



Atrix Consult GmbH

Hardturmstrasse 169

CH-8005 Zürich

Tel: 044 / 445 70 40

Fax: 044 / 440 51 71

admin@atrix.ch

www.atrix.ch

Richtlinien

Softwareentwicklung

Atrix Consult GmbH



Atrix Consult GmbH
Hardturmstrasse 169
CH-8005 Zürich
Tel: 044 / 445 70 40
Fax: 044 / 440 51 71
admin@atrix.ch
www.atrix.ch

Inhaltsverzeichnis

| | |
|--|----------|
| Allgemeine Regeln | 3 |
| Sprache der Richtlinien | 4 |
| VB.NET-Quelltext | 4 |
| Gross- und Kleinschreibung | 4 |
| Klassen, Strukturen, Schnittstellen | 6 |
| Verschiedene Ableitungen | 6 |
| Enumerationen | 7 |
| Member von Datentypen | 7 |
| Eigenschaften und Felder | 7 |
| Methoden | 8 |
| Parameter von Methoden | 8 |
| Änderungskontrolle | 9 |



Atrix Consult GmbH
Hardturmstrasse 169
CH-8005 Zürich
Tel: 044 / 445 70 40
Fax: 044 / 440 51 71
admin@atrix.ch
www.atrix.ch

Allgemeine Regeln

Lesbarkeit und Klarheit gelten als oberste Gebote:

- Wählen Sie leicht lesbare Bezeichner Namen aus. Beispielsweise ist eine Eigenschaft mit dem Namen *HorizontalAlignment* im Englischen besser lesbar als *AlignmentHorizontal*
- Ziehen Sie Lesbarkeit einer kurzen Bezeichnung vor. Der Eigenschaftsname *CanScrollHorizontally* ist besser als *ScrollableX* (ein unklarer Verweis auf die X-Achse).
- Verwenden Sie keine Bindestriche oder andere Sonderzeichen (darunter fallen auch Umlaute, %, etc.).
- Verwenden Sie keine Bezeichner, die Konflikte mit Schlüsselwörtern weit verbreiteter Programmiersprachen verursachen.
- keine Abkürzungen oder Akronyme verwenden.
- Verwenden Sie keine Abkürzungen oder Zusammenziehungen als Bestandteile von Bezeichnernamen.
Verwenden Sie beispielsweise *OnButtonClick* und nicht *OnBtnClick*.
- Vermeiden Sie Namen, die auf eine bestimmte Programmiersprache hindeuten.
- Verwenden Sie als Typnamen allgemein interessante Namen statt sprachspezifischer Schlüsselwörter.
Beispielsweise ist *GetLength* besser als *GetInt*.
- Verwenden Sie in den seltenen Fällen, in denen ein Bezeichner über keine inhaltliche Bedeutung außer seinem Typ verfügt, einen .NET-Typnamen und keinen sprachspezifischen Namen.
Beispielsweise sollte eine Methode, die Daten in *Int16* konvertiert, mit *ToInt16* und nicht mit *ToShort* benannt werden.
- Wo sinnvoll, sollen im Programmcode Kommentare zur Verarbeitung hinterlegt sein. Bei komplexen Funktionen / Prozeduren soll einleitend der Zweck der Verarbeitung beschrieben werden.



Atrix Consult GmbH
Hardturmstrasse 169
CH-8005 Zürich
Tel: 044 / 445 70 40
Fax: 044 / 440 51 71
admin@atrix.ch
www.atrix.ch

Sprache der Richtlinien

Die Implementationssprache ist in der Regel in Englisch. Die Norm zu führen kann schneller in Foren verstanden werden und auch schneller gelöst werden.

VB.NET-Quelltext

```
Public Class CustomerDataTable Inherits DataTable ...
```

```
End Class
```

```
Private Sub TitleInput_Validate(ByVal sender As Object, ByVal e As EventArgs) ...
```

```
End Sub
```

Gross- und Kleinschreibung

Es gibt drei Varianten für die Groß- und Kleinschreibung von Bezeichnern:

- Großbuchstaben werden für gängige Abkürzungen mehrteiliger Begriffe (Akronym genannt) verwendet.
Beispiel: **IO**
- Bei Pascal-Schreibweise wird der erste Buchstabe groß, der Rest klein geschrieben.
Beispiel: **Car**
- Bei Camel-Schreibweise wird alles klein geschrieben.
Beispiel: **mycar**

Abkürzungen für einteilige Begriffe sind nicht vorgesehen. Ausnahmen sind ID für identifier und OK; diese sind aber nach den „normalen“ Regeln in Pascal als „Id“ und „Ok“ sowie in Camel als id und ok zu schreiben.

Es gibt folgende Grundsätze:

- Alle öffentlichen Bezeichner erhalten die Pascal-Schreibweise.
- Alle privaten Bezeichner erhalten die Camel-Schreibweise.
- Bei allen Zusammensetzungen werden der zweite und folgende Bestandteil in Pascal-Schreibweise angefügt; es werden keine Striche (Unterstrich '_' oder Bindestrich '-') dazwischen gesetzt.



Atrix Consult GmbH
 Hardturmstrasse 169
 CH-8005 Zürich
 Tel: 044 / 445 70 40
 Fax: 044 / 440 51 71
admin@atrix.ch
www.atrix.ch

| Verwendungsbereich | Schreibweise | Einfache Begriffe | Zusammensetzungen |
|--|--------------|-------------------|--|
| Klasse | Pascal | Program | AppDomain |
| Enumerationstyp | Pascal | Rule | ErrorLevel |
| Enumerationswerte | Pascal | Monday | FatalError |
| Ereignis | Pascal | Click | ValueChanged |
| Ausnahmeklasse | Pascal | Exception | WebException |
| Schnittstelle | Pascal | IDisposable | ICustomFormatter |
| Methode | Pascal | Main | ToString |
| Namespace | Pascal | System.Drawing | System.ComponentModel |
| Parameter | Camel | current | typeName |
| Eigenschaft | Pascal | Name | BackColor |
| Feld | Camel | name | backColor |
| lokale Variable | Camel | name | nameValue |
| Zusammensetzungen mit Akronymen | | | |
| öffentlich | Pascal | IO Xml | IOStream XmlSerializer OdbcCommand |
| privat | Camel | io xml | ioStream xmlDocument odbcParameter |



Atrix Consult GmbH
Hardturmstrasse 169
CH-8005 Zürich
Tel: 044 / 445 70 40
Fax: 044 / 440 51 71
admin@atrix.ch
www.atrix.ch

Klassen, Strukturen, Schnittstellen

Für die Namen von Typen gelten die folgenden Regeln:

- Verwenden Sie die Pascal-Schreibweise, also große Anfangsbuchstaben.
- Verwenden Sie kein Präfix (also kein 'C' als Kennzeichen für eine Klasse).
- Abweichend davon ist eine Schnittstelle mit 'I' zu bezeichnen, z.B. IComparable.

Für bestimmte Klassen mit besonderer Bedeutung soll dies bereits im Namen erkennbar sein.

Verschiedene Ableitungen

Abgeleitete Klassen sollen als letzten Bestandteil (Suffix) den Namen der Basisklasse tragen. Dies gilt vor allem für die folgenden Typen:

| Art eines Typen | Basisklasse | Namensteil | Beispiel |
|---------------------------------|------------------|--------------|------------------------------|
| Ausnahmen | System.Exception | Exception | OutOfMemoryException |
| Datenstrom | System.IO.Stream | Stream | MemoryStream |
| Attributklassen | System.Attribute | Attribute | AttributeUsageAttribute |
| Delegate für Ereignisse | System.Delegate | EventHandler | AssemblyLoadEventHandler |
| andere Delegates | | Callback | TimerCallback ^[6] |
| Argumente für Ereignisse | System.EventArgs | EventArgs | MouseEventArgs |



Atrix Consult GmbH
Hardturmstrasse 169
CH-8005 Zürich
Tel: 044 / 445 70 40
Fax: 044 / 440 51 71
admin@atrix.ch
www.atrix.ch

Enumerationen

Auch für Aufzählungstypen gelten die allgemeinen Regeln: kein Präfix, kein Suffix (auch nicht Enum oder Flags), zu beachten:

- Der Name der Enumeration darf nicht im Namen der Elemente wiederholt werden.
- Nur Namen für Enumerationen, deren Werte Bitfelder sind, stehen im Plural. *Diese Aufzählungen werden als Flags-Enumerationen bezeichnet.*

Ein Beispiel für richtige Bezeichnungen lautet:

Name der Enumeration: DayOfWeek im System-Namespace

Elemente: Monday, Tuesday etc.

DayOfWeek.Monday

Member von Datentypen

Wie schon eingangs erwähnt, gelten diese Regeln eigentlich nur für öffentliche und geschützte Elemente. Es bietet sich aber an, sie auch für private Elemente anzuwenden und dafür die Camel- statt der Pascal-Schreibweise zu verwenden.

Eigenschaften und Felder

Eigenschaften bezeichnen Daten, die in den Objekten enthalten sind. Dafür sind Substantive, substantivische Ausdrücke und Adjektive sinnvoll.

Für **private Felder** gilt: Sofern eine Eigenschaft (vor allem über Getter und Setter) direkt mit einem privaten Feld verbunden ist, kann so verfahren werden:

- Sofern die Programmiersprache zwischen Groß- und Kleinschreibung unterscheidet, wird für die Eigenschaft die Pascal-Schreibweise eines Begriffs und für das Feld der gleiche Begriff in Camel-Schreibweise verwendet.
- **Boolesche Eigenschaften** sollen einen positiven Begriff bezeichnen. Notfalls kann auch ein Präfix Is, Can, Has o.ä. vorangestellt werden.
- Doppelte Verwendung eines Begriffs als Eigenschaft und in einer Get-Methode ist nicht zulässig. Beispiel: GetNumber als Methode und Number als Eigenschaft.



Atrix Consult GmbH
Hardturmstrasse 169
CH-8005 Zürich
Tel: 044 / 445 70 40
Fax: 044 / 440 51 71
admin@atrix.ch
www.atrix.ch

Methoden

Verben oder verbale Ausdrücke als Bezeichner sind zu verwenden.

Als Bezeichnung ist eine Beschreibung dafür zu wählen, *was* die Methode macht, aber nicht dafür, *wie* dies erledigt wird. Die .NET-Entwickler erklären das so: „*Verwenden Sie also keine Implementierungsdetails als Methodennamen.*“ Zur größeren Klarheit sind durchaus mehrteilige Begriffe geeignet.

Parameter von Methoden

Die Namen der Parameter geben dem Programmierer, der eine Methode aufrufen und eine passende Methode auswählen soll, wichtige Hinweise zur Verwendung der Methode. Ein guter Parametername hilft deshalb ungemein bei der Benutzerfreundlichkeit einer Klassenbibliothek und soll angeben, auf welche Daten er sich auswirkt oder zu welcher Aufgabe er gehört.

Parameter werden in Camel-Schreibweise angegeben, weil sie lokal (innerhalb der Methode) verwendet werden.



Atrix Consult GmbH
Hardturmstrasse 169
CH-8005 Zürich
Tel: 044 / 445 70 40
Fax: 044 / 440 51 71
admin@atrix.ch
www.atrix.ch

Änderungskontrolle

| Version | Datum | Bearbeiter | Bemerkungen |
|------------|-----------|------------|---|
| 1.0 | 16.3.2013 | JP. Rosano | Erstellung der Dokumentation |
| 1.1 | 26.2.2014 | JP. Rosano | Allgemeines Überarbeitung Dokument |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |