

# Задание 1: проектирование CLI

Жилкин Феодор

Смирнов Александр

28 сентября 2020 г.

## Поведенческая диаграмма

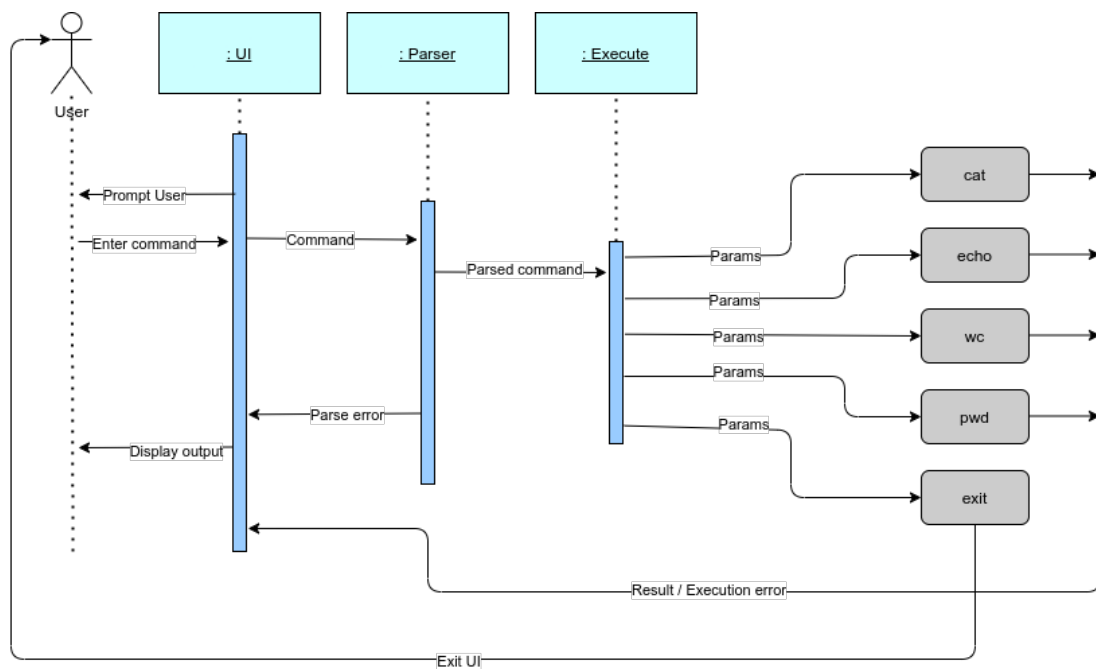


Рис. 1: Поведенческая диаграмма

## Диаграмма классов

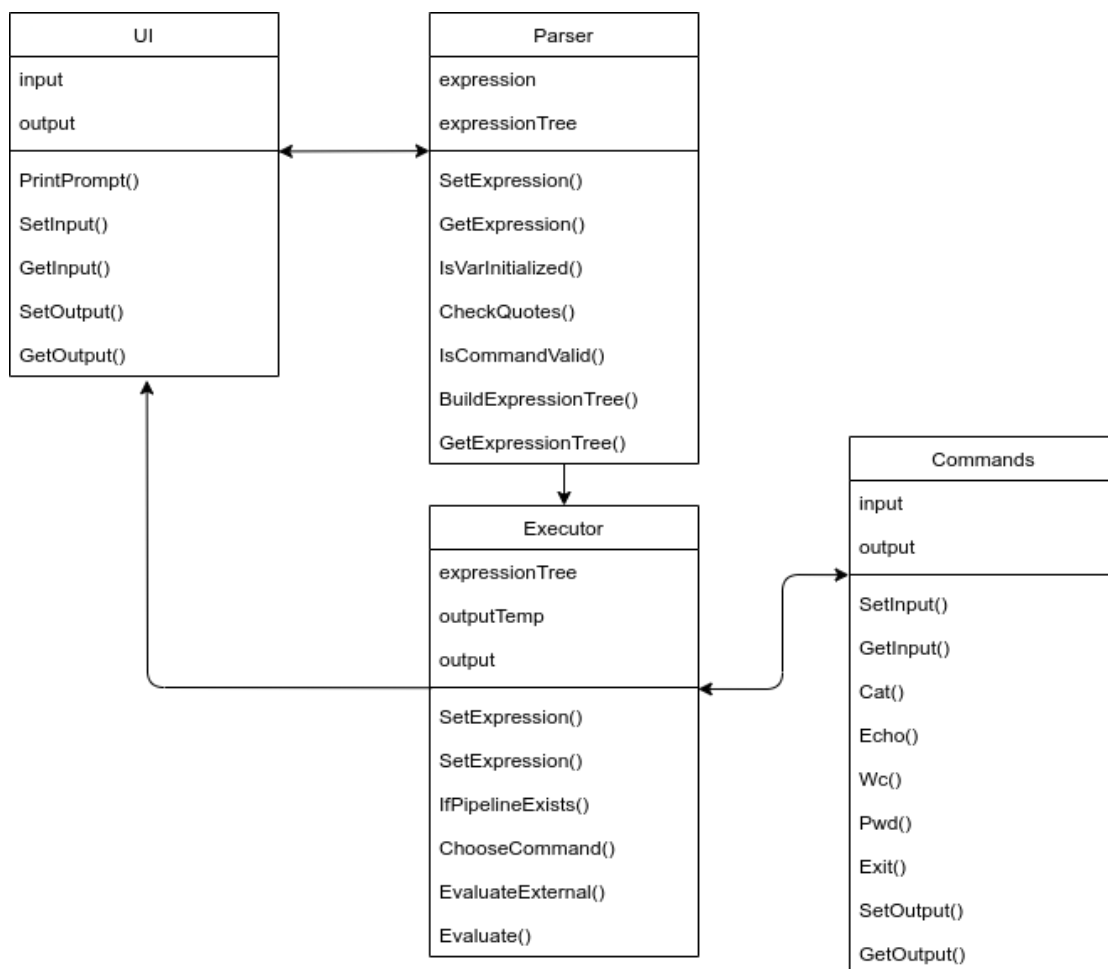


Рис. 2: Диаграмма классов

## Схема работы

- UI запрашивает у пользователя входные данные
- Пользователь вводит команду
- Команда попадает в Parser
  - Parser строит синтаксическое дерево разбора
    - \* Представляет сложную команду набором простых команд
    - \* Если встречаем переменную, проверяем, проинициализирована ли она в окружении
    - \* Если встречаем переменную в двойных кавычках, подставляем значение вместо переменной
  - Проверяет команду на валидность
    - \* Существует ли команда в \$PATH
    - \* Если команда валидна, передаем её в Execute
    - \* Если команда не валидна, возвращаем ошибку в UI
- Если в команде присутствует pipeline, то отправляем на исполнение первую команду
  - Если введённая команда не поддерживается интерпретатором, то исполняем внешнюю команду
    - \* Подаём внешней команде аргументы на stdin
    - \* Если команда завершилась с ошибкой, передаём ошибку из stderr в UI
    - \* Если команда завершилась успешно, возвращаем результат работы команды из stdout
  - Если введённая команда поддерживается интерпретатором, то выбираем команду для исполнения
    - \* Сравниваем введённую команду с множеством команд интерпретатора
    - \* Если это exit, то выходим из UI
    - \* Подаём команде аргументы на stdin
    - \* Если команда завершилась с ошибкой, передаём ошибку из stderr в UI
    - \* Если команда завершилась успешно, возвращаем результат работы команды из stdout
  - Результат команды передаём в Execute в качестве stdin следующей команде
- Если в команде отсутствует pipeline, то отправляем команду на исполнение и возвращаем результат в UI