Python技术管理规范文档

一、A强制规范（20条）

编码标准

1. 必须使用UTF-8编码，禁止使用其他编码格式
2. 必须通过pylint --score=10质量门禁（得分≥9.5）

依赖管理

1. 必须通过requirements.txt声明依赖，版本号使用~=符号限定主版本
2. 禁止直接修改site-packages目录安装包

安全规范

1. 必须使用secrets模块替代random生成敏感数据
2. 必须通过bandit静态扫描（严重漏洞数=0）

测试要求

1. 必须达到90%+行覆盖率（使用coverage报告）
2. 必须包含pytest.mark.integration集成测试标记

部署规范

1. 必须通过Dockerfile定义运行环境，基础镜像版本锁定
2. 必须使用poetry进行虚拟环境管理

异常处理

1. 必须使用自定义异常继承BaseException体系
2. 禁止捕获泛型Exception（允许特定场景）

并发控制

1. 必须使用concurrent.futures替代直接threading调用
2. 必须设置线程池最大数量（默认=CPU核心数×5）

日志规范

1. 必须使用logging模块，禁止print输出生产日志
2. 必须配置日志轮转（默认保留7天，大小≤100MB）

类型提示

1. 公共API必须包含PEP 484类型注解
2. 禁止使用Any类型（特殊场景需注释说明）

代码结构

1. 必须使用\_\_init\_\_.py显式声明包结构
2. 禁止相对导入（允许测试代码例外）

二、B推荐规范（25条）

性能优化

1. 推荐使用lru\_cache装饰器缓存计算密集型函数结果
2. 推荐对I/O密集型操作使用asyncio异步编程

代码风格

1. 推荐使用Black自动格式化（line-length=120）
2. 推荐使用isort自动排序import语句

API设计

1. 推荐使用Pydantic进行数据验证
2. 推荐对REST API使用FastAPI框架

配置管理

1. 推荐使用python-decouple管理环境变量
2. 推荐将配置分为development/staging/production三级

监控集成

1. 推荐使用OpenTelemetry实现分布式追踪
2. 推荐通过Prometheus暴露/metrics端点

文档规范

1. 推荐使用Sphinx生成API文档（包含doctest示例）
2. 推荐对复杂逻辑添加# noqa: E501行注释说明

测试策略

1. 推荐使用hypothesis进行属性测试
2. 推荐对关键路径实施混沌工程测试

CI/CD

1. 推荐在GitHub Actions中配置pre-commit钩子
2. 推荐使用tox管理多环境测试矩阵

代码复用

1. 推荐通过\_\_all\_\_显式导出公共接口
2. 推荐使用functools.cached\_property替代属性装饰器

异常日志

1. 推荐使用logging.exception记录异常堆栈
2. 推荐对第三方API调用设置重试逻辑（指数退避）

三、C允许规范（7条）

1. 允许在性能关键路径使用@cython.locals优化
2. 允许对科学计算使用Numba JIT加速
3. 允许在数据管道使用Pandas UDF
4. 允许对遗留代码使用# type: ignore注释
5. 允许在开发环境使用pdb调试器
6. 允许对原型开发使用Jupyter Notebook（需转换）
7. 允许在测试代码使用unittest.mock