

## Esercizio su iteratore per Poly

In riferimento all'esercizio relativo alla classe Poly (indifferente se denso o meno, se implementato con Vector o TreeSet), progettare e implementare quanto serve per fare in modo che il tipo di dato astratto Poly disponga anche di un'operazione che produce un iteratore sui termini del polinomio.

In particolare aggiungere al tipo Poly il seguente metodo:

```
// produce un iteratore sul polinomio this  
// che consente di "estrarre" i singoli termini di this  
// che hanno coefficiente diverso da 0  
// ad es. se it e' un tale iteratore su 2x^3+4x^6 allora  
// it.next() -> 2x^3  
// it.next() -> 4x^6  
public Iterator<Poly> termIterator()
```

e implementare tutte le classi e i metodi che servono per far funzionare l'iteratore.

NB “Estrarre” nel senso di produrre in uscita; non di togliere dal polinomio di partenza, che rimane un oggetto immutabile.

Riscrivere il metodo `differentiate` (riportato di seguito; per calcolare la derivata simbolica) in modo da usare l'iteratore e invocarlo su qualche esempio di polinomio e produrre su standard output una stampa dei termini prodotti via via dall'iteratore sul risultato di `differentiate`.

```
/**  
 * @param p not null  
 * @return the result of symbolic differentiation of a poly  
 * @Throws NullPointerException if p==null  
 */  
static public Poly differentiate (Poly p) {  
    Objects.requireNonNull(p);  
    Poly q = new Poly();  
    try {  
        for (int i=1; i < p.degree(); i++){  
            q = q.add(new Poly(p.coefficient(i)*i, i-1));  
        }  
    } catch (NegativeExponentException nee){  
        assert false : "this cannot happen";  
    }  
    return(q);  
}
```