1.00
Tutorial 7

Swing I

Agenda

- Swing Components
- Event Handling

  (download Tutorial7.java and Tutorial7Inner.java)

  - By implementing ActionListner Interface (Example 3)
  - Using Anonymous Inner Class (Example 4)
- Problem Set 6

Review - Swing Toolkit

- **Top Level Windows (`JFrame`, `JDialog`)**

  - Not contained in other containers

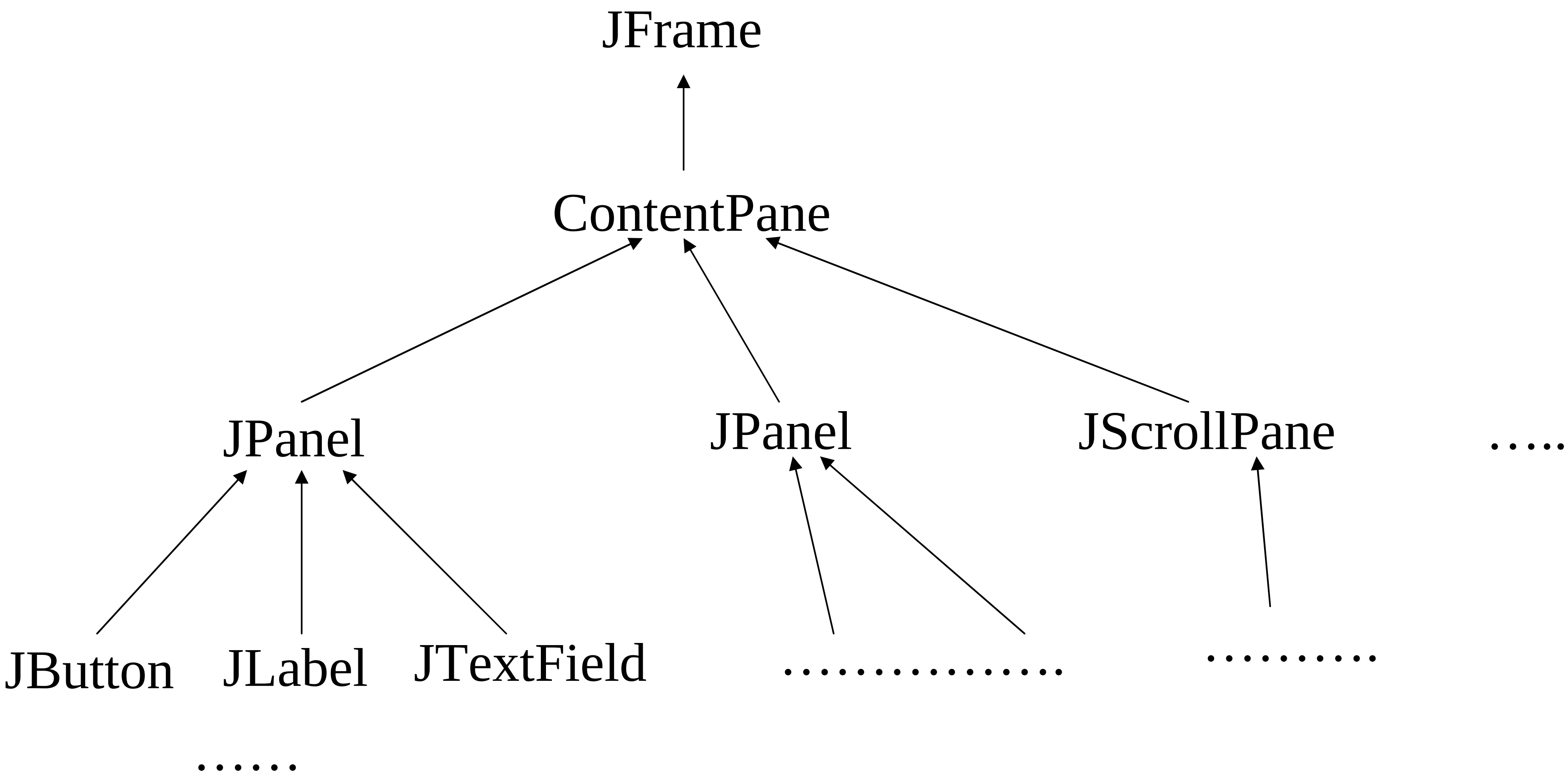
  - Interact with native windowing system
- **Containers (`JPanel`, `JScrollPane`)**

  - Hold components
- **JComponents (`JButton`, `JLabel`)**

  - Present GUI elements

  - Interact with user

Composition of a Swing GUI



JFrame

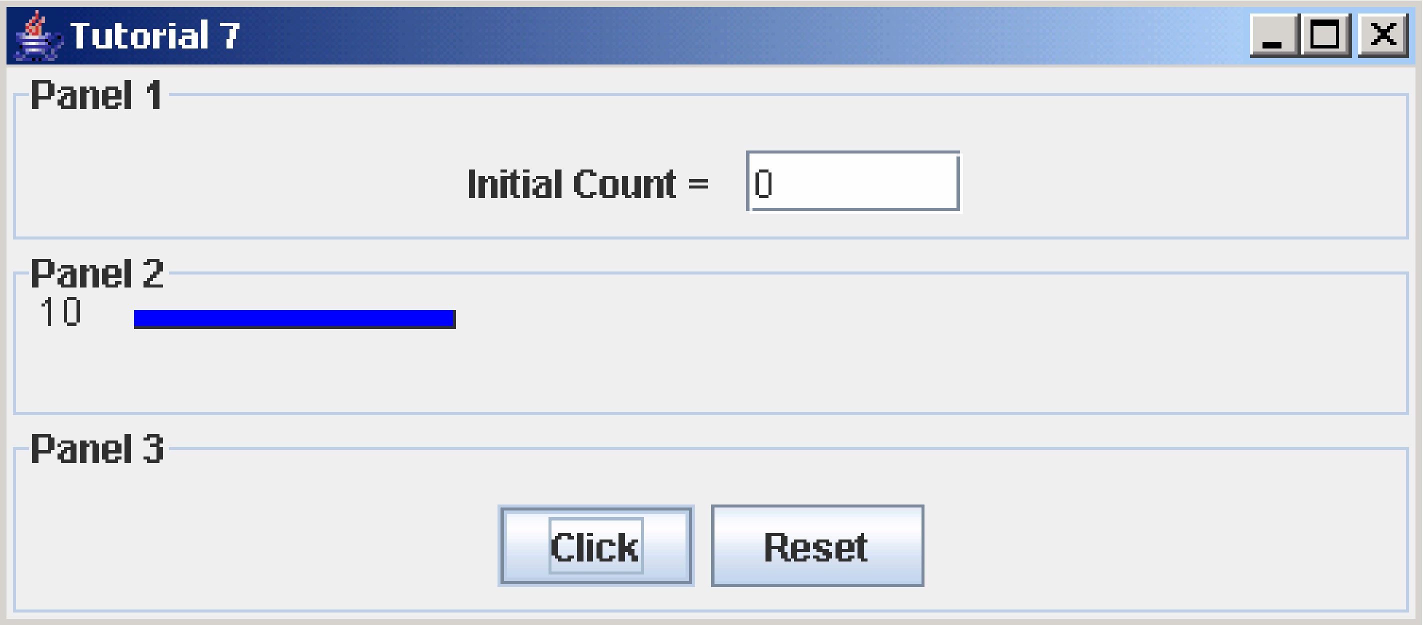
ContentPane

JPanel JPanel JScrollPane …..

JButton JLabel JTextField ……………. ……….

……

Example



- Identify the top level window
- Identify the containers
- Identify the components

Answers

- **Top Level Window**

  - `JFrame`

- **Containers**

  - `JPanels:` Panel 1, Panel 2 (2D Graph), Panel 3

- **Components**

  - `JLabel`, `JTextField`

  - `JButton`

Review - Swing Event Model

- **Event Sources**

  - Components generate event upon user/system interaction

- **Event Objects**

  - Events are encapsulated by instances of EventObject's subclasses (ActionEvent, MouseEvent)

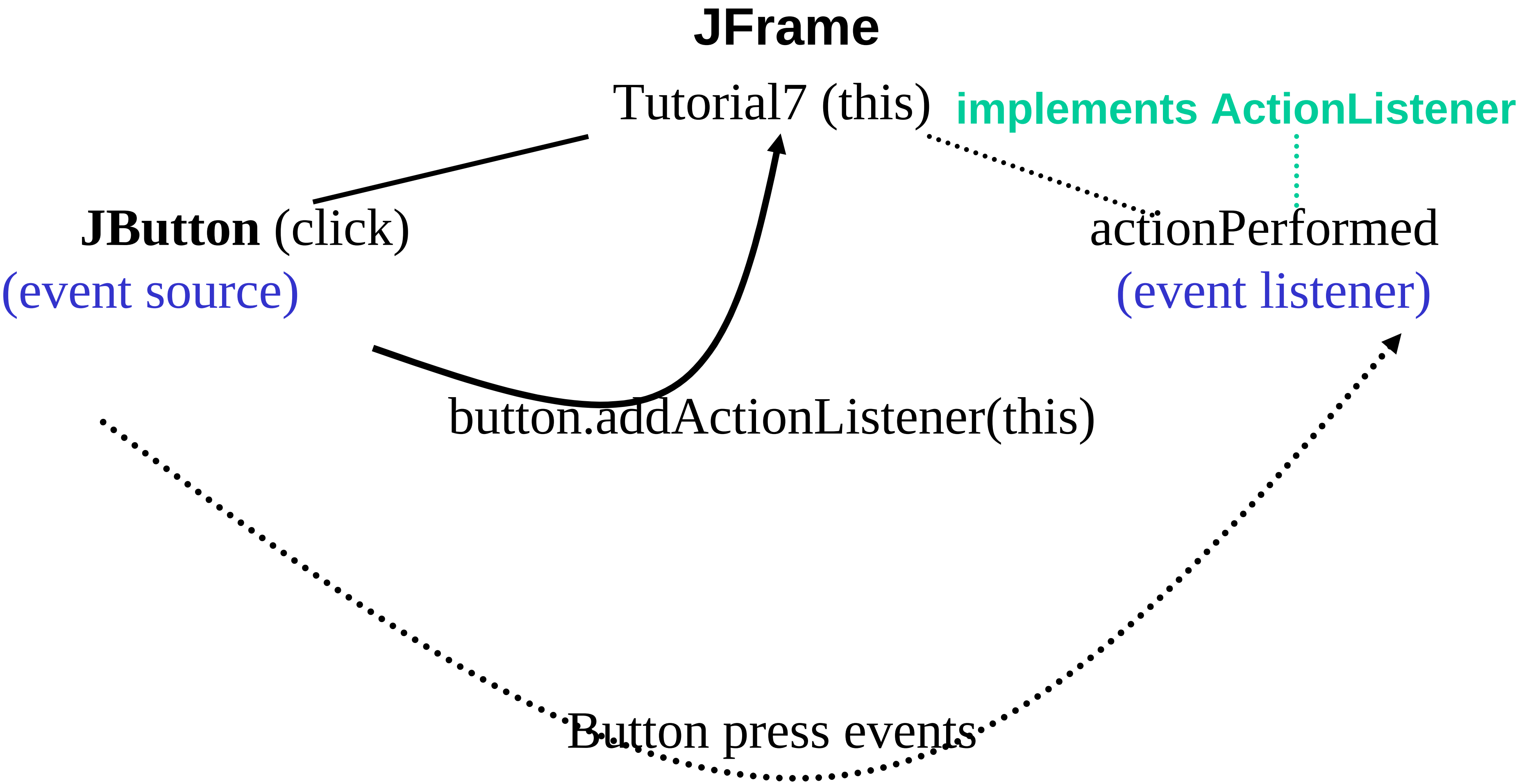
- **Event Listeners**

  - Listen and react to event objects

  - Event sources are delegated to event listeners

  - A source may have many listeners, and a listener may listen to many sources

Button Example Framework



JFrame

Tutorial7 (this) implements ActionListener

JButton (click)
(event source)

actionPerformed
(event listener)

button.addActionListener(this)

Button press events

Event Handling

- by implementing ActionListner Interface

- Make Tutorial7 an ActionListener.
- Register click and reset events with Tutorial7.
- Implement actionPerformed() method
  - If event source is the click button, increment the counter and extend the blue bar in panel 2
  - If event source is the reset button, read the initial counter from the text field and reset the counter and the blue bar.
  - You can reuse the following method:
    - incrementNumber()
    - resetNumber()
  - Notice the exception handling in method resetNumber()

Solution

```
public class Tutorial7 extends JFrame implements ActionListener {
```

Step 2: ActionListener Interface

```
// {


        click.setActionCommand(CLICK);
        click.addActionListener(this);
```

Step 1: Who will handle the event

```
//}
```

Step 3: Event handling using `actionPerformed()` method

```
public void actionPerformed(ActionEvent e) {
   String action = e.getActionCommand();
   //Event 1 : clicked
   if (action.equals(CLICK)) {
        numberOfClicks++;
        area.setText(new Integer(numberOfClicks).toString());
   }
   // Else Handle other action
}


} //end of Class
```

Event Handling

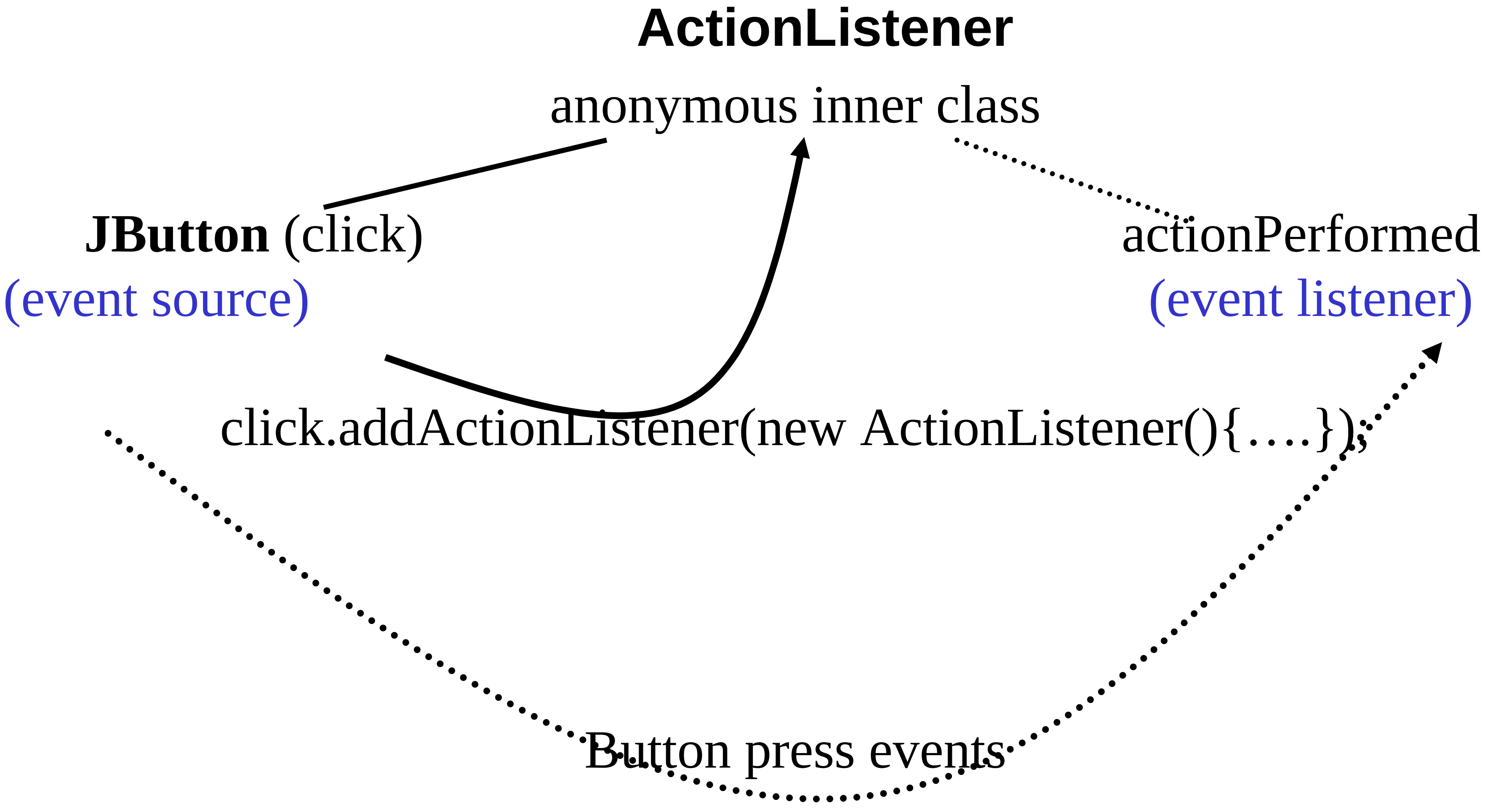
- using Anonymous Inner Classes

- Now write Event Listeners using Anonymous Inner Classes

```
public class Tutorial7Inner extends JFrame {

// {
                 …

     click.addActionListener(new ActionListener() {

         //COMPLETE THE CODE


});

//} End of Code
```

Event Handling

- using Anonymous Inner Classes



ActionListener

anonymous inner class

JButton (click)

(event source)

actionPerformed

(event listener)

click.addActionListener(new ActionListener(){….});

Button press events

Example 4: Solution

(Anonymous Inner Classes)

```
public class Tutorial7Inner extends JFrame {

// {
        ……

     click.addActionListener(new ActionListener() {
          public void actionPerformed(ActionEvent e) {
              incrementNumber();
          }
     });

//} End of Code
```

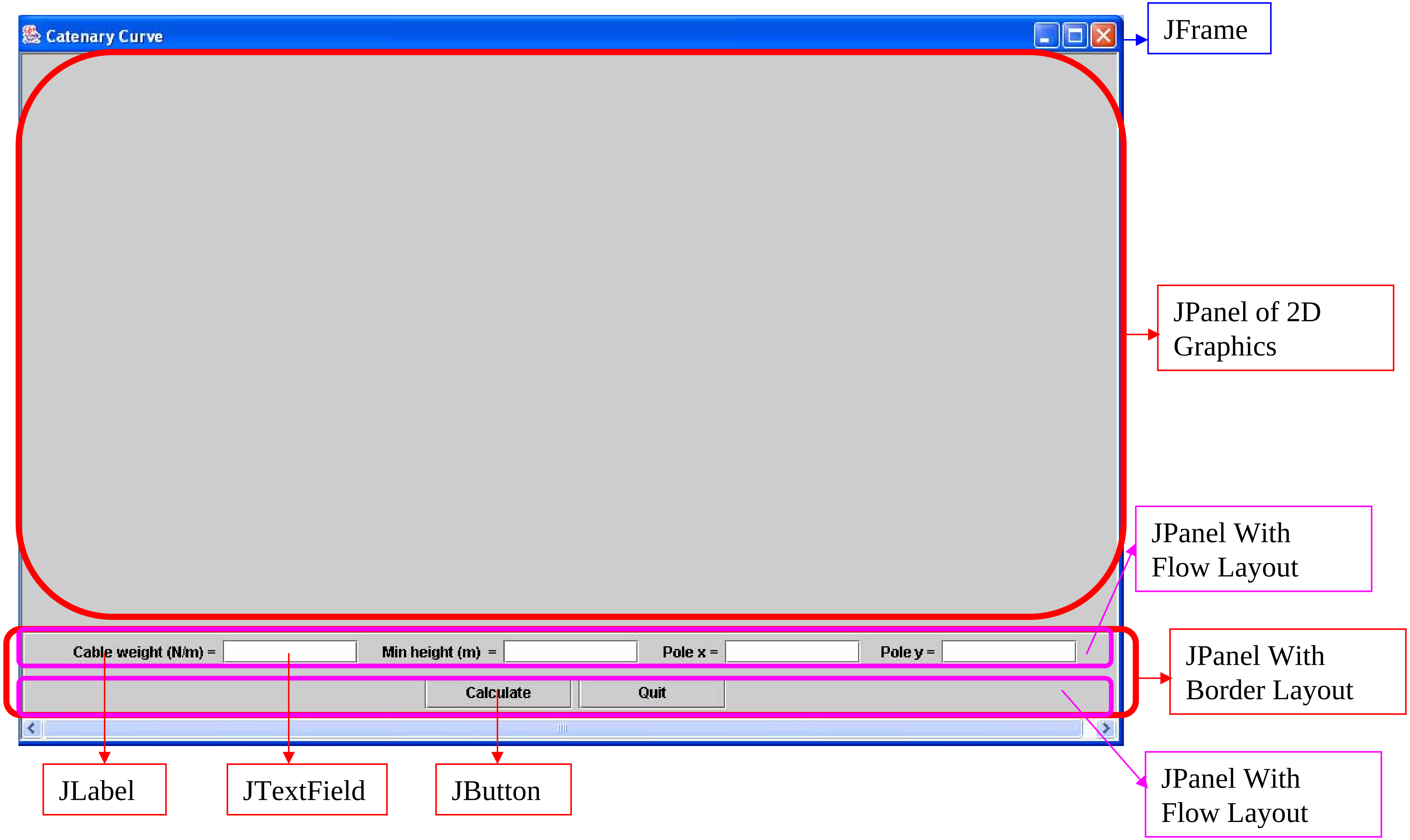
Review – Other Things to Remember

- Component Layout

  - BorderLayout

  - FlowLayout
- **`getContentPane()`**
- **`setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)`**
- **`pack()/setSize()/revalidate()`**
- **`setVisible(true)`**

ProblemSet 6 : GUI for PS2



JFrame

JPanel of 2D Graphics

JPanel With Flow Layout

JPanel With Border Layout

JPanel With Flow Layout

JLabel

JTextField

JButton

PS6: Three Classes

- CatenaryModel
  - Data + algorithms (reuse methods from PS2)
- CatenaryView (To be added in PS7)
  - Extends JPanel
  - Draws 2D Graph
- CatenaryController
  - Extends JFrame
  - Contains input fields and buttons
  - Event Handling (To be added in PS7)

PS6: Design CatenaryModel

- CatenaryModel should be able to provide all the information needed to draw the 2D graph
  - Data:
    - User inputs: unitWeight, minHeight, poleHeight, poleDistance
    - Calculation results: tension
  - Methods:
    - Constructor
    - Method to calculate tension
    - Method to compute y given x
    - Getters and setters