

提高 C#编程水平的 50 个要点

1. 总是用属性 (Property) 来代替可访问的数据成员
2. 在 `readonly` 和 `const` 之间, 优先使用 `readonly`
3. 在 `as` 和 强制类型转换之间, 优先使用 `as` 操作符
4. 使用条件属性 (Conditional Attributes) 来代替条件编译语句 `#if`
5. 总是为自定义类重载 `ToString` 方法
6. 区别值类型和引用类型
7. 使用不可变的值类型 (Immutable Atomic Value Types)
8. 在值类型中, 确保 0 是一个合法的数据
9. 理解 `ReferenceEquals`, `static Equals`, `instance Equals` 和 比较运算符 (`=` `=>`) 之间的关系
10. 理解 `GetHashCode` 方法的缺陷
11. 在编写循环时, 优先使用 `foreach`.
12. 在定义变量的时候就将其初始化
13. 使用静态构造函数来初始化静态成员变量
14. 用多个构造函数时, 利用构造函数链
15. 使用 `using` 和 `try/finally` 来处理资源的释放
16. 尽量避免产生资源垃圾
17. 尽量避免使用装箱 (boxing) 和拆箱 (unboxing)
18. 实现类的 `Dispose` 方法
19. 在接口和继承 (Inheritance) 之间, 优先使用接口 (interface)
20. 区分接口和重载 (overrides)
21. 用委托 (delegate) 来实现回调 (callback)
22. 用事件 (event) 来定义外部接口
23. 避免返回类内部成员的引用
24. 使用元数据来控制程序
25. 优先使用可序列化 (serializable) 类型

26. 对需要排序的对象实现 `Comparable` 和 `Comparer` 接口
27. 避免使用 `Cloneable` 接口
28. 避免使用类型转换操作符
29. 只有当基类加入了与派生类中现有的函数名称相同的函数时，才需要使用 `new` 操作符
30. 尽量使用 `CLS-Compliant`
31. 尽量编写短少，简单的函数
32. 尽量编写比较小的程序集 (`assembly`)
33. 限定类型的可见性 (`visibility`)
34. 编写大粒度的 web API
35. 在使用事件时，优先继承基类事件，而不是重新创建一个事件
36. 多使用 framework 的运行时调试 (`DEBUG`, `TRACE`, `EVENTLOG` 等)
37. 使用 .net 标准的配置机制
38. 使用并且在类中支持 .net 的数据绑定功能 (`Data Binding`)
39. 使用 .net 的验证机制 (`Validation`)
40. 根据你的需求选择正确的集合类 (`Collection`)
41. 在自定义结构中使用 `DataSet`
42. 利用属性 (`Attributes`)
43. 不要过度使用反射 (`Reflection`)
44. 创建完整的，应用程序特定的异常
45. 尽可能多的考虑程序可能出现的异常，并作出处理
46. 尽可能少的使用 `Interop`
47. 尽量使用安全代码 (`safe code`)
48. 多多学习、使用外部工具和资源
49. 准备使用 C# 2.0
50. 学习 ECMA 标准