**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №4

«Разработка универсального алгоритма анализа. Часть 1»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-32Б |  | преподаватель каф. ИУ5 |
| Яковлев Сергей |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

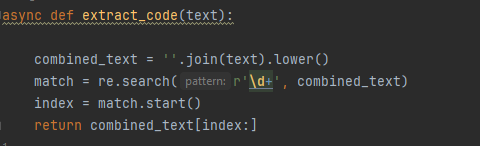
### Задание:

Разработать модуль, что будет отвечать за непосредственную расшифровку полученного клейма.

**Выполнение:**

Т.к. приёмочные клейма в массе своей достаточно линейны, то большой сложности их расшифровка не приставляет.

1. Выделим значимый для нас код:



1. Пройдёмся по БД в поисках значения с нужным кодом:



Пример работы c различным качеством исходного текста:



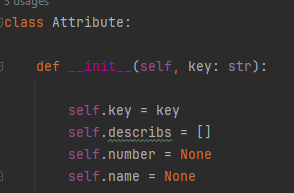


Задача считывания же ЛОЗа оказывается гораздо менее тривиальной. Как выяснилось при анализе исходных данных, клейма далеко не всегда набивались системно (как, например, в Красной армии). Спустя множество попыток лучшим вариантом алгоритма был признан следующий:

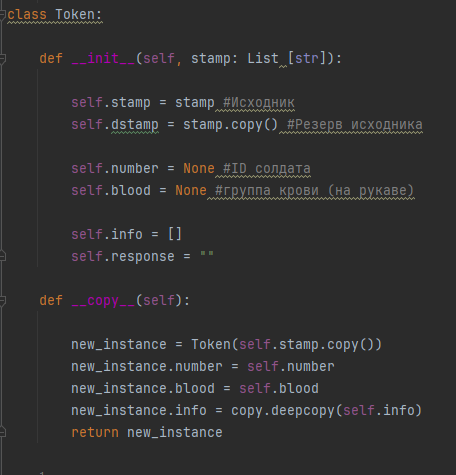
1. Разделим все входящие сокращения на 3 типа:

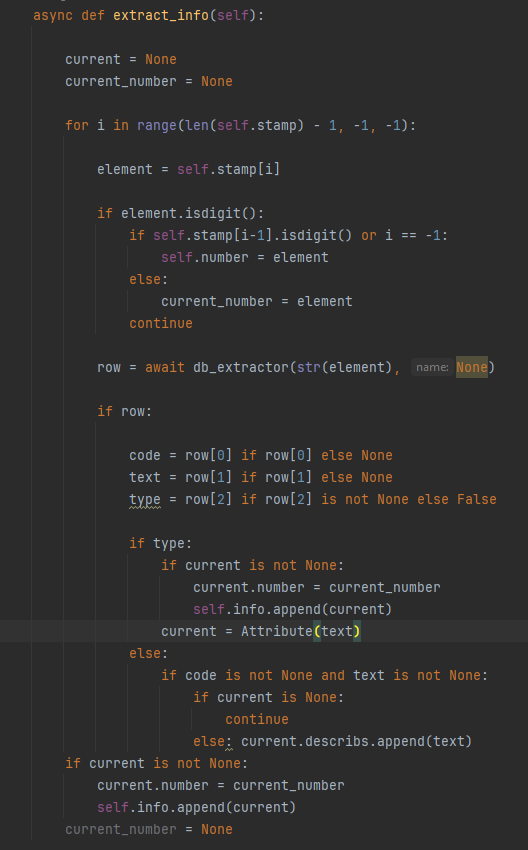
* Числовое - 
* Ключевое - 
* Описание - 

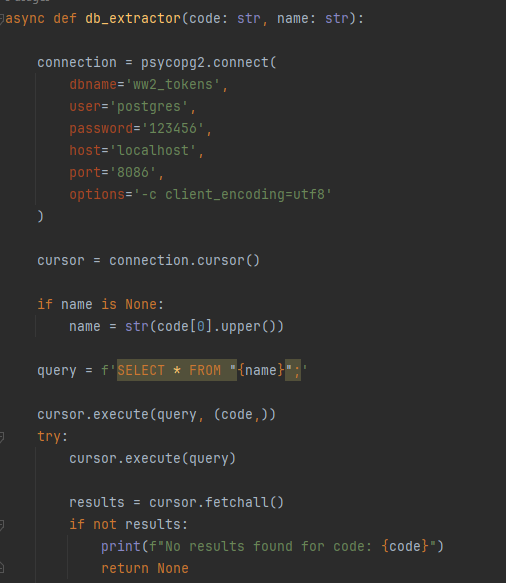
1. Заметим, что описания зачастую идут после ключевых элементов, а числовые – перед. По этому признаку сформируем структуру “Атрибут”, которая будет хранить части исходного текста в следующем виде:
   * + Ключ-значение – содержит ключевое значение для каждого атрибута
     + Описание – массив описаний
     + Номер – числовое значение каждого атрибута
     + Имя – для именных формирований

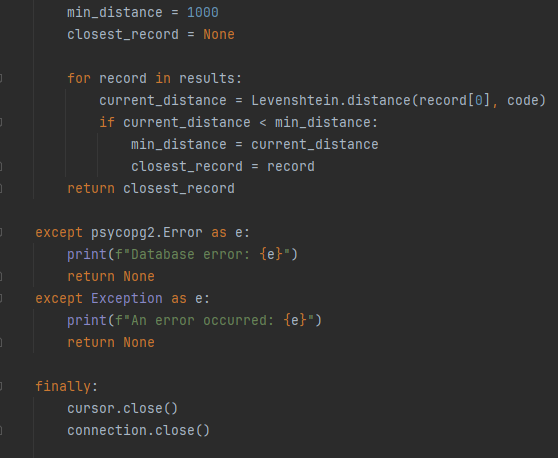


1. Напишем класс ЛОЗа и базовую расшифровку без учета **исключительных** моментов:

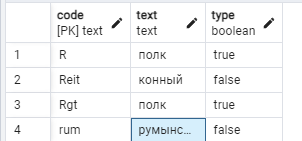








Пример того, как хранятся данные в БД:



Пример того, как на данный момент работает расшифровка:

