

Глубокая гигиена DOCX

Спецификация общей команды для прохода по форматированию и текстовым ошибкам документа. Рабочее пользовательское имя можно держать спокойным: **Глубокая гигиена**

DOCX; внутреннее прозвище команды - "базука".

Разделение ответственности

DOCX anomaly inspector остаётся узким модулем поиска и гигиены ошибок форматирования: секции, ориентация, таблицы, подписи, нумерация, колонтитулы, пустоты, разрывы и другие видимые структурные сбои.

Глубокая гигиена DOCX - общий агрегатор. Он объединяет форматные аномалии с текстовыми правилами: двойные пробелы, пунктуацию, единицы измерения, символы, сокращения и будущие проверки адресных обозначений.

Состав команды

Текущие режимы UI:

- Проверка** - отчёт без изменения документов, тёмно-зелёный кант.
- Корректировка** - применяет только безопасные выбранные правки к копиям DOCX, тёмно-оранжевый кант.

Команда должна собирать единый отчёт из нескольких классов проверок:

- форматные аномалии из **DOCX anomaly inspector**;
- низкоуровневая текстовая гигиена: двойные пробелы, пробелы перед пунктуацией, отсутствие пробела после пунктуации, мягкие переносы и похожий мусор;
- правила аудита: **кв. м**, **куб. м**, **% 1**, проценты, градусы, даты без **г.**, типовые опечатки в названиях таблиц;
- будущие морфологически поддерживаемые правила для единиц измерения, населённых пунктов, улиц, домов и корпусов.

В интерфейсе раздел **СТИЛИ И ГИГИЕНА** сделан как рабочая панель с переключателями **ГИГИЕНА ТЕКСТА**, **АНОМАЛИИ ДОКУМЕНТА**, **СТИЛИ ДОКУМЕНТА**. Параметры сгруппированы рамками со скруглением; чекбоксы стоят сеткой, подпись идёт на одной линии с чекбоксом, а число колонок уменьшается при узкой панели. Кнопка запуска находится в верхней командной строке справа, на уровне **НАЗАД**. Подробные пояснения вынесены в подсказки с фоном `rgb(23, 33, 43)`.

Текущий статус реализации

Первый каркас команды реализован в `system_core\services\office_service.py` как `docx_deep_hygiene`.

Уже работает:

- единая рабочая панель в разделе **СТИЛИ И ГИГИЕНА**;
- **Проход 1: Текстовая гигиена** в режимах **Проверка** и **Корректировка**;
- **Проход 1B: Правила аудита** в режимах **Проверка** и **Корректировка**;
- опциональный **Проход 2: Аномалии документа** в тех же режимах;
- цепочка корректировки: сначала текстовые копии, затем правки аудита поверх них, затем аномалии поверх результата аудита;
- единые отчёты `docx_deep_hygiene.md/.docx/.json`;
- явные границы зон ответственности `domain_boundaries`: что ищет и что исправляет каждый слой, а куда ему нельзя лезть;
- единая карта целей `target_index` и `unified_findings` с `target_fingerprint`, `issue_fingerprint`, владельцем, риском, autofix и статусом `active` / `suppressed_duplicate`;
- реестр владения классами ошибок и заметки о пересечениях в JSON/MD/DOCX отчётах.

Пока не подключён в общий запуск будущий морфологический слой адресов/топонимов.

Двухпроходный режим

Глубокая гигиена DOCX должна работать как двухпроходная объединяющая команда:

1. **Проход 1: Текстовая гигиена** - существующая старая гигиена ошибок в тексте, пунктуации, пробелах, обозначениях и предметных правилах аудита. Внутри текущей

реализации это два последовательных слоя: `docx_text_hygiene_*` и `docx_audit_processor`.

2. **Проход 2: Форматирование** - форматные и структурные аномалии документа: секции, поля, ориентация, таблицы, подписи, нумерация, колонтитулы, разрывы и похожие видимые сбои.

Оба прохода должны поддерживать общие пользовательские режимы:

- **Проверка** - оба прохода дают единый отчёт без изменения документов.
- **Корректировка** - оба прохода применяют только включённые безопасные правки к копиям DOCX и пишут единый отчёт о том, что исправлено, пропущено или требует ручного решения.

База не должна становиться третьим набором правил. Она только последовательно запускает старую текстовую гигиену, слой правил аудита и форматный слой, объединяет результаты и показывает границы зон ответственности. Техническая дедупликация нужна только как защитный механизм, а не как основной смысл команды.

Границы зон ответственности

Главная задача объединённой команды - не смешать проверки в одну массу, а показать, какой слой имеет право искать и исправлять конкретный класс проблем.

Базовое правило:

- **Гигиена текста** исправляет механический видимый текст.
- **Правила аудита** исправляют детерминированные предметные текстовые нормы.
- **Аномалии документа** проверяют и частично исправляют структуру и форматирование документа.
- **Стили документа** назначают Word-стили и не переписывают содержание.
- **Унификация таблиц** - отдельная намеренная команда массового приведения таблиц, а не скрытая часть гигиены.
- **Морфология** будет отдельным контекстным слоем для адресов, топонимов и неоднозначных случаев.

Иными словами, слои не должны сливаться. Если два слоя смотрят на один объект, это не повод смешивать правила: объединённый отчёт должен показать границу и владельца класса.

Рекомендуемые поля объединённой карты результатов:

- `source_module` - внутренний модуль, который нашёл проблему;

- `class_id` - стабильный класс ошибки;
- `ownership_key` - ключ владения классом ошибки;
- `location` - человекочитаемая локация;
- `target_fingerprint` - отпечаток абзаца, таблицы, ячейки, поля или run-range;
- `evidence` - короткое доказательство;
- `proposed_fix` - предлагаемая правка;
- `risk` - `safe`, `review_required`, `report_only`;
- `autofix` - разрешена ли автоправка в текущем режиме.

В текущей реализации эти поля уже пишутся в `unified_findings`. Детальные

`target_fingerprint` строятся для `docx_text_hygiene_*`, `docx_audit_processor` и `DOCX anomaly inspector`, где есть координаты диапазона текста или локация объекта.

Корректирующие pass-ы без детальных координат могут оставаться на точности `file`, но старая text hygiene уже отдаёт range-level находки для своих видимых текстовых правил. XML-мягкие переносы `w:softHyphen` помечаются точностью `xml-element`, потому что это не диапазон внутри `w:t`.

Владение классами и технические пересечения

У каждого класса ошибки должен быть один владелец. `Глубокая гигиена DOCX` только оркестрирует проверки, объединяет отчёты, показывает границы зон и запускает разрешённые правки. Она не должна копировать логику старой гигиены, правил аудита или форматного корректора.

Базовое разделение:

- `docx_text_hygiene_*` - механическая низкоуровневая гигиена текста: двойные пробелы, пробелы около пунктуации, мягкие переносы и похожий мусор.
- `docx_audit_processor` - детерминированные предметные правила: `кв. м`, `куб. м`, `№`, проценты, градусы, даты с `г.`, типовые ошибки аудита.
- `DOCX anomaly inspector/corrector` - форматирование и структура документа: секции, ориентация, поля, колонтитулы, подписи, нумерация, таблицы, пустоты, разрывы, фиксированные высоты строк и запрет переноса текста в ячейках.
- `document_tables_unifier` - намеренная широкая унификация таблиц по команде пользователя: шрифты, границы, поля ячеек, баланс ширин, плотность.

- будущий морфологический слой - адресные обозначения, населённые пункты, улицы, дома, корпуса и неоднозначные единицы с контекстной проверкой.

Если два модуля указывают на один `target_fingerprint` или один диапазон текста, объединяющий слой должен пометить пересечение. Это технический защитный механизм: он нужен, чтобы не применить две правки к одной цели и чтобы пользователь видел, какие домены встретились на одном объекте.

Сейчас база подавляет точные дубли по `issue_fingerprint` и группирует пересечения по одному `target_fingerprint`. Разные классы на одном объекте не скрываются автоматически: они попадают в overlap-группу, чтобы пользователь видел, что несколько правил смотрят на один и тот же фрагмент или объект.

Текущие решения по пересечениям:

- мягкие переносы относятся к текстовой гигиене; правка должна быть у одного владельца, чтобы `docx_text_hygiene_*` и `docx_nonprinting_clean` не делали одно и то же дважды;
- `NBSP` и пробелы внутри единиц допускаются в связке `текстовая гигиена -> правила аудита`, но предметная нормализация единиц остаётся за `docx_audit_processor`;
- общий поиск пробела после точки выключен по умолчанию и как общее правило пропускает типовые сокращения/единицы: `кв.м`, `куб.м`, `руб.`, `г.Энск`, `ул.Садовая`; находить и исправлять их должен `docx_audit_processor` или будущий морфологический слой с учётом контекста;
- текст в таблицах принадлежит текстовой гигиене, а структура/оформление таблиц - anomaly/table unifier;
- стили владеют стилем абзаца, аномалии - логикой подписи, номера, близости объекта и человекочитаемой локации;
- удаление зачёркнутого текста не является обычной гигиеной и должно требовать явного параметра или режима финализации.

Текущий статус морфологии

В portable runtime уже встроен `pymorphy3`; существующий `system_core\morph_replace.py` использует его для поиска по леммам и согласования заменяющей фразы. Это готовая база для будущей глубокой гигиены.

Но полноценного валидатора "единицы измерения + населённые пункты + улицы + дома" пока нет. Сейчас детерминированный процессор правил аудита уже закрывает часть единиц и

символов через строгие правила, но адресные обозначения и топонимы нужно реализовывать отдельным высокоточным слоем.

Морфологический слой

`py morphology3` использовать как помощника, а не как единственный источник истины:

- приводить слова к нормальной форме для поиска вариантов;
- отличать русские словоформы от технических токенов;
- помогать со склонением предлагаемой замены, когда правка явно безопасна;
- подтверждать контекст, но не заменять предметные словари и защитные правила.

Для адресов нужны закрытые словари и контекстные ограничения:

- `г.` может быть "город" или "год";
- `д.` может быть "дом" или "деревня";
- `п.` может быть "пункт", "посёлок" или элемент нумерации;
- формулировки вроде `городской округ город Тюмень` нельзя механически сокращать или переписывать;
- автоправка топонимов по падежу должна быть `review_required`, пока нет доказанного контекста.

Выходные артефакты

Отчёты:

- `report\docx_deep_hygiene.md`
- `report\docx_deep_hygiene.docx`
- `report\docx_deep_hygiene.json`

Исправленные копии:

- `output\hygiene_fixed` - результат текстовой гигиены;
- `output\docx_deep_hygiene_audit_fixed` - результат правок аудита поверх текстовой гигиены;
- `output\docx_deep_hygiene_fixed`

Отчёт должен показывать класс ошибки, место, доказательство, предлагаемую правку, уровень риска и статус автоправки: `none`, `safe`, `review_required`.

Локация ошибок

В объединённом отчёте локация должна быть человекочитаемой. Главный формат: страница, номер раздела, номер таблицы, номер рисунка, ближайшая подпись или заголовок. Это важнее внутренних индексов абзацев: пользователь будет искать ошибку в Word по страницам, разделам, таблицам и рисункам.

Рекомендуемые поля JSON:

- `page` - номер страницы, если удалось получить пагинацию;
- `section_number` и `section_title` - ближайший логический раздел;
- `object_type` - `paragraph`, `table`, `figure`, `caption`, `header`, `footer`, `toc`, `field`;
- `object_number` - номер таблицы, рисунка, приложения или подписи;
- `detail` - строка/колонка, поле, подпись, колонтитул или другой уточнитель;
- `search_text` - короткий фрагмент для поиска в Word;
- `debug_path` - необязательный технический путь: индекс абзаца, таблицы, XML-части или узла.

Для машинной склейки рядом с человекочитаемой локацией используется

`target_fingerprint`. Он не заменяет локацию для пользователя, а нужен для подавления дублей, группировки пересечений и будущей связи с морфологическим слоем.

В MD/DOCX отчётах показывать прежде всего строку вида:

`стр. 18; раздел 1.4 "Существующее положение"; Таблица 5; ячейка R3C2`.

Если страница неизвестна, строка должна оставаться полезной:

`страница: не определена; раздел 3.2; Рисунок 9; подпись оторвана`.

Правила безопасности

- Исходные документы не изменять.
- Все правки делать только в копиях.
- Текстовые правки с неоднозначным падежом или смыслом оставлять как предложения.
- Форматные правки применять только для классов, отмеченных как `safe`.

- Для объединённой команды не скрывать, какой внутренний модуль нашёл ошибку.

Приоритет реализации

1. Сделать двухпроходный оркестратор: сначала старая текстовая/пунктуационная гигиена, затем форматные аномалии.
2. Добавить единый UI с чекбоксами классов проверок и режимами **Проверка** / **Корректировка**.
3. Завести ключи владения, чтобы старая гигиена, правила аудита, anomaly corrector и будущий морфологический слой не правили один диапазон дважды.
4. Расширить правила аудита по единицам измерения закрытыми списками.
5. Реализовать отдельный морфологический слой адресных обозначений с отчётом confidence и ручным подтверждением неоднозначных мест.