

GUI

1. 다음 프로그램 소스를 입력하고 실행하시오. (윈도우 클래스, JFrame)

A. 아래 코드를 실행하면 화면 좌측 상단에 윈도우가 하나 생긴다. 그러나 윈도우를 종료하더라도 Test 프로그램은 종료되지 않는다.

```
public class Test {  
    public static void main(String[] args) {  
        JFrame jFrame=new JFrame(); // 윈도우를 하나 만든다.  
        jFrame.setTitle("나의 첫 프로그램"); // 윈도우 상단 타이틀바(title bar)에 제목을 표시한다.  
        jFrame.setVisible(true); // 윈도우를 화면에 표시한다.  
    }  
}
```

2. 다음 프로그램 소스를 입력하고 실행하시오.

A. 아래 코드를 실행하면 윈도우를 종료할 때 프로그램이 종료된다.

```
public class Test {  
    public static void main(String[] args) {  
        JFrame jFrame=new JFrame(); // 윈도우를 하나 만든다.  
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // 윈도우 종료 시 프로그램도 종료되도록 한다.  
        jFrame.setVisible(true); // 윈도우를 화면에 표시한다.  
    }  
}
```

3. 다음 프로그램 소스를 입력하고 실행하시오. (레이블, JLabel)

A. 레이블을 만들어 윈도우에 추가했을 때 그 레이블이 화면에 가득 차는 것을 확인할 것

```
public class Test {  
    public static void main(String[] args) {  
        JFrame jFrame=new JFrame(); // 윈도우를 하나 만든다.  
        JLabel jLabel=new JLabel("경성대학교");  
        jFrame.add(jLabel); // 윈도우 위에 "경성대학교"라는 레이블을 올린다.  
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        jFrame.setVisible(true);  
    }  
}
```

4. 다음 프로그램 소스를 입력하고 실행하시오. (텍스트필드, JTextField)

A. 텍스트필드를 만들어 윈도우에 추가했을 때 그 텍스트필드가 화면에 가득 차는 것을 확인할 것

```
public class Test {  
    public static void main(String[] args) {  
        JFrame jFrame=new JFrame(); // 윈도우를 하나 만든다.  
        JTextField jTextField=new JTextField(); // 한 줄만 입력 가능한 텍스트필드를 만든다.  
        jFrame.add(jTextField); // 텍스트필드를 윈도우 위에 올린다.  
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        jFrame.setVisible(true);  
    }  
}
```

5. 다음 프로그램 소스를 입력하고 실행하시오. (텍스트영역, JTextArea)

- A. 여러 줄 입력이 가능한 텍스트영역(JTextArea) 객체를 윈도우에 올리는 예제이다. 그러나 여러 줄 계속 입력할 경우에도 스크롤 바가 자동으로 생기지 않는 것을 확인할 것.
- B. pack() 메소드는 윈도우에 추가된 객체의 크기에 맞추어 윈도우 크기를 조정해 준다.

```
public class Test {  
    public static void main(String[] args) {  
        JFrame jFrame=new JFrame();  
        JTextArea jTextArea=new JTextArea();  
        jFrame.add(jTextArea);  
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        jFrame.pack();  
        jFrame.setVisible(true);  
    }  
}
```

6. 다음 프로그램 소스를 입력하고 실행하시오. (스크롤 기능 지원 창, JScrollPane)

- A. 스크롤바 기능을 지원하는 창 JScrollPane의 개체를 만들고 그 위에 텍스트영역(JTextArea)을 올린다.

```
public class Test {  
    public static void main(String[] args) {  
        JFrame jFrame=new JFrame();  
        JTextArea jTextArea=new JTextArea();  
        JScrollPane jScrollPane=new JScrollPane(jTextArea);  
        jFrame.add(jScrollPane);  
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        jFrame.pack();  
        jFrame.setVisible(true);  
    }  
}
```

7. 다음 프로그램 소스를 입력하고 실행하시오. (버튼, JButton)

```
public class Test {  
    public static void main(String[] args) {  
        JFrame jFrame=new JFrame(); // 윈도우를 하나 만든다.  
        JButton jButton=new JButton("OK"); // OK라고 표시된 버튼을 하나 만든다.  
        jFrame.add(jButton); // 버튼을 윈도우 위에 올린다.  
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        jFrame.setVisible(true);  
    }  
}
```

8. 다음 프로그램 소스를 입력하고 실행하시오.

A. 두 개의 레이블을 윈도우에 추가했지만 마지막 레이블만 화면에 표시된다는 것을 확인할 것.

B. 여러 개의 화면 요소들(레이블, 텍스트필드, 버튼 등)을 윈도우 위에 올리는 방법은?

```
public class Test {  
    public static void main(String[] args) {  
        JFrame jFrame=new JFrame();  
        JLabel jLabel1=new JLabel("동해물과 백두산이");  
        jFrame.add(jLabel1);  
        JLabel jLabel2=new JLabel("마르고 닳도록");  
        jFrame.add(jLabel2);  
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        jFrame.setVisible(true);  
    }  
}
```

9. 다음 프로그램 소스를 입력하고 실행하시오. (화면 배치 설정, setLayout(), pack())

- A. 버튼, 레이블, 텍스트필드 등 화면 요소들을 윈도우에 올릴 때에는 배치(Layout) 방식을 정한 다음 올려야 한다.
- B. 아래 프로그램은 배치 방식을 setLayout() 메소드를 통해 FlowLayout으로 설정했다. 이는 화면 요소들이 윈도우 위에서 윈도우 크기에 맞추어 떠다니는 방식으로 배치되는 식이다. 또한 코드 마지막 부분의 pack() 메소드는 화면 요소들 전체를 포함하도록 윈도우 크기를 조정해 준다. pack() 문장을 제거하고 실행해 볼 것.
- C. 배치는 그런대로 되었지만 아무리 버튼을 눌러도 반응이 없다. 버튼을 클릭했을 때 어떤 작업을 수행할 것인지에 대해 프로그램하지 않았기 때문이다.

```
public class Test {  
    public static void main(String[] args) {  
        JFrame jFrame=new JFrame();  
        jFrame.setLayout(new FlowLayout());  
        JLabel jLabel1=new JLabel("First number");  
        jFrame.add(jLabel1);  
        JTextField jTextField1=new JTextField(10);  
        jFrame.add(jTextField1);  
        JLabel jLabel2=new JLabel("Second number");  
        jFrame.add(jLabel2);  
        JTextField jTextField2=new JTextField(10);  
        jFrame.add(jTextField2);  
        JButton jButton=new JButton("Add");  
        jFrame.add(jButton);  
        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        jFrame.pack(); // 배치한 화면 요소들의 크기에 맞게 윈도우 크기를 조정함.  
        jFrame.setVisible(true);  
    }  
}
```

10. 다음 프로그램 소스를 입력하고 실행하시오.

```
public class Test {
    public static void main(String[] args) {
        JFrame jFrame=new JFrame();
        jFrame.setLayout(new FlowLayout());

        JLabel jLabel1=new JLabel("First number");
        JLabel jLabel2=new JLabel("Second number");
        JTextField jTextField1=new JTextField(10);
        JTextField jTextField2=new JTextField(10);
        JButton jButton=new JButton("Add");

        jFrame.add(jLabel1);
        jFrame.add(jTextField1);
        jFrame.add(jLabel2);
        jFrame.add(jTextField2);
        jFrame.add(jButton);

        jButton.addActionListener(
            new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) { // jButton이 클릭되면 이 함수가 호출됨. 버튼 클릭시 수행될 작업이 기술됨.
                    int n1=Integer.parseInt(jTextField1.getText()); // jTextField1.getText()는 jTextField1에 입력된 텍스트를 문자열 반환
                    int n2=Integer.parseInt(jTextField2.getText());
                    JOptionPane.showMessageDialog(null, n1+n2); // 팝업창을 띄워 두 수의 합을 표시한다.
                }
            }
        );

        jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jFrame.pack();
        jFrame.setVisible(true);
    }
}
```

11. 다음 프로그램 소스를 입력하고 실행하시오. 다음 문제의 코드와 비교할 것.

```
public class Test {  
    public static void main(String args[]){  
        JFrame w=new JFrame();  
        w.setVisible(true);  
    }  
}
```

12. 다음 프로그램 소스를 입력하고 실행하시오. 이전 문제의 코드와 비교할 것.

```
public class Test extends JFrame {  
    public static void main(String args[]){  
        Test test=new Test();  
        test.setVisible(true);  
    }  
}
```

13. 다음 프로그램 소스를 입력하고 실행하시오. 다음 문제의 코드와 비교할 것.

```
public class Test {  
    public static void main(String args[]){  
        JFrame w=new JFrame();  
        JButton jButton=new JButton("Click Me ");  
        w.add(jButton);  
        w.setVisible(true);  
    }  
}
```

14. 다음 프로그램 소스를 입력하고 실행하시오. 이전 문제의 코드와 비교할 것.

```
public class Test extends JFrame {  
    JButton jButton=new JButton("Click Me");  
    public Test() {  
        add(jButton);  
    }  
    public static void main(String args[]){  
        Test test=new Test();  
        test.setVisible(true);  
    }  
}
```


15. 다음 프로그램 소스를 입력하고 실행하시오. 다음 문제의 코드와 비교할 것.

```
public class Test {  
    public static void main(String args[]){  
        JFrame w=new JFrame();  
        JButton jButton=new JButton("Click Me ");  
        jButton.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                JOptionPane.showMessageDialog(null, "Hi");  
            }  
        });  
        w.add(jButton);  
        w.setVisible(true);  
    }  
}
```

16. 다음 프로그램 소스를 입력하고 실행하시오. 이전 문제의 코드와 비교할 것.

```
public class Test extends JFrame {  
    JButton jButton=new JButton("Click Me");  
    public Test() {  
        jButton.addActionListener(new ActionListener() {  
            @Override  
            public void actionPerformed(ActionEvent e) {  
                JOptionPane.showMessageDialog(null, "Hi");  
            }  
        });  
        add(jButton);  
    }  
    public static void main(String args[]){  
        Test test=new Test();  
        test.setVisible(true);  
    }  
}
```

17. 다음 프로그램 소스를 입력하고 실행하시오. 이전 문제의 코드와 비교할 것.

```
public class Test extends JFrame {
    JButton jButton=new JButton("Click Me");
    public Test() {
        ActionListener actionListener=new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(null, "Hi");
            }
        };
        jButton.addActionListener(actionListener);
        add(jButton);
    }
    public static void main(String args[]){
        Test test=new Test();
        test.setVisible(true);
    }
}
```

18. 다음 프로그램 소스를 입력하고 실행하시오. 이전 문제의 코드와 비교할 것.

```
public class Test extends JFrame implements ActionListener {
    JButton jButton=new JButton("Click Me");
    public Test() {
        jButton.addActionListener(this);
        add(jButton);
    }
    public static void main(String args[]){
        Test test=new Test();
        test.setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Hi");
    }
}
```

19. 다음 프로그램 소스를 입력하고 실행하시오.

A. 다음은 앞 프로그램과 같은 GUI 덧셈기를 JFrame 클래스를 상속받는 자식 클래스 Window를 통해 구현한 코드임.

B. Window 클래스가 JFrame이므로 new Window()를 하면 new JFrame()한 것처럼 윈도우 객체가 만들어짐.

```
public class Window extends JFrame {
    JTextField jTextField1=new JTextField(20);
    JTextField jTextField2=new JTextField(20);
    JButton jButton=new JButton("ADD");
    public Window() {
        setTitle("덧셈기"); // this.setTitle("덧셈기"); 와 동일
        setLayout(new FlowLayout());
        add(jTextField1);
        add(new JLabel("+"));
        add(jTextField2);
        add(jButton);
        jButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                double n1=Double.parseDouble(jTextField1.getText());
                double n2=Double.parseDouble(jTextField2.getText());
                JOptionPane.showMessageDialog(null, n1+n2);
            }
        });
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
    }
}

public class Test {
    public static void main(String[] args) {
        new Window().setVisible(true);
    }
}
```

20. 다음 프로그램 소스를 입력하고 실행하시오.

- A. Window 클래스를 ActionListener의 자식이 되도록 했으므로, Window 클래스가 곧 ActionListener이다. 따라서 아래의
jButton.addActionListener(this); 문장에서 this 즉 Window 클래스의 객체를 파라미터로 넘기고 있다.

```
public class Window extends JFrame implements ActionListener{
    JTextField jTextField1=new JTextField(20);
    JTextField jTextField2=new JTextField(20);
    JButton jButton=new JButton("ADD");
    public Window() {
        this.setTitle("덧셈기");
        setLayout(new FlowLayout());
        add(jTextField1);
        add(new JLabel("+"));
        add(jTextField2);
        add(jButton);
        jButton.addActionListener(this);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        double n1=Double.parseDouble(jTextField1.getText());
        double n2=Double.parseDouble(jTextField2.getText());
        JOptionPane.showMessageDialog(null, n1+n2);
    }
}

public class Test {
    public static void main(String[] args) {
        new Window().setVisible(true);
    }
}
```

21. 다음 프로그램 소스를 입력하고 실행하시오. 아래 코드는 앞 코드와 같은 동작을 수행하는 GUI 덧셈기이다. 그러나 앞과 달리 윈도우 객체를 만들고 화면에 표시하는 작업을 `SwingUtilities.invokeLater()` 메소드를 통해서 수행하고 있다.

```
public class Window extends JFrame implements ActionListener{
    JTextField jTextField1=new JTextField(20);
    JTextField jTextField2=new JTextField(20);
    JButton jButton=new JButton("ADD");
    public Window() {
        this.setTitle("덧셈기");
        setLayout(new FlowLayout());
        add(jTextField1);
        add(new JLabel("+"));
        add(jTextField2);
        add(jButton);
        jButton.addActionListener(this);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        double n1=Double.parseDouble(jTextField1.getText());
        double n2=Double.parseDouble(jTextField2.getText());
        JOptionPane.showMessageDialog(null, n1+n2);
    }
}

public class Test {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new Window().setVisible(true);
            }
        });
    }
}
```

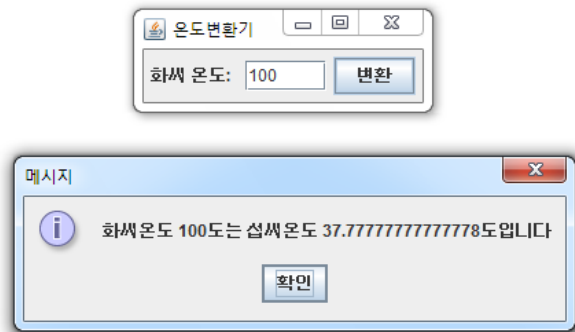
22. 다음 프로그램 소스를 입력하고 실행하시오. (작은 크기의 텍스트파일 내용 전체를 JTextArea에 표시하기)

```
public class Test {  
    public static void main(String[] args) {  
        try {  
            JFrame jFrame=new JFrame();  
            JTextArea jTextArea=new JTextArea();  
            JScrollPane jScrollPane=new JScrollPane(jTextArea);  
            jFrame.add(jScrollPane);  
            File iFile=new File("./src/Test.java");  
            char cbuf[]=new char[(int) iFile.length()];  
            new FileReader(iFile).read(cbuf);  
            jTextArea.setText(new String(cbuf));  
            jFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
            jFrame.pack();  
            jFrame.setVisible(true);  
        } catch (Exception e) {  
        }  
    }  
}
```

23. 실습 과제 I

A. 다음과 같이 동작하는 온도 변환기 TempConv.java를 만드시오.

- i. 화씨(F)를 섭씨(C)로 변환하는 공식: $C = (F - 32.0) / 1.8$



24. 실습 과제 II

- A. 다음과 같이 동작하는 BMI.java를 만드시오. BMI 지수값(예: 24.49648)을 소수점 둘째 자리까지만 출력할 것. 이를 위해 String.format() 함수를 사용할 것 (아래 예제 참고).

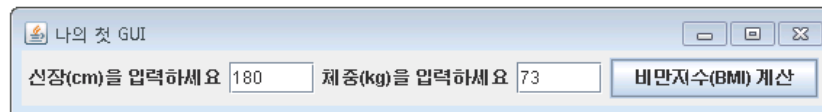
```
double n=3.135495;
```

```
String s=String.format("%.2f", n); // s에는 문자열 "3.14"가 저장됨.
```

$$\text{BMI} = \frac{\text{mass}(\text{kg})}{(\text{height}(\text{m}))^2}$$

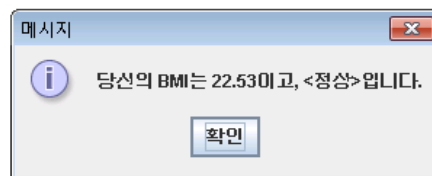
BMI 20 미만→저체중, 20~24→정상, 25~29→과체중, 30~39→비만, 40이상→고도비만

예) 신장: 173 cm, 체중 60 → BMI=60/(1.73×1.73) = 20.05



나의 첫 GUI

신장(cm)을 입력하세요 180 체중(kg)을 입력하세요 73 비만지수(BMI) 계산



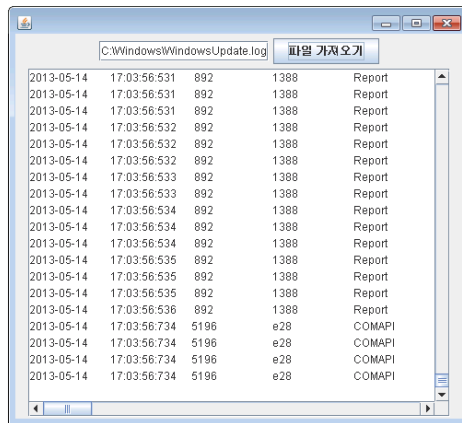
메시지

당신의 BMI는 22.53이고, <정상>입니다.

확인

25. 실습 과제 III

- i. 다음과 같은 GUI를 갖는 텍스트 파일 표시기를 만드시오. 즉, 텍스트 파일의 절대경로를 입력 받아 해당 텍스트 파일의 내용을 JTextArea에 표시하도록 동작해야 함.



References

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

남궁성. 자바의 정석. 도우출판.

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

황기태, 김효수 (2015). 명품 Java Programming. 생능출판사.