

# Domáce zadanie na druhé cvičenie

30. novembra 2022

## 1 Úvod

Tradičný spôsob zápisu aritmetických výrazov je *infixový*, čo znamená, že binárnu operáciu píšeme vždy medzi operandy:  $7 + 12$ ,  $18/9$  a podobne. Tento spôsob má tú nevýhodu, že občas musíme používať zátvorky aby sme určili prioritu operácií:  $(7 + 2) * 3$  je niečo iné ako  $7 + (2 * 3)$ .

Naproti tomu v *postfixovom* spôsobe zápisu dávame operáciu za operandy:

infixový	postfixový
$7+12$	$7\ 12\ +$
$18/9$	$18\ 9\ /$
$(7+2)*3$	$7\ 2\ +\ 3\ *$
$7+(2*3)$	$7\ 2\ 3\ *\ +$

## 2 Zadanie

Napište pythonovský modul `postfix.py` obsahujúci jedinú funkciu `eval_expr(s)`, ktorá ako argument očakáva reťazec pozostávajúci zo slov oddelených bielymi znakmi. Každé zo slov je pritom buďto celé číslo, alebo jedna z binárnych operácií  $\{+, -, *, /\}$ .

Vaša funkcia ten reťazec vypočíta čoby postfixový výraz a vráti výsledok – celé číslo. Funkcia ani kód v hotovom module nemá nič vypisovať, pýtať si vstupy ani nič podobné.

## 3 Vzor

```
>>> import postfix
>>> postfix.eval_expr("1 2 +")
3
>>> postfix.eval_expr("1 2 -")
-1
>>> postfix.eval_expr("2 1 -")
```

```

1
>>> postfix.eval_expr("2 2 + 3 *")
12
>>> postfix.eval_expr("2 2 * 3 +")
7
>>> postfix.eval_expr("2 2 3 * +")
8
>>> postfix.eval_expr("10 2 /")
5
>>> postfix.eval_expr("5 2 * 2 /")
5
>>> postfix.eval_expr("5 2 /")
2
>>> postfix.eval_expr("5 2 / 2 *")
4

```

## 4 Ako to urobiť

Najdôležitejšie je uvedomiť si, že postfixové výrazy je treba počítať pomocou zásobníka. Napríklad výraz "7 2 + 3 \*" vypočítame takto, vždy reagujúc na dané slovo:

slovo	čo sa deje	zásobník
7	dáme 7 na vrch zásobníka	<div><div></div><div>7</div></div>
2	dáme 2 na vrch zásobníka	<div><div></div><div>2</div><div>7</div></div>
+	vyberieme horné dve čísla, aplikujeme + a výsledok dáme na vrch	<div><div></div><div>9</div></div>
3	dáme 3 na vrch zásobníka	<div><div></div><div>3</div><div>9</div></div>
*	vyberieme horné dve čísla, aplikujeme * a výsledok dáme na vrch	<div><div></div><div>27</div></div>

Po tomto je výsledok na vrchu zásobníka a vaša funkcia ho má vrátiť. Ako vidíte, je to pomerne jednoduché.

- Ako zásobník použijete zoznam; pridávate pomocou `list.append(...)`, vyberáte pomocou `list.pop()`
- Reťazec sa na slová rozdelí pomocou `str.split`.

```
>>> "kobyła ma maly bok".split()
['kobyła', 'ma', 'maly', 'bok']
```

- Ak slovo nie je binárna operácia, môžete predpokladať, že to je reťazec reprezentujúci celé číslo. Na celé číslo ho prevediete pomocou `int(...)`.
- Nemusíte riešiť delenie nulou.
- Budete potrebovať Váš modul testovať. Toto môžete urobiť napríklad tak, že si napíšete oddelený skript, ktorý bude Váš modul importovať a vyskúša ho povedzme cez testovacie výpisy. Nedávajte testovacie výpisy priamo do modulu, pretože to znamená že sa budú spúšťať pri importe modulu, to je nežiadúce.