Human-Computer Interaction

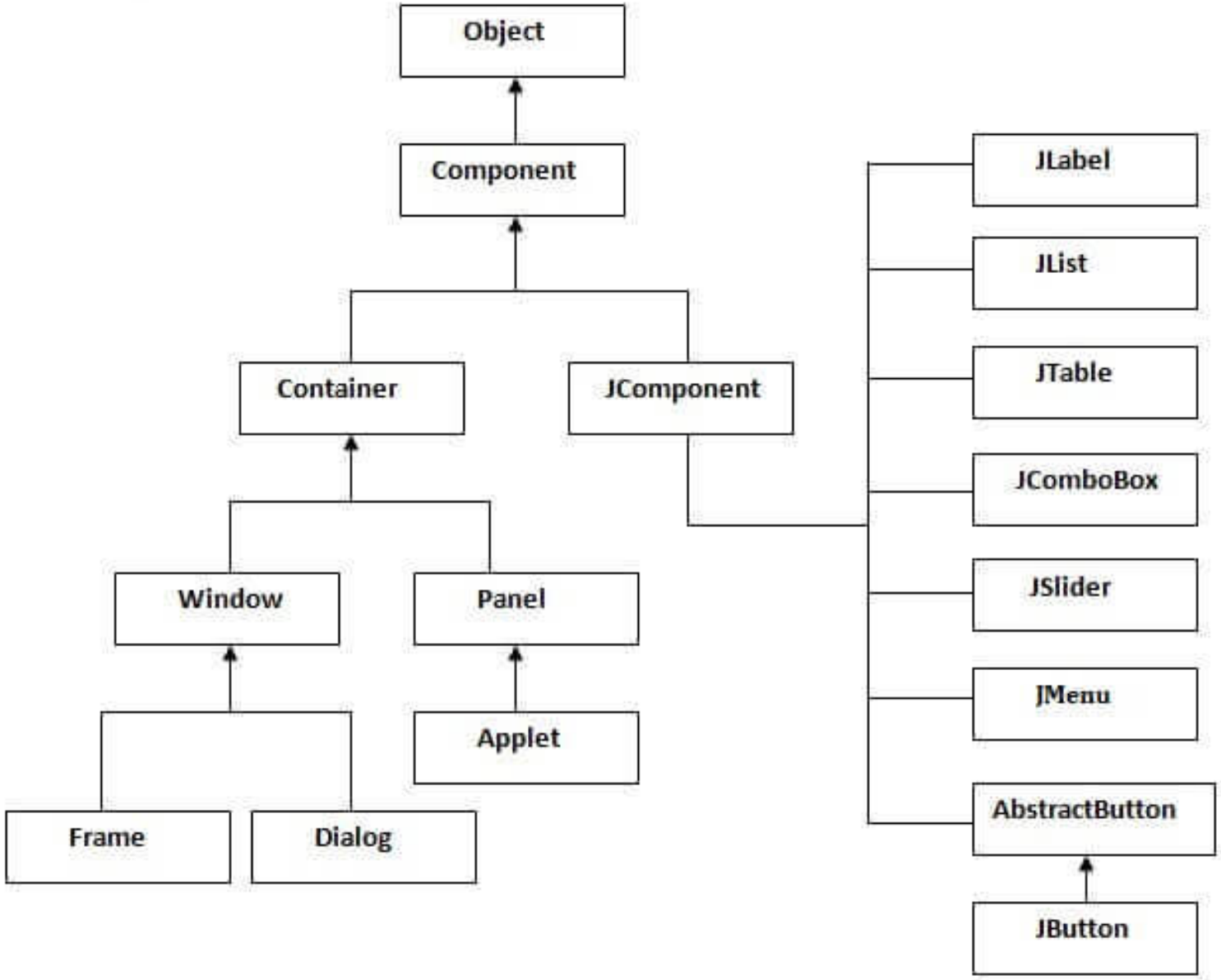
Lab session

Introduction to Java Swing

Java Swing

- **Java Swing** is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.
- Unlike AWT, Java Swing provides platform-independent and lightweight components.
- The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

Hierarchy of Java Swing classes



Commonly used Methods of Component class

Method	Description
public void add(Component c)	add a component on another component.
public void setSize(int width,int height)	sets size of the component.
public void setLayout(LayoutManager m)	sets the layout manager for the component.
public void setVisible(boolean b)	sets the visibility of the component. It is by default false.

Java Swing Examples

- There are two ways to create a frame:
- By creating the object of Frame class (association)
- By extending Frame class (inheritance)
- We can write the code of swing inside the main(), constructor or any other method.

- JButton class declaration
- **public class** JButton **extends** AbstractButton **implements** Accessible

Methods	Description
void setText(String s)	It is used to set specified text on button
String getText()	It is used to return the text of the button.
void setEnabled(boolean b)	It is used to enable or disable the button.
void setIcon(Icon b)	It is used to set the specified Icon on the button.
Icon getIcon()	It is used to get the Icon of the button.
void setMnemonic(int a)	It is used to set the mnemonic on the button.
void addActionListener(ActionListener a)	It is used to add the action listener to this object.

- **public class** JLabel **extends** JComponent **implements** SwingConstants, Accessible
- **public class** JTextField **extends** JTextComponent **implements** SwingConstants
- **public class** JTextArea **extends** JTextComponent
- **public class** JPasswordField **extends** JTextField
- **public class** JMenuBar **extends** JComponent **implements** MenuElement, Accessible

  public class JMenu **extends** JMenuItem **implements** MenuElement, Accessible
- **public class** JMenuItem **extends** AbstractButton **implements** Accessible, MenuElement
- **public class** JColorChooser **extends** JComponent **implements** Accessible

- **public class** JDialog **extends** Dialog **implements** WindowConstants, Accessible, RootPaneContainer

Create a Frame

- Use the class JFrame to create an object of JFrame

```
JFrame frame = new JFrame("Login page");
```

Set a size using the setSize() method

```
frame.setSize(400, 400);
```

- Set the visibility to true to make it visible using the setVisible() method.

```
frame.setVisible(true);
```

JOptionPane

- The JOptionPane class is used to provide standard dialog boxes such as message dialog box, confirm dialog box and input dialog box.

- **public class** JOptionPane **extends** JComponent **implements** Accessible

Methods	Description
JDialog createDialog(String title)	It is used to create and return a new parentless JDialog with the specified title.
static void showMessageDialog(Component parentComponent, Object message)	It is used to create an information-message dialog titled "Message".
static void showMessageDialog(Component parentComponent, Object message, String title, int messageType)	It is used to create a message dialog with given title and messageType.
static int showConfirmDialog(Component parentComponent, Object message)	It is used to create a dialog with the options Yes, No and Cancel; with the title, Select an Option.
static String showInputDialog(Component parentComponent, Object message)	It is used to show a question-message dialog requesting input from the user parented to parentComponent.
void setInputValue(Object newValue)	It is used to set the input value that was selected or input by the user.

Example

```
public class Messages {
    JFrame f;
    Messages(){
        f=new JFrame();
        JOptionPane.showMessageDialog(f,"Id is not
Exist","Error",JOptionPane.ERROR_MESSAGE);
    }
    public static void main(String[] args) {
        new Messages();
    }
}
```

Graphics:

public abstract void drawString(String str, int x, int y):

is used to draw the specified string.

public void drawRect(int x, int y, int width, int height):

draws a rectangle with the specified width and height.

public abstract void fillRect(int x, int y, int width, int height):

is used to fill rectangle with the default color and specified width and height.

public abstract void drawOval(int x, int y, int width, int height):

is used to draw oval with the specified width and height.

public abstract void fillOval(int x, int y, int width, int height):

is used to fill oval with the default color and specified width and height.

public abstract void drawLine(int x1, int y1, int x2, int y2):

is used to draw line between the points(x1, y1) and (x2, y2).

public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):

is used draw a circular or elliptical arc.

public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):

is used to fill a circular or elliptical arc.

public abstract void setColor(Color c):

is used to set the graphics current color to the specified color.

public abstract void setFont(Font font):

is used to set the graphics current font to the specified font.

public abstract void setColor(Color c):

is used to set the graphics current color to the specified color.

public abstract void setFont(Font font):

is used to set the graphics current font to the specified font.

public abstract void drawString(String str, int x, int y):

is used to draw the specified string.

public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):

is used draw the specified image.

Example

```
JFrame f=new JFrame(" Lab Sessions");
final JTextField tf=new JTextField();
tf.setBounds(50,50, 150,20);
JButton b=new JButton("Click Here");
b.setBounds(50,100,95,30);
```

- f.add(b);
- f.add(tf);
  f.setSize(400,400);

- f.setLayout(null);
  f.setVisible(true);

```
b.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
      tf.setText("Welcome to Class HCI");
    }
});
```

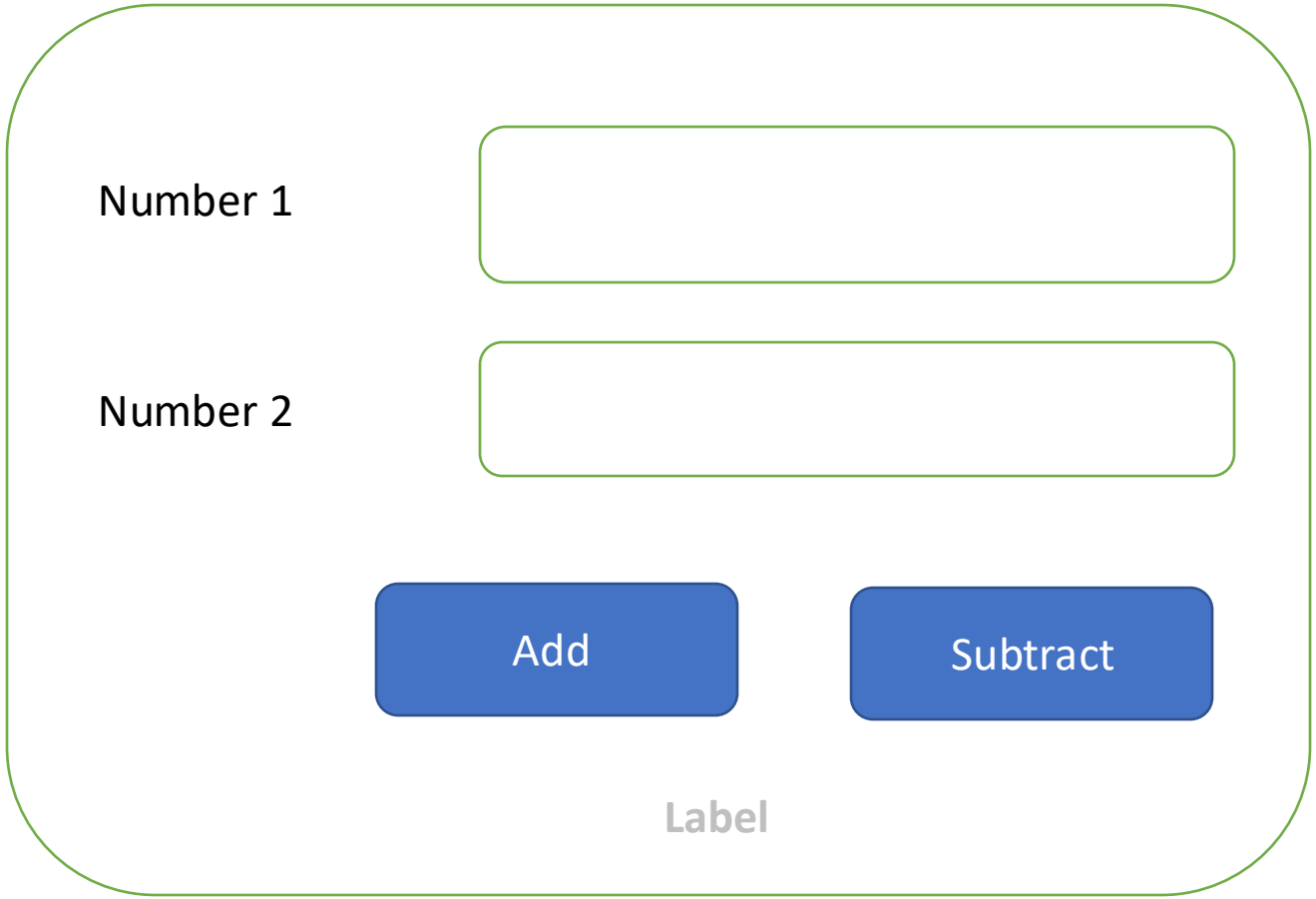
Label

- JLabel lb;
  lb=new JLabel("First lab session");
  lb.setBounds(100,150, 100,30);
  f.add(lb);

Password text field

- final JPasswordField value = new JPasswordField();
  value.setBounds(100,75,100,30);

Activity



Using a constructor

- Main(){

```
JFrame f= new JFrame();
tf1=new JTextField();
tf1.setBounds(50,50,150,20);
tf2=new JTextField();
tf2.setBounds(50,100,150,20);
tf3=new JTextField();
tf3.setBounds(50,150,150,20);
tf3.setEditable(false);
b1=new JButton("+");
b1.setBounds(50,200,50,50);
b2=new JButton("-");
b2.setBounds(120,200,50,50);
b1.addActionListener(this);
b2.addActionListener(this);
```

```
 f.add(tf1);
f.add(tf2);
f.add(tf3);
f.add(b1);
f.add(b2);
    f.setSize(300,300);
    f.setLayout(null);
    f.setVisible(true);
}
```

```
public void actionPerformed(ActionEvent e) {
    String s1=tf1.getText();
    String s2=tf2.getText();
    int a=Integer.parseInt(s1);
    int b=Integer.parseInt(s2);
    int c=0;
    if(e.getSource()==b1){
        c=a+b;
    }else if(e.getSource()==b2){
        c=a-b;
    }
    String result=String.valueOf(c);
    tf3.setText(result);
}
```

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

Menu

- JMenu menu, submenu;
  JMenuItem i1, i2, i3, i4, i5;
  MenuExample(){
  JFrame f= new JFrame("Menu and MenuItem Example");
  JMenuBar mb=new JMenuBar();
  menu=new JMenu("Menu");

  }

```
submenu=new JMenu("Sub Menu");
    i1=new JMenuItem("Save");
    i2=new JMenuItem("Edit");
    i3=new JMenuItem("Delete");
    i4=new JMenuItem("Item 4");
    i5=new JMenuItem("Item 5");
```

```
menu.add(i1); menu.add(i2); menu.add(i3);
    submenu.add(i4); submenu.add(i5);
    menu.add(submenu);
    mb.add(menu);
    f.setJMenuBar(mb);
    f.setSize(400,400);
```

- f.setLayout(null);
  f.setVisible(true);

Graphics

- public class Main extends Canvas
- public void paint(Graphics g)

- g.drawOval(30,130,50, 60);
  setForeground(Color.*RED*);
  g.fillOval(130,130,50, 60);

```
public static void main(String[] args) {
    Main m=new Main();
    JFrame f=new JFrame();
    f.add(m);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
}
```

Draw a specific image

```
Toolkit t=Toolkit.getDefaultToolkit();
Image i=t.getImage("icon.gif");
g.drawImage(i, 120,100,this);
```

Colour chooser

- Main(){

```
f=new JFrame(“Choose a colour.");
b=new JButton("Pad Color");
b.setBounds(200,250,100,30);
ta=new JTextArea();
ta.setBounds(10,10,300,200);
b.addActionListener(this);
f.add(b);f.add(ta);
f.setLayout(null);
f.setSize(400,400);
f.setVisible(true);
}
```

```
public void actionPerformed(ActionEvent e){
    Color
c=JColorChooser.showDialog(this,"Choose",Color.CYAN);
    ta.setBackground(c);
}
```

Activity

- Use the knowledge you learnt throughout the lectures to Implement the UIs of an app for selling stationaries (2-3 interfaces).

Useful links

- Java Swing https://www.javatpoint.com/java-swing
- Java Fundamentals https://www.javatpoint.com/java-tutorial
  https://www.tutorialspoint.com/java/index.htm

Thank you