# Теоретическая часть

## 1. Метрики программного обеспечения: назначение и применение

Метрики программного обеспечения представляют собой количественные показатели, предназначенные для анализа характеристик кода, включая его сложность, читаемость и удобство сопровождения.

Основные направления использования метрик:

1. **Оценка сложности кода**
   * Позволяют обнаружить сложные участки, которые нуждаются в рефакторинге.
   * Например, высокая цикломатическая сложность сигнализирует о запутанной логике.
2. **Прогнозирование трудозатрат**
   * Дают возможность оценить время, необходимое для разработки и исправления ошибок.
3. **Контроль качества**
   * Обеспечивают соответствие кода установленным стандартам и лучшим практикам.
4. **Улучшение стиля кода**
   * Способствуют повышению читаемости и единообразия кода.

**Пример применения:**  
Во время код-ревью метрики могут выявить избыточную сложность функции, что позволит упростить её логику.

## 2. Метрики сложности

Код с высокой сложностью труднее понимать, тестировать и поддерживать, что повышает вероятность возникновения ошибок.

**Проблемы высокосложного кода:**

* Затруднения при отладке.
* Сложности при внесении изменений.
* Повышенный риск появления багов.

**Другие метрики сложности:**

1. **Глубина вложенности** — количество уровней вложенности условий и циклов.
2. **Количество параметров функции** — избыточное число параметров усложняет её понимание.

## 3. Метрики стилистики

Эти метрики оценивают читаемость и удобство поддержки кода, что особенно важно при командной работе.

**Примеры метрик стилистики:**

1. **Длина идентификаторов**
   * Слишком короткие имена (например, x, tmp) ухудшают понимание кода.
2. **Количество комментариев**
   * Оптимальное соотношение комментариев и кода повышает 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.

# 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 количество 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), что снижает 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.