Recitation No. 10: SWT GUI Package

Software 1

The GUI Development Process

- **GUI**: Graphical User Interface
  - User Interface Engineer
  - Graphic Designer
  - GUI Programmer

**GUI Application**

- When implementing a GUI application one should specify:
  - the GUI elements
  - the 2D arrangement of the GUI elements
  - the behavior of the GUI elements
- Java GUI libraries:
  - AWT (Abstract Windowing Toolkit)
  - Swing
  - SWT (Standard Widget Toolkit)

**Model-View Separation**

- Separate between the application logic (model) part and the GUI (view) part.
- Ensures that view changes have no effect on the basic model
- Enables us to maintain one model for several different views

**Example: Address Book**

- Address Book Application
  - AddressBook class
  - GIAddressBookViewer class
- The Model
  - <class>AddressBook
    - void add(Contact c);
    - Contact get(String name);
    - void delete(String name);
    - void modify(Contact c);
    - Contact search(String prefix);
    - Iterator<Contact> getContacts();
    - int getCount();
    - Iterator<Contact> search(String prefix);
    - void save(String filename);
    - void load(String filename);
- The View
  - <class>Contact
    - String name;
    - String email;
    - String telephone;
    - String street;
    - String zipCode;
    - String country;
    - String address
      - has-a Address
  - <class>Address
    - String street;
    - String zipCode;
    - String country;

The View

The class diagram:

The implementation:
- based on the SWT GUI library

 SWT

 Online Documentation:
- SWT HomePage: http://www.eclipse.org/swt/
- JavaDoc
- Snippets
- Getting Started with Eclipse and the SWT: http://www.cs.umanitoba.ca/~eclipse/

Widgets

Layouts

A Layout controls the position and size of Control widgets in a Composite.
GridLayout

- Lays out the Control widgets in a grid.

Each column is as wide as Wide Button 2

GridLayout (cont.)

GridData:
- Use GridData objects to configure the Control widgets in a GridLayout.
- Use the setLayoutData() to set a GridData object into a Control, e.g.
  label.setLayoutData(new GridData(…));
- Do not reuse GridData objects

GridLayout (cont.)

GridData Configuration Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gridDataMustHaveSize</td>
<td>SWT.DEFAULT</td>
<td>If true, the width/height of the widget will be as large as possible to fit the remaining space.</td>
</tr>
<tr>
<td>gridDataMustHaveSize</td>
<td>SWT.DEFAULT</td>
<td>A minimum width/height for the widget.</td>
</tr>
<tr>
<td>horizontalIndent</td>
<td>0</td>
<td>The number of indentation pixels along the left side of the cell.</td>
</tr>
<tr>
<td>horizontalSpan</td>
<td>1</td>
<td>The number of column/row cells that the widget will take up.</td>
</tr>
<tr>
<td>horizontalAlignment</td>
<td>GridData.BEFORE</td>
<td>How controls will be positioned horizontally/vertically within a cell.</td>
</tr>
<tr>
<td>verticalAlignment</td>
<td>GridData.CENTER</td>
<td>How controls will be positioned horizontally/vertically within a cell.</td>
</tr>
<tr>
<td>marginHeight</td>
<td>0</td>
<td>The size of the horizontal/vertical space between the grid cells.</td>
</tr>
<tr>
<td>marginWidth</td>
<td>0</td>
<td>The size of the horizontal/vertical margins of the layout.</td>
</tr>
</tbody>
</table>

FormLayout

- A very flexible layout

FormLayouts

- A very flexible layout
- FormLayout Configuration Properties:

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>marginHeight</td>
<td>0</td>
<td>the margin width/height</td>
</tr>
<tr>
<td>marginWidth</td>
<td>0</td>
<td>the margin width/height</td>
</tr>
<tr>
<td>spacing</td>
<td>0</td>
<td>the number of pixels between the edge of one control and the edge of its neighbouring control.</td>
</tr>
</tbody>
</table>
FormLayouts (cont.)

- Use FormData objects to configure the Control widgets in a FormLayout.
- Use the setLayoutData() to set a FormData object into a Control widget.
- A FormData object has a FormAttachment object for each edge of the Control.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>width/height</td>
<td>the desired width/height in pixels.</td>
</tr>
<tr>
<td>top/bottom/left/right</td>
<td>specifies the position of the control attachment.</td>
</tr>
</tbody>
</table>

FormLayouts (cont.)

- A FormAttachment defines where to attach the side of a Control by using the equation: \( y = ax + b \).

\[ y = \frac{\text{numerator}}{\text{denominator}} \times \text{offset} + \text{control's width/height} \]

FormLayouts (cont.)

**Main FormAttachment Constructors:**

- public FormAttachment(Control control)
- public FormAttachment(Control control, int offset)
- public FormAttachment(int numerator)
- public FormAttachment(int numerator, int offset)

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>control</td>
<td>Parent Control</td>
</tr>
<tr>
<td>numerator</td>
<td>100</td>
</tr>
<tr>
<td>denominator</td>
<td>100</td>
</tr>
<tr>
<td>offset</td>
<td>0</td>
</tr>
</tbody>
</table>