Praktický příklad – RPN kalkulátor

Abstraktní datové typy

V programování se často užívají dva abstraktní datové typy. Abstraktní proto, že nebývají v programovacích jazycích přímo implementovány. V následujícím textu je budeme interpretovat pomocí seznamu. Jsou to:

- **Fronta** – odpovídá postupu při spravedlivé obsluze, či způsobu skladování zboží, které podléhá zkáze. Metodou přístupu je metoda **FIFO** (*First In First Out*) – tedy „kdo dřív přijde, ten dřív mele“. Z fronty vychází jako první datová položka, která první do ní vstoupila. V jazyku Python pro přidání prvku na konec fronty použijeme metodu `append()`, pro odebrání prvního prvku pak metodu `pop()`, které předáme index `0` (ta tudíž odstraní a vrátí první prvek celého seznamu):

```
>>> fronta = ["Python", "Perl", "PHP"]
>>> fronta.append("Ruby")
>>> fronta.append("Lisp")
>>> fronta.pop(0)
'Python'
>>> fronta.pop(0)
'Perl'
>>> fronta
['PHP', 'Ruby', 'Lisp']
```

- **Zásobník** – odpovídá postupu při obsluze přerušení (bez priorit) či způsobu ukládání zboží, které nepodléhá zkáze (např. talíře, cihly apod.). Metodou přístupu je **LIFO** (*Last In First Out)* – tedy „kdo poslední přijde, ten první mele“. Pomocí metod `append()` a `pop()` lze ze seznamů vytvořit zásobníky. Poslední přidaný prvek bude odebrán jako první. Pro přidání prvku na vrchol zásobníku použijeme metodu `append()`, pro odebrání prvku metodu `pop()`. Metoda `pop()` bez dalších argumentů vrátí poslední prvek seznamu, čili prvek na vrcholu zásobníku:

```
>>> zasobnik = [3, 4, 5]
>>> zasobnik.append(6)
>>> zasobnik.append(7)
>>> zasobnik
[3, 4, 5, 6, 7]
>>> zasobnik.pop()
7
>>> zasobnik
[3, 4, 5, 6]
>>> zasobnik.pop()
6
>>> zasobnik.pop()
5
>>> zasobnik
[3, 4]
```

Reverzní polská notace

Obecně lze aritmetický výraz zapisovat notací:

- **Prefixovou** – operátor přechází oba operandy. Např. chceme sečíst dvě čísla – 5 a 8. V prefixové notaci to zapíšeme +(5,8).
- **Interfixovou** – operátor leží mezi dvěma operandy. Tentýž příklad zapíšeme 5 + 8.
- **Postfixovou** – operátor leží až za oběma operandy. Opět stejný příklad – (5,8)+.

V praxi se užívá interfixová notace v kombinaci s prefixovou (zejména unární mínus). **Postfixová notace** zvaná též **reverzní polská** se rozšířila u některých kalkulátorů, protože umožňuje zapisovat aritmetický výraz bez užití závorek.

Teoretická realizace kalkulátoru v RPN. Předpokládejme, že operandy se ukládají na zásobník. Kalkulátor ovládá operátory pro sčítání (+), odčítání (-), násobení (*) a dělení (/). Pokud kalkulátor načte některý z těchto operátorů provede příslušnou operaci tak, že jako první operand vezme ten, který je první pod vrcholem (označuje se jako *NOS – Next Of top of Stack*) a jako druhý ten, který je na vrcholu zásobníku (označuje se *TOS – Top Of Stack*). Označení NOS a TOS pochází z programovacího jazyka Forth. Je třeba si uvědomit, že zásobník se obvykle kreslí „vzhůru nohama“, takže TOS je úplně dole.

Příklad:

Chceme zapsat a vypočítat v RPN výraz: (5 *3 – 8) * (4 + 2) / (3 *7) =

Řešení:

5	**3**	*****	**8**	**-**	**4**	**2**	**+**	*****	**3**	**7**	*****	**/**	**=**
5	**5**	**15**	**15**	**7**	**7**	**7**	**7**	**42**	**42**	**42**	**42**	**2**	
	3		**8**		**4**	**4**	**6**		**3**	**3**	**21**		
						2				**7**			

Programy – RPNKalk a RPNKalk v Tk

RPN v Tk je rozšířen o další funkce a výsledkem je plnohodnotný simulátor kalkulátoru v RPN.

Literatura:

[1] Rubeš, J.: Nebojte se programovat, ComputerMedia, Bedihošť 2001