

MER Java Foundation Classes (JFC)

Swing

Johan Elsson

Swing

- Många klasser
- Jättemånga klasser
- Vettigt att lära sig dem utantill
 - Tror inte det

Johan Elsson

Hur lära sig?

- Principerna
 - Model - view - controller
 - Observer - observed
- Öva, dvs skriv kod
 - Ofta och mycket
- Titta igenom biblioteken (men försök inte lära dig dem utantill).
- Kolla på exempel

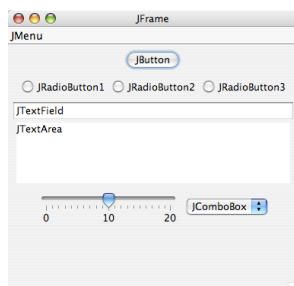
Johan Elsson

Vad är farorna

- Personlig erfarenhet
 - Dålig dokumentation
 - Logik på fel ställe

Johan Elsson

Exempel på några av komponenterna



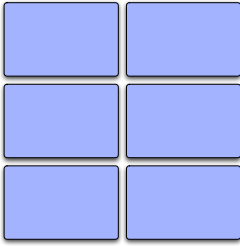
Johan Elsson

LayoutManagers

- BorderLayout - Har vi tittat på
- FlowLayout - Har vi nämnt
- GridLayout
- CardLayout
- GridBagLayout
- etc

Johan Elsson

GridLayout



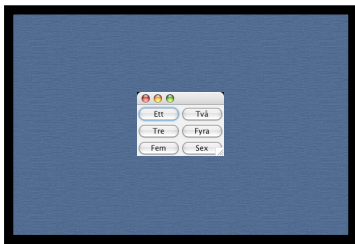
Johan Eliasson

```
public class GridDemo
    extends JFrame
{
    public GridDemo()
    {
        Container c = getContentPane();

        c.setLayout( new GridLayout( 3, 3 ) );
        c.add( new JButton( "Ett" ) );
        c.add( new JButton( "Två" ) );
        c.add( new JButton( "Tre" ) );
        c.add( new JButton( "Fyra" ) );
        c.add( new JButton( "Fem" ) );
        c.add( new JButton( "Sex" ) );

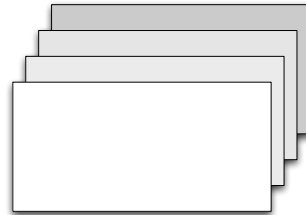
        pack();
        setVisible( true );
    }
}
```

Johan Eliasson



Johan Eliasson

Card layout



Johan Eliasson

```
CardLayout c1 = new CardLayout();
JPanel tp = new JPanel( c1 );

public CardDemo()
{
    Container c = getContentPane();
    JPanel bp = new JPanel();

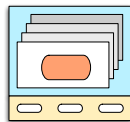
    c.add( tp, BorderLayout.NORTH );
    c.add( bp, BorderLayout.SOUTH );

    tp.add( "one", new JButton( "Ett" ) );
    tp.add( "two", new JButton( "Två" ) );
    tp.add( "three", new JButton( "Tre" ) );
    tp.add( "four", new JButton( "Fyra" ) );
    tp.add( "five", new JButton( "Fem" ) );
    tp.add( "six", new JButton( "Sex" ) );

    JButton btn = new JButton( "First" );
    btn.addActionListener( new ActionListener() {
        public void actionPerformed( ActionEvent ae )
        { c1.first( tp ); } } );
    bp.add( btn );

    ...

    pack();
    setVisible( true );
}
```

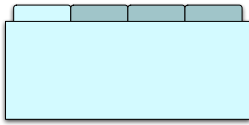


Johan Eliasson



Johan Eliasson

Tabbed pane



Johan Eliasson

```
public class TabbedDemo
    extends JFrame
{
    public TabbedDemo( )
    {
        super( "Tabbed Demo" );
        JTabbedPane tp = new JTabbedPane( );
        setContentPane( tp );

        JPanel jp = new JPanel( );
        jp.add( new JButton( "Ett" ) );
        tp.addTab( "One", jp );

        jp = new JPanel( );
        jp.add( new JButton( "Två" ) );
        tp.addTab( "Two", jp );

        jp = new JPanel( );
        jp.add( new JButton( "Tre" ) );
        tp.addTab( "Three", jp );

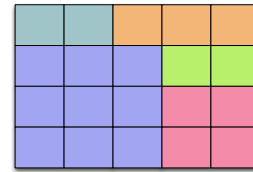
        pack();
        setVisible( true );
    }
}
```

Johan Eliasson



Johan Eliasson

GridBagLayout



Johan Eliasson

"GridBagLayout was contributed to Javasoft by a programmer who wanted to support the Java effort. It was intended as a proof that the AWT offered enough features for programmers to write their own layout managers.

It wasn't designed with human factors and ease of use in mind. If it bothers you (it bothers me) then just don't use it.

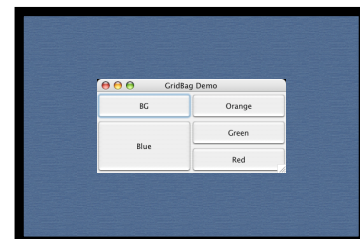
Create your GUI on several panels and use the other layout managers as appropriate to get the exact effect you want.

JavaSoft's AWT team project leader has said that eventually another layout manager will be added that does similar things to GBL, but is more intuitive.

Bottom line: nobody has to waste any effort on GBL, there are better alternatives available."

Peter van der Linden

Johan Eliasson



Johan Eliasson



Johan Eliasson

Fler LayoutManagers

- Finns en bunt till
- Undersök de klasser som implementerar någon av interfacen LayoutManager eller LayoutManager2

Johan Eliasson

Händelser

- Olika typer
 - AdjustmentEvent
 - ComponentEvent
 - MouseEvent

Johan Eliasson

Rita

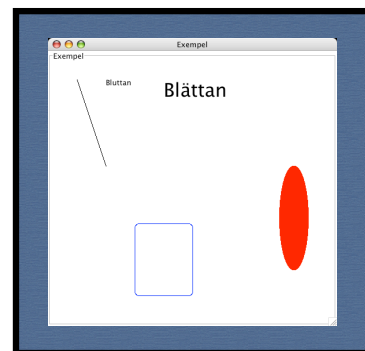
- Händelsestyrt - styr inte själv
- paint
- update
- paintComponent
- repaint, repaint(x)

Johan Eliasson

Graphics

- drawString
- drawLine
- drawRect
- drawPolygon
- fillRect
- fillPolygon
- setColor
- setFont

Johan Eliasson



Johan Eliasson

```

public class DrawingPanel
    extends JPanel
{
    public DrawingPanel( )
    {
        setBackground( Color.white );
        setBorder( BorderFactory.createTitledBorder( "Exempel" ) );
    }

    public void paintComponent( Graphics g )
    {
        super.paintComponent( g );
        g.drawString( "Bluttan", 100, 60 );

        Font cFont    = g.getFont();
        Font newFont  = new Font( cFont.getName(), cFont.getStyle(),
                                cFont.getSize() + 20 );

        g.setFont( newFont );
        g.drawString( "Blättan", 200, 80 );
        g.drawLine( 50, 50, 100, 200 );
        g.setColor( Color.blue );
        g.drawRoundRect( 150, 300, 100, 125, 15, 15 );
        g.setColor( Color.red );
        g.fillOval( 400, 200, 50, 180 );
    }
}

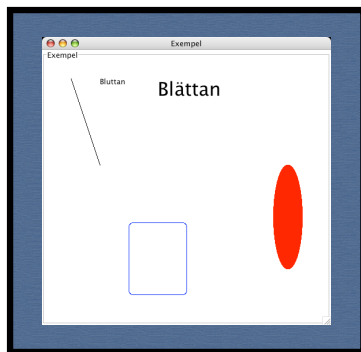
```

Johan Eliasson

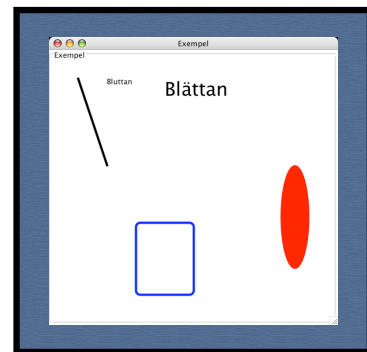
Java 2D

- Förbättrad grafik så att det blir lite snyggare
- Anti-aliasing
- Fyll med mönster, gradienter, etc
- Bättre typsnittshantering
- figurer, linjer, transformeringar

Johan Eliasson



Johan Eliasson



Johan Eliasson

```

public void paintComponent( Graphics g )
{
    super.paintComponent( g );
    Graphics2D g2 = (Graphics2D) g;
    g2.setRenderingHint( RenderingHints.KEY_ANTIALIASING,
                        RenderingHints.VALUE_ANTIALIAS_ON );
    g2.setStroke( new BasicStroke( 4.0f ) );
    g2.drawString( "Bluttan", 100, 60 );

    Font cFont    = g2.getFont();
    Font newFont  = new Font( cFont.getName(),
                            cFont.getStyle(),
                            cFont.getSize() + 20 );

    g2.setFont( newFont );
    g2.drawString( "Blättan", 200, 80 );

    Line2D line = new Line2D.Double( 50.0, 50.0, 100.0, 200.0 );
    g2.draw( line );

    g2.setPaint( Color.blue );
    RoundRectangle2D rect = new RoundRectangle2D.Double(
        150.0, 300.0, 100.0, 125.0, 15.0, 15.0 );

    g2.draw( rect );

    g2.setPaint( Color.red );
    Ellipse2D oval = new Ellipse2D.Double( 400.0, 200.0, 50.0, 180.0 );
    g2.fill( oval );
}

```

Johan Eliasson

Fler prylar

- JOptionPane
- JScrollPane
- JPanel
- JSlider, JSpinner
- JLabel
- Listor, popupmenyer och träd
- Flera knappar
- ImageIcon
- Menyter, toolbars, etc
- Adaptor klasser

Johan Eliasson

JOptionPane

- Enkelt att göra olika "meddelandedialoger"
- Finns färdiga
- Kan modifiera och göra egna

Johan Eliasson

```
JOptionPane.showMessageDialog(  
    null, "Kolla", "Vad kul",  
    JOptionPane.ERROR_MESSAGE );  
  
x = JOptionPane.showConfirmDialog(  
    null, "Hello world" );  
  
s = JOptionPane.showInputDialog(  
    null, "Skriv nåt" );  
  
JOptionPane.showMessageDialog(  
    null, "Du skrev: " + s,  
    "Meddelande",  
    JOptionPane.INFORMATION_MESSAGE );
```



Johan Eliasson

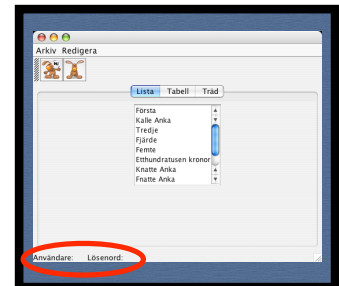
JPanel

- Används för att organisera andra objekt
- Har layout manager (default - FlowLayout)

Johan Eliasson

JLabel

Enkel text
Eller en bild
Eller båda



Johan Eliasson

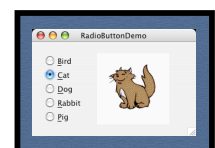
```
private JLabel statusLine = new JLabel( );
```

```
statusLine.setText( "Användare: " + username + "      Lösenord: " + password );
```

Johan Eliasson

Knappar

- Push buttons
- Check buttons
- Radio buttons
- Grupper av radio buttons



Johan Eliasson

```
JRadioButton catButton = new JRadioButton(catString);
catButton.setMnemonic(KeyEvent.VK_C);
catButton.setActionCommand(catString);
```

```
ButtonGroup group = new ButtonGroup();
group.add(birdButton);
group.add(catButton);
group.add(dogButton);
group.add(rabbitButton);
group.add(pigButton);
```

```
catButton.addActionListener(this);
```

```
public void actionPerformed(ActionEvent e) {
    picture.setIcon(createImageIcon("images/"
        + e.getActionCommand()
        + ".gif"));
}
```

Johan Eliasson

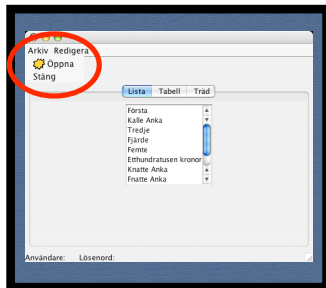
ImageIcon

- Enkelt sätt att få symboler på olika ställen
- Klarar JPEG, GIF och PNG bilder

Johan Eliasson

Menyer

- Menyrad
- Menyer
- Menyval



Johan Eliasson

```
JMenuBar menubar = new JMenuBar();
setJMenuBar( menubar );
```

```
JMenu file = new JMenu( "Arkiv" );
JMenu edit = new JMenu( "Redigera" );
```

```
menubar.add( file );
menubar.add( edit );
```

```
JMenuItem fileClose = new JMenuItem( "Stäng" );
```

```
fileClose.addActionListener( new ActionListener() {
    public void actionPerformed( ActionEvent ae )
    {
        // Stäng fönstret
        AddressBook.this.dispose();
    }
} );
```

```
file.add( fileClose );
```

Johan Eliasson

```
class OpenAction
    extends AbstractAction
{
    OpenAction( String name )
    {
        super( name, new ImageIcon( "bilder/middle.gif" ) );
    }

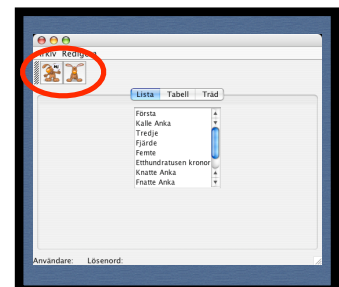
    public void actionPerformed( ActionEvent ae )
    {
        statusLine.setText( "Inte öppnas nåt nu inte!!" );
    }
}
```

```
file.add( new OpenAction( "Öppna" ) );
```

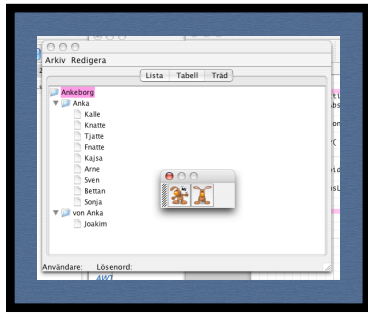
Johan Eliasson

Toolbars

- Ligger längst upp
- Flyter omkring



Johan Eliasson



Johan Eliasson

```
JToolBar tb = new JToolBar( );
tb.add( new HiAction( ) );
tb.add( new PlainAction( ) );
```

```
class PlainAction
    extends AbstractAction
{
    PlainAction( )
    {
        super( "plain", new ImageIcon( "bilder/rabbit.png" ) );
    }

    public void actionPerformed( ActionEvent ae )
    {
        statusLine.setText( "Jag vill ha en morot" );
    }
}
```

Johan Eliasson

Popup-menyer

- JPopupMenu
- JComboBox



Johan Eliasson

```
String[] petStrings = { "Bird", "Cat", "Dog", "Rabbit", "Pig" };
JComboBox petList = new JComboBox( petStrings );
petList.setSelectedIndex( 4 );
petList.addActionListener( this );
```

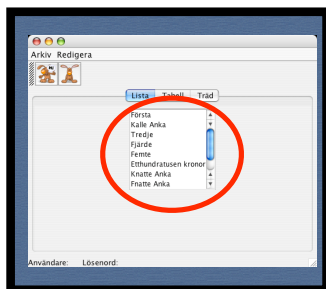
```
public void actionPerformed( ActionEvent e ) {
    JComboBox cb = (JComboBox)e.getSource();
    String petName = (String)cb.getSelectedItem();
    updateLabel( petName );
}
```

```
JComboBox patternList = new JComboBox( patternExamples );
patternList.setEditable( true );
patternList.addActionListener( this );
```

Johan Eliasson

Listor

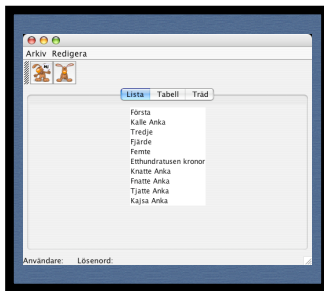
- JList



Johan Eliasson

```
jl = new JList( new Object[] {
    "Första",
    new Person( "Kalle", "Anka" ),
    "Tredje",
    "Fjärde",
    "Femte",
    "Etthundratusen kronor",
    new Person( "Knatte", "Anka" ),
    new Person( "Fnatte", "Anka" ),
    new Person( "Tjatte", "Anka" ),
    new Person( "Kajsa", "Anka" ),
} );
```

Johan Eliasson



Johan Eliasson

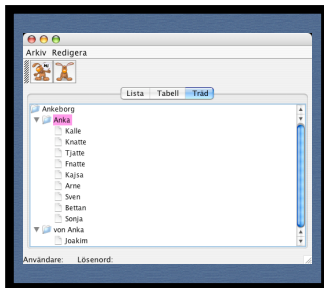
JScrollPane

- Implementerar själva scrollningen

```
JScrollPane sp = new JScrollPane( jl );
add( sp );
```

Johan Eliasson

Träd



Johan Eliasson

```
DefaultMutableTreeNode root = new DefaultMutableTreeNode( "Ankeborg" );
```

```
JTree tree = new JTree( root );
JScrollPane sp = new JScrollPane( tree );
```

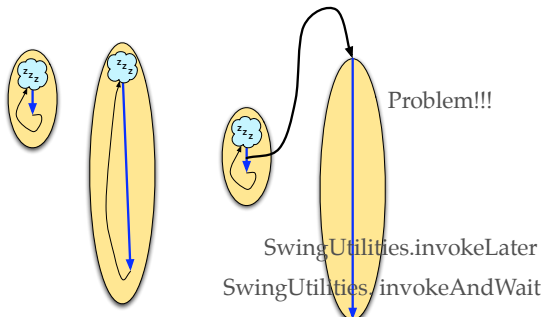
```
DefaultMutableTreeNode duck = new DefaultMutableTreeNode( "Anka" );
duck.add( new DefaultMutableTreeNode( "Kalle" ) );
duck.add( new DefaultMutableTreeNode( "Kotte" ) );
duck.add( new DefaultMutableTreeNode( "Tjatte" ) );
duck.add( new DefaultMutableTreeNode( "Fritte" ) );
duck.add( new DefaultMutableTreeNode( "Kajsa" ) );
duck.add( new DefaultMutableTreeNode( "Arne" ) );
duck.add( new DefaultMutableTreeNode( "Sven" ) );
duck.add( new DefaultMutableTreeNode( "Bettan" ) );
duck.add( new DefaultMutableTreeNode( "Sonja" ) );
```

```
root.add( duck );
```

```
DefaultMutableTreeNode vonduck = new DefaultMutableTreeNode( "von Anka" );
vonduck.add( new DefaultMutableTreeNode( "Joakim" ) );
root.add( vonduck );
```

Johan Eliasson

Händelser som tar tid



Johan Eliasson

XML

Introduktion

Johan Eliasson

Problemet

- Gammalt problem: arkivera/byta data
- Binärt
- Text
- Struktur

Johan Eliasson

Principer

- En strukturerad textfil
- Använder taggar för att märka ut informationen
- Informationen kan lätt flyttas mellan olika system
- Informationen kan lätt läsas i olika programspråk

Johan Eliasson

XML Exempel

ID	Namn
1	Kalle Anka
2	Kajsa Anka
3	Knatte Anka
4	Fnatte Anka
5	Tjatte Anka

Johan Eliasson

```
<?xml version="1.0" encoding="UTF-8"?>
<annikastuff>
  <values>
    <l1 d0="1" d1="2" d2="3" />
    <l2 d0="4" d1="5" />
    <l3 d0="6" d1="7" d2="8" />
    <px d0="11" d1="12" />
    <py d0="13" d1="14" />
  </values>
  <students>
    <student id="1" name="Kalle Anka" l1="0" l2="0" l3="0" px="0" py="0" />
    <student id="2" name="Kajsa Anka" l1="0" l2="0" l3="0" px="0" py="0" />
    <student id="3" name="Knatte Anka" l1="0" l2="0" l3="0" px="0" py="0" />
    <student id="4" name="Fnatte Anka" l1="0" l2="0" l3="0" px="0" py="0" />
    <student id="5" name="Tjatte Anka" l1="0" l2="0" l3="0" px="0" py="0" />
  </students>
</annikastuff>
```

Johan Eliasson

```
<?xml version="1.0" encoding="UTF-8"?>
<annikastuff>
  <values>
    <variable name="l1">
      <value value="1" />
      <value value="2" />
      <value value="3" />
    </variable>
    <variable name="l2">
      <value value="4" />
      <value value="5" />
    </variable>
    <variable name="l3">
      <value value="6" />
      <value value="7" />
      <value value="8" />
    </variable>
    <variable name="px">
      <value value="11" />
      <value value="12" />
    </variable>
    <variable name="py">
      <value value="13" />
      <value value="14" />
    </variable>
  </values>
  <students>
    <student id="1" l1="0" l2="0" l3="0" name="Kalle Anka" px="0" py="0" />
    <student id="2" l1="0" l2="0" l3="0" name="Kajsa Anka" px="0" py="0" />
    <student id="3" l1="0" l2="0" l3="0" name="Knatte Anka" px="0" py="0" />
    <student id="4" l1="0" l2="0" l3="0" name="Fnatte Anka" px="0" py="0" />
    <student id="5" l1="0" l2="0" l3="0" name="Tjatte Anka" px="0" py="0" />
  </students>
</annikastuff>
```

Johan Eliasson

XML och Java

- Java har naturligtvis stöd för detta
- SAX och DOM
- XSLT

Johan Eliasson

```

import java.io.FileOutputStream;
import javax.xml.parsers.*;
import javax.xml.transform.*;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import org.w3c.dom.*;

public class GenerateData
{
    ...

    public static void main( String[] argv )
        throws Exception
    {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        DOMImplementation domImpl = builder.getDOMImplementation();
        Document doc = domImpl.createDocument( null,
            "annikastuff", null );

        Element root = doc.getDocumentElement();

        root.appendChild( buildXMLValues( doc ) );
        root.appendChild( buildXMLStudents( doc ) );

        TransformerFactory tf = TransformerFactory.newInstance();
        Transformer trans = tf.newTransformer();
        Source src = new DOMSource( doc );
        FileOutputStream fos = new FileOutputStream( argv[0] );
        Result output = new StreamResult( fos );
        trans.transform( src, output );
    }
}

```

Johan Eliasson

```

private static Element buildValues( Document doc, String n, Integer... vals )
{
    Element val = doc.createElement( "variable" );
    Element tmp;

    val.setAttribute( "name", n );

    for( int counter = 0; counter < vals.length; counter++ ){
        tmp = doc.createElement( "value" );
        tmp.setAttribute( "value", vals[counter].toString() );
        val.appendChild( tmp );
    }
    return val;
}

private static Element buildXMLValues( Document doc )
{
    Element val = doc.createElement( "values" );

    val.appendChild( buildValues( doc, "l1", 1, 2, 3 ) );
    val.appendChild( buildValues( doc, "l2", 4, 5 ) );
    val.appendChild( buildValues( doc, "l3", 6, 7, 8 ) );
    val.appendChild( buildValues( doc, "px", 11, 12 ) );
    val.appendChild( buildValues( doc, "py", 13, 14 ) );
    return val;
}

```

Johan Eliasson

```

private static Element buildStudent( Document doc, String s, Integer id,
    Integer l1, Integer l2, Integer l3, Integer px, Integer py )
{
    Element val = doc.createElement( "student" );

    val.setAttribute( "name", s );
    val.setAttribute( "id", id.toString() );
    val.setAttribute( "l1", l1.toString() );
    val.setAttribute( "l2", l2.toString() );
    val.setAttribute( "l3", l3.toString() );
    val.setAttribute( "px", px.toString() );
    val.setAttribute( "py", py.toString() );

    return val;
}

private static Element buildXMLStudents( Document doc )
{
    Element val = doc.createElement( "students" );

    val.appendChild( buildStudent( doc, "Kalle Anka", 1, 0, 0, 0, 0, 0 ) );
    val.appendChild( buildStudent( doc, "Kajsa Anka", 2, 0, 0, 0, 0, 0 ) );
    val.appendChild( buildStudent( doc, "Knatte Anka", 3, 0, 0, 0, 0, 0 ) );
    val.appendChild( buildStudent( doc, "Fnatte Anka", 4, 0, 0, 0, 0, 0 ) );
    val.appendChild( buildStudent( doc, "Tjatte Anka", 5, 0, 0, 0, 0, 0 ) );

    return val;
}

```

Johan Eliasson

```

<?xml version="1.0" encoding="UTF-8"?>
<annikastuff>
  <values>
    <variable name="l1">
      <value value="1" />
      <value value="2" />
      <value value="3" />
    </variable>
    <variable name="l2">
      <value value="4" />
      <value value="5" />
    </variable>
    <variable name="l3">
      <value value="6" />
      <value value="7" />
      <value value="8" />
    </variable>
    <variable name="px">
      <value value="11" />
      <value value="12" />
    </variable>
    <variable name="py">
      <value value="13" />
      <value value="14" />
    </variable>
  </values>
  <students>
    <student id="1" l1="0" l2="0" l3="0" name="Kalle Anka" px="0" py="0" />
    <student id="2" l1="0" l2="0" l3="0" name="Kajsa Anka" px="0" py="0" />
    <student id="3" l1="0" l2="0" l3="0" name="Knatte Anka" px="0" py="0" />
    <student id="4" l1="0" l2="0" l3="0" name="Fnatte Anka" px="0" py="0" />
    <student id="5" l1="0" l2="0" l3="0" name="Tjatte Anka" px="0" py="0" />
  </students>
</annikastuff>

```

Johan Eliasson

```

import javax.xml.parsers.*;
import org.xml.sax.InputSource;
import org.w3c.dom.*;

public class ConsumeData
{
    ...

    public static void main( String[] args )
        throws Exception
    {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder parser = factory.newDocumentBuilder();
        Document document = parser.parse( new InputSource( args[0] ) );

        Element data = document.getDocumentElement();

        // Det ska bara vara en (struktur i felkoll)
        NodeList el = data.getElementsByTagName( "values" );
        displayVariables( (Element) el.item(0) );

        // Det ska bara vara en (struktur i felkoll)
        el = data.getElementsByTagName( "students" );
        displayStudents( (Element) el.item(0) );
    }
}

```

Johan Eliasson

```

private static void displayOneVariable( Element v )
{
    int nrOfValues;

    System.out.print( v.getAttribute( "name" ) + " = { " );

    NodeList vals = v.getElementsByTagName( "value" );

    nrOfValues = vals.getLength();
    for( int counter = 0; counter < nrOfValues; counter++ ){
        if( counter < nrOfValues-1 ){
            System.out.print( ( (Element) vals.item(counter) ).getAttribute( "value" ) + ", " );
        } else {
            System.out.print( ( (Element) vals.item(counter) ).getAttribute( "value" ) );
        }
    }
    System.out.println( " } " );
}

private static void displayVariables( Element v )
{
    NodeList vars = v.getElementsByTagName( "variable" );

    for( int counter = 0; counter < vars.getLength(); counter++ ){
        displayOneVariable( (Element) vars.item( counter ) );
    }
}

```

Johan Eliasson

```

private static void displayOneStudent( Element s )
{
    System.out.println( " Namn ; " + s.getAttribute( "name" ) );
    System.out.println( " ID ; " + s.getAttribute( "id" ) );
    System.out.println( " l1 ; " + s.getAttribute( "l1" ) );
    System.out.println( " l2 ; " + s.getAttribute( "l2" ) );
    System.out.println( " l3 ; " + s.getAttribute( "l3" ) );
    System.out.println( " px ; " + s.getAttribute( "px" ) );
    System.out.println( " py ; " + s.getAttribute( "py" ) );
}

private static void displayStudents( Element v )
{
    NodeList students = v.getElementsByTagName( "student" );

    for( int counter = 0; counter < students.getLength(); counter++ ){
        displayOneStudent( (Element) students.item( counter ) );
    }
}

```

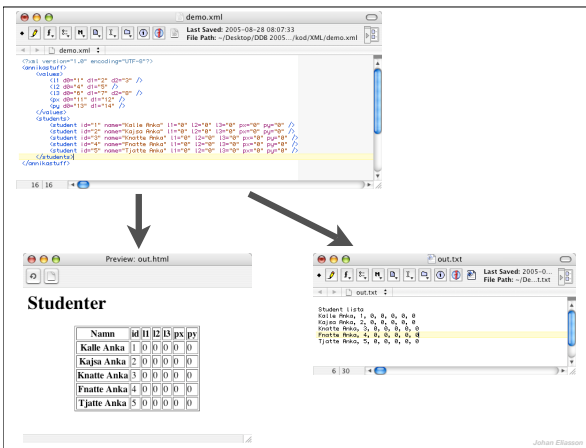
Johan Eliasson

```

> java ConsumeData bejsan
l1 = [1, 2, 3]
l2 = [4, 5]
l3 = [6, 7, 8]
px = [11, 12]
py = [13, 14]
Namn : Kalle Anka
ID : 1
l1 : 0
l2 : 0
l3 : 0
px : 0
py : 0
Namn : Kajsa Anka
ID : 2
l1 : 0
l2 : 0
l3 : 0
px : 0
py : 0
Namn : Knatte Anka
ID : 3
l1 : 0
l2 : 0
l3 : 0
px : 0
py : 0
Namn : Fnatte Anka
ID : 4
l1 : 0
l2 : 0
l3 : 0
px : 0
py : 0
Namn : Tjatte Anka
ID : 5
l1 : 0
l2 : 0
l3 : 0
px : 0
py : 0

```

Johan Eliasson



Johan Eliasson

```

import java.io.FileOutputStream;
import javax.xml.transform.*;
import javax.xml.transform.stream.*;

public class FormatStudentData
{
    public static void main( String[] args )
        throws Exception
    {
        TransformerFactory factory = TransformerFactory.newInstance();
        Transformer xslt = factory.newTransformer(
            new StreamSource( args[0] ) );
        StreamSource xmlsrc = new StreamSource( args[1] );
        StreamResult output = new StreamResult(
            new FileOutputStream( args[2] ) );
        xslt.transform( xmlsrc, output );
    }
}

```

```

> java FormatStudentData studenttext.xslt demo.xml out.txt
> java FormatStudentData studenthtml.xslt demo.xml out.html

```

Johan Eliasson

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
    version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:output method="html" />
    <xsl:template match="/">

        <html>
            <head>
                <title>Student lista</title>
            </head>
            <body>
                <h1>Studenter</h1>
                <xsl:apply-templates select="annikastuff/students" />
            </body>
        </html>
    </xsl:template>

    <xsl:template match="students">
        <table align="center" border="1">
            <tr><th>Namn</th><th>id</th><th>chp1</th><th>chp12</th><th>chp13</th><th>chp14</th><th>chp15</th><th>chp16</th><th>chp17</th><th>chp18</th><th>chp19</th><th>chp20</th></tr>
            <xsl:apply-templates select="student" />
        </table>
    </xsl:template>

    <xsl:template match="student">
        <tr>
            <th><xsl:value-of select="#name" /></th>
            <td><xsl:value-of select="#id" /></td>
            <td><xsl:value-of select="#l1" /></td>
            <td><xsl:value-of select="#l2" /></td>
            <td><xsl:value-of select="#l3" /></td>
            <td><xsl:value-of select="#px" /></td>
            <td><xsl:value-of select="#py" /></td>
        </tr>
    </xsl:template>
</xsl:stylesheet>

```

Johan Eliasson

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
    version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:output method="text" />
    <xsl:template match="/">
        Student lista
        <xsl:apply-templates select="annikastuff/students" />
    </xsl:template>

    <xsl:template match="students">
        <xsl:preserve-space elements="student"/>
        <xsl:for-each select="student">
            <xsl:value-of select="#name" />, <xsl:value-of select="#id" />,
            <xsl:value-of select="#l1" />, <xsl:value-of select="#l2" />,
            <xsl:value-of select="#l3" />, <xsl:value-of select="#px" />,
            <xsl:value-of select="#py" />
        </xsl:for-each>
    </xsl:template>
</xsl:stylesheet>

```

Johan Eliasson