|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный  исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по индивидуальной работе №2  по дисциплине «Язык программирования Python»  Вариант 4 | | |
|  | | Работу выполнил  студент группы ИТ-4-2024 1 курса  Михайлов К.Н.  «9» Июня 2024 г. |
| Работу проверил  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фамилия И.О.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Постановка задачи 3](#_Toc153130027)

[Алгоритм решения 3](#_Toc153130028)

[Тестирование 3](#_Toc153130029)

[Код программы 3](#_Toc153130030)

# Постановка задачи

Наша задача - сделать абстрактную вычислительную машину, которая преобразует выражение в постфиксной форме в набор инструкций. Запрещено использовать сторонние библиотеки, а так же нужно выполнить задание с использованием списков.

# Алгоритм решения

Класс LinkedList — класс для создания односвязный списков с методами «push», «pop», «reverse» а так же «\_\_init\_\_» и «\_\_str\_\_».

Класс Node — класс для создания узла для списка. Есть поля: «data» и «next».

Класс Engine — класс для преобразования постфиксной формы в набор инструкций из заданий.

Конвертируем постфиксное выражение в односвязный список с помощью класса LinkedList в классе Engine. В условии варианта прописано, чтобы мы использовали списки, поэтому было решено использовать cписок. Конвертируем в классе Engine, в списке «stack» мы указываем все буквы алфавита, далее из этих элемента списка «stack « мы берем по два элемента и оператор из словаря «operators» и записываем в в другой список «instructions». Далее мы по случаю создаем переменные в начинающие с T и заканчивающиеся переменной «temp\_count» (T1, T2 … Tn). Данные переменные мы добавляем в «stack» и снова проходимся по циклу. После окончания цикла мы получаем список «instructions» который содержит отзеркаленный список с ответом. В функции «run» мы вызываем функцию «compile» и выводим ответ.

Мы только что превратили выражение в список, потом в список инструкций и вывели содержимое списка инструкций.

# Тестирование

* Вводные данные из примера: «ABC\*+DE-/» - ответ полностью соответствует примеру.
* Неправильные вводные данные (лишние оператор «\*»): «AB+\*» - программа выдала «Недостаточно операндов», программа выполнилась правильно.
* Неправильные вводные данные (отсутствие оператора): «AB» - программа выдала «Некорректный список», программа выполнилась правильно.
* Обычные вводные данные: «AB+» - программа выдала «LD A» и «AD B»

# Код программы

https://github.com/KirillMikhailov442/ikm-2-python/blob/main/README.md