

Esercizio (variazione sull'es. 14.3 di Horstmann)

Scrivete una classe Monomio che rappresenta il termine di un polinomio: la classe conterrà il coefficiente e l'esponente. Ad esempio $5x^{10}$ verrà rappresentato con coeff=5 e espon=10.

Implementate poi una classe Polinomio come una collezione (lista) di un numero arbitrario di Monomio.

Fornite il metodo normalizza() che fa sì che nel polinomio non vi siano più monomi con uguale esponente (es: $p(x) = 9x^7 + 3x^7 - 10$ dopo la normalizzazione deve diventare $p(x) = 12x^7 - 10$. Nota: non occorre che il polinomio abbia gli esponenti disposti in ordine, va bene anche $p(x) = 9x^7 + 3x^3 + 5x^5$.

Fornite metodi per sommare, sottrarre, moltiplicare e visualizzare polinomi, oltre a un costruttore che crea un polinomio a partire da un singolo termine.

Implementate il metodo `toString`: per il polinomio $9x^7 - 3x^3 + 5x^5$ la rappresentazione sarà $9 x^7 - 3 x^3 + 5 x^5$.

Il polinomio $p(x) = 5x^{10} + 9x^7 - x - 10$, ad esempio, potrebbe essere costruito in questo modo:

```
Polinomio p = new Polinomio(new Monomio(-10, 0));
p.add(new Polinomio (new Monomio (-1, 1)));
p.add(new Polinomio (new Monomio (9, 7)));
p.add(new Polinomio (new Monomio (5, 10)));
```

Successivamente calcolate $p(x) \times p(x)$, $p(x) + p(x)$, $p(x) - p(x)$.

Gestire correttamente il caso che il coefficiente sia 0, eliminando il corrispondente monomio.

Al termine, ripensate il progetto in termini di Map invece che di List.