|  |
| --- |
| **Система распределённого оптического распознавания символов Nebulous** |
| Спецификация |
|  |
| Соглашения разработки оформления кода. |
|  |
| **Игорь Калашников, Павел Пересторонин** |
| **26.10.2011** |
|  |

[1 Введение 3](#_Toc307406020)

[2 Соглашения по разработке и написанию кода 3](#_Toc307406021)

[2.1 Создание нового проекта в Visual Studio 3](#_Toc307406022)

[2.2 Соглашения по именованию 3](#_Toc307406023)

[2.3 Сокращения 3](#_Toc307406024)

[2.4 Скобки 3](#_Toc307406025)

[2.5 Пространства имен 3](#_Toc307406026)

[2.6 Оформление 4](#_Toc307406027)

[2.7 Другое 4](#_Toc307406028)

# Введение

Этот документ описывает единый стиль кода, которого следует придерживаться при разработке данного проекта. В первую очередь соглашения относятся к языку C#, но применимы и для других языков с учетом их специфики.

# Соглашения по разработке и написанию кода

## Создание нового проекта в Visual Studio

После создания нового проекта с использованием мастера:

1. Изменить Default Namespace на: Nebulous.<название проекта>.
2. Для всех конфигураций поставить Treat warnings as errors: All.
3. Подписать сборку ключом Nebulous.snk.
4. Для всех конфигураций включить Code Analysis on Build.
5. Изменить AssemblyInfo.cs по образу и подобию существующих проектов.

## Подписывание сборок

Visual Studio по умолчанию при выборе ключа для подписи копирует его в папку проекта. Для того, чтобы использовать рашаренный на все проекты ключ:

1. Add – Existing Item…
2. Указать .snk файл и вместо пункта Add выбрать Add As Link
3. В свойствах проекта выбрать этот ключ для подписи. (Несмотря на то, что будет указан полный путь, в действительности в файле проекта сохранится относительный).

## Соглашения по именованию

Стандартные для C#.

1. Имена даются с учетом контекста. Так, для коллекции необходимо использовать Count, а не CountOfElementsOfCollection, а для класса WebPage метод должен называться Update(), а не UpdateWebPage().

## Сокращения

1. Не используйте аббревиатуры или неполные слова в идентификаторах, если только они не являются общепринятыми. Например, пишите GetWindow, а не GetWin.
2. Не используйте акронимы, если они не общеприняты в области информационных технологий.
3. Широко распространенные акронимы используйте для замены длинных фраз. Например, *UI* вместо User Interface или *Olap* вместо On-line Analytical Processing.
4. Если имеется идентификатор длиной менее трех букв, являющийся сокращением, то его записывают заглавными буквами, например System.**IO,** System.Web.**UI.** Имена длиннее двух букв записывайте в стиле Паскаль или Кэмел, например Guid, Xml, xmlDocument.

## Скобки

1. Открывающаяся фигурная скобка ставится на отдельной строке, за исключением случая объявления атрибутов:

public Example { get; set; }

1. После закрывающейся фигурной скобки ставится пустая строка.
2. После if ставится пробел.
3. Перед и после скобок в вызове и объявлении функции не ставятся.

## Пространства имен

1. Пространство имен верхнего уровня – название проекта.
2. Не более одного в одном файле.
3. Именование соответствует каталогам проекта. Вложенность обозначается точкой.
4. Множественное число следует использовать в случае, если пространство имен объединяет некоторое количество разных, но семантически похожих сущностей. И наоборот, когда пространство имен содержит некую подсистему, стоит использовать единственное число.
5. Не используйте одно и то же имя для класса и пространства имен.

## Оформление

1. Используйте **табуляцию**, а не пробелы для отступов. Размер табуляции равен четырем пробелам.
2. При форматировании текста (кроме отступа в начале строки) используйте пробелы.
3. Избегайте строк длиннее 100 символов, переносите инструкцию на другую строку при необходимости.
4. При переносе части кода инструкций и описаний на другую строку вторая и последующая строки должны быть отбиты вправо на один отступ (табуляцию).
5. Оставляйте запятую на предыдущей строке так же, как вы это делаете в обычных языках (русском, например).
6. Избегайте лишних скобок, обрамляющих выражения целиком. Лишние скобки усложняют восприятие кода и увеличивают возможность ошибки. Если вы не уверены в приоритете операторов, лучше загляните в соответствующий раздел документации.
7. Не размещайте несколько инструкций на одной строке. Каждая инструкция должна начинаться с новой строки.

## Другое

1. Директивы using помещаются вне пространства имен.
2. Неиспользуемые using-и удаляются.