

# Архитектура CLI

1. Вся информация между запусками команд хранится в глобальном Controller. Внутри него прописана логика пайплайна CLI, а также хранятся переменные окружения.
2. UI считывает строки ввода и передаёт их в лексер. Также он выводит итоговый результат работы команды.
3. Лексер разбирает приходящую строку на токены. Нюансы токенизации в приложении.
4. Подстановщик делает подстановку переменных окружения.
5. Лексер повторно токенизирует строку.
6. Парсер выделяет команды и их аргументы.
7. В фабрике для каждой команды инстанцируется класс.
8. Интерпретатор по очереди выполняет метод `execute` у каждого объекта, передавая вывод предыдущей команды следующей. Вывод последней команды интерпретатор возвращает в контроллер, а он передает его в UI.
9. Обработка исключений происходит в функции `run` контроллера. Все тело цикла `while True` оборачивается в контекстный менеджер, который будет перехватывать исключения и передавать сообщения об ошибках в UI пользователю.
10. При создании конструктор команды принимает список строк - аргументы и в случае `UnknownCommand` имя команды. У команды есть метод `run`, который принимает строку (`stdin`) и возвращает строку (`stdout`).

## Примечания

На диаграммах белым цветом обозначены компоненты, которые будут реализованы в первую итерацию. Светло-бежевым обозначены компоненты, которые будут реализованы на второй итерации, а градиентом от белого к бежевому обозначены компоненты, которые будут реализованы на первой итерации, но которые придётся расширить во время второй итерации.

## Приложение

Токенизация разбивает строку сначала по '[', которые располагаются вне кавычек. Затем каждый блок разбирается по отдельности.

Выделяются части, обрамлённые одинарными кавычками. Такая часть сразу становится токеном. Выделяются части, обрамлённые двойными кавычками. Внутри двойных кавычек ищем \$ и дальнейший текст после \$ до следующего разделителя, состоящий из букв, цифр, \_. Получившиеся фрагменты выделяются в токены. Остальное разделяется по пробелам и равно на токены.