Generator for structogram
(Nassi - Shneiderman)

Peter Lenser
Peter.Lenser@gmail.com

printet on: January 20, 2013

The program translates methods, written in Java, into a *.tex file. The translation routine uses the package struktex, written by Jobst Hoffmann, to depict the code as a structogram (Nassi-Shneiderman chart). It works well with ©Miktex. The generated structogram is useable for being displayed by an overhead or LCD projector or could be included in other LaTeXfiles.

The program is written and tested with ©Eclipse 3.n and ©Eclipse Juno under ©Windows 7. The methods should be formatted by eclipse with the default preferences. Other settings or code without format will probably produce errors. Eclipse encodes characters under Windows with CP1252 under Linux with UTF8. So, if the program is used under Linux, the source code needs some modifications because mutated vowels will be automatically removed. I advise to avoid mutated vowels - even in the comment.

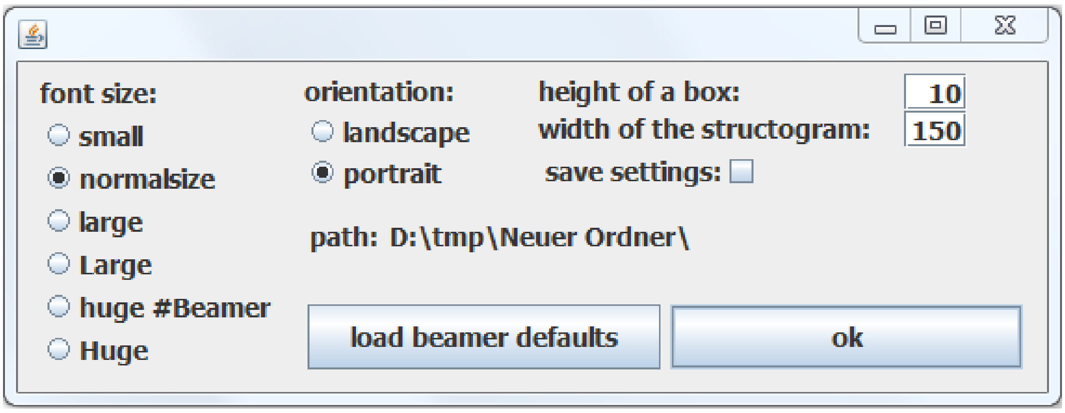
How to use:

Copy a method from your sourcecode in eclipse into a text editor and save it as a text file. Example:

```
private String until_semicolon(StringBuilder sB) {
    int i = 0;
    int length = sB.length();
    String word;
    while ((i < length) && (sB.charAt(i) != ';')) {
        i++;
    }
    word = format_text(new StringBuilder(sB.substring(0, i)));
    sB = sB.delete(0, i + 1);
    return word;
}
```

Save the file as 'until_semicolon.txt'. You can choose any name, only the file extension .txt is important.

The program starts with a little user interface:



font size:
○ small
● normalsize
○ large
○ Large
○ huge #Beamer
○ Huge

orientation:
○ landscape
● portrait

height of a box: 10
width of the structogram: 150
save settings: □

path: D:\tmp\Neuer Ordner\

In this example I chose for orientation portrait, for font size normalsize and for width of the structogram 150. After pressing the ok button you will see a file selection dialog. Select the .txt file you want to translate. After the translation is completed you can enter a filename and save the file in .tex format. In this example I choose until_semicolon.tex. You can open this file direct with TeXworks from MikTex. Here is the file:

```
1  \documentclass[a4paper,11pt]{letter}
2  \usepackage[utf8]{inputenc}
3  \usepackage[T1]{fontenc}
4  \usepackage{anysize}
5  \usepackage{lmodern}
6  \usepackage{german}
7  \marginsize{1 cm}{1 cm}{0 cm}{2 cm}
8  \usepackage[pict2e,german]{struktex}
9  \pFonts{\large\sffamily}{\large\sffamily\bfseries}{\large\sffamily\slshape}
10 \begin{document}
11 \normalsize
12 \begin{struktogramm}(150,80)
13     [private\ String\ until\_semicolon(StringBuilder\ sB)]
14 \assign[10]{\(int\ \ i\ \gets 0\)}
15 \assign[10]{\(int\ \ length\ \gets sB.length()\)}
16 \assign[10]{\(String\ \ word\)}
17 \while[10]{\(while \ ((i\ <\ length)\ \&\&\ (sB.charAt(i)\ !=\ ';'))\)}
18 \assign[10]{\(i++\ \)}
19 \whileend
20 \assign[10]{\(word\ \ \gets
21     format\_text(new\ StringBuilder(sB.substring(0,\ i)))\)}
22 \assign[10]{\(sB\ \ \gets sB.delete(0,\ i\ +\ 1)\)}
23 \assign[10]{\(\gets \\ word\)}
24 \end{struktogramm}\\
25 \end{document}
```

The first line is generated in according to the selected orientation. The lines 2 to 9 are copied from the file firstlines.txt. This File can be customized to your preferences.

Result:

private String until_semicolon(StringBuilder sB)

```
int i ← 0
int length ← sB.length()
String word
while ((i < length) && (sB.charAt(i) ! = ';'))
    i++
word ← format_text(new StringBuilder(sB.substring(0, i)))
sB ← sB.delete(0, i + 1)
←                                   word
```

Explanatory notes and known problems

The program converts the Java switch-statement into the case-statement from Nassi - Shneiderman. This is unfortunately not correct. If so, the program puts the break command unnecessarily into the structogram. The correct translation without “break“ at the end of every case would be a series of if–else branches. But for my application in teaching this solution is still adequate.

The program is also not able to:

- wrap long lines and correct the height of a box
- make the declaration of variable better or remove the declaration
- arrange long programms on multiple pages
- describe and set parameters
- calculate the correct high of the complete structogram, if the structogram contains nested structograms
- and much more . . .

Because it is very easy to correct these settings in the Tex-document itself, I do not plan to add or correct any features of the converter at the moment.

Copyright