

Этот гайд поможет вам писать код на C# красиво. Перед отправкой решения на ulearn пробегитесь по списку. Каждый пункт - это потенциальный комментарий к вашему решению.

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;
```

2. Используйте var

```
string s = "abc";
```

```
var s = "abc";
```

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

```
var procent = 0.05;  
string[] stroki;
```

```
var percent = 0.05;  
string[] lines;
```

4. Названия переменных должны быть осмысленными. Чем больше область видимости, тем развернутее должно быть название

```
var p = 0.05;
```

```
var percent = 0.05;
```

5. Не используйте of, the и a в названиях

```
ColorOfCar  
theSun  
aStar
```

```
CarColor  
sun  
star
```

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

```
var flag = false;
```

```
var isSorted = false;
```

## 7. Названия переменных в lowerCamelCase

```
var percent_per_month = 0.05;  
var PercentPerMonth = 0.05;
```

```
var percentPerMonth = 0.05;
```

## 8. Не оставляйте пустых строк после объявления метода и после return

```
int GetOne()  
{  
  
    return 1;  
  
}
```

```
int GetOne()  
{  
    return 1;  
}
```

## 9. Стирайте неактуальные комментарии

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

```
int GetOne()  
{  
    return 1;  
}
```

## 10. Тело if всегда в новой строке. Тело цикла всегда в новой строке.

```
if (a > b) return 0;
```

```
if (a > b)  
    return 0;
```

## 11. 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;
```

## 12. Если в теле if больше одной строки, используйте скобки

```
if (a > b)
```

```
if (a > b)
```

<pre>if (a &lt; c)     return 0;</pre>	<pre>{     if (a &lt; c)         return 0; }</pre>
--	--

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

<pre>var lenAB = Math.Sqrt((ax-bx)*(ax-bx) + (ay-by)*(ay-by)); var lenBC = Math.Sqrt((bx-cx)*(bx-cx) + (by-cy)*(by-cy));</pre>	<pre>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);</pre>
--	---

14. Не сравнивайте bool с константой с помощью ==

<pre>if (success == true) ... if (success == false)</pre>	<pre>if (success) ... if (!success)</pre>
---	---

15. Названия методов в UpperCamelCase

<pre>public static double get_length() public static double getLength()</pre>	<pre>public static double GetLength()</pre>
---	---

16. Названия методов должна начинаться с глагола

<pre>public static double Length()</pre>	<pre>public static double GetLength()</pre>
--	---

17. Используйте множественное число в названии коллекций

<pre>Apple[] Apple</pre>	<pre>Apple[] Apples</pre>
--------------------------	---------------------------

18. 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
```

19. Всегда явно указывайте модификаторы доступа

```
class Vector
```

```
public class Vector
```

20. Соблюдайте порядок элементов внутри класса. Сначала поля и свойства, потом конструктор, затем публичные методы, в конце приватные методы.

21. Между методами оставляйте пустую строку

```
public static void DoSomething()  
{  
    ...  
}  
public static void DoSomethingElse()  
{  
    ...  
}
```

```
public static void DoSomething()  
{  
    ...  
}  
  
public static void DoSomethingElse()  
{  
    ...  
}
```

22. После for, if, ... ставьте пробел

```
if(a > b)  
    return 0;
```

```
if (a > b)  
    return 0;
```

23. Обрамляйте арифметические, логические операторы, оператор присваивания пробелами

```
if (a>b)  
    c=0;
```

```
if (a > b)  
    c = 0;
```

24. Правильно переносите длинный тернарный оператор

```
return needAutocomplete ?  
Command.AutoComplete : Command.Ignore;
```

```
return needAutocomplete  
    ? Command.AutoComplete
```

```
: Command.Ignore;
```

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

```
if (longExpression &&  
    anotherExpression)
```

```
if (longExpression  
    && anotherExpression)
```

26. Для записи однострочных свойств используйте =>

```
public int Length  
{  
    get {return bytes.Length;}  
}
```

```
public int Length => bytes.Length;
```

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

```
return lines.Where(...)  
    .Select(...)...
```

```
return lines  
    .Where(...)  
    .Select(...)  
    ...
```