

Dokumentacja

Projekt „Język do operacji na walutach”

Mateusz Śledź

Implementacja

Program napisany jest w języku python. Wykorzystane przeze mnie biblioteki to między innymi „ply”, która została użyta do implementacji lexera oraz parsera. Zaimplementowany interpreter przetwarza drzewo AST wygenerowane przez parser.

Testowanie

Testy w module *test_interpreter* wywołują program sprawdzając poprawność wartości przez niego zwracanych w zależności od przekazanych instrukcji.

Aby wykonać wszystkie testy należy wykonać komendę:

```
python3 -m pytest test_interpreter/
```

Opis użytkowy

Wspierane elementy

- instrukcje warunkowe `if`
- pętla `while`
- operatory: `+`, `-`, `*`, `/`, `|`, `&`, `==`, `!=`, `>`, `<`, `>=`, `<=`
- funkcje – możliwość zwracania wartości przy pomocy `return`

1. Deklaracja zmiennej

```
var euro = 42 [EUR]
var dolary = 55 [USD]
var i = 5
```

2. Przykład pętli `while`. Ten prosty fragment kodu oblicza ile euro (całkowitoliczbowo) mieści się w 100 dolarach.

```
var dolary = 100 [USD]
var i = -1

while(dolary > 0 [USD])
{
    dolary = dolary - 1 [EUR]
    i = 1 + i
}

print(i)
```

3. Definicja i wywołanie funkcji

```
def func( x, y) {  
  
    . . .  
}
```

Wywołanie :

```
func(1,2)
```

Lub jeżeli funkcja zwraca jakąś wartość:

```
var x = 0  
x = func(1,2)
```

Jeżeli funkcja zwraca wartość, która jest walutą konieczne jest zadeklarowanie x jako zmiennej o wybranej walucie (wartość zwracana może być w dowolnej walucie):

```
var x = 0 [USD]  
x = func(1,2)
```

4. Zwracanie wartości z funkcji.

Po słowie kluczowym return może występować zmienna, liczba, liczba z przypisaną walutą jak również wyrażenie np.:

```
return 0  
return a + b*2  
return 1 + b/10  
return 2*12 [USD] + c
```

Uruchomienie program

Komenda:

```
python3 __main__.py program.txt
```

program.txt – plik z kodem