CISC 3120
Design & Implementation of Software Applications I

Lecture #7 – More Input and Output

M. Meyer

Course Page:
http://www.sci.brooklyn.cuny.edu/~meyer/CISC3120
Content

- Text Fields Revisited
- Layout
- Panels
- Canvas
- Scrollbars
TextFields Revisited

- A text field allows for input/display of a single line of text.
- TextFields can generate events: example contents of the text field change.
- The methods `getText` and `setText` of the `TextField` class retrieve/store the value of the text field.
  - The contents of the text field is viewed as a String.
  - Non-string values (such as integer) must be converted to/from String when storing/retrieving the contents of the field.
A Scrollbar is an interface component that allows the user to easily move within a specified range of values.

The scrollbar may be manipulated:
- via the arrows at either end
- By dragging the 'thumb' (the little box in the scrollbar's gutter)

At any point the value of the scrollbar may be retrieved/modified via the getValue, setValue methods.
Listening to a Scrollbar

- As with buttons, the applet must register as a listener to the scroll bar.
- In the case of the scroll bar, the event generated is known as an adjustment event and the applet registers as an AdjustmentListener.
- The event-handling method for scrollbar events is adjustmentValueChanged.
- If multiple scrollbar's are present, the getSource method can be used, just as in actionPerformed.
Applet_07
Applets possess a default way in which components added to the applet are positioned
- Left-to-right/top-to-bottom

This behavior can be modified by specifying an alternative layout manager object for the applet.
- The layout manager object is responsible for the positioning of the components on the applet's surface
- Each layout manager (and there are several to choose from) provides a different way of positioning the components
- The default layout manager for an applet is a FlowLayout object
Border Layout

- The BorderLayoutManager class provides a layout consisting of five (5) regions
  - North - the upper portion of the surface
  - South - the lower portion of the surface
  - West - the left side of the surface
  - East - the right side of the surface
  - Center - the middle portion of the surface

- A slightly different form of the add method is used when working with BorderLayout
  - A 2\textsuperscript{nd} parameter is introduced specifying the desired region:
    - add(button1, BorderLayout.NORTH);
  - BorderLayout class provides constants for various regions
Border Layout

```
+----------+---------+----------+
| NORTH    | CENTER  | EAST     |
| WEST     |         |          |
|          |         | SOUTH    |
+----------+---------+----------+
```
Understanding Borders

- The sizing of the component when using BorderLayout takes some getting used to:
  - North and South components are stretched for the entire width of the applet's surface
  - East and West components are stretched for the entire height between North and South
  - Center is stretched to fill in the entire middle

- Note that only 5 components can be placed when using BorderLayout
**GridLayout**

- The GridLayout class provides a layout consisting of a rectangular grid of regions
  - The number of rows/columns may be specified in the constructor.
  - Adding components fills the grid from left-to-right, top-to-bottom, according to the number of rows/columns.
  - Blank panels (discussed later) can be used to leave a grid position empty.
GridLayout
Other Layout Managers

- **Flow Layout**
  - This is the default L/R T/B managers

- **Box Layout**
  - Allows you to stack your components on top of one another or from side to side.

- **Card Layout**
  - Useful when you want to display panels or containers one at a time: you assign an order to panels and use commands like first, next and last to cycle through them.

- **GridBag Layout**
  - Extension of the Grid Layout, and lets you manipulate the cells and positioning to a far greater level. Very powerful, and also quite complex.
BoxLayout

![BoxLayoutDemo](image1)

![Horizontal Box Te...](image2)
CardLayout

CardLayout Demo

TextField on Card 2

First    Next    Previous    Last
GridBagLayout
Applet_08
Panels

- An applet can use panels to further subdivide a applet's real estate, allows for better layout control.
- A panel is an interface component whose sole use is to act as a container for other components.
- Panels provide the ability to group components
  - Placing a Panel in a region of a BorderLayout allows multiple components to be positioned in that region (by adding them to the Panel)
- Like Applets (in fact an Applet is a form of Panel), the default layout for Panel is FlowLayout
  - This default can be overridden using Panel's setLayout method
Applet_09
Interface Components and Graphics

- An applet's surface can typically contain
  - Interface components, such as Button
    - These are placed on the applet's surface by the layout manager
  - Graphics, such as lines, rectangles, circles, drawn text
    - These are drawn by the paint method
  - The above two placements are performed independently of each other
    - This leads to graphics and components overlapping and blocking each other on the surface
The Canvas class provides a dedicated area for graphics to be drawn
  ◦ In fact, that is all a Canvas does: provide a drawing surface
  ◦ You can think of this as being similar to the Graphics object (g) that we have been passing into the paint method.

The Canvas can be laid out like any other component
  ◦ Once created, it's simply added to the applet (or panel)

Since it is its own component (with its own surface), no other components can overlap to block the drawn graphics on the canvas surface
SubClasSsing the Canvas Class

- The Canvas class provides drawing support, but what is to be drawn differs.
- Your Canvas class must be subclassed (extended) from the Canvas superclass.
- Your Canvas subclass must have:
  - A constructor to perform initialization
    - Canvas subclasses typically receive information from the applet in the form of parameters to its constructor
  - A paint method
    - It is here that the applet-specific drawing is performed
Applet_10
You can do this!