Events and Their Handling

Object Oriented Programming
2024 First Semester
Shin-chi Tadaki (Saga University)

1. Event driven
2. Example: `guiWithAction`
3. Example: `ColorChoice`
4. Class design for `ColorPanel` class

Today's program

`https://github.com/oop-mc-saga/ColorChoice`

Event Driven

- Events in GUI applications are generated by
  - User's action though mouse, keyboard
  - Changes in widgets, etc.
- Mechanisms driven by events are called *Event Driven*.
- Event driven mechanisms are common in GUI applications

Event Driven in Java

- Event classes are defined as extensions of `java.util.EventObject`.
- `EventListener` (receivers of events) classes are defined as extensions of `java.util.EventListener`.
- `EventListener` instances have methods for handling events.

Actions in guiWithAction
Examples of events and event handling

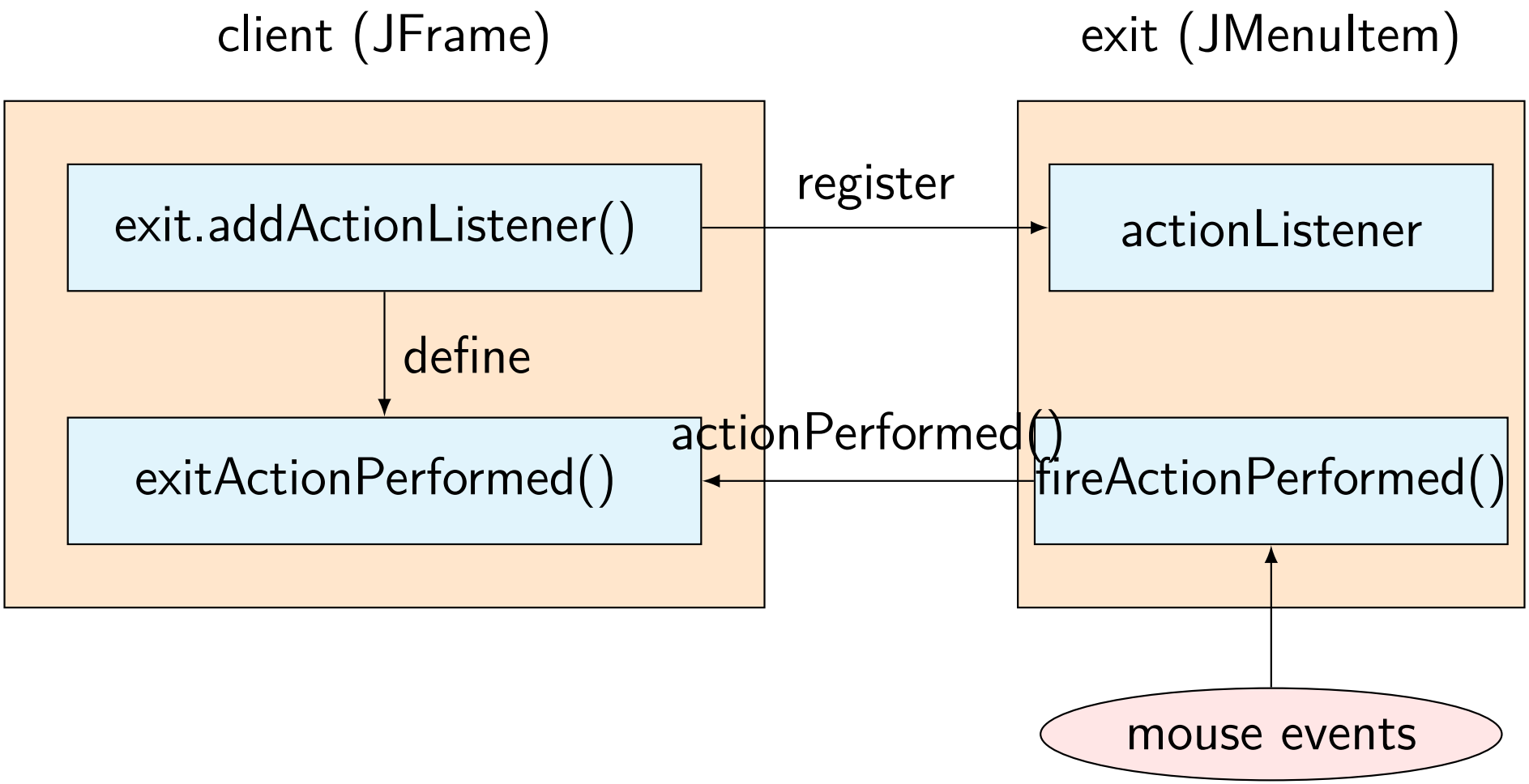
- guiWithAction responds to button events.
- Registering actionListener
  - JFrame: `exit.addActionListener()`
  - JMenuItem: Register as an instance of `ActionListener`
- Event handling
  - JFrame: call `exitActionPerformed()`
  - JMenuItem: call `actionPerformed()` in `fireActionListener()`

Code in guiWithAction

```
1  exit.addActionListener(new java.awt.event.ActionListener() {
2      public void actionPerformed(java.awt.event.ActionEvent evt) {
3          exitActionPerformed(evt);
4      }
5  });
```

- The `JMenuItem` instance `exit` has a method `addActionListener()`.
- The method registers an instance of the event listener interface `ActionListener`.
- The method `actionPerformed()` is implemented to call `exitActionPerformed()`.

Relations between widgets and actions in guiWithAction



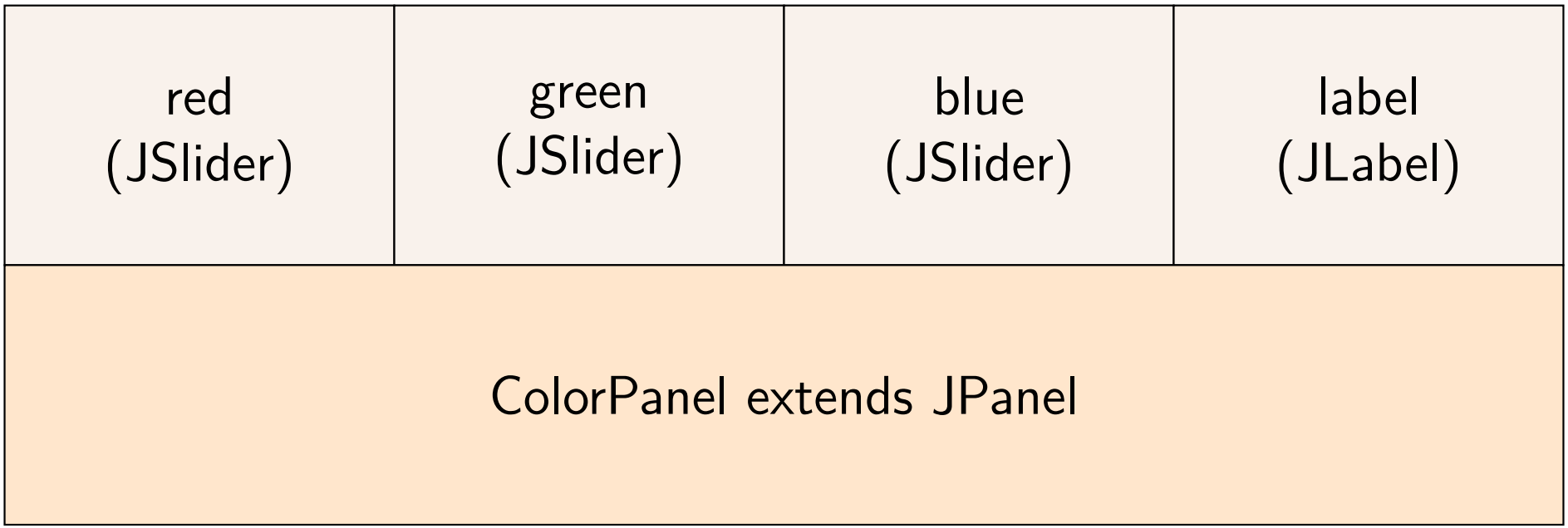
Actions in event generators

- An instance or a list of instances of action listeners
- `addActionListener()` method registers listeners
- At an occurrence of events
  - Notify event to listeners through `fireActionPerformed()`
  - In the `fireActionPerformed()` method, call `actionPerformed()` of each listener.

Example: ColorChoice

- The application specifies color by rgb components using sliders.
- Sliders
  - `JSlider` class
  - Generate `ChangeEvent`
- Place three color sliders for rgb components in a panel
  - Place `JSlider` instances in `JPanel`
  - `JPanel` usually does not generate `ChangeEvent`
  - Need to define event handling mechanisms

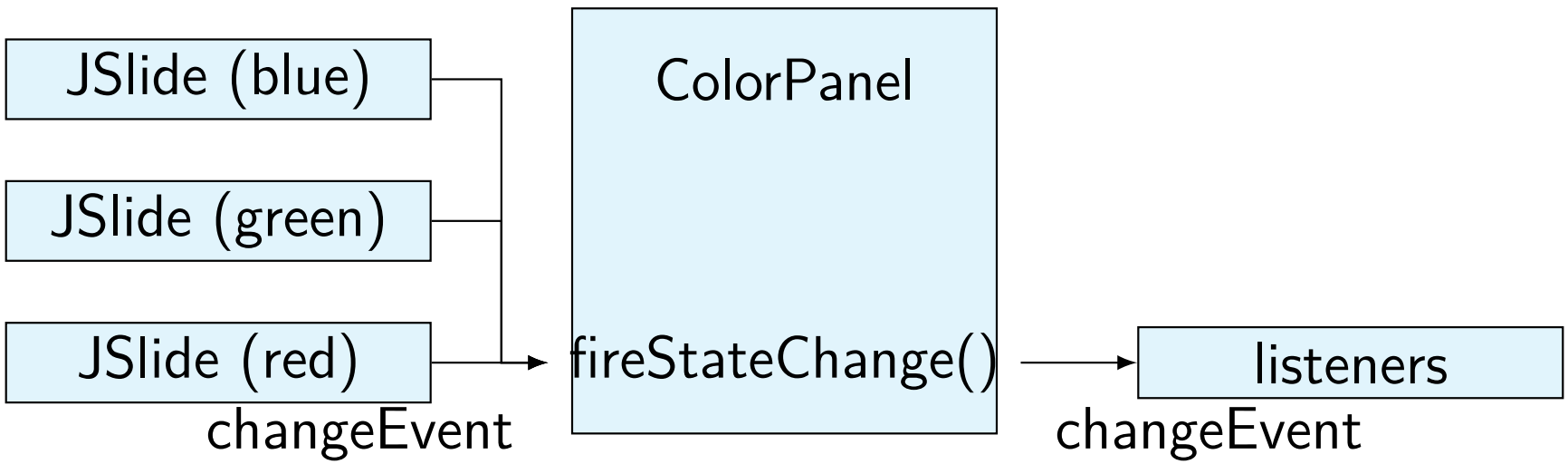
Widget composition



Class design for ColorPanel class

- List of changeListener instances
- fireStateChange() method
  - notify changes to listeners
- Three JSlider instances for rgb
  - Each stateChange() calls fireStateChange() method

Event propagation in ColorPanel



ColorPanel

```
1  public ColorPanel() {
2      initComponents();
3      listeners = new ArrayList<>();
4      setColor();
5      redSlider.addChangeListener(e->fireStateChanged(e));
6      greeSlider.addChangeListener(e->fireStateChanged(e));
7      blueSlider.addChangeListener(e->fireStateChanged(e));
8  }
```

- Define actions as calling `fireStateChanges()` at state change events of sliders.
- Instances of `ChangeListener` are registered using lambda expressions.
  - The interface `ChangeListener` has a single method `stateChanged()`.

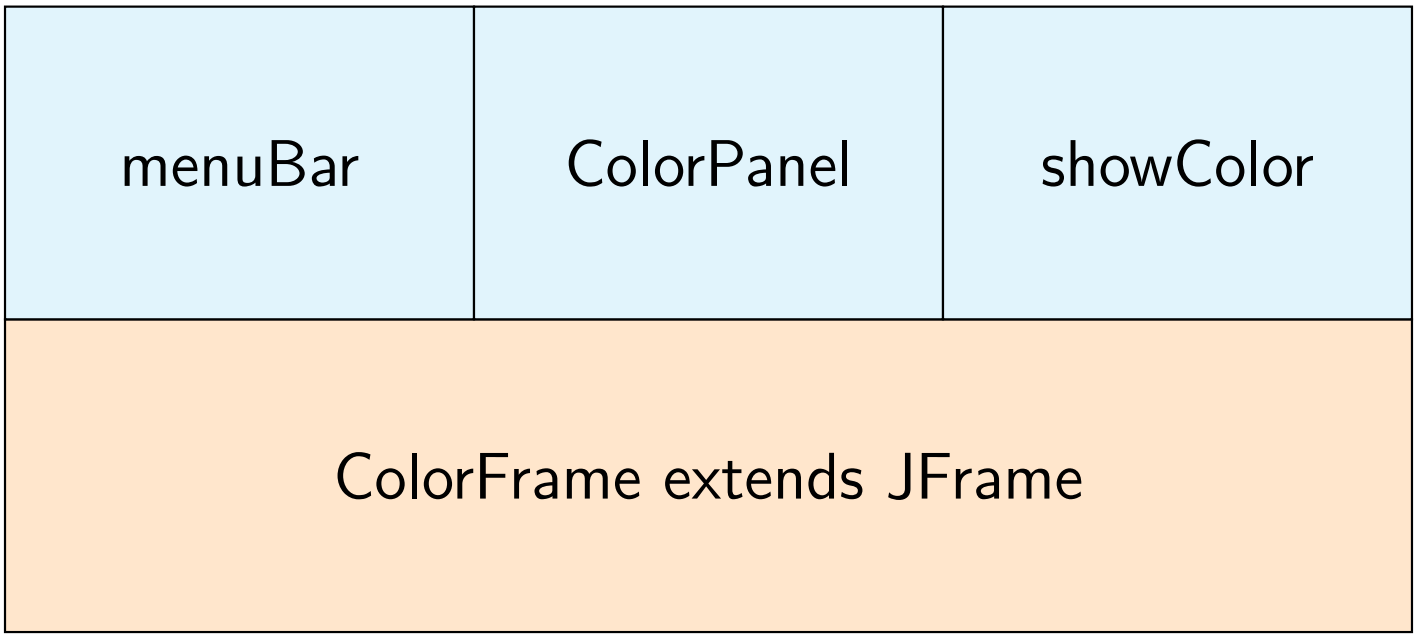
ColorPanel

```
1   /**
2   * Add change listener
3   *
4   * @param listener
5   */
6   public void addChangeListener(ChangeListener listener) {
7       listeners.add(listener);
8   }
9
10  /**
11  * Notify listeners of event
12  *
13  * @param e
14  */
15  protected void fireStateChanged(ChangeEvent e) {
16      setColor();
17      listeners.forEach(li -> li.stateChanged(e));
18  }
```

Notify an event to listeners at state change events.

```
1   /**
2   * Get color values from three sliders and set color
3   */
4   private void setColor() {
5       int r = redSlider.getValue();
6       int g = greeSlider.getValue();
7       int b = blueSlider.getValue();
8       color = new Color(r, g, b);
9       setLabel(r, g, b);
10  }
```

Widgets in ColorFrame



Exercise

Get color from `Colors` and set the color to `colorPanel` (see quiz).