CSE 1325 OBJECT-ORIENTED PROGRAMMING

Exam #2 –=# HANDOUT REV 1 #=– Spring 2023

Note: "extends" in this document includes multi-generational inheritance: Superclass members are also available to the subclass. Square brackets [ ] denote optional parameters via overloaded members.

Superclasses

Component

Rectangle getBounds()

Gets the bounds of this component in the form of a Rectangle object.

void repaint()

Repaints this component at the earliest opportunity.

Container extends Component

Component add(Component comp, [int index])

Appends the specified component at the given position or end of the container.

void add(Component comp, Object constraints, [int index])

Adds the specified component with constraints at the given position or end of the container.

void setLayout(LayoutManager manager)

Sets the LayoutManager.

JComponent extends Container

void paintComponent(Graphics g)

Calls the UI delegate's paint method, if the UI delegate is non-null.

void setBackground(Color bg)

Sets the background color of this component.

void setBorder(Border border)

Sets the border of this component.

void setEnabled(boolean enabled)

Sets whether or not this component is enabled.

void setForeground(Color fg)

Sets the foreground color of this component.

void setToolTipText(String text)

Registers the text to display in a tool tip.

void setVisible(boolean aFlag)

Makes the component visible or invisible.

Window extends Container

void setSize(int width, int height)

Layout Managers

BorderLayout

BorderLayout([int hgap, int vgap])

Constraints: PAGE_START, PAGE_END, LINE_START, LINE_END, CENTER

GridBagConstraints

GridBagConstraints()

int anchor

Determines where, within the display area, to place the component: PAGE_START, PAGE_END, LINE_START, LINE_END, FIRST_LINE_START, FIRST_LINE_END, LAST_LINE_START, LAST_LINE_END

int fill

Determines how to resize the component: NONE, HORIZONTAL, VERTICAL, BOTH

int gridheight, gridwidth

Specifies the number of cells in a row and column for the component's display area. REMAINDER means use all remaining cells. RELATIVE means use all remaining cells except one.

Insets insets

Specifies the external padding of the component, the minimum amount of space between the component and the edges of its display area. The constructor for Insets is `Insets(int top, int left, int bottom, int right)`

int ipadx, ipady

Specifies the internal padding of the component, how much space to add to the minimum width of the component.

double gridx, gridy

Specifies the cell containing the leading edge of the component's display area, where the first cell in a row has gridx=0 and in a column has gridy=0. RELATIVE specifies that the component be placed just below the component that was added to the container just before this component was added.

double weightx, weighty

Specifies how to distribute extra horizontal and vertical space.

GridBagLayout

GridBagLayout()

void setConstraints(Component comp, GridBagConstraints constraints)

Sets the constraints for the specified component in this layout.

GridLayout

GridLayout([int rows, int cols, [int hgap, int vgap]])

BoxLayout

BoxLayout(Container target, int axis)

Axis: LINE_AXIS, PAGE_AXIS, X_AXIS, Y_AXIS

FlowLayout

FlowLayout([int align, [int hgap, int vgap]])

Align: CENTER, LEFT, RIGHT, LEADNG, TRAILING

Primary Containers (all implement ImageObserver)

JFrame extends Window

JFrame(String title)

Container getContentPane()

Returns the contentPane object for this frame.

void setDefaultCloseOperation(int operation)

Sets the operation that will happen by default when the user initiates a "close" on this frame (DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, EXIT_ON_CLOSE, HIDE_ON_CLOSE).

void setJMenuBar(JMenuBar menubar)

Sets the menubar for this frame.

JDialog extends Window

JDialog(Frame owner, String title)

void setDefaultCloseOperation(int operation)

Sets the operation that will happen by default when the user initiates a "close" on this dialog (DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, EXIT_ON_CLOSE, HIDE_ON_CLOSE).

JPanel extends JComponent

JPanel()

Object[] objects = { // Array of widgets to display
compOpt, compOpts,
name, names,
model, models};

Dialogs

JOptionPane

messageType (icon): PLAIN_MESSAGE, INFORMATION_MESSAGE, QUESTION_MESSAGE, WARNING_MESSAGE, ERROR_MESSAGE

optionType: YES_NO_CANCEL_OPTION, YES_NO_OPTION, OK_CANCEL_OPTION

responseType: YES_OPTION, NO_OPTION, CANCEL_OPTION, OK_OPTION, CLOSED_OPTION

this

static int showConfirmDialog(Component parentComponent, Object message, [String title, int optionType, [int messageType, [Icon icon]]])

Brings up dialog with icon, where number of choices is determined by optionType parameter.

static String showInputDialog(Component parentComponent, Object message[, Object initialSelectionValue])

Shows question-message dialog requesting input from user and parented to parentComponent.

static String showInputDialog(Component parentComponent, Object message, String title, int messageType)

Shows dialog requesting input from user with dialog having title and messageType.

static Object showInputDialog(Component parentComponent, Object message, String title, int messageType, Icon icon, Object[] selectionValues, Object initialSelectionValue)

Prompts user for input in a blocking dialog where all options can be specified.

static void showMessageDialog(Component parentComponent, Object message, [String title, int messageType, [Icon icon]])

Brings up dialog displaying a message, specifying all parameters.

static int showOptionDialog(Component parentComponent, Object message, String title, int optionType, int messageType, Icon icon, Object[] options, Object initialValue)

Brings up dialog with specified icon, where initial choice is determined by initialValue and number of choices is determined by optionType.

JDialog (see Primary Containers)

JFileChooser

JFileChooser([String currentDirectoryPath])

void addChoosableFileFilter(FileFilter filter)

Adds a filter to the list of user choosable file filters.

File getSelectedFile()

Returns the selected file.

void setFileFilter(FileFilter filter)

Sets the current file filter.

int showOpenDialog(Component parent)

Pops up an "Open File" file chooser dialog.

int showSaveDialog(Component parent)

Pops up a "Save File" file chooser dialog.

FileNameExtensionFilter extends FileFilter

FileNameExtensionFilter(String description, String... extensions)

JColorChooser

static Color showDialog(Component component, String title, Color initialColor)
Shows a modal color-chooser dialog and blocks until the dialog is hidden.

Widgets (all extend JComponent)

AbstractButton

void addActionListener(ActionListener l)
Adds an ActionListener to the widget.
boolean isSelected()
Returns the state of the button.
void setSelected(boolean b)
Sets the state of the button.
void setIcon(Icon defaultIcon) | setSelectedIcon(Icon selectedIcon)
Sets the button's default (unselected) and selected icon, respectively.

JButton extends AbstractButton

JButton([String text], [Icon icon]])

JCheckBox extends AbstractButton

JCheckBox([String text], [Icon icon], [boolean selected])

JRadioButton extends AbstractButton

JRadioButton(String text, Icon icon, boolean selected)

JToggleButton extends AbstractButton

JToggleButton([String text], [Icon icon], [boolean selected])

JComboBox<E>

JComboBox(E[] items)

void addActionListener(ActionListener l)
Adds an ActionListener to the widget.

void addItem(E item)
Adds an item to the item list.

E getItemAt(int index)
Returns the list item at the specified index.

int getItemCount()
Returns the number of items in the list.

int getSelectedIndex()
Returns the first item in the list that matches the given item.

Object getSelectedItem()
Returns the current selected item.

void setEditable(boolean aFlag)
Determines whether the JComboBox field is editable.

void setSelectedIndex(int anIndex)
Selects the item at index anIndex.

void setSelectedItem(Object anObject)
Sets the selected item in the combo box display area to the object in the argument.

JLabel

JLabel([String text], [Icon icon], [int horizontalAlignment])
horizontalAlignment: SwingConstants.LEFT, CENTER, RIGHT, LEADING, TRAILING

String getText()
Returns the text string that the label displays.

Icon getIcon()
Returns the graphic image (glyph, icon) that the label displays.

void setIcon(Icon icon)
Defines the icon this component will display.

void setText(String text)
Defines the single line of text this component will display.

JMenu

JMenu(String s)

JMenuItem add(JMenuItem menuItem)
Appends a menu item to the end of this menu.

JMenuBar

JMenuBar()

JMenu add(JMenu c)

Appends the specified menu to the end of the menu bar.

JMenuItem

JMenuItem(String text, [Icon icon])

void addActionListener(ActionListener l)

Adds an ActionListener to the widget.

JProgressBar

JProgressBar([int orient], [int min, int max])

orient: SwingConstants.VERTICAL, HORIZONTAL

double getPercentComplete()

Returns the percent complete for the progress bar.

void setValue(int n)

Sets the progress bar's current value to n.

JSlider

JSlider([int orientation], [int min, int max], [int value])

void addChangeListener(ChangeListener l)

Adds a ChangeListener to the slider.

int getValue()

Returns the slider's current value.

void setValue(int n)

Sets the slider's current value to n.

JSpinner

JSpinner(SpinnerModel model)

void addChangeListener(ChangeListener l)

Adds a listener to be notified each time a change to the model occurs.

Object getValue()

Returns the current value of the model, typically this value is displayed by the editor.

void setValue(Object value)

Changes current value of the model, typically this value is displayed by the editor.

JTextField

JTextField([String text], [int columns])

void addActionListener(ActionListener l)

Adds an ActionListener to the widget.

String getText()

Returns the text contained in this TextComponent.

void setText(String t)

Sets the text of this TextComponent to the specified text.

JToolBar

JToolBar()

void addSeparator([Dimension size])

Appends a separator of a specified size to the end of the tool bar.

Models

SpinnerNumberModel implements SpinnerModel

SpinnerNumberModel(int value, int minimum, int maximum, int stepSize)

SpinnerNumberModel(double value, double minimum, double maximum, double stepSize)

Image Classes

BasicStroke implements Stroke

BasicStroke(float width)

Constructs a solid BasicStroke with the specified line width .

BufferedImage extends Image

Color

Color(float r, float g, float b, [float a])

Color(int r, int g, int b, [int a])

Color(int rgb)

int getRGB()

Returns the RGB value representing the color in the default sRGB ColorModel.

Color BLACK, BLUE, CYAN, DARK_GRAY, GRAY, GREEN, LIGHT_GRAY, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW

Dimension

Dimension(int width, int height)

int height, width

Number of pixels in each direction

Font

Font(String name, int style, int size)

Name SERIF, SANS_SERIF, DIALOG, DIALOG_INPUT, MONOSPACE

Style PLAIN, BOLD, ITALIC, BOLD|ITALIC

Graphics2D extends Graphics

Graphics create()

Creates a new Graphics object that is a copy of this Graphics object.

boolean drawImage(Image img, int x, int y, ImageObserver observer)

Draws as much of the specified image as is currently available.

void drawLine(int x1, int y1, int x2, int y2)

Draws a line, using the current color, between the points (x1, y1) and (x2, y2) in this graphics context's coordinate system.

void drawRect(int x, int y, int width, int height)

Draws the outline of the specified rectangle.

void drawString(String str, int x, int y)

Draws the text given by the specified string, using this graphics context's current font and color.

void setBackground(Color color)

Sets the background color for the Graphics2D context.

void setColor(Color c)

Sets this graphics context's current color to the specified color.

void setFont(Font font)

Sets this graphics context's font to the specified font.

void setStroke(Stroke s)

Sets the Stroke (including line width) for the Graphics2D context.

void translate(int x, int y)

Translates the origin of the graphics context to the point (x, y) in the current coordinate system.

ImageIcon implements Icon

ImageIcon([String filename, [String description]])

ImageIO

static BufferedImage read(File input)

Returns a BufferedImage as the result of decoding a supplied File with an ImageReader chosen automatically from among those currently registered. BufferedImage image = ImageIO.read(new File("pix.png"));

Rectangle

Rectangle([int x, int y], int width, int height)

int height, width

Number of pixels in each direction

int x, y

Location on the Cartesian plane

Events

Interface ActionListener

void actionPerformed(ActionEvent e)

Invoked when an action occurs.

ActionEvent

String getActionCommand()

Returns the command string associated with this action.

int getModifiers()

Returns the bitwise OR of modifier keys (ALT_MASK, CTRL_MASK, META_MASK, SHIFT_MASK).

Interface ChangeListener

void stateChanged(ChangeEvent e)

Invoked when the target of the listener has changed its state.

ChangeEvent

Object getSource()

The object on which the Event initially occurred.