

# МОНОГРАФИЯ БАЗОВЫЙ ПРОЕКТ KLT-RVD

v171 - Том 0, главы 0-4: карта чтения, claim-status, Evidence-D constitution, примеры, словарь и библиография

Автор идей, конструкций, терминов, программной линии и реперной базы данных:

Курпишев Иван Борисович

Ivan Borisovich Kurpishev, Independent Researcher, Kaliningrad

Статус: volume0\_reader\_map\_evidence\_constitution\_without\_truth\_status\_inflation. Документ является рабочей редакцией новой многотомной монографии, а не финальным изданием. Все утверждения получают статус, источник, D/Dom и место в proof-ledger.

# Оглавление v171

0. Задача v171 и место тома 0 в многотомнике
1. Reader Map: как читать проект без потери смысла
2. Claim-status ladder: лестница статусов утверждений
3. Evidence-D constitution: источник, домен и достаточное основание
4. Примеры: смета, формульная цепочка, физика, химия, белок, ДНК
5. Словарь тома 0
6. Библиография: internal canon и external prior art
7. План v172-v177
8. Финальная фиксация v171

# 0. Задача v171 и место тома 0 в многотомнике

Точка v171 продолжает новую монографию "МОНОГРАФИЯ БАЗОВЫЙ ПРОЕКТ KLT-RBD". После v169, где был закреплён source canon и архитектура томов, и после v170, где появился объясняющий слой с примерами, схемами, словарём и внешней библиографией, v171 начинает полноценное развёртывание Тома 0. Том 0 нужен как входной мост: он учит читателя отличать авторскую конструкцию от классического фона, гипотезу от теоремы, математическую формулу от проверенного результата, а красивую аналогию от доменного доказательства.

Именно здесь фиксируется дисциплина чтения: каждый термин сначала объясняется обычным языком, затем получает пример, затем формулу, затем статус и только после этого включается в RBD. Да, это медленнее, чем сразу объявить всё фундаментальным законом. Зато потом не приходится вытаскивать проект из болота преждевременных победных заявлений.

Главная цель v171 - сделать так, чтобы весь дальнейший многотомник был понятен внешнему рецензенту, инженеру, программисту, экономисту, физику, химику и биологу. Для этого вводится повторяемая структура чтения:



Схема: повторяемый путь чтения и проверки в новой монографии

Эта схема не является декоративной. Она задаёт метод написания всех следующих томов: любой раздел должен показать, из какого источника взят объект, как он разложен на PIX, как собран в PEAKS, как получил Reper(R,I,U;D), какой статус ему назначен и как он входит в RBD/RPD.

## 0.1. Что именно продолжает v171

Предыдущая точка	Что она сделала	Что добавляет v171
v169	source canon, архитектура томов, первые рабочие главы	делает Том 0 самостоятельным методологическим входом
v170	объясняющая редакция, примеры, схемы, словарь, библиография	разворачивает главы 0-4 и фиксирует Evidence-D constitution
v32-v168	последовательный development-ledger физики, химии, био, ДНК и сайта	переводит развитие в единый метод чтения и проверки
PILOT-01	formula-chain audit, gap theorem, Fano barrier	становится примером claim-status и gap-discipline
v129R/v130	prelock, CGI*, publication_verified_status, физический status operator	входит в Evidence-D constitution как обязательная публичная дисциплина

## 0.2. Редакционный принцип

Том 0 не должен быть набором общих слов о методе. Он должен быть процедурным документом: как читать, как проверять, как не повышать статус, как записывать blockers, как ссылаться на prior art и как защищать авторскую новизну без присвоения

классического фона.

Reader -> example -> term -> formula -> proof/status gate -> RBD card -> domain application

Правило v171: ни один термин не должен входить в основной корпус без простого объяснения, доменного примера, формального ядра и статуса. Иначе он остаётся в словаре или appendix, а не в доказательном тексте.

## 1. Reader Map: как читать проект без потери смысла

Reader Map - это карта входа в проект. Внутренний корпус "ЛОГИКИ КУРПИШЕВА 2" велик, многослоен и местами плотен до состояния геологической породы. Поэтому внешний читатель не обязан начинать с NAPG, FOS, V\*P, химии, ASD, RBD и APK одновременно. Том 0 распределяет входы по типам читателя.

Тип читателя	Первый вопрос	Первый вход	Что не надо читать первым
Математик	каковы определения, статусы и теоремы?	C@C, Reper, lambda, D/Dom, PILOT-01, NAPG	прикладной аудит смет как старт
Физик	что реально проверено в данных?	Evidence-D, ASD, Hol_D, T129T, status operator	гносеологию и антропологию как доказательство физики
Химик	есть ли строгая химическая гипотеза?	H_DHC_chem^strict, n>=20, prelock, negative discipline	музыкальные lambda-совпадения как доказательство
Биолог	как связаны белок и ДНК?	accession, sequence hash, coordinate convention, gene->protein morphism	анalogии без source-bound morphism
Инженер/аудитор	что даёт программа для смет?	PIX, PEAKS, Reper, gap, overlap, missing-D report	глубокую NAPG-геометрию в первый день
Философ	какова теория познания?	no-promotion rule, D, Dom, статус знания	превращение философской глубины в физическое доказательство

### 1.1. Три слоя чтения

В каждом томе нужно различать три слоя: объясняющий, формальный и проверочный. Объясняющий слой отвечает на вопрос "зачем это нужно". Формальный слой отвечает на вопрос "как это определено". Проверочный слой отвечает на вопрос "что даёт право повышать статус".

Слой	Функция	Тип текста	Ошибка, которую он предотвращает
Пояснительный	вводит смысл и пример	рассказ, таблица, схема	читатель не понимает термин
Формальный	даёт определение, лемму, теорему	формулы, аксиомы, доказательство	термин остаётся метафорой
Проверочный	фиксирует gates и blockers	status ledger, evidence route	формула получает завышенный статус

### 1.2. Пять запретов для читателя и автора

Эти запреты выглядят сухо, зато они экономят годы последующих объяснений.

1. beautiful formula != proof
2. local pattern != global theorem
3. review signal != publication\_verified\_status
4. philosophical depth != empirical proof
5. registered software != automatic validation of all claims

В проекте допускаются сильные авторские гипотезы, но они должны называться гипотезами. Допускаются theorem-candidates, но они не должны жить под видом теорем. Допускаются формальные теоремы при аксиомах, но они не становятся физическими законами без внешнего корпуса, воспроизводимости и доменных gates.

## 2. Claim-status ladder: лестница статусов утверждений

Claim-status ladder - это центральная дисциплина Тома 0. Она нужна потому, что в большом корпусе неизбежно соседствуют разные типы объектов: определения, примеры, леммы, теоремы, доменные сигналы, программные результаты, философские выводы, заявки на внедрение и открытые blockers. Без статусов всё сливается в один суп, а суп, как известно, плохо рецензируется.

**Лестница статусов: переход вверх разрешён только через gates**

<b>6. verified</b>	публичный verified
<b>5. review</b>	проверочный сигнал
<b>4. formal</b>	формально при аксиомах
<b>3. thm-cand</b>	теорема-кандидат
<b>2. lemma-cand</b>	лемма-кандидат
<b>1. definition</b>	определение
<b>0. source</b>	источник

### 2.1. Основные статусы

Статус	Что означает	Что разрешено утверждать	Что запрещено
source	есть источник или файл	объект существует в корпусе	объект доказан
definition_core	дано определение	термин можно использовать	эмпирический вывод
formal_core	формальное правило внутри KLT	внутренняя связность	публикационную верификацию
lemma_candidate	есть локальный путь к лемме	поставлена задача доказательства	называть леммой без proof-object
theorem_candidate	есть устойчивый кандидат	строить proof-ledger	продавать как теорему
formal_theorem_under_axioms	доказано при явных аксиомах	использовать как математическую импликацию	объявлять физическим законом
review_signal	найден вычислительный сигнал	требовать проверки	повышать до verified
blocker	закрывающий дефект	планировать устранение	маскировать как мелочь
publication_verified_status	прошёл все gates	публично заявлять verified	получать без внешней репродукции

### 2.2. Неповышение статуса как теорема дисциплины

Новая монография должна повторять формальный запрет: если объект имеет lambda-сигнал или красивое проективное совпадение, но не имеет D, Dom, proof-object или Evidence-D gates, он не может получить publication\_verified\_status. Это не

осторожность ради осторожности. Это условие выживания проекта как научного текста.

```
If claim C lacks D or Dom attached to the exact transition,
then Status(C) <= gap / blocker / review_seed / theorem_candidate.
Status(C) != publication_verified_status.
```

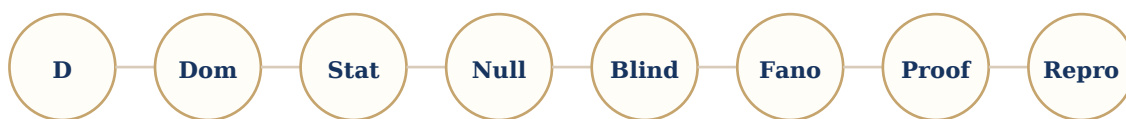
Смысл для читателя прост: метод Курпишева не обещает, что каждая найденная структура уже истина. Он обещает построить карту того, что найдено, что доказано, что требует проверки, где разрыв и что делать дальше. Да, это менее эффективно, чем лозунг "мы всё предсказали", зато намного труднее разбить рецензией.

## 2.3. Пример: одно и то же утверждение в разных статусах

Фраза	Минимальный статус	Что нужно для повышения
"lambda близка к -1"	review_signal	prelock, null, blind, D/Dom, effect size
"локально видна Fano-like структура"	review_seed / barrier	compatible identification morphism
"Hol_D превышает kappa"	review_signal	loop certificate, Reper connection, K_D faithfulness
"реакция химически допустима"	hypothesis_review	n>=20, source-bound chemistry channels, external reproduction
"смета содержит risk gap"	audit finding	документальная трасса D, экспертная проверка, protocol report

## 3. Evidence-D constitution: источник, домен и достаточное основание

Evidence-D constitution - это набор правил, по которым объект допускается к серьёзной проверке. Буква D в KLT-RBD не является украшением формулы. D - это sufficient foundation: источник, документ, proof-object, расчёт, акт, договор, accession, sequence hash, unit, uncertainty, coordinate convention или другой воспроизводимый якорь.



Evidence-D: источник и достаточное основание должны пройти все ворота

### 3.1. Минимальная структура Evidence-D

$Evidence\_D(C) = (Source, Object, Dom, D, Units, Uncertainty, Channel, Status, Blockers)$   
 $\Gamma\_C = D\_C \times Dom\_C \times Stat\_C \times Null\_C \times Blind\_C \times Fano\_C \times Proof\_C \times Repro\_C \rightarrow Status\_C$

Внутри этой конструкции CGI\* не является статистикой доказательства. CGI\* - это blocker-veto: он показывает, что есть незакрытый дефект. Доказательную нагрузку несут source, D, Dom, статистика, matched-null, blind prediction, proof-ledger и external reproduction там, где они применимы.

### 3.2. Почему D различается по доменам

Домен	Что является D	Типичный blocker
Математика	аксиома, определение, proof-object, morphism, source	нет Dom или proof dependency
Физика	source, unit, uncertainty, channel, instrument/data route	derived проху выдан за прямой канал
Химия	reaction record, условия, источник, stoich/energy/kin channels	нет $n \geq 20$ или kinetic rows
Белки	accession, organism, taxon_id, sequence_hash, structure_id	protein signal без gene/protein morphism
ДНК	accession, genome build, coordinate convention, variant_id, sequence hash	координаты без версии генома
Сметы	проектный лист, расчёт, договор, акт, график, норматив	позиция есть, основания нет

### 3.3. Prelock как защита от подгонки

Prelock нужен, чтобы метод не выбирал классы, каналы и пороги после того, как увидел удобный результат. В монографии это должно быть сказано спокойно и жёстко: если SourceManifest, ClassRegistry, ChannelRegistry, NullFamily, Thresholds, MultiplicityPlan и FailureRules не закреплены до анализа, результат остаётся exploratory seed, а не confirmatory evidence.

Prelock = SourceManifest + ClassRegistry + ChannelRegistry + AnchorPolicy + HiddenChannelPolicy + Null

Принцип: exploratory run может рождать гипотезы; confirmatory run требует prelock. Иначе проект будет выглядеть так, будто он ищет истину после того, как уже выбрал красивую цифру. Люди и так слишком часто так делают.

### 3.4. No Metaphysical Promotion Rule

В философской части проекта допускаются большие обобщения: гносеология, антропология, философия естествознания, человек как Rereg разворота, память как пересборка Rereg прошлого. Но философская глубина не повышает математический или физический статус. Для такого повышения нужен морфизм, сохраняющий D, Dom, source, proof/evidence и status gates.

PhilosophyLayer(Y) cannot promote Status\_domain(X)  
unless there exists a status-preserving morphism  $\mu: Y \rightarrow X$   
preserving D, Dom, source, proof/evidence and gates.

## 4. Примеры: как Том 0 должен обучать читателя

Примеры в Томе 0 не являются украшением. Они задают читательский навык: смотреть не на красоту формулы или важность темы, а на Rereg, D, Dom и статус. Ниже фиксируется шесть базовых примеров, которые затем должны разворачиваться в томах I-IX.

### 4.1. Смета: missing-D позиция

Строка: "Монтаж металлоконструкций - 120 т - 48 000 000 руб.". Обычный аудит видит цену и объём. KLT-RBD видит Rereg-кандидат.

Компонент	Содержание	Вопрос
R	монтаж металлоконструкций, 120 т, 48 млн руб.	что реально заявлено?
I	тип работ, норматив, проектная часть	какой инвариант или норма?
U	альтернативные поставки, этапы, смежные работы	какое поле допустимых связей?
D	проектный лист, договор, акт, расчёт	есть ли достаточное основание?

Если D отсутствует, вывод звучит так: "позиция требует документарной проверки". Не "коррупция доказана", не "смета обрушена", а missing-D или underfunding/duplication/overlap risk. Точность формулировки защищает метод от юридической и научной самоуверенности.

## 4.2. Формульная цепочка: красивое равенство без Dom

В математическом тексте найден переход: из  $cr(U,I;R,D) = -1$  сразу выводится truth-status. PILOT-01 показывает, что без admissible domain и sufficient foundation такой переход создаёт gap-node. Это идеальный пример для Тома 0: формула может быть центральной, но её статус зависит от Dom и D.

```
cr(U,I;R,D) = -1
without Dom or D
=> GAP-DOMAIN-MISSING or GAP-FOUNDATION-MISSING
```

## 4.3. Физика: Hol\_D и статус formal theorem under axioms

В физической ветке v129T/v130 Hol\_D связывается с Reper curvature, associator contraction и reduced obstruction только при явных аксиомах и условиях. Это не означает, что любое числовое превышение проху становится физическим законом. Текст тома должен различать theorem under axioms и empirical law.

```
Hol_D(gamma) > kappa_D + epsilon_num
+ Reper connection
+ faithful K_D contraction
=> non-flat predictive corridor under axioms
!= empirical physical law
```

## 4.4. Химия: отрицательная дисциплина

Химический пример особенно важен для честности проекта. Если нет prelocked class C с  $n \geq 20$ , D/Dom, p-value, q\_FDR, effect size, CGI\* $<1$ , blind prediction и external reproduction, то результат не становится химической закономерностью. Отрицательная дисциплина - это не поражение, а фильтр качества. Увы, отрицательные результаты менее праздничны, зато полезнее для настоящей науки.

## 4.5. Белок: нельзя повышать ДНК-статус без морфизма

Белковая ветка может иметь accession, sequence hash, length, features and structure\_id. Но protein signal не повышает DNA signal без source-bound morphism gene -> transcript -> protein. Такой морфизм должен сохранять organism/taxon\_id, coordinate convention, sequence hash, source, D, Dom and status gates.

## 4.6. ДНК: координата без сборки генома почти пуста

В ДНК-варианте запись "позиция 12345" без genome build и coordinate convention не является воспроизводимым основанием. Поэтому D в ДНК - это не философская приставка, а техническая необходимость: accession, сборка генома, координатная система, вариант, источник, hash.

# 5. Словарь тома 0

Словарь должен быть не только списком терминов, а навигационным устройством. Каждый термин получает простое объяснение, формальное ядро и типичный blocker.

Термин	Простое объяснение	Формальное ядро / статус
C@C	событие@состояние, минимальный объект чтения	$c=(e,s)$ , definition_core
PIX	извлечённый первичный элемент	строка, формула, число, дата, object-id
PEAKS	устойчивый смысловой узел	состояние, класс, этап, домен
Reper	четвёрка проверки факта	$Rep=(R,I,U;D)$
D	достаточное основание	документ, proof, source, hash, coordinate convention
Dom	допустимый домен	граница применимости утверждения
lambda	проективно-гармоническая координата согласованности	$lambda=cr(U,I;R,D)$
delta_truth	дефект гармонического замыкания	$ lambda+1 $
CGI*	blocker-veto, не статистика истины	critical blocker index
Evidence-D	протокол source-bound проверки	$D/Dom + gates$
RBD	активная база реперов	source, Reper, edge, status, blocker
RPD	реперно-проективный слой формульных цепочек	formula-chain audit
Fano barrier	запрет автоматической глобализации локального паттерна	requires compatible morphism
proof-ledger	реестр доказательств, зависимостей и blockers	Claim -> ProofObject -> Status
publication_verified_status	публичный verified status после всех gates	не равен internal truth-status

## 6. Библиография: internal canon и external prior art

Библиография новой монографии должна быть устроена не как вежливый список имён в конце, а как карта различия: что является внутренним источником проекта, что является классическим prior art, что является технической документацией для source-bound данных, и где именно проходит граница авторского вклада.

### 6.1. Internal canon

Источник	Роль в новой монографии
Монографии 2.0, 2.1, 2.3	исторический понятийный слой KLT/RBD
Monograph 5.0	master corpus: C@C, Reper, lambda-truth, NAPG, KLT/RBD
Monograph 5.1	no-loss extension, RPD/RBD, KLT-RPM/LIM, architecture of mathematics
Monograph 6.0	publication package and no-loss QA
PILOT-01	formula-chain audit, gap theorem, Fano barrier
SIGMA / NAPG	stratified time, Pack, Hodge, associator, reduced obstruction
v32-v168	development ledger: physics, chemistry, biology, DNA, software and site
KLT4-14, KLT5-1 SDK/APK, KLT-RBD5.10	software layer and future industrial implementation

### 6.2. External prior art and technical sources

Группа	Авторы / источники	Роль
Проективная геометрия	Klein, Coxeter, Veblen & Young, Hartshorne	cross-ratio, projective spaces, Fano background
Основания и структуры	Hilbert, Bourbaki, Mac Lane	аксиомы, структуры, категории
Стратификации	Goresky-MacPherson, Pflaum	standard stratified spaces
Hodge, G2, deformation	Bryant, Joyce, Karigiannis, Gelfand-Manin	Hodge/G2/deformation-theoretic background

Группа	Авторы / источники	Роль
Физические данные	NIST ASD, NIST transition probabilities	source-bound spectra routes
Химические данные	PubChem PUG-REST, NIST Chemistry WebBook, NIST Kinetics	compound/reaction/thermo/kinetics routes
Био/ДНК	NCBI Datasets, Ensembl REST, UniProt, RCSB PDB	sequence/protein/structure Evidence-D routes
Внешний корпус	arXiv API	independent non-KLT corpus route

В тексте следует постоянно повторять: projective geometry, Fano plane, Hodge theory, G2, categories and stratified spaces are prior art. Авторская новизна Курпишева И.Б. состоит в реперно-проективной архитектуре связи источника, формулы, D, Dom, статуса, proof-ledger, RBD и междоменных Evidence-D маршрутов.

## 7. График редакционного развития



График показывает не истинность, а готовность текста. Том 0 и Том I получают приоритет, потому что без понятного входа и базовых определений весь многотомник снова превратится в лабиринт. Том II и III затем закрепят геометрию и NAPG. Том IV свяжет RBD/RPD с предсказательным методом. Далее идут доменные тома и программный слой.

## 8. План v172-v177

Точка	Содержание	Ожидаемый результат
v172	Volume I chapters 1-5: C@C, Reper, lambda, D/Dom, FOS	понятное и строгое ядро Reper-логики
v173	Volume II: projective and packet geometry consolidated	cross-ratio, harmonicity, packet incidence, Fano barrier
v174	Volume III: NAPG, stratified time, Hodge/associator separation	формальный мост к NAPG
v175	Volume IV: RBD/RPD and predictive method	база как активный proof/status engine
v176	Volume V/VI: PHYS/CHEM unified Evidence-D	физика и химия без status inflation
v177	Volume VII/VIII: BIO/DNA and KLT-RBD5.10	био/ДНК маршруты и программный слой

## 9. Финальная фиксация v171

```
v171 = Volume 0 reader map
+ claim-status ladder
+ Evidence-D constitution
+ examples across math / estimates / physics / chemistry / biology / DNA
+ glossary
+ bibliography map
```

```
truth_status_promoted_count = 0
publication_verified_status_count = 0
```

Финальная дисциплина v171: новая монография должна быть полной, но понятной; строгой, но не перегруженной в первом входе; авторской, но без присвоения prior art; амбициозной, но без повышения статуса там, где gates ещё не закрыты.

## 10. Развёрнутый рабочий текст Тома 0, главы 0-4

Ниже фиксируется не план, а рабочий текст для будущего Тома 0. Он написан так, чтобы его можно было включать в полный многотомник после дальнейшей литературной и математической шлифовки. Главная задача этого блока - удержать понятность без потери строгости. Это редкий жанр, потому что обычно люди выбирают либо ясность без математики, либо математику без милосердия к читателю.

### Глава 0. Договор с читателем

Читатель новой монографии должен с самого начала понимать: перед ним не обычный учебник проективной геометрии, не каталог баз данных, не философская система в старом смысле и не рекламное описание программы. Перед ним авторский проект Курпишева Ивана Борисовича, где математическая идея, доказательная дисциплина, машинная база RBD и прикладной аудит должны быть сведены в одну архитектуру. Такой проект можно читать только при условии честного разделения слоёв.

Первый слой - классический фон. Он включает проективную геометрию, cross-ratio, harmonic division, плоскость Фано, категории, стратифицированные пространства, Hodge-операторы, G2-геометрию, когомологию и теорию деформаций. Эти объекты не объявляются изобретением KLT-RBD. Они являются prior art, то есть математическим фоном, на котором строится авторская архитектура.

Второй слой - авторская конструкция. Здесь вводятся  $C@C$  как событие@состояние,  $Reper(R,I,U;D)$  как проверочная четвёрка,  $\lambda$ -truth как проективно-гармонический критерий с обязательным  $D$ , Evidence-D gates, RBD/RPD как активная база доказательных карт, KLT-RBD5.10 как программный контур текстового, документарного и научного аудита.

Третий слой - статус. Он отвечает на вопрос: что именно доказано, что только определено, что является гипотезой, что является review-signal, что является blocker и что вообще пока нельзя утверждать. Без этого слоя большая монография превращается в торжественный туман, а туман, как известно, плохо проходит экспертную комиссию.

Договор с читателем: каждая сильная формулировка получает статус. Если статус не указан, текст считается editorial draft, а не доказанным результатом.

### Глава 1. Как читать формулу $Reper$

Формула  $Reper=(R,I,U;D)$  выглядит простой, но её сила именно в том, что она не разрешает факту существовать без основания.  $R$  - это то, что заявлено или дано:

строка сметы, формула, спектральный переход, реакция, белковая последовательность, ДНК-вариант, тезис статьи. I - это смысловая ось: инвариант, норма, правило, конструкция, проектный смысл. U - это поле возможностей: допустимые варианты, смежные объекты, возможные продолжения, класс сравнения. D - это достаточное основание: документ, proof-object, источник, расчёт, hash, coordinate convention, воспроизводимая трасса.

Если убрать D, получится красивая, но безответственная структура. Она может быть полезной как идея, но не как основание для доказательства, аудита или публикационного verified-статуса. Поэтому в новой монографии D всегда пишется после точки с запятой: (R,I,U;D). Эта пунктуация не декоративна. Она показывает, что основание не растворено среди других координат, а выполняет особую охранную функцию.

Пример из сметы: R - заявленная работа, I - норматив или проектная функция этой работы, U - допустимое поле аналогов и связанных работ, D - проектный лист или акт. Пример из математики: R - формула, I - инвариант или утверждение, U - класс допустимых объектов, D - доказательство или ссылка на аксиомы. Пример из физики: R - численная строка, I - физический канал, U - класс переходов или измерений, D - источник, unit, uncertainty и доменная рамка.

```
Rep_i = (R_i, I_i, U_i; D_i)
D_i absent => no_publication_verified_status
```

## Глава 2. Почему lambda не является магической истиной

Проективно-гармоническая координата lambda занимает важное место в проекте. Но Том 0 должен защитить её от неправильного чтения. Lambda не должна подаваться как мистический датчик истины. Она является координатой согласованности Rerep-четвёрки. Центральное значение -1 соответствует гармоническому замыканию. Дефект delta\_truth показывает удаление от этого замыкания.

Однако даже точное попадание в -1 не заменяет proof-ledger. Если нет D, Dom, source, proof-object или Evidence-D route, формула остаётся review structure, а не доказанный результат. Это особенно важно для прикладных областей: в смете, химии, физике или ДНК нельзя подменять документарную и экспериментальную проверку красивым числом.

Том 0 должен повторять это правило в разных формах: lambda может показать напряжение, согласованность, symmetry candidate, missing-D или interesting alignment. Но она не отменяет обязанность закрыть gates. Иначе метод превратится в numerology with rerepwork, а мы всё-таки пытаемся строить математику, а не гадальный сервис с базой данных.

```
lambda = cr(U,I;R,D)
delta_truth = |lambda + 1|
small_delta_truth => review_interest
small_delta_truth + closed_gates => higher_status_candidate
```

## Глава 3. Evidence-D как юрико-математическая гигиена

Evidence-D вводится потому, что большой междоменный проект обязан отвечать не только на вопрос "что найдено", но и на вопрос "почему этому можно доверять". В математике основанием служит доказательство. В физике - источник данных, единицы, uncertainty, instrument route. В химии - условия реакции, источник, stoich, thermodynamics, kinetics. В биологии - accession, organism, sequence hash, structure id. В ДНК - coordinate convention, genome build, variant id, source hash.

Evidence-D не делает проект слабее. Наоборот, он показывает, где именно сила результата, а где только карта следующей работы. Когда блокер закрыт, результат поднимается по claim-status ladder. Когда блокер открыт, результат остаётся честным. В нормальной науке это называется дисциплина. В человеческой практике это почему-то иногда воспринимается как занудство. Что ж, занудство часто спасает мосты от обрушения.

В прикладном аудите Evidence-D означает, что каждая критическая позиция должна иметь трассу: источник -> извлечение -> нормализация -> Reper -> статус -> отчёт. Если трасса оборвана, отчёт не должен обвинять, он должен указывать место проверки. Это делает KLT-RBD пригодным не только для научной работы, но и для юридически осторожного аудита.

source -> extraction -> normalization -> Reper -> gates -> report

## Глава 4. Почему Fano barrier нужен всей монографии

Fano barrier впервые становится для читателя ясным на простом примере: локально найденная красивая семиточечная структура ещё не является глобальной плоскостью Фано. Чтобы перейти от локального рисунка к глобальному carrier, нужны compatible identification maps. Это правило защищает не только геометрию, но и физику, химию, биологию, финансы и текстовый аудит.

В физике нельзя сказать: найден локальный паттерн, значит есть универсальный закон. В химии нельзя сказать: несколько красивых lambda-отношений, значит реакционная теория подтверждена. В ДНК нельзя сказать: похожая структура в белке автоматически доказывает ДНК-морфизм. В смете нельзя сказать: найдено совпадение, значит доказан умысел. Везде нужен переход через барьер: морфизм, source preservation, D/Dom, proof/status gates.

local Fano-like pattern  
without compatible identification morphism  
!= global Fano carrier

Поэтому Fano barrier в новой монографии является не частной геометрической деталью, а методологическим законом: локальная красота не равна глобальной истине. Видимо, человечеству это приходится объяснять отдельно, потому что красивые картинки слишком хорошо продаются.

## Глава 4.5. Как новая монография должна писать примеры

Каждый пример должен быть написан по одной форме. Сначала обычная ситуация. Затем разложение на R,I,U,D. Затем возможный дефект. Затем статус. Затем следующий шаг. Это делает монографию пригодной для чтения и для машинной обработки KLT-RBD5.10.

Шаг примера	Что писать	Минимальная польза
Ситуация	одна строка сметы / одна формула / один спектр / одна реакция	читатель понимает объект
Reper	R, I, U, D	видно, что проверяется
Дефект	missing-D, missing-Dom, overlap, proxy, no-morphism	видно, где проблема
Статус	gap, blocker, review, theorem-candidate	нет завышения
Следующий шаг	какой документ, морфизм, proof или test нужен	есть план работы

# 11. Рабочие вставки для будущего полного Тома 0

Следующие фрагменты являются заготовками для полной редакции. Их можно будет расширять, но нельзя разрушать их статусную функцию.

## 11.1. Вставка: что такое RBD простыми словами

RBD - это не полка с файлами. Это карта того, как источники превращаются в проверяемые узлы. Обычная база хранит запись: автор, название, дата, ссылка. RBD хранит больше: какой объект извлечён, как он представлен как Reper, какие есть связи, где есть gaps, какие theorem-candidates возникли, какие blockers мешают повышению статуса, какие внешние источники нужны для проверки.

Поэтому RBD можно представить как память проекта, но память не пассивную, а рабочую. Она не просто помнит старые результаты; она указывает, что делать дальше. Это важная авторская новизна: база данных превращается в предсказательно-доказательную архитектуру, а не в каталог красивых файлов.

## 11.2. Вставка: что такое KLT-RBD5.10

KLT-RBD5.10 в новой монографии следует описывать не как волшебный программный интеллект, а как disciplined software layer. Его задача - принимать источник, нормализовать его, выделять PIX, строить PEAKS, собирать Reper-карты, вычислять lambda-каналы, запускать gates, строить blockers и формировать отчёты. Программа не должна обещать абсолютное знание. Она должна давать воспроизводимую карту проверки.

Такая формулировка сильнее рекламной. Рекламное описание быстро разбивается вопросом эксперта: где валидация, где тесты, где источник, где ошибка? Дисциплинированное описание показывает: вот путь данных, вот статусы, вот ограничения, вот что уже реализовано, вот что требует пилота. Чудес меньше, доверия больше. Мир, к сожалению, устроен именно так.

## 11.3. Вставка: как писать об авторстве

Авторство Курпишева Ивана Борисовича должно быть подчёркнуто в каждом томе, но не в форме присвоения классического prior art. Правильная формулировка: классические объекты используются как известный математический фон; авторская новизна состоит в реперно-проективной, доказательно-статусной и программно-базовой архитектуре KLT-RBD, которая связывает источник, D, Dom, Reper, lambda, RBD и Evidence-D gates.

Authorial novelty = C@C + Reper(R,I,U;D) + lambda with D barrier + RBD/RPD + Evidence-D + KLT-RBD5.10

Такой стиль защищает проект от двух крайностей: от недооценки авторской новизны и от опасного утверждения, будто вся классическая математика изобретена заново. Первая крайность несправедлива, вторая уязвима. Новая монография должна быть сильной, а не удобной мишенью.

## Приложение А. Карта авторской новизны

Авторский слой Курпишева И.Б.	Классический фон, который не присваивается
C@C как минимальная онтология документа, формулы, события и состояния	классическая логика фактов и объектов
Reper(R,I,U;D) как проверочная четвёрка с обязательным D	общие идеи основания и контекста
lambda-truth как KLT-использование harmonic closure с proof/status gates	cross-ratio and harmonic division
RBD/RPD как активная база Reper-узлов, gaps, blockers and theorem-candidates	библиографические базы и графы знаний вообще
Evidence-D как междоменный source-bound протокол	общая методология данных и воспроизводимости
Fano barrier как KLT/RPD-правило запрета автоматической глобализации	классическая Fano plane
KLT-RBD5.10 как программная реализация TextWork / audit / dashboard	SQLite, CLI, software engineering как таковые

## Приложение В. Минимальный шаблон главы

Chapter template:

1. Plain-language entry
2. Example
3. Formal definition
4. Formula / theorem / protocol
5. Proof-status and blockers
6. Internal sources
7. External prior art
8. RBD card / next action