Interface Documentation

Software Engineering

Chair of Programming Methodology

Agenda for Today

Interface Documentation

1. Motivation
2. Concepts
3. Javadoc
4. Discussion

Objective:

- *Document your Code!*

Maintaining Interface Documentation

- Interface documentation starts with class diagrams
- Source code often automatically generated
- But then modified by the developers

Keep the documentation and the source current!

- **Integrate the interface documentation in the source**

Different Roles of Developers

- Class Implementer
  Realizes the functionality of a class
- Class Extender
  Develops specializations of classes
- Class User
  Uses a class to achieve a task

Visibility in UML

- Private: only for this class
  -
- Protected: this class and all descendant classes
  #
- Public: any class
  +

Constraints

- Preconditions and Postconditions for methods
- Invariants for classes

- UML: Object Constraint Language OCL
- Eiffel: Design-by-Contract
- Java: only assertions
- JML: full behavioral interface specification
- Javadoc: informal specification

Object Constraint Language

- Formal specification of constraints
- Boolean expressions
- Java/C++ like syntax
- Attach as notes or separate document

<<precondition>>
value < max

```
context Class::method(params) pre:
      value < max
```

- Many other features, e.g. sets and quantifiers

Java assertions

- Low-level, no distinction in pre-/post-conditions
- Invariants have to be modeled explicitly
- Can be turned on or off for execution

`assert` check;
`assert` check : msg;

Java Modeling Language JML

- Pre- and Post-Conditions
- Invariants
- Modifies-Clauses
- Many model classes
- Different tools to verify software

- See: `http://jmlspecs.org/`

Javadoc Comments

- Special Java comments for interface documentation
- Always starts with `/**`
- Two parts:
  - main description
  - tag section
- Description can contain HTML markup
- Block/standalone tags: @tag comment
- In-line tags: {@tag comment}

Example Javadoc

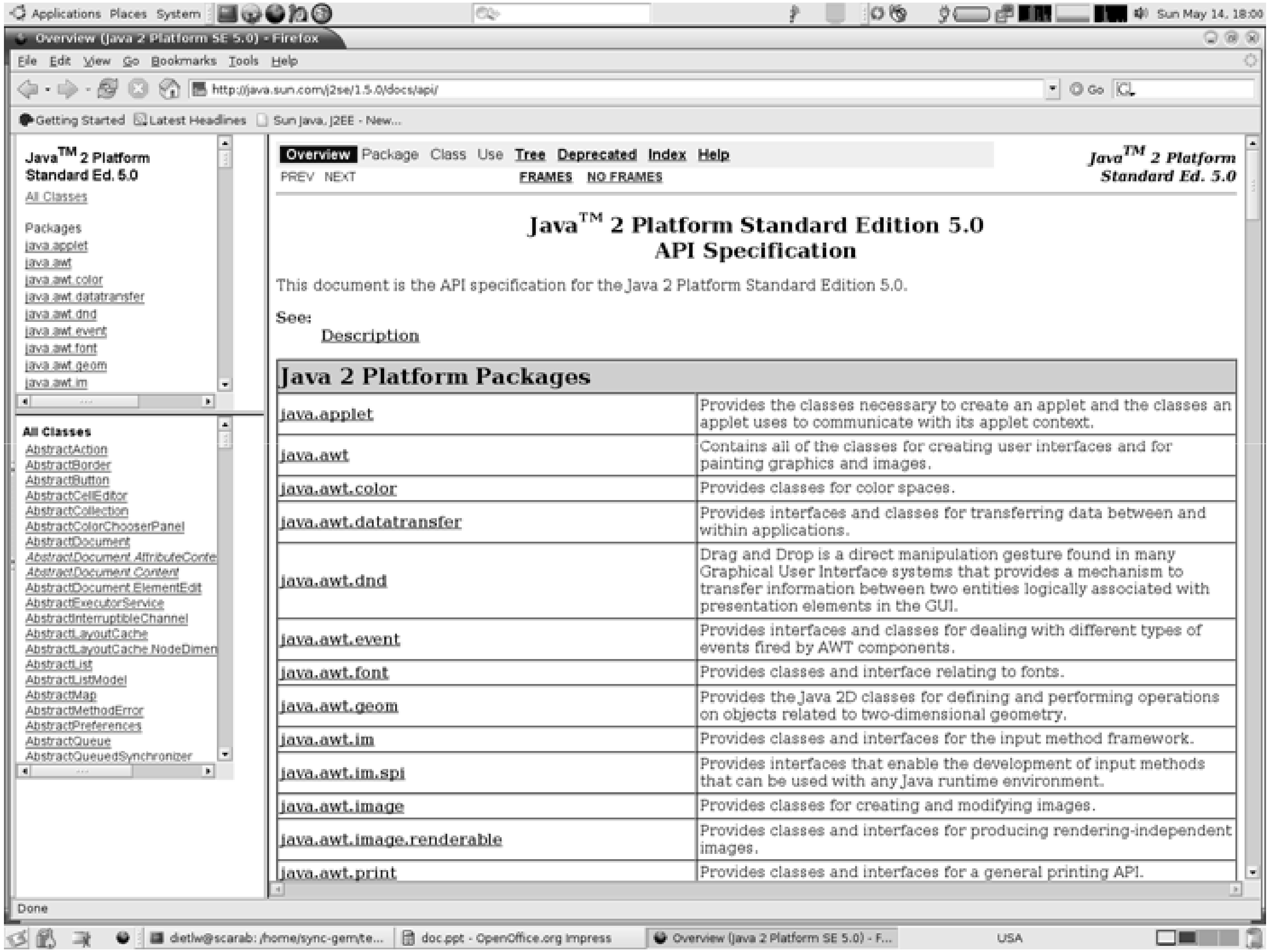
```
/**
 * Summary sentence till first dot.
 * Stars in the beginning optional.
 * <p>
 * More description.
 *
 * @param p the useful parameter
 */
void m(Param p) {
...
```

Javadoc Tool

- Goes through Java source files
- Extracts interface documentation
- Produces output file
- Output by default in HTML format
- Doclets for other output formats

```
javadoc [options][packagenames][sourcefiles]
```

Example Output


Applications Places System
Sun May 14, 18:00
Overview (Java 2 Platform SE 5.0) - Firefox
File Edit View Go Bookmarks Tools Help
http://java.sun.com/j2se/1.5.0/docs/api/
Go
Getting Started
Latest Headlines
Sun Java, J2EE - New...
Overview Package Class Use Tree Deprecated Index Help
PREV NEXT
FRAMES NO FRAMES
Done
dietlw@scarab: /home/sync-gem/te...
doc.ppt - OpenOffice.org Impress
Overview (Java 2 Platform SE 5.0) - F...
USA

Java^{TM} 2 Platform
Standard Ed. 5.0

All Classes

Packages
java.applet
java.awt
java.awt.color
java.awt.datatransfer
java.awt.dnd
java.awt.event
java.awt.font
java.awt.geom
java.awt.im

All Classes
AbstractAction
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeConte
AbstractDocument.Content
AbstractDocument.ElementEdit
AbstractExecutorService
AbstractInterruptibleChannel
AbstractLayoutCache
AbstractLayoutCache.NodeDimen
AbstractList
AbstractListModel
AbstractMap
AbstractMethodError
AbstractPreferences
AbstractQueue
AbstractQueuedSynchronizer

Java^{TM} 2 Platform
Standard Ed. 5.0

Java^{TM} 2 Platform Standard Edition 5.0
API Specification

This document is the API specification for the Java 2 Platform Standard Edition 5.0.

See:
Description

Java 2 Platform Packages	
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
java.awt.dnd	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.
java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
java.awt.font	Provides classes and interface relating to fonts.
java.awt.geom	Provides the Java 2D classes for defining and performing operations on objects related to two-dimensional geometry.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
java.awt.print	Provides classes and interfaces for a general printing API.

NetBeans Integration

- Helps in creating documentation
- Create Javadoc in separate directory
  Menu: Run → Generate Javadoc for “...”
- Help to document all features of a class
  Menu: Tools → Analyze Javadoc

References

- http://java.sun.com/javase/6/docs/technotes/guides/javadoc/index.html
- http://java.sun.com/j2se/javadoc/
- http://java.sun.com/j2se/javadoc/writingdoccomments/index.html
- http://java.sun.com/javase/6/docs/api/

Discussion

- Look at Javadoc Guidelines
- Ensure complete docs with Auto Comment
- Think of the readers and their different roles