

# Учебный конфигурационный язык

## Инструмент командной строки для трансляции в YAML

### Описание

В рамках данного задания реализован инструмент командной строки для учебного конфигурационного языка. Инструмент выполняет разбор входного текста, написанного на специальном конфигурационном языке, выявляет синтаксические ошибки и преобразует корректное описание конфигурации в формат **YAML**.

Для разбора синтаксиса используется специализированный инструмент — **Lark**, что соответствует требованиям задания.

---

### Возможности языка

Учебный конфигурационный язык поддерживает следующие конструкции:

#### 1. Многострочные комментарии

```
=begin  
Это многострочный  
комментарий  
=end
```

Комментарии полностью игнорируются при трансляции.

---

#### 2. Числа

Поддерживаются целые числа:

```
10  
-42
```

#### 3. Строки

Строки задаются в одинарных кавычках:

```
'Это строка'
```

---

#### 4. Словари

Словари задаются в специальном синтаксисе:

```
([
  name : 'server',
  port : 8080
])
```

Результат трансляции:

```
name: server
port: 8080
```

## 5. Объявление глобальных констант

Константы вычисляются **на этапе трансляции** и могут использоваться в дальнейшем:

```
global A = 10
global B = 3
```

## 6. Константные выражения (постфиксная форма)

Поддерживается вычисление выражений в обратной польской записи (RPN):

```
.(A B +).
.(A B mod).
```

Поддерживаемые операции:

- + — сложение
- - — вычитание
- \* — умножение
- mod — остаток от деления

## Примеры конфигураций

### Пример 1. Математические вычисления

```
global A = 10
global B = 3

.(A B +).
.(A B mod).
```

Результат (YAML):

```
- 13
- 1
```

### Пример 2. Конфигурация сервера

```
global PORT = 8080

([
  host : 'localhost',
  port : PORT,
  workers : .(PORT 1000 /).
])
```

### Пример 3. Графическая сцена

```
global WIDTH = 1920
global HEIGHT = 1080

([
  window : ([
    width : WIDTH,
    height : HEIGHT
  ]),
  pixels : .(WIDTH HEIGHT *).
])
```

## Установка зависимостей

Для работы инструмента необходимы следующие библиотеки:

```
pip install lark pyyaml
```

## Запуск программы

### Базовый запуск

```
python config_tool.py --input examples/math.cfg --output out.yaml
```

### Запуск с отладочным выводом

```
python config_tool.py --input examples/math.cfg --output out.yaml --test
```

В режиме `--test` выводятся значения глобальных констант и результат интерпретации.

## Обработка ошибок

При наличии синтаксической ошибки во входном файле программа завершает работу с сообщением об ошибке:

```
Syntax error:  
Unexpected token ...
```

Также выявляются ошибки использования необъявленных идентификаторов.

---

## Реализация

Решение реализовано с использованием следующих компонентов:

- **Lark** — построение синтаксического дерева
- **Transformer** — преобразование дерева в абстрактное синтаксическое представление (AST)
- **Интерпретатор** — вычисление константных выражений и формирование результата
- **PyYAML** — сохранение результата в формате YAML

Ручной синтаксический разбор не используется.

---

## Заключение

Разработанный инструмент удовлетворяет всем требованиям задания:

- используется специализированный парсер,
- реализованы все конструкции учебного языка,
- обеспечено выявление синтаксических ошибок,
- поддерживается трансляция в YAML,
- представлены примеры конфигураций из разных предметных областей.