Теоретическая часть

1. Метрики ПО: Назначение и Применение

Метрика программного обеспечения — это количественный показатель, используемый для оценки характеристик кода, таких как сложность, читаемость и сопровождаемость.

Ключевые направления применения метрик:

1. Оценка сложности кода

- Помогает выявить сложные участки, требующие рефакторинга.

- Пример: высокая цикломатическая сложность указывает на запутанную логику.

2. Прогнозирование трудозатрат

- Позволяет оценить время на разработку и исправление ошибок.

3. Контроль качества

- Обеспечивает соответствие кода стандартам и лучшим практикам.

4. Улучшение стиля кода

- Помогает повысить читаемость и единообразие кода.

Пример применения:

Метрики могут выявить избыточную сложность функции во время код-ревью, что позволит упростить её логику.

2. Метрики Сложности

Высокосложный код сложнее понимать, тестировать и поддерживать, что увеличивает риск ошибок.

Проблемы высокосложного кода:

- Трудности при отладке.

- Сложность модификации.

- Высокая вероятность появления багов.

Другие метрики сложности:

1. Глубина вложенности — количество уровней вложенности условий и циклов.

2. Количество параметров функции — избыточное число параметров усложняет понимание.

3. Метрики Стилистики

Они оценивают читаемость и удобство поддержки кода, что критично для командной работы.

Примеры метрик стилистики:

1. Длина идентификаторов

- Слишком короткие имена (x, tmp) ухудшают понимание.

2. Количество комментариев

- Оптимальное соотношение комментариев и кода улучшает ясность.

3. Длина строки кода

- Слишком длинные строки (более 79 символов) затрудняют чтение.

4. Глубина вложенности

- Большая вложенность усложняет восприятие логики.

Пример плохой практики:

Использование x вместо current\_value снижает понятность кода.

2. Практическая часть

1. Метрики Стилистики

|  |  |  |
| --- | --- | --- |
| Идентификаторы | Понятность (1 - 5) | длина |
| currencies | 5 | 10 |
| currency\_data | 5 | 12 |
| end\_date | 5 | 8 |
| a, b, r | 2 | 1 |
| arr, m, x | 3 | 1-3 |

Средняя длина идентификаторов:

- Основной код: 8–10 символов.

- Функция calc: 1 символ.

- Функция find\_max: 1–3 символа.

Комментарии

- Общее строк кода: 87.

- Строк с комментариями: 58.

- Процент комментариев: 66%.

- Оценка достаточности: 3.

Структура

- Средняя длина строки кода: 50 символов.

- Нарушения PEP 8:

1. Отсутствие пробелов вокруг операторов (например, current\_date=start\_date).

2. Слишком короткие имена переменных в функциях (a, b, r).

2. Анализ и Рекомендации

Сильные стороны:

1. Хорошая читаемость основного кода благодаря осмысленным именам переменных (currency\_data, normalized\_value).

2. Наличие комментариев, поясняющих ключевые шаги.

Слабые стороны:

1. Слишком короткие имена переменных в функциях (calc, find\_max), снижающие читаемость.

2. Недостаточное количество комментариев для сложных логических блоков.

Рекомендации:

1. Переименовать переменные в функциях:

- a → operand1, b → operand2, r → result.

2. Добавить комментарии к логике парсинга XML и обработки данных.

3. Заключение

Метрики ПО позволяют объективно оценить качество кода, выявить слабые места и предложить конкретные улучшения. Их использование способствует повышению читаемости, сопровождаемости и надежности программного обеспечения. В данном анализе метрики помогли выявить проблемы с именованием переменных и недостатком комментариев, что позволит улучшить код при доработке.