

Состав команды:

Саврасов Михаил

Носивской Владислав

Левин Лев

Широков Кирилл

Введение

Реализация на Python.

Pipeline

Приложение получает строку, ее обрабатывает Lexer, возвращая список токенов. Parser по списку токенов строит AST. Executor выполняет AST.

Lexer

Для лексера пишутся правила разбора лексем. Правила -- это обертки над регулярными выражениями в формате библиотеки [ply](#).

Parser

Для построения дерева вызовов парсер использует грамматические конструкции из токенов, которые сопоставляются соответствующим им функциям. Согласно формату библиотеки, в одной функции пишется как грамматическое правило, так и построение новой вершины AST.

Executor

Обходит AST, поочередно вызывая метод ехес у вершин дерева.

Работа с модулями программы

Lexer

Лексер оперирует правилами, которые сопоставляют подстроке входной строки заданную нами лексему.

Parser

Используемые парсером правила имеют следующий формат:

<последовательность слов грамматики> —> <имя функции> <список аргументов>

Executor

Принимает <имя функции> <список аргументов> в каждой вершине дерева, вызывает функцию с переданными аргументами.

Работа с переменными окружения

Осуществляет Executor при обходе AST, вызывая код функций '\$' (выдача значения переменной окружения) и '=' (присваивание переменной значения окружения)

Многопоточность

Единственный тонкий момент многопоточного выполнения программ -- одновременная запись в переменные окружения или доступ к файлам. Переменные окружения хранятся в отдельных файлах. На каждый файл, к которому предоставляется доступ, берется lock.