# Введение

В данном документе изложен рекомендуемый стиль оформления кода.

Берите пример со стиля именования в базовой библиотеке классов.

Помните что не так важно какой именно стиль кодирования вы предпочитаете как

## Цитаты

«Код пишется для людей, а не для компьютера»

“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.” Martin Fowler

“Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.”

## Рекомендуемая литература

* «Совершенный код. Мастер-класс», Стив Макконнелл, Издательство «Русская редакция».  
  Оригинальное название: Code Complete, Steve McConnell, Microsoft Press

## Обозначения

* UpperCamelCase – первая буква каждого слова пишется заглавной, остальные прописными. Например: WriteLine, BackroundColor.
* lowerCamelCase – первое слово пишется прописными, остальные по правилам UpperCamelCase. Например:
* \_lowerCamelCase – lowerCamelCase с символом подчеркивания в начале
* UPPER\_CASE – все буквы заглавные; отдельные слова разделены символом подчеркивания.

# Форматирование

# Идентификаторы

## Локальные переменные и аргументы функций

Название идентифкатора должно отражать его назначение

|  |  |
| --- | --- |
| * Используйте lowerCamelCase * Используйте существительные в единственном числе для одинарных значений * Используйте существительные в множествнном числе для коллекций и массивов. | * Не используйте «венгерскую» нотацию * Не используйте транслитерацию с кириллицы * Избегайте сложных аббревиатур |

## Константы

Используйте UPPER\_CASE.

## Члены enum

Используйте UpperCamelCase.

## Имена типов

Используйте UpperCamelCase

Избегайте использования префиксов типа T, C (кроме I для интерфейсов). Например, названия классов TObject, CObject и т.п. явлются плохим стилем.

Избегайте использования суффиксов типа Class, Struct и т.п. За исключением исключечений и атрибутов

### Имена классов для исключений

Имя класса должно заканчиваться на Exception.

### Имена классов для атрибутов

Имя класса должно заканчиваться на Attribute.

### Имена интерфейсов

Имя интерфейса должно начинаться с I и дальше по правилам UpperCamelCase.

### Имена тип-аргументов для обобщенных типов

Если тип-аргумент всего один, то используем T

Если тип-аргумент несколько, то ???

## Члены типов

### Private поля

Используйте \_lowerCamelCase. Подчеркивание позволяет быстро отличать private поля от локальных переменных и избавляет от необходимости использовать this для разрешения конфликтов.

### Protected и Public поля

Используйте UpperCamelCase.

### Методы

Используйте UpperCamelCase.

#### Обработчики событий

Используйте схему именования On<ИсточникСобытия><ИмяСобытия> или On<ИсточникСобытия>\_<ИмяСобытия>.

## Неявно типизированные локальные переменные (var)

Используем для локальных переменных которым присваивается значение.

Используем для инициализации обобщенных типов т.к. это обычно сильно сокращает код.

# Комментарии

* Комментарий должен описывать суть кода, а не повторять его
* Лучший комментарий – это пустой комментарий. Это означает, что код написан так что не требует дополнительных пояснений
* Избегайте шутливых комментариев
* Не пишите комментариев в начале файла с описанием сделанных изменений. Для этого есть системы контроля версий.
* Не пишите нецензурные комментарии

# Директивы using

* Размещаются в начале файла в виде групп. Внутри каждой группы пространства имен идут в алфавитном порядке
  + System.\*
  + Пространства имен текущего проекта (project) и решения (solution)
  + Пространства имен внешних библиотек
  + Псевдонимы (using Alias = Some.Namespace;)

# Настройки Visual Studio

Tools -> Options

Text Editor -> C# -> Tabs

Text Editor -> C# -> Formatting

# Общие советы

## Функции

Чем короче, тем лучше

Single responsibility

Params[] аргументы

Bool-ловушка

# Приложения

## Общепринятые пары глаголов

Add…/Remove…

Open…/Close…

Insert…/Delete…

Create…/Destroy…

Increment/…Decrement…

Acquire…/Release…

Lock…/Unlock…

Up…/Down…

Begin…/End…

Show…/Hide…

Fetch…/Store…

Start…/Stop…

To…/From… (подразумевается преобразование типов)