

Техническое задание на KLT 5.1

1. Наименование

KLT 5.1 Unified Projective Construction Graph Engine - объединённая программа проектного lambda-анализа, проверки проектной документации, смет, графиков работ, ресурсов, поставок, финансовых отклонений и генерации схем.

2. Основание разработки

KLT 5.1 объединяет четыре ветки:

1. KLT 5.0 как переход от анализа текстов к универсальной проверке проектных данных: документы, чертежи, сметы, модели, графики, ресурсы и договорные параметры.
2. Сборку 0.1 Projective Construction Graph Engine: PIX -> PEAKS -> Reper(U,I,R,D) -> lambda -> defect -> scheme -> report.
3. KLT 5.01 Alpha: выбор языка анализа, варианты реализации PEAKS-first / PIX-first / Sync / Guided, вычисление D*, ранжирование реализаций по приближению lambda -> -1.
4. Следующую ветку KLT 5.02-FIN/Market: финансовые PIX, отчётность, план/факт, риск, деловая активность и сценарная аналитика.

3. Цель

Создать единую Flutter-готовую программу, которая принимает проектные данные, строит граф целостности, вычисляет lambda-дефекты и формирует визуальные материалы:

- план-схемы;
- чертежи/схематические разрезы;
- графики риска и lambda-метрик;
- отчёты Markdown/JSON/SVG;
- основу для будущего PDF-экспорта и API-интеграций.

4. Функциональные требования

4.1. Ввод данных

Программа должна принимать JSON-проект следующей структуры:

- works - работы, зоны, даты, ресурсы, предшественники, поставки, Reper;
- resources - трудовые, машинные и материальные ресурсы;
- deliveries - поставки и документальные основания;
- costs - сметные позиции, родительские связи, суммы, коэффициенты, основания;
- finance - финансовые и план-факт строки;
- c_min, c_offer - контрольная минимальная стоимость и предложение.

В будущих версиях допускается подключение импорта TXT, CSV, XLSX, PDF, DOCX, DWG/BIM через отдельные адаптеры.

4.2. Расчётное ядро

Ядро должно реализовывать:

$$\lambda = \frac{(U-R)(I-D)}{(U-D)(I-R)}$$

$$\Delta = |\lambda + 1|, \text{Auth} = \frac{1}{1 + \Delta}$$

$$D^* = \frac{2UI - UR - IR}{U + I - 2R}$$

$$MSI = \lfloor \log(C_b/C_a) \rfloor, DI = \max(0, (C_{\min} - C_{\text{offer}})/C_{\min})$$

$$CRI = \text{norm}(\text{severities}, \Delta, MSI, DI)$$

4.3. Проверки

KLT 5.1 должна выявлять:

- высокий λ -дефект;
- сингулярный Reper;
- неполное состояние PEAKS;
- технологическое наложение работ;
- отсутствующих предшественников;
- ресурсные конфликты;
- поставки без документального основания;
- коэффициенты без основания;
- скачки сметного масштаба через MSI;
- опасную зону дефицита через DI;
- финансовые план-факт отклонения.

4.4. Визуализация

Программа должна генерировать:

1. План-схему: граф PIX@PEAKS с узлами, связями и подсветкой дефектов.
2. Чертёж: схематическое размещение работ по зонам и срокам.
3. График: бар-граф CRI, средней Δ , MSI, DI и доли дефектов.

В текущей сборке визуализации создаются как Canvas в приложении и как SVG-строки/файлы в reference-core.

4.5. Интерфейс

Интерфейс должен быть трехязычным:

- русский;
- английский;
- китайский.

Требования к интерфейсу:

- режим Material 3;
- переключатель языка;
- вкладки Dashboard / Input / Diagrams / Report / Method;
- копирование отчёта, JSON и SVG;
- демонстрационный проект для проверки запуска.

5. Нефункциональные требования

- прозрачность формул;
- возможность аудита каждого дефекта;
- отсутствие скрытых вычислений в MVP;
- расширяемость под API и серверный режим;
- сохранение юридической оговорки: результат является аналитическим индикатором и не заменяет официальную экспертизу.

6. Состав поставки

- Flutter-проект app/;
- Python reference-core core_python/;
- демонстрационные данные examples/;
- документация docs/;
- проверочные журналы verification/;
- исходные архивы предыдущих сборок legacy_sources/.

7. Приёмочные критерии

1. Пакет распаковывается без ошибок.
2. Python reference-core выполняет smoke-run на examples/demo_project.json.
3. Генерируются result.json, report.md, scheme.svg, blueprint.svg, graph.svg.
4. Flutter-проект содержит валидный pubspec.yaml и полный каталог lib/.
5. Приложение имеет RU/EN/ZH строки.
6. В документации присутствуют ТЗ, принципы работы и возможности проекта.