



Костыли и велосипеды: как не нужно писать код

Полезно многим, а то и всем

4.1



**Писать код –
это легко**



Угадай, если сможешь...



Угадай, если сможешь...

- 1 Method
- 2 ShowNumbers
- 3 CreateArray
- 4 ShowNums
- 5 Ar
- 6 Numbers



Как мы пишем код?

```
void Method(int maximum)
{
    int minimum;
    minimum = -maximum;
    while (minimum <= maximum)
    {
        Console.Write(minimum + " ");
        minimum++;
    }
}
```



Как мы пишем код?

```
void func7support(int m, int n)
{
    if (m<=n)
    {
        Console.WriteLine (m);
        func7support(m+1, n);
    }
}
```



Как мы пишем код?

```
int[] CreateArray(int N)
{
    int[] arrayA = new int[N * 2 + 1];
    for (int i = -N; i <= N; i++)
    {
        arrayA[i + N] = i;
    }
    return arrayA;
}
```



Как мы пишем код?

```
int Ar (int N)  //задаем метод
{
    int x= -N;  //первая цифра -N (задаем цикл)
    while (x <=N )  //до тех пор пока x меньше или равен N
    {
        Console.WriteLine (x);  //выводим в консоль "x"
        x++;  //инкремент
    }
    return x;
}
```



Как мы пишем код?

```
void Numbers(int n)
{
    int length = n + n;
    for (int i = 0; i < length + 1; i++)
    {
        Console.WriteLine(-n + i);
    }
}
```



Как мы пишем код?

```
string ShowNums(int N)
{
    string NumShow = "";
    for (int i = -N; i < N; i++)
    {
        NumShow = NumShow + i + " ";
    }
    return NumShow;
}
```



Как мы пишем код?



Как мы пишем код?

Задача: Показать числа от $-N$ до N



Как мы пишем код?

Как написал бы я



Как мы пишем код?

```
int af = -5;  
int uf = 5;  
Console.WriteLine($"{af} .. {uf}");
```



Как мы пишем код?

```
int af = -5;  
int uf = 5;  
Console.WriteLine($"{af} .. {uf}");
```



**Что в этом коде
можно улучшить?**



Что в этом коде можно улучшить?

- 1 Имя переменных
- 2 Имя метода
- 3 Имена аргументов метода

и ещё кое-что...



Что в этом коде можно улучшить?

Design guidelines:

<https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/>



Что в этом коде можно улучшить?

Код читается больше, чем пишется,
не нужно экономить на понятности
и чистоте кода ради скорости.

[url](#)



Что в этом коде можно улучшить?

Старайтесь **не использовать сокращения** кроме общепринятых в разрабатываемом продукте.



Что в этом коде можно улучшить?

Когда придумываете название для общедоступной единицы, старайтесь **не использовать имена**, потенциально или явно **конфликтующие со стандартными** идентификаторами.



Что в этом коде можно улучшить?

Используйте имена с простым написанием.
Их проще читать и набирать.

Избегайте (в разумных пределах)
использования слов с двойными буквами,
сложным чередованием согласных.

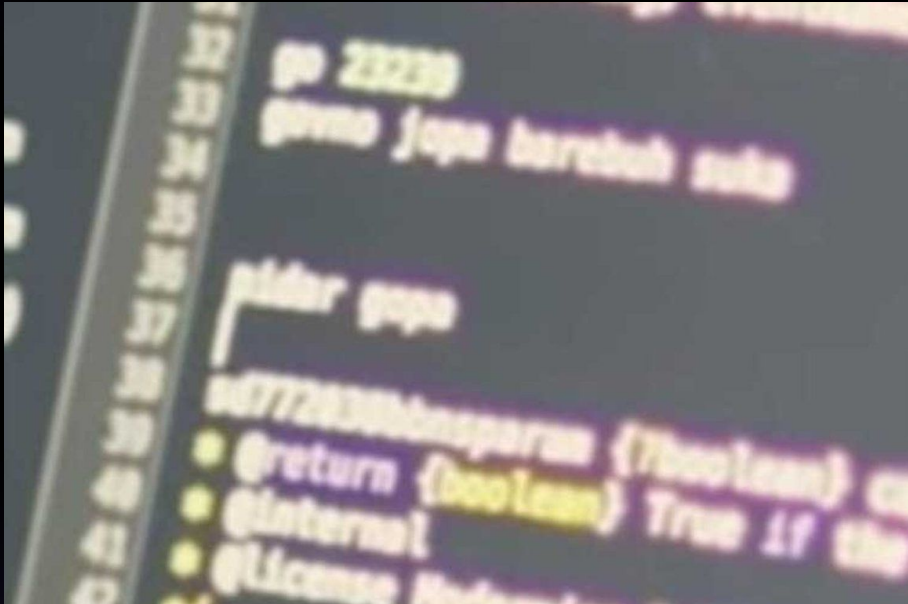


Что в этом коде можно улучшить?

Никогда не используйте матерные (запрещённые) слова в коде (в том числе в комментариях).



История с рекламным роликом гугла



Например:



Например:

- Идентификатор **HorizontalAlignment** легче читается на английском, чем **AlignmentHorizontal**
- Имя **CanScrollHorizontally** лучше, чем **ScrollableX** (неочевидная ссылка на ось X)
- **НЕ используйте** знаки подчеркивания, дефисы и другие символы, не являющиеся буквенно-цифровыми - **Can_Scroll_Horizontally**

```
int _x;
```



Например:

- Идентификатор **HorizontalAlignment** легче читается на английском, чем **AlignmentHorizontal**
- Имя **CanScrollHorizontally** лучше, чем **ScrollableX** (неочевидная ссылка на ось X)
- **НЕ используйте** знаки подчеркивания, дефисы и другие символы, не являющиеся буквенно-цифровыми - **Can_Scroll_Horizontally**
- **ИЗБЕГАЙТЕ** использования идентификаторов, совпадающих с ключевыми словами широко используемых языков программирования
string class = "1B";
<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/>
<https://docs.swift.org/swift-book/ReferenceManual/LexicalStructure.html>



Например:

- Идентификатор **HorizontalAlignment** легче читается на английском, чем **AlignmentHorizontal**
- Имя **CanScrollHorizontally** лучше, чем **ScrollableX** (неочевидная ссылка на ось X)
- **НЕ используйте** знаки подчеркивания, дефисы и другие символы, не являющиеся буквенно-цифровыми - **Can_Scroll_Horizontally**
- **ИЗБЕГАЙТЕ** использования идентификаторов, совпадающих с ключевыми словами широко используемых языков программирования
`string class = "1B";`
- **Старайтесь** не использовать сокращения, **GetWindow**, проще понять, чем **GetWin**



- **НЕ используйте** акронимы, которые не являются широко принятыми, и в целом используется акронимы только при необходимости - **SendPM**(int i, string t) точно хуже, чем **SendMessagePrivate**(int id, string text)
- **Используйте** универсальные имена платформы, не относящиеся к конкретному языку : **ConvertToByte**(string value); против **ConvertToUnsignedChar**(string value);
- **Используйте** общие, не привязанные к контексту имена, когда это нужно **ConvertToByte**(string value); лучше, чем **ConvertToByte**(string str);
- Именованное методов, аргументов, переменных и других системных единиц отличается.



Нотации:

Pascal – указание этого стиля оформления идентификатора обозначает, что первая буква заглавная и все последующие первые буквы слов тоже заглавные.

Например: BackColor, LastModified, DateTime.

Camel – указание этого стиля обозначает, что первая буква строчная, а остальные первые буквы слов заглавные.

Например: borderColor, accessTime, templateName.



Ещё:

Избегайте длинных строк, переносите инструкцию на новую строку.

Не размещайте несколько инструкций на одной строке.
Каждая инструкция должна начинаться с новой строки.

При разбивке длинных выражений на несколько строк,
оставляйте операторы на новой строке (пример ниже).



Избегайте длинных строк, переносите инструкцию на новую строку.

Не размещайте несколько инструкций на одной строке.
Каждая инструкция должна начинаться с новой строки.

При разбивке длинных выражений на несколько строк,
оставляйте операторы на новой строке (пример ниже).

```
int result = (1 + 2) - (3 + 4)
            / (15 * 16) - (17 + 19)
            / (25 * 26) - (27 + 29)
            / (35 * 36) - (37 + 39)
            ;
```

```
if (a > b
    && c > d
    && e > a
    || a == 1
    || c == f
    ^ !(g != h))
{
    Console.WriteLine("vse kruta");
}
```



Используйте автоформатирование кода вашей IDE после модификации кода.

```
if (a>b
    && c>d
    && e> a
    || a ==1
|| c == f
    ^ !(g!= h))
{
    Console.WriteLine("vse kruta");
}
```



Используйте автоформатирование кода вашей IDE после модификации кода.

```
if (a > b
    && c > d
    && e > a
    || a == 1
    || c == f
    ^ !(g != h))
{
    Console.WriteLine("vse kruta");
}
```



Моё любимое! :D

```
bool Metodi(int chislo)
{
    bool resultati = false;
    if( chislo % 2==0 )
{ resultati = true;
    }
    else{resultati = false; }
    if (resultati == false)
{ return false;
    }      else {
return true;
}
}
```



Моё любимое! :D

```
bool Metodi(int chislo)
{
    bool resultati = false;
    if (chislo % 2 == 0)
    {
        resultati = true;
    }
    else
    {
        resultati = false;
    }
    if (resultati == false)
    {
        return false;
    }
    else
    {
        return true;
    }
}
```



Моё любимое! :D

```
bool Metodi(int chislo)
{
    bool resultati = false;

    if (chislo % 2 == 0) { resultati = true; }
    else { resultati = false; }

    if (resultati == false) { return false; }
    else { return true; }
}
```



Моё любимое! :D

```
bool Metodi(int chislo)
{
    bool resultati = false;

    if (chislo % 2 == 0) { resultati = true; }
    else { resultati = false; }

    return resultati;
}
```



Моё любимое! :D

```
bool Metodi(int chislo)
{
    bool resultati = false;

    if (chislo % 2 == 0) { return true; }
    else { return false; }

}
```



Моё любимое! :D

```
bool Metodi(int chislo)
{

    if (chislo % 2 == 0) { return true; }
    else { return false; }

}
```



Моё любимое! :D

```
bool Metodi(int chislo)
{
    if (chislo % 2 == 0) return true;
    else return false;
}
```



Моё любимое! :D

```
bool Metodi(int chislo)
{
    return chislo % 2 == 0;
}
```



Моё любимое! :D

```
bool Metodi(int chislo)
{
    bool resultati = false;
    if (chislo % 2 == 0)
    {
        resultati = true;
    }
    else
    {
        resultati = false;
    }
    if (resultati == false)
    {
        return false;
    }
    else
    {
        return true;
    }
}
```



Моё любимое! :D

```
bool Metodi(int chislo)
{
    return chislo % 2 == 0;
}
```



Моё любимое! :D

```
bool IsEven(int value)
{
    return value % 2 == 0;
}
```



Моё любимое! :D

```
* bool IsEven(int value) => value % 2 == 0;
```



Моё любимое! :D

```
** var IsEven = (int v) => v % 2 == 0;
```



Ещё про bool

Не используйте проверки вида `b == false`.
Вместо этого используйте `!b`



**Такой метод
никому не нужен)**



Я тоже так думал, но потом нашёл
<https://www.npmjs.com//is-even>



↓ Weekly Downloads

164 768



↓ 2021-03-04 to 2021-03-10

210 828



↓ 2021-10-07 to 2021-10-13

217 344



Дополнение



Красивый код: какой он?

- Для методов используйте нотацию Pascal (вне зависимости от области видимости метода)

`GetStream()`

- Если метод асинхронный - он заканчивается на "Async"

`GetStreamAsync()`



Красивый код: какой он?

- Для именования переменных используйте нотацию Camel.
- Используйте неявную типизацию (`var`) для локальных переменных в случаях, когда тип переменной понятен из правой части назначения или когда точный тип не важен.

ДЕМОНСТРАЦИЯ



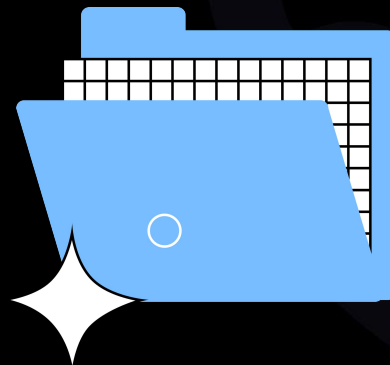
Красивый код: какой он?

- Для именования переменных используйте нотацию Camel
- Используйте неявную типизацию (`var`) для локальных переменных в случаях, когда тип переменной понятен из правой части назначения или когда точный тип не важен.
- Объявляйте переменные непосредственно перед их использованием.



Красивый код: какой он?

- Счетчики в циклах i , j , k , l , m , n , если этого не требует условие.
- Инициализируйте переменные при объявлении, если есть такая возможность.



Красивый код: какой он?

Цепочки вызовов стоят особняком



Красивый код: какой он?

Цепочки вызовов стоят особняком

```
SaveToFile (
    PrintArraySumElements (
        SumElements (
            GetArrayElements (
                PrintMatrix (
                    MultiplicationMatrixByNumber (
                        PrintMatrix (
                            FillMatrix (
                                CreateMatrix(5, 5)),
                                10)))))
```



Красивый код: какой он?

- Порядок функций, констант, методов, свойств
- Исключения, пространства имен, классы, интерфейсы, свойства, индексаторы и блаблабла
- ...а ещё в других языках



Красивый код: какой он?

Комментарии и документирование



Красивый код: какой он?

Комментарии и документирование

- Внутри блока с кодом отделяйте текст комментария одним пробелом «// Текст комментария.»
- Неиспользуемый код не комментируйте, а удаляйте.
- Если код комментируется временно - он должен быть с пометкой «// TODO: причина почему код закомментирован».
- *Используйте XML-документирование для описания публичных типов и их элементов. Начинайте текст комментария с большой буквы и завершайте текст комментария точкой.
- **Использование словарей для подсветки ошибок русского\английского языка.




Домашнее задание





Спасибо
за внимание

A yellow smiley face is drawn over the text. It has two vertical lines for eyes and a curved line for a mouth.