



WebbButton

-DEFAULT BORDER SIZE : int = 10 {readOnly}

-borderSize : int -textColor : Color -text : String

-image : BufferedImage

-font : Font

-drawBackground : boolean -backgroundColor : Color

<<constructor>> WebbButton(text : String, clickListener :

ClickListener)

<<constructor>> WebbButton(text: String, imageIn :

BufferedImage, width: int, height: int, clickListener, ClickListener) <<constructor>> WebbButton(imageIn : BufferedImage, width :

int, height : int, clickListener : ClickListener)

<<constructor>> WebbButton(clickListener : ClickListener)

+setClickReleaseListener(clickReleaseListener: clickReleaseListener)

+setBorderSize(borderSize: int) +setTextColor(textColor: setTextColor)

+setDrawBackground(drawBackground : boolean)

+setBackgroundColor(backgroundColor : Color)

+setFont(font : Font) +aetFont(): Font +setText(text : String)

+getText(): String +paint(g : Graphics)

WebbBackButton

<<constructor>> WebbBackButton(contentPane : Component, layout : SpringLayout, clickListener : ClickListener)

WebbCustomScrollpane

-TRACK_COLOR : Color {readOnly} -BAR_COLOR : Color {readOnly}

-BAR_COLOR_HOVER : Color {readOnly}

-BAR_COLOR_CLICK : Color {readOnly}

-ARC SIZE : int=15 {readOnly}

<<constructor>> WebbCustomScrollpane(view : Component, trackColor : Color, barColor : Color, barHoverColor : Color, barClickColor : Color) +setVerticalScrollbarWidth(width: int)

WebbScrollbarUI

-EMPTY_DIMENSION : Dimension {readOnly}

#createDecreaseButton(orientation: int): JButton #createIncreaseButton(orientation: int): JButton #paintTrack(g : Graphics, c : JComponent, r : Rectangle) #paintThumb(g Grahpics, c : JComponent, r : Rectangle) #setThumbBounds(x:int, y:int, width:int, height:int)

WebbProgressBar

-COLOR BACKGROUND : Color {readOnly}

-COLOR_BAR : Color {readOnly}

<<constructor>>

WebbProgressBar(background : Color,

barColor: Color)

WebProgressBarUI

-STROKE_WIDTH : int=3 {readOnly}

#paintDeterminate(g : Graphics, c : JComponent)

WebbRoundedJPanel

rWidth : int {readOnly} rHeight : int {readOnly}

<<constructor>>WebbRoundedJPanel()

<<constructor>>WebbRoundedJPanel(roundWidth: int,

roundhHeight : int)

#paintComponent(g : Graphics)

WebbSimpleImage

image : BufferedImage {readOnly}

<<constructor>> WebbSingleImage(image : BufferedImage, width : int, height : int)

#paintComponent(g : Graphics)

WebbTable

-alternatingColors : boolean

-c1 : Color -c2 : Color

-headerColor : Color

-isEditingAllowed : boolean

<<constructor>> WebbTable(columnNames : String[],

data : String[][]) #setColWidth(col : int, width : int) {leaf}

#removeCellLines() {leaf} +setFont(font : Font)

#setAlternatingColors(c1 : Color, c2 : Color) {leaf}

#setHeaderColor(c: Color) {leaf}

#setTableEditingAllowed(b : boolean) {leaf} +isCellEditable(row : int, column : int) : boolean +isCellSelected(row : int, column : int) : boolean

-updateCellRenderer()

User - Username : String

- Password : String

- MaxSolveTime : Double

AverageSolveTime : Double

- CompletedPuzzles : List<Boolean>

- PuzzleProgress : List<Puzzle>

+ getMaxSolveTime() : Double

+ getAverageSolveTime() : Double

+ getCompletedPuzzles() : List(Boolean>

+ getPuzzleProgress(index : Int) : Puzzle

Level

- Identifier : Int

puzzle

+ changeType(c : Cell, rclick : boolean, visible : boolean)

Puzzle cell - gridSize : Int - numStars ; Int - row : Int - regions : List<List<Cell>> - col : Int regions {List<List>} 1..+ - board : List<List<Cell>> type : cellType - solution : List(Cell> + getRegions(): List<List<Cell>> + getRow(): Int getSolution(): List(Cell> + getCol(): Int board {List<List>} 1..+ + getNumStars(): Int + getType() : String + getTotalStars(): Int + setEmpty() + getcell() : cell + setStar() + setMarker() + checkBoard(visible : Boolean) solution {List} - checkRow(visible : Boolean, current : Cell) + setInvalid() - checkCol(visible : Boolean, current : Cell) + changeType(rClick : Boolean) - checkRegions() - mark/Adjacent(current : Cell) - clearAdjacent(current : Cell)