

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

SEARCH:

Index

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

B

- [board1](#) - Static variable in interface [PegSolitaireGame](#)
- [board2](#) - Static variable in interface [PegSolitaireGame](#)
- [board3](#) - Static variable in interface [PegSolitaireGame](#)
- [board4](#) - Static variable in interface [PegSolitaireGame](#)
- [board5](#) - Static variable in interface [PegSolitaireGame](#)
- [board6](#) - Static variable in interface [PegSolitaireGame](#)

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

SEARCH:

reset

Index

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#)|[All Packages](#)|[Serialized Form](#)

C

- [calculateScore\(PegSolitaireGame.grid_type\[\]\[\]\)](#) - Method in class [PegSoliati](#)[re](#)
- [calculateScore\(PegSolitaireGame.grid_type\[\]\[\]\)](#) - Method in interface [PegSolitaireGame](#)
- [check_down\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in class [PegSoliati](#)[re](#)
- [check_down\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in interface [PegSolitaireGame](#)
- [check_down_manuel\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in class [PegSoliati](#)[re](#)
- [check_game_is_done\(PegSolitaireGame.grid_type\[\]\[\], int, int\)](#) - Method in class [PegSoliati](#)[re](#)
- [check_game_is_done\(PegSolitaireGame.grid_type\[\]\[\], int, int\)](#) - Method in interface [PegSolitaireGame](#)
- [check_left\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in class [PegSoliati](#)[re](#)
- [check_left\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in interface [PegSolitaireGame](#)
- [check_left_manuel\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in class [PegSoliati](#)[re](#)
- [check_right\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in class [PegSoliati](#)[re](#)
- [check_right\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in interface [PegSolitaireGame](#)
- [check_right_manuel\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in class [PegSoliati](#)[re](#)
- [check_up\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in class [PegSoliati](#)[re](#)
- [check_up\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in interface [PegSolitaireGame](#)
- [check_up_manuel\(PegSolitaireGame.grid_type\[\]\[\], int, int, int, int\)](#) - Method in class [PegSoliati](#)[re](#)
- [computer_game\(PegSolitaireGame.grid_type\[\]\[\], int, int, String\)](#) - Method in class [PegSoliati](#)[re](#)
- [computer_game\(PegSolitaireGame.grid_type\[\]\[\], int, int, String\)](#) - Method in interface [PegSolitaireGame](#)
- [computer_game1\(PegSolitaireGame.grid_type\[\]\[\], int, int, String\)](#) - Method in class [PegSoliati](#)[re](#)

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#)|[All Packages](#)|[Serialized Form](#)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

SEARCH:

Index

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

E

[equal\(PegSolitaireGame.grid_type\[\]\[\], String\)](#) - Method in class [PegSoliature](#)

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

SEARCH:

reset

Index

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

I

- [initial_choose_function\(\)](#) - Method in class [PegSoliature](#)
function prepares the user interface initially when the program starts
- [InitilazePanel](#) - Class in [Unnamed Package](#)
- [InitilazePanel\(\)](#) - Constructor for class [InitilazePanel](#)

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

SEARCH:

reset

Index

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

L

- [load_file\(String\)](#) - Method in class [PegSoliatire](#)
- [load_file\(String\)](#) - Method in interface [PegSolitaireGame](#)

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

SEARCH:

reset

Index

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#)|[All Packages](#)|[Serialized Form](#)

M

- [main](#) - Class in [Unnamed Package](#)
- [main\(\)](#) - Constructor for class [main](#)
- [main\(String\[\]\)](#) - Static method in class [main](#)
- [manage_game\(PegSolitaireGame.grid_type\[\]\[\], int, int, String, int\)](#) - Method in class [PegSoliature](#)
- [manage_game\(PegSolitaireGame.grid_type\[\]\[\], int, int, String, int\)](#) - Method in interface [PegSolitaireGame](#)

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#)|[All Packages](#)|[Serialized Form](#)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

SEARCH:

reset

Index

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#)|[All Packages](#)|[Serialized Form](#)

P

[peg](#) - Enum constant in enum class [PegSolitaireGame.grid_type](#)

[PegSoliatire](#) - Class in [Unnamed Package](#)

[PegSolitaireGame](#) - Interface in [Unnamed Package](#)

[PegSolitaireGame.grid_type](#) - Enum Class in [Unnamed Package](#)

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#)|[All Packages](#)|[Serialized Form](#)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

SEARCH:

reset

Index

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

S

[space](#) - Enum constant in enum class [PegSolitaireGame.grid_type](#)

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

SEARCH:

reset

Index

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

V

[valueOf\(String\)](#) - Static method in enum class [PegSolitaireGame.grid_type](#)
Returns the enum constant of this class with the specified name.

[values\(\)](#) - Static method in enum class [PegSolitaireGame.grid_type](#)
Returns an array containing the constants of this enum class, in the order they are declared.

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

SEARCH:

reset

Index

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

W

[wall](#) - Enum constant in enum class [PegSolitaireGame.grid_type](#)

[B](#) [C](#) [E](#) [I](#) [L](#) [M](#) [P](#) [S](#) [V](#) [W](#)
[All Classes and Interfaces](#) | [All Packages](#) | [Serialized Form](#)

raptor not found.



- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)
- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH:

reset

Class InitilazePanel

[java.lang.Object](#)
[java.awt.Component](#)
[java.awt.Container](#)
[java.awt.Window](#)
[java.awt.Frame](#)
[javax.swing.JFrame](#)
InitilazePanel

All Implemented Interfaces:
[ImageObserver](#), [MenuContainer](#), [Serializable](#), [Accessible](#), [RootPaneContainer](#), [WindowConstants](#)

public class InitilazePanel extends [JFrame](#)

See Also:

- [Serialized Form](#)
- **Nested Class Summary**

Nested classes/interfaces inherited from class java.awt.[Window](#)

[Window.Type](#)

Nested classes/interfaces inherited from class java.awt.[Component](#)

[Component.BaselineResizeBehavior](#)

• Field Summary

Fields inherited from class java.awt.[Frame](#)

[CROSSHAIR_CURSOR](#), [DEFAULT_CURSOR](#), [E_RESIZE_CURSOR](#), [HAND_CURSOR](#), [ICONIFIED](#), [MAXIMIZED_BOTH](#), [MAXIMIZED_HORIZ](#), [MAXIMIZED_VERT](#), [MOVE_CURSOR](#), [N_RESIZE_CURSOR](#), [NE_RESIZE_CURSOR](#), [NORMAL](#), [NW_RESIZE_CURSOR](#), [S_RESIZE_CURSOR](#), [SE_RESIZE_CURSOR](#), [SW_RESIZE_CURSOR](#), [TEXT_CURSOR](#), [W_RESIZE_CURSOR](#), [WAIT_CURSOR](#)

Fields inherited from class java.awt.[Component](#)

[BOTTOM_ALIGNMENT](#), [CENTER_ALIGNMENT](#), [LEFT_ALIGNMENT](#), [RIGHT_ALIGNMENT](#), [TOP_ALIGNMENT](#)

Fields inherited from interface java.awt.image.[ImageObserver](#)

[ABORT](#), [ALLBITS](#), [ERROR](#), [FRAMEBITS](#), [HEIGHT](#), [PROPERTIES](#), [SOMEBITS](#), [WIDTH](#)

Fields inherited from interface javax.swing.[WindowConstants](#)

[DISPOSE_ON_CLOSE](#), [DO_NOTHING_ON_CLOSE](#), [EXIT_ON_CLOSE](#), [HIDE_ON_CLOSE](#)

• Constructor Summary

Constructors
Constructor
Description
[InitilazePanel\(\)](#)

• Method Summary

Methods inherited from class [javax.swing.JFrame](#)

[getAccessibleContext](#), [getContentPane](#), [getDefaultCloseOperation](#), [getGlassPane](#), [getGraphics](#), [getJMenuBar](#), [getLayeredPane](#), [getRootPane](#), [getTransferHandler](#), [isDefaultLookAndFeelDecorated](#), [remove](#), [repaint](#), [setContentPane](#), [setDefaultCloseOperation](#), [setDefaultLookAndFeelDecorated](#), [setGlassPane](#), [setIconImage](#), [setJMenuBar](#), [setLayeredPane](#), [setLayout](#), [setTransferHandler](#), [update](#)

Methods inherited from class [java.awt.Frame](#)

[addNotify](#), [getCursorType](#), [getExtendedState](#), [getFrames](#), [setIconImage](#), [getMaximizedBounds](#), [getMenuBar](#), [getState](#), [getTitle](#), [isResizable](#), [isUndecorated](#), [remove](#), [removeNotify](#), [setBackground](#), [setCursor](#), [setExtendedState](#), [setMaximizedBounds](#), [setMenuBar](#), [setOpacity](#), [setResizable](#), [setShape](#), [setState](#), [setTitle](#), [setUndecorated](#)

Methods inherited from class [java.awt.Window](#)

[addPropertyChangeListener](#), [addPropertyChangeListener](#), [addWindowFocusListener](#), [addWindowListener](#), [addWindowStateListener](#), [applyResourceBundle](#), [applyResourceBundle](#), [createBufferStrategy](#), [createBufferStrategy](#), [dispose](#), [setBackground](#), [getBufferStrategy](#), [getFocusableWindowState](#), [getFocusCycleRootAncestor](#), [getFocusOwner](#), [getFocusTraversalKeys](#), [getIconImages](#), [getInputContext](#), [getListeners](#), [getLocale](#), [getModalExclusionType](#), [getMostRecentFocusOwner](#), [getOpacity](#), [getOwnedWindows](#), [getOwner](#), [getOwnerlessWindows](#), [getShape](#), [getToolkit](#), [getType](#), [getWarningString](#), [getWindowFocusListeners](#), [getWindowListeners](#), [getWindows](#), [getWindowStateListeners](#), [hide](#), [isActive](#), [isAlwaysOnTop](#), [isAlwaysOnTopSupported](#), [isAutoRequestFocus](#), [isFocusableWindow](#), [isFocusCycleRoot](#), [isFocused](#), [isLocationByPlatform](#), [isOpaque](#), [isShowing](#), [isValidateRoot](#), [pack](#), [paint](#), [postEvent](#), [removeWindowFocusListener](#), [removeWindowListener](#), [removeWindowStateListener](#), [reshape](#), [setAlwaysOnTop](#), [setAutoRequestFocus](#), [setBounds](#), [setBounds](#), [setCursor](#), [setFocusableWindowState](#), [setFocusCycleRoot](#), [setIconImages](#), [setLocation](#), [setLocation](#), [setLocationByPlatform](#), [setLocationRelativeTo](#), [setMinimumSize](#), [setModalExclusionType](#), [setSize](#), [setSize](#), [setType](#), [setVisible](#), [show](#), [toBack](#), [toFront](#)

Methods inherited from class [java.awt.Container](#)

[add](#), [add](#), [add](#), [add](#), [add](#), [addContainerListener](#), [applyComponentOrientation](#), [areFocusTraversalKeysSet](#), [countComponents](#), [deliverEvent](#), [doLayout](#), [findComponentAt](#), [findComponentAt](#), [getAlignmentX](#), [getAlignmentY](#), [getComponent](#), [getComponentAt](#), [getComponentAt](#), [getComponentCount](#), [getComponents](#), [getComponentZOrder](#), [getContainerListeners](#), [getFocusTraversalPolicy](#), [getInsets](#), [getLayout](#), [getMaximumSize](#), [getMinimumSize](#), [getMousePosition](#), [getPreferredSize](#), [insets](#), [invalidate](#), [isAncestorOf](#), [isFocusCycleRoot](#), [isFocusTraversalPolicyProvider](#), [isFocusTraversalPolicySet](#), [layout](#), [list](#), [list](#), [locate](#), [minimumSize](#), [paintComponents](#), [preferredSize](#), [print](#), [printComponents](#), [remove](#), [removeAll](#), [removeContainerListener](#), [setComponentZOrder](#), [setFocusTraversalKeys](#), [setFocusTraversalPolicy](#), [setFocusTraversalPolicyProvider](#), [setFont](#), [transferFocusDownCycle](#), [validate](#)

Methods inherited from class [java.awt.Component](#)

[action](#), [add](#), [addComponentListener](#), [addFocusListener](#), [addHierarchyBoundsListener](#), [addHierarchyListener](#), [addInputMethodListener](#), [addKeyListener](#), [addMouseListener](#), [addMouseMotionListener](#), [addMouseWheelListener](#), [bounds](#), [checkImage](#), [checkImage](#), [contains](#), [contains](#), [createImage](#), [createImage](#), [createVolatileImage](#), [createVolatileImage](#), [disable](#), [dispatchEvent](#), [enable](#), [enable](#), [enableInputMethods](#), [firePropertyChange](#), [firePropertyChange](#), [firePropertyChange](#), [firePropertyChange](#), [firePropertyChange](#), [firePropertyChange](#), [getBaseline](#), [getBaselineResizeBehavior](#), [getBounds](#), [getBounds](#), [getColorModel](#), [getComponentListeners](#), [getComponentOrientation](#), [getCursor](#), [getDropTarget](#), [getFocusListeners](#), [getFocusTraversalKeysEnabled](#), [getFont](#), [getFontMetrics](#), [getForeground](#), [getGraphicsConfiguration](#), [getHeight](#), [getHierarchyBoundsListeners](#), [getHierarchyListeners](#), [getIgnoreRepaint](#), [getInputMethodListeners](#), [getInputMethodRequests](#), [getKeyListeners](#), [getLocation](#), [getLocation](#), [getLocationOnScreen](#), [getMouseListeners](#), [getMouseMotionListeners](#), [getMousePosition](#), [getMouseWheelListeners](#), [getName](#), [getParent](#), [getPropertyChangeListeners](#), [getPropertyChangeListeners](#), [getSize](#), [setSize](#), [getTreeLock](#), [getWidth](#), [getX](#), [getY](#), [getFocus](#), [handleEvent](#), [hasFocus](#), [imageUpdate](#), [inside](#), [isBackgroundSet](#), [isCursorSet](#), [isDisplayable](#), [isDoubleBuffered](#), [isEnabled](#), [isFocusable](#), [isFocusOwner](#), [isFocusTraversable](#), [isFontSet](#), [isForegroundSet](#), [isLightweight](#), [isMaximumSizeSet](#), [isMinimumSizeSet](#), [isPreferredSizeSet](#), [isValid](#), [isVisible](#), [keyDown](#), [keyUp](#), [list](#), [list](#), [list](#), [location](#), [lostFocus](#), [mouseDown](#), [mouseDrag](#), [mouseEnter](#), [mouseExit](#), [mouseMove](#), [mouseUp](#), [move](#), [nextFocus](#), [paintAll](#), [prepareImage](#), [prepareImage](#), [printAll](#), [removeComponentListener](#), [removeFocusListener](#), [removeHierarchyBoundsListener](#), [removeHierarchyListener](#), [removeInputMethodListener](#), [removeKeyListener](#), [removeMouseListener](#), [removeMouseMotionListener](#), [removeMouseWheelListener](#), [removePropertyChangeListener](#), [removePropertyChangeListener](#), [repaint](#), [repaint](#), [repaint](#), [requestFocus](#), [requestFocus](#), [requestFocusInWindow](#), [requestFocusInWindow](#), [resize](#), [resize](#), [revalidate](#), [setComponentOrientation](#), [setDropTarget](#), [setEnabled](#), [setFocusable](#), [setFocusTraversalKeysEnabled](#), [setForeground](#), [setIgnoreRepaint](#), [setLocale](#), [setMaximumSize](#), [setMixingCutoutShape](#), [setName](#), [setPreferredSize](#), [show](#), [size](#), [toString](#), [transferFocus](#), [transferFocusBackward](#), [transferFocusUpCycle](#)

Methods inherited from class java.lang.[Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Methods inherited from interface java.awt.[MenuContainer](#)

[getFont](#), [postEvent](#)

• **Constructor Details**

- **InitilazePanel**

public InitilazePanel()

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

SEARCH:

reset

Class main

[java.lang.Object](#)
main

public class main extends [Object](#)

• Constructor Summary

Constructors
Constructor
Description
[main\(\)](#)

• Method Summary

All MethodsStatic MethodsConcrete Methods

Modifier and Type
Method
Description
static void
[main](#)([String](#)[] args)

Methods inherited from class java.lang.[Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

• Constructor Details

- **main**

public main()

• Method Details

- **main**

public static void main([String](#)[] args)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

SEARCH:

reset

Class main

[java.lang.Object](#)
main

public class main extends [Object](#)

• Constructor Summary

Constructors
Constructor
Description
[main\(\)](#)

• Method Summary

All MethodsStatic MethodsConcrete Methods

Modifier and Type
Method
Description
static void
[main](#)([String](#)[] args)

Methods inherited from class java.lang.[Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

• Constructor Details

- **main**

public main()

• Method Details

- **main**

public static void main([String](#)[] args)

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)

- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)

- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH:

reset

Class PegSoliatire

[java.lang.Object](#)
PegSoliatire

All Implemented Interfaces:
[PegSolitaireGame](#)

public class PegSoliatire extends [Object](#) implements [PegSolitaireGame](#)

• Nested Class Summary

Nested classes/interfaces inherited from interface [PegSolitaireGame](#)

[PegSolitaireGame.grid_type](#)

• Field Summary

Fields inherited from interface [PegSolitaireGame](#)

[board1](#), [board2](#), [board3](#), [board4](#), [board5](#), [board6](#)

• Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type		
Method		
Description		
int	calculateScore (PegSolitaireGame.grid_type [][] board)	
int	check_down (PegSolitaireGame.grid_type [][] board, int i, int j, int N1, int N2)	
int	check_down_manuel (PegSolitaireGame.grid_type [][] board, int first_row, int first_col, int sec_row, int sec_col)	
int	check_game_is_done (PegSolitaireGame.grid_type [][] board, int N1, int N2)	
int	check_left (PegSolitaireGame.grid_type [][] board, int i, int j, int N1, int N2)	
int	check_left_manuel (PegSolitaireGame.grid_type [][] board, int first_row, int first_col, int sec_row, int sec_col)	
int	check_right (PegSolitaireGame.grid_type [][] board, int i, int j, int N1, int N2)	

```

int
check\_right\_manuel(PegSolitaireGame.grid\_type[][] board, int first_row, int first_col, int sec_row, int sec_col)

int
check\_up(PegSolitaireGame.grid\_type[][] board, int i, int j, int N1, int N2)

int
check\_up\_manuel(PegSolitaireGame.grid\_type[][] board, int first_row, int first_col, int sec_row, int sec_col)

void
computer\_game(PegSolitaireGame.grid\_type[][] board, int N1, int N2, String board_type)

void
computer\_game1(PegSolitaireGame.grid\_type[][] board, int N1, int N2, String board_type)

PegSolitaireGame.grid\_type[][]
equal(PegSolitaireGame.grid\_type[][] obj, String board_type)

void
initial\_choose\_function()
function prepares the user interface initially when the program starts
PegSolitaireGame.grid\_type[][]
load\_file(String filename)

PegSolitaireGame.grid\_type[][]
manage\_game(PegSolitaireGame.grid\_type[][] board1_temp, int N1, int N2, String board_type, int z)

```

Methods inherited from class java.lang.[Object](#)

[equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

• Method Details

◦ **equal**

```
public PegSolitaireGame.grid\_type[][] equal(PegSolitaireGame.grid\_type[][] obj, String board_type)
```

Parameters:

obj -

board_type - : type of the board for indication as a string such as "board1"

Returns:

returns the grid type which is cloned

◦ **check_left**

```
public int check\_left(PegSolitaireGame.grid\_type[][] board, int i, int j, int N1, int N2)
```

Specified by:

[check_left](#) in interface [PegSolitaireGame](#)

Parameters:

board - : grid that will be controlled for legal left move

i - : row number

j - : col number

N1 - i dimension of the grid

N2 - j dimension of the grid

Returns:

1 if the given values on the given grid has a legal move for left direction

◦ **check_right**

```
public int check\_right(PegSolitaireGame.grid\_type[][] board, int i, int j, int N1, int N2)
```

Specified by:

[check_right](#) in interface [PegSolitaireGame](#)

Parameters:

board - : grid that will be controlled for legal right move

i - : row number

j - : col number

N1 - i dimension of the grid

N2 - j dimension of the grid

Returns:

1 if the given values on the given grid has a legal move for right direction

- **load_file**

public [PegSolitaireGame.grid_type](#)[][] load_file([String](#) filename)

Specified by:

[load_file](#) in interface [PegSolitaireGame](#)

Parameters:

filename - name of the file

Returns:

returns the grid on the file

- **check_up**

public int check_up([PegSolitaireGame.grid_type](#)[][] board, int i, int j, int N1, int N2)

Specified by:

[check_up](#) in interface [PegSolitaireGame](#)

- **check_down**

public int check_down([PegSolitaireGame.grid_type](#)[][] board, int i, int j, int N1, int N2)

Specified by:

[check_down](#) in interface [PegSolitaireGame](#)

Parameters:

board - : grid that will be controlled for legal down move

i - : row number

j - : col number

N1 - i dimension of the grid

N2 - j dimension of the grid

Returns:

1 if the given values on the given grid has a legal move for down direction

- **check_up_manuel**

public int check_up_manuel([PegSolitaireGame.grid_type](#)[][] board, int first_row, int first_col, int sec_row, int sec_col)

Parameters:

board - grid type

first_row - row number of the first touched button

first_col - col number of the first touched button

sec_row - row number of the second touched button

sec_col - col number of the second touched button

Returns:

1 if the given values on the given grid has a legal move for up direction

- **check_down_manuel**

public int check_down_manuel([PegSolitaireGame.grid_type](#)[][] board, int first_row, int first_col, int sec_row, int sec_col)

Parameters:

board - grid type

first_row - row number of the first touched button

first_col - col number of the first touched button

sec_row - row number of the second touched button

sec_col - col number of the second touched button

Returns:

1 if the given values on the given grid has a legal move for down direction

- **check_left_manuel**

public int check_left_manuel([PegSolitaireGame.grid_type](#)[][] board, int first_row, int first_col, int sec_row, int sec_col)

Parameters:

board - grid type

first_row - row number of the first touched button

first_col - col number of the first touched button

sec_row - row number of the second touched button

sec_col - col number of the second touched button

Returns:

1 if the given values on the given grid has a legal move for left direction

- **check_right_manuel**

```
public int check_right_manuel(PegSolitaireGame.grid\_type[][] board, int first_row, int first_col, int sec_row, int sec_col)
```

Parameters:

board - grid type
first_row - row number of the first touched button
first_col - col number of the first touched button
sec_row - row number of the second touched button
sec_col - col number of the second touched button

Returns:

1 if the given values on the given grid has a legal move for right direction

- **check_game_is_done**

```
public int check_game_is_done(PegSolitaireGame.grid\_type[][] board, int N1, int N2)
```

Specified by:

[check_game_is_done](#) in interface [PegSolitaireGame](#)

Parameters:

board - grid type
N1 - row dimension of the grid
N2 - col dimension of the grid

Returns:

1 if the game is done else returns 0

- **initial_choose_function**

```
public void initial_choose_function()
```

function prepares the user interface initially when the program starts

- **manage_game**

```
public PegSolitaireGame.grid\_type[][] manage_game(PegSolitaireGame.grid\_type[][] board1_temp, int N1, int N2, String board_type, int z)
```

Specified by:

[manage_game](#) in interface [PegSolitaireGame](#)

Parameters:

board1_temp - : static board type.
N1 - : row parameter for board
N2 - : col parameter for board
board_type - : type of the board as a string such as "board1" for indication
z - : initial parameter to understand if the game is just started or not

Returns:

manages the game, makes the moves, creates the GridLayout, buttons inside it has action listeners to make the moves that user selected

- **calculateScore**

```
public int calculateScore(PegSolitaireGame.grid\_type[][] board)
```

Specified by:

[calculateScore](#) in interface [PegSolitaireGame](#)

Parameters:

board - board type, it's number of peg's left will be calculated inside the function

Returns:

the number of peg left in the game

- **computer_game**

```
public void computer_game(PegSolitaireGame.grid\_type[][] board, int N1, int N2, String board_type)
```

Specified by:

[computer_game](#) in interface [PegSolitaireGame](#)

Parameters:

board - board type
N1 - : row dimension of the board
N2 - : col dimension of the board
board_type - type of the board as string Manages the computer game. Makes the moves until end and finally shows the final board to screen.

- **computer_game1**

```
public void computer_game1(PegSolitaireGame.grid\_type[][] board, int N1, int N2, String board_type)
```

Parameters:

board - board type

N1 - : row dimension of the board

N2 - : col dimension of the board

board_type - type of the board as string Manages the computer game but if it is loaded from file. Makes the moves until end and finally shows the final board to screen.

- [Package](#)
- [Class](#)
- [Tree](#)
- [Index](#)
- [Help](#)
- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)
- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

SEARCH:

reset

Interface PegSolitaireGame

All Known Implementing Classes:
[PegSoliatire](#)

public interface PegSolitaireGame

• Nested Class Summary

Nested Classes
Modifier and Type
Interface
Description
static enum
PegSolitaireGame.grid_type

• Field Summary

Fields
Modifier and Type
Field
Description
static final PegSolitaireGame.grid_type [][] board1
static final PegSolitaireGame.grid_type [][] board2
static final PegSolitaireGame.grid_type [][] board3
static final PegSolitaireGame.grid_type [][] board4
static final PegSolitaireGame.grid_type [][] board5
static final PegSolitaireGame.grid_type [][] board6

• Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type		
Method		
Description		

```
int
calculateScore(PegSolitaireGame.grid\_type[][] board)

int
check\_down(PegSolitaireGame.grid\_type[][] board, int i, int j, int N1, int N2)

int
check\_game\_is\_done(PegSolitaireGame.grid\_type[][] board, int N1, int N2)

int
check\_left(PegSolitaireGame.grid\_type[][] board, int i, int j, int N1, int N2)

int
check\_right(PegSolitaireGame.grid\_type[][] board, int i, int j, int N1, int N2)

int
check\_up(PegSolitaireGame.grid\_type[][] board, int i, int j, int N1, int N2)

void
computer\_game(PegSolitaireGame.grid\_type[][] board, int N1, int N2, String board_type)

PegSolitaireGame.grid\_type[][]
load\_file(String filename)

PegSolitaireGame.grid\_type[][]
manage\_game(PegSolitaireGame.grid\_type[][] board1_temp, int N1, int N2, String board_type, int z)
```

• Field Details

- **board1**

static final [PegSolitaireGame.grid_type](#)[][] board1

- **board2**

static final [PegSolitaireGame.grid_type](#)[][] board2

- **board3**

static final [PegSolitaireGame.grid_type](#)[][] board3

- **board4**

static final [PegSolitaireGame.grid_type](#)[][] board4

- **board5**

static final [PegSolitaireGame.grid_type](#)[][] board5

- **board6**

static final [PegSolitaireGame.grid_type](#)[][] board6

• Method Details

- **check_left**

int [check_left](#)([PegSolitaireGame.grid_type](#)[][] board, int i, int j, int N1, int N2)

- **check_right**

int [check_right](#)([PegSolitaireGame.grid_type](#)[][] board, int i, int j, int N1, int N2)

- **check_up**

int [check_up](#)([PegSolitaireGame.grid_type](#)[][] board, int i, int j, int N1, int N2)

- **check_down**

int [check_down](#)([PegSolitaireGame.grid_type](#)[][] board, int i, int j, int N1, int N2)

- **check_game_is_done**

int check_game_is_done([PegSolitaireGame.grid_type](#)[][] board, int N1, int N2)

- **manage_game**

[PegSolitaireGame.grid_type](#)[][] manage_game([PegSolitaireGame.grid_type](#)[][] board1_temp, int N1, int N2, [String](#) board_type, int z)

- **calculateScore**

int calculateScore([PegSolitaireGame.grid_type](#)[][] board)

- **computer_game**

void computer_game([PegSolitaireGame.grid_type](#)[][] board, int N1, int N2, [String](#) board_type)

- **load_file**

[PegSolitaireGame.grid_type](#)[][] load_file([String](#) filename)