

Evaluarea Expresiilor Aritmetice

Ciuperceanu Vlad-Mihai
Grupa 151

1 Aspecte ale proiectării codului

Pentru a construi arborele binar care să ilustreze expresia dată (*expression tree*), vom începe prin a defini clasa *Node*, în care reținem un string *token*, în care punem operatorul operației curente. De asemenea, vom mai ține un atribut *value*, în care păstrăm valoarea expresiei, precum și doi pointeri *left* și *right*, pentru cei doi fii.

Apoi, în clasa *ExpressionTree*, vom construi arborele, ce are o rădăcină (*root*). Pentru inițializare, vom folosi o funcție privată clasei - *parseExpression*, ce are ca parametrii un string cu expresia și un indice transmis prin referință, ce arată unde anume ne aflăm în parsarea expresiei.

Funcția recursivă *parseExpression* va fi folosită pentru a construi arborele și pentru a obține valorile operațiilor. Aceasta va returna un nod cu informația obținută din subarborele său. Practic, vom construi recursiv nodurile arborelui, plecând dinspre rădăcină spre frunze.

La fiecare indice, verificăm mai întâi dacă am terminat expresia, caz în care ne oprim și returnăm un nod fără informație. Apoi, dacă încă mai avem de parsat din expresie, avem 2 cazuri: întâlnim un număr sau o să întâlnim un operator. Indicele din parametru ne va ajuta să avansăm prin expresie, parsând în continuare și trecând peste spații sau paranteze.

Dacă întâlnim un număr, extragem tot numărul și actualizăm informația din *value* a nodului creat, apoi îl returnăm. Dacă urmează un operator, actualizăm câmpul *token* al nodului curent, apoi îi vom crea cei doi fii, fiecare reținând unul dintre termenii operației, prin apelarea funcției *parseExpression* de la indicele la care ne aflăm. Acest lucru este corect, deoarece indicele este transmis prin referință, astfel că după prima apelare, indicele se va afla pe poziția corectă, de unde mai avem de procesat. Apoi, în funcție de operatorul citit, efectuăm operația citită cu informația obținută din cei doi fii.

La final, vom găsi rezultatul expresiei în atributul *value* al rădăcinii, pe care îl vom returna prin funcția *result*.

2 Exemple

În primul rând, trebuie menționat faptul că toate inputurile vor fi de forma celor de mai jos, adică vom avea spații doar între numerele din cadrul unei operații, în rest simbolurile vor fi unul lângă celălalt. Mai jos se află câteva exemple pe care programul a fost testat:

$(*(-31)(+25))$, pentru care rezultatul obținut este 14
 $(/7(*42))$, pentru care rezultatul obținut este 0.875
 $(*(+34)(+56))$, pentru care rezultatul obținut este 77
 $(/(-(*82)10)(+31))$, pentru care rezultatul obținut este 1.5
 $(+(*(-62)3)5)$, pentru care rezultatul obținut este 17