

Стандарт оформления кода C#

- **Camel case** – первая буква первого слова в идентификаторе в нижнем регистре, все первые буквы последующих слов – в верхнем.

Пример: theCategory;

Правила именования идентификаторов

Общие правила именования идентификаторов

- При именовании идентификаторов не используются аббревиатуры или сокращения, если только они не являются общепринятыми.

Пример: GetWindow(), а не GetWin();

Использование верхнего и нижнего регистра в именах

- Запрещается создавать два различных **namespace**'а, функции, типа или свойства с одинаковыми именами, отличающиеся только регистром. Запрещается создавать функции с именами параметров, отличающимися только регистром. Ниже приведены примеры **Неправильных** названий.

Пример:

*KeywordManager u Keywordmanager;
KeywordManager.Keyword u KeywordManager.KEYWORD;
int id {get, set} u int ID {get, set};*

*findByID(int id) u FindByID(int id);
void MyFunction(string s, string S).*

Правила именования классов

- Следует избегать имен классов, совпадающих с именами классов **.NET Framework**;
- В качестве имен классов используются имена существительные;
- Если класс представляет собой сущность, хранимую в базе данных – имя класса соответствует имени таблицы. В этом случае имя класса – это название сущности в единственном числе, имя таблицы – во множественном числе.

Пример: таблица Users, класс User;

Правила именования функций

- Имена функций должны давать четкое представление о том, какое действие эта функция выполняет. Имя функции начинается с глагола, указывающего на то, какое действие она выполняет;

Правила именования параметров функций

- Для именования параметров используется стиль **camel case**;
- Имена параметров должны давать четкое представление о том для чего используется параметр, и какое значение следует передать при вызове функции.

Пример:

*public void EncodeString(string **sourceString**, ref string **encodedString**),
a **ne** public void EncodeString(string **string1**, ref string **string2**).*

Правила именования свойств

- Название свойства типа **bool** должно представлять из себя вопрос, требующий ответа да или нет.

Примеры названий: “CanDownload”, “HasKeywords”, “IsChecked”, “NeedsUpdate”.

Правила именования переменных

- Для именования переменных используется стиль **camel case**;
- Переменные объявляются согласно следующему шаблону:

<Тип> <Название поля>;

Пример: int userID = null;

Форматирование кода

- В одном файле не объявляется больше одного namespace'а и одного класса (исключение – небольшие вспомогательные private классы);
- Фигурные скобки размещаются всегда на отдельной строке;
- В условии **if-else** всегда используются фигурные скобки;
- Использование строк длиннее 80-100 символов не желательно. При необходимости инструкция переносится на другую строку. При переносе части кода на другую строку вторая и последующая строки сдвигаются вправо на один символ табуляции;
- Каждая переменная объявляется на отдельной строке;