



Практическое задание

Тема: Осознанная стилизация диаграммы (узлы, классы, тема, связи)

Цель

Закрепить понимание иерархии стилизации в Mermaid:

Тема -> Классы -> Индивидуальный style -> linkStyle

Вы должны продемонстрировать, что умеете:

- применять `style` осознанно,
- проектировать и использовать `classDef`,
- корректно работать с темой и `themeVariables`,
- аккуратно стилизовать связи через `linkStyle`,
- избегать избыточной стилизации.

Задание (шаги)

Создать **небольшую архитектурную диаграмму** (flowchart LR), включающую:

- Клиент
- API
- Сервис
- База данных
- Внешняя система
- Обработчик ошибок

Шаг 1. Базовая структура

1. Создать `flowchart LR`
2. Добавить 6 узлов
3. Добавить связи:

```
Client -> API
API -> Service
Service -> Database
Service -> External System
Service -> Error Handler
```

Никакой стилизации на этом этапе.

Шаг 2. Подключить тему

1. Использовать встроенную тему **forest**
2. Через `themeVariables` изменить:
 - фон диаграммы
 - базовый цвет линий

Не менять больше 2–3 переменных.

Шаг 3. Создать классы

Создать минимум **3 класса**:

- `service`
- `database`
- `external`

Требования:

- Использовать `classDef`
- Применить классы к соответствующим узлам
- Не называть классы по цвету

Дополнительно:

- Создать класс `error`

Шаг 4. Индивидуальная стилизация

Использовать `style` ровно для **одного** узла — например, выделить `Client`.

Только один узел. Не больше.

Шаг 5. Стилизация связей

Использовать `linkStyle` :

- Сделать внешний вызов пунктирным
- Сделать поток ошибки красным и толще

Обязательно:

- Прокомментировать индексы связей
- Проверить корректность нумерации

Подсказки по ключевым частям

1. Инициализация темы

Используется блок:

```
%%{init: { "theme": "forest" }}%%
```

Для переменных:

```
%%{init: { "theme": "forest", "themeVariables": { "background": "#FFFFFF" } }}%%
```

2. Создание класса

```
classDef service fill:#E8F5E9,stroke:#2E7D32,stroke-width:2px,color:#1B5E20;
```

Назначение:

```
class Service service;
```

3. Индексация связей

Mermaid считает связи сверху вниз:

```
0
1
2
3
4
```

Стилизуется так:

```
linkStyle 4 stroke:#B71C1C,stroke-width:3px;
```

4. Пунктирная линия

```
stroke-dasharray:5 5;
```

Что проверить перед отправкой (чек-лист)

Структура

- ☐ Есть ровно 6 узлов
- ☐ Есть 5 связей
- ☐ Нет лишних элементов

Темы

- ☐ Использована тема forest
- ☐ Изменены только базовые переменные
- ☐ Диаграмма остаётся читаемой

Классы

- ☐ Минимум 3 класса
- ☐ Классы названы по роли, а не по цвету
- ☐ Нет дублирования inline-стилей внутри классов
- ☐ Каждый класс применяется минимум к одному узлу

Индивидуальный стиль

- ☐ Использован только для одного узла
- ☐ Не дублирует стили класса

Стили связей

- ☐ Прокомментированы индексы
- ☐ Пунктир используется осмысленно
- ☐ Красная стрелка только для error-flow
- ☐ После правок индексы не сломаны

Советы по улучшению работы

1. Минимизируйте стили

Если вы чувствуете, что хочется «ещё немного покрасить» — остановитесь. Избыточная стилизация почти всегда ухудшает читаемость.

2. Проверяйте контраст

Особенно в теме `forest` .

Зелёные оттенки могут конфликтовать.

3. Проверяйте иерархию

Убедитесь, что:

- Тема задаёт фон
- Классы задают семантику
- `style` — редкий акцент
- `linkStyle` — подчёркивание потока

Критерий успешного выполнения

Диаграмма:

- читается без объяснений,
- использует стили осмысленно,
- демонстрирует понимание архитектуры стилизации,
- не перегружена.