Doxygen

21. November 2003

Was ist Doxygen?

- erstellt Dokumentation in HTML, Latex, PDF oder Unix man pages
- kann speziell formatierte Kommentare aus Quelltext extrahieren
- kann die Code Struktur aus großen undokumentierten Quellen extrahieren

Warum Kommentare?

- Programme sollen von anderen verstanden werden
- 80% der Kosten/Zeit liegen in Software-Wartung
- Quelltext ist die einzige verlässliche Dokumentation

Was macht diese Funktion? (1)

```
 /undokumentierte Funktion
float calculate(float a,float b, float c){
    return (1/(sqrt(2*3.1415)*c))*exp(((a-b)*(a-b))/(2*c*c))
}
```

Was macht diese Funktion? (2)

```
/** Berechnet die Dichte der
Normalverteilung mit Erwartungswert
mu und Standardabweichung sigma an Stelle s. */

float Normalverteilung(float s, float mu, float sigma){
    return (1/(sqrt(2*3.1415)*sigma))*
       exp((s-mu)*(s-mu))/(2*sigma*sigma));
}
```

Konfigurationsdatei (1)

- jedes Projekt hat eine Konfigurationsdatei
- der Aufruf "doxygen -g" erstellt Standarddatei "Doxyfile"
- Format dieser Datei:

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

- Konfigurationsdatei ist sehr gut dokumentiert!

Konfigurationsdatei (2)

- wichtige Tagnames:
  - INPUT - Definition der Quelldateien und/oder der Quellverzeichnisse
  - FILE_PATTERNS - definiert wildcard-patterns der Quelldateien
  - RECURSIVE - gibt an ob auch Unterverzeichnisse gelsesen werden sollen
  - EXTRACT_ALL - auch nicht dokumentierte Elemente kommen in Dokumentation
  - OUTPUT_DIRECTORY - in diesem Verzeichnis wird die Dokumentation gespeichert

Dokumentation des Programmcodes (1)

- vor einer Programmeinheit (z.B. Member, Klasse, Namespace)
  - Kurzbeschreibung (brief description)
  - ausführliche Beschreibung (detailed description)

```
/**
    \brief Dies ist die Kurzbeschreibung der Einheit.

    Dies ist die etwas ausführlichere Beschreibung der
    Programmeinheit.   Beispiel: normales_Kommentar.c
*/
...
```

Dokumentation des Programmcodes (2)

- nach einer Programmeinheit (z.B. *Datei*, Member, Klasse, Namespace)

```
int numberOfFingers; /**< die Anzahl der Roboterfinger. */
```

Alternativen

- statt /** Kommentar */ kann auch

```
/*! Kommentar */

///
/// Kommentar
///

//!
//! Kommentar
//!
```

verwendet werden

Mehrere Möglichkeiten für brief description

```
/**
    \brief Kurzbeschreibung

    Ausführliche Beschreibung.
*/
```

oder

```
///Kurzbeschreibung
/**
    Ausführliche Beschreibung.
*/
```

Automatische Kurzbeschreibung

- wenn Tagname JAVADOC_AUTOBRIEF auf YES gsetzt wurde

```
/** Dies ist Kurzbeschreibung. Und
    dies ist schon die ausführliche
    Beschreibung.
*/
```

Structural Commands (1)

- man kann alle Programmeinheiten auch von beliebigen Stellen aus dokumentieren
- dazu benötigt man structural commands
- Wichtig für Dateien!:

```
/** \file
    Beschreibung der Datei
*/
```

Beispiel: file_kommentierung.c

Structural Commands (2)

- Beispiel \class

```
/** \class Testklasse
    \brief Kurzbeschreibung der Klasse

    Ausfuehrliche Beschreibung der Klasse
*/
```

Weitere structural commands

\struct	kommentieren eines C-struct
\enum	kommentieren eines enumeration Typs
\var	kommentieren einer Variablen, eines typedefs oder eines enum-Wertes
\def	kommentieren eines #define.
\namespace	kommentieren eines namespace
\fn	kommentieren einer Funktion

Special commands

- für Funktionen
  - \param - kommentiert ein Argument der Funktion
  - \sa - generiert einen Link auf eine andere Funktion (see also)
  - \return - kommentiert den return-Wert der Funktion

Special commands

- Beispiel:

```
/**
    Addition zweier Werte
    \param a Erster Wert
    \param b Zweiter Wert
    \return Summe der beiden Werte a und b
    \sa sub, mul, div
*/
int add(int a, int b){
    return a+b;
}
```

Special commands

- Beispiel Code Fragment einbinden mit \example

```
/** \example example.cc
    Dies ist ein Beispielprogarmm.
*/
```

Special commands

- Hauptseite gestalten mit \mainpage

```
/** \mainpage Titel der Hauptseite
    \section sec1 Erste Überschrift
    ...Text...
*/
```

Beispiel: mainpage.c

Special commands

- zusätzliche Seite erstellen mit \page

```
/** \page page1 Titel der zusätzlichen Seite
\section sec1 Erste Überschrift
    ...Text...
*/
```

- es erscheint ein neuer Menüpunkt "Related Pages"

Beispiel: add_page.c

Elemente in den Kommentaren: Listen

- es können sehr leicht Listen eingefügt werden: Beispiel: liste.c

```
<ul>
  <li>Listenelement 1</li>
  <li>Listenelement 2</li>
  <ul>
    <li>Unterlistenelement 1</li>
    <li>Unterlistenelement 2</li>
  </ul>
  <li>Listenelement 3</li>
</ul>
```

Elemente in den Kommentaren: Formeln (1)

- auch mathematische Formeln können eingefügt werden
- diese werden als Latex-Formeln in den Kommentar geschrieben
- inline-Formeln werden eingeleitet und beendet durch \f$
- separate Formeln werden eingeleitet durch \f[und beendet mit \f]

Elemente in den Kommentaren: Formeln (2)

```
/**
    \f[
       N_s(\mu,\sigma)=
      \frac{1}{\sqrt{2 \pi} \sigma}\exp(\frac{(s-\mu)^2}{2 \sigma^2})
    \f]
*/
```

$$N_s(\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}\exp(\frac{(s-\mu)^2}{2\sigma^2})$$

Elemente in den Kommentaren: HTML-Formatierungen

- viele HTML-Formatierungen werden unterstützt: z.B.

```
<B>Fettdruck</B>
<BR>Zeilenumbruch
<I>Kursive Schrift</I>

<CENTER>Text zentrieren</CENTER>
<CODE>Alle Zeichen haben die gleiche Breite</CODE>
<IMG SRC='image.png'> (Nur im HTML-Output)

...... und viele mehr!
```

Generierung der Dokumentation

- Aufruf von "doxygen [config-file]" startet Generierung
- Dokumentation wird im Verzeichnis OUTPUT_DIRECTORY abgelget

Quellen

`http://www.stack.nl/~dimitri/doxygen/`

bzw.

`http://www.stack.nl/~dimitri/doxygen/manual.html`