C#语言技术管理规范文档

2021141090183 付文君

1. 命名规范

| **#** | **规则** | **级别** |
| --- | --- | --- |
| 1 | 类名、接口名、枚举名使用 PascalCase（大驼峰） | a |
| 2 | 方法名使用 PascalCase | a |
| 3 | 变量名使用 camelCase | a |
| 4 | 常量名使用全大写+下划线，如 MAX\_RETRY\_COUNT | b |
| 5 | 接口名前缀使用 I，如 IRepository | a |
| 6 | 异步方法以 Async 结尾，如 GetUserAsync | a |
| 7 | 命名应避免缩写（如 usr），除非行业通用（如 ID、URL） | b |
| 8 | 静态字段使用 \_camelCase 命名风格 | b |
| 9 | 不使用匈牙利命名（如 strName、iCount） | a |
| 10 | 文件名应与类名一致，一个文件一个顶级类型 | a |

1. 代码结构与格式化

| **#** | **规则** | **级别** |
| --- | --- | --- |
| 11 | 使用 4 个空格缩进（禁止使用 Tab） | a |
| 12 | 每个类、接口、结构必须有访问修饰符 | a |
| 13 | 类、方法、属性之间用 1 行空行分隔 | a |
| 14 | 方法参数、调用参数过多时可换行对齐 | b |
| 15 | 属性声明使用简洁写法 { get; set; }（如无逻辑） | b |
| 16 | 空语句块 {} 不应省略大括号，即使是单行 | a |
| 17 | 避免一行多个语句 | b |
| 18 | 类的顺序推荐为：字段 → 构造函数 → 属性 → 方法 | b |

1. 编码风格

| **#** | **规则** | **级别** |
| --- | --- | --- |
| 19 | 使用 var 代替明确类型，当类型明确可读性强时 | b |
| 20 | 避免无意义的 else，例如在 if (return) 后继续 else | b |
| 21 | 尽可能使用表达式体成员（=>）简化代码 | b |
| 22 | 避免魔法值，使用常量或枚举代替 | a |
| 23 | 异常处理时，禁止捕获后不处理（catch {}） | a |
| 24 | 异常日志必须记录错误信息和堆栈 | b |
| 25 | 异步方法内部应使用 await，避免返回未等待的 Task | a |
| 26 | 禁止在非异步方法中调用 .Result 或 .Wait() 阻塞异步代码 | a |

1. 注释与文档

| **#** | **规则** | **级别** |
| --- | --- | --- |
| 27 | 所有 public 方法、类、接口必须添加 XML 注释 | a |
| 28 | 注释应准确、简洁，不能写废话注释（如 // get user name） | a |
| 29 | 使用 /// 风格 XML 注释（支持 IntelliSense） | a |
| 30 | 方法内如存在复杂业务或关键逻辑，需适当块注释解释 | b |
| 31 | 不允许注释掉废弃代码，应直接删除 | a |

1. 面向对象设计

| **#** | **规则** | **级别** |
| --- | --- | --- |
| 32 | 每个类应只关注一个职责（Single Responsibility） | a |
| 33 | 类之间使用接口解耦，禁止依赖具体类（Use DI） | a |
| 34 | 禁止方法超过 50 行或嵌套超过 3 层 | b |
| 35 | 业务逻辑代码不得出现在 Controller 中，应下沉至 Service 层 | a |

1. 测试与可维护性

| **#** | **规则** | **级别** |
| --- | --- | --- |
| 36 | 所有公共业务方法必须具备单元测试 | a |
| 37 | 单元测试命名格式：MethodName\_Scenario\_ExpectedBehavior | b |
| 38 | 测试类命名格式为：类名 + Tests，如 UserServiceTests | a |
| 39 | 测试用例应覆盖正常、边界、异常路径 | b |
| 40 | 使用 Mock 隔离外部依赖，不可连接真实数据库 | a |