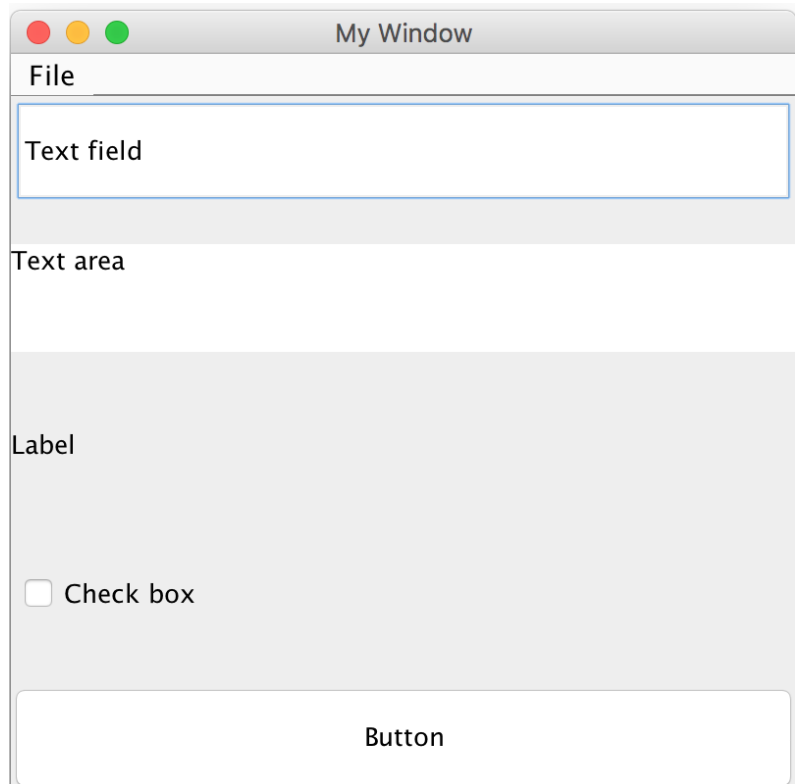
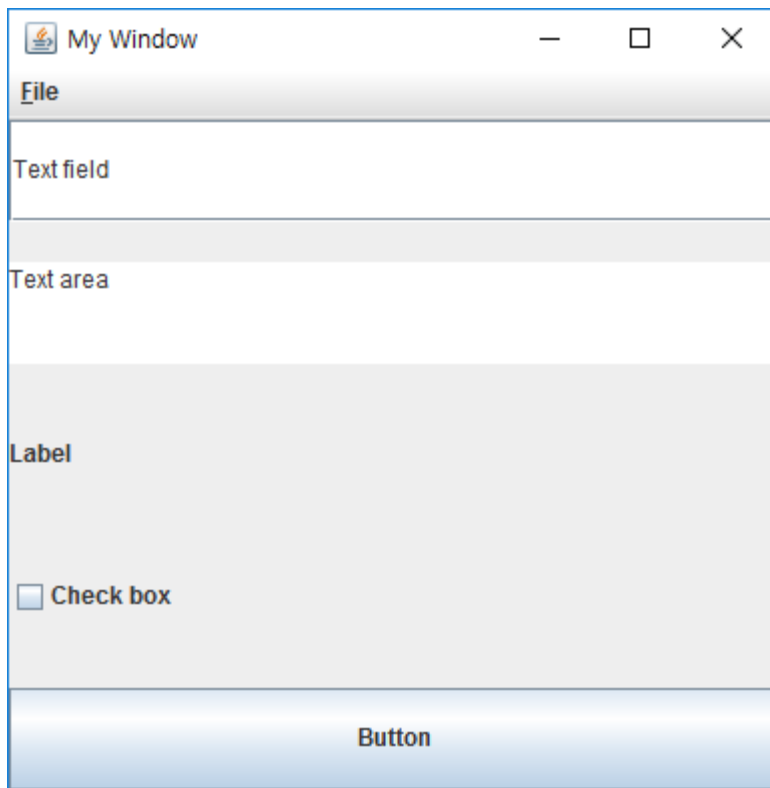


컴퓨터 프로그래밍

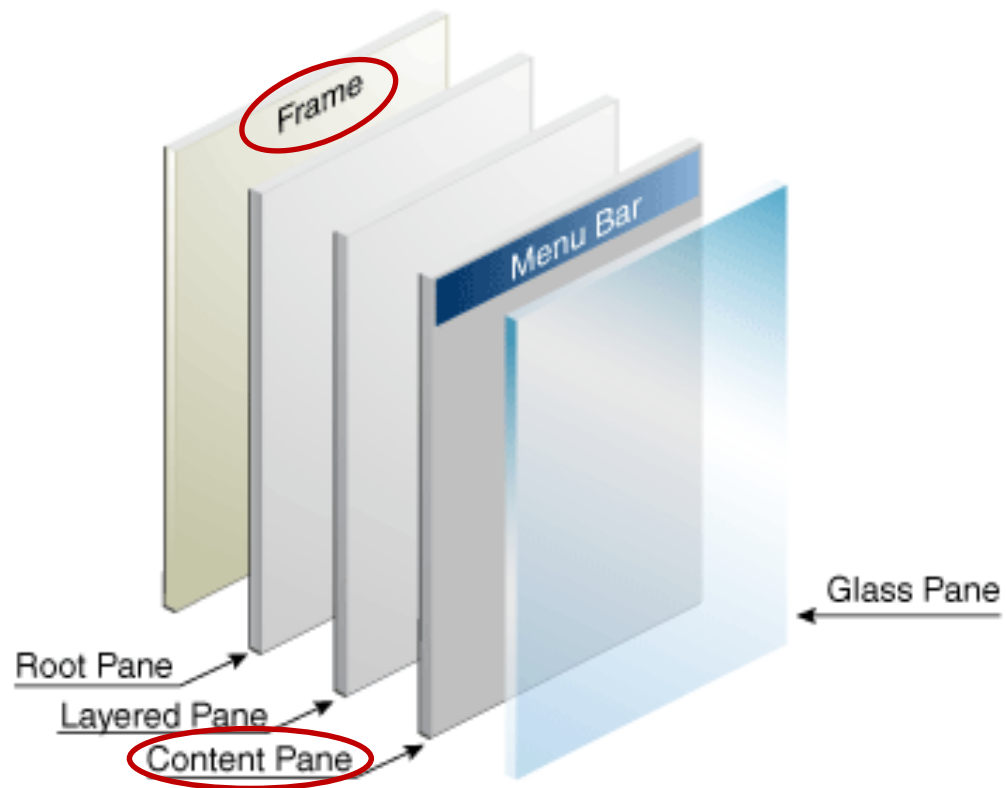
GUI

# class for GUI

- AWT (Abstract Window Toolkit)
  - Standard API for providing a GUI for a Java program
- Swing
  - Provides a more sophisticated set of GUI components

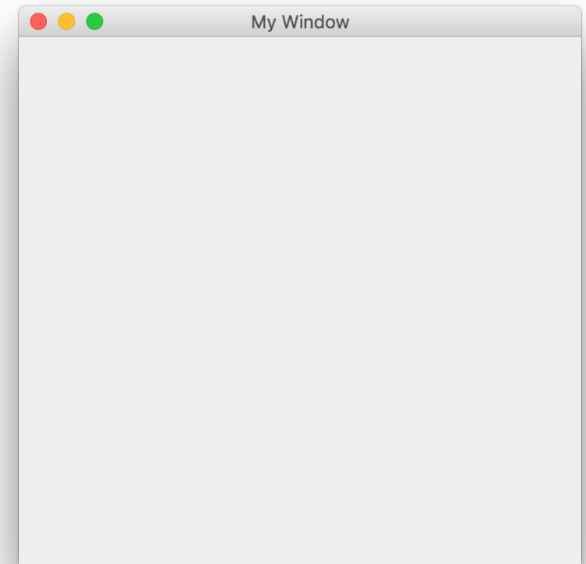


# Structure



# Frame

```
class MainFrame {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setTitle("My Window");  
        frame.setSize(400, 400);  
        frame.setVisible(true);  
    }  
}
```



# content Pane

```
class MainFrame {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        container contentPane = frame.getContentPane();  
        frame.setTitle("My window");  
        frame.setSize(400, 400);  
        frame.setVisible(true);  
    }  
}
```

# Layout Managers

- Each container can have a layout manager
- determines the size and position of the components within a container
- FlowLayout
- BorderLayout
- GridLayout
- ...
- <https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

# Layout Managers

```
class MainFrame {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        container contentPane = frame.getContentPane();  
        contentPane.setLayout(new GridLayout(5, 1, 0, 20));  
        frame.setTitle("My window");  
        frame.setSize(400, 400);  
        frame.setVisible(true);  
    }  
}
```

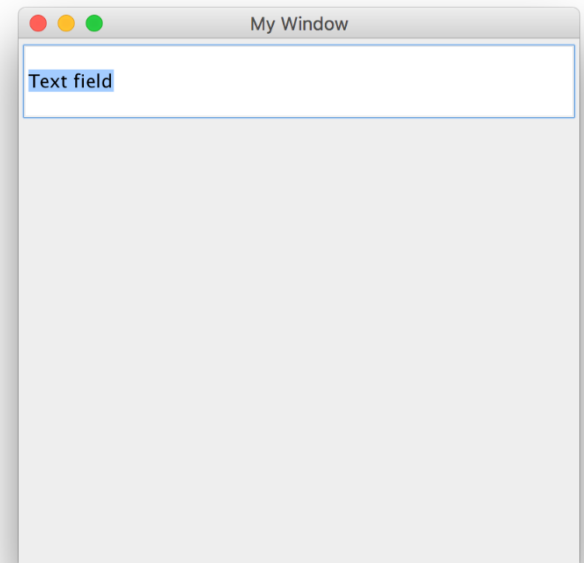
# components

- JLabel
  - displays unselectable text and images
- JTextField
  - enables the user to type a single line text
- JTextArea
  - displays multiple lines of text and optionally allows the user to edit the text
- JCheckBox
  - provides a check box button
- JButton
  - provides a common button
- JPanel
  - generic container
- JSlider, JSpinner, JComboBox, JProgressBar, ...



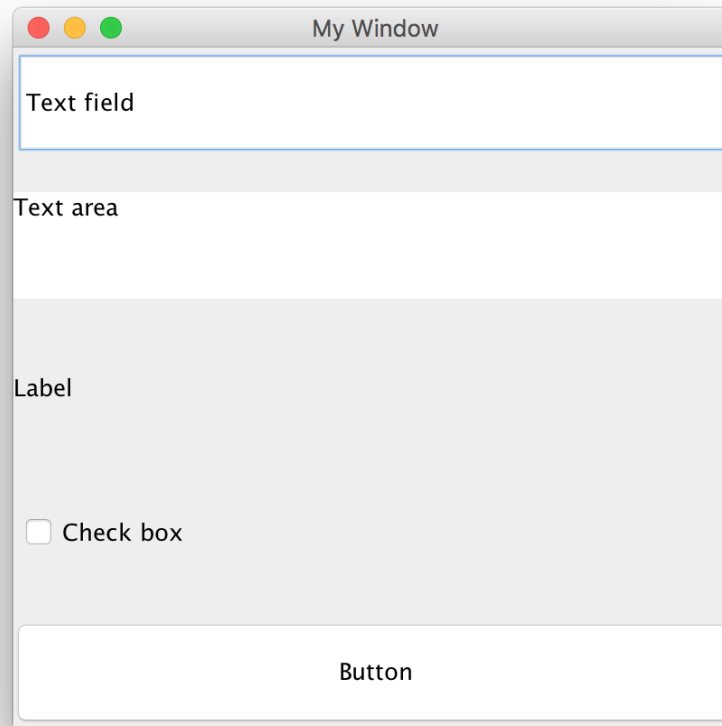
# components

```
class MainFrame {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        container contentPane = frame.getContentPane();  
        contentPane.setLayout(new GridLayout(5, 1, 0, 20));  
        JTextField textField = new JTextField("Text field");  
        contentPane.add(textField);  
        frame.setTitle("My window");  
        frame.setSize(400, 400);  
        frame.setVisible(true);  
    }  
}
```



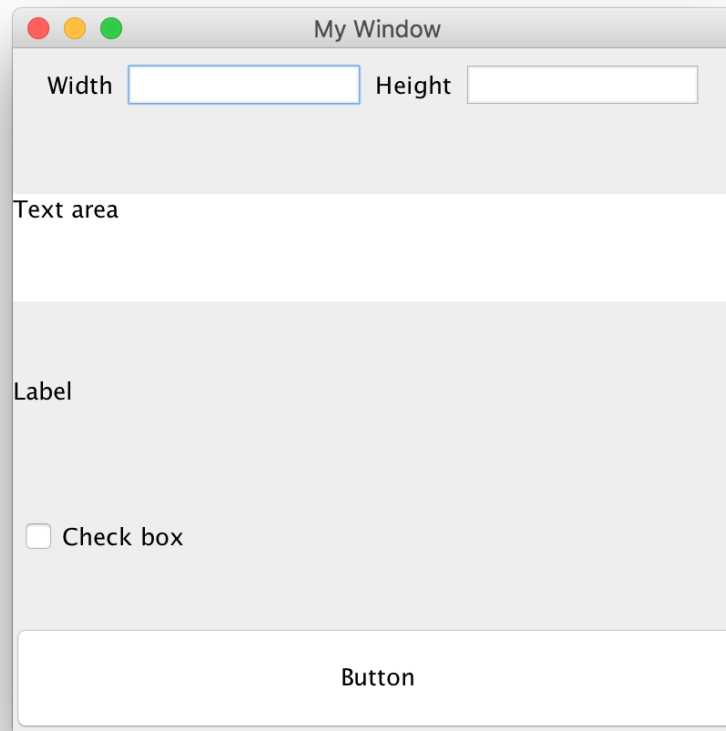
시작

- component들을 추가하여 아래 화면과 같이 UI를 구성하시오



시작

- component들을 추가하여 아래 화면과 같이 UI를 구성하시오  
— JPanel



# Event-driven Programming

- Event

- 사용자가 UI component에 취하는 행위

- 이벤트의 발생에 의해 프로그램의 흐름이 결정되는 방식

- 이벤트를 받아 처리하고자 하는 컴포넌트에 event listener 등록

- 이벤트가 발생하면 event listener 실행

# Event Listener

- JDK에 있는 interface 정리

```
public interface KeyListener extends EventListener {  
    public void keyTyped(KeyEvent e);  
    public void keyPressed(KeyEvent e);  
    public void keyReleased(KeyEvent e);  
}
```

```
public interface MouseListener extends EventListener {  
    public void mouseClicked(MouseEvent e);  
    public void mousePressed(MouseEvent e);  
    public void mouseReleased(MouseEvent e);  
    public void mouseEntered(MouseEvent e);  
    public void mouseExited(MouseEvent e);  
}
```

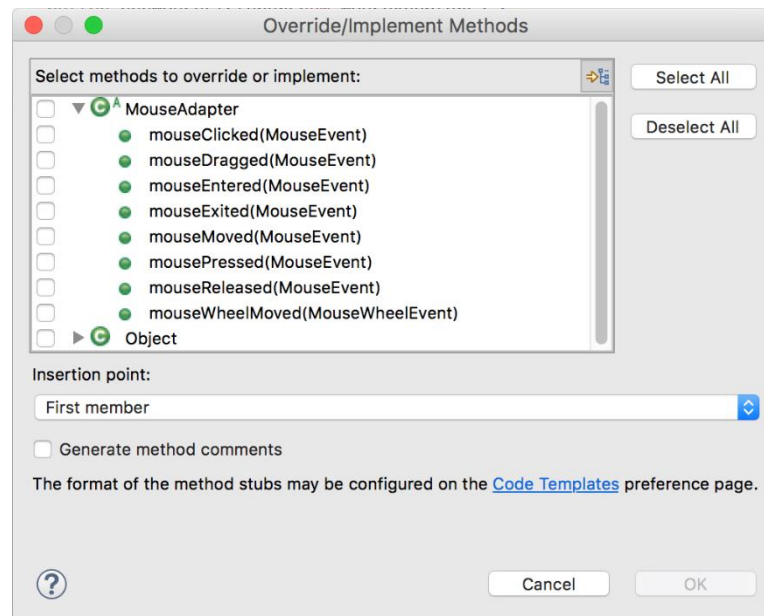
# Add MouseListener to Button

```
button.addMouseListener(new MouseListener() {  
    @Override  
    public void mouseClicked(MouseEvent e) {  
    }  
    @Override  
    public void mousePressed(MouseEvent e) {  
    }  
    @Override  
    public void mouseReleased(MouseEvent e) {  
    }  
    @Override  
    public void mouseEntered(MouseEvent e) {  
    }  
    @Override  
    public void mouseExited(MouseEvent e) {  
    }  
});
```

# Add MouseListener to Button

```
button.addMouseListener(new MouseAdapter() {  
  
});
```

- Set cursor into MouseAdapter body
- Source -> Override/Implement Methods...



# Add MouseListener to Button

- MouseListener vs. MouseAdapter

```
public interface MouseListener extends EventListener {  
    public void mouseClicked(MouseEvent e);  
    public void mousePressed(MouseEvent e);  
    public void mouseReleased(MouseEvent e);  
    public void mouseEntered(MouseEvent e);  
    public void mouseExited(MouseEvent e);  
};
```

should implement  
all methods

```
public abstract class MouseAdapter implements MouseListener, MouseWheelListener,  
                                                MouseMotionListener {  
    public void mouseClicked(MouseEvent e) {}  
    public void mousePressed(MouseEvent e) {}  
    public void mouseReleased(MouseEvent e) {}  
    public void mouseEntered(MouseEvent e) {}  
    public void mouseExited(MouseEvent e) {}  
    public void mouseWheelMoved(MouseWheelEvent e) {}  
    public void mouseDragged(MouseEvent e) {}  
    public void mouseMoved(MouseEvent e) {}  
};
```

can implement  
what we need



# Add ActionListener to Button

```
@Override
```

```
public void mouseClicked(MouseEvent e) {  
    ((JButton)e.getSource()).setText("clicked");  
}
```

```
@Override
```

```
public void mouseEntered(MouseEvent e) {  
    ((JButton)e.getSource()).setForeground(color.RED);  
}
```

```
@Override
```

```
public void mouseExited(MouseEvent e) {  
    ((JButton)e.getSource()).setForeground(color.BLACK);  
}
```

사각형

- 사각형의 가로, 세로 값을 입력받고(JTextField) 버튼을 누르면 사각형의 넓이를 구하여 표시(JLabel)해주는 UI

My Window

Width 10 Height 20

Calculate Area

200.0

시스  
템

- 문자열을 입력받고(JTextArea) 영어 소문자가 입력되면 대문자로 바꿔 표시해주는 UI

