {
        if(args.length != 2) {
            System.out.println("USAGE:     java     Exercise15_5     TARGET_FILE");
            System.exit(0);
        }

        String inputFile = args[0];
        String outputFile = args[1];

        try {
            BufferedReader input
                = new BufferedReader(new FileReader(inputFile));
            HtmlTagFilterWriter output
                = new HtmlTagFilterWriter(new FileWriter(outputFile));

            int ch = 0;

            while((ch=input.read())!=-1) {
                output.write(ch);
            }

            input.close();
            output.close();

        } catch(IOException e) {}
    }
}

class HtmlTagFilterWriter extends FilterWriter {
    StringWriter tmp = new StringWriter();
    boolean inTag = false;

    HtmlTagFilterWriter(Writer out) {
        super(out);
    }

    public void write(int c) throws IOException {
        /*
         * (1) 아래의 로직에 맞게 코드를 작성하시오.
         * 1. 출력할 문자(c)가 '<'이면 inTag의 값을 true로 한다.
         * 2. 출력할 문자(c)가 '>'이면 inTag의 값을 false로 한다.
         *     새로운 StringWriter를 생성한다. (기존 StringWriter의 내용을 버린다.)
         * 3. inTag의 값이 true이면, StringWriter에 문자(c)를 출력하고
        */
    }
}
```

`inTag`의 값이 `false`이면, `out`에 문자(`c`)를 출력한다.

[참고] 태그가 시작되면 `StringWriter`에 출력하고 태그가 끝나면  
`StringWriter`는 비워진다.

```
    */
}

public void close() throws IOException {
    out.write(tmp.toString()); // StringWriter의 내용을 출력하고
    super.close();           // 조상의 close()를 호출해서 기반 스트림을 닫는다.
}
}
```

### 【실행결과】

```
C:\jdk1.8\work\ch15>java Exercise15_5 test.html result.txt
```

```
C:\jdk1.8\work\ch15>type result.txt
```

```
New Document
```

```
> 안녕하세요. 태그 없애기 테스트용 파일입니다.
```

```
< 태그가 열린 채로 끝난 것은 태그로 처리하지 마세요.
```

```
C:\jdk1.8\work\ch15>type test.html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE> New Document </TITLE>
</HEAD>
<BODY>
> 안녕하세요. 태그 없애기 테스트용 파일입니다.
</BODY>
< 태그가 열린 채로 끝난 것은 태그로 처리하지 마세요.
```

**[15-6]** 다음은 콘솔 명령어 중에서 디렉토리를 변경하는 cd명령을 구현한 것이다. 알맞은 코드를 넣어 cd()를 완성하시오.

**[연습문제]/ch15/Exercise15\_6.java**

```
import java.io.*;
import java.util.*;
import java.util.regex.*;

class Exercise15_6 {
    static String[] argArr;      // 입력한 매개변수를 담기위한 문자열배열
    static File curDir;          // 현재 디렉토리

    static {
        try {
            curDir = new File(System.getProperty("user.dir"));
        } catch(Exception e) {}
    }

    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        while(true) {
            try {
                String prompt = curDir.getCanonicalPath() + ">>";
                System.out.print(prompt);

                // 화면으로부터 라인단위로 입력받는다.
                String input = s.nextLine();

                input = input.trim(); // 입력받은 값에서 불필요한 앞뒤 공백을 제거한다.
                argArr = input.split(" +");

                String command = argArr[0].trim();

                if("").equals(command)) continue;

                command = command.toLowerCase(); // 명령어를 소문자로 바꾼다.

                if(command.equals("q")) { // q 또는 Q를 입력하면 실행종료한다.
                    System.exit(0);
                } else if(command.equals("cd")) {
                    cd();
                } else {
                    for(int i=0; i < argArr.length;i++) {
                        System.out.println(argArr[i]);
                    }
                }
            } catch(Exception e) {
                e.printStackTrace();
                System.out.println("입력오류입니다.");
            }
        } // while(true)
    } // main

    public static void cd() {
```

```
if(argArr.length==1) {
    System.out.println(curDir);
    return;
} else if(argArr.length > 2) {
    System.out.println("USAGE : cd directory");
    return;
}

String subDir = argArr[1];

/*
(1) 아래의 로직에 맞게 코드를 작성하시오.
1. 입력된 디렉토리(subDir)가 ".."이면,
   1.1 현재 디렉토리의 조상 디렉토리를 엣어서 현재 디렉토리로 지정한다.
       (File클래스의 getParentFile()을 사용)
2. 입력된 디렉토리(subDir)가 "."이면, 단순히 현재 디렉토리의 경로를 화면에 출력한다.
3. 1 또는 2의 경우가 아니면,
   3.1 입력된 디렉토리(subDir)가 현재 디렉토리의 하위디렉토리인지 확인한다.
   3.2 확인결과가 true이면, 현재 디렉토리(curDir)을 입력된 디렉토리(subDir)로
       변경한다.
   3.3 확인결과가 false이면, "유효하지 않은 디렉토리입니다."고 화면에 출력한다.
*/
} // cd()
}
```

#### 【실행결과】

```
C:\jdk1.8\work\ch15>java Exercise15_6
C:\jdk1.8\work\ch15>>
C:\jdk1.8\work\ch15>>cd ch15
유효하지 않은 디렉토리입니다.
C:\jdk1.8\work\ch15>>cd ..
C:\jdk1.8\work>>cd ch15
C:\jdk1.8\work\ch15>>
C:\jdk1.8\work\ch15>>cd .
C:\jdk1.8\work\ch15
C:\jdk1.8\work\ch15>>q

C:\jdk1.8\work\ch15>
```

**[15-7]** 다음의 코드는 대화내용을 파일에 저장할 수 있는 채팅 프로그램이다. ‘저장’ 버튼을 누르면 대화내용이 저장되도록 알맞은 코드를 넣어 완성하시오.

**[연습문제]/ch15/ChatWin.java**

```
import java.io.*;
import java.awt.*;
import java.awt.event.*;

class ChatWin extends Frame {
    String nickname = "";
    TextArea ta = new TextArea();
    Panel p = new Panel();
    TextField tf = new TextField(30);
    Button bSave = new Button("저장");

    ChatWin(String nickname) {
        super("Chatting");
        this.nickname = nickname;

        setBounds(200, 100, 300, 200);

        p.setLayout(new FlowLayout());
        p.add(tf);
        p.add(bSave);

        add(ta, "Center");
        add(p, "South");

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }

    bSave.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            FileDialog fileSave = new FileDialog(ChatWin.this,
                "파일저장", FileDialog.SAVE);
            fileSave.setVisible(true);
            String fileName = fileSave.getDirectory()
                + fileSave.getFile();
            saveAs(fileName);
        }
    });

    EventHandler handler = new EventHandler();
    ta.addFocusListener(handler);
    tf.addFocusListener(handler);
    tf.addActionListener(handler);

    ta.setText("#" + nickname + "님 즐거운 채팅되세요.");
    ta.setEditable(false);

    setResizable(false);
    setVisible(true);
    tf.requestFocus();
}
```

```
void saveAs(String fileName) {
    /*
     * (1) 알맞은 코드를 넣어 완성하시오.
     */
} // saveAs메서드의 끝

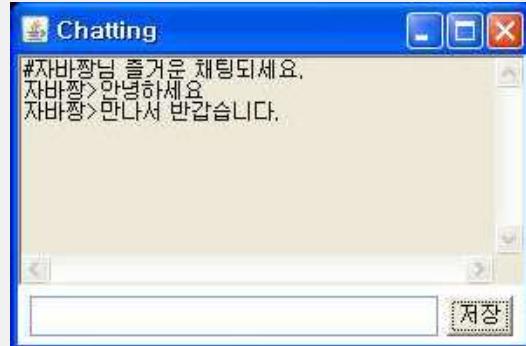
public static void main(String[] args) {
    if(args.length != 1) {
        System.out.println("USAGE : java ChatWin NICKNAME");
        System.exit(0);
    }

    new ChatWin(args[0]);
} // main

class EventHandler extends FocusAdapter implements ActionListener {
    public void actionPerformed(ActionEvent ae) {
        String msg = tf.getText();
        if(".".equals(msg)) return;

        ta.append("\r\n" + nickname + ">" + msg);
        tf.setText("");
    }

    public void focusGained(FocusEvent e) {
        tf.requestFocus();
    }
} // class EventHandler
} // class
```

**【실행결과】**

**[15-8]** 다음의 코드는 파일로부터 한 줄 씩 데이터를 읽어서 보여주는 프로그램이다. 버튼을 이용해서 첫 줄, 다음 줄, 이전 줄, 마지막 줄로 이동할 수 있으며, 각 줄의 개행문자는 ‘|’를 사용한다. (1)~(2)에 알맞은 코드를 넣어 완성하시오.

**【연습문제】/ch15/WordStudy.java**

```
import java.util.*;
import java.io.*;
import java.awt.*;
import java.awt.event.*;

class WordStudy extends Frame {
    Button first = new Button("<<");  
    Button prev = new Button("<");  
    Button next = new Button(">");  
    Button last = new Button(">>");  
    Panel buttons = new Panel();  
    TextArea ta = new TextArea();  
    ArrayList wordList = new ArrayList();

    final String WORD_FILE = "word_data.txt";
    final String CR_LF = System.getProperty("line.separator");

    int pos = 0;

    WordStudy(String title) {
        super(title);

        buttons.add(first);
        buttons.add(prev);
        buttons.add(next);
        buttons.add(last);
        add("South", buttons);
        add("Center", ta);

        EventHandler handler = new EventHandler();
        addWindowListener(handler);
        first.addActionListener(handler);
        prev.addActionListener(handler);
        next.addActionListener(handler);
        last.addActionListener(handler);

        loadFile(WORD_FILE);

        setBackground(Color.BLACK);
        setSize(300, 200);
        setLocation(200, 200);
        setResizable(true);
        setVisible(true);

        showFirst();
    }

    void showFirst() {
        pos = 0;
        display(pos);
    }
}
```

```

void showPrevious() {
    pos = (pos > 0) ? --pos : 0;
    display(pos);
}

void showNext() {
    int size = wordList.size();
    pos = (pos < size-1) ? ++pos : size-1;
    display(pos);
}

void showLast() {
    pos = wordList.size()-1;
    display(pos);
}

void display(int pos) { // pos위치에 있는 라인의 내용을 보여준다.
/*
(1) 아래의 로직에 맞게 코드를 작성하시오.
1. wordList에서 pos번쨰의 위치에 있는 데이터를 읽어온다.
2. StringTokenizer를 이용해서 '|'를 구분자로 자른다.
3. 잘라진 Token에 개행문자(CR_LF)를 붙여서 StringBuffer에 붙인다. (append)
4. StringBuffer의 내용을 뽑아서 TextArea에 보여준다.
*/
}

void loadFile(String fileName) {
/*
(2) 아래의 로직에 맞게 코드를 작성하시오.
1. BufferedReader와 FileReader를 이용해서 파일의 내용을 라인 단위로 읽는다.
2. 읽어온 라인을 wordList에 저장한다.
3. 만일 예외가 발생하면 프로그램을 종료한다.
*/
}

public static void main(String args[]) {
    WordStudy mainWin = new WordStudy("Word Study");
}

class EventHandler extends WindowAdapter implements ActionListener {
    public void actionPerformed(ActionEvent ae) {
        Button b = (Button)ae.getSource();

        if(b==first) {
            showFirst();
        } else if(b==prev) {
            showPrevious();
        } else if(b==next) {
            showNext();
        } else if(b==last) {
            showLast();
        }
    }

    public void windowClosing(WindowEvent e) {
}

```

```
        System.exit(0);
    }
} // class EventHandler
}
```

**【실행결과】****【word\_data.txt】**

mandatory | 1. 명령의 | 2. 통치를 위임받은 | 3. 강제의, 의무의(obligatory); 필수의  
preliminary | 1. 사전 준비 | 2. 예비 시험 | 3. 주 경기 이전에 펼쳐지는 개막 경기  
commitment | 1. 위탁, 위임; 위원회 회부 | 2. 인도; 투옥, 구류, 수감 | 3. 연질[공약]을 주기  
prominent | 1. 현저한, 두드러진 | 2. 둘기한, 양각된 | 3. 탁월한, 걸출한, 유명한; 중요한  
Tell me the reason for coming here. | 여기에 온 이유를 내게 말해라.  
There is something different about you today. | 너 오늘 평소와 좀 다르구나.  
He jumped up and down when he got the news. | 그는 뉴스를 듣고 펄쩍 뛰었다.  
When I opened it, I found a surprise. | 그 것을 열었을 때, 나는 놀라운 것을 발견했다.  
I have known him since he was a child. | 나는 그를 어려서부터 알고 있다.

**[15-9]** 다음은 메모장 프로그램의 일부인데, `fileOpen()`과 `saveAs()`가 아직 구현되어 있지 않다. 이 두 메서드를 구현하여 프로그램을 완성하시오.

**[연습문제]/ch15/Exercise15\_9.java**

```

import java.awt.*;
import java.awt.event.*;
import java.io.*;

class Exercise15_9 extends Frame {
    String fileName;
    TextArea content;
    MenuBar mb;
    Menu mFile;
    MenuItem miNew, miOpen, miSaveAs, miExit;

    Exercise15_9(String title) {
        super(title);
        content = new TextArea();
        add(content);

        mb      = new MenuBar();
        mFile   = new Menu("File");

        miNew   = new MenuItem("New");
        miOpen  = new MenuItem("Open");
        miSaveAs = new MenuItem("Save As...");
        miExit  = new MenuItem("Exit");

        mFile.add(miNew);
        mFile.add(miOpen);
        mFile.add(miSaveAs);
        mFile.addSeparator(); // 메뉴 분리선(separator)을 넣는다.
        mFile.add(miExit);

        mb.add(mFile);           // MenuBar에 Menu를 추가한다.
        setMenuBar(mb);         // Frame에 MenuBar를 포함시킨다.

        // 메뉴에 이벤트 핸들러를 등록한다.
        MyHandler handler = new MyHandler();
        miNew.addActionListener(handler);
        miOpen.addActionListener(handler);
        miSaveAs.addActionListener(handler);
        miExit.addActionListener(handler);

        setSize(300, 200);
        setVisible(true);
    }

    // 선택된 파일의 내용을 읽어서 TextArea에 보여주는 메서드
    void fileOpen(String fileName) {
        /*
        (1) 아래의 로직에 맞게 코드를 작성하시오.
        1. BufferedReader와 FileReader를 이용해서 지정된 파일을 읽는다.
        2. StringWriter에 출력한다.
        3. StringWriter의 내용을 content(TextArea)에 보여준다.
        */
    } // fileOpen메서드의 끝

    // TextArea의 내용을 지정된 파일에 저장하는 메서드
}

```

```

void saveAs(String fileName) {
    /*
        (2) 아래의 로직에 맞게 코드를 작성하시오.
        1. BufferedWriter와 FileWriter를 생성한다.
        2. content에 있는 내용을 가져와서 BufferedWriter에 출력한다.
        3. BufferedWriter를 닫는다.
    */
} // saveAs메서드의 끝

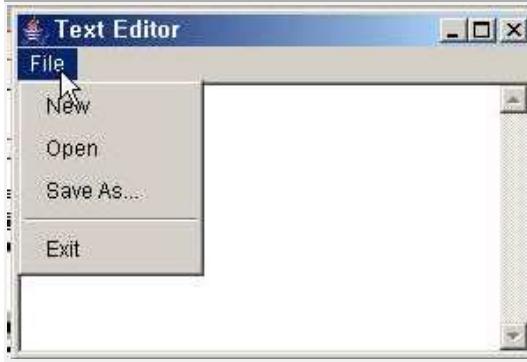
public static void main(String args[]) {
    Exercise15_9 mainWin = new Exercise15_9("Text Editor");
} // main메서드의 끝

// 메뉴를 클릭했을 때 메뉴별 처리코드
class MyHandler implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();

        if(command.equals("New")) {
            content.setText("");
        } else if(command.equals("Open")) {
            FileDialog fileOpen = new FileDialog(Exercise15_9.this,
                "파일열기");
            fileOpen.setVisible(true);
            fileName = fileOpen.getDirectory() + fileOpen.getFile();
            System.out.println(fileName);
            // 선택된 파일의 내용을 TextArea에 보여준다.
            fileOpen(fileName);
        } else if(command.equals("Save As...")) {
            FileDialog fileSave = new FileDialog(Exercise15_9.this,
                "파일저장",FileDialog.SAVE);
            fileSave.setVisible(true);
            fileName = fileSave.getDirectory() + fileSave.getFile();
            System.out.println(fileName);
            // 현재 TextArea의 내용을 선택된 파일에 저장한다.
            saveAs(fileName);
        } else if(command.equals("Exit")) {
            System.exit(0); // 프로그램을 종료시킨다.
        }
    }
} // class MyHandler
} // Exercise15_9클래스의 끝

```

## 【실행결과】



# Chapter 16

---

네트워킹

Networking

## [ 연습문제 ]

**[16-1]** ip주소가 192.168.10.100이고 서브넷 마스크(subnet mask)가 255.255.255.0일 때, 네트워크 주소와 호스트 주소를 계산하여 화면에 출력하는 프로그램을 작성하시오. 단, 비트연산자를 사용해서 계산해야 한다.

### 【실행결과】

```
네트워크 주소:192.168.10.0.  
호스트 주소:0.0.0.100.
```

**[16-2]** 다음 중 TCP의 특징이 아닌 것은?

- a. 전화와 같은 1:1 연결기반의 프로토콜이다.
- b. 데이터의 전송순서가 보장된다.
- c. UDP보다 전송속도가 빠르다.
- d. 데이터의 수신여부를 확인한다.

**[16-3]** JTextField에 URL을 입력하고 Enter키를 누르면 해당 페이지의 소스를 보여주는 'Source Viwer'프로그램이다. 예제16-4를 참고해서 (1)에 알맞은 코드를 넣어 완성하시오.

### 【연습문제】/ch16/Exercise16\_3.java

```
import java.net.*;  
import java.io.*;  
import java.awt.*;  
import java.awt.event.*;  
  
class SourceViewer extends Frame {  
    TextField tf = new TextField();  
    TextArea ta = new TextArea();  
  
    SourceViewer(String title) {  
        super(title);  
  
        add(tf, "North");  
        add(ta, "Center");  
  
        tf.addActionListener(new ActionListener(){  
            public void actionPerformed(ActionEvent ae) {  
                displaySource();  
            }  
        });  
    }  
    void displaySource() {  
        String url = tf.getText();  
        URL u = null;  
        try {  
            u = new URL(url);  
        } catch (MalformedURLException e) {  
            e.printStackTrace();  
        }  
        BufferedReader br = null;  
        try {  
            br = new BufferedReader(new InputStreamReader(u.openStream()));  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
        String str = null;  
        try {  
            str = br.readLine();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
        ta.setText(str);  
    }  
}
```

```

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });

        setBounds(500, 200, 500, 300);
        setVisible(true);
    }

    void displaySource() {
        URL url = null;
        BufferedReader input = null;
        String address = tf.getText().trim();
        String line = "";

        ta.setText("");

        try {
            if(!address.startsWith("http://"))
                address = "http://" + address;

            /*
             * (1) 알맞은 코드를 넣어 완성하시오.
             */
            input.close();
        } catch(Exception e) {
            ta.setText("유효하지 않은 URL입니다.");
        }
    } // displaySource()

    public static void main(String[] args) {
        SourceViewer mainWin = new SourceViewer("Source Viewer");
    }
}

```

**[실행결과]**

**[16-4]** 다음의 코드는 TCP통신을 하는 예제16-6, 16-7을 결합하여 GUI 채팅프로그램을 작성한 것이다. (1)~(4)에 알맞은 코드를 넣어 프로그램을 완성하시오.

**[연습문제]/ch16/ChatServer.java**

```
import java.net.*;
import java.io.*;
import java.awt.*;
import java.awt.event.*;

class ChatServer extends Frame {
    String nickname = "";

    DataOutputStream out;
    DataInputStream in;

    Panel p = new Panel();
    TextArea ta = new TextArea();
    TextField tf = new TextField();

    ChatServer(String nickname) {
        super("Chatting");
        this.nickname = nickname;

        p.setLayout(new BorderLayout());
        p.add(tf, "Center");

        add(ta, "Center");
        add(p, "South");

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        EventHandler handler = new EventHandler();
        ta.addFocusListener(handler);
        tf.addFocusListener(handler);
        tf.addActionListener(handler);

        ta.setEditable(false);
        setBounds(200, 200, 300, 200);
        setVisible(true);
        tf.requestFocus();
    }

    void startServer() {
        ServerSocket serverSocket = null;
        Socket socket = null;

        try {
            /*
             * (1) 아래의 로직에 맞게 코드를 작성하시오.
             * 1. 서비스켓을 생성하여 7777번 포트와 결합시킨다.
             * 2. ta에 "서버가 준비되었습니다."라고 보여준다.
             * 3. 상대방의 연결을 기다린다.
             * 4. ta에 "상대방과 연결되었습니다."라고 보여준다.
             *     ta.append("\r\n" + "상대방과 연결되었습니다.");
            */
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

5. 연결된 상대방 소켓의 입력스트림과 출력스트림을 얻어온다.
6. 반복문을 이용해서 입력스트림이 null이 아닌 동안
   입력스트림으로부터 데이터를 읽어서 변수 msg에 저장한다.

/*
ta.append("\r\n" + msg);
} catch (Exception e) {
e.printStackTrace();
}
}

public static void main(String[] args) {
if(args.length != 1) {
System.out.println("USAGE : java ChatServer NICKNAME");
System.exit(0);
}

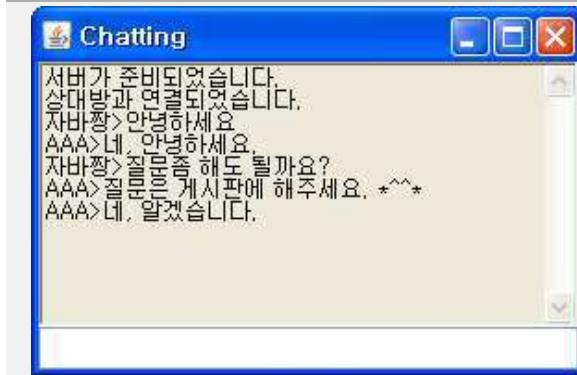
ChatServer chatWin = new ChatServer(args[0]);
chatWin.startServer();
} // main

class EventHandler extends FocusAdapter implements ActionListener {
public void actionPerformed(ActionEvent ae) {
String msg = tf.getText();

if("").equals(msg)) return;
/*
(2) 알맞은 코드를 넣어 완성하시오.
*/
ta.append("\r\n" + nickname +">" + msg);
tf.setText("");
}

public void focusGained(FocusEvent e) {
tf.requestFocus();
}
} // class EventHandler
} // class

```

**[실행결과]**

**[연습문제]/ch16/ChatClient.java**

```
import java.net.*;
import java.io.*;
import java.awt.*;
import java.awt.event.*;

class ChatClient extends Frame {
    String nickname = "";
    String serverIp = "";
    int serverPort = 0;

    DataOutputStream out;
    DataInputStream in;

    Panel p = new Panel();
    TextArea ta = new TextArea();
    TextField tf = new TextField();

    ChatClient(String nickname, String serverIp, String serverPort) {
        super("Chatting with " + serverIp + ":" + serverPort);
        this.nickname = nickname;
        this.serverIp = serverIp;
        this.serverPort = Integer.parseInt(serverPort);

        setBounds(600, 200, 300, 200);

        p.setLayout(new BorderLayout());
        p.add(tf, "Center");

        add(ta, "Center");
        add(p, "South");

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        EventHandler handler = new EventHandler();
        ta.addFocusListener(handler);
        tf.addFocusListener(handler);
        tf.addActionListener(handler);

        ta.setEditable(false);

        setVisible(true);
        tf.requestFocus();
    }

    void startClient() {
        try {
            /*
             * (3) 아래의 로직에 맞게 코드를 작성하시오.
             * 1. 소켓을 생성하여 serverIp의 serverPort에 연결한다.
             * 2. ta에 "상대방과 연결되었습니다."라고 보여준다.
             *      ta.setText("상대방과 연결되었습니다.");
             * 3. 연결된 상대방 소켓의 입력스트림과 출력스트림을 얻어온다.
             * 4. 반복문을 이용해서 입력스트림이 null이 아닌 동안
            */
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

입력스트림으로부터 데이터를 읽어서 변수 msg에 저장한다.

```

/*
ta.append("\r\n" + msg);
} catch(ConnectException ce) {
    ta.setText("상대방과 연결할 수 없습니다.");
    ce.printStackTrace();
} catch(IOException ie) {
    ie.printStackTrace();
} catch(Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    if(args.length != 3) {
        System.out.println("USAGE : java ChatClient NICKNAME SERVER_IP
SERVER_PORT");
        System.exit(0);
    }

    ChatClient chatWin = new ChatClient(args[0],args[1],args[2]);
    chatWin.startClient();
} // main

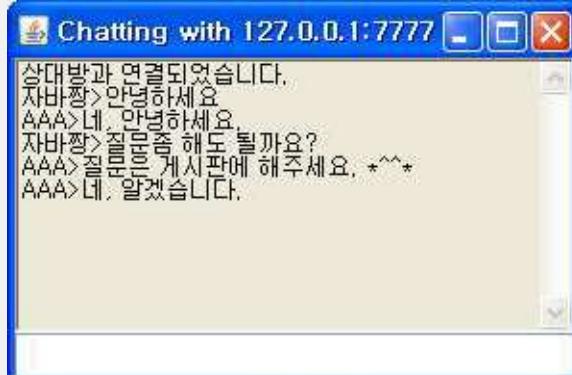
class EventHandler extends FocusAdapter implements ActionListener {
    public void actionPerformed(ActionEvent ae) {
        String msg = tf.getText();

        if("".equals(msg)) return;
        /*
         * (4) 알맞은 코드를 넣어 완성하시오.
        */
        ta.append("\r\n" + nickname +">" + msg);
        tf.setText("");
    }

    public void focusGained(FocusEvent e) {
        tf.requestFocus();
    }
} // class EventHandler
} // class

```

#### 【실행결과】



## [ 연습문제 - 모범답안 ]

[2-1] 다음 표의 빈 칸에 8개의 기본형(primitive type)을 알맞은 자리에 넣으시오.

크기 종류	1 byte	2 byte	4 byte	8 byte
논리형	boolean			
문자형		char		
정수형	byte	short	int	long
실수형			float	double