

Appello TAP del 18/01/2018

Scrivere nome, cognome e matricola sul foglio protocollo, indicando anche se avete nel piano di studi TAP da 6 CFU (quello attuale) o da 8 CFU (quello “vecchio”). Avete a disposizione due ore.

Esercizio 1 (10 punti)

Scrivere l’extension-method `CountUntil` che conta, in una sequenza di elementi di tipo `T`, quanti elementi soddisfano un certo predicato.

Il metodo dovrà prendere come parametri:

- “this” `seq`, la sequenza sorgente. Nota: la sequenza può essere infinita.
- `matchPredicate`, il predicato che indica se un elemento va contato o meno
- `exitCondition`, un predicato *opzionale* che indica se terminare il conteggio prima della fine della sequenza. Nota: `exitCondition` ha una priorità maggiore di `matchPredicate`, quindi se `exitCondition` restituisce `true` su un certo elemento, questo non deve essere conteggiato (e il metodo deve restituire il numero di elementi contati fino a quel momento).

Il metodo `CountUntil` deve sollevare l’eccezione `ArgumentNullException` se `seq` o `matchPredicate` sono `null`.

Esempi

Date s_1 la sequenza dei primi 10 numeri interi positivi, e s_2 la sequenza di tutti gli interi positivi ci aspettiamo che `CountUntil`:

1. non termini su s_2 quando `exitCondition` non è presente o restituisce `false` su argomenti positivi
2. restituisca 0 su `seq="cippalippa"` e `matchPredicate=Char.IsDigit`
3. restituisca 2 su `seq="sottocoppe di peltro"` e `matchPredicate="elemento uguale al carattere spazio"`
4. restituisca 5 su `seq=s1` e `matchPredicate="numero pari"`
5. restituisca 5 su `seq=s1`, `matchPredicate="numero pari"` ed `exitCondition="numero diverso da -42"`
6. restituisca 2 su `seq=s1` quando `matchPredicate="numero pari"` ed `exitCondition="numero maggiore di 4"`.
7. restituisca 2 su `seq=s2` quando `matchPredicate="numero pari"` ed `exitCondition="numero maggiore di 4"`.

Esercizio 2 (10 punti)

1. Implementare, usando NUnit ed eventualmente Moq, dei test che coprano gli esempi da (2) a (6) dell’esercizio precedente. Nota: il numero di test e quello degli esempi non deve necessariamente coincidere, purché i test coprano tutti i casi.
2. Sarebbe possibile coprire anche l’esempio (7) con un test? Se sì, darne un’implementazione, altrimenti motivare perché sarebbe impossibile.

Esercizio 3 (10 punti)

Implementare, sfruttando solo `CountUntil` (=nelle vostre implementazioni non è permesso invocare *nessun* altro metodo), i seguenti metodi:

- `static int CountPositive(int[] array)`, che restituisce quanti elementi di `array` sono (strettamente) maggiori di 0
- `static int StringLength(string s)`, che calcola (in modo “buffo”/strano) la lunghezza di `s`, *senza* usare la proprietà `Length`
- `static int FirstDigitPosition(String s)`, che restituisce la posizione della prima cifra contenuta in `s`, oppure la sua lunghezza se non sono presenti cifre. Qui vi potrebbe tornare utile il metodo `Char.IsDigit`, che potete *usare* (ma non invocare).