Abschnitt 5.7 auf Seite 159:
C-Quellcode-Zähler

ccount.l:

```
%{
#include <stdio.h>
int codeLines = 0, commentLines = 0, blankLines = 0;
int currCodeLine = 0, currCommentLine = 0;
%}
%option noyywrap
%x COMMENT
%%
"/*"            { BEGIN COMMENT; currCommentLine = 1; }
<COMMENT>"*/"   { BEGIN INITIAL; currCommentLine = 1; }
<*>[ \t]        { /* Blank, Tab ignorieren */ }
.               { currCodeLine = 1; }
<COMMENT>.      { currCommentLine = 1; }
<*>"\n"         { if(currCodeLine)
                      codeLines++;
                  else if(currCommentLine)
                          commentLines++;
                      else
                          blankLines++;
                  currCodeLine = 0;
                  currCommentLine = 0;
                }
        /* Die letzte Zeile muss mit einem Newline abgeschlossen sein. */
%%
int main() {
    yylex();
    printf(
      "%d Code-Zeilen, %d Nur-Kommentar-Zeilen, %d ganz leere Zeilen\n",
      codeLines, commentLines, blankLines);
}
```

▶▶ make ccount

▶▶ ./ccount < cexample.c

Zusatzaufgabe:
`ccount2.l`:

```
%{
#include <stdio.h>
int codeLines = 0, commentLines = 0, blankLines = 0;
int currCodeLine = 0, currCommentLine = 0;
%}
%option noyywrap
%x COMMENT
%x STRING
%%
"/*"             { BEGIN COMMENT; currCommentLine = 1; }
<COMMENT>"*/"    { BEGIN INITIAL; currCommentLine = 1; }
"\""             { BEGIN STRING; currCodeLine = 1; }
<STRING>"\""     { BEGIN INITIAL; currCodeLine = 1; }
<STRING>"\\\\"   { currCodeLine = 1; }
<STRING>"\\\""   { currCodeLine = 1; }
<STRING>"\n"     { codeLines++; /* String-Fortsetzungszeile */
                   currCodeLine = 1;
                   currCommentLine = 0;
                 }
<INITIAL,COMMENT>[ \t]  { /* Blank, Tab ignorieren */ }
<INITIAL,STRING>.       { currCodeLine = 1; }
<COMMENT>.       { currCommentLine = 1; }
<INITIAL,COMMENT>"\n"   {
                   if(currCodeLine)
                       codeLines++;
                   else if(currCommentLine)
                           commentLines++;
                       else
                           blankLines++;
                   currCodeLine = 0;
                   currCommentLine = 0;
                 }
        /* Die letzte Zeile muss mit einem Newline abgeschlossen sein. */
%%
int main() {
    yylex();
    printf(
      "%d Code-Zeilen, %d Nur-Kommentar-Zeilen, %d ganz leere Zeilen\n",
      codeLines, commentLines, blankLines);
}
```

▶▶ `make ccount2`

▶▶ `./ccount2 < cexample2.c`

Anmerkung: Die Lösung im Lex&Yacc-Buch ist komplizierter, aber kümmert sich nicht um Strings.