Практическая часть

1)Метрики Стилистики

Идентификаторы:

1. Список идентификаторов

- currencies

- currency\_data

- end\_date

- start\_date

- current\_date

- formatted\_date

- url

- response

- root

- date\_record

- valute

- char\_code

- value

- nominal

- normalized\_value

- writer

- df

2. Оценка понятности

- Все идентификаторы: 4

3. Средняя длина идентификаторов

- Средняя длина = (9 + 12 + 10 + 11 + 12 + 14 + 3 + 8 + 4 + 9 + 5 + 5 + 7 + 15 + 17 + 6) / 16 = 9.75 символов

Комментарии:

1. Процент строк с комментариями

- Комментарии: 18 из 71 строк кода ≈ 25%

2. Оценка достаточности комментариев

- 4 (достаточно для понимания кода)

Структура:

1. Средняя длина строки кода

- Пример: 92 символа

2. Соблюдение PEP 8

- Нарушения:

- Отступы в 1-2 местах вместо 4 пробелов.

- Отсутствие пробелов вокруг оператора: value = float(...) вместо value = float (...).

2)Метрики стилистики важны для поддержания читаемости и сопровождаемости кода. Они вносят ясность и упрощают совместную работу, что снижает время на обучение и исправление ошибок.

Примеры метрик стилистики:

- Длина идентификаторов: Помогает понять назначение переменной; длинные имена делают код понятнее, но слишком длинные могут быть неудобными.

- Количество комментариев: Соотношение комментариев к коду помогает понять логические связи; недостаток комментариев затрудняет восприятие.

- Длина строки кода: Оптимальная длина улучшает читаемость; слишком длинные строки затрудняют понимание.

- Глубина вложенности: Чем меньше уровень вложенности, тем легче следить за логикой выполнения.

Слишком короткие имена переменных (x, tmp) делают код неясным, затрудняя понимание их роли.

Анализ:

Сильные стороны:

1. Хорошая структура функций - легко переиспользовать код.

2. Сбалансированное количество комментариев - помогает понять общую логику.

Слабые стороны:

1. Длинные строки кода - затрудняют чтение.

2. Короткие имена переменных - неясные.

Улучшения:

1. Разделить длинные строки кода на более короткие, используя вспомогательные функции.

2. Переименовать переменные в более описательные имена для улучшения понимания.

Заключение:

Метрики позволяют количественно оценить качество кода, предоставляя четкие и измеримые критерии для анализа его структуры и стиля. С их помощью можно выявить проблемные области, такие как чрезмерная сложность или недостаточная читаемость, что упрощает принятие решений о необходимых улучшениях. Кроме того, использование метрик способствует стандартизации подходов к программированию в команде.