

Demo for Package listings

Herbert Voß

September 14, 2003

At first we want to see the first Java code for a little program, called Welcome (see Listing 1).

Listing 1: My first Java Program

```
1  /* Das erste Beispielprogramm, absolut nutzlos! :-)
2  Der externe Dateiname MUSS Welcome1.java lauten,
3  sonst ist das System nicht in der Lage, die
4  Klassen auseinander zu halten!      */
5
6  public class Welcome1 {                               // ein
7  erstes Javaprogramm
8  public static void main(String[] args ) { // Die
9  HAUPTmethode
10 System.out.println( "Hallo Fans!\nWillkommen zur
    Java Programmierung!" );
11 }
12 }
```

Another way is to write an own command for the inputlisting, which has a better layout. This one is called `lst` with 2 parameters:

1. The filename of the Sourcefile
2. The caption

```
\newcommand{\lst}[2]{%
  \noindent\rule[-1ex]{\textwidth}{0.3mm}\vspace{-1ex}
  \lstinputlisting[caption={#2},%
  label={#1},%
  showstringspaces=false,%
```

```

frame=tb ,%
extendedchars=true ,%
basicstyle=\footnotesize\tt ,%
numbers=left ,%
stepnumber=1 ,%
numberstyle=\tiny ,%
keywordstyle=\color{red} ,%
language=Java ,
breaklines=true ]{#1}%
\vspace{1ex}%
}

```

Now the listing is called by

```
\lst{Welcome1.java}{My first Java Program}
```

The lst-command creates by default a label which is the filename, here {Welcome1.java}.

Listing 2: My first Java Program

```

1  /* Das erste Beispielprogramm, absolut nutzlos! :-)
2  Der externe Dateiname MUSS Welcome1.java lauten,
3  sonst ist das System nicht in der Lage, die
4  Klassen auseinander zu halten!      */
5
6  public class Welcome1 {                               // ein erstes
7      Javaprogramm
8      public static void main(String[] args ) { // Die
9          HAUPTmethode
10         System.out.println( "Hallo Fans!\nWillkommen zur Java
                Programmierung!" );
    }
}

```

A reference to this listings is therefore no problem, look at listing 2. This reference is set by `\ref{Welcome1.java}`.

Changing the fontstyle to courier is possible with:

```
basicstyle=\fontfamily{pcr}\fontseries{m}\selectfont\footnotesize
```

or with loading the package `courier.sty`.

Listing 3: My first Java Program

```

/* Das erste Beispielprogramm, absolut nutzlos! :-)
Der externe Dateiname MUSS Welcome1.java lauten,

```

```

sonst ist das System nicht in der Lage, die
Klassen auseinander zu halten!      */

public class Welcome1 {                // ein erstes
    Javaprogramm
    public static void main(String[] args ) { // Die
        HAUPTmethode
        System.out.println( "Hallo_Fans!\nWillkommen_zur_Java_
        Programmierung!" );
    }
}

```

The baselineskip (vertical space) can also be changed. As an additional command to the options of `basicstyle` write:

```
\renewcommand\baselinestretch{0.8}
```

and the listings becomes:

Listing 4: My first Java Program

```

/* Das erste Basispielprogramm, absolut nutzlos! :-)
Der externe Dateiname MUSS Welcome1.java lauten,
sonst ist das System nicht in der Lage, die
Klassen auseinander zu halten!      */

public class Welcome1 {                // ein erstes
    Javaprogramm
    public static void main(String[] args ) { // Die
        HAUPTmethode
        System.out.println( "Hallo_Fans!\nWillkommen_zur_Java_
        Programmierung!" );
    }
}

```

A new language which is not part of the current list, can easily be defined with the `\lstdefinlanguage`:

```

\lstdefinlanguage{bash}{%
    morekeywords={%
        for, if, else, then, fi, echo, case, in,%
        elif, esac, for, do, done, tar, while, until,
        shift,%
        mv, mkdir ;;, \$\#, \$1, \$2},%
    sensitive=t,%
    morecomment=[l]{\#} ,%
    morestring=[d]{"}%
}

```

With this definition an output of a bashfile is shown with listings 5.

Listing 5: A listing for a new defined Language

```
#!/bin/bash
# voss/010622
#
NOMENCL_STYLE=~voss/.lyx/styles/nomencl.ist # or path to nomencl-
package
INDEX_STYLE=~voss/.lyx/styles/Letter.ist # my default
#TEXPATH=/usr/share/texmf/teTeX/bin/i386-linux-glibc
TEXPATH=/usr/share/texmf/teTeX/bin/i386-linux-libc6
FILENAME=$3 # lyx calls it with makeindex -p -s
filename
NAME=${FILENAME%%.*} # without suffix
echo "This is myMakeIndex ..."
echo "File: $NAME"
#
$TEXPATH/makeindex -s $INDEX_STYLE -c -q $3
# if there is no glossary it doesn't matter
if [ -f $name.gls ] ; then
    $TEXPATH/makeindex $NAME.gls -s $NOMENCL_STYLE -o $NAME.gls
fi
```

A list of listings is possible with `\lstlistoflistings?`:

Listings

1	My first Java Program	1
2	My first Java Program	2
3	My first Java Program	2
4	My first Java Program	3
5	A listing for a new defined Language	3