



# Object-Oriented Programming with Java

Recitation No. 11:  
SWT GUI Package

# The GUI Development Process

- **GUI: Graphical User Interface**

User Interface Engineer



Graphic Designer



GUI Programmer

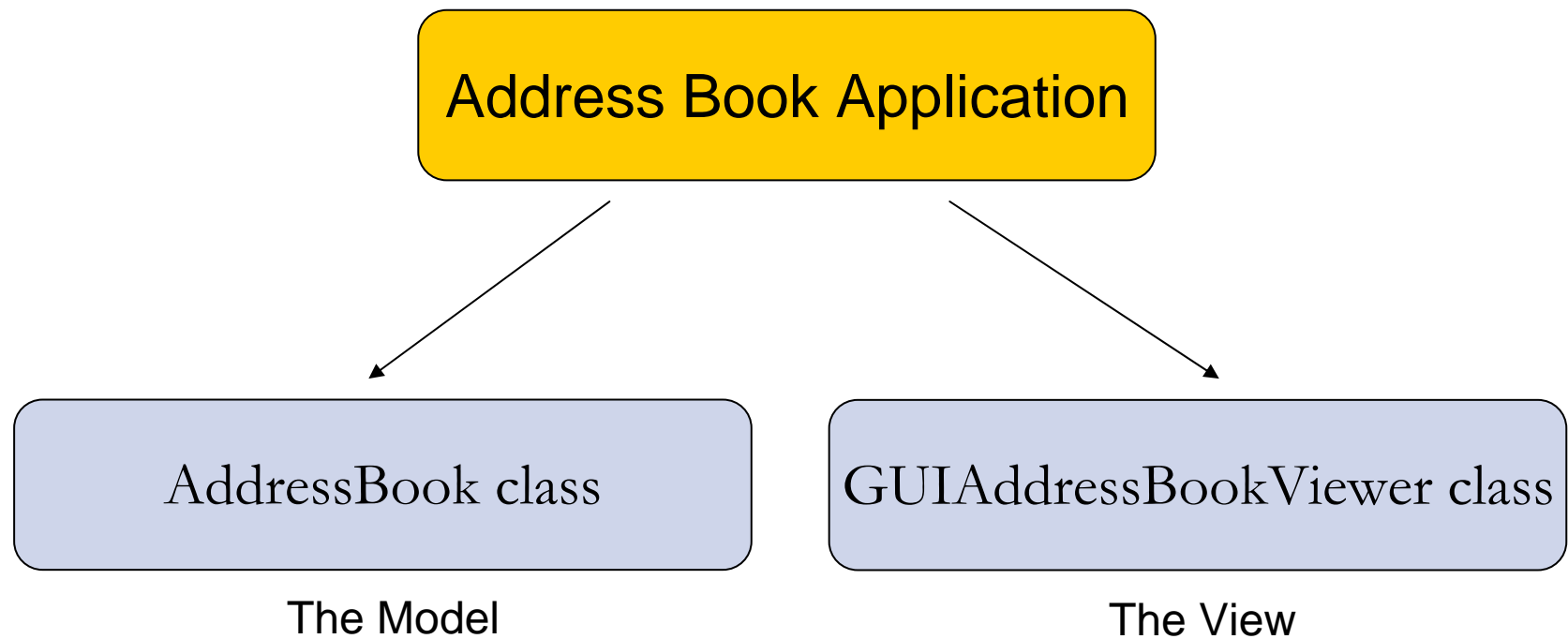
# GUI Application

- 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 (**A**bstract **W**indowing **T**oolkit)
  - Swing
  - SWT (**S**tandard **W**idget **T**oolkit)

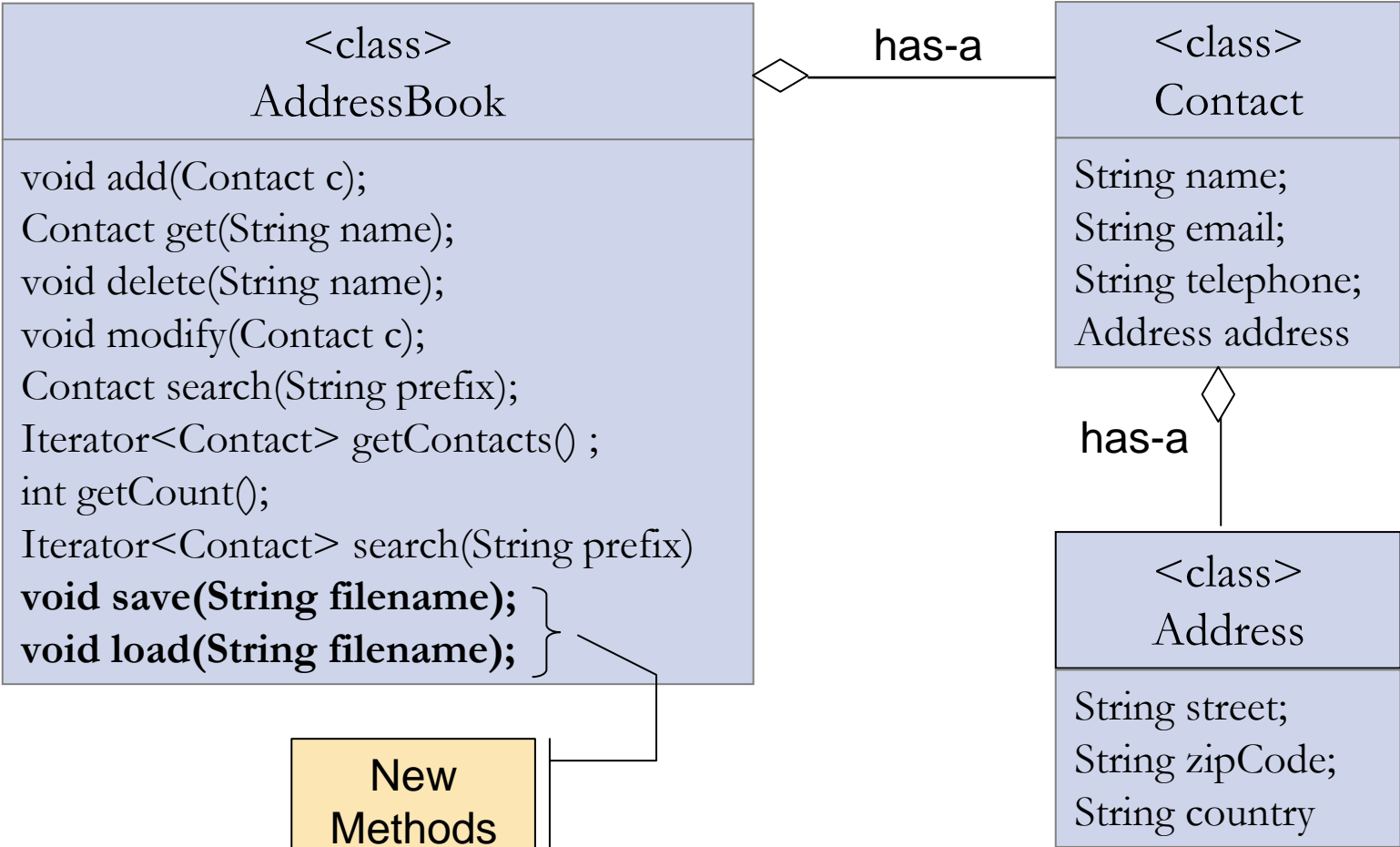
# Model-View Separation

- 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



# The Model



# The View

New Address Book Ctrl+N  
Open Ctrl+O  
Save Ctrl+S

New Contact Ctrl+M  
Edit  
Delete

The screenshot shows a window titled "Address Book" with a menu bar containing "File", "Contacts", and "Help". The main area displays a table of contacts with columns for Name, Email, and Phone Number. The contact "Smith, John" is selected. Below the table is a form for editing the selected contact's details.

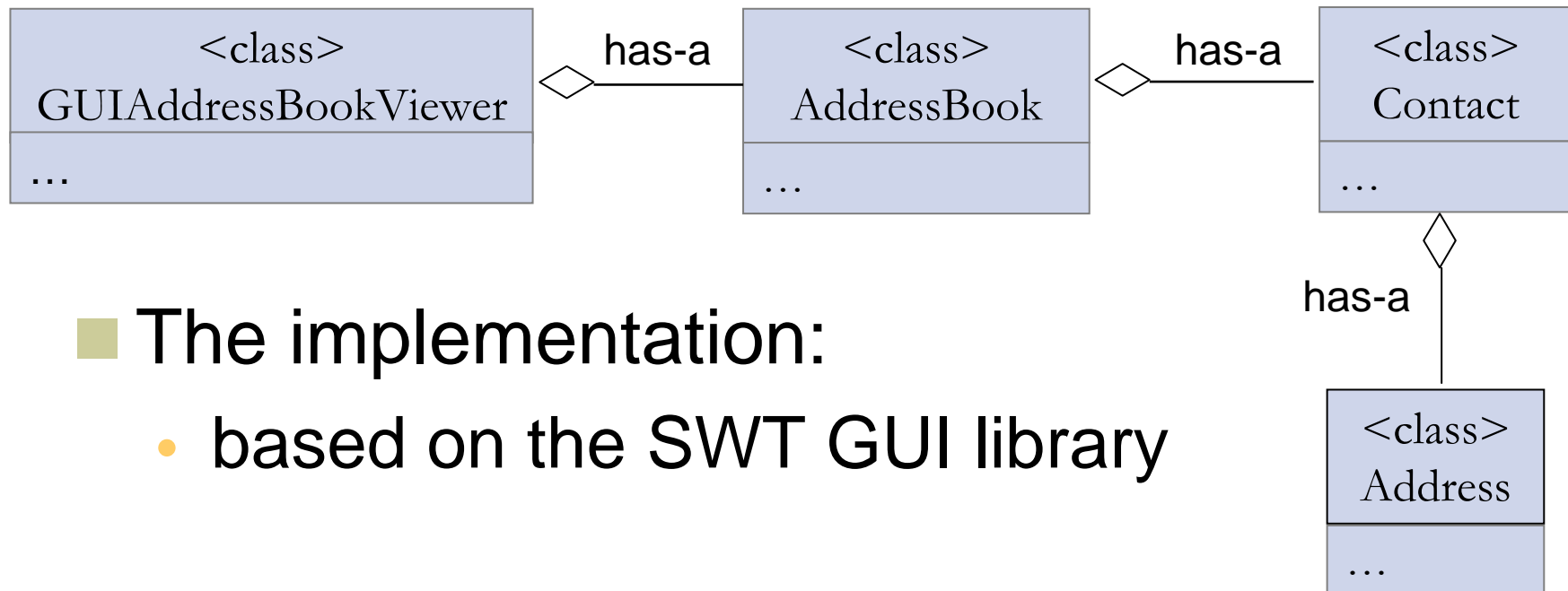
Name	Email	Phone Number
Altman, David	david@gmail.com	
Altman, Rebecca	rebecca@gmail.com	03-9414324
<b>Smith, John</b>	<b>smith@gmail.com</b>	<b>03-6404324</b>
Stein, Rita	rita@gmail.com	03-5524324

Name	Smith, John
Email	smith@gmail.com
Phone	03-6404324
Street Address	23 Laskov St.
City	Tel-Aviv
Zip Code	56743
Country	Israel

# The View

## ■ The class diagram:



## ■ The implementation:

- based on the SWT GUI library



# SWT

## ■ Online Documentation:

- SWT HomePage:

<http://dev.eclipse.org/viewcvs/index.cgi/%7Echeckout%7E/platform-swt-home/dev.html>

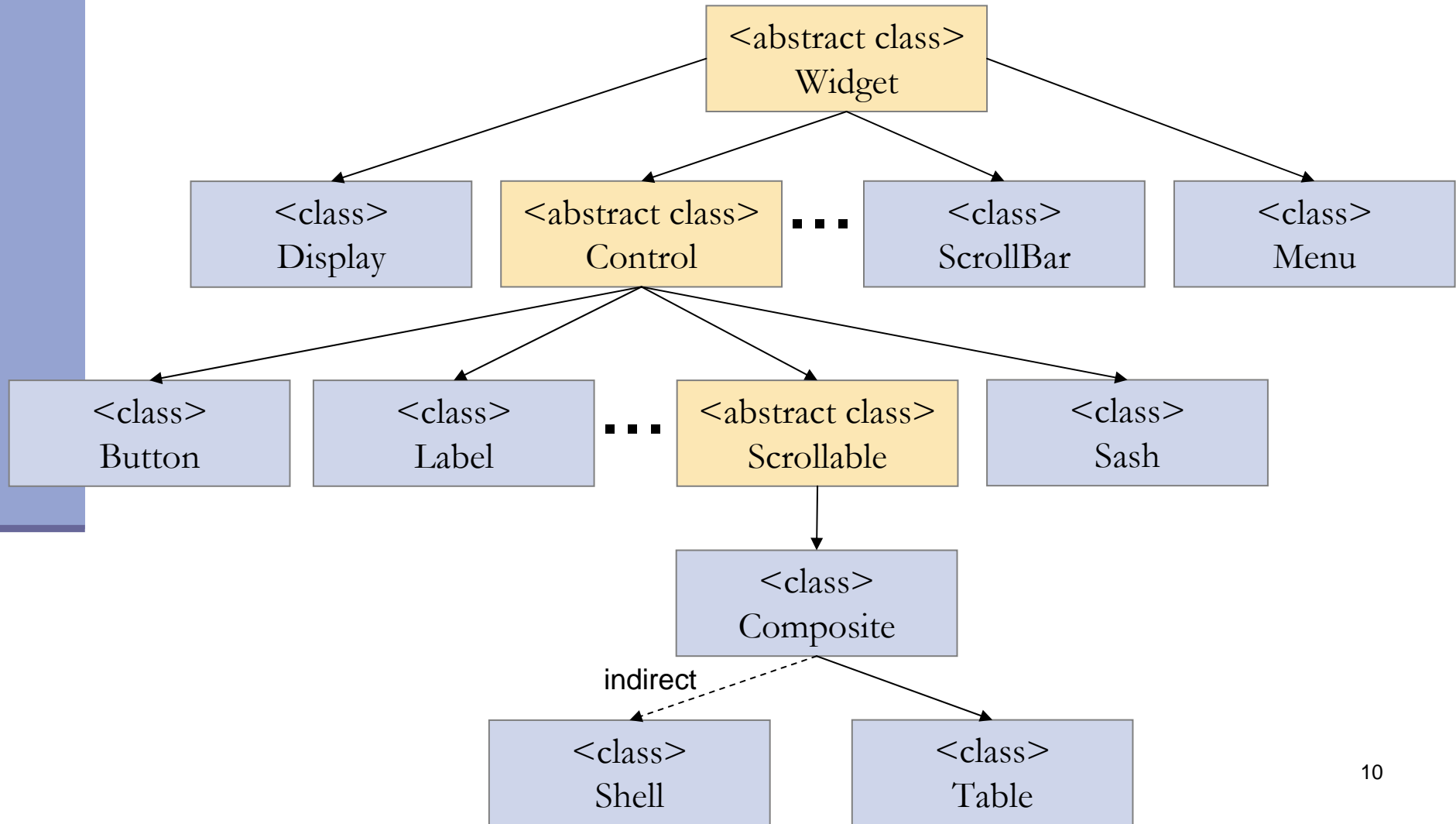
- JavaDoc

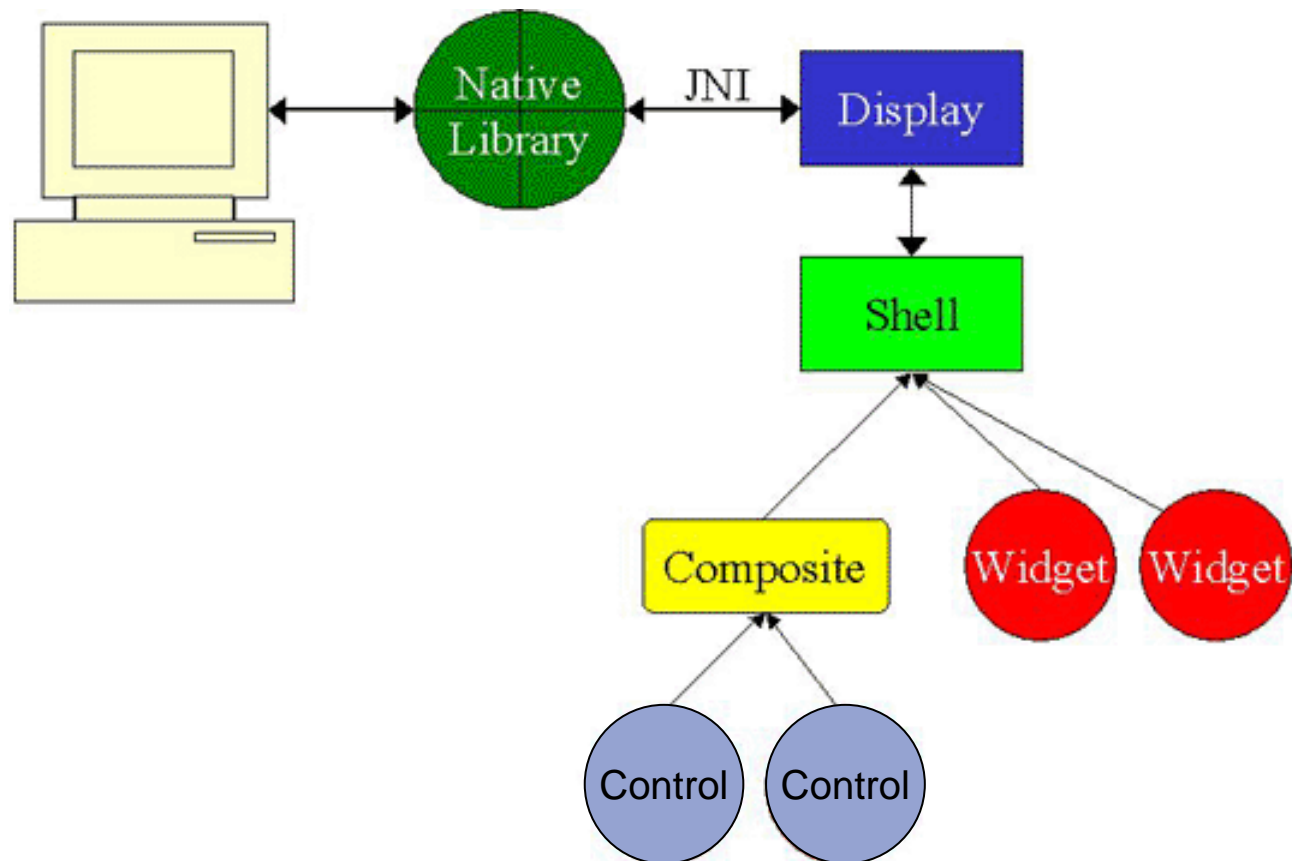
- Snippets

- Getting Started with Eclipse and the SWT:

<http://www.cs.umanitoba.ca/~eclipse/>

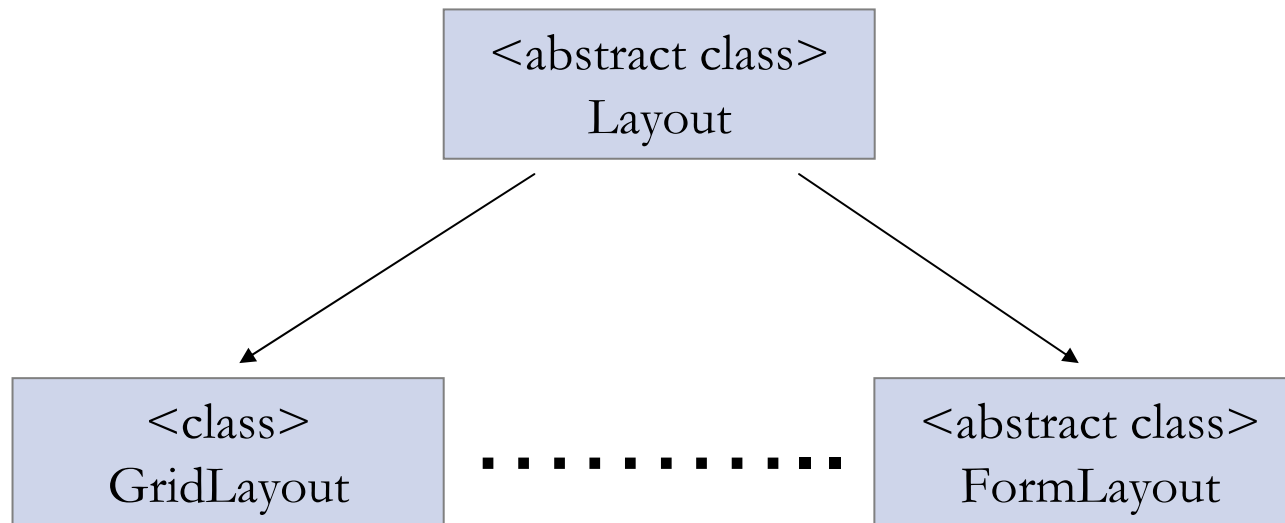
# Widgets





# Layouts

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



# GridLayout

- Lays out the Control widgets in a grid.
- GridLayout Configuration fields:

Field	Default	Description
horizontalSpacing verticalSpacing	5	Horizontal/vertical space between the grid cells
marginHeight marginWidth	5	The size of the horizontal/vertical margins of the layout
numColumns	1	Number of columns
makeColumnsEqualWidth	false	If true, all columns will have the same size

# 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:

Field	Default	Description
grabExcessHorizontalSpace grabExcessVerticalSpace	false	If true, the width/length of the widget will be as large as possible to fit the remaining space.
heightHint widthHint	SWT.DEFAULT (no minimum)	A minimum width/height for the widget.
horizontalSpan verticalSpan	1	the number of column/row cells that the widget will take up.
horizontalIndent	0	the number of indentation pixels along the left side of the cell.
horizontalAlignment verticalAlignment	GridData.BEGINNING GridData.CENTER	how controls will be positioned horizontally/vertically within a cell.

# FormLayouts

- A very flexible layout
- FormLayout Configuration Properties:

Field	Default	Description
marginHeight marginWidth	0	the margin width/height
spacing	0	the number of pixels between the edge of one control and the edge of its neighbouring control.



## 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.

Field	Description
width/height	the desired width/height in pixels.
top/bottom/left/right	Specifies the position of the control attachment.

# FormLayouts (cont.)

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

A fraction defined by:  
-**numerator**  
-**denominator**

an **offset**, in pixels

the width/height of a `Control` to which the control side is attached (**control**).

# 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)`

Field	Default
control	Parent Composite
numerator	100
denominator	100
offset	0

$$y = \frac{\textit{numerator}}{\textit{denominnator}} \bullet x + \textit{offet}$$

$x$  – control's width/height