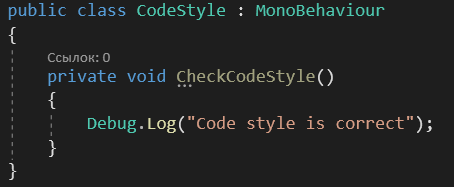
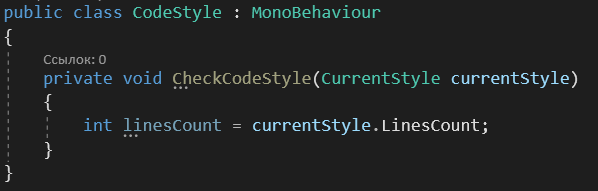
**NAMING RULES:**

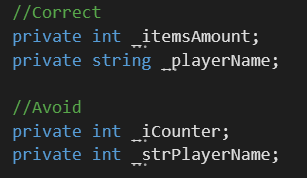
**1**.Do use PascalCasing for class names and method names



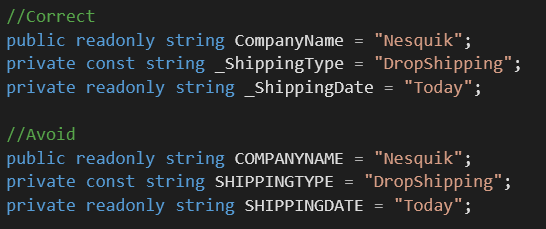
**2**.Do use camelCasing for method arguments and local variables



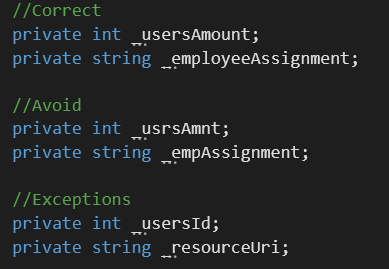
**3**.Do not use Hungarian notation or any other type identification in identifiers



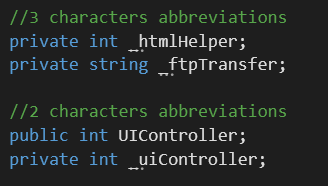
**4**.Do not use Screaming Caps for constants or readonly variables, name it in same way with common fields, but always with the first letter in uppercase



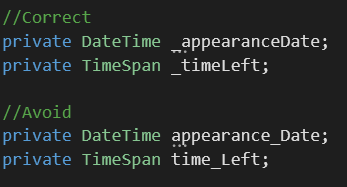
**5**.Avoid using Abbreviations. Exceptions: abbreviations commonly used as names, such as Id, Xml, Ftp, Uri.



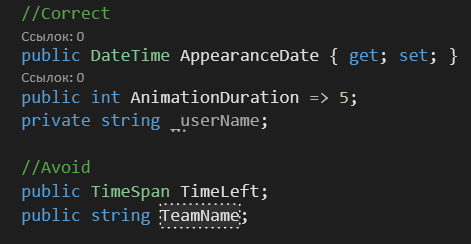
**6**.Do use PascalCasing for abbreviations 3 characters or more (2 chars are both uppercase)



**7**.Do not use Underscores in identifiers. Exception: prefix private fields with an underscore



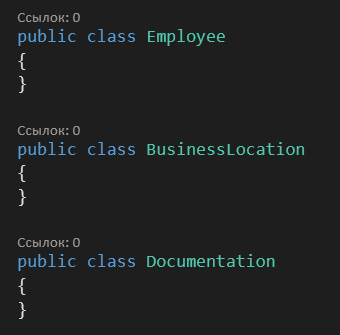
**8**.Avoid of using public fields, better to use properties



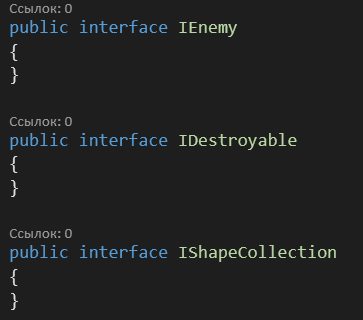
**9**.Do use implicit type var for local variable declarations when it use word new or Factory Method . Exception: primitive types (int, string, double, etc) use predefined names



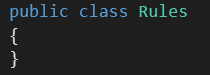
**10**.Do use noun or noun phrases to name a class



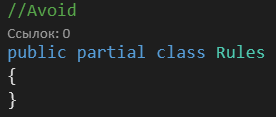
**11**.Do prefix interfaces with the letter I. Interface names are noun (phrases) or adjectives.



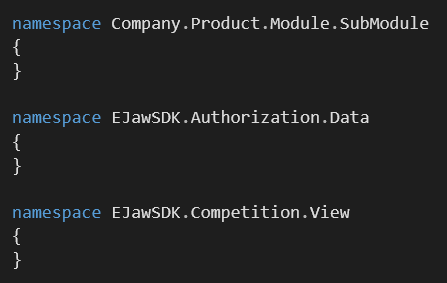
**12**.Do name source files according to their main classes.



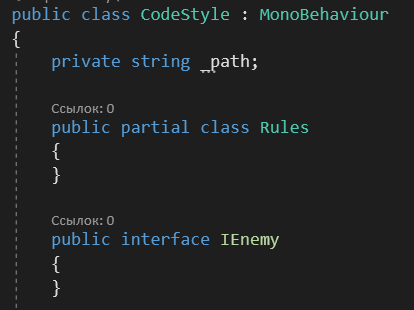
**13**.Do not use partial classes.



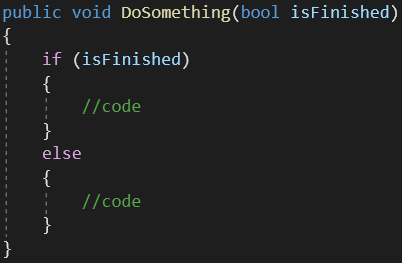
**14**.Do organize namespaces with a clearly defined structure



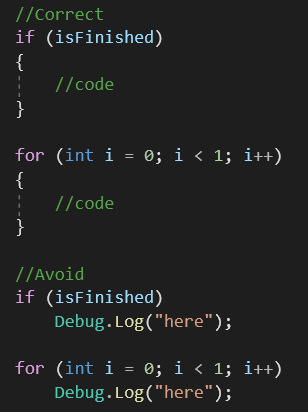
**15**.Do vertically align curly brackets



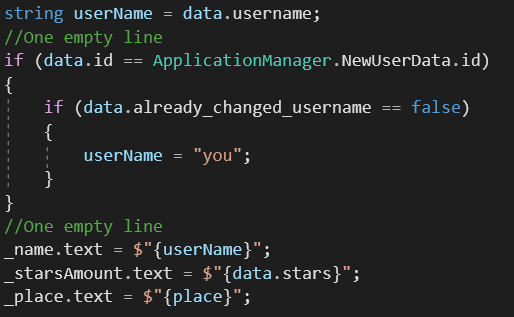
**16**.Do put the curly braces on a separate line. Here's a sample



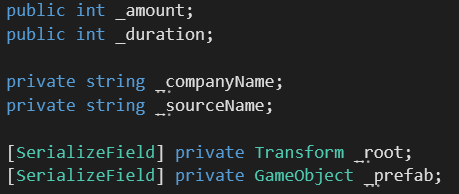
**17**.Do add the curly braces even if you have only one line run.



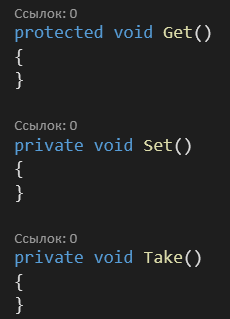
**18**. Group the code blocks by logic, separating them from the rest with an empty line



**19**. Group fields by access modifiers

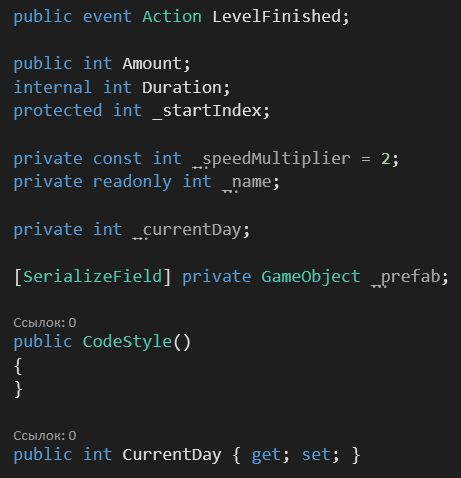


**20**. Separate methods with an empty line

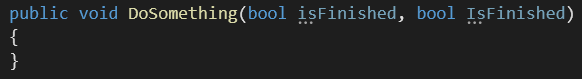


**21**. One file - one class

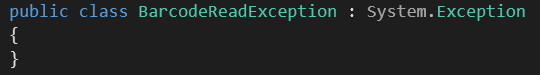
**22**.Sort by access modifiers public events ,public, internal, protected, private, [SerializeField] private, constructors, properties



**23**.Do not create names of parameters in methods (or constructors) which differ only by the register:

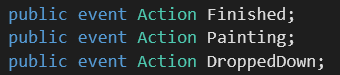


**24**.Do use suffix Exception at creation of the new classes comprising the information on exception:



**25**.Events naming rules:

**-1.**Do name events with a verb or a verb phrase.

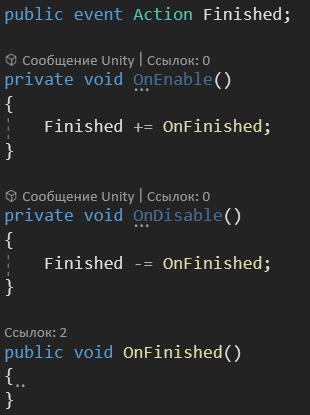


**-2.Do NOT use "Before" or "After" prefixes or postfixes to indicate pre- and post-events. Use present and past tenses as just described.**

**3.Do give events names with a concept of before and after, using the present and past tenses.**

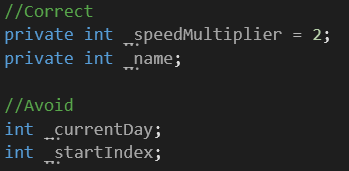
****

**26**.Do use prefix On when handle event



**27**.Delete all unused code and usings, use usings sorting

**28**.Explicitly specify access modifiers



**29**.Save comments ONLY if you will need it later, otherwise delete it. Your code should be self - documented

**30**. Use sealed if class will not be inherited

**31**. Do use prefix On and postfix Click when handle button click event

