

Wybrane narzędzia wspomagające dokumentowanie programu

Krzysztof Gogól
Paweł Konieczny

Uniwersytet Warszawski

18 III 2008

Plan prezentacji

- 1 Wprowadzenie
- 2 Konwencje
 - Wstęp
 - Dobre praktyki
 - Przykłady
- 3 Doxygen
 - Doxygen - co to?
 - Doxygen - pierwsze kroki
 - Doxygen - podsumowanie
- 4 Javadoc
 - Wprowadzenie
 - Działanie
- 5 Inne programy
 - Rozwiązanie komercyjne
 - Koniec

Generator dokumentacji

Generator dokumentacji (documentation generator) jest narzędziem programistycznym, którego zadaniem jest tworzenie dokumentacji na potrzeby programistów lub końcowych użytkowników systemu na podstawie zbioru specjalnie zakomentowanych plików źródłowych z kodem lub plików binarnych.

Typy dokumentów

Wyróżniamy następujące typy dokumentów:

- Batch documents
- Interactive documents
- Text block correspondence
- Forms

Każdy dokument należy do jednej z powyższych kategorii.

Typy dokumentów

Wyróżniamy następujące typy dokumentów:

- Batch documents
- Interactive documents
- Text block correspondence
- Forms

Każdy dokument należy do jednej z powyższych kategorii.

Typy dokumentów

Wyróżniamy następujące typy dokumentów:

- Batch documents
- Interactive documents
- Text block correspondence
- Forms

Każdy dokument należy do jednej z powyższych kategorii.

Typy dokumentów

Wyróżniamy następujące typy dokumentów:

- Batch documents
- Interactive documents
- Text block correspondence
- Forms

Każdy dokument należy do jednej z powyższych kategorii.

Typy dokumentów

Wyróżniamy następujące typy dokumentów:

- Batch documents
- Interactive documents
- Text block correspondence
- Forms

Każdy dokument należy do jednej z powyższych kategorii.

Typy dokumentów

Wyróżniamy następujące typy dokumentów:

- Batch documents
- Interactive documents
- Text block correspondence
- Forms

Każdy dokument należy do jednej z powyższych kategorii.

Wstęp

Podobnie jak w innych dziedzinach życia, także przy pisaniu komentarzy poddanie się niewielkiej liczbie ograniczeń może zdziałać cuda. Podczas pisania dobrze jest mieć na uwadze parę zasad, dzięki którym dokumentacja staje się jasna i przejrzysta.

Formułowanie treści

- Pierwsze zdanie komentarza zazwyczaj wykorzystywane jest jako podsumowanie całego bloku, powinno więc być jednocześnie krótkie i samowystarczalne, ale także zawierać wszystkie kluczowe informacje.
- Dla ujednoczenia formy dokumentu, warto trzymać się zasady, że opis metody lub atrybutu pisany jest w 3-ciej osobie.
- Unikać zbędnych wprowadzeń, np: "Sprawdza poprawność ruchu" zamiast "Ta metoda sprawdza poprawność ruchu".

Drobne uwagi

- Unikać odnośników do dobrze znanych elementów, do pozostałych wystarczy po jednym odnośniku (np ciągłe tworzenie odnośników do klasy String jest złym pomysłem).
- Odwołując się do określonej metody unikać pisania nawiasów i zestawu argumentów - chyba że chodzi o konkretną metodę spośród wielu o tej samej nazwie (np `sprawdzZakresRuchu`, ale `sprawdzZakresRuchu(int, int, int, Pionek)`).
- Unikać ujawniania szczegółów implementacyjnych - chyba że zależą one od platformy - wówczas można nadmienić w jaki sposób kawałek kodu działa na określonej platformie (np *W systemie Windows działanie metody `loadLibrary` jest takie samo jak metody `loadLibrary` z Windows API.*)

Jak pisać

- Komentarze do obiektu powinny być umieszczone bezpośrednio przed nim i zostać zawarte między znacznikami `/**` oraz `*/`
- Pisząc komentarz można używać elementów formatowania HTML, np. znacznika `<p>` do oznaczania akapitów.
- Odnośniki do innych części dokumentacji tworzy się używając znacznika `{@link URL}`, np. bardzo niewskazane `{@link String}`.
- Między opisem, a listą tzw. tagów należy umieścić pojedynczą wolną linię.
- W metodzie implementującej interfejs lub przeddefiniowującej jakąś metodę, dokumentacja jest automatycznie generowana z odpowiednim odnośnikiem (np. `overrides` lub `implements`), nie trzeba zatem kopiować komentarzy.

Tagi

Lista tagów, które można umieścić w komentarzu (powinno się przestrzegać podanej kolejności):

- @param - argument metody.
- @return - wartość wyniku.
- @throws - wyjątki rzucone przez metodę.
- @author - autor (tylko w klasach i interfejsach)
- @version - wersja (tylko w klasach i interfejsach)
- @see - obiekty, z którymi warto się zapoznać dla zrozumienia działania metody.
- @deprecated - oznaczanie elementu jako przestarzały.

Przykład

```
/**
 * Returns an Image object that can then be painted on the screen.
 * The url argument must specify an absolute {@link URL}. The name
 * argument is a specifier that is relative to the url argument.
 * <p>
 * This method always returns immediately, whether or not the
 * image exists. When this applet attempts to draw the image on
 * the screen, the data will be loaded. The graphics primitives
 * that draw the image will incrementally paint on the screen.
 *
 * @param url an absolute URL giving the base location of the image
 * @param name the location of the image, relative to the url argument
 * @return the image at the specified URL
 * @see Image
 */
public Image getImage(URL url, String name) {
    try {
        return getImage(new URL(url, name));
    } catch (MalformedURLException e) {
        return null;
    }
}
```

Kontrprzykład

```
/**  
 * Sets the tool tip text.  
 *  
 * @param text the text of the tool tip  
 */  
public void setToolTipText(String text) {}
```


Czym jest Doxygen?

Doxygen jest systemem tworzenia dokumentacji obsługujący języki:

- C++,
- C,
- Java,
- Objective-C,
- Python,
- IDL (Corba and Microsoft flavors),
- Fortran,
- VHDL,
- PHP,
- C#

Po co z niego korzystać?

- Doxygen generuje dokumentację w postaci strony (w HTMLu) oraz jako maunal napisany w Latex'u na podstawie dowolnego zbioru plików źródłowych. Wynikiem jego działania mogą również być pliki RTF (MS-Word), PostScript, PDF lub strony uniksowego podręcznika.
- Dzięki odpowiednim ustawieniom możliwe jest wygenerowanie dokumentacji opisującej niezakomentowany kod.
- Możliwość tworzenia dokumentacji nie tylko na podstawie plików z kodem. Dodatkowo istnieje możliwość zilustrowania zależności między klasami za pomocą różnych diagramów.

Po co z niego korzystać?

- Doxygen generuje dokumentację w postaci strony (w HTMLu) oraz jako maunal napisany w Latex'u na podstawie dowolnego zbioru plików źródłowych. Wynikiem jego działania mogą również być pliki RTF (MS-Word), PostScript, PDF lub strony uniksowego podręcznika.
- Dzięki odpowiednim ustawieniom możliwe jest wygenerowanie dokumentacji opisującej niezakomentowany kod.
- Możliwość tworzenia dokumentacji nie tylko na podstawie plików z kodem. Dodatkowo istnieje możliwość zilustrowania zależności między klasami za pomocą różnych diagramów.

Po co z niego korzystać?

- Doxygen generuje dokumentację w postaci strony (w HTMLu) oraz jako maunal napisany w Latex'u na podstawie dowolnego zbioru plików źródłowych. Wynikiem jego działania mogą również być pliki RTF (MS-Word), PostScript, PDF lub strony uniksowego podręcznika.
- Dzięki odpowiednim ustawieniom możliwe jest wygenerowanie dokumentacji opisującej niezakomentowany kod.
- Możliwość tworzenia dokumentacji nie tylko na podstawie plików z kodem. Dodatkowo istnieje możliwość zilustrowania zależności między klasami za pomocą różnych diagramów.

Na jakich systemach działa?

Doxygen jest rozwijany pod Linuxem i Mac OS X, ale jest programem łatwo przenośnym. Działa również pod innymi Unix flavors a nawet pod Windowsem.

Przykładowe projekty

Przykłady wykorzystania Doxygen'a:

- The KDevelop API Documentation
- The Xerces-C++ Documentation
- D-Bus documentation
- Yahoo Music Engine
- IBM's International Components for Unicode

Przykładowe projekty

Przykłady wykorzystania Doxygen'a:

- The KDevelop API Documentation
- The Xerces-C++ Documentation
- D-Bus documentation
- Yahoo Music Engine
- IBM's International Components for Unicode

Przykładowe projekty

Przykłady wykorzystania Doxygen'a:

- The KDevelop API Documentation
- The Xerces-C++ Documentation
- D-Bus documentation
- Yahoo Music Engine
- IBM's International Components for Unicode

Przykładowe projekty

Przykłady wykorzystania Doxygen'a:

- The KDevelop API Documentation
- The Xerces-C++ Documentation
- D-Bus documentation
- Yahoo Music Engine
- IBM's International Components for Unicode

Przykładowe projekty

Przykłady wykorzystania Doxygen'a:

- The KDevelop API Documentation
- The Xerces-C++ Documentation
- D-Bus documentation
- Yahoo Music Engine
- IBM's International Components for Unicode

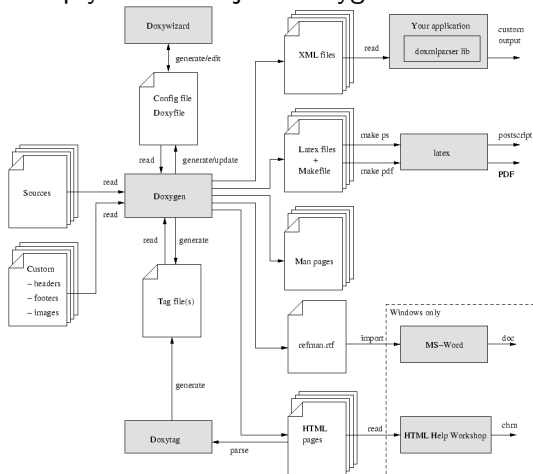
Przykładowe projekty

Przykłady wykorzystania Doxygen'a:

- The KDevelop API Documentation
- The Xerces-C++ Documentation
- D-Bus documentation
- Yahoo Music Engine
- IBM's International Components for Unicode

Przeływ informacji

Przeływ informacji w Doxygen



Konfiguracja

Doxygen generuje dokumentację na podstawie pliku konfiguracyjnego. Każdy projekt powinien posiadać własny plik konfiguracyjny. Projekt może składać się z jednego pliku lub drzewa katalogów.

Do łatwego i szybkiego tworzenia pliku konfiguracyjnego służy komenda:

```
doxygen -g <config-file >
```

Plik konfiguracyjny Doxygena ma podobną budowę do Makefile'a.

Składa się z wielu przypisań postaci:

```
TAGNAME = VALUE lub
```

```
TAGNAME = VALUE1 VALUE2 ...
```

Doxywizard

Dla osób, które nie chcą konfigurować programów za pośrednictwem edytorów tekstowych powstał DOXYWIZARD – GUI systemu Doxygen. Umożliwia on tworzenie, czytanie i pisanie do pliku konfiguracyjnego doxygen za pośrednictwem miłych dla oka okienek.

Pierwsze kroki

Do generowania dokumentacji służy komenda:

```
doxygen <config-file >
```

W zależności od ustawień program wygeneruje katalog html, rtf, latex, xml lub man w wyjściowym katalogu. Tak jak sugerują nazwy, te katalogi zawierają wygenerowaną dokumentację w HTML, RTF, XML lub Unix-Man page format.

Domyślnie OUTPUT jest umieszczany w katalogu, w którym jest uruchamiany Doxygen. Ustawienia te można zmienić w pliku konfiguracyjnym (*OUTPUT DIRECTORY*).

Podsumowanie

- Doxygen jest prostym w obsłudze narzędziem wspomagającym dokumentację.
- Jest w pełni zgodny z Javadoc i QT-doc.
- Dzięki możliwości tworzenia dokumentacji nawet na podstawie nieudokumentowanego kodu, w łatwy i przyjazny sposób można poznać strukturę projektu.

Podsumowanie

- Doxygen jest prostym w obsłudze narzędziem wspomagającym dokumentację.
- Jest w pełni zgodny z Javadoc i QT-doc.
- Dzięki możliwości tworzenia dokumentacji nawet na podstawie nieudokumentowanego kodu, w łatwy i przyjazny sposób można poznać strukturę projektu.

Podsumowanie

- Doxygen jest prostym w obsłudze narzędziem wspomagającym dokumentację.
- Jest w pełni zgodny z Javadoc i QT-doc.
- Dzięki możliwości tworzenia dokumentacji nawet na podstawie nieudokumentowanego kodu, w łatwy i przyjazny sposób można poznać strukturę projektu.

Gdzie szukać informacji?

- Strona główna Doxygen'a
<http://www.stack.nl/~dimitri/doxygen/>
- Podręcznik użytkownika
<http://www.stack.nl/~dimitri/doxygen/manual.html>
- Lista wszystkich poleceń specjalnych
<http://www.stack.nl/~dimitri/doxygen/commands.html>

Gdzie szukać informacji?

- Strona główna Doxygen'a
<http://www.stack.nl/~dimitri/doxygen/>
- Podręcznik użytkownika
<http://www.stack.nl/~dimitri/doxygen/manual.html>
- Lista wszystkich poleceń specjalnych
<http://www.stack.nl/~dimitri/doxygen/commands.html>

Gdzie szukać informacji?

- Strona główna Doxygen'a
<http://www.stack.nl/~dimitri/doxygen/>
- Podręcznik użytkownika
<http://www.stack.nl/~dimitri/doxygen/manual.html>
- Lista wszystkich poleceń specjalnych
<http://www.stack.nl/~dimitri/doxygen/commands.html>

Wstęp

Javadoc jest częścią komponentu SDK do tworzenia aplikacji w języku Java firmy Sun Microsystems. Oznacza to, że nie wymaga instalowania żadnego dodatkowego oprogramowania i można się spodziewać, że narzędzia programistyczne dla środowiska Java będą go obsługiwać. Potrafi generować nie tylko dokumentację API programu, ale także dla całego pakietu lub zastawu pakietów. Działa pod wszystkimi popularnymi systemami operacyjnymi, jednak rozpoznaje tylko język Java, zaś standardowo generuje dokumentację jedynie w formacie HTML.

Opis

Javadoc wymaga kompilatora Javy, aby działać. Najpierw wykonuje część instrukcji `javac`, aby wygenerować drzewo analizy składniowej (parse tree). Na jego podstawie powstaje dokumentacja. Wykonywane instrukcje kompilatora zupełnie ignorują kod źródłowy, możliwe jest więc wygenerowanie dokumentacji dla pustego interfejsu, bez treści metod. Zastosowanie takiej techniki umożliwia tworzenie dokumentacji dokładnie odzwierciedlającej implementację - np. kompilator sam tworzy domyślne konstruktory dla klas, jeśli żaden nie został zdefiniowany, po czym umieszcza go w opisie API programu.

Konfiguracja

Konfigurowanie działania Javadoc wymaga zmiany jego kodu źródłowego, a konkretnie edycji plików

`Standard.java`, `ClassWriter.java`, `HtmlStandardWriter.java`,
`PackageWriter.java`, `PackageIndexWriter.java`

Zazwyczaj jednak domyślna konfiguracja jest dobrze wypracowanym kompromisem.

Zmiana wyglądu dokumentacji jest już dużo łatwiejsza - jako że dokument jest w formacie HTML, wystarczy wykorzystać pliki ze stylami CSS. W katalogu z wygenerowaną dokumentacją należy umieścić plik `stylesheet.css` i w standardowy sposób modyfikować wygląd poszczególnych znaczników HTML.

Za mało informacji?

- Strona domowa Javadoc
<http://java.sun.com/j2se/javadoc/>
- Ładny opis na Wikipedii
<http://en.wikipedia.org/wiki/Javadoc>
- Poradnik odnośnie konwencji pisania
<http://java.sun.com/j2se/javadoc/writingdoccomments/>

Za mało informacji?

- Strona domowa Javadoc
<http://java.sun.com/j2se/javadoc/>
- Ładny opis na Wikipedii
<http://en.wikipedia.org/wiki/Javadoc>
- Poradnik odnośnie konwencji pisania
<http://java.sun.com/j2se/javadoc/writingdoccomments/>

Za mało informacji?

- Strona domowa Javadoc
<http://java.sun.com/j2se/javadoc/>
- Ładny opis na Wikipedii
<http://en.wikipedia.org/wiki/Javadoc>
- Poradnik odnośnie konwencji pisania
<http://java.sun.com/j2se/javadoc/writingdoccomments/>

Doc-O-Matic

Jest to komercyjny generator dokumentacji, działający wyłącznie pod systemem Windows. Jego zalety:

- Znaczną liczbę rozpoznawanych języków programowania.
- Spory wybór formatów wyjściowych dokumentacji.
- Jednym z formatów wyjściowych jest WinHelp.
- Potrafi naśladować działanie innych generatorów, aby uniknąć zmieniania konwencji pisania.
- Posiada miły dla oka interfejs.
- Dostarcza narzędzi do wizualizacji hierarchii klas i elastyczny system raportowania.

Ze względu na platformę

	Windows	Mac OS X	Linux	BSD	Unix
Autoduck	Yes	No	No	No	No
BI Documenter	Yes	No	No	No	No
DB Manual	Yes	No	No	No	No
DBScribe	Yes	No	No	No	No
classdoc	Yes	Yes	Yes	Yes	Yes
CppDoc	Yes	No	Yes	No	No
Ddoc	Yes	Yes ³	Yes	Yes ³	No
Doc-O-Matic	Yes	No	No	No	No
DOC++	Yes	No?	Yes	No?	No?
Document! X	Yes	No	No	No	No
Doxygen	Yes	Yes	Yes	Yes	Yes
Epydoc	Yes	Yes	Yes	Yes	Yes
GenHelp	Yes	No	No	No	No
HeaderDoc	No	Yes	Yes	Yes	Yes
Help Generator	Yes	No	No	No	No
Javadoc	Yes	Yes	Yes	Yes	Yes
JSDoc	Yes	Yes	Yes	Yes	Yes
KDOC	Partial	Yes	Yes	Yes	Yes
Natural Docs	Yes	Yes	Yes	Yes	Yes
NDoc	Yes	No	No	No	No
phpDocumentor	Yes	Yes	Yes	Yes	Yes
Project Analyzer	Yes	No	No	No	No
RDoc	Yes	Yes	Yes	Yes	Yes

Ze względu na obsługiwane języki

	C/C++	Java	C#	VB / VBScript Delphi / Pascal	Ada	D	IDL	.NET ¹	Access
Autoduck	Yes	No	No	Yes	No	No	No	No	No
BI Documenter	No	No	No	No	No	No	No	Yes	No
DB Manual	No	No	No	No	No	No	No	Yes	No
DBScribe	No	No	No	No	No	No	No	No	No
classdoc	No	Yes	No	No	No	No	No	No	No
CppDoc	Yes	No	No	No	No	No	No	No	No
Ddoc	No	No	No	No	No	Yes	No	No	No
Doc-O-Matic	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No
DOC++	Yes	Yes	No	No	No	No	No	Yes	No
Document! X	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes
Doxygen	Yes	Yes	Yes	No	No	No	Partial	Yes	No
Epydoc	No	No	No	No	No	No	No	No	No
GenHelp	Yes	No	Yes	No	Yes	No	No	Yes	No
HeaderDoc	Yes	Yes	No	No	Yes	No	No	No	No
Help Generator	Yes	No	Yes	Yes	No	No	No	Yes	Yes
Javadoc	No	Yes	No	No	No	No	No	No	No
JSDoc	No	No	No	No	No	No	No	No	No
KDDC	Yes	No	No	No	No	No	Yes	No	No
Natural Docs	Partial	Partial	Yes	Partial	Partial	Partial	No	No	No
NDoc	No	No	Yes	No	No	No	No	Yes	No
phpDocumentor	No	No	No	No	No	No	No	No	No
Project Analyzer	No	No	No	Yes	No	No	No	Yes	No
RDoc	No	No	No	No	No	No	No	No	No

Ze względu na obsługiwane języki - c.d.

	PHP	Perl	Python	Ruby	JavaScript	ActionScript	PL/SQL	Tcl	Haskell	Any With Comments ²
Autoduck	No	No	No	No	No	No	No	No	No	No
BI Documenter	No	No	No	No	No	No	No	No	No	No
DB Manual	No	No	No	No	No	No	No	No	No	No
DBScribe	No	No	No	No	No	No	Yes	No	No	No
classdoc	No	No	No	No	No	No	No	No	No	No
CppDoc	No	No	No	No	No	No	No	No	No	No
Ddoc	No	No	No	No	No	No	No	No	No	No
Doc-O-Matic	No	No	No	No	Yes	No	No	No	No	No
DOC++	No	No	No	No	No	No	No	No	No	No
Document! X	No	No	No	No	No	No	Yes	No	No	No
Doxygen	Yes	No	Yes	No	No	No	No	No	No	No
Epydoc	No	No	Yes	No	No	No	No	No	No	No
GenHelp	No	No	No	No	No	No	No	No	No	No
HeaderDoc	Yes	Yes	No	No	Yes	No	No	No	No	No
Javadoc	No	No	No	No	No	No	No	No	No	No
JSDoc	No	No	No	No	Yes	No	No	No	No	No
KDOC	No	No	No	No	No	No	No	No	No	No
Natural Docs	Partial	Yes	Partial	Partial	Partial	Yes	Partial	Partial	No	Yes
NDoc	No	No	No	No	No	No	No	No	No	No
phpDocumentor	Yes	No	No	No	No	No	No	No	No	No
Project Analyzer	No	No	No	No	No	No	No	No	No	No
RDoc	No	No	No	Yes	No	No	No	No	No	No

Ze względu na format wyjściowy

	HTML	CHM	RTF	PDF	LaTeX	PostScript	man pages	DocBook	XML
Autoduck	No	No	Yes	No	No	No	No	No	No
BI Documenter	Yes	Yes	No	No	No	No	No	No	No
DB Manual	Yes	Yes	No	No	No	No	No	No	No
DBScribe	Yes	Yes	Yes	No	No	No	No	No	No
classdoc	Yes	No	No	No	No	No	No	No	No
CppDoc	Yes	No	No	No	No	No	No	No	No
Ddoc	Yes	Yes ⁴	No	Yes ⁴	Yes ⁴	Yes ⁴	Yes ⁴	No	Yes ⁴
Doc-O-Matic	Yes	Yes	Yes	Yes	No	No	No	No	Yes
DOC++	Yes	No	No	No	Yes	No	No	No	No
Document! X	Yes	Yes	No	No	No	No	No	No	No
Doxygen	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Epydoc	Yes	No	No	Yes	Indirectly ⁵	Indirectly ⁵	No	No	No
GenHelp	Yes	Yes	Yes	No	No	No	No	No	No
HeaderDoc	Yes	No	No	No	No	No	Yes	No	Yes
Help Generator	Yes	Yes	Yes	No	No	No	No	No	Yes
Javadoc	Yes	No	No	No	No	No	No	No	No
JSDoc	Yes	No	No	No	No	No	No	No	(planned)
KDOC	Yes	No	No	No	No	No	No	No	No
Natural Docs	Yes	No	No	No	No	No	No	No	No
NDoc	Yes	Yes	No	No	No	No	No	No	No
phpDocumentor	Yes	Yes	No	Yes	No	No	No	Yes	Yes
Project Analyzer	Yes	Yes	Yes	Yes	No	No	No	No	No
RDoc	Yes	No	No	No	No	No	No	No	No

Koniec

Dziękujemy za uwagę.