Python 技术管理文档

学号：2022141461129　　姓名：张雅秋

一、 背景

Python 是当今重要的开发语言。本文档主要列举了 Python 程序应采纳和避免的编码风格规范。每条规范根据其重要性分为以下三个级别：

- A（强制，Mandatory）：必须严格遵守的规则，违反会导致代码可读性、可维护性或兼容性问题。
- B（推荐，Recommended）：建议遵循的规则，在大多数情况下能提升代码质量，但允许在合理场景下例外。
- C（允许，Permitted）：非强制性规则，可根据团队或项目需求灵活选择是否采用。

下文中的规范标注 A、B 或 C，分别对应上述三个级别。

二、 Python 代码风格规范

2.1 分号

A 禁止在行尾添加分号，禁止使用分号将两条语句合并到同一行。

2.2 行宽

A 每行最大长度不得超过 80 个字符，但在以下例外情况下可以超过：长的导入语句、注释中的 URL 或路径名、不便于换行的模块级长字符串常量、Pylint 禁用注释。

A 禁止使用反斜杠进行显式续行，必须使用圆括号、中括号或花括号实现隐式续行。

2.3 括号

B 避免不必要的括号使用，元组可以但不强制使用括号包围，返回语句或条件语句中除非用于隐式续行或表示元组，否则不应使用括号。

2.4 缩进

A 必须使用 4 个空格作为缩进，禁止使用制表符。使用隐式续行时，应对齐括号内元素或使用 4 个空格的悬挂缩进。

2.5 序列尾部逗号

B 仅当闭合符号与最后一个元素不在同一行时，推荐在序列尾部添加逗号。

2.6 Shebang 行

C 主程序文件可以但不强制添加 Shebang 行，格式应为 #!/usr/bin/env python3 或 #!/usr/bin/python3。

2.7 注释和文档字符串

A 模块、函数、方法的文档字符串必须使用三重双引号 """ 格式，遵循 PEP-257 规范。
A 模块级文档字符串应包含一行概述和详细描述，函数文档字符串应描述调用语法和语义。
B 复杂代码块前应添加注释解释，行尾注释应与代码间隔至少 2 个空格。

2.8 标点符号和语法

B 注释应使用正确的标点符号、拼写和语法，保持与记叙文一样的可读性。

2.9 字符串

A 优先使用 f-string 格式化字符串，其次是%运算符或 format 方法。
A 禁止在循环中使用+或+=拼接字符串，应使用列表 join 或 StringIO。
B 保持同一文件中字符串引号的一致性，多行字符串推荐使用"""。

2.10 日志

A 日志函数的第一个参数必须使用字符串字面量而非 f-string，用%占位符格式。

2.11 错误信息

A 错误信息必须精确匹配错误条件，插入片段清晰可辨，便于自动化处理。

2.12 资源和状态管理

A 使用完文件、套接字等资源后必须显式关闭，优先使用 with 语句管理资源。

2.13 TODO 注释

B TODO 注释应以全大写 TODO 开头，后跟括号包含的上下文标识符和具体说明。

2.14 导入格式

A 导入语句必须各自独占一行（typing 和 collections.abc 导入除外）。
A 导入应按标准库、第三方库、本地代码的顺序分组，每组间空一行，组内按字典序排序。

2.15 语句

B 通常每个语句应独占一行，但简单的 if 语句可以与执行体放在同一行。

2.16 访问器和设置器

B 当访问或设置变量值会产生有意义的作用时，应使用 get_foo()和 set_foo()形式的访问器/设置器。

2.17 命名

A 模块名使用小写加下划线，类名使用驼峰式，函数/方法名使用小写加下划线，常量使用全大写加下划线。
A 避免使用单字符名称（除计数器和迭代器等特殊情况）、连字符名称、首尾双下划线名称。

2.18 主程序

A 可执行文件的主要功能应放在 main()函数中，并在 `if __name__ == '__main__':`条件

下调用。

2.19 函数长度

B 函数应保持小巧专一，超过 40 行应考虑拆分。

2.20 类型注解规范

A 方法中的 self 或 cls 参数不需要类型注解，除非有额外类型信息。
B 公开 API 必须包含类型注解，内部函数可酌情省略。

2.21 换行

B 类型注解导致签名过长时，应按参数换行，返回值类型可单独一行。

2.22 前向声明

A 使用未定义类名时应通过 from __future__ import annotations 或字符串实现。

2.23 默认值

A 同时有类型注解和默认值的参数，等号周围应加空格。

2.24 NoneType

A 可能为 None 的变量必须显式声明，使用 X | None 或 Optional[X]形式。

2.25 类型别名

B 复杂类型应定义别名，命名采用驼峰式，私有别名加下划线前缀。

2.26 忽略类型

C 可使用# type: ignore 禁用特定行类型检查，但应谨慎使用。

2.27 变量类型标注

B 难以自动推断类型的内部变量应使用带类型注解的赋值操作。

2.28 容器类型

A 列表类型注解应指定单一元素类型，元组可指定多个位置类型。

2.29 类型变量

A 类型变量应有描述性名称，除非完全不可见且无约束。

2.30 字符串类型

A 新代码应使用 str 表示文本数据，bytes 表示二进制数据。

2.31 导入类型

A 从 typing 和 collections.abc 导入符号时应直接导入符号本身。

2.32 条件导入

C 仅在特殊情况下可使用条件导入，放在 if TYPE_CHECKING:块内。

2.33 循环依赖

B 应重构代码避免类型注解导致的循环依赖，必要时可使用 Any 临时解决。

2.34 泛型

A 使用泛型类型时必须指定类型参数，不应默认为 Any。

2.35 文件组织

A 测试文件应与被测试代码保持相同目录结构，以 test_前缀命名。

2.36 代码审查

A 所有代码变更必须经过代码审查，强制规范项必须修正。

2.37 静态分析

A 必须使用 flake8 或 pylint 进行代码风格检查，推荐使用 mypy 进行类型检查。

2.38 格式化工具

B 推荐使用 black 自动格式化代码，使用 isort 自动排序 import 语句。

2.39 测试要求

A 核心功能必须包含单元测试，覆盖率不低于 80%。

2.40 文档要求

A 公共 API 必须包含完整的 docstring，描述参数、返回值和可能抛出的异常。

2.41 依赖管理

A 必须使用 requirements.txt 或 Pipenv/Poetry 管理依赖，生产环境必须固定版本。

2.42 安全实践

A 禁止使用 eval()和 exec()处理用户输入，SQL 查询必须使用参数化查询或 ORM。