CSC 308 Lecture Notes Week 8
GUI Prototyping in Java Swing

I. Administration

    A. Milestones 7, 8, and 9 -- see the handouts.

    B. A very good Swing reading resource is the online Java Swing Tutorial, at

            `download.oracle.com/javase/tutorial/uiswing/index.html`

    C. Similarly good tutorial-style resources are available for Java FX at http://docs.oracle.com/javafx/

    D. These notes are based on Swing, with the Swing concepts readily transferable to Java FX.

II. Prototyping in Java Swing.

    A. Java provides a standard library of common user interface elements.

    B. Using the Java Swing tutorial, and the example code cited below, you can build a reasonably rapid prototype of the user interface for your 308 project.

    C. In upcoming notes, we'll discuss further details of how prototyping fits into the overall software engineering process.

    D. These notes are intended to get you started with concrete GUI prototyping.

    E. A GUI prototype differs from a full software implementation in that it focuses on just the GUI itself, without any underlying computation being performed.

        1. For example instead of a command like 'File->Save' actually saving a file, it just appears on a pull-down menu, and prints a message like "File->Save" selected.

        2. A command that brings up a data-entry dialog causes the dialog to appear on the screen, but pressing the 'OK' button in the dialog does not cause data to be stored in some collection.

        3. In this way, a GUI prototype is like a "store front", that shows how the interface appears, but does not do any real work.

III. Swing packages and classes.

    A. In 308, we will be doing prototyping using what's called the "Swing" library of Java GUI tools.

    B. The Swing library is rooted in the package named `javax.swing`, which contains many classes and sub-packages for building GUIs.

    C. Key Swing classes include the following:
- `Box` -- *a simple way to layout GUI components.*
- `ButtonGroup` -- *for grouping buttons, particular radio buttons.*
- `JButton` -- *a standard command button; used all over the place.*
- `JCheckBox` -- *a typical on/off check box.*
- `JCheckBoxMenuItem` -- *a menu itme with a check box next to it.*
- `JColorChooser` -- *a standard-looking color selection dialog.*
- `JComboBox` -- *a pulldown that allows typing too.*
- `JComponent` -- *the top-level of the Swing component hierarchy.*
- `JDialog` -- *a handy pop-up dialog.*
- `JEditorPane` -- *a way to view and edit text, in particular HTML.*
- `JFileChooser` -- *a standard-looking file chooser.*
- `JFrame` -- *the outermost container for a GUI window.*
- `JLabel` -- *a simple piece of text within a GUI.*
- `JLayeredPane` -- *a way to layer GUI frames in a 3D stack.*
- `JList` -- *a list of GUI components, typically with a scroll bar.*
- `JMenu` -- *a pulldown or pop-up menu.*

- `JMenuBar` -- *a standard menubar, typically at the top of a JFrame.*
- `JMenuItem` -- *an item in a JMenu.*
- `JOptionPane` -- *a parent class for a set of standard option dialogs.*
- `JPanel` -- *Typically the inner-wrapper of a JFrame, for managing GUI layout.*
- `JPasswordField` -- *a way to enter passwords without echoing.*
- `JProgressBar` -- *a typical-looking "throbber"*
- `JRadioButton` -- *a typical-looking radio button*
- `JScrollBar` -- *horizontal or vertical scrollbar, typically in a JScrollPane.*
- `JSeparator` -- *spacing in a menu.*
- `JSlider` -- *typical-looking slider, typically for numeric input.*
- `JTabbedPane` -- *tabbing pane for organizing things like preferences.*
- `JTable` -- *a two-dimensional table, with many display options.*
- `JTextArea` -- *a simple, unformatted multi-line text area.*
- `JTextField` -- *a single-line text field.*
- `JToggleButton` -- *an on/off button.*
- `JToolBar` -- *a container for buttons that select other tools.*
- `JToolTip` -- *roll-over help for tool buttons or menu items.*
- `JTree` -- *a hierarchical tree display, in a Windows Explorer style.*

D. Key Swing subpackages are:
- `javax.swing.colorchooser` -- *color chooser support classes*
- `javax.swing.event` -- *low-level classes representing GUI events*
- `javax.swing.filechooser` -- *file chooser support classes*
- `javax.swing.table` -- *JTable support classes*
- `javax.swing.text` -- *text display and editing support classes*
- `javax.swing.text.html` -- *HTML support (there is also XML support)*
- `javax.swing.text.html.parser` -- *low-level HTML support*
- `javax.swing.tree` -- *JTree support classes*
- `javax.swing.undo` -- *simple undo/redo support*

IV. There is also a companion package **`java.awt`** for lower-level GUI support.
  A. "awt" stands for the "abstract windowing tools".
  B. The classes and interfaces include the following:
- `Color` -- *low-level color support*
- `Component` -- *THE most generic GUI component, parent of JComponent*
- `Container` -- *THE most generic GUI container, parent of JFrame*
- `Event` -- *all the details of a GUI event*
- `Graphics2D` -- *the way to draw graphic shapes, e.g., lines, circles, etc.*
- `GridLayout` -- *a nasty way to do 2D layout (I like Boxes much better)*
- `Image` -- *a GIF or JPEG image*
- `java.awt.`*ActionListener -- the way buttons and menu items list for events*

V. Designing GUIs with swing components.

  A. The list of the Java swing components given above are those that you are most likely to use in the GUI interfaces for your CSC 308 and 309 projects.

  B. Figure 1 shows an annotated version of a typical menubar and its menus, indicating which swing components are used for which pieces.

  C. Figure 2 shows an annotated version of a typical editing dialog indicating which swing components are used for which pieces.

  D. Figure 3 shows an annotated version of the editing dialog showing how components are laid out using Swing `Boxes`; layout can also be done using `GradBags` and other forms of layout managers, , but I find these much more tedious than simple `Boxes`.

VI. The milestone 8 prototype example illustrate the use of Swing components to build an initial GUI prototype for
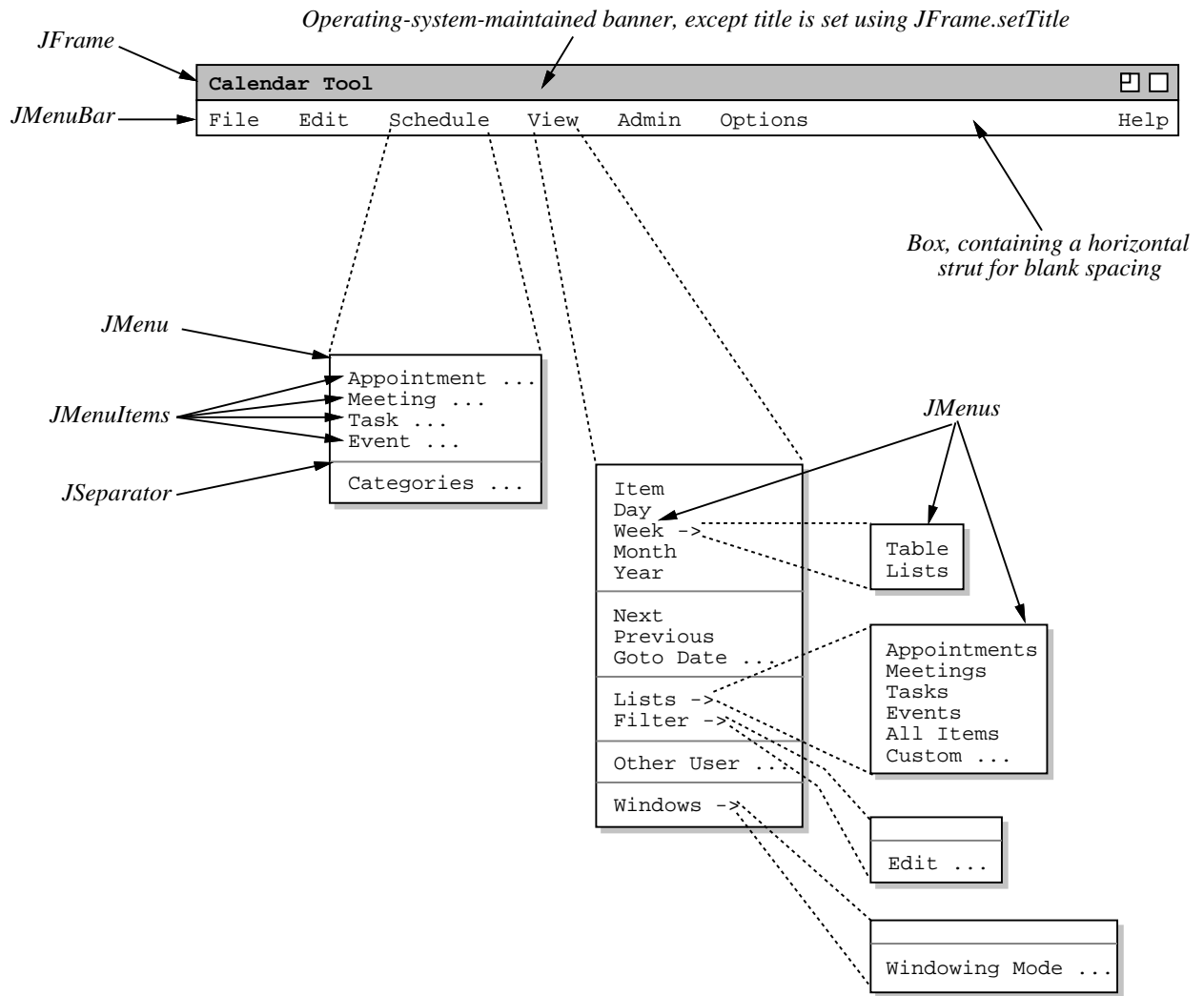
JFrame

Operating-system-maintained banner, except title is set using JFrame.setTitle

Calendar Tool

JMenuBar

File   Edit   Schedule   View   Admin   Options   Help

Box, containing a horizontal strut for blank spacing

JMenu

JMenuItems

Appointment ...
Meeting ...
Task ...
Event ...

JSeparator

Categories ...

JMenus

Item
Day
Week ->
Month
Year

Next
Previous
Goto Date ...

Lists ->
Filter ->

Other User ...

Windows ->

Table
Lists

Appointments
Meetings
Tasks
Events
All Items
Custom ...

Edit ...

Windowing Mode ...

Figure 1: Annotated menus showing swing components used.

JFrame

Schedule an Appointment   □ 凸

JLabels

Title:

Date: Start Time:

End Date: Duration: hr   min

JTextFields

JTextField (disabled)

Recurring? ☐ Interval: weekly ▼   S M T W Th F S
                                   ☐ ☐ ☐ ☐ ☐ ☐ ☐

JCheckBoxes (disabled)

JCheckBox

Category: ▼   Security: public ▼

JComboBox

Location: Priority: must ▼

Remind? ☐ 15 minutes before ▼ on screen ▼

Details:

JTextArea

△
▽

JScrollBar in a JScrollPane

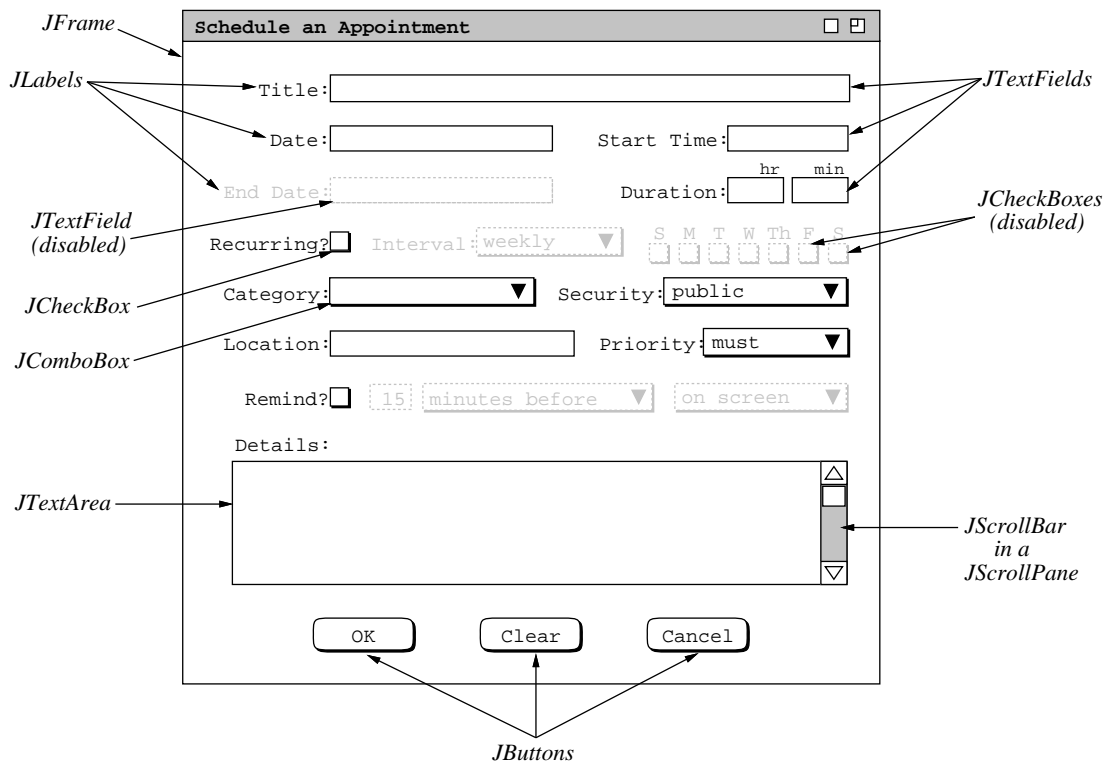[OK]   [Clear]   [Cancel]

JButtons

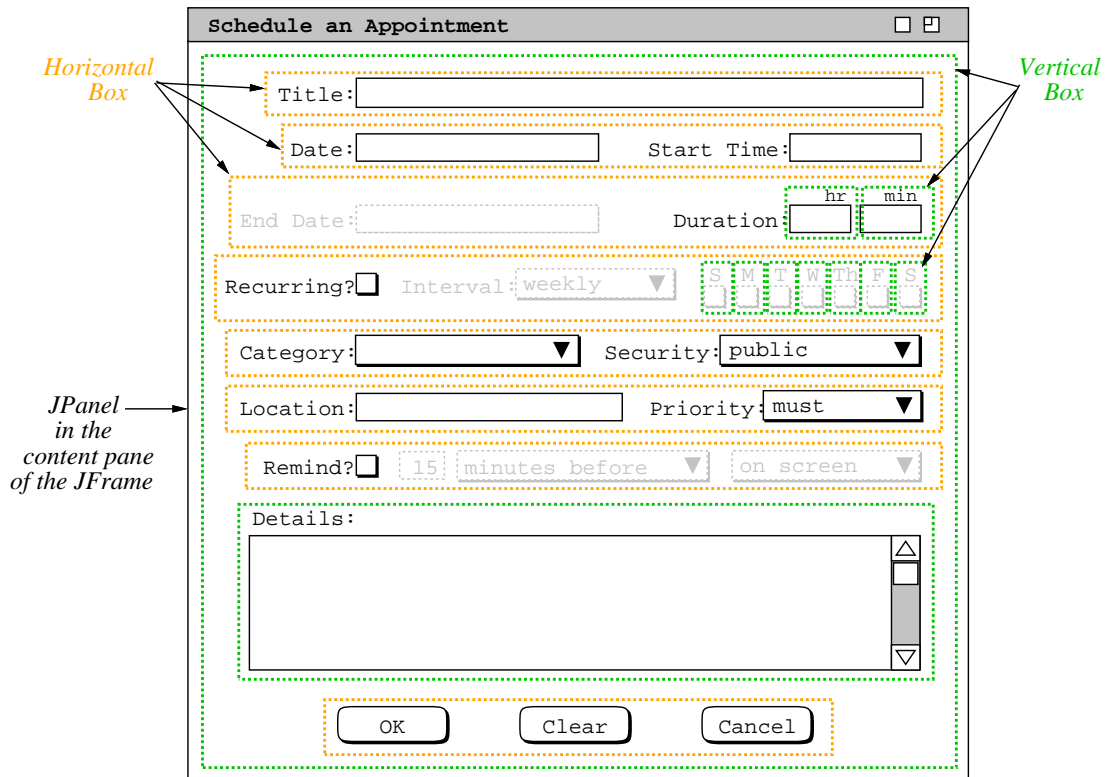**Figure 2:** Annotated dialog showing swing components used.

Figure 3: Annotated dialog showing layout using boxes.

the Calendar Tool.

    A. This example is comparable to what you need to build for your project's Milestone 8.

    B. The code and supporting material is online at `classes/308/examples/milestone8/prototype/`.

VII. We'll walk through the code and look at some highlights during lecture.