

Конвертер TOML и обработчик констант

Этот скрипт предназначен для преобразования конфигурационных файлов в формате TOML в кастомный формат, а также для обработки констант в коде. Он поддерживает парсинг TOML-файлов, замену значений констант в выражениях и перевод данных в заданный формат.

Описание

Скрипт включает в себя следующие ключевые функции:

- **Парсинг TOML:** Преобразует строку в формате TOML в словарь Python.
- **Перевод значений:** Преобразует значения из словаря Python в кастомный формат:
 - Строки переводятся в формат `'строка'`
 - Булевы значения `True/False` заменяются на `true/false`
 - Списки и словари преобразуются в формат `{ элемент1. элемент2. ... }` и `[ключ => значение]`.
- **Обработка констант:** Разбирает строки с определениями констант (например, `def pi = 3.14159`) и заменяет их в коде на соответствующие значения.
- **Замена констант:** В коде осуществляется замена выражений вида `!(константа)` на значение константы, если оно было определено.

Установка

1. Скачайте или клонируйте репозиторий.
2. Убедитесь, что у вас установлен Python 3.6 или выше.
3. Установите зависимость для работы с TOML:

```
pip install toml
```

Использование

Запуск через командную строку

1. Запустите скрипт, передав входной текст через стандартный ввод:

```
Get-Content input_file.toml | python config_language.py | Out-File  
output_file.txt
```

Здесь `input_file.toml` — это файл, содержащий исходный текст конфигурации, а `output_file.txt` — файл, куда будет записан результат.

2. Скрипт поддерживает замену значений констант в коде, а также преобразование TOML-данных в нужный формат.

Пример работы

Входной текст (input_file.txt):

```
def pi = 3.14159
radius = 10
area = !(pi) * radius * radius

[settings]
resolution = "1920x1080"
fullscreen = true
sound_volume = 80
```

Выходной текст (output_file.txt):

```
area = 3.14159 * radius * radius

settings => [
  resolution => '1920x1080',
  fullscreen => true,
  sound_volume => 80
]
```

Функции класса Translator: parse_toml(input_text) Парсит строку с данными в формате TOML и возвращает соответствующий словарь Python. При ошибке разбора выводит сообщение об ошибке.

translate_value(value) Преобразует значение в кастомный формат:

Преобразует строки в формат 'строка' Преобразует булевы значения в true/false Преобразует числа в строковый формат Преобразует списки и словари в соответствующий формат

translate_constants(input_text) Ищет определения констант в коде (например, def pi = 3.14159) и сохраняет их для дальнейшего использования.

replace_constants(text) Заменяет в тексте все вхождения вида !(константа) на значение константы. Если константа не найдена, заменяет на 'undefined'.

translate(toml_data) Преобразует данные в формате TOML в кастомный формат.

Пример констант и их замены

```
def pi = 3.14159
radius = 10
area = !(pi) * radius * radius
```

После замены:

```
area = 3.14159 * radius * radius
```

```
PS C:\Users\redmi\Documents\Config\Config--Yura\Config--3> python test_config_language.py
test_task_manager_config PASSED
test_cloud_storage_config PASSED

All tests passed successfully!
```