Этот гайд поможет вам писать код на C# красиво.

Гайд носит рекомендательный характер: если вы категорически не согласны с каким-то пунктом, обсудите это с преподавателем после пары. Не разжигайте холиваров на пустом месте.

### 

1. Скобки после return не нужны

|  |  |
| --- | --- |
| return (a + b + c); | return a + b + c; |

1. Используйте var

|  |  |
| --- | --- |
| string s = "abc"; | var s = "abc"; |

1. Все названия на английском языке (используйте переводчик при затруднениях)

|  |  |
| --- | --- |
| var procent = 0.05;  string[] stroki; | var percent = 0.05;  string[] lines; |

1. Названия переменных должны быть осмысленными. Чем больше область видимости, тем более развернутым должно быть название

|  |  |
| --- | --- |
| var p = 0.05; | var percent = 0.05; |

1. Не используйте of, the и a в названиях

|  |  |
| --- | --- |
| ColorOfCar  theSun  aStar | CarColor  sun  star |

1. Булевы переменные называйте в виде вопросов, на которые можно ответить да или нет. Не используйте название flag

|  |  |
| --- | --- |
| var flag = false; | var isSorted = false; |

1. Названия переменных в lowerCamelCase

|  |  |
| --- | --- |
| var percent\_per\_month = 0.05;  var PercentPerMonth = 0.05; | var percentPerMonth = 0.05; |

1. Не оставляйте пустых строк после объявления метода и после return

|  |  |
| --- | --- |
| int GetOne()  {  return 1;  } | int GetOne()  {  return 1;  } |

1. Стирайте неактуальные комментарии

|  |  |
| --- | --- |
| int GetOne()  {  //TODO Напишите здесь return 1  return 1;  } | int GetOne()  {  return 1;  } |

1. Тело if всегда в новой строке. Тело цикла всегда в новой строке.

|  |  |
| --- | --- |
| if (a > b) return 0; | if (a > b)  return 0; |

1. Не пишите else, если выше везде выходите из тела if

|  |  |
| --- | --- |
| if (a > b)  {  return 10;  }  else  {  …  return 20;  } | if (a > b)  {  return 10;  }  …  return 20; |

1. Если в теле if больше одной строки, используйте скобки

|  |  |
| --- | --- |
| if (a > b)  if (a < c)  return 0; | if (a > b)  {  if (a < c)  return 0;  } |

1. Пишете одну и ту же строчку два раза - выделите в метод

|  |  |
| --- | --- |
| var lenAB = Math.Sqrt((ax-bx)\*(ax-bx) + (ay-by)\*(ay-by));  var lenBC = Math.Sqrt((bx-cx)\*(bx-cx) + (by-cy)\*(by-cy)); | double GetLength(  double ax,  double ay,  double bx,  double by)  {  return Math.Sqrt((ax-bx)\*(ax-bx) + (ay-by)\*(ay-by));  }  …  var lenAB = GetLength(ax, ay, bx, by);  var lenBC = GetLength(bx, by, cx, cy); |

1. Не сравнивайте bool c константой с помощью ==

|  |  |
| --- | --- |
| if (success == true)  …  if (success == false) | if (success)  …  if (!success) |

1. Названия методов в UpperCamelCase

|  |  |
| --- | --- |
| public static double get\_length()  public static double getLength() | public static double GetLength() |

1. Названия методов должны начинаться с глагола

|  |  |
| --- | --- |
| public static double Length() | public static double GetLength() |

1. Используйте множественное число в названии коллекций

|  |  |
| --- | --- |
| Apple[] Apple | Apple[] Apples |

1. Не пишите тип коллекции в названии. Исключение, если есть коллекция другого типа с теми же данными

|  |  |
| --- | --- |
| Orange[] ArrayOfOranges  List<Apple> AppleList | Orange[] Oranges  List<Apple> Apples |

1. Всегда явно указывайте модификаторы доступа

|  |  |
| --- | --- |
| class Vector | public class Vector |

1. Соблюдайте порядок элементов внутри класса. Сначала поля и свойства, потом конструктор, затем публичные методы, в конце приватные методы.
2. Между методами оставляйте пустую строку

|  |  |
| --- | --- |
| public static void DoSomething()  {  ...  }  public static void DoSomethingElse()  {  ...  } | public static void DoSomething()  {  ...  }  public static void DoSomethingElse()  {  ...  } |

1. После for, if, … ставьте пробел

|  |  |
| --- | --- |
| if(a > b)  return 0; | if (a > b)  return 0; |

1. Обрамляйте арифметические, логические операторы, оператор присваивания пробелами

|  |  |
| --- | --- |
| if (a>b)  c=0; | if (a > b)  c = 0; |

1. Правильно переносите длинный тернарный оператор

|  |  |
| --- | --- |
| return needAutocomplete ? Сommand.AutoComplete : Command.Ignore; | return needAutocomplete  ? Сommand.AutoComplete  : Command.Ignore; |

1. Переносите выражения так, чтобы новая строка начиналась с оператора

|  |  |
| --- | --- |
| if (longExpression &&  anotherExpression) | if (longExpression  && anotherExpression) |

1. Для записи однострочных свойств используйте =>

|  |  |
| --- | --- |
| public int Length  {  get {return bytes.Length;}  } | public int Length => bytes.Length; |

1. При записи длинной цепочки вызовов методов пишите каждый вызов в новой строке

|  |  |
| --- | --- |
| return lines.Where(...)  .Select(...)... | return lines  .Where(...)  .Select(...)  ... |

ИГРА: <https://cleancodegame.github.io/>