C++ Namenskonvention Übersicht

In C++ ist eine konsistente Namensgebung von entscheidender Bedeutung, um Code lesbar, wartbar und kollisionsfrei zu gestalten. Die Einhaltung von Namenskonventionen erleichtert die Zusammenarbeit in Teams und fördert eine klare Strukturierung des Codes.

Klassennamen:

Klassennamen werden üblicherweise in CamelCase geschrieben, beginnend mit einem Großbuchstaben, ohne Unterstriche. Zum Beispiel: MeineKlasse, AutoMotor, BenutzerInterface.

Klassenmethoden:

Methoden innerhalb einer Klasse folgen ebenfalls dem CamelCase und beginnen mit einem Kleinbuchstaben. Es ist ratsam, klare, beschreibende Namen zu wählen, die die Funktionalität der Methode widerspiegeln. Zum Beispiel: berechneWert(), setzeName(string name), holeDaten().

Namespace:

Namespaces dienen dazu, den Bereich von Identifiern zu organisieren und Kollisionen zwischen verschiedenen Teilen des Codes zu vermeiden. Namen für Namespaces sollten sinnvoll und beschreibend sein, oft in kleingeschriebenen Buchstaben und ggf. separiert durch Unterstriche. Zum Beispiel: namespace geometrie { /* ... */ }, namespace rechnungen { /* ... */ }.

Attribute (Klassenvariablen):

Attribute innerhalb einer Klasse folgen in der Regel der gleichen Konvention wie Klassenmethoden: CamelCase mit einem kleinen Anfangsbuchstaben. Klare und aussagekräftige Namen sind hier besonders wichtig, um die Verwendung und Bedeutung der Attribute zu verdeutlichen. Zum Beispiel: anzahlVersuche, aktuellerWert, letztesErgebnis.

Konstante Attribute:

Konstante Werte werden oft in Großbuchstaben mit Unterstrichen zwischen den Wörtern benannt, um ihre Konstanz deutlich zu machen. Zum Beispiel: MAX_ANZAHL_ELEMENTE, STANDARD_WARTUNGSZEIT, PI_WERT.

Statische Attribute und Methoden:

Statische Attribute und Methoden sind auf Klassenebene definiert und gehören nicht zu einer spezifischen Instanz der Klasse. Ihre Namen folgen üblicherweise den gleichen Konventionen wie nicht-statische Attribute und Methoden. Jedoch könnten einige Entwickler eine zusätzliche Kennzeichnung verwenden, wie z. B. ein Präfix wie s_ für statische Attribute (s_anzahlObjekte) oder

eine spezielle Benennung für statische Methoden, um ihre Natur hervorzuheben (berechneSumme() vs. statischeBerechneSumme()).

Einheitlichkeit in der Namensgebung ist entscheidend, um die Lesbarkeit und Wartbarkeit des Codes sicherzustellen. Es ist auch wichtig, die spezifischen Konventionen eines Teams oder einer Codebasis zu respektieren und zu befolgen, um eine konsistente Codebasis zu gewährleisten.

Pointer-Variablen:

Pointer-Variablen werden oft durch das Hinzufügen von ptr oder einem ähnlichen Hinweis auf ihre Natur als Zeiger gekennzeichnet. Die Verwendung von klar erkennbaren Namen ist wichtig, um zu verdeutlichen, dass es sich um eine Zeiger-Variante handelt. Zum Beispiel: benutzerPtr, datenZeiger, listePtr.

Es ist entscheidend, bei der Benennung von Pointer-Variablen darauf zu achten, dass sie sowohl den Typ als auch die Referenz klar widerspiegeln. So können Fehler vermieden und die Lesbarkeit des Codes verbessert werden.

Die Beachtung dieser Namenskonventionen für Pointer-Variablen hilft dabei, die Handhabung von Zeigern transparent zu machen und erleichtert das Verständnis des Codes für andere Entwickler, die darauf aufbauen oder ihn warten müssen.