

# Style Guide

Olaf Ippisch

Um leicht lesbare, wartbare und fehlerarme Programme zu schreiben, sollten folgende Regeln beachtet werden:

- Der Programmtext sollte eingerückt werden, insbesondere Blöcke in Schleifen, Bedingungen, Funktionen ...
- Zur Einrückung sollten immer Leerzeichen verwendet werden, keine Tabulatoren (die meisten Editoren lassen sich so konfigurieren), da die Tabweite variieren kann und dann die Formatierung zerstört wird.
- Das Programm sollte durch Kommentare erklärt werden.
- Kommentare sollten notwendig und angemessen sein. Vermeide:  

```
for(int i = 0; i < 10; ++i) // Diese Anweisung startet die Schleife
```
- Variablennamen sollten sprechend sein, d.h. einen Namen haben der ihre Funktion deutlich macht (abgesehen von Schleifenzählern).
- Variablen sollten definiert werden, wenn sie das erste mal gebraucht werden.
- Es sollten immer angemessene Datentypen verwendet werden (z.B. nicht `int` für `bool` oder `float` statt `int` ...)
- Der Gültigkeitsbereich von Variablen sollte so klein wie möglich sein. Schleifenzähler in `for`-Schleifen sollten z.B. direkt im Schleifenkopf definiert werden.
- Variablen müssen vor der ersten Verwendung initialisiert werden.
- Keine Variablen definieren die nicht verwendet werden.
- Variablen die nur in einer Schleife gebraucht werden, sollten auch innerhalb der Schleife definiert werden. Dadurch wird es wesentlich einfacher zu verstehen was eine Schleife tun soll und ob sie es tatsächlich tut.

Ein schlechtes Beispiel ist:

```
int main()
{
    bool flag = false;
    [...]
    for(int i = 0; i < 10; ++i)
    {
        // do something which might
        // change flag
        if(flag)
        {
            // ...
        }
        flag = false;
    }
}
```

besser wäre:

```
int main()
{
    [...]
    for(int i = 0; i < 10; ++i)
    {
        bool flag = false;
        // do something which might
        // change flag
        if(flag)
        {
            // ...
        }
    }
}
```

- Es sollten nur Headerdateien eingebunden werden, die auch notwendig sind.
- Vermeiden von `if` oder `else` Zweigen die nie ausgeführt werden.
- Bedingungen bei denen eine Alternative ein Sonderfall der anderen Alternative ist, sollten vermieden werden:

Ein schlechtes Beispiel ist:

```
if(y == 2)
    x = 1;
else
    x = y - 1;
```

besser wäre:

```
x = y - 1;
```

- Unnötig komplizierte Anweisungen sind zu vermeiden, z.B. verwende `var += 2` statt `++(++var)`