

## Progetto POO - Valutazione di una espressione aritmetica intera con le priorità degli operatori della matematica

### Premessa

Gli operandi sono interi senza segno. Gli operatori sono +, -, \*, %, ^ e valgono le usuali precedenze della matematica, cioè:  $\pi(^)>\pi(*,/,%)>\pi(+,-)$ . A parità di priorità, si assume l'associatività a sinistra. Eventualmente, si possono usare le parentesi ( e ) per alterare le priorità intrinseche: un'espressione in parentesi va sempre valutata prima.

### Algoritmo di valutazione

Si usano due stack: il primo è uno stack di operandi, il secondo è uno stack di caratteri operatori. Quando arriva un operando, lo si inserisce in cima allo stack di operandi. Quando arriva un operatore, sia esso `opc` (`op(eratore c(orrente))`), si procede come segue:

- A) se `opc` è più prioritario dell'operatore affiorante dallo stack di operatori o tale stack è vuoto, si inserisce `opc` in cima allo stack degli operatori.
- B) Se `opc` non è più prioritario rispetto alla cima dello stack operatori, si preleva l'operatore **op** al top dello stack operatori, quindi si prelevano due operandi `o2` (top) e `o1` (top-1) dallo stack operandi (in caso di eccezioni, l'espressione è malformata). Si esegue `o1 op o2`. Il risultato è inserito in cima allo stack operandi. Si continua ad eseguire il passo B) se `opc` risulta ancora non più prioritario dell'operatore affiorante la cima dello stack operatori. Dopo questo, o perché `opc` è più prioritario dell'operatore in cima allo stack operatori o perché tale stack è vuoto, si applica il caso A).

Quando l'espressione termina, se lo stack operatori non è vuoto, si estraggono uno alla volta gli operatori presenti e si applicano ai rispettivi due operandi prelevati dallo stack operandi, come spiegato al punto B), inserendo ogni volta il risultato in cima allo stack operandi.

Quando lo stack operatori è vuoto, allora lo stack operandi dovrebbe contenere un solo elemento che è il risultato dell'espressione. Ogni altra situazione (stack operandi vuoto o con più di un elemento) denota una situazione di espressione malformata.

### Gestione parentesi tonde

Quando si incontra una parentesi aperta '(' si invoca ricorsivamente la procedura di valutazione (diciamola `valutaEspressione()`). Quando si incontra una parentesi chiusa ')', si ritorna l'operando in cima allo stack operandi (un solo elemento o l'espressione è malformata).

Materialmente, il metodo `valutaEspressione()` potrebbe introdurre i due stack come variabili locali. Al termine di `valutaEspressione()`, il metodo ritorna l'unico elemento (o la sotto espressione è malformata) in cima allo stack operandi locale.

Il metodo `valutaEspressione()` potrebbe ricevere come parametro uno string tokenizer inizialmente aperto sulla stringa espressione ricevuta da input.

### Malformazioni

Alcune evidenti malformazioni (es. utilizzo di operatori non ammessi) possono essere immediatamente identificate mediante pattern matching della stringa espressione letta da input su un'espressione regolare che esprime la "corretta costituzione" (condizione necessaria) dell'espressione aritmetica.

### GUI di interazione

Il progetto dovrebbe includere la messa a punto di una GUI amichevole di interazione.