

# **Java Awesome Alphabet**

**Mark Jaleel Mike Vivek Levi**

Boston University

---

**Package**

**edu.bu.cs673.AwesomeAlphabet.controller**

## edu.bu.cs673.AwesomeAlphabet.controller Class AlphabetPageController

java.lang.Object

```

+--edu.bu.cs673.AwesomeAlphabet.controller.PageController
    |
    +--edu.bu.cs673.AwesomeAlphabet.controller.AlphabetPageController
  
```

public class **AlphabetPageController**  
extends [PageController](#)

This class defines the Alphabet Page Controller.

### Field Summary

private	<a href="#">m_alphabet</a>
---------	----------------------------

Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[m\\_pageObserver](#)

### Constructor Summary

public	<a href="#">AlphabetPageController</a> ( <a href="#">IPageObserver</a> pageObserver, <a href="#">Alphabet</a> alphabet) Class constructor.
--------	---

### Method Summary

java.util.Iterator	<a href="#">GetLetterIterator</a> () Gets a Letter iterator from the Alphabet model.
boolean	<a href="#">GoToLetterPage</a> ( <a href="#">Letter</a> cLetter) Causes the Letter Page to be shown.
boolean	<a href="#">GoToTitlePage</a> () Causes the Title Page to be shown.
boolean	<a href="#">PlayAlphabetSong</a> () Plays the alphabet song.

Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[GoToPage](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

(continued on next page)

(continued from last page)

## m\_alphabet

```
private edu.bu.cs673.AwesomeAlphabet.model.Alphabet m_alphabet
```

## Constructors

### AlphabetPageController

```
public AlphabetPageController(IPageObserver pageObserver,  
                               Alphabet alphabet)
```

Class constructor.

**Parameters:**

pageObserver - A page observer reference so that page transitions may be requested. For example, this may refer to the main window.

alphabet - The Alphabet model.

## Methods

### GetLetterIterator

```
public java.util.Iterator GetLetterIterator()
```

Gets a Letter iterator from the Alphabet model.

**Returns:**

Letter Iterator.

### PlayAlphabetSong

```
public boolean PlayAlphabetSong()
```

Plays the alphabet song.

**Returns:**

True if song was played successfully.

### GoToTitlePage

```
public boolean GoToTitlePage()
```

Causes the Title Page to be shown.

**Returns:**

True if Title Page is able to be shown.

### GoToLetterPage

```
public boolean GoToLetterPage(Letter cLetter)
```

Causes the Letter Page to be shown.

**Parameters:**

cLetter - The letter to be shown on the Letter Page.

(continued from last page)

**Returns:**

True if Letter Page is able to be shown.

## edu.bu.cs673.AwesomeAlphabet.controller Class ButtonHandler

java.lang.Object

└─ edu.bu.cs673.AwesomeAlphabet.controller.ButtonHandler

### All Implemented Interfaces:

java.awt.event.ActionListener

public class **ButtonHandler**  
extends java.lang.Object  
implements java.awt.event.ActionListener

This Class handles user actions on the buttons on the GUI

### Field Summary

static	<a href="#">log</a>
private	<a href="#">method</a>
private	<a href="#">pv</a>

### Constructor Summary

public	<a href="#">ButtonHandler</a> ( <a href="#">PageView</a> pv, java.lang.String method) Constructor for ButtonHandler.
--------	---

### Method Summary

void	<a href="#">actionPerformed</a> (java.awt.event.ActionEvent ae) Method actionPerformed.
------	--

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

#### Methods inherited from interface java.awt.event.ActionListener

actionPerformed

### Fields

#### pv

private edu.bu.cs673.AwesomeAlphabet.view.PageView **pv**

## method

```
private java.lang.reflect.Method method
```

---

## log

```
static org.apache.log4j.Logger log
```

---

# Constructors

## ButtonHandler

```
public ButtonHandler(PageView pv,  
                     java.lang.String method)
```

Constructor for ButtonHandler.

### Parameters:

pv - PageView  
method - String

# Methods

## actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent ae)
```

Method actionPerformed.

### Parameters:

ae - ActionEvent

### See Also:

`ActionListener.actionPerformed(java.awt.event.ActionEvent)`

## edu.bu.cs673.AwesomeAlphabet.controller Class LabelClickHandler

java.lang.Object

└─ edu.bu.cs673.AwesomeAlphabet.controller.LabelClickHandler

### All Implemented Interfaces:

java.awt.event.MouseListener

public class **LabelClickHandler**  
 extends java.lang.Object  
 implements java.awt.event.MouseListener

### Field Summary

static	<a href="#">log</a>
private	<a href="#">method</a>
private	<a href="#">pv</a>

### Constructor Summary

public	<a href="#">LabelClickHandler</a> ( <a href="#">PageView</a> pv, java.lang.String method) Constructor for LabelClickHandler.
--------	---

### Method Summary

void	<a href="#">mouseClicked</a> (java.awt.event.MouseEvent e) Method mouseClicked.
void	<a href="#">mouseEntered</a> (java.awt.event.MouseEvent e) Method mouseEntered.
void	<a href="#">mouseExited</a> (java.awt.event.MouseEvent e) Method mouseExited.
void	<a href="#">mousePressed</a> (java.awt.event.MouseEvent e) Method mousePressed.
void	<a href="#">mouseReleased</a> (java.awt.event.MouseEvent e) Method mouseReleased.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Methods inherited from interface java.awt.event.MouseListener

mouseClicked, mouseEntered, mouseExited, mousePressed, mouseReleased



## Fields

### **pv**

```
private edu.bu.cs673.AwesomeAlphabet.view.PageView pv
```

### **method**

```
private java.lang.reflect.Method method
```

### **log**

```
static org.apache.log4j.Logger log
```

## Constructors

### **LabelClickHandler**

```
public LabelClickHandler(PageView pv,  
                        java.lang.String method)
```

Constructor for LabelClickHandler.

#### **Parameters:**

pv - PageView  
method - String

## Methods

### **mouseClicked**

```
public void mouseClicked(java.awt.event.MouseEvent e)
```

Method mouseClicked.

#### **Parameters:**

e - MouseEvent

#### **See Also:**

`MouseListener.mouseClicked(java.awt.event.MouseEvent)`

### **mouseEntered**

```
public void mouseEntered(java.awt.event.MouseEvent e)
```

Method mouseEntered.

#### **Parameters:**

e - MouseEvent

(continued from last page)

**See Also:**`MouseListener.mouseEntered( java.awt.event.MouseEvent )`

---

**mouseExited**

```
public void mouseExited( java.awt.event.MouseEvent e )
```

Method mouseExited.

**Parameters:**

e - MouseEvent

**See Also:**`MouseListener.mouseExited( java.awt.event.MouseEvent )`

---

**mousePressed**

```
public void mousePressed( java.awt.event.MouseEvent e )
```

Method mousePressed.

**Parameters:**

e - MouseEvent

**See Also:**`MouseListener.mousePressed( java.awt.event.MouseEvent )`

---

**mouseReleased**

```
public void mouseReleased( java.awt.event.MouseEvent e )
```

Method mouseReleased.

**Parameters:**

e - MouseEvent

**See Also:**`MouseListener.mouseReleased( java.awt.event.MouseEvent )`

---

## edu.bu.cs673.AwesomeAlphabet.controller Class LetterPageController

java.lang.Object



public class **LetterPageController**  
extends [PageController](#)

This class defines the Letter Page Controller.

### Field Summary

private	<a href="#">m_alphabet</a>
private	<a href="#">m_view</a>

Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[m\\_pageObserver](#)

### Constructor Summary

public	<a href="#">LetterPageController</a> ( <a href="#">IPageObserver</a> pageObserver, <a href="#">LetterPageView</a> view, <a href="#">Alphabet</a> alphabet) Class constructor.
--------	--

### Method Summary

javax.swing.Icon	<a href="#">GetIcon</a> (int width, int height) This method gets the current word picture from the Letter model.
char	<a href="#">GetLowercaseLetterAsChar</a> () This method gets the current letter from the model in lower-case.
void	<a href="#">GetNextExample</a> () This method causes the controller to update the current Letter model so that the next word example is shown.
boolean	<a href="#">GetNextLetter</a> () This method causes the controller to update the Alphabet model so that the next letter is displayed.
boolean	<a href="#">GetPreviousLetter</a> () This method causes the controller to update the Alphabet model so that the previous letter is displayed.
char	<a href="#">GetUppercaseLetterAsChar</a> () This method gets the current letter from the model in upper-case.
java.lang.String	<a href="#">GetWord</a> () This method gets the current word from the Letter model.

void	<a href="#">GoToAlphabetPage()</a> Causes the Alphabet Page to be shown.
void	<a href="#">GoToTitlePage()</a> Causes the Title Page to be shown.
void	<a href="#">LetterClicked()</a> This method is used to notify the controller that the image of the letter was clicked.
void	<a href="#">ObserveCurrentLetter()</a>
void	<a href="#">PictureClicked()</a> This method is used to notify the controller that the word's picture was clicked.
void	<a href="#">StopSound()</a> ( <a href="#">Letter</a> letter)

Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[GoToPage](#)

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait`

## Fields

### **m\_view**

```
private edu.bu.cs673.AwesomeAlphabet.view.LetterPageView m_view
```

### **m\_alphabet**

```
private edu.bu.cs673.AwesomeAlphabet.model.Alphabet m_alphabet
```

## Constructors

### **LetterPageController**

```
public LetterPageController(IPageObserver pageObserver,
LetterPageView view,
Alphabet alphabet)
```

Class constructor.

#### **Parameters:**

`pageObserver` - A page observer reference so that page transitions may be requested. For example, this may refer to the main window.  
`view` - The view.  
`alphabet` - The Alphabet model.

(continued from last page)

## Methods

### StopSound

```
private void StopSound(Letter letter)
```

### LetterClicked

```
public void LetterClicked()
```

This method is used to notify the controller that the image of the letter was clicked. This will cause a phonetic letter sound to be played.

### PictureClicked

```
public void PictureClicked()
```

This method is used to notify the controller that the word's picture was clicked. This will cause the word sound to be played.

### GetNextLetter

```
public boolean GetNextLetter()
```

This method causes the controller to update the Alphabet model so that the next letter is displayed.

**Returns:**

True if next letter can be displayed.

### GetPreviousLetter

```
public boolean GetPreviousLetter()
```

This method causes the controller to update the Alphabet model so that the previous letter is displayed.

**Returns:**

True if previous letter can be displayed.

### GoToAlphabetPage

```
public void GoToAlphabetPage()
```

Causes the Alphabet Page to be shown.

### GoToTitlePage

```
public void GoToTitlePage()
```

Causes the Title Page to be shown.

### GetNextExample

```
public void GetNextExample()
```

This method causes the controller to update the current Letter model so that the next word example is shown.

---

## GetLowercaseLetterAsChar

```
public char GetLowercaseLetterAsChar()
```

This method gets the current letter from the model in lower-case.

**Returns:**

Current letter as a lower-case char.

---

## GetUppercaseLetterAsChar

```
public char GetUppercaseLetterAsChar()
```

This method gets the current letter from the model in upper-case.

**Returns:**

Current letter as an upper-case char.

---

## GetIcon

```
public javax.swing.Icon GetIcon(int width,  
                                int height)
```

This method gets the current word picture from the Letter model.

**Returns:**

Word picture as an Icon.

---

## GetWord

```
public java.lang.String GetWord()
```

This method gets the current word from the Letter model.

**Returns:**

Letter word as a String.

---

## ObserveCurrentLetter

```
public void ObserveCurrentLetter()
```

## edu.bu.cs673.AwesomeAlphabet.controller Class OptionsPageController

java.lang.Object

```

+--edu.bu.cs673.AwesomeAlphabet.controller.PageController
    +--edu.bu.cs673.AwesomeAlphabet.controller.OptionsPageController
  
```

public class **OptionsPageController**  
extends [PageController](#)

### Field Summary

	<a href="#">m_view</a>
--	------------------------

Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[m\\_pageObserver](#)

### Constructor Summary

public	<a href="#">OptionsPageController</a> ( <a href="#">IPageObserver</a> pageObserver, <a href="#">OptionsPageView</a> view)
--------	---

### Method Summary

void	<a href="#">GoToManageSettingsPage</a> ()
void	<a href="#">GoToThemesPage</a> ()
void	<a href="#">GoToTitlePage</a> ()
void	<a href="#">GoToWordEditPage</a> ()

Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[GoToPage](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

#### **m\_view**

edu.bu.cs673.AwesomeAlphabet.view.OptionsPageView **m\_view**

(continued from last page)

## Constructors

### OptionsPageController

```
public OptionsPageController(IPageObserver pageObserver,  
                             OptionsPageView view)
```

## Methods

### GoToTitlePage

```
public void GoToTitlePage()
```

---

### GoToThemesPage

```
public void GoToThemesPage()
```

---

### GoToWordEditPage

```
public void GoToWordEditPage()
```

---

### GoToManageSettingsPage

```
public void GoToManageSettingsPage()
```



# edu.bu.cs673.AwesomeAlphabet.controller

## Class PageController

```
java.lang.Object
```

```
└-edu.bu.cs673.AwesomeAlphabet.controller.PageController
```

### Direct Known Subclasses:

[AlphabetPageController](#), [LetterPageController](#), [OptionsPageController](#), [SettingController](#), [ThemeController](#), [TitlePageController](#), [WordEditController](#), [WPSController](#)

```
public abstract class PageController
extends java.lang.Object
```

This class defines the Page Controller. It is the abstract parent class of all page controller classes, including AlphabetPageController, LetterPageController, etc. It manages a Page Observer so that page transitions may be requested.

## Field Summary

private	<a href="#">m_pageObserver</a>
---------	--------------------------------

## Constructor Summary

public	<a href="#">PageController</a> ( <a href="#">IPageObserver</a> pageObserver) Class constructor.
--------	--

## Method Summary

boolean	<a href="#">GoToPage</a> ( <a href="#">PageName</a> page) Commands a page to be shown.
---------	---

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### m\_pageObserver

```
private edu.bu.cs673.AwesomeAlphabet.view.IPageObserver m_pageObserver
```

## Constructors

### PageController

```
public PageController(IPageObserver pageObserver)
```

Class constructor.

(continued from last page)

**Parameters:**

`pageObserver` - A page observer reference so that page transitions may be requested. For example, this may refer to the main window.

## Methods

### GoToPage

protected boolean **GoToPage**([PageName](#) page)

Commands a page to be shown.

**Parameters:**

`page` - The name of the page.

**Returns:**

True if page is able to be shown.

## edu.bu.cs673.AwesomeAlphabet.controller Class SettingController

java.lang.Object

└- [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)  
└- [edu.bu.cs673.AwesomeAlphabet.controller.SettingController](#)

public class **SettingController**  
extends [PageController](#)

### Field Summary

	<a href="#">alphabet</a>
static	<a href="#">log</a>

Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[m\\_pageObserver](#)

### Constructor Summary

public	<a href="#">SettingController</a> ( <a href="#">Alphabet</a> pAlphabet, <a href="#">IPageObserver</a> pageObserver)
--------	---

### Method Summary

java.lang.String	<a href="#">getDisplayOrder</a> ()
int	<a href="#">getMaxExamples</a> ()
void	<a href="#">GoToOptionsMenu</a> ()
void	<a href="#">updateDisplayOrder</a> (java.lang.String displayOrder)
void	<a href="#">updateMaxExamples</a> (int limit)

Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[GoToPage](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

(continued from last page)

## Fields

### **log**

```
static org.apache.log4j.Logger log
```

---

### **alphabet**

```
edu.bu.cs673.AwesomeAlphabet.model.Alphabet alphabet
```

## Constructors

### **SettingController**

```
public SettingController(Alphabet pAlphabet,  
                        IPageObserver pageObserver)
```

## Methods

### **GoToOptionsMenu**

```
public void GoToOptionsMenu()
```

---

### **updateDisplayOrder**

```
public void updateDisplayOrder(java.lang.String displayOrder)
```

---

### **updateMaxExamples**

```
public void updateMaxExamples(int limit)
```

---

### **getDisplayOrder**

```
public java.lang.String getDisplayOrder()
```

---

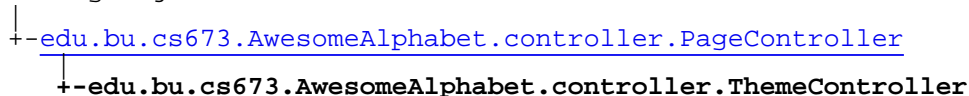
### **getMaxExamples**

```
public int getMaxExamples()
```

---

## edu.bu.cs673.AwesomeAlphabet.controller Class ThemeController

java.lang.Object



public class **ThemeController**  
extends [PageController](#)

This class defines the Themes Page Controller.

### Field Summary

private	<a href="#">m_themeMgr</a>
private	<a href="#">m_view</a>

Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[m\\_pageObserver](#)

### Constructor Summary

public	<a href="#">ThemeController</a> ( <a href="#">IPageObserver</a> pageObserver, <a href="#">IThemeControllerView</a> view, <a href="#">ThemeManager</a> themeMgr) Constructor
--------	--

### Method Summary

boolean	<a href="#">addTheme</a> (java.lang.String themeName) Adds a new theme.
boolean	<a href="#">changeThemeName</a> (java.lang.String oldThemeName, java.lang.String newThemeName) Changes the name of a theme.
boolean	<a href="#">deleteTheme</a> (java.lang.String themeName) Adds a new theme.
java.lang.String	<a href="#">getCurrentTheme</a> () Finds the current theme.
java.util.Iterator	<a href="#">getThemeNamesIterator</a> () Gets a theme name iterator.
void	<a href="#">GoToOptionsPage</a> ()
boolean	<a href="#">setCurrentTheme</a> (java.lang.String themeName) Sets the current theme.

Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[GoToPage](#)**Methods inherited from class** `java.lang.Object``clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait`

## Fields

**m\_view**

```
private edu.bu.cs673.AwesomeAlphabet.view.IThemeControllerView m_view
```

**m\_themeMgr**

```
private edu.bu.cs673.AwesomeAlphabet.model.ThemeManager m_themeMgr
```

## Constructors

**ThemeController**

```
public ThemeController(IPageObserver pageObserver,  
                       IThemeControllerView view,  
                       ThemeManager themeMgr)
```

Constructor

**Parameters:**

`pageObserver` - A page observer reference so that page transitions may be requested. For example, this may refer to the main window.

`view` - The view.

`themeMgr` - The Theme Manager.

## Methods

**getThemeNamesIterator**

```
public java.util.Iterator getThemeNamesIterator()
```

Gets a theme name iterator.

**Returns:**

A theme name iterator.

**addTheme**

```
public boolean addTheme(java.lang.String themeName)
```

Adds a new theme.

**Parameters:**

`themeName` - The name of the theme.

---

(continued from last page)

**Returns:**

True if theme was added. False if theme already exists or other error.

---

## deleteTheme

```
public boolean deleteTheme(java.lang.String themeName)
```

Adds a new theme.

**Parameters:**

themeName - The name of the theme.

**Returns:**

True if theme was added. False if theme already exists or other error.

---

## changeThemeName

```
public boolean changeThemeName(java.lang.String oldThemeName,  
    java.lang.String newThemeName)
```

Changes the name of a theme.

**Parameters:**

oldThemeName - Old theme name.

newThemeName - New theme name.

**Returns:**

True if theme name was changed.

---

## setCurrentTheme

```
public boolean setCurrentTheme(java.lang.String themeName)
```

Sets the current theme.

**Parameters:**

themeName - The theme name.

**Returns:**

True if current theme was set; false otherwise. This function will return false if themeName does not specify an existing theme and is not null.

---

## getCurrentTheme

```
public java.lang.String getCurrentTheme()
```

Finds the current theme.

**Returns:**

The name of the current theme, or a string indicating that no theme is currently active.

---

## GoToOptionsPage

```
public void GoToOptionsPage()
```

## edu.bu.cs673.AwesomeAlphabet.controller Class TitlePageController

```
java.lang.Object
  |
+- edu.bu.cs673.AwesomeAlphabet.controller.PageController
    |
    +- edu.bu.cs673.AwesomeAlphabet.controller.TitlePageController
```

public class **TitlePageController**  
extends [PageController](#)

This class defines the Title Page Controller.

Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[m\\_pageObserver](#)

### Constructor Summary

public	<a href="#">TitlePageController</a> ( <a href="#">IPageObserver</a> pageObserver, <a href="#">TitlePageView</a> view) Class constructor.
--------	---

### Method Summary

boolean	<a href="#">Options</a> ()
boolean	<a href="#">Start</a> () This method is used to notify the controller that the Start button was clicked.

Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[GoToPage](#)

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait`

## Constructors

### TitlePageController

```
public TitlePageController(IPageObserver pageObserver,
                           TitlePageView view)
```

Class constructor.

#### Parameters:

`pageObserver` - A page observer reference so that page transitions may be requested. For example, this may refer to the main window.  
`view` - The view.



(continued from last page)

## Methods

### Start

```
public boolean Start()
```

This method is used to notify the controller that the Start button was clicked. This will cause the Alphabet Page to be shown.

**Returns:**

True if Alphabet Page is able to be shown.

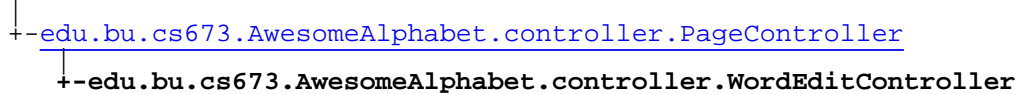
---

### Options

```
public boolean Options()
```

## edu.bu.cs673.AwesomeAlphabet.controller Class WordEditController

java.lang.Object



public class **WordEditController**  
extends [PageController](#)

### Field Summary

	<a href="#">m_model</a>
	<a href="#">m_themeMgr</a>
	<a href="#">m_view</a>

Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[m\\_pageObserver](#)

### Constructor Summary

public	<a href="#">WordEditController</a> ( <a href="#">IPageObserver</a> pageObserver, <a href="#">WordEditView</a> view, <a href="#">ThemeManager</a> themeMgr, <a href="#">Alphabet</a> model)
--------	--

### Method Summary

void	<a href="#">CancelEditWord</a> ()
java.lang.String	<a href="#">getAbsImageFilePath</a> (java.lang.String wordText)
java.lang.String	<a href="#">getAbsSoundFilePath</a> (java.lang.String wordText)
<a href="#">WordPictureSound</a>	<a href="#">getCurrentWordEditing</a> ()
int	<a href="#">getLetterIndex</a> (char wordLetter)
java.util.Iterator	<a href="#">getThemeNamesIterator</a> ()
void	<a href="#">SaveEditWord</a> (java.lang.String wordText, char associated_letter, java.lang.String imageFile, java.lang.String soundFile, java.lang.String themeName)
void	<a href="#">SaveNewWord</a> (java.lang.String wordText, char letter_c, java.lang.String imageFile, java.lang.String soundFile, java.lang.String themeName)

boolean

[wordExists](#)(java.lang.String sWord)

Returns true if word exists in the current model.

Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)[GoToPage](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### **m\_view**

edu.bu.cs673.AwesomeAlphabet.view.WordEditView **m\_view**

### **m\_themeMgr**

edu.bu.cs673.AwesomeAlphabet.model.ThemeManager **m\_themeMgr**

### **m\_model**

edu.bu.cs673.AwesomeAlphabet.model.Alphabet **m\_model**

## Constructors

### **WordEditController**

```
public WordEditController(IPageObserver pageObserver,
                          WordEditView view,
                          ThemeManager themeMgr,
                          Alphabet model)
```

## Methods

### **getThemeNamesIterator**

```
public java.util.Iterator getThemeNamesIterator()
```

### **CancelEditWord**

```
public void CancelEditWord()
```

## SaveNewWord

```
public void SaveNewWord(java.lang.String wordText,  
    char letter_c,  
    java.lang.String imageFile,  
    java.lang.String soundFile,  
    java.lang.String themeName)
```

---

## SaveEditWord

```
public void SaveEditWord(java.lang.String wordText,  
    char associated_letter,  
    java.lang.String imageFile,  
    java.lang.String soundFile,  
    java.lang.String themeName)
```

---

## getCurrentWordEditing

```
public WordPictureSound getCurrentWordEditing()
```

---

## getAbsSoundFilePath

```
public java.lang.String getAbsSoundFilePath(java.lang.String wordText)
```

---

## getAbsImageFilePath

```
public java.lang.String getAbsImageFilePath(java.lang.String wordText)
```

---

## getLetterIndex

```
public int getLetterIndex(char wordLetter)
```

---

## wordExists

```
public boolean wordExists(java.lang.String sWord)
```

Returns true if word exists in the current model.

### Parameters:

sWord - The name of the word.

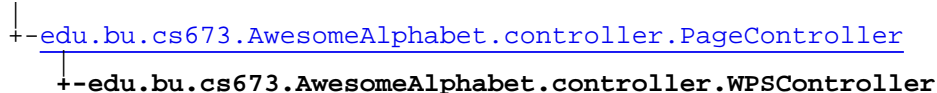
### Returns:

True if the word exists in the current model.

---

## edu.bu.cs673.AwesomeAlphabet.controller Class WPSController

java.lang.Object



public class **WPSController**  
extends [PageController](#)

### Field Summary

private	<a href="#">m_model</a>
private	<a href="#">m_view</a>

Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[m\\_pageObserver](#)

### Constructor Summary

public	<a href="#">WPSController</a> ( <a href="#">IPageObserver</a> pageObserver, <a href="#">WPSView</a> wpsView, <a href="#">Alphabet</a> model)
--------	--

### Method Summary

void	<a href="#">CreateNewWord</a> ()
void	<a href="#">DeleteWord</a> (java.lang.String word)
void	<a href="#">EditWord</a> (java.lang.String word)
java.util.Iterator	<a href="#">getWords</a> ()
void	<a href="#">GoToOptionsMenu</a> ()
void	<a href="#">SubstringSearch</a> (java.lang.String newText)

Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.controller.PageController](#)

[GoToPage](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### **m\_view**

```
private edu.bu.cs673.AwesomeAlphabet.view.WPSView m_view
```

---

### **m\_model**

```
private edu.bu.cs673.AwesomeAlphabet.model.Alphabet m_model
```

---

## Constructors

### **WPSController**

```
public WPSController(IPageObserver pageObserver,  
                    WPSView wpsView,  
                    Alphabet model)
```

---

## Methods

### **getWords**

```
public java.util.Iterator getWords()
```

---

### **GoToOptionsMenu**

```
public void GoToOptionsMenu()
```

---

### **CreateNewWord**

```
public void CreateNewWord()
```

---

### **EditWord**

```
public void EditWord(java.lang.String word)
```

---

### **DeleteWord**

```
public void DeleteWord(java.lang.String word)
```

---

---

## SubstringSearch

```
public void SubstringSearch(java.lang.String newText)
```

---

**Package**

**edu.bu.cs673.AwesomeAlphabet.main**



## edu.bu.cs673.AwesomeAlphabet.main Class AAConfig

java.lang.Object

└-edu.bu.cs673.AwesomeAlphabet.main.AAConfig

public class **AAConfig**  
extends java.lang.Object

### Field Summary

private static final	<a href="#">BASE_DIR</a> Value: <b>dir.location</b>
private static	<a href="#">baseDirName</a>
private static final	<a href="#">CONFIG_PROPS</a> Value: <b>config.properties</b>
private static	<a href="#">cwd</a>
private static final	<a href="#">DEFAULT_LETTERS</a> Value: <b>prop.letters</b>
private static final	<a href="#">GRAPHICS_DIR</a> Value: <b>dir.graphics</b>
private static	<a href="#">graphicsSubDir</a>
private static	<a href="#">letterProps</a>
private static	<a href="#">letterPropsName</a>
private static	<a href="#">letterPropsPersistent</a>
private static final	<a href="#">loader</a>
protected static	<a href="#">log</a>
private static final	<a href="#">PERSISTENT_RES</a> Value: <b>dir.persistent_resources</b>
private static	<a href="#">persistentResDir</a>
private static final	<a href="#">SOUNDS_DIR</a> Value: <b>dir.sounds</b>

private static	<a href="#">soundsSubDir</a>
----------------	------------------------------

## Constructor Summary

public	<a href="#">AAConfig()</a>
--------	----------------------------

## Method Summary

static int	<a href="#">addImageResource</a> (java.lang.String srcFileName, java.lang.String destFileName) Add a word image file.
static int	<a href="#">addSoundResource</a> (java.lang.String srcFileName, java.lang.String destFileName) Add a sound resource file.
static int	<a href="#">addWordToIndex</a> (char letter, java.lang.String wordText, java.lang.String Theme) Add a word to letter.properties file
static int	<a href="#">copy_dir</a> (java.lang.String srcDirName, java.lang.String destDirName) copy dir
static int	<a href="#">copy_file</a> (java.lang.String srcFileName, java.lang.String destFileName) copy file
static boolean	<a href="#">copy_res_to_file</a> (java.lang.String resource, java.lang.String destFileName, int type) copy resource to a file
static int	<a href="#">copy_stream</a> (java.io.InputStream inStream, java.io.OutputStream outStream)
static java.io.InputStream	<a href="#">getGraphicsResource</a> (java.lang.String filename)
static java.lang.String	<a href="#">getGraphicsResourceDir</a> ()
static java.lang.String	<a href="#">getGraphicsResourceDirAbs</a> ()
static java.lang.String	<a href="#">getGraphicsResourceDirPersistentAbs</a> ()
static java.io.InputStream	<a href="#">getGraphicsResourcePersistent</a> (java.lang.String filename)
static java.lang.String	<a href="#">getLetterPropFileName</a> ()
static java.util.Properties	<a href="#">getLetterProps</a> ()
static java.util.Properties	<a href="#">getLetterPropsPersistent</a> ()
static java.lang.String	<a href="#">getResourceDirAbs</a> ()

static java.lang.String	<a href="#"><u>getResourceDirPersistentAbs()</u></a>
static java.io.InputStream	<a href="#"><u>getSoundResource</u></a> (java.lang.String filename)
static java.lang.String	<a href="#"><u>getSoundResourceDir</u></a> ()
static java.lang.String	<a href="#"><u>getSoundResourceDirAbs</u></a> ()
static java.lang.String	<a href="#"><u>getSoundResourceDirPersistentAbs</u></a> ()
static java.io.InputStream	<a href="#"><u>getSoundResourcePersistent</u></a> (java.lang.String filename)
static int	<a href="#"><u>removeImageResource</u></a> (java.lang.String srcFileName) Remove a image resource file.
static int	<a href="#"><u>removeSoundResource</u></a> (java.lang.String srcFileName) Remove a sound resource file.
static int	<a href="#"><u>removeWordFromIndex</u></a> (char letter, java.lang.String wordText) Remove a word from letter.properties file

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### CONFIG\_PROPS

```
private static final java.lang.String CONFIG_PROPS
```

Constant value: **config.properties**

### BASE\_DIR

```
private static final java.lang.String BASE_DIR
```

Constant value: **dir.location**

### GRAPHICS\_DIR

```
private static final java.lang.String GRAPHICS_DIR
```

Constant value: **dir.graphics**

### SOUNDS\_DIR

```
private static final java.lang.String SOUNDS_DIR
```

(continued from last page)

Constant value: **dir.sounds**

---

## DEFAULT\_LETTERS

```
private static final java.lang.String DEFAULT_LETTERS
```

Constant value: **prop.letters**

---

## PERSISTENT\_RES

```
private static final java.lang.String PERSISTENT_RES
```

Constant value: **dir.persistent\_resources**

---

## loader

```
private static final java.lang.ClassLoader loader
```

---

## baseDirName

```
private static java.lang.String baseDirName
```

---

## graphicsSubDir

```
private static java.lang.String graphicsSubDir
```

---

## soundsSubDir

```
private static java.lang.String soundsSubDir
```

---

## letterPropsName

```
private static java.lang.String letterPropsName
```

---

## persistentResDir

```
private static java.lang.String persistentResDir
```

---

## cwd

```
private static java.lang.String cwd
```

---

---

## letterProps

```
private static java.util.Properties letterProps
```

---

## letterPropsPersistent

```
private static java.util.Properties letterPropsPersistent
```

---

## log

```
protected static org.apache.log4j.Logger log
```

---

## Constructors

### AAConfig

```
public AAConfig()
```

## Methods

### getLetterPropFileName

```
public static java.lang.String getLetterPropFileName()
```

---

### getGraphicsResource

```
public static java.io.InputStream getGraphicsResource(java.lang.String filename)
```

---

### getGraphicsResourcePersistent

```
public static java.io.InputStream getGraphicsResourcePersistent(java.lang.String filename)
```

---

### getResourceDirAbs

```
public static java.lang.String getResourceDirAbs()
```

---

### getResourceDirPersistentAbs

```
public static java.lang.String getResourceDirPersistentAbs()
```

---

(continued from last page)

---

## **getGraphicsResourceDir**

```
public static java.lang.String getGraphicsResourceDir()
```

---

## **getGraphicsResourceDirAbs**

```
public static java.lang.String getGraphicsResourceDirAbs()
```

---

## **getGraphicsResourceDirPersistentAbs**

```
public static java.lang.String getGraphicsResourceDirPersistentAbs()
```

---

## **getSoundResource**

```
public static java.io.InputStream getSoundResource(java.lang.String filename)
```

---

## **getSoundResourcePersistent**

```
public static java.io.InputStream getSoundResourcePersistent(java.lang.String filename)
```

---

## **getSoundResourceDir**

```
public static java.lang.String getSoundResourceDir()
```

---

## **getSoundResourceDirAbs**

```
public static java.lang.String getSoundResourceDirAbs()
```

---

## **getSoundResourceDirPersistentAbs**

```
public static java.lang.String getSoundResourceDirPersistentAbs()
```

---

## **getLetterProps**

```
public static java.util.Properties getLetterProps()
```

---

(continued from last page)

## getLetterPropsPersistent

```
public static java.util.Properties getLetterPropsPersistent()
```

---

## copy\_stream

```
public static int copy_stream(java.io.InputStream inStream,  
    java.io.OutputStream outStream)
```

---

## copy\_file

```
public static int copy_file(java.lang.String srcFileName,  
    java.lang.String destFileName)
```

copy file

### Parameters:

srcFileName: - Full path to source file

dstFileName: - Full path to dest file

---

## copy\_res\_to\_file

```
public static boolean copy_res_to_file(java.lang.String resource,  
    java.lang.String destFileName,  
    int type)
```

copy resource to a file

### Parameters:

resource: - resource file name

dstFileName: - Full path to dest file

type: - type of resource. 0 - top level dir resource, 1 - graphics resource, 2- sound resource

---

## copy\_dir

```
public static int copy_dir(java.lang.String srcDirName,  
    java.lang.String destDirName)
```

copy dir

### Parameters:

srcDirName: - Absolute path to source Dir

dstDirName: - Absolute path to dest Dir

---

## addSoundResource

```
public static int addSoundResource(java.lang.String srcFileName,  
    java.lang.String destFileName)
```

Add a sound resource file.

### Parameters:

srcFileName: - Full path name to source sound file

destFileName: - file name (no path info) of dest file

---

(continued from last page)

**Returns:**

---

## addImageResource

```
public static int addImageResource(java.lang.String srcFileName,  
    java.lang.String destFileName)
```

Add a word image file.

**Parameters:**

srcFileName: - Full path name to source image file  
destFileName: - file name (no path info) of dest file

**Returns:**

---

## addWordToIndex

```
public static int addWordToIndex(char letter,  
    java.lang.String wordText,  
    java.lang.String Theme)
```

Add a word to letter.properties file

**Parameters:**

srcFileName: - Full path name to source image file  
destFileName: - file name (no path info) of dest file

**Returns:**

---

## removeSoundResource

```
public static int removeSoundResource(java.lang.String srcFileName)
```

Remove a sound resource file.

**Parameters:**

srcFileName: - file name to source sound file (no dir)

**Returns:**

0 on success.

---

## removeImageResource

```
public static int removeImageResource(java.lang.String srcFileName)
```

Remove a image resource file.

**Parameters:**

srcFileName: - file name to source sound file (no dir)

**Returns:**

0 on success.

---



(continued from last page)

## **removeWordFromIndex**

```
public static int removeWordFromIndex(char letter,  
    java.lang.String wordText)
```

Remove a word from letter.properties file

### **Parameters:**

letter - char word belongs to  
actual - word string

### **Returns:**

0 on success

## edu.bu.cs673.AwesomeAlphabet.main Class AwesomeAlphabetApp

java.lang.Object

└-edu.bu.cs673.AwesomeAlphabet.main.AwesomeAlphabetApp

```
public class AwesomeAlphabetApp
    extends java.lang.Object
```

This class contains the application's main() method.

### Field Summary

static	<a href="#">log</a>
--------	---------------------

### Constructor Summary

public	<a href="#">AwesomeAlphabetApp()</a>
--------	--------------------------------------

### Method Summary

static void	<a href="#">main</a> (java.lang.String[] args)
-------------	--

Main entry point into the application.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

#### log

static org.apache.log4j.Logger **log**

### Constructors

#### AwesomeAlphabetApp

```
public AwesomeAlphabetApp()
```

### Methods

(continued from last page)

## **main**

```
public static void main(java.lang.String[] args)
```

Main entry point into the application. It is responsible for creating the models, views, controllers, and main window. In addition, it processes the resource file and causes the Title Page to be shown.

### **Parameters:**

args - Application arguments.

### **Throws:**

Exception

## edu.bu.cs673.AwesomeAlphabet.main Class Settings

java.lang.Object

└─edu.bu.cs673.AwesomeAlphabet.main.Settings

public class **Settings**  
extends java.lang.Object

### Field Summary

public static final	<a href="#">DISPLAY_ORDER</a> Value: <b>Display_Order</b>
private static final	<a href="#">loader</a>
private static	<a href="#">log</a>
public static final	<a href="#">MAXIMUM_EXAMPLES</a> Value: <b>Maximum_Examples</b>
public static	<a href="#">props</a>
private static final	<a href="#">SettingPropertiesFile</a> Value: <b>settings.properties</b>

### Constructor Summary

public	<a href="#">Settings()</a>
--------	----------------------------

### Method Summary

static java.lang.String	<a href="#">getDisplayOrder()</a> Return value of display order
static int	<a href="#">getMaxExamples()</a> Return value of display order
java.io.InputStream	<a href="#">loadFileFromClassPath()</a> Read the default setting properties from the classpath
java.io.InputStream	<a href="#">loadFileFromPersitentDirectory()</a> Read the user setting properties from the current directory
void	<a href="#">loadSettingProperties()</a> Read and load setting properties from the file
static void	<a href="#">saveSettingProperties()</a> Write setting properties to the files

---

**Methods inherited from class** `java.lang.Object``clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait`

---

**Fields****log**`private static org.apache.log4j.Logger log`

---

**DISPLAY\_ORDER**`public static final java.lang.String DISPLAY_ORDER`

Constant value: **Display\_Order**

---

**MAXIMUM\_EXAMPLES**`public static final java.lang.String MAXIMUM_EXAMPLES`

Constant value: **Maximum\_Examples**

---

**SettingPropertiesFile**`private static final java.lang.String SettingPropertiesFile`

Constant value: **settings.properties**

---

**loader**`private static final java.lang.ClassLoader loader`

---

**props**`public static java.util.Properties props`

---

**Constructors****Settings**`public Settings()`

---

**Methods**

(continued from last page)

## loadSettingProperties

```
public void loadSettingProperties()
```

Read and load setting properties from the file

---

## loadFileFromClassPath

```
private java.io.InputStream loadFileFromClassPath()
```

Read the default setting properties from the classpath

**Returns:**

---

## loadFileFromPersitentDirectory

```
private java.io.InputStream loadFileFromPersitentDirectory()
```

Read the user setting properties from the current directory

**Returns:**

---

## saveSettingProperties

```
public static void saveSettingProperties()
```

Write setting properties to the files

**Throws:**

IOException

FileNotFoundException

---

## getDisplayOrder

```
public static java.lang.String getDisplayOrder()
```

Return value of display order

---

## getMaxExamples

```
public static int getMaxExamples()
```

Return value of display order

---

---

**Package**

**edu.bu.cs673.AwesomeAlphabet.model**

## edu.bu.cs673.AwesomeAlphabet.model Class Alphabet

```

java.lang.Object
  |
  +- java.util.Observable
      |
      +- edu.bu.cs673.AwesomeAlphabet.model.Alphabet
  
```

```

public class Alphabet
extends java.util.Observable
  
```

The class defines the Alphabet model. It creates and maintains references to 26 Letter objects; one for each letter of the English alphabet. It also maintains the current letter selection and has methods for getting the Letter object, changing the letter selection, and loading resources.

### Field Summary

protected static final	<a href="#">AA_ALPHABET_SIZE</a> Value: <b>26</b>
protected static	<a href="#">log</a>
private	<a href="#">m_alphabetsong</a>
private	<a href="#">m_currentWordEditing</a>
private	<a href="#">m_db</a>
public	<a href="#">m_iCurLetterIndex</a>
private	<a href="#">m_letters</a>
private	<a href="#">m_themeMgr</a>
private	<a href="#">m_word_cache</a>

### Fields inherited from class java.util.Observable

changed, obs

### Constructor Summary

public	<a href="#">Alphabet</a> ( <a href="#">ThemeManager</a> themeMgr) Class constructor.
--------	---

### Method Summary



int	<a href="#">addNewWord</a> (java.lang.String wordText, char associatedLetter, java.lang.String imageName, java.lang.String soundName, java.lang.String themeName) Add a new word
boolean	<a href="#">createLoadPersistentResourceDir</a> (java.util.Properties prop)
int	<a href="#">deleteWord</a> (java.lang.String wordText) delete a word
int	<a href="#">editWord</a> (java.lang.String wordText, char associated_letter, java.lang.String imageName, java.lang.String soundName, java.lang.String themeName) Edit properties of existing word
java.lang.String	<a href="#">getAbsImageFilePath</a> (java.lang.String wordText)
java.lang.String	<a href="#">getAbsSoundFilePath</a> (java.lang.String wordText)
<a href="#">Letter</a>	<a href="#">GetCurrentLetter</a> () Returns the selected letter.
<a href="#">WordPictureSound</a>	<a href="#">getCurrentWordEditing</a> ()
java.util.Iterator	<a href="#">GetIterator</a> () Gets an iterator to the list of Letter objects.
int	<a href="#">GetLetterIndex</a> (char c) Gets the array index of the specified letter.
java.util.Iterator	<a href="#">GetWordCacheIterator</a> () Gets an iterator to the list of String objects.
java.util.Iterator	<a href="#">GetWordCacheIterator</a> (java.lang.String regex)
<a href="#">WordPictureSound</a>	<a href="#">getWordPictureSound</a> (java.lang.String word) Given a word, return matching WordPictureSound object, if it exists.
java.util.Iterator	<a href="#">getWords</a> ()
java.util.Iterator	<a href="#">getWords</a> (java.lang.String prefix)
<a href="#">Letter</a>	<a href="#">GoToNextLetter</a> () Changes letter selection to next letter.
<a href="#">Letter</a>	<a href="#">GoToPreviousLetter</a> () Changes letter selection to previous letter.
void	<a href="#">Initialize</a> () Responsible for creating the Letter objects.
void	<a href="#">loadAlphabetSong</a> (java.util.Properties prop)
void	<a href="#">LoadDatabase</a> ()
void	<a href="#">loadLetterPhonicSound</a> (char letter_c)

void	<a href="#"><code>loadLettersFromDatabase</code></a> (char letter_c)
void	<a href="#"><code>loadLetterSound</code></a> (char letter_c)
void	<a href="#"><code>LoadResources</code></a> (java.util.Properties prop) Loads word, picture, and sound resources into Letter objects.
void	<a href="#"><code>PlayAlphabetSong</code></a> ()
void	<a href="#"><code>resetListOrder</code></a> () Sort the examples for each letter by id
<a href="#"><code>Letter</code></a>	<a href="#"><code>setCurrentLetter</code></a> ( <a href="#"><code>Letter</code></a> letter) Sets the letter selection.
boolean	<a href="#"><code>setCurrentWordEditing</code></a> (java.lang.String wordText)
void	<a href="#"><code>shuffleExamples</code></a> () Shuffle the examples for each letter
void	<a href="#"><code>sortLetterExamples</code></a> () Sort the examples for each letter by word
void	<a href="#"><code>StopAlphabetSound</code></a> ()
boolean	<a href="#"><code>unsetCurrentWordEditing</code></a> ()

**Methods inherited from class** java.util.Observable

addObserver, clearChanged, countObservers, deleteObserver, deleteObservers, hasChanged, notifyObservers, notifyObservers, setChanged

**Methods inherited from class** java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### AA\_ALPHABET\_SIZE

protected static final int **AA\_ALPHABET\_SIZE**

Constant value: **26**

### log

protected static org.apache.log4j.Logger **log**

(continued from last page)

---

## **m\_letters**

```
private edu.bu.cs673.AwesomeAlphabet.model.Letter m_letters
```

---

## **m\_iCurLetterIndex**

```
public int m_iCurLetterIndex
```

---

## **m\_alphabetsong**

```
private edu.bu.cs673.AwesomeAlphabet.model.GameSound m_alphabetsong
```

---

## **m\_themeMgr**

```
private edu.bu.cs673.AwesomeAlphabet.model.ThemeManager m_themeMgr
```

---

## **m\_db**

```
private edu.bu.cs673.AwesomeAlphabet.model.Database m_db
```

---

## **m\_currentWordEditing**

```
private java.lang.String m_currentWordEditing
```

---

## **m\_word\_cache**

```
private java.util.List m_word_cache
```

---

# Constructors

## **Alphabet**

```
public Alphabet(ThemeManager themeMgr)
```

Class constructor.

# Methods

## **Initialize**

```
public void Initialize()
```

Responsible for creating the Letter objects.

---

## GetLetterIndex

```
public int GetLetterIndex(char c)
```

Gets the array index of the specified letter.

**Parameters:**

c - The letter, represented as a char.

**Returns:**

The index of the letter in the array or -1 if the letter is invalid.

---

## GetIterator

```
public java.util.Iterator GetIterator()
```

Gets an iterator to the list of Letter objects.

**Returns:**

An iterator to the list of Letter objects.

---

## getWords

```
public java.util.Iterator getWords(java.lang.String prefix)
```

---

## getWords

```
public java.util.Iterator getWords()
```

---

## GetWordCacheIterator

```
public java.util.Iterator GetWordCacheIterator()
```

Gets an iterator to the list of String objects.

**Returns:**

An iterator to the list of String objects.

---

## GetWordCacheIterator

```
public java.util.Iterator GetWordCacheIterator(java.lang.String regex)
```

---

## SetCurrentLetter

```
public Letter SetCurrentLetter(Letter letter)
```

Sets the letter selection.

**Parameters:**

letter - The letter to select.

---

(continued from last page)

**Returns:**

A reference to the newly selected Letter object or null if the selection was invalid.

---

**GetCurrentLetter**

```
public Letter GetCurrentLetter()
```

Returns the selected letter.

**Returns:**

The selected letter.

---

**GoToPreviousLetter**

```
public Letter GoToPreviousLetter()
```

Changes letter selection to previous letter.

**Returns:**

The previous letter object or null if there are no previous letters.

---

**GoToNextLetter**

```
public Letter GoToNextLetter()
```

Changes letter selection to next letter.

**Returns:**

The next letter object or null if there are no more letters.

---

**createLoadPersistentResourceDir**

```
public boolean createLoadPersistentResourceDir(java.util.Properties prop)
```

---

**LoadDatabase**

```
private void LoadDatabase()
```

---

**loadLetterSound**

```
private void loadLetterSound(char letter_c)
```

---

**loadLetterPhonicSound**

```
private void loadLetterPhonicSound(char letter_c)
```

---

**loadAlphabetSong**

```
private void loadAlphabetSong(java.util.Properties prop)
```

---

(continued from last page)

---

## loadLettersFromDatabase

```
private void loadLettersFromDatabase(char letter_c)
```

---

## LoadResources

```
public void LoadResources(java.util.Properties prop)
```

Loads word, picture, and sound resources into Letter objects.

**Parameters:**

prop - The property list containing resource information.

---

## PlayAlphabetSong

```
public void PlayAlphabetSong()
```

---

## StopAlphabetSound

```
public void StopAlphabetSound()
```

---

## getWordPictureSound

```
public WordPictureSound getWordPictureSound(java.lang.String word)
```

Given a word, return matching WordPictureSound object, if it exists.

**Parameters:**

word - to match

**Returns:**

WordPictureSound object associated with word

---

## addNewWord

```
public int addNewWord(java.lang.String wordText,  
    char associatedLetter,  
    java.lang.String imageName,  
    java.lang.String soundName,  
    java.lang.String themeName)
```

Add a new word

**Returns:**

0 on success. Failure otherwise.

---

## shuffleExamples

```
public void shuffleExamples()
```

Shuffle the examples for each letter

---

## sortLetterExamples

```
public void sortLetterExamples()
```

Sort the examples for each letter by word

---

## resetListOrder

```
public void resetListOrder()
```

Sort the examples for each letter by id

---

## deleteWord

```
public int deleteWord(java.lang.String wordText)
```

delete a word

**Returns:**

0 on success. Failure otherwise.

---

## editWord

```
public int editWord(java.lang.String wordText,  
                    char associated_letter,  
                    java.lang.String imageName,  
                    java.lang.String soundName,  
                    java.lang.String themeName)
```

Edit properties of existing word

**Returns:**

0 on success. Failure otherwise.

---

## setCurrentWordEditing

```
public boolean setCurrentWordEditing(java.lang.String wordText)
```

---

## unsetCurrentWordEditing

```
public boolean unsetCurrentWordEditing()
```

---

## getCurrentWordEditing

```
public WordPictureSound getCurrentWordEditing()
```

---

## getAbsImagePath

```
public java.lang.String getAbsImagePath(java.lang.String wordText)
```

---

---

## **getAbsSoundFilePath**

```
public java.lang.String getAbsSoundFilePath(java.lang.String wordText)
```



## edu.bu.cs673.AwesomeAlphabet.model Class Database

java.lang.Object

└-edu.bu.cs673.AwesomeAlphabet.model.Database

public class **Database**  
extends java.lang.Object

This class defines the interface to the database. It uses the singleton pattern.

### Nested Class Summary

class	<a href="#">Database.WordData</a> Database.WordData
-------	--

### Field Summary

protected static	<a href="#">log</a>
private	<a href="#">m_con</a>
private static	<a href="#">m_db</a>
private final	<a href="#">m_iDefThemeId</a> Value: 0

### Constructor Summary

private	<a href="#">Database()</a> Constructor.
---------	--

### Method Summary

boolean	<a href="#">addTheme</a> (java.lang.String themeName) Adds a theme to the database.
boolean	<a href="#">addWord</a> (java.lang.String wordName, java.lang.String picturePath, java.lang.String soundPath, char letter, java.lang.String themeName) Adds a Word entry to the database.
boolean	<a href="#">changeThemeName</a> (java.lang.String oldThemeName, java.lang.String newThemeName) Changes the name of the theme in the database.
boolean	<a href="#">changeWordData</a> (java.lang.String oldWordName, java.lang.String newWordName, java.lang.String picturePath, java.lang.String soundPath, char letter, java.lang.String themeName) Modified all fields of an existing word

boolean	<a href="#"><code>changeWordName</code></a> (java.lang.String oldWordName, java.lang.String newWordName) Changes the name of an existing word in the Word table.
void	<a href="#"><code>createDefaultTheme</code></a> () Creates the default theme record if it does not already exist.
void	<a href="#"><code>createTables</code></a> () Creates all tables in the database if they do not already exist.
boolean	<a href="#"><code>deleteTheme</code></a> (java.lang.String themeName) Deletes a theme from the database.
boolean	<a href="#"><code>deleteWord</code></a> (java.lang.String wordName) Deletes a word from the Word table
void	<a href="#"><code>EnableForeignKeySupport</code></a> () Enables database foreign key support
static <a href="#"><code>Database</code></a>	<a href="#"><code>getDatabaseInstance</code></a> () This method returns the Database object.
int	<a href="#"><code>getNumberRowsWordTable</code></a> ()
int	<a href="#"><code>getThemeId</code></a> (java.lang.String themeName) Gets the theme ID based on the theme name.
java.lang.String	<a href="#"><code>getThemeName</code></a> (int themeId) Gets the theme name based on the theme ID.
java.util.Iterator	<a href="#"><code>getThemes</code></a> () Retrieves an iterator to a collection of theme names that are stored in the database.
java.util.Iterator	<a href="#"><code>getWordData</code></a> (char letter) Retrieves an iterator to a collection of row data retrieved form the Word table based on the specified letter.
java.util.Iterator	<a href="#"><code>getWordNames</code></a> () Retrieves an iterator to a collection of word names that are stored in the database.
int	<a href="#"><code>hasTheme</code></a> (java.lang.String themeName) Checks if theme exists in database.
int	<a href="#"><code>hasWord</code></a> (java.lang.String wordName) Checks if word exists in database
void	<a href="#"><code>initializeDatabase</code></a> () Initializes the database.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

(continued from last page)

---

## m\_db

```
private static edu.bu.cs673.AwesomeAlphabet.model.Database m_db
```

---

## log

```
protected static org.apache.log4j.Logger log
```

---

## m\_con

```
private java.sql.Connection m_con
```

---

## m\_iDefThemeId

```
private final int m_iDefThemeId
```

Constant value: 0

## Constructors

### Database

```
private Database()
```

Constructor.

## Methods

### getDatabaseInstance

```
public static Database getDatabaseInstance()
```

This method returns the Database object. It is part of the singleton pattern.

**Returns:**

The database object.

---

### initializeDatabase

```
private void initializeDatabase()
```

Initializes the database. As part of this, all tables and default records are created if they do not already exist.

---

### createTables

```
private void createTables()
```

Creates all tables in the database if they do not already exist.

---

(continued from last page)

---

## createDefaultTheme

```
private void createDefaultTheme()
```

Creates the default theme record if it does not already exist.

---

## EnableForeignKeySupport

```
private void EnableForeignKeySupport()
```

Enables database foreign key support

---

## addTheme

```
public boolean addTheme(java.lang.String themeName)
```

Adds a theme to the database.

**Parameters:**

themeName - The name of the theme.

**Returns:**

True if theme was added successfully; false otherwise.

---

## deleteTheme

```
public boolean deleteTheme(java.lang.String themeName)
```

Deletes a theme from the database.

**Parameters:**

themeName - The theme name.

**Returns:**

True if theme was successfully deleted; false otherwise.

---

## changeThemeName

```
public boolean changeThemeName(java.lang.String oldThemeName,  
    java.lang.String newThemeName)
```

Changes the name of the theme in the database.

**Parameters:**

oldThemeName - Old theme name.

newThemeName - New theme name.

**Returns:**

True if theme name was changed successfully; otherwise false.

---

## hasTheme

```
public int hasTheme(java.lang.String themeName)
```

Checks if theme exists in database.

**Parameters:**

themeName

---

(continued from last page)

**Returns:**1 = Theme exists, 0 = Theme does not exist, -1 = Error querying database

---

**getThemeId**

```
private int getThemeId(java.lang.String themeName)
```

Gets the theme ID based on the theme name.

**Parameters:**

themeName - Theme Name

**Returns:**>=0: Theme ID <0: Error querying database

---

**getThemeName**

```
private java.lang.String getThemeName(int themeId)
```

Gets the theme name based on the theme ID.

**Parameters:**

themeID - Theme ID

**Returns:**The theme name or null if theme ID not found.

---

**getThemes**

```
public java.util.Iterator getThemes()
```

Retrieves an iterator to a collection of theme names that are stored in the database.

**Returns:**The iterator.

---

**getNumberRowsWordTable**

```
public int getNumberRowsWordTable()
```

---

**addWord**

```
public boolean addWord(java.lang.String wordName,  
    java.lang.String picturePath,  
    java.lang.String soundPath,  
    char letter,  
    java.lang.String themeName)
```

Adds a Word entry to the database.

**Parameters:**

word - The word.

picturePath - The picture path name.

soundPath - The sound path name

letter - The letter that the word is associated with.

themeName - The theme name. Must be a valid theme name.

---

(continued from last page)

**Returns:**

---

**deleteWord**

```
public boolean deleteWord(java.lang.String wordName)
```

Deletes a word from the Word table

**Parameters:**

wordName - Word Name

**Returns:**

True if word was deleted successfully; otherwise false.

---

**changeWordName**

```
public boolean changeWordName(java.lang.String oldWordName,  
    java.lang.String newWordName)
```

Changes the name of an existing word in the Word table.

**Parameters:**

oldWordName - Old word name.

newWordName - New word name.

**Returns:**

True if word name was changed successfully; otherwise false.

---

**changeWordData**

```
public boolean changeWordData(java.lang.String oldWordName,  
    java.lang.String newWordName,  
    java.lang.String picturePath,  
    java.lang.String soundPath,  
    char letter,  
    java.lang.String themeName)
```

Modified all fields of an existing word

**Parameters:**

oldWordName - Old word name

newWordName - New word name. May be the same as the old word name

picturePath - The picture path name.

soundPath - The sound path name.

letter - The letter word is associated with.

themeName - The theme name.

**Returns:**

True if word data was changed successfully; otherwise false.

---

**hasWord**

```
public int hasWord(java.lang.String wordName)
```

Checks if word exists in database

**Parameters:**

word - The word.

---

(continued from last page)

**Returns:**

1 = Word exists, 0 = Word does not exist, -1 = Error querying database

---

**getWordNames**

```
public java.util.Iterator getWordNames()
```

Retrieves an iterator to a collection of word names that are stored in the database.

**Returns:**

The iterator.

---

**getWordData**

```
public java.util.Iterator getWordData(char letter)
```

Retrieves an iterator to a collection of row data retrieved form the Word table based on the specified letter.

**Returns:**

The iterator.

## edu.bu.cs673.AwesomeAlphabet.model Class Database.WordData

java.lang.Object

└─ edu.bu.cs673.AwesomeAlphabet.model.Database.WordData

```
public class Database.WordData
extends java.lang.Object
```

Inner class that defines row data returned by Word query functions.

### Field Summary

public	<a href="#">letter</a>
public	<a href="#">picturePath</a>
public	<a href="#">soundPath</a>
public	<a href="#">theme</a>
public	<a href="#">word</a>

### Constructor Summary

public	<a href="#">Database.WordData</a> (java.lang.String word, java.lang.String picturePath, java.lang.String soundPath, char letter, java.lang.String theme)
--------	--

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

#### word

```
public java.lang.String word
```

#### picturePath

```
public java.lang.String picturePath
```



(continued from last page)

**soundPath**

```
public java.lang.String soundPath
```

---

**letter**

```
public char letter
```

---

**theme**

```
public java.lang.String theme
```

---

## Constructors

**Database.WordData**

```
public Database.WordData(java.lang.String word,  
                        java.lang.String picturePath,  
                        java.lang.String soundPath,  
                        char letter,  
                        java.lang.String theme)
```

# edu.bu.cs673.AwesomeAlphabet.model

## Class GameImage

java.lang.Object

└-edu.bu.cs673.AwesomeAlphabet.model.GameImage

public class **GameImage**  
extends java.lang.Object

### Field Summary

static	<a href="#">log</a>
--------	---------------------

### Constructor Summary

public	<a href="#">GameImage()</a>
--------	-----------------------------

### Method Summary

static java.awt.Image	<a href="#">getImage</a> (java.lang.String filename) Constructor.
-----------------------	--

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

#### log

static org.apache.log4j.Logger **log**

### Constructors

#### GameImage

public **GameImage**()

### Methods

#### getImage

public static java.awt.Image **getImage**(java.lang.String filename)

(continued from last page)

Constructor. This prepends the resource directory to the image's filename.

**Parameters:**

filename - the name of the file containing the graphics

**Returns:**

an Image containing the graphic read from the jar or the file system.

# edu.bu.cs673.AwesomeAlphabet.model

## Class GameSound

java.lang.Object

└-edu.bu.cs673.AwesomeAlphabet.model.GameSound

```
public class GameSound
    extends java.lang.Object
```

### Field Summary

private	<a href="#">curr_clip</a>
static	<a href="#">log</a>
private	<a href="#">soundName</a>

### Constructor Summary

public	<a href="#">GameSound</a> (java.lang.String soundName) Constructor.
--------	--

### Method Summary

void	<a href="#">PlaySound</a> () Plays the sound for this object.
void	<a href="#">StopSound</a> ()

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Fields

### log

static org.apache.log4j.Logger **log**

### soundName

private java.lang.String **soundName**

**curr\_clip**

```
private javax.sound.sampled.Clip curr_clip
```

## Constructors

**GameSound**

```
public GameSound(java.lang.String soundName)
```

Constructor. This prepends the directory to the sound's filename

**Parameters:**

soundName - sound's filename

## Methods

**PlaySound**

```
public void PlaySound()
```

Plays the sound for this object. Does not return until the sound has finished playing. This is limited to short-duration sounds only (2 seconds or less).

**StopSound**

```
public void StopSound()
```

# edu.bu.cs673.AwesomeAlphabet.model

## Class GuiUpdate

```

java.lang.Object
  |
  +- java.util.Observable
        |
        +- edu.bu.cs673.AwesomeAlphabet.model.GuiUpdate
  
```

class **GuiUpdate**  
 extends java.util.Observable

### Fields inherited from class java.util.Observable

changed, obs

## Constructor Summary

[GuiUpdate\(\)](#)

## Method Summary

void [forceUpdate\(\)](#)

### Methods inherited from class java.util.Observable

addObserver, clearChanged, countObservers, deleteObserver, deleteObservers, hasChanged, notifyObservers, notifyObservers, setChanged

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

## Constructors

### GuiUpdate

**GuiUpdate()**

## Methods

### forceUpdate

public void **forceUpdate()**

## edu.bu.cs673.AwesomeAlphabet.model Class Letter

```

java.lang.Object
  |
  +- java.util.Observable
        |
        +- edu.bu.cs673.AwesomeAlphabet.model.Letter

```

### All Implemented Interfaces:

```
java.util.Observer
```

```

public class Letter
extends java.util.Observable
implements java.util.Observer

```

This class defines the Letter model. It represents a letter in the English alphabet and contains a list words that start with the letter, along with associated pictures and sounds. An index is maintained for the word list to keep track of which word was most recently shown.

### Nested Class Summary

class	<a href="#">Letter.Sound_Type</a> Letter.Sound_Type
-------	--

### Field Summary

private	<a href="#">curr_sound</a>
protected static	<a href="#">log</a>
private	<a href="#">m_cLetter</a>
private	<a href="#">m_index</a>
private	<a href="#">m_LetterSound</a>
private	<a href="#">m_PhonicSound</a>
private	<a href="#">m_themeMgr</a>
private	<a href="#">m_wps</a>

### Fields inherited from class java.util.Observable

changed, obs

### Constructor Summary

public	<a href="#">Letter</a> (char cLetter, <a href="#">ThemeManager</a> themeMgr) Class constructor.
--------	--

## Method Summary

void	<a href="#"><code>addLetterSoundResource</code></a> (java.lang.String soundName)
void	<a href="#"><code>addPhonicSoundResource</code></a> (java.lang.String soundName)
void	<a href="#"><code>addResource</code></a> (java.lang.String imageName, java.lang.String soundName, java.lang.String wordText, <a href="#"><code>Theme</code></a> theme) Adds a new WordPictureSound object to the word list.
javax.swing.Icon	<a href="#"><code>getIcon</code></a> (int width, int height) Gets the image associated with the current word.
java.util.ListIterator	<a href="#"><code>GetIterator</code></a> () Gets an iterator to the list of WordPictureSound objects.
char	<a href="#"><code>GetLetterAsChar</code></a> () Gets the lower-case letter that this object represents.
char	<a href="#"><code>GetUppercaseLetter</code></a> () Gets the upper-case letter that this object represents.
java.lang.String	<a href="#"><code>getWord</code></a> () Gets the text of the current word.
<a href="#"><code>WordPictureSound</code></a>	<a href="#"><code>getWPSData</code></a> (int index) Gets the WordPictureSound object stored at the provided list index.
void	<a href="#"><code>nextExample</code></a> () Advances the word list index and notifies observers that the word example has changed.
void	<a href="#"><code>playSound</code></a> () Plays the sound associated with the current word.
void	<a href="#"><code>playSoundLetter</code></a> ()
void	<a href="#"><code>playSoundPhonic</code></a> ()
void	<a href="#"><code>removeAllEntries</code></a> ()
int	<a href="#"><code>removeResource</code></a> ( <a href="#"><code>WordPictureSound</code></a> wps)
void	<a href="#"><code>resetDeafultOrder</code></a> () Reset default order of the examples as they were initially loaded/added to the list
void	<a href="#"><code>shuffleList</code></a> () Shuffle the list of words/examples
void	<a href="#"><code>sortList</code></a> () Sort the list of words/examples
void	<a href="#"><code>stopSound</code></a> ()
void	<a href="#"><code>update</code></a> (java.util.Observable o, java.lang.Object arg)



**Methods inherited from class** `java.util.Observable`

`addObserver`, `clearChanged`, `countObservers`, `deleteObserver`, `deleteObservers`, `hasChanged`, `notifyObservers`, `notifyObservers`, `setChanged`

**Methods inherited from class** `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `registerNatives`, `toString`, `wait`, `wait`, `wait`

**Methods inherited from interface** `java.util.Observer`

`update`

---

## Fields

**m\_cLetter**

`private char m_cLetter`

---

**m\_wps**

`private java.util.List m_wps`

---

**m\_LetterSound**

`private edu.bu.cs673.AwesomeAlphabet.model.GameSound m_LetterSound`

---

**m\_PhonicSound**

`private edu.bu.cs673.AwesomeAlphabet.model.GameSound m_PhonicSound`

---

**m\_index**

`private int m_index`

---

**curr\_sound**

`private edu.bu.cs673.AwesomeAlphabet.model.Letter.Sound_Type curr_sound`

---

**m\_themeMgr**

`private edu.bu.cs673.AwesomeAlphabet.model.ThemeManager m_themeMgr`

---

(continued from last page)

---

## log

```
protected static org.apache.log4j.Logger log
```

---

## Constructors

### Letter

```
public Letter(char cLetter,  
             ThemeManager themeMgr)
```

Class constructor.

**Parameters:**

cLetter - The letter that this object will represent.

---

## Methods

### GetLetterAsChar

```
public char GetLetterAsChar()
```

Gets the lower-case letter that this object represents.

**Returns:**

The letter as a char.

---

### GetUppercaseLetter

```
public char GetUppercaseLetter()
```

Gets the upper-case letter that this object represents.

**Returns:**

The letter as a char.

---

### GetIterator

```
public java.util.ListIterator GetIterator()
```

Gets an iterator to the list of WordPictureSound objects.

**Returns:**

An iterator to the list of WordPictureSound objects.

---

### addResource

```
public void addResource(java.lang.String imageName,  
                        java.lang.String soundName,  
                        java.lang.String wordText,  
                        Theme theme)
```

Adds a new WordPictureSound object to the word list.

**Parameters:**

(continued from last page)

imageName - The filename of the word image.  
soundName - The filename of the word sound.  
wordText - The text of the word.  
theme - The theme.

---

## addLetterSoundResource

```
public void addLetterSoundResource(java.lang.String soundName)
```

---

## addPhonicSoundResource

```
public void addPhonicSoundResource(java.lang.String soundName)
```

---

## getWPSData

```
private WordPictureSound getWPSData(int index)
```

Gets the WordPictureSound object stored at the provided list index.

**Parameters:**

index - The index into the list.

**Returns:**

The WordPictureSound object or null if the index is invalid.

---

## getWord

```
public java.lang.String getWord()
```

Gets the text of the current word.

**Returns:**

The text of the current word as a String.

---

## getIcon

```
public javax.swing.Icon getIcon(int width,  
int height)
```

Gets the image associated with the current word.

**Returns:**

The image associated with the current word.

---

## nextExample

```
public void nextExample()
```

Advances the word list index and notifies observers that the word example has changed.

---

## playSound

```
public void playSound()
```

---

(continued from last page)

Plays the sound associated with the current word.

---

## **playSoundLetter**

```
public void playSoundLetter()
```

---

## **playSoundPhonic**

```
public void playSoundPhonic()
```

---

## **stopSound**

```
public void stopSound()
```

---

## **removeAllEntries**

```
public void removeAllEntries()
```

---

## **update**

```
public void update(java.util.Observable o,  
                  java.lang.Object arg)
```

---

## **removeResource**

```
public int removeResource(WordPictureSound wps)
```

---

## **shuffleList**

```
public void shuffleList()
```

Shuffle the list of words/examples

---

## **sortList**

```
public void sortList()
```

Sort the list of words/examples

---

## **resetDeafultOrder**

```
public void resetDeafultOrder()
```

Reset default order of the examples as they were initially loaded/added to the list

---

## edu.bu.cs673.AwesomeAlphabet.model Class Letter.Sound\_Type

java.lang.Object

└─ java.lang.Enum

└─ edu.bu.cs673.AwesomeAlphabet.model.Letter.Sound\_Type

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

private static final class **Letter.Sound\_Type**

extends java.lang.Enum

### Field Summary

public static final	<a href="#">LETTER</a>
public static final	<a href="#">NONE</a>
public static final	<a href="#">PHONIC</a>
public static final	<a href="#">WPS</a>

Fields inherited from class java.lang.Enum

name, ordinal

### Constructor Summary

private	<a href="#">Letter.Sound_Type()</a>
---------	-------------------------------------

### Method Summary

static <a href="#">Letter.Sound_Type</a>	<a href="#">valueOf</a> (java.lang.String name)
static <a href="#">Letter.Sound_Type[]</a>	<a href="#">values</a> ()

Methods inherited from class java.lang.Enum

clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, readObject, readObjectNoData, toString, valueOf

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

Methods inherited from interface java.lang.Comparable

```
compareTo
```

---

## Fields

### NONE

```
public static final edu.bu.cs673.AwesomeAlphabet.model.Letter.Sound_Type NONE
```

---

### WPS

```
public static final edu.bu.cs673.AwesomeAlphabet.model.Letter.Sound_Type WPS
```

---

### LETTER

```
public static final edu.bu.cs673.AwesomeAlphabet.model.Letter.Sound_Type LETTER
```

---

### PHONIC

```
public static final edu.bu.cs673.AwesomeAlphabet.model.Letter.Sound_Type PHONIC
```

---

## Constructors

### Letter.Sound\_Type

```
private Letter.Sound_Type()
```

## Methods

### values

```
public static Letter.Sound\_Type\[\] values()
```

---

### valueOf

```
public static Letter.Sound\_Type valueOf(java.lang.String name)
```

# edu.bu.cs673.AwesomeAlphabet.model

## Class PageName



All Implemented Interfaces:  
java.io.Serializable, java.lang.Comparable

```
public final class PageName
extends java.lang.Enum
```

This enumeration defines the page names for each view. To get the name as a String, use the toString() method.

Field Summary	
public static final	<a href="#">AlphabetPage</a>
public static final	<a href="#">LetterPage</a>
public static final	<a href="#">OptionsPage</a>
public static final	<a href="#">SettingPage</a>
private final	<a href="#">stringValue</a>
public static final	<a href="#">ThemePage</a>
public static final	<a href="#">TitlePage</a>
public static final	<a href="#">WordEditPage</a>
public static final	<a href="#">WPSPage</a>

Fields inherited from class java.lang.Enum	
name, ordinal	

Constructor Summary	
private	<a href="#">PageName</a> ( java.lang.String s )

Method Summary	
java.lang.String	<a href="#">toString</a> ( )

static <a href="#">PageName</a>	<a href="#">valueOf</a> (java.lang.String name)
static <a href="#">PageName[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, readObject, readObjectNoData, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

## Fields

### TitlePage

```
public static final edu.bu.cs673.AwesomeAlphabet.model.PageName TitlePage
```

### AlphabetPage

```
public static final edu.bu.cs673.AwesomeAlphabet.model.PageName AlphabetPage
```

### LetterPage

```
public static final edu.bu.cs673.AwesomeAlphabet.model.PageName LetterPage
```

### OptionsPage

```
public static final edu.bu.cs673.AwesomeAlphabet.model.PageName OptionsPage
```

### ThemePage

```
public static final edu.bu.cs673.AwesomeAlphabet.model.PageName ThemePage
```

### WPSPage

```
public static final edu.bu.cs673.AwesomeAlphabet.model.PageName WPSPage
```



(continued from last page)

---

## SettingPage

```
public static final edu.bu.cs673.AwesomeAlphabet.model.PageName SettingPage
```

---

## WordEditPage

```
public static final edu.bu.cs673.AwesomeAlphabet.model.PageName WordEditPage
```

---

## stringValue

```
private final java.lang.String stringValue
```

## Constructors

### PageName

```
private PageName(java.lang.String s)
```

## Methods

### values

```
public static PageName[] values()
```

---

### valueOf

```
public static PageName valueOf(java.lang.String name)
```

---

### toString

```
public java.lang.String toString()
```

## edu.bu.cs673.AwesomeAlphabet.model Class Theme

```

java.lang.Object
  |
+- java.util.Observable
    |
+- edu.bu.cs673.AwesomeAlphabet.model.Theme

```

```

public class Theme
extends java.util.Observable

```

### Field Summary

public static	<a href="#">ALL_THEMES</a>
public static	<a href="#">DEFAULT_THEME_NAME</a>
private	<a href="#">m_db</a>
private	<a href="#">m_letterCount</a>
private	<a href="#">m_themeName</a>

### Fields inherited from class java.util.Observable

changed, obs

### Constructor Summary

public	<a href="#">Theme</a> (java.lang.String themeName) Constructor
--------	---

### Method Summary

boolean	<a href="#">changeThemeName</a> (java.lang.String newThemeName) Changes the theme name.
void	<a href="#">decRefCount</a> ()
int	<a href="#">getCount</a> ()
java.lang.String	<a href="#">getThemeName</a> () Gets the theme name.
void	<a href="#">incRefCount</a> ()
boolean	<a href="#">isEditable</a> () Determines editability.

**Methods inherited from class** `java.util.Observable`

`addObserver`, `clearChanged`, `countObservers`, `deleteObserver`, `deleteObservers`, `hasChanged`, `notifyObservers`, `notifyObservers`, `setChanged`

**Methods inherited from class** `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `registerNatives`, `toString`, `wait`, `wait`, `wait`

## Fields

### **DEFAULT\_THEME\_NAME**

`public static java.lang.String` **DEFAULT\_THEME\_NAME**

### **ALL\_THEMES**

`public static java.lang.String` **ALL\_THEMES**

### **m\_themeName**

`private java.lang.String` **m\_themeName**

### **m\_db**

`private edu.bu.cs673.AwesomeAlphabet.model.Database` **m\_db**

### **m\_letterCount**

`private int` **m\_letterCount**

## Constructors

### **Theme**

`public` **Theme**(`java.lang.String` themeName)

Constructor

#### **Parameters:**

themeName - Theme Name

## Methods

(continued from last page)

## getThemeName

```
public java.lang.String getThemeName()
```

Gets the theme name.

**Returns:**

The theme name.

---

## isEditable

```
public boolean isEditable()
```

Determines editability.

**Returns:**

true if editable.

---

## incRefCount

```
public void incRefCount()
```

---

## decRefCount

```
public void decRefCount()
```

---

## getCount

```
public int getCount()
```

---

## changeThemeName

```
public boolean changeThemeName(java.lang.String newThemeName)
```

Changes the theme name.

**Parameters:**

`newThemeName` - The new theme name.

**Returns:**

True if theme name was changed.

---

## edu.bu.cs673.AwesomeAlphabet.model Class ThemeManager

```

java.lang.Object
  |
  +- java.util.Observable
        |
        +- edu.bu.cs673.AwesomeAlphabet.model.ThemeManager
  
```

```

public class ThemeManager
extends java.util.Observable
  
```

This class manages all themes used by the application.

### Field Summary

private	<a href="#">gui_update</a>
private	<a href="#">m_currentTheme</a>
private	<a href="#">m_db</a>
private	<a href="#">m_themes</a>

### Fields inherited from class java.util.Observable

changed, obs

### Constructor Summary

public	<a href="#">ThemeManager</a> ( ) Constructor.
--------	--

### Method Summary

boolean	<a href="#">addTheme</a> ( java.lang.String themeName) Adds a theme to the Theme Manager.
boolean	<a href="#">changeThemeName</a> ( java.lang.String oldThemeName, java.lang.String newThemeName) Changes the name of an existing theme.
boolean	<a href="#">deleteTheme</a> ( java.lang.String themeName) Deletes a theme from the Theme Manager.
<a href="#">Theme</a>	<a href="#">getCurrentTheme</a> ( ) Gets the current theme.
java.util.Iterator	<a href="#">getIterator</a> ( ) Gets an iterator to be able to walk through all themes
<a href="#">Theme</a>	<a href="#">getTheme</a> ( java.lang.String themeName) Gets the Theme object based on the theme name.

boolean	<a href="#"><code>hasTheme</code></a> ( java.lang.String themeName) Determines if Theme Manager contains the specified theme.
boolean	<a href="#"><code>loadTheme</code></a> ( java.lang.String themeName)
void	<a href="#"><code>notifyWhenUpdatesComplete</code></a> ( java.util.Observer observer)
boolean	<a href="#"><code>ReloadThemesFromDatabase</code></a> ()
boolean	<a href="#"><code>setCurrentTheme</code></a> ( java.lang.String themeName) Sets the current theme.

**Methods inherited from class java.util.Observable**

`addObserver`, `clearChanged`, `countObservers`, `deleteObserver`, `deleteObservers`, `hasChanged`, `notifyObservers`, `notifyObservers`, `setChanged`

**Methods inherited from class java.lang.Object**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `registerNatives`, `toString`, `wait`, `wait`, `wait`

## Fields

**m\_themes**

```
private java.util.ArrayList m_themes
```

**m\_currentTheme**

```
private edu.bu.cs673.AwesomeAlphabet.model.Theme m_currentTheme
```

**m\_db**

```
private edu.bu.cs673.AwesomeAlphabet.model.Database m_db
```

**gui\_update**

```
private edu.bu.cs673.AwesomeAlphabet.model.GuiUpdate gui_update
```

## Constructors

**ThemeManager**

```
public ThemeManager()
```

(continued from last page)

Constructor.

## Methods

### ReloadThemesFromDatabase

```
public boolean ReloadThemesFromDatabase()
```

---

### getIterator

```
public java.util.Iterator getIterator()
```

Gets an iterator to be able to walk through all themes

**Returns:**

The iterator.

---

### getTheme

```
public Theme getTheme(java.lang.String themeName)
```

Gets the Theme object based on the theme name.

**Returns:**

The theme object or null if the theme does not exist.

---

### hasTheme

```
public boolean hasTheme(java.lang.String themeName)
```

Determines if Theme Manager contains the specified theme.

**Parameters:**

themeName - Theme Name.

**Returns:**

True if Theme Manager contains the specified theme.

---

### addTheme

```
public boolean addTheme(java.lang.String themeName)
```

Adds a theme to the Theme Manager.

**Parameters:**

themeName - The theme name.

**Returns:**

True if theme was added successfully or already exists.

---

### loadTheme

```
public boolean loadTheme(java.lang.String themeName)
```

## deleteTheme

```
public boolean deleteTheme(java.lang.String themeName)
```

Deletes a theme from the Theme Manager.

**Parameters:**

themeName - The theme name.

**Returns:**

True if theme was deleted successfully or does not exist.

---

## changeThemeName

```
public boolean changeThemeName(java.lang.String oldThemeName,  
    java.lang.String newThemeName)
```

Changes the name of an existing theme.

**Parameters:**

oldThemeName - Old theme name.

newThemeName - New theme name.

**Returns:**

True if theme name was changed. False if old theme not found.

---

## setCurrentTheme

```
public boolean setCurrentTheme(java.lang.String themeName)
```

Sets the current theme.

**Parameters:**

themeName - Theme name.

**Returns:**

True if current theme was set. False if theme is invalid (ThemeManger does not contain specified theme).

---

## getCurrentTheme

```
public Theme getCurrentTheme()
```

Gets the current theme.

**Returns:**

The current theme.

---

## notifyWhenUpdatesComplete

```
public void notifyWhenUpdatesComplete(java.util.Observer observer)
```



## edu.bu.cs673.AwesomeAlphabet.model Class Version

```
java.lang.Object
├--edu.bu.cs673.AwesomeAlphabet.model.Version
```

```
public class Version
extends java.lang.Object
```

This class contains the application version number and related information.

### Field Summary

public static final	<a href="#">sAppName</a> Value: <b>Awesome Alphabet</b>
public static final	<a href="#">sVersionNum</a> Value: <b>0.3.0.0</b>

### Constructor Summary

public	<a href="#">Version()</a>
--------	---------------------------

### Method Summary

java.lang.String	<a href="#">getAppName()</a>
java.lang.String	<a href="#">getVersion()</a>

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

#### sAppName

```
public static final java.lang.String sAppName
```

Constant value: **Awesome Alphabet**

#### sVersionNum

```
public static final java.lang.String sVersionNum
```

Constant value: **0.3.0.0**

## Constructors

### Version

```
public Version()
```

## Methods

### getVersion

```
public java.lang.String getVersion()
```

---

### getAppName

```
public java.lang.String getAppName()
```

## edu.bu.cs673.AwesomeAlphabet.model Class WordPictureSound

java.lang.Object

└─ edu.bu.cs673.AwesomeAlphabet.model.WordPictureSound

### All Implemented Interfaces:

java.lang.Comparable

```
public class WordPictureSound
extends java.lang.Object
implements java.lang.Comparable
```

### Field Summary

public static	<a href="#">compareById</a> Compare objects by id
	<a href="#">id</a>
protected static	<a href="#">log</a>
private	<a href="#">m_cLetter</a>
private	<a href="#">m_image</a>
private	<a href="#">m_sound</a>
private	<a href="#">m_theme</a>
private	<a href="#">m_word</a>

### Constructor Summary

public	<a href="#">WordPictureSound</a> (char Letter, java.lang.String word, java.lang.String imageFile, java.lang.String soundFile, <a href="#">Theme</a> theme, int id)
--------	--

### Method Summary

void	<a href="#">changeTheme</a> ( <a href="#">Theme</a> newTheme) Changes the theme associated with this object.
int	<a href="#">compareTo</a> (java.lang.Object o) Compare two objects of type WordPictureSound
<a href="#">Theme</a>	<a href="#">getTheme</a> () Gets the current theme associated with this object.

javax.swing.ImageIcon	<a href="#">GetWordImage</a> (int width, int height)
char	<a href="#">getWordLetter</a> () Get the letter associated with this word.
java.lang.String	<a href="#">GetWordString</a> ()
void	<a href="#">PlaySound</a> ()
void	<a href="#">StopSound</a> ()

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

## Fields

### **m\_word**

```
private java.lang.String m_word
```

### **m\_sound**

```
private edu.bu.cs673.AwesomeAlphabet.model.GameSound m_sound
```

### **m\_image**

```
private java.awt.Image m_image
```

### **m\_theme**

```
private edu.bu.cs673.AwesomeAlphabet.model.Theme m_theme
```

### **m\_cLetter**

```
private char m_cLetter
```

(continued from last page)

**id**

```
java.lang.Integer id
```

---

**log**

```
protected static org.apache.log4j.Logger log
```

---

**compareTo**

```
public static java.util.Comparator compareTo
```

Compare objects by id

## Constructors

**WordPictureSound**

```
public WordPictureSound(char Letter,  
                        java.lang.String word,  
                        java.lang.String imageFile,  
                        java.lang.String soundFile,  
                        Theme theme,  
                        int id)
```

## Methods

**PlaySound**

```
public void PlaySound()
```

---

**StopSound**

```
public void StopSound()
```

---

**GetWordString**

```
public java.lang.String GetWordString()
```

---

**GetWordImage**

```
public javax.swing.ImageIcon GetWordImage(int width,  
                                             int height)
```

---

(continued from last page)

## getTheme

```
public Theme getTheme( )
```

Gets the current theme associated with this object.

**Returns:**

The current theme.

---

## changeTheme

```
public void changeTheme(Theme newTheme)
```

Changes the theme associated with this object.

**Parameters:**

newTheme - The new theme.

---

## getWordLetter

```
public char getWordLetter( )
```

Get the letter associated with this word.

**Parameters:**

wps - The word we want letter for

**Returns:**

returns char word is associated with

---

## compareTo

```
public int compareTo(java.lang.Object o)
```

Compare two objects of type WordPictureSound

**Parameters:**

anotherWordPictureSoundObject

**Returns:**

---

**Package**

**edu.bu.cs673.AwesomeAlphabet.value**

## edu.bu.cs673.AwesomeAlphabet.value Class ThemeViewData

java.lang.Object

└-edu.bu.cs673.AwesomeAlphabet.value.ThemeViewData

```
public class ThemeViewData
extends java.lang.Object
```

### Field Summary

private final	<a href="#">count</a>
private final	<a href="#">editable</a>
private final	<a href="#">themeName</a>

### Constructor Summary

public	<a href="#">ThemeViewData</a> (java.lang.String themeName, boolean editable, int count)
--------	---

### Method Summary

int	<a href="#">count</a> ()
boolean	<a href="#">editable</a> ()
java.lang.String	<a href="#">name</a> ()

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

#### themeName

```
private final java.lang.String themeName
```

#### editable

```
private final boolean editable
```



---

**count**

```
private final int count
```

## Constructors

**ThemeViewData**

```
public ThemeViewData(java.lang.String themeName,  
                     boolean editable,  
                     int count)
```

## Methods

**name**

```
public java.lang.String name()
```

---

**editable**

```
public boolean editable()
```

---

**count**

```
public int count()
```

## edu.bu.cs673.AwesomeAlphabet.value Class WPSViewData

java.lang.Object

└-edu.bu.cs673.AwesomeAlphabet.value.WPSViewData

public class **WPSViewData**  
extends java.lang.Object

### Field Summary

public final	<a href="#">m_letter</a>
public final	<a href="#">m_themeName</a>
public final	<a href="#">m_word</a>

### Constructor Summary

public	<a href="#">WPSViewData</a> (java.lang.String word, java.lang.String letter, java.lang.String themeName)
--------	--

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Fields

#### **m\_word**

public final java.lang.String **m\_word**

#### **m\_letter**

public final java.lang.String **m\_letter**

#### **m\_themeName**

public final java.lang.String **m\_themeName**

### Constructors

(continued from last page)

## WPSViewData

```
public WPSViewData(java.lang.String word,  
                   java.lang.String letter,  
                   java.lang.String themeName)
```

---

**Package**

**edu.bu.cs673.AwesomeAlphabet.view**

## edu.bu.cs673.AwesomeAlphabet.view Class AAJPanel

```

java.lang.Object
  |-- java.awt.Component
        |-- java.awt.Container
              |-- javax.swing.JComponent
                    |-- javax.swing.JPanel
                          |-- edu.bu.cs673.AwesomeAlphabet.view.AAJPanel

```

### All Implemented Interfaces:

java.io.Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver,  
javax.swing.TransferHandler.HasGetTransferHandler, java.io.Serializable, javax.accessibility.Accessible

```

public class AAJPanel
extends javax.swing.JPanel

```

This class defines a customized JPanel. It has the ability to add a background image.

## Field Summary

	<a href="#">m_backgroundImage</a>
	<a href="#">m_originalBackgroundImage</a>

### Fields inherited from class javax.swing.JPanel

uiClassID

### Fields inherited from class javax.swing.JComponent

aaTextInfo, accessibleContext, actionMap, ACTIONMAP\_CREATED, alignmentX, alignmentY, ANCESTOR\_INPUTMAP\_CREATED, ANCESTOR\_NOTIFIER\_KEY, ANCESTOR\_USING\_BUFFER, ancestorInputMap, autoscrolls, AUTOSCROLLS\_SET, border, clientProperties, COMPLETELY\_OBSCURED, componentObtainingGraphicsFrom, componentObtainingGraphicsFromLock, CREATED\_DOUBLE\_BUFFER, DEBUG\_GRAPHICS\_LOADED, defaultLocale, flags, FOCUS\_INPUTMAP\_CREATED, FOCUS\_TRAVERSAL\_KEYS\_BACKWARD\_SET, FOCUS\_TRAVERSAL\_KEYS\_FORWARD\_SET, focusController, focusInputMap, INHERITS\_POPUP\_MENU, INPUT\_VERIFIER\_KEY, INPUT\_VERIFIER\_SOURCE\_KEY, inputVerifier, IS\_DOUBLE\_BUFFERED, IS\_OPAQUE, IS\_PAINTING\_TILE, IS\_PRINTING, IS\_PRINTING\_ALL, IS\_REPAINTING, isAlignmentXSet, isAlignmentYSet, KEY\_EVENTS\_ENABLED, KEYBOARD\_BINDINGS\_KEY, listenerList, managingFocusBackwardTraversalKeys, managingFocusForwardTraversalKeys, NEXT\_FOCUS, NOT\_OBSCURED, OPAQUE\_SET, paintingChild, PARTIALLY\_OBSCURED, popupMenu, readObjectCallbacks, REQUEST\_FOCUS\_DISABLED, RESERVED\_1, RESERVED\_2, RESERVED\_3, RESERVED\_4, RESERVED\_5, RESERVED\_6, tempRectangles, TOOL\_TIP\_TEXT\_KEY, TRANSFER\_HANDLER\_KEY, ui, uiClassID, UNDEFINED\_CONDITION, verifyInputWhenFocusTarget, vetoableChangeSupport, WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT, WHEN\_FOCUSED, WHEN\_IN\_FOCUSED\_WINDOW, WHEN\_IN\_FOCUSED\_WINDOW\_BINDINGS, WIF\_INPUTMAP\_CREATED, windowInputMap, WRITE\_OBJ\_COUNTER\_FIRST, WRITE\_OBJ\_COUNTER\_LAST

### Fields inherited from class java.awt.Container

component, containerListener, containerSerializedDataVersion, dbg, descendantsCount, dispatcher, EMPTY\_ARRAY, focusCycleRoot, focusTraversalPolicy, focusTraversalPolicyProvider, INCLUDE\_SELF, layoutMgr, listeningBoundsChildren, listeningChildren, mixingLog, modalAppContext, modalComp, numOfHWComponents, numOfLWComponents, preserveBackgroundColor, printing, printingThreads, SEARCH\_HEAVYWEIGHTS, serialPersistentFields, serialVersionUID

#### Fields inherited from class `java.awt.Component`

acc, accessibleContext, actionListenerK, adjustmentListenerK, appContext, background, backgroundEraseDisabled, BOTTOM\_ALIGNMENT, boundsOp, bufferStrategy, CENTER\_ALIGNMENT, changeSupport, coalesceEventsParams, coalesceMap, coalescingEnabled, componentListener, componentListenerK, componentOrientation, componentSerializedDataVersion, compoundShape, containerListenerK, cursor, dbg, dropTarget, enabled, eventCache, eventMask, FOCUS\_TRAVERSABLE\_DEFAULT, FOCUS\_TRAVERSABLE\_SET, FOCUS\_TRAVERSABLE\_UNKNOWN, focusable, focusListener, focusListenerK, focusLog, focusTraversalKeyPropertyNames, focusTraversalKeys, focusTraversalKeysEnabled, font, foreground, graphicsConfig, height, hierarchyBoundsListener, hierarchyBoundsListenerK, hierarchyListener, hierarchyListenerK, ignoreRepaint, incRate, inputMethodListener, inputMethodListenerK, isAddNotifyComplete, isFocusTraversableOverridden, isInc, isPacked, itemListenerK, keyListener, keyListenerK, LEFT\_ALIGNMENT, locale, LOCK, log, maxSize, maxSizeSet, minSize, minSizeSet, mixingCutoutRegion, mixingLog, mouseListener, mouseListenerK, mouseMotionListener, mouseMotionListenerK, mouseWheelListener, mouseWheelListenerK, name, nameExplicitlySet, nativeInLightFixer, newEventsOnly, objectLock, ownedWindowK, parent, peer, peerFont, popups, prefSize, prefSizeSet, privateKey, requestFocusController, RIGHT\_ALIGNMENT, serialVersionUID, textListenerK, TOP\_ALIGNMENT, valid, visible, width, windowClosingException, windowFocusListenerK, windowListenerK, windowStateListenerK, x, y

#### Fields inherited from interface `java.awt.image.ImageObserver`

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

## Constructor Summary

public	<a href="#">AAJPanel()</a> Class constructor.
--------	--

## Method Summary

void	<a href="#">paint()</a> ( <code>java.awt.Graphics g</code> ) Overridden paint method.
void	<a href="#">setBackgroundImage()</a> ( <code>java.awt.Image image</code> ) Sets the background image.
void	<a href="#">setBackgroundImage()</a> ( <code>java.lang.String filename</code> ) Sets the background image.

#### Methods inherited from class `javax.swing.JPanel`

getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI, writeObject

#### Methods inherited from class `javax.swing.JComponent`

```

_paintImmediately, addAncestorListener, addNotify, addVetoableChangeListener,
adjustPaintFlags, alwaysOnTop, checkIfChildObscuredBySibling, clientPropertyChange,
componentInputMapChanged, computeVisibleRect, computeVisibleRect,
compWriteObjectNotify, contains, createToolTip, deregisterNextFocusableComponent,
disable, dndDone, dropLocationForPoint, enable, fetchRectangle, firePropertyChange,
firePropertyChange, firePropertyChange, fireVetoableChange, getAccessibleContext,
getActionForKeyStroke, getActionMap, getActionMap, getAlignmentX, getAlignmentY,
getAncestorListeners, getAncestorNotifier, getAutoscrolls, getBaseline,
getBaselineResizeBehavior, getBorder, getBounds, getClientProperties,
getClientProperty, getComponentGraphics, getComponentPopupMenu,
getConditionForKeyStroke, getCreatedDoubleBuffer, getDebugGraphicsOptions,
getDefaultLocale, getFlag, getFontMetrics, getGraphics, getGraphicsInvoked,
getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputMap,
getInputVerifier, getInsets, getInsets, getListeners, getLocation,
getManagingFocusBackwardTraversalKeys, getManagingFocusForwardTraversalKeys,
getMaximumSize, getMinimumSize, getNextFocusableComponent, getObscuredState,
getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize,
getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor,
getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget,
getVetoableChangeListeners, getVisibleRect, getWidth, getWriteObjCounter, getX, getY,
grabFocus, isComponentObtainingGraphicsFrom, isDoubleBuffered,
isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled,
isPainting, isPaintingForPrint, isPaintingOrigin, isPaintingTile,
isRequestFocusEnabled, isValidateRoot, paint, paintBorder, paintChildren,
paintComponent, paintForceDoubleBuffered, paintImmediately, paintImmediately,
paintToOffscreen, paramString, print, printAll, printBorder, printChildren,
printComponent, processComponentKeyEvent, processKeyBinding, processKeyBindings,
processKeyBindingsForAllComponents, processKeyEvent, processMouseEvent,
processMouseMotionEvent, putClientProperty, readObject, rectangleIsObscured,
recycleRectangle, registerKeyboardAction, registerKeyboardAction,
registerNextFocusableComponent, registerNextFocusableComponent,
registerWithKeyboardManager, registerWithKeyboardManager, removeAncestorListener,
removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus,
requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow,
resetKeyboardActions, reshape, revalidate, safelyGetGraphics, safelyGetGraphics,
scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls,
setBackground, setBorder, setComponentPopupMenu, setCreatedDoubleBuffer,
setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setDropLocation,
setEnabled, setFlag, setFocusTraversalKeys, setFont, setForeground,
setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize,
setNextFocusableComponent, setOpaque, setPaintingChild, setPreferredSize,
setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI, setUIProperty,
setVerifyInputWhenFocusTarget, setVisible, setWriteObjCounter, shouldDebugGraphics,
superProcessMouseMotionEvent, unregisterKeyboardAction,
unregisterWithKeyboardManager, unregisterWithKeyboardManager, update, updateUI,
writeObject

```

**Methods inherited from class** java.awt.Container

```

add, add, add, add, add, addContainerListener, addDelicately, addImpl, addNotify,
addPropertyChangeListener, addPropertyChangeListener, adjustDecendantsOnParent,
adjustDescendants, adjustListeningChildren, applyComponentOrientation,
areFocusTraversalKeysSet, canContainFocusOwner, checkAdding, checkAddToSelf, checkGD,
checkNotAWindow, clearCurrentFocusCycleRootOnHide, clearMostRecentFocusOwnerOnHide,
containsFocus, countComponents, countHierarchyMembers, createChildHierarchyEvents,
createHierarchyEvents, decreaseComponentCount, deliverEvent, dispatchEventImpl,
dispatchEventToSelf, doLayout, eventEnabled, findComponentAt, findComponentAt,
findComponentAt, findComponentAtImpl, findTraversalRoot, getAccessibleAt,
getAccessibleChild, getAccessibleChildrenCount, getAlignmentX, getAlignmentY,
getBottommostComponentIndex, getComponent, getComponentAt, getComponentAt,
getComponentCount, getComponents_NoClientCode, getComponents, getComponentZOrder,
getContainerListeners, getDropTargetEventTarget, getFocusTraversalKeys,
getFocusTraversalPolicy, getHeavyweightContainer, getInsets, getLayout, getListeners,
getListenersCount, getMaximumSize, getMinimumSize, getMouseEventTarget,
getMouseEventTarget, getMouseEventTargetImpl, getMousePosition, getOpaqueShape,
getPreferredSize, getTopmostComponentIndex, hasHeavyweightDescendants,
hasLightweightDescendants, increaseComponentCount, initializeFocusTraversalKeys,
initIDs, insets, invalidate, invalidateTree, isAncestorOf, isFocusCycleRoot,
isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet,
isParentOf, isRemoveNotifyNeeded, isSameOrAncestorOf, layout, lightweightPaint,
lightweightPrint, list, list, locate, minimumSize, mixOnReshaping, mixOnShowing,
mixOnValidating, mixOnZOrderChanging, nextFocusHelper, numListening, paint,
paintComponents, paintHeavyweightComponents, paramString, postProcessKeyEvent,
postsOldMouseEvents, preferredSize, preProcessKeyEvent, print, printComponents,
printHeavyweightComponents, processContainerEvent, processEvent, proxyEnableEvents,
readObject, recursiveApplyCurrentShape, recursiveApplyCurrentShape,
recursiveApplyCurrentShape, recursiveSubtractAndApplyShape,
recursiveSubtractAndApplyShape, recursiveSubtractAndApplyShape, remove, remove,
removeAll, removeContainerListener, removeDelicately, removeNotify, reparentChild,
reparentTraverse, setComponentZOrder, setFocusCycleRoot, setFocusTraversalKeys,
setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setFont, setLayout,
startLWModal, stopLWModal, transferFocusBackward, transferFocusDownCycle, update,
validate, validateTree, writeObject

```

**Methods inherited from class** java.awt.Component





```
removeComponentListener, removeFocusListener, removeHierarchyBoundsListener,
removeHierarchyListener, removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener, removeMouseWheelListener,
removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, repaint, repaintParentIfNeeded, requestFocus, requestFocus,
requestFocus, requestFocus, requestFocusHelper, requestFocusHelper,
requestFocusInWindow, requestFocusInWindow, requestFocusInWindow,
requestFocusInWindow, resetGC, reshape, reshapeNativePeer, resize, resize,
setBackground, setBounds, setBounds, setBoundsOp, setComponentOrientation, setCursor,
setDropTarget, setEnabled, setFocusable, setFocusTraversalKeys_NoIDCheck,
setFocusTraversalKeys, setFocusTraversalKeysEnabled, setFont, setForeground,
setGCFromPeer, setIgnoreRepaint, setLocale, setLocation, setLocation, setMaximumSize,
setMinimumSize, setName, setPreferredSize, setRequestFocusController, setSize,
setSize, setVisible, show, show, size, subtractAndApplyShape,
subtractAndApplyShapeBelowMe, toString, transferFocus, transferFocusBackward,
transferFocusUpCycle, update, updateCursorImmediately, validate, writeObject
```

#### Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait
```

#### Methods inherited from interface `java.awt.image.ImageObserver`

```
imageUpdate
```

#### Methods inherited from interface `java.awt.MenuContainer`

```
getFont, postEvent, remove
```

#### Methods inherited from interface `javax.swing.TransferHandler.HasGetTransferHandler`

```
getTransferHandler
```

#### Methods inherited from interface `javax.accessibility.Accessible`

```
getAccessibleContext
```

## Fields

### **`m_originalBackgroundImage`**

```
java.awt.Image m_originalBackgroundImage
```

### **`m_backgroundImage`**

```
java.awt.Image m_backgroundImage
```

## Constructors

(continued from last page)

## AAJPanel

```
public AAJPanel()
```

Class constructor.

## Methods

### SetBackgroundImage

```
public void SetBackgroundImage(java.lang.String filename)
```

Sets the background image.

**Parameters:**

filename - The filename of the image.

---

### SetBackgroundImage

```
public void SetBackgroundImage(java.awt.Image image)
```

Sets the background image.

**Parameters:**

image - The image.

---

### paint

```
public void paint(java.awt.Graphics g)
```

Overridden paint method. Paints the background image.

## edu.bu.cs673.AwesomeAlphabet.view

### Class AlphabetPageView

java.lang.Object

```

+--edu.bu.cs673.AwesomeAlphabet.view.PageView
    +--edu.bu.cs673.AwesomeAlphabet.view.AlphabetPageView
  
```

#### All Implemented Interfaces:

java.util.Observer

public class **AlphabetPageView**

extends [PageView](#)

This class defines the Alphabet Page View. From this view, the user can play the alphabet song, go to the Title Page, or click on a letter button to go to a specific Letter Page.

### Field Summary

private	<a href="#">letterPanel</a>
static	<a href="#">log</a>
private	<a href="#">m_controller</a>

#### Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[AA\\_NAV\\_BUTTON\\_ALPHABET\\_PAGE](#), [AA\\_NAV\\_BUTTON\\_ALPHABET\\_SONG](#), [AA\\_NAV\\_BUTTON\\_HOME](#),  
[AA\\_NAV\\_BUTTON\\_MANAGE\\_SETTINGS](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_THEMES](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_WORDS](#),  
[AA\\_NAV\\_BUTTON\\_NEXT\\_EXAMPLE](#), [AA\\_NAV\\_BUTTON\\_NEXT\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_OPTIONS](#),  
[AA\\_NAV\\_BUTTON\\_PREV\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_RETURN\\_HOME](#), [AA\\_NAV\\_BUTTON\\_START](#),  
[AA\\_NAV\\_BUTTON\\_TITLE\\_PAGE](#), [backgroundColor](#), [border](#), [headingBackground](#), [headingFont](#), [infoFont](#),  
[letterFont](#), [m\\_panel](#), [m\\_sPageName](#), [wordFont](#)

### Constructor Summary

public	<a href="#">AlphabetPageView</a> (java.lang.String sPageName) Class constructor.
--------	---

### Method Summary

void	<a href="#">activated</a> ()
void	<a href="#">OnLetterButtonClick</a> ( <a href="#">Letter</a> cLetter) Called when a letter button is clicked and causes the Letter Page to be shown.
void	<a href="#">OnPlayAlphabetSongButtonClick</a> () Called when the Alphabet Song button is clicked and causes the alphabet song to be played.
void	<a href="#">OnTitlePageButtonClick</a> () Called when the Title Page button is clicked and causes the Title Page to be shown.

void	<a href="#">SetController</a> ( <a href="#">AlphabetPageController</a> controller) Sets the controller associated with this view and creates the letter buttons.
void	<a href="#">update</a> (java.util.Observable o, java.lang.Object arg) Causes the view to be updated when the model changes.

Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[activated](#), [getButtonImage](#), [getPageName](#), [getPagePanel](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

Methods inherited from interface java.util.Observer

update

## Fields

### **m\_controller**

```
private edu.bu.cs673.AwesomeAlphabet.controller.AlphabetPageController m_controller
```

### **letterPanel**

```
private javax.swing.JPanel letterPanel
```

### **log**

```
static org.apache.log4j.Logger log
```

## Constructors

### **AlphabetPageView**

```
public AlphabetPageView(java.lang.String sPageName)
```

Class constructor.

#### **Parameters:**

sPageName - The page name associated with this view.

## Methods

### **SetController**

```
public void SetController(AlphabetPageController controller)
```

(continued from last page)

Sets the controller associated with this view and creates the letter buttons.

**Parameters:**

controller - The controller.

---

## update

```
public void update(java.util.Observable o,  
                  java.lang.Object arg)
```

Causes the view to be updated when the model changes. Currently this method does nothing. It is required for the observer pattern.

---

## OnLetterButtonClick

```
public void OnLetterButtonClick(Letter cLetter)
```

Called when a letter button is clicked and causes the Letter Page to be shown.

**Parameters:**

cLetter - The letter to show.

---

## OnTitlePageButtonClick

```
public void OnTitlePageButtonClick()
```

Called when the Title Page button is clicked and causes the Title Page to be shown.

---

## OnPlayAlphabetSongButtonClick

```
public void OnPlayAlphabetSongButtonClick()
```

Called when the Alphabet Song button is clicked and causes the alphabet song to be played.

---

## activated

```
public void activated()
```

## edu.bu.cs673.AwesomeAlphabet.view Interface IPageObserver

All Known Implementing Classes:

[MainWindow](#)

```
public interface IPageObserver
extends
```

This interface defines a Page Observer. It is used to to help manage page transitions without requiring classes to have knowledge of the user interface.

### Method Summary

boolean	<a href="#">GoToPage</a> ( java.lang.String sPageName) Commands a page to be shown.
---------	--

### Methods

#### GoToPage

```
public boolean GoToPage( java.lang.String sPageName)
```

Commands a page to be shown.

**Parameters:**

sPageName - The name of the page.

**Returns:**

True if page is able to be shown.

## edu.bu.cs673.AwesomeAlphabet.view Interface IThemeControllerView

All Known Implementing Classes:

[ThemePageView](#)

```
public interface IThemeControllerView
extends java.util.Observer
```

This interface defines a Theme Controller View. Any view that wants to use the ThemeController must implement this interface.

### Method Summary

void	<a href="#">SetController</a> ( <a href="#">ThemeController</a> controller)
------	---

### Methods inherited from interface `java.util.Observer`

`update`

### Methods

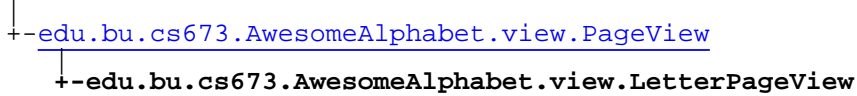
#### SetController

```
public void SetController(ThemeController controller)
```



## edu.bu.cs673.AwesomeAlphabet.view Class LetterPageView

java.lang.Object



### All Implemented Interfaces:

java.util.Observer

public class **LetterPageView**  
extends [PageView](#)

This class defines the Letter Page View. From this view, the user is able to see the current letter; see words and pictures associated with the letter; and play letter and word sounds. The user is also able to go to the Title and Alphabet pages and see the previous and next letters.

### Field Summary

static	<a href="#">log</a>
	<a href="#">m_controller</a>
	<a href="#">m_image</a>
	<a href="#">m_lowercase</a>
	<a href="#">m_uppercase</a>
	<a href="#">m_word</a>

### Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[AA\\_NAV\\_BUTTON\\_ALPHABET\\_PAGE](#), [AA\\_NAV\\_BUTTON\\_ALPHABET\\_SONG](#), [AA\\_NAV\\_BUTTON\\_HOME](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_SETTINGS](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_THEMES](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_WORDS](#), [AA\\_NAV\\_BUTTON\\_NEXT\\_EXAMPLE](#), [AA\\_NAV\\_BUTTON\\_NEXT\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_OPTIONS](#), [AA\\_NAV\\_BUTTON\\_PREV\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_RETURN\\_HOME](#), [AA\\_NAV\\_BUTTON\\_START](#), [AA\\_NAV\\_BUTTON\\_TITLE\\_PAGE](#), [backgroundColor](#), [border](#), [headingBackground](#), [headingFont](#), [infoFont](#), [letterFont](#), [m\\_panel](#), [m\\_sPageName](#), [wordFont](#)

### Constructor Summary

public	<a href="#">LetterPageView</a> (java.lang.String sPageName) Class constructor.
--------	---

### Method Summary

void	<a href="#">activated</a> () Lets the controller know that this view has become active.
------	--

void	<a href="#"><u>OnAlphabetPageButtonClick()</u></a> Called when the Alphabet Page button is clicked and causes the Alphabet Page to be shown.
void	<a href="#"><u>OnGetNextExampleButtonClick()</u></a> Called when the Get Next Example button is clicked and causes the next word and picture to be shown for the current letter.
void	<a href="#"><u>OnLetterClick()</u></a> Called when the letter text/image is clicked and causes the controller to play a phonetic letter sound.
void	<a href="#"><u>OnNextLetterButtonClick()</u></a> Called when the Next Letter button is clicked and causes the next letter to be displayed (if available).
void	<a href="#"><u>OnPictureClick()</u></a> Called when the word's picture is clicked and causes the controller to play the word's sound.
void	<a href="#"><u>OnPreviousLetterButtonClick()</u></a> Called when the Previous Letter button is clicked and causes the previous letter to be displayed (if available).
void	<a href="#"><u>OnTitlePageButtonClick()</u></a> Called when the Title Page button is clicked and causes the Title Page to be shown.
void	<a href="#"><u>SetController()</u></a> ( <a href="#"><u>LetterPageController</u></a> controller) Sets the controller associated with this view.
void	<a href="#"><u>update()</u></a> (java.util.Observable o, java.lang.Object arg) Causes the view to be updated when the model changes.

#### Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[activated](#), [getButtonImage](#), [getPageName](#), [getPagePanel](#)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

#### Methods inherited from interface java.util.Observer

update

## Fields

### **m\_controller**

edu.bu.cs673.AwesomeAlphabet.controller.LetterPageController **m\_controller**

### **m\_uppercase**

javax.swing.JButton **m\_uppercase**

---

## m\_lowercase

```
javax.swing.JButton m_lowercase
```

---

## m\_image

```
javax.swing.JButton m_image
```

---

## m\_word

```
javax.swing.JLabel m_word
```

---

## log

```
static org.apache.log4j.Logger log
```

---

# Constructors

## LetterPageView

```
public LetterPageView(java.lang.String sPageName)
```

Class constructor.

### Parameters:

sPageName - The page name associated with this view.

# Methods

## SetController

```
public void SetController(LetterPageController controller)
```

Sets the controller associated with this view.

### Parameters:

controller - The controller.

---

## update

```
public void update(java.util.Observable o,  
    java.lang.Object arg)
```

Causes the view to be updated when the model changes. It is required for the observer pattern.

---

## OnLetterClick

```
public void OnLetterClick()
```

---

(continued from last page)

Called when the letter text/image is clicked and causes the controller to play a phonetic letter sound.

---

## **OnPictureClick**

```
public void OnPictureClick()
```

Called when the word's picture is clicked and causes the controller to play the word's sound.

---

## **OnNextLetterButtonClick**

```
public void OnNextLetterButtonClick()
```

Called when the Next Letter button is clicked and causes the next letter to be displayed (if available).

---

## **OnPreviousLetterButtonClick**

```
public void OnPreviousLetterButtonClick()
```

Called when the Previous Letter button is clicked and causes the previous letter to be displayed (if available).

---

## **OnAlphabetPageButtonClick**

```
public void OnAlphabetPageButtonClick()
```

Called when the Alphabet Page button is clicked and causes the Alphabet Page to be shown.

---

## **OnTitlePageButtonClick**

```
public void OnTitlePageButtonClick()
```

Called when the Title Page button is clicked and causes the Title Page to be shown.

---

## **OnGetNextExampleButtonClick**

```
public void OnGetNextExampleButtonClick()
```

Called when the Get Next Example button is clicked and causes the next word and picture to be shown for the current letter.

---

## **activated**

```
public void activated()
```

Lets the controller know that this view has become active.

---

## edu.bu.cs673.AwesomeAlphabet.view Class MainWindow

java.lang.Object

└-edu.bu.cs673.AwesomeAlphabet.view.MainWindow

All Implemented Interfaces:

[IPageObserver](#)

public class **MainWindow**  
 extends java.lang.Object  
 implements [IPageObserver](#)

### Field Summary

protected static final	<a href="#">AA_JFRAME_SIZE_HEIGHT</a> Value: <b>800</b>
protected static final	<a href="#">AA_JFRAME_SIZE_WIDTH</a> Value: <b>600</b>
static	<a href="#">log</a>
private	<a href="#">m_cl</a>
private	<a href="#">m_curView</a>
private	<a href="#">m_frame</a>
private	<a href="#">m_pageHash</a>

### Constructor Summary

public	<a href="#">MainWindow()</a>
--------	------------------------------

### Method Summary

javax.swing.JFrame	<a href="#">getJFrame()</a>
boolean	<a href="#">GoToPage</a> ( java.lang.String sPageName)
void	<a href="#">registerPage</a> ( <a href="#">PageView</a> page)
void	<a href="#">Show()</a>

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait
```

Methods inherited from interface [edu.bu.cs673.AwesomeAlphabet.view.IPageObserver](#)

[GoToPage](#)

## Fields

### AA\_JFRAME\_SIZE\_HEIGHT

```
protected static final int AA_JFRAME_SIZE_HEIGHT
```

Constant value: **800**

### AA\_JFRAME\_SIZE\_WIDTH

```
protected static final int AA_JFRAME_SIZE_WIDTH
```

Constant value: **600**

### log

```
static org.apache.log4j.Logger log
```

### m\_frame

```
private javax.swing.JFrame m_frame
```

### m\_curView

```
private javax.swing.JPanel m_curView
```

### m\_cl

```
private java.awt.CardLayout m_cl
```

### m\_pageHash

```
private java.util.Hashtable m_pageHash
```

## Constructors

(continued from last page)

## MainWindow

```
public MainWindow()
```

## Methods

### registerPage

```
public void registerPage(PageView page)
```

---

### Show

```
public void Show()
```

---

### GoToPage

```
public boolean GoToPage(java.lang.String sPageName)
```

---

### getJFrame

```
public javax.swing.JFrame getJFrame()
```

## edu.bu.cs673.AwesomeAlphabet.view Class OptionsPageView

java.lang.Object

└- [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

└- **edu.bu.cs673.AwesomeAlphabet.view.OptionsPageView**

All Implemented Interfaces:

java.util.Observer

public class **OptionsPageView**

extends [PageView](#)

### Field Summary

private static	<a href="#">log</a>
private	<a href="#">m_bOptionsLocked</a>
private	<a href="#">m_controller</a>
private final	<a href="#">m_iUnlockResponseCount</a> Value: 3
private	<a href="#">m_lockUnlockBtn</a>
private	<a href="#">m_lockUnlockControlsBox</a>
private	<a href="#">m_mainMenuBtn</a>
private	<a href="#">m_manageSettingsBtn</a>
private	<a href="#">m_manageThemesBtn</a>
private	<a href="#">m_manageWordsBtn</a>
private	<a href="#">m_sExpectedUnlockResponse</a>
private	<a href="#">m_unlockAnswerTextField</a>
private	<a href="#">m_unlockInstructionsLbl</a>

Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)



[AA\\_NAV\\_BUTTON\\_ALPHABET\\_PAGE](#), [AA\\_NAV\\_BUTTON\\_ALPHABET\\_SONG](#), [AA\\_NAV\\_BUTTON\\_HOME](#),  
[AA\\_NAV\\_BUTTON\\_MANAGE\\_SETTINGS](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_THEMES](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_WORDS](#),  
[AA\\_NAV\\_BUTTON\\_NEXT\\_EXAMPLE](#), [AA\\_NAV\\_BUTTON\\_NEXT\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_OPTIONS](#),  
[AA\\_NAV\\_BUTTON\\_PREV\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_RETURN\\_HOME](#), [AA\\_NAV\\_BUTTON\\_START](#),  
[AA\\_NAV\\_BUTTON\\_TITLE\\_PAGE](#), [backgroundColor](#), [border](#), [headingBackground](#), [headingFont](#), [infoFont](#),  
[letterFont](#), [m\\_panel](#), [m\\_sPageName](#), [wordFont](#)

## Constructor Summary

public	<a href="#">OptionsPageView</a> (java.lang.String pageName)
--------	---

## Method Summary

void	<a href="#">activated</a> ()
void	<a href="#">EnableOptionControls</a> (boolean bEnable)
java.lang.String	<a href="#">GetNumberWord</a> (int iNum)
java.lang.String	<a href="#">GetUnlockQuestion</a> ()
java.lang.String	<a href="#">GetUnlockResponse</a> (int iIndex)
void	<a href="#">OnLockUnlockBtnClick</a> ()
void	<a href="#">OnManageSettingsClick</a> ()
void	<a href="#">OnManageThemesClick</a> ()
void	<a href="#">OnManageWordsClick</a> ()
void	<a href="#">OnReturnHomeClick</a> ()
void	<a href="#">SetController</a> ( <a href="#">OptionsPageController</a> controller) Sets the controller associated with this view.
void	<a href="#">update</a> (java.util.Observable arg0, java.lang.Object arg1)

### Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[activated](#), [getButtonImage](#), [getPageName](#), [getPagePanel](#)

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

### Methods inherited from interface java.util.Observer

update

## Fields

### **m\_controller**

```
private edu.bu.cs673.AwesomeAlphabet.controller.OptionsPageController m_controller
```

---

### **log**

```
private static org.apache.log4j.Logger log
```

---

### **m\_iUnlockResponseCount**

```
private final int m_iUnlockResponseCount
```

Constant value: 3

---

### **m\_bOptionsLocked**

```
private boolean m_bOptionsLocked
```

---

### **m\_manageWordsBtn**

```
private javax.swing.JButton m_manageWordsBtn
```

---

### **m\_manageThemesBtn**

```
private javax.swing.JButton m_manageThemesBtn
```

---

### **m\_manageSettingsBtn**

```
private javax.swing.JButton m_manageSettingsBtn
```

---

### **m\_mainMenuBtn**

```
private javax.swing.JButton m_mainMenuBtn
```

---

### **m\_lockUnlockControlsBox**

```
private javax.swing.Box m_lockUnlockControlsBox
```

---

---

### **m\_lockUnlockBtn**

```
private javax.swing.JButton m_lockUnlockBtn
```

---

### **m\_unlockInstructionsLbl**

```
private javax.swing.JLabel m_unlockInstructionsLbl
```

---

### **m\_unlockAnswerTextField**

```
private javax.swing.JTextField m_unlockAnswerTextField
```

---

### **m\_sExpectedUnlockResponse**

```
private java.lang.String m_sExpectedUnlockResponse
```

---

## Constructors

### **OptionsPageView**

```
public OptionsPageView(java.lang.String pageName)
```

## Methods

### **update**

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

---

### **activated**

```
public void activated()
```

---

### **OnReturnHomeClick**

```
public void OnReturnHomeClick()
```

---

### **OnManageThemesClick**

```
public void OnManageThemesClick()
```

---

(continued from last page)

---

## OnManageWordsClick

```
public void OnManageWordsClick()
```

---

## OnManageSettingsClick

```
public void OnManageSettingsClick()
```

---

## OnLockUnlockBtnClick

```
public void OnLockUnlockBtnClick()
```

---

## SetController

```
public void SetController(OptionsPageController controller)
```

Sets the controller associated with this view.

### Parameters:

controller - The controller.

---

## EnableOptionControls

```
private void EnableOptionControls(boolean bEnable)
```

---

## GetUnlockQuestion

```
private java.lang.String GetUnlockQuestion()
```

---

## GetNumberWord

```
private java.lang.String GetNumberWord(int iNum)
```

---

## GetUnlockResponse

```
public java.lang.String GetUnlockResponse(int iIndex)
```

## edu.bu.cs673.AwesomeAlphabet.view Class PageView

java.lang.Object

└-edu.bu.cs673.AwesomeAlphabet.view.PageView

### All Implemented Interfaces:

java.util.Observer

### Direct Known Subclasses:

[AlphabetPageView](#), [LetterPageView](#), [OptionsPageView](#), [SettingPageView](#), [ThemePageView](#), [TitlePageView](#), [WordEditView](#), [WPSView](#)

public abstract class **PageView**  
 extends java.lang.Object  
 implements java.util.Observer

This class defines the Page View. It is the abstract parent class of all page view classes, including AlphabetPageView, LetterPageView, etc. It manages the page name and panel.

### Field Summary

protected static final	<a href="#">AA_NAV_BUTTON_ALPHABET_PAGE</a> Value: <b>AlphabetPage.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_ALPHABET_SONG</a> Value: <b>Music.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_HOME</a> Value: <b>Home.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_MANAGE_SETTINGS</a> Value: <b>ManageOptions.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_MANAGE_THEMES</a> Value: <b>ManageThemes.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_MANAGE_WORDS</a> Value: <b>ManageWords.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_NEXT_EXAMPLE</a> Value: <b>NextExample.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_NEXT_LETTER</a> Value: <b>NextArrow.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_OPTIONS</a> Value: <b>OptionsButton.png</b>

protected static final	<a href="#">AA_NAV_BUTTON_PREV_LETTER</a> Value: <b>PreviousArrow.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_RETURN_HOME</a> Value: <b>ReturnHome.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_START</a> Value: <b>StartButton.png</b>
protected static final	<a href="#">AA_NAV_BUTTON_TITLE_PAGE</a> Value: <b>Home.png</b>
protected static	<a href="#">backgroundColor</a>
protected static	<a href="#">border</a>
protected static	<a href="#">headingBackground</a>
protected static	<a href="#">headingFont</a>
protected static	<a href="#">infoFont</a>
protected static	<a href="#">letterFont</a>
protected	<a href="#">m_panel</a>
private	<a href="#">m_sPageName</a>
protected static	<a href="#">wordFont</a>

## Constructor Summary

public	<a href="#">PageView</a> ( java.lang.String sPageName) Class constructor.
--------	--

## Method Summary

abstract void	<a href="#">activated</a> ()
javax.swing.JButton	<a href="#">getButtonImage</a> ( java.lang.String filename, java.lang.String defaultName) Gets a button image from the graphics code.
java.lang.String	<a href="#">getPageName</a> () Gets the page name.
javax.swing.JPanel	<a href="#">getPagePanel</a> () Gets the panel.

**Methods inherited from class** java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,  
toString, wait, wait, wait
```

#### Methods inherited from interface `java.util.Observer`

```
update
```

## Fields

### **m\_sPageName**

```
private java.lang.String m_sPageName
```

### **m\_panel**

```
protected edu.bu.cs673.AwesomeAlphabet.view.AAJPanel m_panel
```

### **letterFont**

```
protected static java.awt.Font letterFont
```

### **headingFont**

```
protected static java.awt.Font headingFont
```

### **wordFont**

```
protected static java.awt.Font wordFont
```

### **infoFont**

```
protected static java.awt.Font infoFont
```

### **border**

```
protected static javax.swing.border.Border border
```

### **backgroundColor**

```
protected static java.awt.Color backgroundColor
```

(continued from last page)

---

## headingBackground

protected static java.awt.Color **headingBackground**

---

## AA\_NAV\_BUTTON\_TITLE\_PAGE

protected static final java.lang.String **AA\_NAV\_BUTTON\_TITLE\_PAGE**

Constant value: **Home.png**

---

## AA\_NAV\_BUTTON\_ALPHABET\_SONG

protected static final java.lang.String **AA\_NAV\_BUTTON\_ALPHABET\_SONG**

Constant value: **Music.png**

---

## AA\_NAV\_BUTTON\_HOME

protected static final java.lang.String **AA\_NAV\_BUTTON\_HOME**

Constant value: **Home.png**

---

## AA\_NAV\_BUTTON\_NEXT\_LETTER

protected static final java.lang.String **AA\_NAV\_BUTTON\_NEXT\_LETTER**

Constant value: **NextArrow.png**

---

## AA\_NAV\_BUTTON\_PREV\_LETTER

protected static final java.lang.String **AA\_NAV\_BUTTON\_PREV\_LETTER**

Constant value: **PreviousArrow.png**

---

## AA\_NAV\_BUTTON\_NEXT\_EXAMPLE

protected static final java.lang.String **AA\_NAV\_BUTTON\_NEXT\_EXAMPLE**

Constant value: **NextExample.png**

---

## AA\_NAV\_BUTTON\_ALPHABET\_PAGE

protected static final java.lang.String **AA\_NAV\_BUTTON\_ALPHABET\_PAGE**

Constant value: **AlphabetPage.png**

---



(continued from last page)

## AA\_NAV\_BUTTON\_START

```
protected static final java.lang.String AA_NAV_BUTTON_START
```

Constant value: **StartButton.png**

## AA\_NAV\_BUTTON\_OPTIONS

```
protected static final java.lang.String AA_NAV_BUTTON_OPTIONS
```

Constant value: **OptionsButton.png**

## AA\_NAV\_BUTTON\_MANAGE\_THEMES

```
protected static final java.lang.String AA_NAV_BUTTON_MANAGE_THEMES
```

Constant value: **ManageThemes.png**

## AA\_NAV\_BUTTON\_MANAGE\_WORDS

```
protected static final java.lang.String AA_NAV_BUTTON_MANAGE_WORDS
```

Constant value: **ManageWords.png**

## AA\_NAV\_BUTTON\_MANAGE\_SETTINGS

```
protected static final java.lang.String AA_NAV_BUTTON_MANAGE_SETTINGS
```

Constant value: **ManageOptions.png**

## AA\_NAV\_BUTTON\_RETURN\_HOME

```
protected static final java.lang.String AA_NAV_BUTTON_RETURN_HOME
```

Constant value: **ReturnHome.png**

## Constructors

### PageView

```
public PageView(java.lang.String sPageName)
```

Class constructor.

#### Parameters:

sPageName - The name of the page associated with this view.

## Methods

### getPageName

```
public java.lang.String getPageName()
```

(continued from last page)

Gets the page name.

**Returns:**

The page name as a String.

---

## getPagePanel

```
public javax.swing.JPanel getPagePanel()
```

Gets the panel.

**Returns:**

The panel as a JPanel.

---

## getButtonImage

```
public javax.swing.JButton getButtonImage(java.lang.String filename,  
                                             java.lang.String defaultName)
```

Gets a button image from the graphics code. If the image could not be loaded, then it creates text button using the default text.

**Parameters:**

filename - the graphics file to read

defaultName - the name of the text button used as a fall-back

**Returns:**

a button ready for displaying

---

## activated

```
public abstract void activated()
```

## edu.bu.cs673.AwesomeAlphabet.view Class SettingPageView

java.lang.Object

```

├- edu.bu.cs673.AwesomeAlphabet.view.PageView
│   └- edu.bu.cs673.AwesomeAlphabet.view.SettingPageView

```

### All Implemented Interfaces:

java.util.Observer

public class **SettingPageView**  
extends [PageView](#)

### Field Summary

private	<a href="#">displayOptionsArray</a>
private	<a href="#">displayOrderOptions</a>
static	<a href="#">log</a>
private	<a href="#">m_controller</a>
private	<a href="#">wordsLimitField</a>

### Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[AA\\_NAV\\_BUTTON\\_ALPHABET\\_PAGE](#), [AA\\_NAV\\_BUTTON\\_ALPHABET\\_SONG](#), [AA\\_NAV\\_BUTTON\\_HOME](#),  
[AA\\_NAV\\_BUTTON\\_MANAGE\\_SETTINGS](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_THEMES](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_WORDS](#),  
[AA\\_NAV\\_BUTTON\\_NEXT\\_EXAMPLE](#), [AA\\_NAV\\_BUTTON\\_NEXT\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_OPTIONS](#),  
[AA\\_NAV\\_BUTTON\\_PREV\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_RETURN\\_HOME](#), [AA\\_NAV\\_BUTTON\\_START](#),  
[AA\\_NAV\\_BUTTON\\_TITLE\\_PAGE](#), [backgroundColor](#), [border](#), [headingBackground](#), [headingFont](#), [infoFont](#),  
[letterFont](#), [m\\_panel](#), [m\\_sPageName](#), [wordFont](#)

### Constructor Summary

public	<a href="#">SettingPageView</a> (java.lang.String sPageName)
--------	--

### Method Summary

void	<a href="#">activated</a> ()
void	<a href="#">OnCancelClicked</a> ()
void	<a href="#">OnSaveClicked</a> ()

void	<a href="#">SetController</a> ( <a href="#">SettingController</a> controller) Sets the controller associated with the setting page's view.
void	<a href="#">update</a> (java.util.Observable arg0, java.lang.Object arg1)

Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[activated](#), [getButtonImage](#), [getPageName](#), [getPagePanel](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

Methods inherited from interface java.util.Observer

update

## Fields

### log

static org.apache.log4j.Logger **log**

### m\_controller

private edu.bu.cs673.AwesomeAlphabet.controller.SettingController **m\_controller**

### displayOptionsArray

private java.lang.String **displayOptionsArray**

### displayOrderOptions

private javax.swing.JComboBox **displayOrderOptions**

### wordsLimitField

private javax.swing.JTextField **wordsLimitField**

## Constructors

(continued from last page)

## SettingPageView

```
public SettingPageView(java.lang.String sPageName)
```

## Methods

### update

```
public void update(java.util.Observable arg0,  
    java.lang.Object arg1)
```

---

### activated

```
public void activated()
```

---

### OnSaveClicked

```
public void OnSaveClicked()
```

---

### OnCancelClicked

```
public void OnCancelClicked()
```

---

## SetController

```
public void SetController(SettingController controller)
```

Sets the controller associated with the setting page's view.

#### Parameters:

controller - The controller.

## edu.bu.cs673.AwesomeAlphabet.view Class ThemePageView

java.lang.Object

```

  |
+- edu.bu.cs673.AwesomeAlphabet.view.PageView
  |
  +- edu.bu.cs673.AwesomeAlphabet.view.ThemePageView

```

All Implemented Interfaces:

[IThemeControllerView](#), java.util.Observer

```

public class ThemePageView
extends PageView
implements java.util.Observer, IThemeControllerView

```

### Field Summary

static	<a href="#">log</a>
private	<a href="#">m_controller</a>
private	<a href="#">m_currentTheme</a>
private	<a href="#">m_deleteButton</a>
private	<a href="#">m_modifyButton</a>
private	<a href="#">m_textField</a>
private	<a href="#">m_themeModel</a>
private	<a href="#">m_themeTable</a>

### Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[AA\\_NAV\\_BUTTON\\_ALPHABET\\_PAGE](#), [AA\\_NAV\\_BUTTON\\_ALPHABET\\_SONG](#), [AA\\_NAV\\_BUTTON\\_HOME](#),  
[AA\\_NAV\\_BUTTON\\_MANAGE\\_SETTINGS](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_THEMES](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_WORDS](#),  
[AA\\_NAV\\_BUTTON\\_NEXT\\_EXAMPLE](#), [AA\\_NAV\\_BUTTON\\_NEXT\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_OPTIONS](#),  
[AA\\_NAV\\_BUTTON\\_PREV\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_RETURN\\_HOME](#), [AA\\_NAV\\_BUTTON\\_START](#),  
[AA\\_NAV\\_BUTTON\\_TITLE\\_PAGE](#), [backgroundColor](#), [border](#), [headingBackground](#), [headingFont](#), [infoFont](#),  
[letterFont](#), [m\\_panel](#), [m\\_sPageName](#), [wordFont](#)

### Constructor Summary

public	<a href="#">ThemePageView</a> (java.lang.String pageName)
--------	---

### Method Summary

void	<a href="#">activated()</a>
void	<a href="#">OnClearCurrentTheme()</a>
void	<a href="#">OnReturnHomeClick()</a>
void	<a href="#">OnSetThemeClicked()</a>
void	<a href="#">OnThemeAddClicked()</a>
void	<a href="#">OnThemeDeleteClicked()</a>
void	<a href="#">OnThemeModifyClicked()</a>
void	<a href="#">repopulateThemes()</a>
void	<a href="#">SetController</a> ( <a href="#">ThemeController</a> themeController)
void	<a href="#">update</a> (java.util.Observable o, java.lang.Object arg)

#### Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[activated](#), [getButtonImage](#), [getPageName](#), [getPagePanel](#)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

#### Methods inherited from interface java.util.Observer

update

#### Methods inherited from interface [edu.bu.cs673.AwesomeAlphabet.view.IThemeControllerView](#)

[SetController](#)

#### Methods inherited from interface java.util.Observer

update

## Fields

### log

static org.apache.log4j.Logger **log**

(continued from last page)

---

## **m\_controller**

```
private edu.bu.cs673.AwesomeAlphabet.controller.ThemeController m_controller
```

---

## **m\_themeTable**

```
private javax.swing.JTable m_themeTable
```

---

## **m\_themeModel**

```
private edu.bu.cs673.AwesomeAlphabet.view.ThemeTableModel m_themeModel
```

---

## **m\_textField**

```
private javax.swing.JTextField m_textField
```

---

## **m\_modifyButton**

```
private javax.swing.JButton m_modifyButton
```

---

## **m\_deleteButton**

```
private javax.swing.JButton m_deleteButton
```

---

## **m\_currentTheme**

```
private javax.swing.JLabel m_currentTheme
```

# Constructors

## **ThemePageView**

```
public ThemePageView(java.lang.String pageName)
```

# Methods

## **update**

```
public void update(java.util.Observable o,  
                  java.lang.Object arg)
```



---

## activated

```
public void activated()
```

---

---

## repopulateThemes

```
private void repopulateThemes()
```

---

---

## OnReturnHomeClick

```
public void OnReturnHomeClick()
```

---

---

## OnThemeAddClicked

```
public void OnThemeAddClicked()
```

---

---

## OnThemeModifyClicked

```
public void OnThemeModifyClicked()
```

---

---

## OnThemeDeleteClicked

```
public void OnThemeDeleteClicked()
```

---

---

## OnSetThemeClicked

```
public void OnSetThemeClicked()
```

---

---

## OnClearCurrentTheme

```
public void OnClearCurrentTheme()
```

---

---

## SetController

```
public void SetController(ThemeController themeController)
```

---

## edu.bu.cs673.AwesomeAlphabet.view Class ThemeTableModel

```

java.lang.Object
  |
  +- javax.swing.table.AbstractTableModel
    |
    +- edu.bu.cs673.AwesomeAlphabet.view.ThemeTableModel
  
```

### All Implemented Interfaces:

java.io.Serializable, javax.swing.table.TableModel

```

public class ThemeTableModel
extends javax.swing.table.AbstractTableModel
  
```

### Field Summary

private static	<a href="#">m_themeTable</a>
private static final	<a href="#">serialVersionUID</a> Value: <b>-224567130880087561</b>

### Fields inherited from class javax.swing.table.AbstractTableModel

<a href="#">listenerList</a>
------------------------------

### Constructor Summary

public	<a href="#">ThemeTableModel()</a>
--------	-----------------------------------

### Method Summary

void	<a href="#">addElement</a> ( <a href="#">ThemeViewData</a> next)
int	<a href="#">getColumnCount</a> ()
java.lang.String	<a href="#">getColumnName</a> (int column)
int	<a href="#">getRowCount</a> ()
java.lang.String	<a href="#">getThemeName</a> (int selectedRow)
java.lang.Object	<a href="#">getValueAt</a> (int row, int column)
void	<a href="#">removeAllElements</a> ()
boolean	<a href="#">rowIsEditable</a> (int selectedRow)

void	<a href="#">sort()</a>
------	------------------------

**Methods inherited from class** `javax.swing.table.AbstractTableModel`

```
addTableModelListener, findColumn, fireTableCellUpdated, fireTableChanged,
fireTableDataChanged, fireTableRowsDeleted, fireTableRowsInserted,
fireTableRowsUpdated, fireTableStructureChanged, getColumnClass, getColumnName,
getListeners, getTableModelListeners, isCellEditable, removeTableModelListener,
setValueAt
```

**Methods inherited from class** `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait
```

**Methods inherited from interface** `javax.swing.table.TableModel`

```
addTableModelListener, getColumnClass, getColumnCount, getColumnName, getRowCount,
getValueAt, isCellEditable, removeTableModelListener, setValueAt
```

## Fields

**serialVersionUID**

```
private static final long serialVersionUID
```

Constant value: **-224567130880087561**

**m\_themeTable**

```
private static java.util.ArrayList m_themeTable
```

## Constructors

**ThemeTableModel**

```
public ThemeTableModel()
```

## Methods

**getColumnName**

```
public java.lang.String getColumnName(int column)
```

**getColumnCount**

```
public int getColumnCount()
```

(continued from last page)

---

## getRowCount

```
public int getRowCount()
```

---

## getValueAt

```
public java.lang.Object getValueAt(int row,  
                                     int column)
```

---

## removeAllElements

```
public void removeAllElements()
```

---

## addElement

```
public void addElement(ThemeViewData next)
```

---

## sort

```
public void sort()
```

---

## getThemeName

```
public java.lang.String getThemeName(int selectedRow)
```

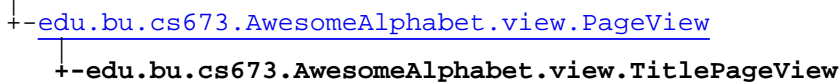
---

## rowIsEditable

```
public boolean rowIsEditable(int selectedRow)
```

## edu.bu.cs673.AwesomeAlphabet.view Class TitlePageView

java.lang.Object



### All Implemented Interfaces:

java.util.Observer

public class **TitlePageView**  
extends [PageView](#)

This class defines the Title Page View. From this view, the user is able to see the application's name and version and go to the Alphabet Page.

### Field Summary

static	<a href="#">log</a>
private	<a href="#">m_controller</a>

### Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[AA\\_NAV\\_BUTTON\\_ALPHABET\\_PAGE](#), [AA\\_NAV\\_BUTTON\\_ALPHABET\\_SONG](#), [AA\\_NAV\\_BUTTON\\_HOME](#),  
[AA\\_NAV\\_BUTTON\\_MANAGE\\_SETTINGS](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_THEMES](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_WORDS](#),  
[AA\\_NAV\\_BUTTON\\_NEXT\\_EXAMPLE](#), [AA\\_NAV\\_BUTTON\\_NEXT\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_OPTIONS](#),  
[AA\\_NAV\\_BUTTON\\_PREV\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_RETURN\\_HOME](#), [AA\\_NAV\\_BUTTON\\_START](#),  
[AA\\_NAV\\_BUTTON\\_TITLE\\_PAGE](#), [backgroundColor](#), [border](#), [headingBackground](#), [headingFont](#), [infoFont](#),  
[letterFont](#), [m\\_panel](#), [m\\_sPageName](#), [wordFont](#)

### Constructor Summary

public	<a href="#">TitlePageView</a> (java.lang.String sPageName) Class constructor.
--------	--

### Method Summary

void	<a href="#">activated</a> ()
void	<a href="#">OnOptionsButtonClick</a> () Called when the Options button is clicked and causes the options page to be shown.
void	<a href="#">OnStartButtonClick</a> () Called when the Start Button is clicked and causes the Alphabet Page to be shown.
void	<a href="#">SetController</a> ( <a href="#">TitlePageController</a> controller) Sets the controller associated with this view.
void	<a href="#">update</a> (java.util.Observable o, java.lang.Object arg) Causes the view to be updated when the model changes.

---

**Methods inherited from class** [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[activated](#), [getButtonImage](#), [getPageName](#), [getPagePanel](#)

**Methods inherited from class** java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

**Methods inherited from interface** java.util.Observer

update

---

## Fields

### **m\_controller**

```
private edu.bu.cs673.AwesomeAlphabet.controller.TitlePageController m_controller
```

---

### **log**

```
static org.apache.log4j.Logger log
```

---

## Constructors

### **TitlePageView**

```
public TitlePageView(java.lang.String sPageName)
```

Class constructor.

**Parameters:**

sPageName - The page name associated with this view.

## Methods

### **SetController**

```
public void SetController(TitlePageController controller)
```

Sets the controller associated with this view.

**Parameters:**

controller - The controller.

---

### **update**

```
public void update(java.util.Observable o,  
                  java.lang.Object arg)
```

---

(continued from last page)

Causes the view to be updated when the model changes. Currently this method does nothing. It is required for the observer pattern.

---

## **OnStartButtonClick**

```
public void OnStartButtonClick()
```

Called when the Start Button is clicked and causes the Alphabet Page to be shown.

---

## **OnOptionsButtonClick**

```
public void OnOptionsButtonClick()
```

Called when the Options button is clicked and causes the options page to be shown.

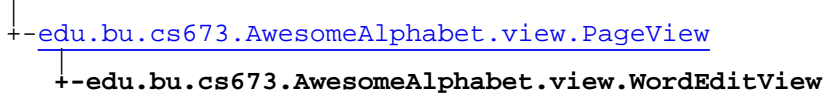
---

## **activated**

```
public void activated()
```

## edu.bu.cs673.AwesomeAlphabet.view Class WordEditView

java.lang.Object



All Implemented Interfaces:

java.util.Observer

public class **WordEditView**

extends [PageView](#)

### Field Summary

private static final	<a href="#">chooser</a>
private	<a href="#">letters</a>
protected static	<a href="#">log</a>
private	<a href="#">m_controller</a>
private	<a href="#">m_imageFileField</a>
private	<a href="#">m_letterChoice</a>
private	<a href="#">m_soundFileField</a>
private	<a href="#">m_sUnselectedThemeName</a>
private	<a href="#">m_themeChoice</a>
private	<a href="#">m_wordField</a>

### Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[AA\\_NAV\\_BUTTON\\_ALPHABET\\_PAGE](#), [AA\\_NAV\\_BUTTON\\_ALPHABET\\_SONG](#), [AA\\_NAV\\_BUTTON\\_HOME](#),  
[AA\\_NAV\\_BUTTON\\_MANAGE\\_SETTINGS](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_THEMES](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_WORDS](#),  
[AA\\_NAV\\_BUTTON\\_NEXT\\_EXAMPLE](#), [AA\\_NAV\\_BUTTON\\_NEXT\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_OPTIONS](#),  
[AA\\_NAV\\_BUTTON\\_PREV\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_RETURN\\_HOME](#), [AA\\_NAV\\_BUTTON\\_START](#),  
[AA\\_NAV\\_BUTTON\\_TITLE\\_PAGE](#), [backgroundColor](#), [border](#), [headingBackground](#), [headingFont](#), [infoFont](#),  
[letterFont](#), [m\\_panel](#), [m\\_sPageName](#), [wordFont](#)

### Constructor Summary

public	<a href="#">WordEditView</a> (java.lang.String pageName)
--------	--



## Method Summary

void	<a href="#"><code>activated()</code></a>
void	<a href="#"><code>OnCancelClicked()</code></a>
void	<a href="#"><code>OnSaveClicked()</code></a>
void	<a href="#"><code>SetController</code></a> ( <a href="#"><code>WordEditController</code></a> controller) Sets the controller associated with this view.
void	<a href="#"><code>update</code></a> (java.util.Observable arg0, java.lang.Object arg1)

### Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[`activated`](#), [`getButtonImage`](#), [`getPageName`](#), [`getPagePanel`](#)

### Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `registerNatives`, `toString`, `wait`, `wait`, `wait`

### Methods inherited from interface java.util.Observer

`update`

## Fields

### **m\_controller**

```
private edu.bu.cs673.AwesomeAlphabet.controller.WordEditController m_controller
```

### **m\_wordField**

```
private javax.swing.JTextField m_wordField
```

### **m\_themeChoice**

```
private javax.swing.JComboBox m_themeChoice
```

### **m\_imageFileField**

```
private javax.swing.JTextField m_imageFileField
```

(continued from last page)

**m\_soundFileField**

```
private javax.swing.JTextField m_soundFileField
```

---

**chooser**

```
private static final javax.swing.JFileChooser chooser
```

---

**letters**

```
private java.lang.String letters
```

---

**m\_sUnselectedThemeName**

```
private java.lang.String m_sUnselectedThemeName
```

---

**m\_letterChoice**

```
private javax.swing.JComboBox m_letterChoice
```

---

**log**

```
protected static org.apache.log4j.Logger log
```

---

## Constructors

**WordEditView**

```
public WordEditView(java.lang.String pageName)
```

## Methods

**OnCancelClicked**

```
public void OnCancelClicked()
```

---

**OnSaveClicked**

```
public void OnSaveClicked()
```

---

---

## update

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

---

## activated

```
public void activated()
```

---

## SetController

```
public void SetController(WordEditController controller)
```

Sets the controller associated with this view.

### Parameters:

controller - The controller.

## edu.bu.cs673.AwesomeAlphabet.view Class WPSTableModel

```

java.lang.Object
  |
  +- javax.swing.table.AbstractTableModel
    |
    +- edu.bu.cs673.AwesomeAlphabet.view.WPSTableModel
  
```

### All Implemented Interfaces:

java.io.Serializable, javax.swing.table.TableModel

```

public class WPSTableModel
extends javax.swing.table.AbstractTableModel
  
```

### Field Summary

private static final	<a href="#">columnNames</a>
	<a href="#">m_words</a>
private static final	<a href="#">serialVersionUID</a> Value: <b>-6728415049308133929</b>

### Fields inherited from class javax.swing.table.AbstractTableModel

<a href="#">listenerList</a>
------------------------------

### Constructor Summary

public	<a href="#">WPSTableModel()</a>
--------	---------------------------------

### Method Summary

void	<a href="#">addElement(<a href="#">WPSTableViewData</a> wpsViewData)</a>
int	<a href="#">getColumnCount()</a>
java.lang.String	<a href="#">getColumnName(int column)</a>
int	<a href="#">getRowCount()</a>
java.lang.String	<a href="#">getSelectedWord(int selectedRow)</a>
java.lang.Object	<a href="#">getValueAt(int row, int col)</a>
void	<a href="#">removeAllElements()</a>

void	<a href="#">sort()</a>
------	------------------------

**Methods inherited from class** javax.swing.table.AbstractTableModel

```
addTableModelListener, findColumn, fireTableCellUpdated, fireTableChanged,
fireTableDataChanged, fireTableRowsDeleted, fireTableRowsInserted,
fireTableRowsUpdated, fireTableStructureChanged, getColumnClass, getColumnName,
getListeners, getTableModelListeners, isCellEditable, removeTableModelListener,
setValueAt
```

**Methods inherited from class** java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives,
toString, wait, wait, wait
```

**Methods inherited from interface** javax.swing.table.TableModel

```
addTableModelListener, getColumnClass, getColumnCount, getColumnName, getRowCount,
getValueAt, isCellEditable, removeTableModelListener, setValueAt
```

## Fields

**serialVersionUID**

```
private static final long serialVersionUID
```

Constant value: **-6728415049308133929**

**columnNames**

```
private static final java.lang.String columnNames
```

**m\_words**

```
java.util.ArrayList m_words
```

## Constructors

**WPSTableModel**

```
public WPSTableModel()
```

## Methods

**getColumnName**

```
public java.lang.String getColumnName(int column)
```

(continued from last page)

---

**getColumnCount**

```
public int getColumnCount()
```

---

**getRowCount**

```
public int getRowCount()
```

---

**getValueAt**

```
public java.lang.Object getValueAt(int row,  
                                   int col)
```

---

**sort**

```
public void sort()
```

---

**removeAllElements**

```
public void removeAllElements()
```

---

**addElement**

```
public void addElement(WPSViewData wpsViewData)
```

---

**getSelectedWord**

```
public java.lang.String getSelectedWord(int selectedRow)
```

## edu.bu.cs673.AwesomeAlphabet.view Class WPSView

java.lang.Object



All Implemented Interfaces:

java.util.Observer

public class **WPSView**  
extends [PageView](#)

### Field Summary

static	<a href="#">log</a>
private	<a href="#">m_controller</a>
private	<a href="#">m_textField</a>
private	<a href="#">m_wordModel</a>
private	<a href="#">m_wordTable</a>

### Fields inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[AA\\_NAV\\_BUTTON\\_ALPHABET\\_PAGE](#), [AA\\_NAV\\_BUTTON\\_ALPHABET\\_SONG](#), [AA\\_NAV\\_BUTTON\\_HOME](#),  
[AA\\_NAV\\_BUTTON\\_MANAGE\\_SETTINGS](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_THEMES](#), [AA\\_NAV\\_BUTTON\\_MANAGE\\_WORDS](#),  
[AA\\_NAV\\_BUTTON\\_NEXT\\_EXAMPLE](#), [AA\\_NAV\\_BUTTON\\_NEXT\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_OPTIONS](#),  
[AA\\_NAV\\_BUTTON\\_PREV\\_LETTER](#), [AA\\_NAV\\_BUTTON\\_RETURN\\_HOME](#), [AA\\_NAV\\_BUTTON\\_START](#),  
[AA\\_NAV\\_BUTTON\\_TITLE\\_PAGE](#), [backgroundColor](#), [border](#), [headingBackground](#), [headingFont](#), [infoFont](#),  
[letterFont](#), [m\\_panel](#), [m\\_sPageName](#), [wordFont](#)

### Constructor Summary

public	<a href="#">WPSView</a> (java.lang.String pageName)
--------	---

### Method Summary

void	<a href="#">activated</a> ()
void	<a href="#">OnReturnHomeClicked</a> ()
void	<a href="#">OnWordAddClicked</a> ()
void	<a href="#">OnWordDeleteClicked</a> ()

void	<a href="#"><u>OnWordEditClicked()</u></a>
void	<a href="#"><u>refreshWordList()</u></a>
void	<a href="#"><u>SetController</u></a> ( <a href="#"><u>WPSController</u></a> controller) Sets the controller associated with this view.
void	<a href="#"><u>update</u></a> (java.util.Observable arg0, java.lang.Object arg1)
void	<a href="#"><u>updateWordList</u></a> (java.util.Iterator words)

#### Methods inherited from class [edu.bu.cs673.AwesomeAlphabet.view.PageView](#)

[activated](#), [getButtonImage](#), [getPageName](#), [getPagePanel](#)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, registerNatives, toString, wait, wait, wait

#### Methods inherited from interface java.util.Observer

update

## Fields

### **m\_wordTable**

```
private javax.swing.JTable m_wordTable
```

### **m\_wordModel**

```
private edu.bu.cs673.AwesomeAlphabet.view.WPSTableModel m_wordModel
```

### **m\_textField**

```
private javax.swing.JTextField m_textField
```

### **m\_controller**

```
private edu.bu.cs673.AwesomeAlphabet.controller.WPSController m_controller
```

### **log**

```
static org.apache.log4j.Logger log
```



(continued from last page)

## Constructors

### WPSView

```
public WPSView(java.lang.String pageName)
```

## Methods

### refreshWordList

```
private void refreshWordList()
```

---

### updateWordList

```
public void updateWordList(java.util.Iterator words)
```

---

### update

```
public void update(java.util.Observable arg0,  
                  java.lang.Object arg1)
```

---

### activated

```
public void activated()
```

---

### OnWordAddClicked

```
public void OnWordAddClicked()
```

---

### OnWordEditClicked

```
public void OnWordEditClicked()
```

---

### OnWordDeleteClicked

```
public void OnWordDeleteClicked()
```

---

### OnReturnHomeClicked

```
public void OnReturnHomeClicked()
```

## SetController

```
public void SetController(WPSController controller)
```

Sets the controller associated with this view.

**Parameters:**

controller - The controller.

# Index

## A

AA\_ALPHABET\_SIZE 50  
AA\_JFRAME\_SIZE\_HEIGHT 118  
AA\_JFRAME\_SIZE\_WIDTH 118  
AA\_NAV\_BUTTON\_ALPHABET\_PAGE 128  
AA\_NAV\_BUTTON\_ALPHABET\_SONG 128  
AA\_NAV\_BUTTON\_HOME 128  
AA\_NAV\_BUTTON\_MANAGE\_SETTINGS 129  
AA\_NAV\_BUTTON\_MANAGE\_THEMES 129  
AA\_NAV\_BUTTON\_MANAGE\_WORDS 129  
AA\_NAV\_BUTTON\_NEXT\_EXAMPLE 128  
AA\_NAV\_BUTTON\_NEXT\_LETTER 128  
AA\_NAV\_BUTTON\_OPTIONS 129  
AA\_NAV\_BUTTON\_PREV\_LETTER 128  
AA\_NAV\_BUTTON\_RETURN\_HOME 129  
AA\_NAV\_BUTTON\_START 128  
AA\_NAV\_BUTTON\_TITLE\_PAGE 128  
AAConfig 37  
AAJPanel 106  
actionPerformed 7  
activated 110, 116, 123, 130, 133, 137, 143, 147, 153  
addElement 140, 150  
addImageResource 40  
addLetterSoundResource 75  
addNewWord 54  
addPhonicSoundResource 75  
addResource 74  
addSoundResource 39  
addTheme 22, 60, 87  
addWord 61  
addWordToIndex 40  
ALL\_THEMES 83  
Alphabet 51  
alphabet 20  
AlphabetPage 80  
AlphabetPageController 4  
AlphabetPageView 109  
AwesomeAlphabetApp 42

## B

backgroundColor 127

BASE\_DIR 35  
baseDirName 36  
border 127  
ButtonHandler 7

## C

CancelEditWord 27  
changeTheme 94  
changeThemeName 23, 60, 84, 88  
changeWordData 62  
changeWordName 62  
chooser 146  
columnNames 149  
compareById 93  
compareTo 94  
CONFIG\_PROPS 35  
copy\_dir 39  
copy\_file 39  
copy\_res\_to\_file 39  
copy\_stream 39  
count 97  
createDefaultTheme 59  
createLoadPersistentResourceDir 53  
CreateNewWord 30  
createTables 59  
curr\_clip 69  
curr\_sound 73  
cwd 36

## D

Database 59  
decRefCount 84  
DEFAULT\_LETTERS 36  
DEFAULT\_THEME\_NAME 83  
deleteTheme 23, 60, 88  
DeleteWord 30  
deleteWord 55, 62  
DISPLAY\_ORDER 45  
displayOptionsArray 132  
displayOrderOptions 132

## E

editable 96, 97

EditWord 30

editWord 55

EnableForeignKeySupport 60

EnableOptionControls 124

## F

forceUpdate 70

## G

GameImage 66

GameSound 69

getAbsImageFilePath 28, 55

getAbsSoundFilePath 28, 56

getAppName 90

getButtonImage 130

getColumnCount 139, 150

getColumnName 139, 149

getCount 84

GetCurrentLetter 53

getCurrentTheme 23, 88

getCurrentWordEditing 28, 55

getDatabaseInstance 59

getDisplayOrder 20, 46

getGraphicsResource 37

getGraphicsResourceDir 38

getGraphicsResourceDirAbs 38

getGraphicsResourceDirPersistentAbs 38

getGraphicsResourcePersistent 37

GetIcon 14

getIcon 75

getImage 66

GetIterator 52, 74

getIterator 87

getJFrame 119

GetLetterAsChar 74

GetLetterIndex 52

getLetterIndex 28

GetLetterIterator 4

getLetterPropFileName 37

getLetterProps 38

getLetterPropsPersistent 38

GetLowercaseLetterAsChar 14

getMaxExamples 20, 46

GetNextExample 13

GetNextLetter 13

getNumberRowsWordTable 61

GetNumberWord 124

getPageName 129

getPagePanel 130

GetPreviousLetter 13

getResourceDirAbs 37

getResourceDirPersistentAbs 37

getRowCount 140, 150

getSelectedWord 150

getSoundResource 38

getSoundResourceDir 38

getSoundResourceDirAbs 38

getSoundResourceDirPersistentAbs 38

getSoundResourcePersistent 38

getTheme 87, 93

getThemeId 61

getThemeName 61, 83, 140

getThemeNamesIterator 22, 27

getThemes 61

GetUnlockQuestion 124

GetUnlockResponse 124

GetUppercaseLetter 74

GetUppercaseLetterAsChar 14

getValueAt 140, 150

getVersion 90

GetWord 14

getWord 75

GetWordCacheIterator 52

getWordData 63

GetWordImage 93

getWordLetter 94

getWordNames 63

getWordPictureSound 54

getWords 30, 52

GetWordString 93

getWPSData 75

GoToAlphabetPage 13

GoToLetterPage 4

GoToManageSettingsPage 16

GoToNextLetter 53

GoToOptionsMenu 20, 30

GoToOptionsPage 23

GoToPage 18, 111, 119  
GoToPreviousLetter 53  
GoToThemesPage 16  
GoToTitlePage 4, 13, 16  
GoToWordEditPage 16  
GRAPHICS\_DIR 35  
graphicsSubDir 36  
gui\_update 86  
GuiUpdate 70

## H

hasTheme 60, 87  
hasWord 62  
headingBackground 128  
headingFont 127

## I

id 92  
incRefCount 84  
infoFont 127  
Initialize 51  
initializeDatabase 59  
isEditable 84

## L

LabelClickHandler 9  
LETTER 78  
Letter 74  
letter 65  
LetterClicked 13  
letterFont 127  
LetterPage 80  
LetterPageController 12  
LetterPageView 115  
letterPanel 109  
letterProps 37  
letterPropsName 36  
letterPropsPersistent 37  
letters 146  
loadAlphabetSong 53  
LoadDatabase 53  
loader 36, 45

loadFileFromClassPath 46  
loadFileFromPersistentDirectory 46  
loadLetterPhonicSound 53  
loadLettersFromDatabase 54  
loadLetterSound 53  
LoadResources 54  
loadSettingProperties 45  
loadTheme 87  
log 7, 9, 19, 37, 42, 45, 50, 59, 66, 68, 74, 93, 109, 115, 118, 122, 132, 135, 142, 146, 152

## M

m\_alphabet 3, 12  
m\_alphabetsong 51  
m\_backgroundImage 106  
m\_bOptionsLocked 122  
m\_cl 118  
m\_cLetter 73, 92  
m\_con 59  
m\_controller 109, 114, 122, 132, 135, 142, 145, 152  
m\_currentTheme 86, 136  
m\_currentWordEditing 51  
m\_curView 118  
m\_db 51, 58, 83, 86  
m\_deleteButton 136  
m\_frame 118  
m\_iCurLetterIndex 51  
m\_iDefThemeId 59  
m\_image 92, 115  
m\_imageFileField 145  
m\_index 73  
m\_iUnlockResponseCount 122  
m\_letter 98  
m\_letterChoice 146  
m\_letterCount 83  
m\_letters 50  
m\_LetterSound 73  
m\_lockUnlockBtn 123  
m\_lockUnlockControlsBox 122  
m\_lowercase 115  
m\_mainMenuBtn 122  
m\_manageSettingsBtn 122  
m\_manageThemesBtn 122  
m\_manageWordsBtn 122

m\_model 27, 30  
m\_modifyButton 136  
m\_originalBackgroundImage 106  
m\_pageHash 118  
m\_pageObserver 17  
m\_panel 127  
m\_PhonicSound 73  
m\_sExpectedUnlockResponse 123  
m\_sound 92  
m\_soundFileField 145  
m\_sPageName 127  
m\_sUnselectedThemeName 146  
m\_textField 136, 152  
m\_theme 92  
m\_themeChoice 145  
m\_themeMgr 22, 27, 51, 73  
m\_themeModel 136  
m\_themeName 83, 98  
m\_themes 86  
m\_themeTable 136, 139  
m\_unlockAnswerTextField 123  
m\_unlockInstructionsLbl 123  
m\_uppercase 114  
m\_view 12, 15, 22, 27, 30  
m\_word 92, 98, 115  
m\_word\_cache 51  
m\_wordField 145  
m\_wordModel 152  
m\_words 149  
m\_wordTable 152  
m\_wps 73  
main 42  
MainWindow 118  
MAXIMUM\_EXAMPLES 45  
method 7, 9  
mouseClicked 9  
mouseEntered 9  
mouseExited 10  
mousePressed 10  
mouseReleased 10

## N

name 97  
nextExample 75

NONE 78  
notifyWhenUpdatesComplete 88

## O

ObserveCurrentLetter 14  
OnAlphabetPageButtonClick 116  
OnCancelClicked 133, 146  
OnClearCurrentTheme 137  
OnGetNextExampleButtonClick 116  
OnLetterButtonClick 110  
OnLetterClick 115  
OnLockUnlockBtnClick 124  
OnManageSettingsClick 124  
OnManageThemesClick 123  
OnManageWordsClick 124  
OnNextLetterButtonClick 116  
OnOptionsButtonClick 143  
OnPictureClick 116  
OnPlayAlphabetSongButtonClick 110  
OnPreviousLetterButtonClick 116  
OnReturnHomeClick 123, 137  
OnReturnHomeClicked 153  
OnSaveClicked 133, 146  
OnSetThemeClicked 137  
OnStartButtonClick 143  
OnThemeAddClicked 137  
OnThemeDeleteClicked 137  
OnThemeModifyClicked 137  
OnTitlePageButtonClick 110, 116  
OnWordAddClicked 153  
OnWordDeleteClicked 153  
OnWordEditClicked 153  
Options 25  
OptionsPage 80  
OptionsPageController 16  
OptionsPageView 123

## P

PageController 17  
PageName 81  
PageView 129  
paint 107  
PERSISTENT\_RES 36

persistentResDir 36  
PHONIC 78  
PictureClicked 13  
picturePath 64  
PlayAlphabetSong 4, 54  
PlaySound 69, 93  
playSound 75  
playSoundLetter 76  
playSoundPhonic 76  
props 45  
pv 6, 9

## R

refreshWordList 153  
registerPage 119  
ReloadThemesFromDatabase 87  
removeAllElements 140, 150  
removeAllEntries 76  
removeImageResource 40  
removeResource 76  
removeSoundResource 40  
removeWordFromIndex 40  
repopulateThemes 137  
resetDeafultOrder 76  
resetListOrder 55  
rowIsEditable 140

## S

sAppName 89  
SaveEditWord 28  
SaveNewWord 28  
saveSettingProperties 46  
serialVersionUID 139, 149  
SetBackgroundImage 107  
SetController 109, 112, 115, 124, 133, 137, 142, 147, 154  
SetCurrentLetter 52  
setCurrentTheme 23, 88  
setCurrentWordEditing 55  
SettingController 20  
SettingPage 81  
SettingPageView 132  
SettingPropertiesFile 45  
Settings 45

Show 119  
shuffleExamples 54  
shuffleList 76  
sort 140, 150  
sortLetterExamples 55  
sortList 76  
Sound\_Type 78  
soundName 68  
soundPath 64  
SOUNDS\_DIR 35  
soundsSubDir 36  
Start 24  
StopAlphabetSound 54  
StopSound 12, 69, 93  
stopSound 76  
stringValue 81  
SubstringSearch 31  
sVersionNum 89

## T

Theme 83  
theme 65  
ThemeController 22  
ThemeManager 86  
themeName 96  
ThemePage 80  
ThemePageView 136  
ThemeTableModel 139  
ThemeViewData 97  
TitlePage 80  
TitlePageController 24  
TitlePageView 142  
toString 81

## U

unsetCurrentWordEditing 55  
update 76, 110, 115, 123, 133, 136, 142, 147, 153  
updateDisplayOrder 20  
updateMaxExamples 20  
updateWordList 153

## V

valueOf 78, 81

values 78, 81

Version 89

## W

word 64

WordData 65

WordEditController 27

WordEditPage 81

WordEditView 146

wordExists 28

wordFont 127

WordPictureSound 93

wordsLimitField 132

WPS 78

WPSController 30

WPSPage 80

WPSTableModel 149

WPSView 153

WPSViewData 98