

Programmierstil:

Korrekt, aber schlecht:

```
void j(FILE *i, FILE *o, int w) {int i,j,k,sw,ps; sw=(w-lw)/ng;
ps=(w-lw)%ng; ... }

void j(FILE *i, FILE *o, int w)
{int i,j,k,sw,ps;
sw=(w-lw)/ng;
ps=(w-lw)%ng; ... }
```

Die Deluxe-Version:

```
/****** Justify - Text im Puffer im Blocksatz ausgeben *****/
void Justify (
FILE* in, /* Input File */
FILE* out, /* Output File */
int width /* gewünschte Breite fuer den Blocksatz */
) {
int i,j,k; /* Laufvariablen */
int spacewidth; /* Anzahl Spaces zwischen zwei Woertern */
int plusspaces; /* Anzahl Zwischenraeume mit einem Space mehr
*/
spacewidth = (width - linewidth) / nofgaps;
plusspaces = (width - linewidth) % nofgaps;
...
}
```

Ein bisschen schlichter:

```
void Justify (FILE *inputfile, FILE *outputfile, int width) {
int i,j,k;
int spacewidth, plusspaces;

spacewidth= (width - linewidth) / nofgaps;
plusspaces = (width - linewidth) % nofgaps;
...
}
```

Tips für einen guten Programmierstil:

- Der Text sollte **konsistent** formatiert sein, am besten mit einer der **Standard-Formatierungen**.
- Der Text sollte visuell einen **guten Eindruck** machen und einen **Überblick** über das Programm geben.
- Die **visuelle Struktur** sollte die **Struktur des Programms** wiedergeben.
- Der Text sollte so geschrieben sein, dass **andere** ihn lesen können.
- **Kommentare** an geeigneten Stellen können viel zum Verständnis beitragen. Zu viele Kommentare stören.
- **Gute Namen und Bezeichnungen** für Variablen, Prozeduren, Funktionen, Konstanten etc. können sehr viele Kommentare überflüssig machen.

⇒ **Überlege Dir etwas beim Formatieren
Deiner Quellen!**