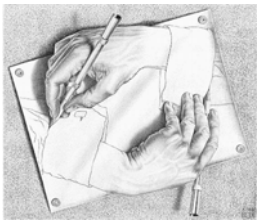
Lecture 24



Event handling

Chapter 17
Event handling

Introduction

- Our graphical user interface (GUI) programs are event based

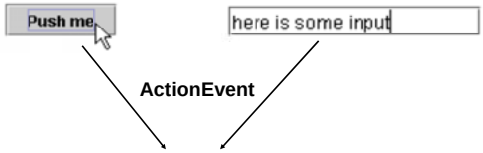
Java Events

- There are different object types in Java to represent different kinds of events

Event	Occurs when
ActionEvent	click a JButton, or press "enter" in a JTextField
MouseEvent	press or release the mouse button, or move the mouse
KeyEvent	press a key on the keyboard

ActionEvents

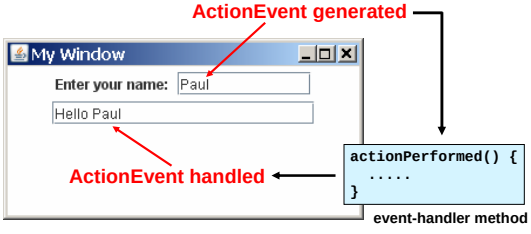
- We process ActionEvents by defining a method which is automatically called whenever an ActionEvent occurs



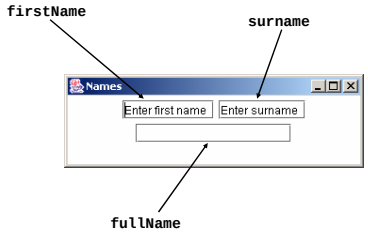

```
public void actionPerformed(ActionEvent e) {
            ...
}
```

actionPerformed()

- The actionPerformed() method is an "event-handler" because it *handles* the event



ActionEvents from JTextFields



```
public class MyJPanel extends JPanel {

	private JTextField firstName;
	private JTextField surname;
	private JTextField fullName;

	public MyJPanel() {
		firstName = new JTextField("Enter first name");
		surname = new JTextField("Enter surname");
		fullName = new JTextField(15);

		add(firstName);
		add(surname);
		add(fullName);



	}









}
```

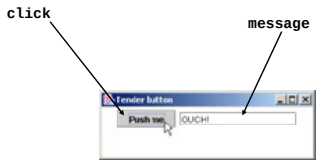
```
public class MyJPanel extends JPanel implements ActionListener {

	private JTextField firstName;
	private JTextField surname;
	private JTextField fullName;

	public MyJPanel() {
		firstName = new JTextField("Enter first name");
		surname = new JTextField("Enter surname");
		fullName = new JTextField(15);

		add(firstName);
		add(surname);
		add(fullName);



	}

	public void actionPerformed(ActionEvent e){




	}
}
```

```
public class MyJPanel extends JPanel implements ActionListener {

	private JTextField firstName;
	private JTextField surname;
	private JTextField fullName;

	public MyJPanel() {
		firstName = new JTextField("Enter first name");
		surname = new JTextField("Enter surname");
		fullName = new JTextField(15);

		add(firstName);
		add(surname);
		add(fullName);

		firstName.addActionListener(this);
		surname.addActionListener(this);
	}

	public void actionPerformed(ActionEvent e){




	}
}
```

```
public class MyJPanel extends JPanel implements ActionListener {

	private JTextField firstName;
	private JTextField surname;
	private JTextField fullName;

	public MyJPanel() {
		firstName = new JTextField("Enter first name");
		surname = new JTextField("Enter surname");
		fullName = new JTextField(15);

		add(firstName);
		add(surname);
		add(fullName);

		firstName.addActionListener(this);
		surname.addActionListener(this);
	}

	public void actionPerformed(ActionEvent e){
		String first = firstName.getText();
		String second = surname.getText();
		String full = first + " " + second;
		fullName.setText(full);
	}
}
```

```
public class MyJPanel extends JPanel implements ActionListener {

	private JTextField firstName;
	private JTextField surname;
	private JTextField fullName;

	public MyJPanel() {
		firstName = new JTextField("Enter first name");
		surname = new JTextField("Enter surname");
		fullName = new JTextField(15);

		add(firstName);
		add(surname);
		add(fullName);

		firstName.addActionListener(this);
		surname.addActionListener(this);
	}

	public void actionPerformed(ActionEvent e){
		String first = firstName.getText();
		String second = surname.getText();
		String full = first + " " + second;
		fullName.setText(full);
	}
}
```

ActionEvents from JButtons



```
public class MyJPanel extends JPanel {

	private JTextField message;
	private JButton click;

	public MyJPanel() {
		click = new JButton("Push me");
		message = new JTextField(15);

		add(click);
		add(message);


	}



}
```

```
public class MyJPanel extends JPanel implements ActionListener {

	private JTextField message;
	private JButton click;

	public MyJPanel() {
		click = new JButton("Push me");
		message = new JTextField(15);

		add(click);
		add(message);


	}

	public void actionPerformed(ActionEvent e){

	}
}
```

```
public class MyJPanel extends JPanel implements ActionListener {

	private JTextField message;
	private JButton click;

	public MyJPanel() {
		click = new JButton("Push me");
		message = new JTextField(15);

		add(click);
		add(message);

		click.addActionListener(this);
	}

	public void actionPerformed(ActionEvent e){

	}
}
```

```
public class MyJPanel extends JPanel implements ActionListener {

	private JTextField message;
	private JButton click;

	public MyJPanel() {
		click = new JButton("Push me");
		message = new JTextField(15);

		add(click);
		add(message);

		click.addActionListener(this);
	}

	public void actionPerformed(ActionEvent e){
		message.setText("OUCH!");
	}
}
```

```
public class MyJPanel extends JPanel implements ActionListener {

	private JTextField message;
	private JButton click;

	public MyJPanel() {
		click = new JButton("Push me");
		message = new JTextField(15);

		add(click);
		add(message);

		click.addActionListener(this);
	}

	public void actionPerformed(ActionEvent e){
		message.setText("OUCH!");
	}
}
```

The Scope of the JTextFields

- The JTextFields (in the previous examples) are instance variables because:
  - they are referred to in the constructor method of the MyJPanel class when they are created
  - they are also referred to in the actionPerformed() method so that we can get and set the text in them.
- For this reason they must be defined as instance variables so that both methods can refer to them

Which component?

- In the body of the actionPerformed() method, we can determine which component caused the event by calling:

`e.getSource()`


```
public void actionPerformed(ActionEvent e) {
	if (e.getSource() == happyButton) {
		stateOfMind.setText("You are happy.");
	} else if (e.getSource() == sadButton) {
		stateOfMind.setText("You are sad.");
	}
}
```