

Coding Style

Belegarbeit Software Engineering II

Gruppe 8 Bibliotheksprozesse unterstützen

Gliederung

1. <u>Einleitung</u>	3
2. <u>Formatierung</u>	4
3. <u>Bezeichner</u>	5

1. Einleitung

Die Festlegung und Durchsetzung eines klar definierten Coding Styles für ein Softwaresystem ist ein wichtiges Element für die Sicherung der Qualität des Systems. Durch einen einheitlichen Style im Quellcode kann schnelle Übersichtlichkeit, einfache Verständlichkeit und gute Änderbarkeit garantiert werden. So bietet ein gut strukturierter und formatierter Code die Grundlage für spätere Arbeit mit dem Programm.

2. Formatierung

Die maximale **Zeilenlänge** sollte nicht über 130 Zeichen liegen.

Die öffnenden und schließenden **geschwungenen Klammern** bei Funktionen und Anweisungen sollten immer auf eigenen Zeilen stehen.

Bsp: falsch `int foo(){
 ...}`

 richtig `int foo()
 {
 ...
 }`

Außerdem wird der Code innerhalb eines Blocks immer um 4 Zeichen (einen Tab in den meisten IDEs) eingerückt.

Bei Anweisungen die nur von einem Ausdruck gefolgt werden kann auf die Klammern verzichtet werden allerdings wird der Ausdruck auf eine eigene Zeile geschrieben und eingerückt.

Bsp: falsch `if (foo > bar) exit();`

 richtig `if (foo > bar)
 exit();`

Beim **initialisieren mehrerer Variablen** vom gleichen Typ wird entweder jede Variable auf eine eigene Zeile mit dem Typ geschrieben oder aber der Typ und alle Variablen in einer Aufzählung in die gleiche Zeile geschrieben.

Bsp: falsch `int foo,
 bar;`

 richtig `int foo;
int bar;`

 oder `int foo, bar;`

3. Bezeichner

Die Bezeichner für Funktionen und Variablen werden möglichst informativ gewählt das heißt es dürfen und sollen auch mehrere Wörter verwendet werden um die Funktion bzw. den Inhalt des Bezeichners zu verdeutlichen. Dabei werden die Wörter aneinander gereiht und jedes Wort wird groß geschrieben.

Bsp: falsch bool connecttodatabase
richtig bool connectToDataBase

Bezeichner für Variablen die von den Windows-Forms verwendet werden sollten sowohl die Funktion die sie innerhalb der GUI erfüllen (z.B. Button, Label, Textbox) als auch die Funktionalität innerhalb des Programms beschreiben.

Bsp: falsch button1, label2
richtig buttonAusleihen, labelTitel