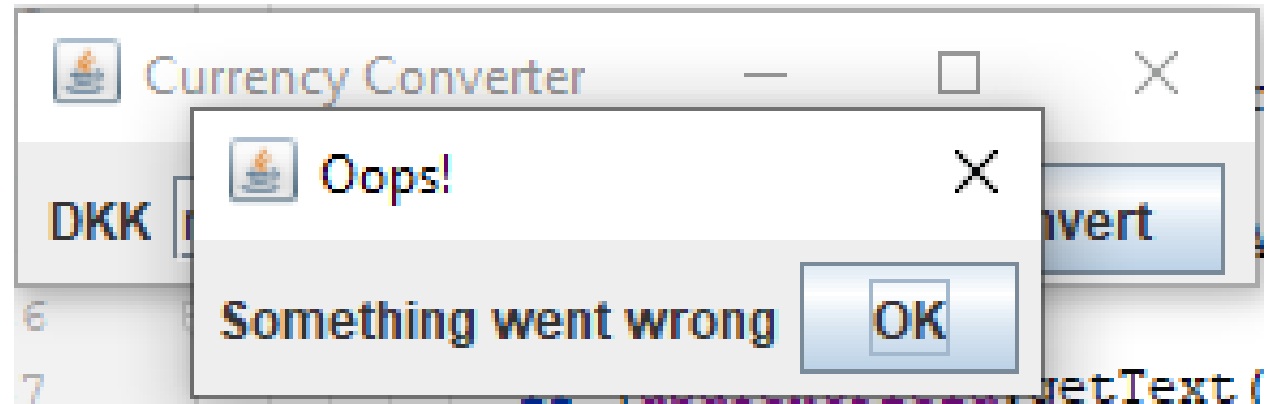


Java Swing for cool GUIs

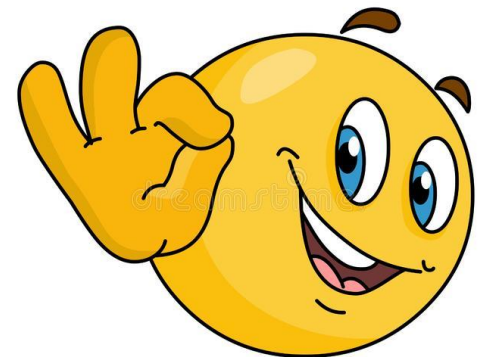
Module 2

Patrick Agergaard, paag@kea.dk



Learning goals

- Refactoring
- `validate()` and `repaint()`
- `JButton`
- `ActionListener`
- Anonymous `ActionListener`
- `ActionListener` in separate class



Refactoring

The bottom of the slide features two horizontal blue bars. The first bar is a solid medium blue rectangle spanning most of the width. The second bar is a slightly lighter blue rectangle that starts where the first bar ends, creating a stepped effect on the right side of the slide.

Refactoring.com

a disciplined technique for
restructuring an existing body of code,
altering its internal structure without
changing its external behavior

<https://refactoring.com/>



EXERCISE

Exercise 2-1: Refactoring

Refactoring

```
public class HelloWorldRefactored {  
    private JFrame frame;  
    private JLabel label;  
  
    HelloWorldRefactored() {  
        frame = new JFrame("HelloWorldRefactored");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.add(label = new JLabel("The label"));  
        frame.setSize(300, 200);  
        frame.setLocationRelativeTo(null);  
        frame.setVisible(true);  
    }  
  
    void run() {  
        try { Thread.sleep(1000); } catch (Exception e) { }  
        label.setText("Label's new text");  
    }  
  
    public static void main(String[] args) {  
        new HelloWorldRefactored().run();  
    }  
}
```

validate() and repaint()



EXERCISE

Exercise 2-2:
ValidateAndRepaint

validate() and repaint()

```
ValidateAndRepaint() {  
    frame = new JFrame("ValidateAndRepaint");  
    frame.add(panel = new JPanel());  
    panel.add(label = new JLabel("The label"));  
    frame.setSize(300, 250);  
    frame.setLocationRelativeTo(null);  
    frame.setVisible(true);  
}  
  
void run() {  
    JButton button = new JButton("Click me");  
    panel.add(button);  
    frame.validate();  
    frame.repaint();  
}
```

JButton





EXERCISE

Exercise 2-3:

Add a JButton - or two

HelloButton

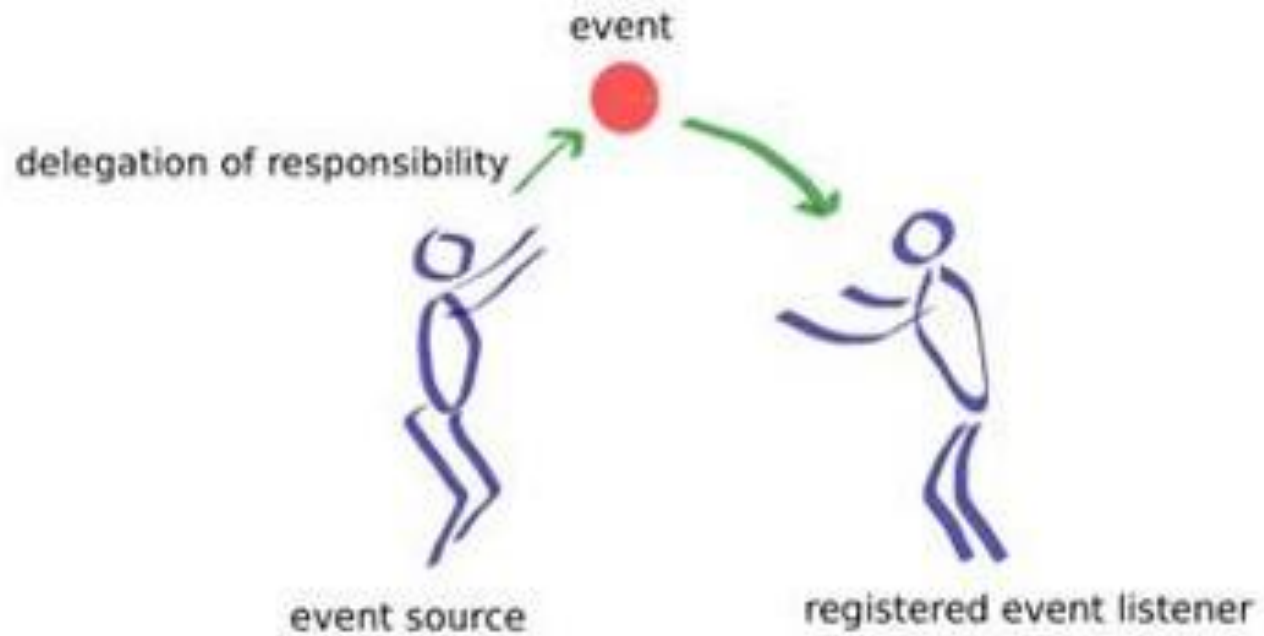
```
public class HelloButton {  
    private JFrame frame;  
    private JPanel panel;  
    private JButton button;  
  
    HelloButton() {  
        frame = new JFrame("HelloButton");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(300, 250);  
        frame.setLocationRelativeTo(null);  
        frame.add(panel = new JPanel());  
        panel.add(button = new JButton("Click me!"));  
        frame.setVisible(true);  
    }  
  
    void run() { }  
  
    public static void main(String[] args) {  
        new HelloButton().run();  
    }  
}
```



YOU WOULDN'T LET THIS
HAPPEN TO YOUR PHONE.
DON'T LET IT HAPPEN TO YOU EITHER.
SELF-CARE IS A PRIORITY
NOT A LUXURY.

ActionListener

The bottom of the slide features two horizontal blue bars. The first bar is a solid medium blue rectangle. The second bar is a slightly lighter shade of blue, positioned to the right of the first bar and slightly offset downwards, creating a layered or 3D effect.



1. The OS fires an EVENT
2. The LISTENER picks up on it
3. The COMPONENT (button, etc.) invokes an action (method)

Interface ActionListener

All Superinterfaces:

EventListener

All Known Subinterfaces:

Action

```
public interface ActionListener  
extends EventListener
```

The listener interface for receiving action events. The class that is interested in processing an action event implements this interface, and the object created with that class is registered with a component, using the component's `addActionListener` method. When the action event occurs, that object's `actionPerformed` method is invoked.

Method Summary

Methods

Modifier and Type	Method and Description
void	<code>actionPerformed(ActionEvent e)</code> Invoked when an action occurs.

Separate ActionListener class

The bottom of the slide features two horizontal blue bars. The first bar is a solid medium blue rectangle. The second bar is a slightly lighter blue rectangle that is offset to the right and appears to be floating above the first bar, creating a layered effect.

HelloActionListenerInterface

```
public class HelloActionListenerInterface {
    private JFrame frame; private JPanel panel; private JButton button;

    HelloActionListenerInterface() {
        frame = new JFrame("HelloActionListener");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 250); frame.setLocationRelativeTo(null);
        frame.add(panel = new JPanel());
        panel.add(button = new JButton("Click me!"));
        frame.setVisible(true);
        button.addActionListener(new AL2());
    }

    public static void main(String[] args) { new HelloActionListenerInterface(); }
}

class AL2 implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        System.out.println("Click registered by AL2.");
    }
}
```



EXERCISE

Exercise 2-6:

Add an ActionListener from a separate class

Nested ActionListener class

The bottom of the slide features two horizontal blue bars. The first bar is a solid medium blue rectangle. The second bar is a slightly lighter blue rectangle that overlaps the first one on the right side, creating a layered effect.

HelloActionListener

```
public class HelloActionListener {  
    private JFrame frame;  
    private JPanel panel;  
    private JButton button;  
  
    HelloActionListener() {  
        frame = new JFrame("HelloActionListener");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(300, 250);  
        frame.setLocationRelativeTo(null);  
        frame.add(panel = new JPanel());  
        panel.add(button = new JButton("Click me!"));  
        frame.setVisible(true);  
    }  
}
```

```
    ActionListener al = new ActionListener() {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            button.setText("I LOVE YOUR CLICKS!!!!");  
        }  
    };  
    button.addActionListener(al);  
}
```

```
    public static void main(String[] args) { new HelloActionListener(); }  
}
```



EXERCISE

Exercise 2-4:
Add an ActionListener

Anonymous ActionListener

The bottom of the slide features two horizontal blue bars. The first bar is a solid medium blue rectangle. The second bar is a slightly lighter blue rectangle that starts to the right of the first bar's end and extends to the right edge of the slide, creating a stepped effect.

HelloAnonymousActionListener

```
public class HelloAnonymousActionListener {
    private JFrame frame;
    private JPanel panel;
    private JButton button;

    HelloAnonymousActionListener() {
        frame = new JFrame("HelloActionListener");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 250);
        frame.setLocationRelativeTo(null);
        frame.add(panel = new JPanel());
        panel.add(button = new JButton("Click me!"));
        frame.setVisible(true);

        button.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                button.setText("Anonymous AL clicked!");
            }
        });
    }

    public static void main(String[] args) { new HelloAnonymousActionListener(); }
}
```




EXERCISE

Exercise 2-5:
Add an ActionListener

HelloActionListenerNestedClass

```
public class HelloActionListenerNestedClass {  
    private JFrame frame; private JPanel panel; private JButton button;
```

```
    private class AL3 implements ActionListener {  
        public void actionPerformed(ActionEvent e) {  
            System.out.println("AL3 here!");  
        }  
    }
```

```
    HelloActionListenerNestedClass() {  
        frame = new JFrame("HelloActionListener");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(300, 250);  
        frame.setLocationRelativeTo(null);  
        frame.add(panel = new JPanel());  
        panel.add(button = new JButton("Click me!"));  
        frame.setVisible(true);
```

```
        button.addActionListener(new AL3());
```

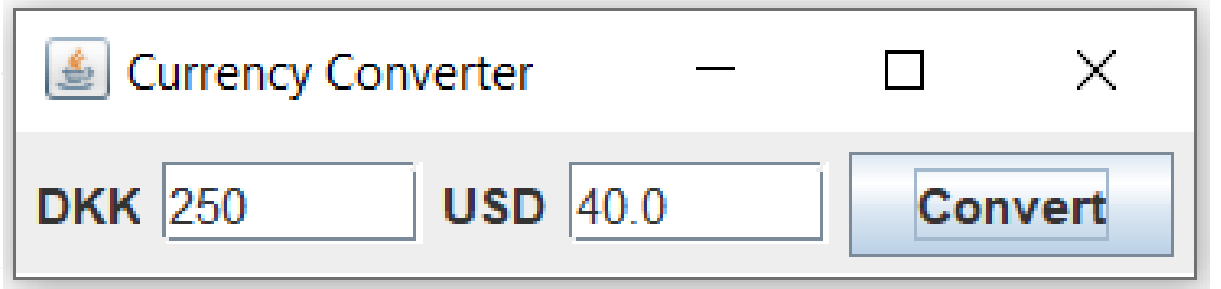
```
    }
```

```
    public static void main(String[] args) {  
        new HelloActionListenerNestedClass();  
    }
```

```
}
```

EXERCISE

Exercise 2-7: Currency Converter



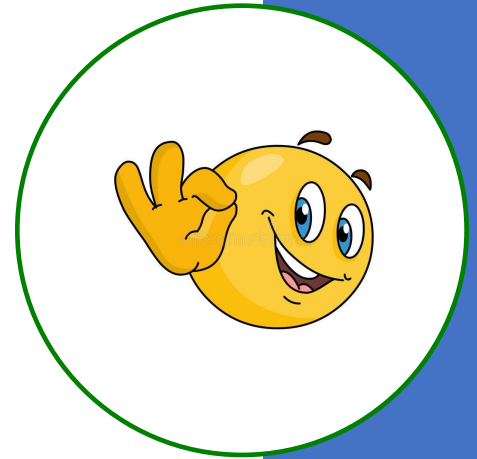
A screenshot of a Windows-style application window titled "Currency Converter". The window has a standard title bar with a minimize button, a maximize button (disabled), and a close button. The main content area contains two currency input fields. The first field is labeled "DKK" and contains the value "250". The second field is labeled "USD" and contains the value "40.0". To the right of these fields is a blue button with the text "Convert".

From	To
DKK 250	USD 40.0

Convert

What was most important today?

- Refactoring
- `validate()` and `repaint()`
- `JButton`
- `ActionListener`
- Anonymous `ActionListener`
- Separate `ActionListener` class





QUESTIONS

- JButton
- Creating and adding ActionListener
- (ActionEvent)