

Appello TAP del 21/01/2016

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)

Dato il tipo

```
public struct Diversity
{
    public int LowerCount { get; private set; }
    public int UpperCount { get; private set; }
    public int DigitCount { get; private set; }
    public Diversity(int lowerCount, int upperCount, int digitCount) {
        LowerCount = lowerCount; UpperCount = upperCount; DigitCount = digitCount; }
}
```

scrivere l’extension-method `DiversityCheck` che, presa una sequenza `alphanumerics` di stringhe alfanumeriche, restituisce la sequenza di oggetti di tipo `Diversity` contenenti ciascuno il numero di minuscole, maiuscole e cifre presenti in tali stringhe, sollevando opportune eccezioni nel caso un elemento di `alphanumerics` non sia una stringa alfanumerica. Per esempio, il seguente frammento di codice

```
foreach (var d in new [] { "Paperotto", "", "Paper8" }.DiversityCheck())
    Console.WriteLine("lower count = {0}; upper count = {1}; digit count = {2}",
        d.LowerCount, d.UpperCount, d.DigitCount);
```

stampa, nell’ordine,

```
lower count = 8; upper count = 1; digit count = 0
lower count = 0; upper count = 0; digit count = 0
lower count = 4; upper count = 1; digit count = 1
```

Il metodo dovrà prendere come parametro “this” `alphanumerics`, la sequenza sorgente. Nota: questa sequenza può anche essere infinita.

Il metodo deve sollevare l’eccezione...

- `ArgumentNullException` se `alphanumerics` o uno dei suoi elementi è `null`;
- `ArgumentOutOfRangeException` se uno degli elementi di `alphanumerics` contiene caratteri che non sono né lettere né cifre.

A seconda della soluzione che volete adottare, potreste trovare utili alcuni dei seguenti metodi statici:

```
Char.IsLetter(Char c)
Char.IsLower(Char c)
Char.IsUpper(Char c)
Char.IsDigit(Char c)
```

Esercizio 2 (3+3+4 = 10 punti)

- Elencare, descrivendoli a parole, i test significativi per il metodo `DiversityCheck`, dell’esercizio precedente.
- Implementare, usando NUnit ed eventualmente Moq, due test della lista precedente; uno che vada a testare un caso “buono” (ovvero, dove ci si aspetta che l’invocazione di `DiversityCheck` vada a buon fine) e uno che vada a testare un caso “cattivo” (ovvero, dove ci si aspetta che l’invocazione di `DiversityCheck` sollevi un’eccezione).
- Implementare, usando NUnit ed eventualmente Moq, un test in cui `DiversityCheck` è istanziato su una sequenza infinita e la chiamata non solleva eccezioni (diverso dai test implementati al punto precedente).

Esercizio 3 (10 punti)

Si implementi una `struct Length` per rappresentare lunghezze, sia in metri che in piedi (feet... per chi non si ricordasse un foot corrisponde a 304,8 millimetri).

La `struct Length` dovrà fornire:

- una proprietà booleana (con `get` e `set` pubblici) vera se l'elemento è espresso in metri (attenzione: a seconda di come implementate il tipo di dato, quando si modifica questa proprietà potrebbe essere necessario modificare anche campi/altre proprietà per continuare a rappresentare la stessa lunghezza);
- un costruttore che preso un `double length` e un booleano (opzionale) `isMeter` produce la lunghezza corrispondente in metri se `isMeter` è vero od omesso, altrimenti in feet (piedi);
- l'override del metodo `ToString()` che su un elemento che rappresenta, ad esempio, 137,42 metri restituisca la stringa `"137,42 m"` e su un elemento che rappresenta 137,42 piedi restituisca la stringa `"137,42 ft"`;
- la ridefinizione degli operatori `<` e `>`;
- la conversione implicita di un `double` alla lunghezza corrispondente, assumendo che il valore sia espresso in metri
- la ridefinizione dell'operatore `+` che prese due lunghezze produce una nuova lunghezza pari alla loro somma; in caso le due lunghezze siano espresse in unità diverse, il risultato sarà espresso in metri (quindi ad esempio sommare 2 metri e un piede darà come risultato 2,3048 metri), altrimenti nell'unità di misura comune ai parametri.