

Appello TAP del 20/01/2017

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 `SlidingWindowAverage` che traduce sequenze di elementi di tipo `double` in sequenze di elementi di tipo `double`. Il metodo `SlidingWindowAverage` invocato su `numbers` e un intero `howMany` restituisce la media di `howMany` elementi consecutivi di `numbers` partendo dal primo elemento, poi dal secondo e così via.

Per esempio, il seguente frammento di codice

```
foreach (var d in new[] {3.2, 3.2, 6.2, -3.4, 6.5}.SlidingWindowAverage(3))
    Console.WriteLine("{0} ", d);
Console.WriteLine();
```

stampa:

```
4.2 2 3.1
```

(perché $(3.2 + 3.2 + 6.2)/3 == 4.2$, $(3.2 + 6.2 - 3.4)/3 == 2$ e $(6.2 - 3.4 + 6.5)/3 == 3.1$)

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

Il metodo deve sollevare l’eccezione...

- `ArgumentNullException` se `numbers` è `null`;
- `ArgumentOutOfRangeException` se `howMany` non è strettamente positivo
- `ArgumentException` se in `numbers` non ci sono almeno `howMany` elementi

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

Implementare, usando NUnit ed eventualmente Moq, i seguenti test relativi al metodo `SlidingWindowAverage`, dell’esercizio precedente.

1. Test parametrico con parametri interi `windowSize` e `numbersLength`.

Input della chiamata sotto test: `numbers` deve essere la sequenza dei primi `numbersLength` interi (convertiti a `double`), partendo da 0 e `howMany` deve essere `windowSize`.

Output atteso: per calcolare il risultato si ricorda che la somma dei numeri $a, a + 1, \dots, b$ è $(a + b) * (b - a + 1) / 2$.

2. Input della chiamata sotto test: `numbers` deve essere la sequenza dei numeri interi (convertiti a `double`), partendo da 100 e `howMany` deve essere 0.

Output atteso: deve essere sollevata un’eccezione di tipo `ArgumentOutOfRangeException`.

3. Input della chiamata sotto test: `numbers` deve essere la sequenza dei numeri interi (convertiti a `double`), partendo da 27 e `howMany` deve essere 15.

Il test deve verificare la correttezza dei primi 1000 elementi del risultato (calcolati come suggerito al primo punto).

Esercizio 3 (10 punti)

Si consideri la seguente interfaccia per descrivere (in modo molto semplificato) beni di prima necessità da raccogliere e distribuire in caso di emergenza.

```
public interface IReliefGood {  
    string Name { get; }  
    string Description { get; }  
    // ...  
}
```

Definire una classe generica `OnlusStorage<T>` che rappresenta la gestione dei beni donati alle onlus. Per semplicità assumiamo che ciascuna onlus gestisca solo un tipo di bene, ad esempio solo vestiti o solo ripari di emergenza o solo strumenti educativi. Il tipo dei beni (che avrà ulteriori caratteristiche specifiche della tipologia di beni, ad esempio per i vestiti la taglia o la stagione in cui si indossa), sarà usato per instanziare `T` e dovrà realizzare l'interfaccia `IReliefGood`.

La classe `OnlusStorage<T>` dovrà fornire metodi (pubblici) per

- ricevere una donazione di un certo numero di esemplari di un bene da aggiungere alle scorte disponibili (ad esempio 100 paia di calze nuove, o 10 giacche invernali usate per una onlus che gestisce vestiti);
- verificare quanti esemplari di un certo bene sono disponibili;
- distribuire un certo numero di esemplari di un dato prodotto, limitatamente alle scorte presenti, aggiornando lo stato del magazzino (ad esempio se sono disponibili 10 giacche invernali, se ne possono distribuire 3 e restare con 7 ancora disponibili);
- stampare a video lo stato delle scorte.

Dovrà inoltre avere proprietà per alcuni dati, quali nome e indirizzo, nonché costruttori appropriati.

Documentate con brevi commenti come vengono segnalati all'utente gli errori e che cosa succede quando un'operazione non può andare a buon fine (ad esempio perché si cerca di distribuire più beni di quelli disponibili, o perché il numero che si è scelto di distribuire non è positivo).