

Esercitazione 8: classi astratte

Esercizio 1 (rappresentazione alternativa per le liste). Si consideri la seguente classe astratta che modifica la classe `List.java` vista a lezione con l'introduzione di alcuni nuovi metodi astratti:

```
public abstract class List
{
    public abstract boolean empty();
    public abstract int size();
    public abstract boolean contains(int x);
    public abstract List insert(int x);
    public abstract List append(List l);
    // NUOVI METODI
    public abstract int sum();
    public abstract int get(int i);
    public abstract List succ();
    public abstract List filter_le(int x);
    public abstract List filter_gt(int x);
    public abstract List intersect(List l);
}
```

per rappresentare liste (vuote o non vuote) di numeri interi ordinati. In ciascuna istanza di `List`, gli interi compaiono in ordine crescente, e non ci sono mai duplicati. Modificare le classi `Nil` e `Cons` viste a lezione per implementare i nuovi metodi di `List`. In particolare, dato un oggetto `p` di tipo `List`, deve essere possibile eseguire le seguenti operazioni:

- `p.sum()` ritorna la somma di tutti gli elementi della lista `p`.
- `p.get(i)` ritorna, se esiste, l'elemento che si trova all'indice `i` nella lista `p`. Al solito, il primo elemento si trova all'indice 0. Gestire in qualche modo il caso in cui `i` sia un indice non valido.
- `p.succ()` ritorna una *nuova* lista contenente tutti gli elementi di `p` incrementati di 1.
- `p.filter_le(x)` ritorna la sotto-lista di `p` che contiene tutti gli elementi di `p` minori o uguali a `x`. La lista ritornata è *nuova*.
- `p.filter_gt(x)` ritorna la sotto-lista di `p` che contiene tutti gli elementi di `p` maggiori di `x`. La lista ritornata è *nuova*.
- `p.intersect(l)` ritorna una nuova lista, intersezione di `p` e `l`.

Per verificare il corretto funzionamento dei metodi implementati è possibile eseguire il programma `TestList.java`, disponibile su Moodle e facente parte del materiale di questa esercitazione (tale programma è l'estensione dell'omonimo programma visto a lezione, con il codice per testare i nuovi metodi).

Nota

- Tutti i metodi da implementare si scrivono in poche righe (molti in una sola riga, spesso corta) di codice. Se vi accorgete di scrivere codice più complicato, molto probabilmente la soluzione pensata non è quella ideale.