

### Calcolatrice RPN – versione 2

Modificare la prima versione, partendo dal presupposto che l'espressione possa contenere degli errori lessicali e sintattici: caratteri non ammessi e/o forma dell'espressione scorretta.

### Note sull'implementazione

Poiché è richiesta la verifica della correttezza dell'espressione, nell'implementazione del metodo di calcolo si possono prevedere due opzioni:

- 1 Il metodo restituisce il risultato come valore *out* e ritorna *true* se l'espressione è corretta, *false* altrimenti. (Vedi metodo `int.TryParse()`)
- 2 Il metodo solleva un'eccezione di tipo `FormatException` (Vedi metodo `int.Parse()`).

Nota bene: i due punti non sono alternativi, possono essere implementati contestualmente, come avviene nel tipo `int`.

### *Sollevare eccezioni*

Le eccezioni hanno la funzione di interrompere l'esecuzione di un metodo quando questo non è in grado di produrre il risultato o l'azione richiesta. L'eccezione, se non gestita dal codice che chiama il metodo, produce il *crash* del programma.

Per creare un'eccezione si usa la parola chiave `throw`. Ad esempio, per la creare l'eccezione `FormatException` si scrive

```
throw new FormatException("Espressione non valida");
```

Nota bene, nel creare l'eccezione si può specificare un messaggio che indichi il motivo dell'errore.