C#: требования и рекомендации по написанию кода

[.NET](http://habrahabr.ru/hub/net/)\*

1. Требования

1.1 Pascal casing

Описываются имена:  
• всех определений типов, в том числе пользовательских классов, перечислений, событий, делегатов и структур;  
• значения перечислений;  
• readonly полей и констант;  
• интерфейсов;  
• методов;  
• пространств имен (namespace);  
• свойств;  
• публичных полей;

namespace SampleNamespace  
{  
  enum SampleEnum  
  {  
    FirstValue,  
    SecondValue  
  }  
  
  struct SampleStruct  
  {  
    public int FirstField;  
    public int SecondField;  
  }  
  
  interface ISampleInterface  
  {  
    void SampleMethod();  
  }  
  
  public class SampleClass: SampleInterface  
  {  
    const int SampleConstValue = 0xffffff;  
  
    readonly int SampleReadonlyField;  
  
    public int SampleProperty  
    {  
      get;  
      set;  
    }  
  
    public int SamplePublicField;  
  
    SampleClass()  
    {  
      SampleReadonlyField = 1;  
    }  
  
    delegate void SampleDelegate();  
    event SampleDelegate SampleEvent;  
  
    public void SampleMethod()  
    {  
    }  
  }  
}\* This source code was highlighted with [Source Code Highlighter](http://poison.qsh.ru/).

1.2 Camel casing

Описываются имена:  
• локальных переменных;  
• аргументов методов;  
• защищенных (protected) полей.

protected int sampleProtectedField;  
  
int SampleMethod(int sampleArgument)  
{  
  int sampleLocalVariable;  
}\* This source code was highlighted with [Source Code Highlighter](http://poison.qsh.ru/).

1.3 Суффиксы и префиксы

Применяются следующие суффиксы и префиксы:  
• имена пользовательских классов исключений всегда заканчиваются суффиксом “Exception”;  
• имена интерфейсов всегда начинаются с префикса «I»;  
• имена пользовательских атрибутов всегда заканчиваются суффиксом «Attribute»;  
• имена делегатов обработчиков событий всегда оканчиваются суффиксом EventHandler, имена классов-наследников от EventArgs всегда заканчиваются суффиксом EventArgs.

public class SampleException: System.Exception  
{  
  public SampleException()   
  {  
  }  
}  
  
interface ISample  
{  
  void SampleMethod();  
}  
  
[System.AttributeUsage(System.AttributeTargets.All, Inherited = false, AllowMultiple = true)]  
sealed class SampleAttribute: System.Attribute  
{  
  public SampleAttribute()  
  {  
  }  
}  public delegate void AnswerCreatedEventHandler(object sender, AnswerCreatedEventArgs e);  
  public class AnswerCreatedEventArgs: EventArgs  
  {  
    public int СreatedId;  
    public int ParentId;  
    public string CreatorName;  
  }\* This source code was highlighted with [Source Code Highlighter](http://poison.qsh.ru/).  
  
\* This source code was highlighted with [Source Code Highlighter](http://poison.qsh.ru/).

1.4 Аббревиатуры

При использовании аббревиатур в именах, капитализации подлежат аббревиатуры с двумя символами, в остальных аббревиатурах необходимо приводить к верхнему регистру только первый символ.

namespace Sample.IO  
{  
}  
  
class HttpUtil  
{  
}\* This source code was highlighted with [Source Code Highlighter](http://poison.qsh.ru/).

2. Рекомендации

2.1 Именование методов

Используйте конструкцию глагол-объект для именования методов

ShowUserInfo()

В частном случае, для методов, которые возвращают значение, используйте в паре глагол-объект для глагола «Get», а для объекта – описание возвращаемого значения.

GetUserId()

2.2 Переменные, поля и свойства

• При именовании переменных избегайте использования сокращенных вариантов вроде **I** и **t**, используйте **index** и **temp**. Не используйте венгерскую нотацию или используйте ее только для закрытых членов. Не сокращайте слова, используйте **number**, а не**num**.  
• Рекомендуется для имен элементов управления указывать префиксы, описывающие тип элемента. Например: txtSample, lblSample, cmdSample или btnSample. Эта же рекомендация распространяется на локальные переменные сложных типов: ThisIsLongTypeName tltnSample = new ThisIsLongTypeName();  
• не используйте публичных или защищенных полей, вместо этого используйте свойства;  
• используйте автоматические свойства;  
• всегда указывайте модификатор доступа private, даже если разрешено его опускать;  
• всегда инициализируйте переменные, даже когда существует автоматическая инициализация.

2.3 Дополнительные рекомендации

• используйте пустую строку между логическими секциями в исходном файле, классе, методе;  
• используйте промежуточную переменную для передачи bool-значения результата функции в условное выражение **if**;

bool boolVariable = GetBoolValue();  
if (boolVariable)  
{  
}\* This source code was highlighted with [Source Code Highlighter](http://poison.qsh.ru/).

2.4 Объем кода

• избегайте файлов с более чем 500 строками кода;  
• избегайте методов с более чем 200 строками кода;  
• избегайте методов с более чем 5 аргументами, используйте структуры для передачи большого числа параметров;  
• одна строка кода не должна иметь длину более 120 символов.

3. Спорные моменты для обсуждения

3.1 Закрытые поля

Первый вариант  
Имена закрытых полей всегда начинаются с префикса «m\_» остальная часть имени описывается с помощью Pascal Casing.

private int m\_SamplePrivateField;

Второй вариант  
Имена закрытых полей всегда начинаются с префикса «m» остальная часть имени описывается с помощью Pascal Casing.

private int mSamplePrivateField;

Третий вариант (пока самый вероятный на окончательный вариант)  
Имена закрытых полей всегда начинаются с префикса «\_» остальная часть имени описывается с помощью Camel Casing.

private int \_samplePrivateField;

3.2 Дополнительные требования

• всегда располагайте открывающие и закрывающие фигурные скобки на новой строке;  
• всегда используйте фигурные скобки для выражения if, даже когда в выражение входит только одна строка кода;