Java OOP: Java Documentation*

Richard Baldwin

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

Abstract

Learn to use Oracle's Java documentation to realize the full potential of programming in Java.

1 Table of Contents

- Preface (p. 1)
  - Viewing tip (p. 2)
    - Figures (p. 2)
- Discussion (p. 2)
- Java Platform, Standard Edition 7 API Specification (p. 3)
- A universal documentation format (p. 7)
- Typical usage of the API documentation (p. 7)
- Summary (p. 7)
- Miscellaneous (p. 7)

2 Preface

This module is one of a series of modules designed to teach you about Object-Oriented Programming (OOP) using Java.

I cannot overemphasize the importance of Oracle's Java documentation [1] to aspiring Java programmers. The documentation package, which can be downloaded for installation and local access or accessed online, contains a wealth of information.

In my opinion, it is not possible to write Java programs of any substance without frequent reference to the documentation. No one can memorize everything that they need to know to be a successful Java programmer.

(Note that each time Oracle releases a new version of the Java Development Kit (JDK), they also release a new version of the documentation. Therefore, as new versions are released, the links provided in this document may become outdated and may not take you to the latest version of the documentation. However, you should be able to find the latest documentation via an online search.)

*Version 1.1: Nov 11, 2012 1:26 pm -0600

†http://creativecommons.org/licenses/by/3.0/

[1] http://docs.oracle.com/javase/7/docs/

2.1 Viewing tip

I recommend that you open another copy of this module in a separate browser window and use the following links to easily find and view the figures while you are reading about them.

2.1.1 Figures

- Figure 1 (p. 4) . Screen shot of the API specification at startup.
- Figure 2 (p. 6) . Screen shot of the API documentation after selecting the JButton class.

3 Discussion

Small core language, large class library

The *Java Platform Standard Edition* programming environment consists of a small core programming language and a large class library.

(As of the date of this writing, you can view The Java Language Specification [2] *online. This specification will tell you just about everything that most programmers need to know about the core language.)*

The size of the class library grows with the release of each new version of the JDK, because each new version provides capabilities that didn't exist in the previous version. New capabilities are added through the addition of new classes and new interfaces to the library.

The true power of Java resides in the class libraries

Once you understand how OOP is implemented in Java and you get beyond **while** loops, **if** statements, and other fundamental programming concepts, virtually all the power of Java resides in:

- Classes, interfaces, and methods in the various class libraries that you use directly.
- Classes, interfaces, and methods in the various class libraries that you extend.
- New classes, interfaces, and methods that you and others define.

You will always use material from Oracle's standard class libraries. You will often use material from other libraries that you create yourself, or that you obtain from sources outside of Oracle *(such as Barb Ericson's multimedia library* [3] *, for example)* .

Top-level online documentation

As of the date of this writing, you can view the top-level page of the documentation for *Java Platform Standard Edition 7* at http://docs.oracle.com/javase/7/docs/ [4] .

A complicated page

When you first view that documentation page, it may appear to be very complicated. Of all the material on the page, the most important is probably the *Application Programming Interface (API)* , which is currently available via a link on the right side of the page labeled Java SE API [5] .

However, you should not ignore the links to other information available on the top-level page [6] . Those links will often contain information that you need, and that information can make you more efficient in using the API documentation.

Search

If you follow the Search link at the top of the page, you will arrive at a search engine page [7] that gives you the ability to do a keyword search on the online documentation.

Downloading and installing Java

[2] http://java.sun.com/docs/books/jls/second_edition/html/j.title.doc.html
[3] http://coweb.cc.gatech.edu/mediaComp-plan/101
[4] http://docs.oracle.com/javase/7/docs/
[5] http://docs.oracle.com/javase/7/docs/api/index.html
[6] http://docs.oracle.com/javase/7/docs/
[7] http://docs.oracle.com/javase/search.html

Of particular interest to most newcomers should be the links titled Installation Instructions [8] and Java SE Downloads [9] .

The Java Tutorials

Also very important is the link to The Java Tutorials [10] . Once you reach those tutorials, of particular importance is the Path and CLASSPATH [11] tutorial. *(Many students trip on the path, the classpath, and the difference between the two.)*

4 Java Platform, Standard Edition 7 API Specification

The *API Specification* [12] is probably the most frequently used section of the entire documentation package.

A screen shot

Figure 1 (p. 4) shows a partial screen shot of what you should see when you first load the API Specification into your browser. *(Note that this screen shot was cropped to force it to fit into this publication format.)*

[8] http://docs.oracle.com/javase/7/docs/webnotes/install/index.html
[9] http://www.oracle.com/technetwork/java/javase/downloads/index.html
[10] http://docs.oracle.com/javase/tutorial/
[11] http://docs.oracle.com/javase/tutorial/essential/environment/paths.html
[12] http://docs.oracle.com/javase/7/docs/api/index.html

Screen shot of the API specification at startup.

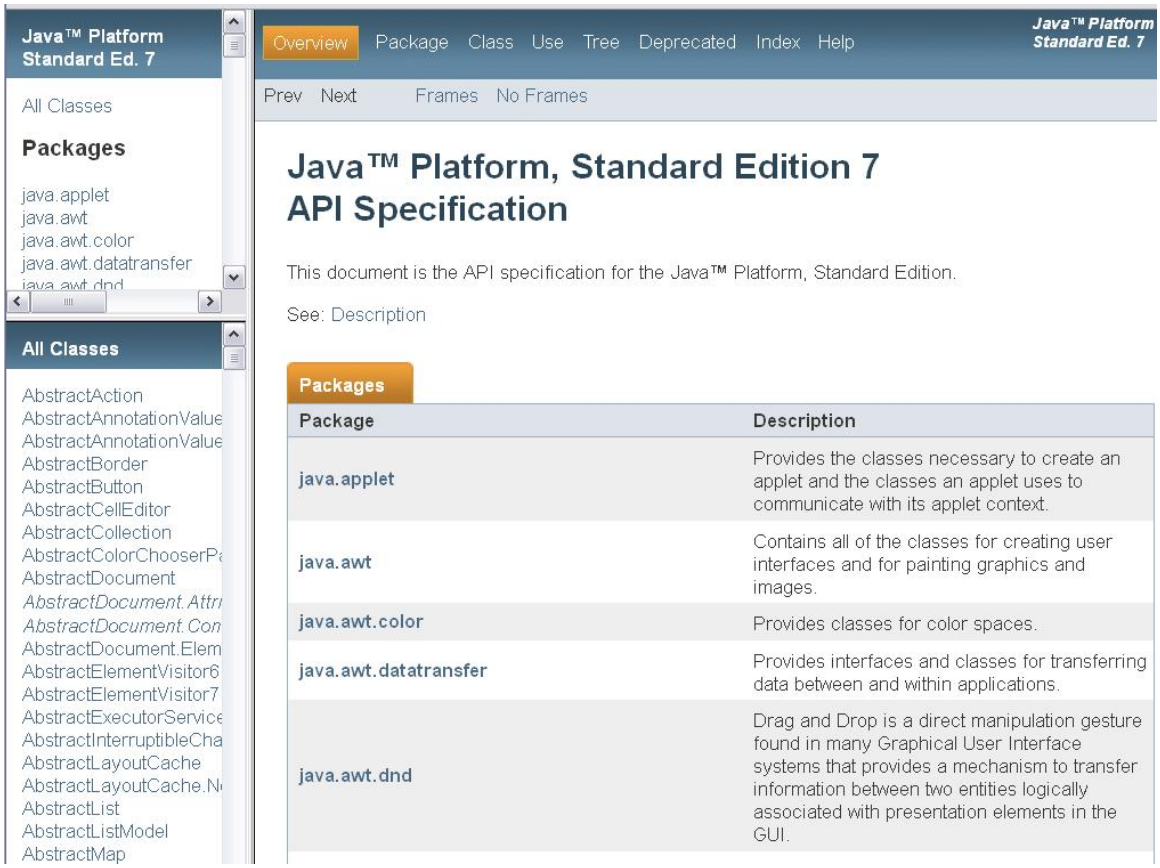

Figure 1: Screen shot of the API specification at startup.

Screen layout

The purpose of providing this screen shot is to give you an idea of the general layout of the material as it appears on your screen. As you can see, the layout consists of three frames when viewed in your HTML browser.

(A version without frames is also available by selecting the "No Frames" link at the top of the page. The version without frames is particularly useful for using the "page search" capability of your browser to find something on the page.)

The two leftmost frames

The upper-left frame contains a list of *packages* . The lower-left frame contains a list of the *classes* and *interfaces* that are contained in the package that is selected in the top-left frame.

(The default package selection in the upper-left frame is "All Classes".)

You make selections in these two frames *(or in the link bar at the top of the page)* to control the contents of the rightmost frame.

The rightmost frame

When you first access the API documentation *(and when you select* ***Overview*** *at the top of the page),*

the rightmost frame contains summary information about all of the packages.

When you select a class in the lower-left frame *(such as the* ***JButton*** *class for example)* , the rightmost frame contains hyperlinked information about that class, including:

- A visual class hierarchy diagram
- Interfaces implemented by the class
- Known subclasses of the class
- Text description of the class
- Summary of nested classes
- Summary of fields defined in the class
- Summary of fields inherited into the class
- Summary and detailed information on constructors defined in the class
- Summary and detailed information on methods defined in the class
- List of methods inherited into the class

Figure 2 (p. 6) shows a screen shot of the browser window after selecting the class named **JButton** in the lower-left frame. If you pull down the thumb in the scroll bar on the right side of the browser window, you will expose all of the information in the above list.

Screen shot of the API documentation after selecting the JButton class.

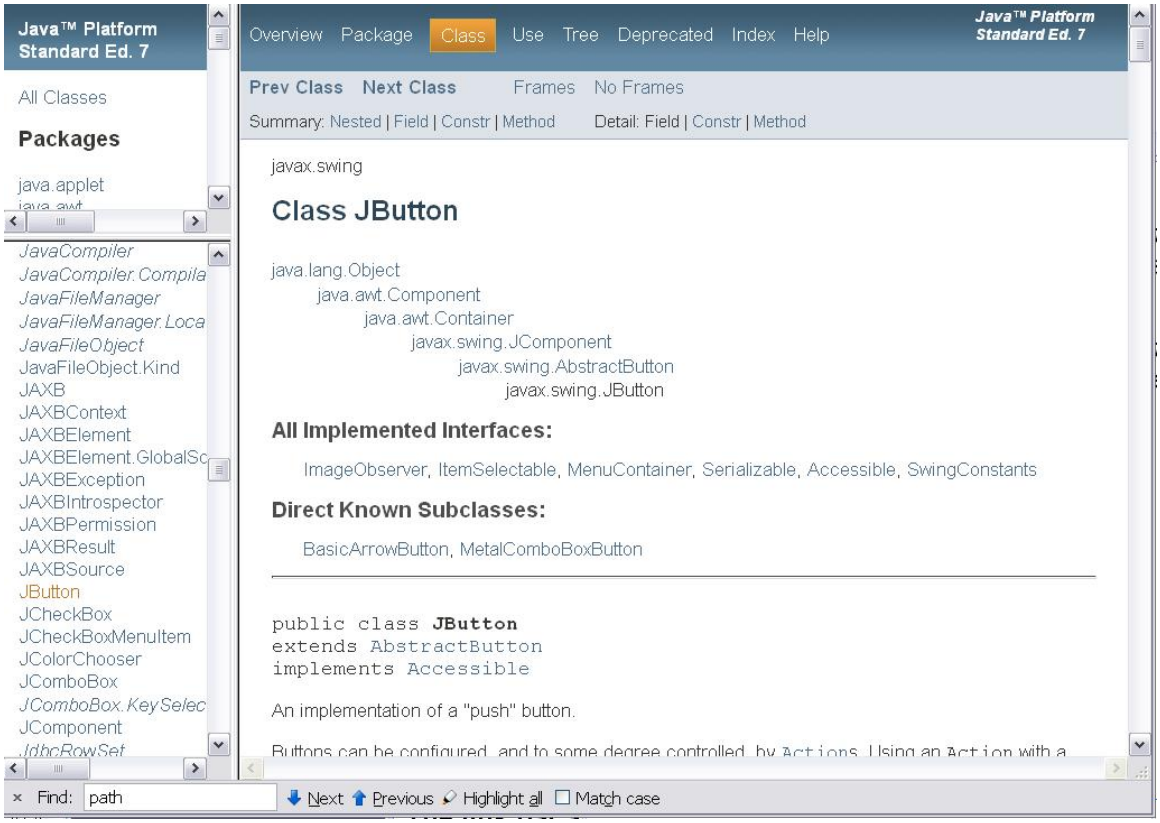

Figure 2: Screen shot of the API documentation after selecting the JButton class.

The link bar at the top of the page

It can sometimes be difficult to find what you are looking for in the documentation package. Referring back to Figure 2 (p. 6) , you can see a link bar at the top of the page in the rightmost frame. This link bar contains the following hyperlinks:

- **Overview** - provides summary information about packages *(shown in Figure 1 (p. 4))*.
- **Package** - provides a description of the package that contains the class selected in the lower-left frame.
- **Class** - provides a description of the class selected in the lower-left frame.
- **Use** - describes how the class selected in the lower-left frame is used in various packages.
- **Tree** - provides detailed inheritance hierarchy information for a selected package.
- **Deprecated** - provides a list of material that has been deprecated *(may not be supported in future versions)*
- **Index** - provides a hyperlinked alphabetized index of interfaces, classes, constructors, variables, and methods.
- **Help** - Describes how the API document is organized.

The various pages that are displayed by selecting these links can often help you to find what you are looking for. Probably the most useful and frequently consulted item in the above list is the *Index*.

The alphabetical index

The alphabetical index can be extremely useful if you know the full or partial spelling of what you are looking for, beginning with the first character.

For example, assume that you remember *(or you can surmise from what you know about Java properties)* that there is a method whose name begins with **getSystemLookAndFeel** , which returns the name of the *LookAndFeel* class that implements the native look and feel for a particular operating system. You can easily look this up under G in the alphabetical index.

What you will find when you look it up is a brief description of a method named **getSystemLookAndFeelClassName** that matches what you are looking for. The name of the method is also a hyperlink to the detailed description of the same method as it appears in the documentation for the class named **UIManager** .

5 A universal documentation format

The documentation for the API is created using a program named **javadoc.exe** that is contained in the JDK. Therefore, the API documentation for class libraries obtained from outside sources should have the same format as that described above *(assuming that the documentation was produced using javadoc.exe)* .

However, the program named **javadoc.exe** only controls the format of the documentation. It does not produce the explanatory content. It is the responsibility of the author of the class library to provide that content.

6 Typical usage of the API documentation

Typical usage of the API documentation consists of the following steps:

- Open the API documentation in your browser.
- Click the class of interest in the lower-left frame.
- Manually search the rightmost frame for summary information about fields, constructors, or methods of interest.
- Click the name of a field, constructor, or method in the summary section to open a detailed description of that field, constructor, or method.

7 Summary

There are several ways to search for useful information in the Oracle documentation. There is no single approach that will serve all of your needs to find information in the documentation. You simply need to become familiar with the different ways to search for information in the documentation and be prepared to use the approach that does the best job in each situation.

8 Miscellaneous

This section contains a variety of miscellaneous information.

NOTE: **Housekeeping material**

- Module name: Java OOP: Java Documentation
- File: Java3140.htm
- Published: 11/11/12

NOTE: **Disclaimers: Financial** : Although the Connexions site makes it possible for you to download a PDF file for this module at no charge, and also makes it possible for you to purchase a pre-printed version of the PDF file, you should be aware that some of the HTML elements in this module may not translate well into PDF.

I also want you to know that, I receive no financial compensation from the Connexions website even if you purchase the PDF version of the module.

In the past, unknown individuals have copied my modules from cnx.org, converted them to Kindle books, and placed them for sale on Amazon.com showing me as the author. I neither receive compensation for those sales nor do I know who does receive compensation. If you purchase such a book, please be aware that it is a copy of a module that is freely available on cnx.org and that it was made and published without my prior knowledge.

Affiliation : I am a professor of Computer Information Technology at Austin Community College in Austin, TX.

-end-