

Doxygen ist ein Tool zum automatischen Erstellen von Dokumentationen aus speziellen Quelltext-Kommentaren. Nach verschiedenen Erweiterungen ist *Doxygen* inzwischen in der Lage solche Code-Referenzen in verschiedenen Formaten, wie z.B. HTML und LaTeX, zu erstellen. Für gewöhnlich wird *Doxygen* per Kommandozeile bedient, allerdings steht für Windows ein sehr praktisches GUI zur Verfügung.

Doxygen-konforme Kommentare

Eine simple Art der *Doxygen*-konformen Kommentierung erreicht man schlicht durch Erweiterung der eigenen Kommentare um führende Anführungszeichen.

Grundlegend sind zwei Arten von Kommentaren vorgesehen: Kurz- und Detailbeschreibungen. Kurzbeschreibungen werden dabei wie folgt formatiert:

```
/*! sets up specific user interface components for this medical data type
    public BloodPressure(String name){...}
```

So eine Kurzbeschreibung eignet sich besonders um sie bei Deklarationen stehen zu haben, z.B. in einer Header-Datei. Sie muss vor dem Quelltext stehen zu dem Bezug genommen wird.

Detailbeschreibungen, die aus mehreren Zeilen bestehen können, werden wie folgt erstellt:

```
/*!
    All needed UI components (and containers to hold these) for this
    medical data type are being instantiated using the Java Swing Founda-
    tion Classes. Component properties are being set and layout managers
    created. Action Listeners are realized as needed.
*/
    public BloodPressure(String name){...}
```

Diese Art der Kommentare eignen sich besonders um vor Definitionen aufgeführt zu werden.

Man kann auch beide Kommentartypen aufeinander folgend benutzen, wie z.B. bei Java-Code angebracht.

Zusätzliche Funktionalität bietet *Doxygen* durch *Tags*, die mit einem Backslash angeführt und mit einem Leerzeichen vom nachfolgenden Text getrennt, innerhalb von Kommentaren eingefügt werden können. Zwei solcher Tags sind `\param` (um Bemerkungen zu Parametern besonders hervorzuheben) und `\sa` (steht für „see also“). Dazu Beispiele:

A)

```
/*!
    All needed UI components (and containers to hold these) for this
    medical data type are being instantiated using the Java Swing Founda-
    tion Classes. Component properties are being set and layout managers
    created. Action Listeners are realized as needed.
    \param name is used to indicate affiliation of the data displayed via
    the UI through labels
*/
    public BloodPressure(String name){...}
```

B)

```
/*!
    Creates the user interface and network communication by calling the
    respective methods
    \sa createAndShowUI() \sa createNetworkCommunication()
*/
public static void main(String[] args) {...}
```