

Appello TAP del 13/07/2016

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)

Scrivere l’extension-method `SplitAndReverse` che, invocato su `sequence`, una sequenza di elementi di tipo `T`, e un ulteriore parametro, `size`, di tipo intero, prende gli elementi di `sequence` a blocchi di dimensione `size` (o quelli che rimangono, se ce ne sono meno di `size`) e li restituisce in ordine inverso all’interno di ciascun blocco.

Per esempio, il seguente frammento di codice

```
foreach (var d in Enumerable.Range(1, 57).SplitAndReverse(10))
    Console.Write("{0}, ", d);
Console.WriteLine();
```

stampa:

```
10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 30, 29, 28, 27, 26,
25, 24, 23, 22, 21, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 50, 49, 48, 47, 46, 45, 44,
43, 42, 41, 57, 56, 55, 54, 53, 52, 51
```

Il metodo dovrà prendere come parametro “this” `sequence`, la sequenza sorgente, e come altro parametro un intero `size`. Nota: la sequenza sorgente può anche essere infinita.

Il metodo deve sollevare l’eccezione...

- `ArgumentNullException` se `sequence` è `null`;
- `ArgumentOutOfRangeException` se `size` non è strettamente positivo.

Esercizio 2 (3+3+4+2 = 12 punti)

Implementare, usando NUnit ed eventualmente Moq, i seguenti test relativi al metodo `SplitAndReverse`, dell’esercizio precedente. (Suggerimento: potreste trovare utile l’extension method `Reverse` in `IEnumerable<T>`)

- Input della chiamata sotto test: `sequence` deve essere la sequenza 1, 2, ..., 57 e `size` deve essere 10.
Output atteso: la sequenza nell’esempio: 10, ..., 1, 20, ..., 11, 30, ..., 21, 40, ..., 31, 50, ..., 41, 57, ..., 51.
- Input della chiamata sotto test: `sequence` deve essere una sequenza di stringhe infinita e `size` deve essere 0.
Output atteso: deve essere sollevata un’eccezione di tipo `ArgumentOutOfRangeException`.
- Input della chiamata sotto test: `sequence` deve essere la sequenza infinita “pippo”, “pluto”, “topolino”, “paperino”, “pippo”, “pluto”, “topolino”, “paperino”, ... e `size` deve essere 4.

Il test deve essere parametrico con un parametro intero `howMany` e verificare che i primi `howMany` del risultato della chiamata siano i primi `howMany` elementi della sequenza “paperino”, “topolino”, “pluto”, “pippo”, “paperino”, “topolino”, “pluto”, “pippo”, “paperino”, “topolino”, “pluto”, “pippo”, ...

È possibile definire un test del metodo `SplitAndReverse` che abbia bisogno di usare dei mock? Se sì, descriverlo, precisando quale (o quali) mock si userebbero e come dovrebbero essere programmati. Se no, motivare la risposta.

Esercizio 3 (8 punti)

Applicare i principi della dependency injection per fare refactoring del codice seguente, eliminando le dipendenze indesiderate.

Se necessario, si possono introdurre ulteriori tipi e/o modificare le classi date.

Nel testo `/*...*/` rappresenta codice non interessante ai fini dell'esercizio

```
public class E{
    public string Name { get; }
    public E(string a){Name = a;}
    /*...*/
}

public class C{
    private E MyE { get; }
    public C(string s)
    {
        MyE = new E(s);
    }

    public E MM(string ss){/*...*/
        return new E(ss);
    }
    /*...*/
}
```