

Класс/Интерфейс	Модуль	Назначение	Методы
<div>PipelineBase (ABC)</div>	<div>src/bioetl/core/ pipeline_base.py</div>	<p>Абстрактный базовый класс табличного ETL-пайплайна. Орkestрует стадии <code>extract→transform→validate→write</code>, обеспечивая единый интерфейс, логирование, валидацию (через Pandera) и атомарную запись результатов ¹ ². Также централизованно управляет ресурсами (например, закрывает подключения) и поддерживает <code>dry_run</code> режим без записи ³ ⁴.</p>	<div>run(output: Path, dry_run: bool) -> RunResult</div> <p>Метод запуска пайплайна (transform) в режиме <code>dry_run</code> (необходимо указать <code>dry_run=True</code> в параметрах). Возвращает результат записи в виде <code>RunResult</code> объекта. ⁶ <code>ex</code></p> <div>pd.DataFrame</div> <p>Метод для загрузки сырых данных из API, файлы и <code>pd.DataFrame</code> объектов. Возвращает <code>pd.DataFrame</code> объект.</p> <p>Абстрактные методы:</p> <ul style="list-style-type: none"> <code>transform</code> - метод трансформации данных. <code>validate</code> - метод валидации данных. <code>write</code> - метод записи данных. <p>⁸ <code>va</code></p> <div>pd.DataFrame</div> <p>Метод для загрузки структуры данных из Pandera-схемы. Возвращает <code>pd.DataFrame</code> объект.</p> <p>Метод для добавления свойство <code>li</code> к <code>DataFrame</code>, в случае отсутствия <code>li</code> упорядочит <code>li</code> и добавит хэши строк ⁹ ¹⁰.</p> <div>pd.DataFrame</div> <p>Метод для записи <code>WriteResult</code> в заданный <code>Path</code> записывает в Parquet по <code>Path</code> успешной записи запуская (на <code>Path</code> информации о количестве записей). ¹² <code>re</code></p> <div>client: Any</div> <p>Метод для внешнего реестра для <code>client</code>. ¹³ <code>cl</code></p> <p>Закрывает все ресурсы, вызывая <code>dispose()</code> метод, но не прерывает</p>

Класс/Интерфейс	Модуль	Назначение	Методы
SourceClientABC	src/bioetl/core/ pipeline/source/ client.py	<p>Интерфейс клиента внешнего источника данных. Описывает unified API для извлечения данных из внешних систем (REST API, баз данных и т.п.), скрывая детали HTTP/DB запросов. Поддерживает одиночные и пагинированные выборки данных</p>	<p>fetch_one Record N возвращает или None, 23 .
 fe ClientRequ Возвращает полученным разбиение п 25 .
 it ClientRequ Возвращает данных (каж записей и и 26 27 .
 me Any] : Пред (например, c маршруты) Закрывает с клиента, есл сессии HTTP</p>

Класс/Интерфейс	Модуль	Назначение	Методы
BaseExternalDataClient	src/bioetl/core/ pipeline/source/ client.py	<p>Базовый класс для клиентов внешних API, реализующий SourceClientABC. Содержит общую логику пагинации и сбор данных из REST API (циклический запрос страниц, объединение параметров и т.п.)³². Конкретные клиенты (ChEMBL, PubChem и др.) наследуются от этого класса, переопределяя при необходимости специфичные части.</p>	<p>fetch_one Record N возвращает страницы ре ничего не по 33 .
 fe ClientRequ Итеративно используя постраничн 25 .
 it ClientRequ Выполняет э результатов параметр сл (page_para 34 .
 me Any] : Возвр (базовый UR 29 .
 cl используем RequestsH определен м</p>
RateLimiterABC	src/bioetl/core/ pipeline/source/ client.py	<p>Интерфейс ограничителя частоты запросов. Обеспечивает соблюдение лимитов нагрузки при обращении к внешнему API (например, не более N запросов в секунду), чтобы избежать превышения квот или блокировок²⁰.</p>	<p>acquire() приостанав пока не наст следующего заданным л время).
 внутренние (например, п окна времен</p>

Класс/Интерфейс	Модуль	Назначение	Методы
TransformerABC	src/bioetl/core/ pipeline/ processing/ transform.py	Интерфейс трансформации данных. Определяет детерминированные преобразования входного DataFrame без взаимодействия с внешними системами – например, очистку значений, переименование колонок, приведение типов – перед дальнейшим обогащением и валидацией ⁴³ .	transform pd.DataFrame чисто функц изменяя вне новый DataF данными. Га одинаковым идентичный (детермини
LookupEnricherABC	src/bioetl/core/ pipeline/ processing/ enrich.py	Интерфейс обогащения данных с помощью внешних справочных данных (lookup). Отвечает за добавление в таблицу недостающей информации из внешних источников/словарей (например, подтягивает данные по внешним ключам) ⁴³ . Использует поставщика побочных данных (SideInputProviderABC) для доступа к справочникам.	enrich(df SideInputP pd.DataFrame DataFrame, и данные из добавлять н значения ид данными из новый DataF информаци
SideInputProviderABC	src/bioetl/core/ pipeline/ processing/ enrich.py	Интерфейс поставщика сторонних (побочных) данных для обогащения. Абстрагирует доступ к справочникам, кэшам или внешним источникам, которые используются в процессах обогащения (lookup) для получения дополнительной информации ⁴³ .	get_data(Предоставля заданному к Например, м словарь с да относящими таблице, есл может осущ файла, БД, А подготовлен
DeduplicatorABC	src/bioetl/core/ pipeline/ processing/ dedup.py	Интерфейс для выявления и удаления дубликатов в данных. Определяет контракт консолидации записей с одинаковыми бизнес-ключами и устранения повторов перед загрузкой данных, обеспечивая идемпотентность пайплайна ⁴³ .	deduplica pd.DataFrame записи в Dat определенн совпадения DataFrame, и (при необхо информаци стратегии сл использует вычисления хэшировани MergeStra дубликатов.

Класс/Интерфейс	Модуль	Назначение	Методы
BusinessKeyDeriverABC	src/bioetl/core/ pipeline/ processing/ dedup.py	Интерфейс вычисления бизнес-ключа записи. Бизнес-ключ – человечески осмысленный уникальный идентификатор сущности (например, комбинация нескольких полей), используемый для определения дубликатов независимо от технических первичных ключей ⁴⁴ ⁴⁵ .	derive_key Вычисляет бизнес-ключ (словаря) на основе подмножества полей. Например, для ключом может быть название+высказанная строку, одноименная бизнес-сущность.
HasherABC	src/bioetl/core/ pipeline/ processing/ dedup.py	Интерфейс хеширования записи. Определяет способ вычисления детерминированного хэша (например, MD5, SHA) по содержимому записи для целей идентификации и сравнения записей ⁴⁶ ⁴⁵ . Хэши применяются для отслеживания изменений и проверки идемпотентности.	hash(record) Вычисляет контрольный хэш. Гарантирует сохранение содержимого записи. Реализация хеширования строки).
MergeStrategyABC	src/bioetl/core/ pipeline/ processing/ dedup.py	Интерфейс стратегии слияния дубликатов. Определяет, как несколько записей, признанных дубликатами, объединяются в одну “консолидированную” запись перед загрузкой, чтобы информация не терялась ⁴⁷ .	merge(records) Объединяет дубликаты записей в одну запись, предпочтительно с максимальной информацией. Реализация объединять запись, содержащую данные.
SchemaProviderABC	src/bioetl/core/ pipeline/ validation/ schema.py	Интерфейс поставщика схемы данных. Предоставляет описание структуры и типов данных, ожидаемых на этапе валидации, абстрагируя конкретную реализацию схем (Pandera, Pydantic и т.п.) от остальной логики пайплайна ⁴⁵ .	get_schema() Возвращает объект схемы данных. Например, DataFrames аналог из других данных, обработанных. Конкретные детали схемы прогрева получать из

Класс/Интерфейс	Модуль	Назначение	Методы
ValidatorABC	src/bioetl/core/ pipeline/ validation/ validator.py	Интерфейс валидатора данных. Обобщает применение схемы и правил качества данных (DQ) к результирующему DataFrame, формируя отчет о найденных нарушениях. Позволяет отделить проверку данных от бизнес-логики пайплайна ⁴⁵ .	validate() Validation на соответс правилам ка SchemaPro и набор DQ проверок (н обязательно результата в сведения о п проверках (н ошибок) ⁴⁵
DQRuleABC	src/bioetl/core/ pipeline/ validation/ dq_rules.py	Интерфейс правила качества данных (Data Quality Rule). Определяет отдельное проверочное правило, не выражаемое лишь схемой (например, межколоночные зависимости, допустимые диапазоны значений, уникальность и пр.). Выделение DQ-правил позволяет гибко расширять проверки качества без изменения схемы ⁴⁸ .	check(df: list[DQIss качества к D обнаруженн пустой спис проверку. Ка проблему (н допустимого включает сс Реализация проверять д результаты.
PathStrategyABC	src/bioetl/core/ pipeline/output/ path_strategy.py	Интерфейс стратегии формирования путей вывода. Определяет, как строятся пути и имена файлов для результатов пайплайна, чтобы обеспечить детерминизм и организованность вывода (например, единообразное именование по сущности, источнику, дате) ⁴⁹ .	make_path str, base_ Генерирует для данного название па запуска и ба может добав подпапки по входных зна одинаковый
WriterABC	src/bioetl/core/ pipeline/output/ writer.py	Интерфейс компонента записи данных. Отвечает за непосредственную запись итогового DataFrame на диск (в файл) с обеспечением атомарности и соблюдением форматов. Инкапсулирует детали сортировки, выбора формата (CSV/Parquet) и предотвращения частично записанных результатов ⁴⁹ .	write(df: > None): Заг DataFrame в должна: отс столбцам (за пайплайна), расширени .parquet), имя и затем атомарности записи долж

Класс/Интерфейс	Модуль	Назначение	Методы
MetadataWriterABC	src/bioetl/core/ pipeline/output/ writer.py	Интерфейс записи метаданных запуска. Отвечает за сохранение вспомогательной информации о выполнении пайплайна – статистики и параметров запуска – в отдельный файл (например, meta.yaml рядом с данными) ⁴⁹ .	write_metadata(meta_path: str) – записывает метаданные в файл. Реализация должна обеспечить необходимую идентификацию времени начала этапов, коли- флаг dry_run – флаг, хэш-человекочитаемый YAML или JSON
ConfigResolverABC	src/bioetl/core/ pipeline/utils/ config.py	Интерфейс загрузки и разрешения конфигурации пайплайна. Инкапсулирует логику чтения конфигурационных файлов (например, YAML), слияния с дефолтами, применения переменных окружения и других преобразований, обеспечивая единый способ получения настроек для запуска пайплайна ⁵¹ .	resolve(config: PipelineConfig) – возвращает пайплайн. Реализация должна возвращать конфигурацию. Также может возвращать секреты и значения. SecretProvider – валидации и проверки.
SecretProviderABC	src/bioetl/core/ pipeline/utils/ config.py	Интерфейс доступа к секретам (пароли, токены API и др.). Предоставляет абстракцию над хранилищем чувствительных данных (например, переменными окружения, Vault, AWS Secrets Manager), позволяя безопасно подключать секреты в конфигурацию пайплайна ⁵¹ .	get_secret(name: str) – Возвращает имя/ключ источника (на окружения с обращение и обеспечение напрямую в может генер
CacheABC	src/bioetl/core/ pipeline/utils/ cache.py	Интерфейс кэша для промежуточных данных. Предоставляет унифицированный доступ к механизму сохранения и повторного использования данных между запусками или между этапами (например, кэширование результатов запросов к API, чтобы не дублировать вызовы) ⁵¹ .	get(key: str) – из кэша по такой записи value: Any заданный объект (перезапись существовал определять размера и п

Класс/Интерфейс	Модуль	Назначение	Методы
LoggerAdapterABC	src/bioetl/core/ pipeline/utils/ logging.py	Интерфейс адаптера логирования. Обеспечивает унифицированный способ структурированного логирования событий пайплайна, не зависящий от конкретной библиотеки логирования. Позволяет привязывать контекст (например, <code>run_id</code> , имя пайплайна) и логировать события пайплайна консистентно ⁵¹ .	<code>bind(**fi</code> Logger Adap логгер/адап контекстным <code>run_id</code> для сообщений) <code>**kwargs:</code> информаци названием/ полями дан значение в <code>str, **kwa</code> событие уро фиксирован ситуаций, со данными.
TracerABC	src/bioetl/core/ pipeline/utils/ tracing.py	Интерфейс трассировки выполнения. Позволяет интегрировать пайплайн с системами трассировки (distributed tracing) или собственным механизмом отслеживания выполнения этапов. Предоставляет методы для обозначения начала/конца важных операций, что упрощает отладку и мониторинг производительности ⁵¹ .	<code>start_spa</code> Отмечает на или этапа с трассировоч например, с распределе трассировке <code>None</code> : Отме операции. И продолжите Реализация <code>start_spa</code> контексту.
ErrorPolicyABC	src/bioetl/core/ pipeline/utils/ error.py	Интерфейс политики обработки ошибок. Определяет стратегию реагирования на исключения, возникающие во время выполнения пайплайна: продолжать выполнение, пропускать текущий элемент, повторять попытку или прерывать пайплайн ⁵² . Позволяет отделить логику обработки сбоев от основного кода пайплайна.	<code>on_error(s</code> Exception) оркестратор на стадии <code>s</code> <code>ErrorActi</code> дальше (нап текущую за конвейер). Р исключения этапа и друг политики.

Класс/Интерфейс	Модуль	Назначение	Методы
ProgressReporterABC	src/bioetl/core/ pipeline/utils/ progress.py	Интерфейс агрегированного репортинга прогресса пайплайна. Собирает и регистрирует статистику по ходу выполнения: количество успешно обработанных записей, количество отброшенных/невалидных, причины отбраковки и пр. – без влияния на сам процесс обработки ⁴⁵ ⁵³ . Разделяет сбор статистики от бизнес-логики, предоставляя единый источник правды о ходе выполнения.	on_stage_ Фиксирует н (например, л логирует ста этапа). str) -> No обработанны (вызывается обработки, л валидаторов указанием причины).<k str) -> No стадии, окон (время, коли отброшенны и сохранены

1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 50	PipelineBase_description.md	https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/_docs/PipelineBase_description.md
2 20 43 44 45 46 49 51	architecture_plan.md	https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/_docs/architecture_plan.md
19 47 48 52	abstract_objects_implementation_plan.md	https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/_docs/abstract_objects_implementation_plan.md
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42	client_abc.py	https://github.com/SatoryKono/ChEMBL_data_acquisition/blob/14d2538df65b6d02dac4cb4703ee7ba34b501e32/src/bioetl/clients/base/client_abc.py
53	ProgressReporterABC.md	https://github.com/SatoryKono/bioactivity_data_acquisition/blob/98f17f432532cddd56d0a07fd4796b37c54677ec/_docs/ProgressReporterABC.md