**Cheatcheet Naming Guidelines (Framework)**

|  |  |
| --- | --- |
| **PascalCasing** | Erstes Zeichen von jedem Wort grossgeschrieben.  Bei Zwei-Buchstaben-Akronyme werden diese grossgeschrieben.  PascalCasing wird für alle öffentliche Member-, Typ- und Namespace-Namen, die aus mehreren Wörter besteht verwendet. |
| **camelCasing** | Wird nur für Parameternamen verwendet, alle ersten Buchstaben von jedem Wort ab dem zweiten werden grossgeschrieben. Auch bei Zwei-Buchstaben-Akronymen |
| **Wortwahl** | In lesbarem Englisch schreiben    anstatt    Keine Unterstriche, Bindestriche nicht alpahnumerische Zeichen |
| **Verwendung von Abkürzungen / Akronymen** | Nicht verwendet (Lieber GetWindow anstatt GetWin)  Sprachspezifische Namen verwenden (Umwandlung zu int64 🡪 ToInt64 nicht ToLong) |
| **Assembly** | Ein Assembly ist die Bereitstellungs- und Identitätseinheit für Programme mit verwaltetem Code. Namen sollte auf das hinweisen, was sich dort befindet (System.Data.)  Diese Assemblys werden normalerweise mit einer DLL eins zu eins zugeordnet. Diese kann man nachfolgendes Muster benennen.  In <Component> können Klausen mit einem oder mehreren Punkten eingefügt werden. (Litware.Controls.dll) |

|  |  |
| --- | --- |
| **Namensräumen (Namespaces)** | Ziel der Benennung ist eine ausreichende Klarheit für den Programmierer, der das Framework verwendetet, was der Inhalt ist.        Beispiel:  Präfixe mit einem Firmennamen  PascalCasing verwenden  Nicht den gleichen Namen für einen Namespace und einen Typ in diesem Namespace nehmen.  Keine Namen wie Debug verwenden, die von anderen Namespaces genutzt werden könnten. Keine Abkürzungen  Klassennamen kein Präfix («C») |
| **Namen von Methoden** | Verben oder Verbalphrasen |
| **Namen von Eigenschaften** | Substantiv, Nominalphrase oder Adjektiv Kein **Get** in dem Namen |
| **Namen von Ereignissen** | Verb oder Verbalphrase      Parameter verwenden |
| **Namen von Felder** | PascalCasing verwenden und ein Nomen, Nominalphrase oder Adjektiv |
| **Benennungsparameter** | Parameternamen 🡪 camelCasing  Beschreibende Namen verwenden und  Namen nach Bedeutung des Parameters bestimmen. |
| **Namen von Klassen, Structs und Interfaces** | Mit Substantiven oder Monialphrasen mit PascalCasing benennen. (Klassen und Strukturen)  Typnamen von Methoden mit Verbalphrasen  Schnittstellen mit Adjektivphrasen oder gelegentlich mit Substantiven oder Nominalphrasen. |

\_ + Variablenname bei einer private Feels oder einer predicted