

C++ Coding Guidelines

Tobias W.

May 2, 2023

Contents

1	Klammern	2
1.1	Geschweifte Klammern	2
2	Naming Conventions	3
2.1	Bei jeder Benennung	3
2.2	Variablen und Felder	3
2.3	Klassen	3
2.4	Methoden und Funktionen	3
3	Dokumentation	4

1 Klammern

1.1 Geschweifte Klammern

```
1 int main()
2 { // Alleingestellte Klammer
3     int input;
4     std::cin >> input;
5
6     // Guard Clause
7     if(4 == input) {return 1}
8
9     // Code here
10
11     return 0;
12 } // Alleingestellte Klammer
13
14 Car::Car()
15 : color(255), wheelCount(4)
16 {} // Leere Funktion
```

- Geschweifte Klammern gehören immer in leere Zeilen, mit Ausnahme von Guard Clauses, welche nur ein Return enthalten sowie leeren Funktionen.
- Verzweigungen und Schleifen werden immer mit Klammern genutzt. Lediglich die einzelnen Fälle in einer Switch Verzweigung sollten nicht mit Klammern umgeben sein. Außer dies ist unbedingt nötig (Scope für Variablen). Dann sollte dies aber über Kommentare gekennzeichnet werden.
- In Ausdrücken, in denen die Operatoren Reihenfolge von Bedeutung ist, sollen der Übersichtlichkeit halber Klammern verwendet werden, auch wenn dies von Compilerseite nicht nötig ist.

2 Naming Conventions

2.1 Bei jeder Benennung

- Benennungen sollten im Englischen erfolgen
- Benennungen sollten sich immer von C++ Key-Wörtern stark unterscheiden. Der Klassenname “Auto” ist somit zum Beispiel nicht zulässig, da er dem Key-Wort “auto” zu ähnlich sieht.

2.2 Variablen und Felder

- Variablen werden in Camelcase geschrieben (z.B. bufferSize)
- Klassenfelder werden ebenfalls in Camelcase geschrieben, Ihnen wird aber ein unterstrich vorangestellt. (z.B. _bufferSize)
- Variablen sollten benennende Namen haben (z.B. carColor, bufferSize)
- Temporäre Variablen die nahe an ihrer Definition verwendet werden dürfen von dieser Regel abweichen. Zum Beispiel “int i” in einer for Schleife. Es muss aber zum Beispiel durch Kommentare sichergestellt werden, dass sich die Definition bei Erweiterung nicht zu weit von der Verwendung entfernt.

2.3 Klassen

- Klassen sollten in PascalCase benannt werden (z.B. BufferObject)
- Klassennamen sollten Substantive sein (z.B. Camera)

2.4 Methoden und Funktionen

- Methoden und Funktionen sollten in PascalCase benannt werden (z.B. GetColor, DrawCircle)
- Methoden/Funktionsnamen sollten Verben sein (z.B. TakePicture()).
- Getter und Setter Methoden müssen mit Add bzw. Set gefolgt von dem betroffenen Feld benannt werden (z.B. AddPitch)

3 Dokumentation

- Die Dokumentation findet in den Header-Dateien statt.
- Dokumentation im Doxygen-Styl
- Jede Funktion/Methode muss dokumentiert sein.
Private Methoden müssen mit @internal dokumentiert werden. Die Dokumentation darf weniger ausführlich sein, da sie nur für Wartungs- und Erweiterungszwecke verwendet wird.
- Public/Protected Felder müssen mit ///
Dokumentiert werden, Private Felder nicht.
- In den CPP-Dateien findet nur eine rudimentäre Implementierungsdokumentation statt. Diese soll “wenn” nötig die Implementierung erklären. Sie findet durch einzelne kurze Kommentare im Quellcode statt (Kein Doxygen-Styl)