**Отчет о проблемах проектирования и сопровождаемости в ветке not-oop**

**1. Отсутствие модульности**

* **Описание проблемы:** В коде отсутствует четкое разделение на модули или классы, которые отвечают за определенные аспекты системы. Это затрудняет повторное использование кода и усложняет его тестирование.
* **Последствия:** Изменения в одной части кода могут непредсказуемо повлиять на другие части. Модификации или улучшения системы требуют значительных усилий, так как приходится вносить изменения сразу в нескольких местах.

**2. Отсутствие инкапсуляции**

* **Описание проблемы:** Данные и методы управления ими не отделены друг от друга. Отсутствие инкапсуляции ведет к тому, что внутренние детали реализации могут быть изменены извне.
* **Последствия:** Нарушение инкапсуляции делает код хрупким, так как любые изменения в структуре данных могут затронуть множество частей кода, вызывая ошибки.

**3. Отсутствие наследования и полиморфизма**

* **Описание проблемы:** В коде нет использования наследования и полиморфизма, что не позволяет переиспользовать общие функции и поведение.
* **Последствия:** Это приводит к дублированию кода, увеличивая его объем и вероятность ошибок. Также усложняет добавление новых функциональностей.

**4. Неразделение ответственности (SRP)**

* **Описание проблемы:** Методы или функции содержат логику, которая должна быть разделена между различными классами или компонентами. Например, один метод может обрабатывать несколько задач сразу, таких как обработка данных и их вывод.
* **Последствия:** Изменение одного аспекта поведения требует изменения всей функции, что ведет к ошибкам и усложняет сопровождение.

**5. Проблемы с тестированием**

* **Описание проблемы:** Отсутствие четко определенных интерфейсов и модулей затрудняет создание юнит-тестов. Функции и методы могут зависеть друг от друга, что усложняет их тестирование в изоляции.
* **Последствия:** Невозможно гарантировать стабильность кода при внесении изменений, что увеличивает риск возникновения ошибок на этапе эксплуатации.

**Заключение**

Код в ветке not-oop имеет множество проблем, связанных с проектированием и сопровождаемостью. Основные из них включают отсутствие модульности, инкапсуляции, использования принципов наследования и полиморфизма. Эти проблемы ведут к снижению качества кода, усложнению его сопровождения и повышению риска возникновения ошибок.