

Nassi-Shneiderman Diagramm Generator

Der Nassi-Shneiderman Diagramm Generator ist ein Kommandozeilenwerkzeug, das aus XML-Dateien PNG-Bilder von Nassi-Shneiderman Diagrammen generiert.

1. Systemvoraussetzungen

Um den Generator zu benutzen, muss Java 7 richtig installiert und in der Kommandozeile verfügbar sein. Ein guter XML-Editor wird empfohlen.

2. Erstellung der XML-Dateien

2.1 Das XML-Schema

Die Datei nsdg.xml enthält ein Schema für die XML-Dateien für den Generator. Bei Bedarf kann es wie folgt in das XML-Dokument eingebunden werden:

```
<diagram xmlns="http://greenlightning.eu/nsdg"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://greenlightning.eu/nsdg pfad/zu/nsdg.xml">
```

2.2 Übersicht über die Tags

<u>Tag</u>	<u>Attribute</u>	<u>Inhalt</u>
• Wurzel-Tag		
diagram	-	(Element-Tag)*
• Element-Tags		
block, procedure, break	-	[String]
branch	[condition=String]	[left], [right]
switch	[expression=String]	(case)+, [default]
loop	[type=test-first, test-last, infinite], [condition=String]	(Element-Tag)*
• Hilfs-Tags		
left, right	[label=String]	(Element-Tag)*
case, default	[label=String]	(Element-Tag)*

Attribute und Inhalte in eckigen Klammern sind optional.

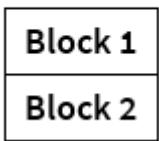
Hinweis: Manche Sonderzeichen (wie '<', '>', '&'...) lassen sich nicht direkt in XML verwenden. Die folgenden Alternativen wurden hinzugefügt, um das Erstellen der XML-Dokumente zu erleichtern, und können an Stelle von XML-Entities ('&') in String-Attributen bzw. String-Inhalten verwendet werden. Zusätzlich zu den Alternativen für XML-Sonderzeichen wurden auch einige Alternativen für Kontrollzeichen wie z. B. den Zeilenumbruch hinzugefügt.

<u>Alternative</u>	<u>Ersatz</u>
\t	<Tabulator>
\b	<Rückschritt>
\n	<Zeilenumbruch>
\r	<Wagenrücklauf>
\f	<Formularvorschub>
\a	&
\g	>
\k (alias \l)	<
\p	'
\q	"
\\"	\

2.3 Tags

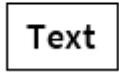
Der Wurzel-Tag des XML-Dokuments muss ein `diagram`-Tag sein. Das XML-Schema kann wie oben gezeigt in den `diagram`-Tag eingebunden werden. Außerdem kann über das `title`-Attribut eine Überschrift für das Diagramm festgelegt werden.

Wie viele andere Tags auch, kann der `diagram`-Tag eine Liste von Element-Tags enthalten. Die Elemente werden untereinander und in der gleichen Reihenfolge wie in der XML-Datei gezeichnet. Wenn der `diagram`-Tag keine Tags enthält, wird das Prozentzeichen angezeigt.

<pre><diagram title="Diagram"> <block>Block 1</block> <block>Block 2</block> </diagram></pre>	Diagram 
<pre><diagram /></pre>	

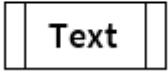
2.3.1 Block

Der Block wird benutzt, um Anweisungen darzustellen. Der `block`-Tag enthält den Text, der innerhalb des Blocks angezeigt werden soll.

<pre><block>Text</block></pre>	Block 
--------------------------------------------	-----------------------------------------------------------------------------------------------------

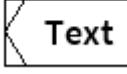
2.3.2 Procedure

Der `procedure`-Tag funktioniert genau wie der `block`-Tag, zeigt aber den Aufruf eines Unterprogramms, wie z. B. einer Prozedur, einer Funktion oder einer Methode an.

<pre><procedure>Text</procedure></pre>	Procedure 
----------------------------------------------------	---------------------------------------------------------------------------------------------------------

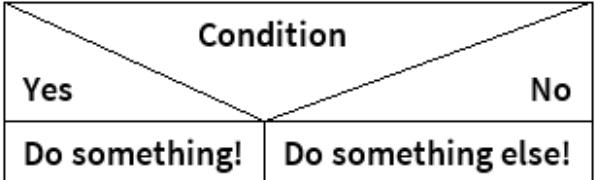
2.3.3 Break

Der `break`-Tag symbolisiert die Beendigung eines Programmteils.

<pre><break>Text</break></pre>	Break 
--------------------------------------------	-------------------------------------------------------------------------------------------------------

2.3.4 Branch

Der `branch`-Tag stellt eine Verzweigung dar. Die Bedingung wird über das `condition`-Attribut angegeben. Der `branch`-Tag kann ein `left`-Tag und / oder ein `right`-Tag enthalten. Diese Tags enthalten weitere Elemente, die in der jeweiligen Hälfte des Verzweigungselementes angezeigt werden. Sollten beide Tags vorhanden sein, muss der `left`-Tag über dem `right`-Tag stehen. Beide Tags können ein `label`-Attribut besitzen, über das die Beschriftung im oberen Teil der Verzweigung gesetzt werden kann. Falls kein `label`-Attribut vorhanden ist, wird keine Beschriftung angezeigt. Falls der `left`- oder der `right`-Tag keine Elemente enthält, wird das Prozentzeichen angezeigt. Sollte einer oder beide der Tags nicht vorhanden sein, wird keine Beschriftung und das Prozentzeichen angezeigt.

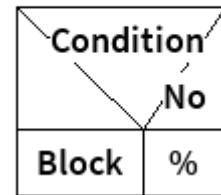
<pre><branch condition="Condition"> <left label="Yes"> <block>Do something!</block> </left> <right label="No"> <block> Do something else! </block> </right> </branch></pre>	Branch 
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------

```

<branch condition="Condition">
    <left>
        <block>Block</block>
    </left>
    <right label="No" />
</branch>

```

Branch



2.3.5 Switch

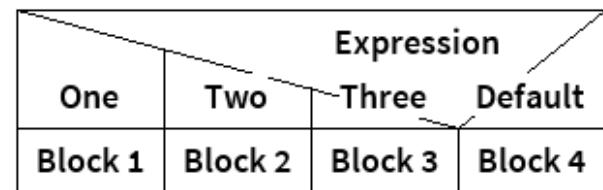
Der `switch`-Tag stellt eine Fallauswahl dar und ist dem `branch`-Tag sehr ähnlich. Es kann ein Ausdruck über das `expression`-Attribut gesetzt werden. Der `switch`-Tag muss mindestens einen, kann aber auch mehrere `case`-Tags enthalten. Als letzter Tag kann ein `default`-Tag eingefügt werden. Die `case`- und `default`-Tags entsprechen den `left`- und `right`-Tags der Verzweigung. Ist der `default`-Tag nicht vorhanden, wird er jedoch gar nicht angezeigt.

```

<switch expression="Expression">
    <case label="One">
        <block>Block 1</block>
    </case>
    <case label="Two">
        <block>Block 2</block>
    </case>
    <case label="Three">
        <block>Block 3</block>
    </case>
    <default label="Default">
        <block>Block 4</block>
    </default>
</switch>

```

Switch

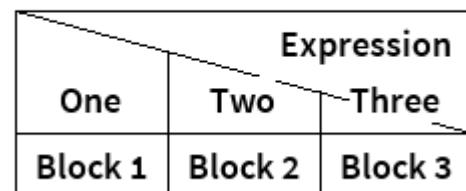


```

<switch expression="Expression">
    <case label="One">
        <block>Block 1</block>
    </case>
    <case label="Two">
        <block>Block 2</block>
    </case>
    <case label="Three">
        <block>Block 3</block>
    </case>
</switch>

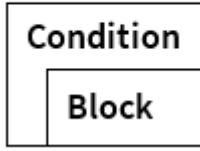
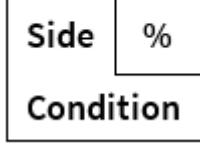
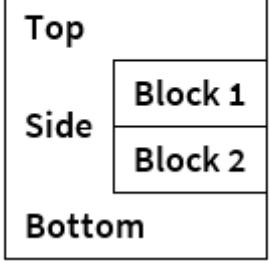
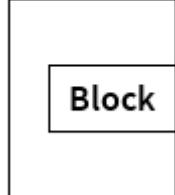
```

Switch



2.3.6 Loop

Der `loop`-Tag erzeugt eine Schleife. Der `loop`-Tag hat ein `type`-Attribut, das entweder `test-first`, `test-last` oder `infinite` sein muss. `test-first` ist der Standardwert. Dieses Attribut bestimmt, wie die Schleife dargestellt wird. Für die Werte `test-first` bzw. `test-last` kann die Schleife mit Hilfe des `condition`- und des `side`-Attributs beschriftet werden. Für den Wert `infinite` stehen die Attribute `top`, `side` und `bottom` zur Verfügung.

<pre><loop condition="Condition"> <block>Block</block> </loop></pre>	<p>Loop</p>  <pre> graph TD Loop[Loop] --- Condition[Condition] Condition --- Block[Block] </pre>
<pre><loop type="test-last" side="Side" condition="Condition" /></pre>	<p>Loop</p>  <pre> graph TD Loop[Loop] --- Side[Side] Side --- Percent[%] Percent --- Condition[Condition] </pre>
<pre><loop type="infinite" top="Top" side="Side" bottom="Bottom"> <block>Block 1</block> <block>Block 2</block> </loop></pre>	<p>Loop</p>  <pre> graph TD Loop[Loop] --- Top[Top] Top --- Side[Side] Side --- Bottom[Bottom] Bottom --- Block1[Block 1] Block1 --- Block2[Block 2] </pre>
<pre><loop type="infinite"> <block>Block</block> </loop></pre>	<p>Loop</p>  <pre> graph TD Loop[Loop] --- Block[Block] </pre>

3. Benutzung des Generators

Aufrufen des Generators:

```
java -jar nsdg.jar <pfad-zu-einer-xml-datei>...
```

Mehrere XML-Dateien können gleichzeitig bearbeitet werden, wenn die Pfade durch Leerzeichen getrennt angegeben werden.

Die PNG-Dateien werden im gleichen Ordner und mit dem gleichen Namen wie die XML-Dateien gespeichert, nur mit der Dateiendung 'png'.

Der Generator überschreibt Dateien ohne Warnung!