Richtlinien zur Formatierung von C-Quellcode

============================================

Ziel: Vereinfachung der Lesbarkeit beim Austausch von Quellcode

\* pro Zeile nur ein Befehl

\* nicht mehr als 70 Zeichen pro Zeile, um Zeilenumbruch zu vermeiden

\* mathematische Grundoperatoren und Gleichheitszeichen durch Leerzeichen von

Variablen und Konstanten abtrennen

z.B.:

a = b + c; /\* richtig \*/

a =b+ c; /\* falsch \*/

a \*= b; /\* richtig \*/

a\*=b; /\* falsch \*/

\* nach Komma kommt ein Leerzeichen

z.B.:

int i, j, x, y; /\* richtig \*/

int i,j,x,y; /\* falsch \*/

\* Vergleichsoperatoren moeglichst mit Leerzeichen umgeben

z.B.:

if (val1 >= val2) ...

\* auf ein C-Schluesselwort folgt ein Leerzeichen

z.B.:

for ( ...) /\* richtig \*/

for(...) /\* falsch \*/

if (...) /\* richtig \*/

if(...) /\* falsch \*/

\* Funktionsaufrufe

z.B.:

funktion\_1( parameter1, parameter2); /\* richtig \*/

funktion\_1 ( parameter1, parameter2); /\* falsch \*/

funktion\_1(parameter1, parameter2); /\* falsch \*/

\* Gestaltung von Schleifenkoerpern und Einrueckungen (1 Tab = 2 Leerzeichen)

z.B.:

for ( i = 0; i < ende\_i; i++)

{

for ( j = 0; j < ende\_j; j++)

{

....

}

}

if ( (x < x\_end) && (y < y\_end) )

{

...

}

do

{

...

} while ( !stop);

\* Kommentare

je mehr Erlaeuterungen im Quelltext, desto einfacher ist die spaetere

Nachbearbeitung oder Veraenderung des Programms

a) Kopf einer Quelltext-Datei

in einer Datei werden Funktionen zusammengefasst, die zu einer

Verarbeitungseinheit gehoeren

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* Datei...: dateiname.c

\* Funktion: kurze Beschreibung

\* Autor...: Vorname Nachname

\* Datum...: 29.02.2009

\* 28.04.2010 H.Musterfrau, 32.05.2011, F.Mustermann

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

b) Kopf einer Funktion

/\*------------------------------------------------------------------

\* funktion\_1()

\*

\* kurze Beschreibung

\*-----------------------------------------------------------------\*/

void

funktion\_1( int parameter1, float parameter2)

{

int i, x; /\* Deklarationen mit 2 Leerzeichen einruecken \*/

float val;

/\* mind. eine Leerzeile \*/

i = 0;

val = (float)i;

.

.

.

}

c) Kommentar fuer einen Programmabschnitt

a = b;

/\*

\* Verarbeitung der Teilsignale

\* unter Beruecksichtigung ...

\*

\*/

c = funk\_ab( ...);

d = funk\_ac( ...);

d) Kommentar fuer eine Programmzeile, Kommentarumgebung ( /\* \*/) moeglichst

mit Tab-Stops positionieren

vec\_len\_h = vec\_len >> 1; /\* halbe Vektorlaenge \*/

process\_signal( signal, vec\_len\_h); /\* Teilsignal verarbeiten \*/