Laboratorio di Algoritmi e Strutture Dati
2016/17
Mini Progetto, Codice: MP2

Docente: Luca Tesei

Scadenza: 13 Gennaio 2017 ore 23.59

1 Implementazione di lista sequenziale

Si definisca una classe pubblica `MyLinkedList<E>` che implementi l'interface `java.util.List<E>` e che utilizzi internamente una *lista concatenata doppia* per mantenere gli elementi. La lista concatenata doppia è una lista concatenata in cui ogni elemento ha un puntatore sia all'elemento successivo nella lista sia all'elemento precedente (a parte i casi speciali del precedente al primo elemento e del successivo all'ultimo elemento per cui si possono usare valori `null`).

La classe deve implementare tutti i metodi richiesti (si vedano le API su `https://docs.oracle.com/javase/7/docs/api/java/util/List.html`) tranne i metodi

- `boolean addAll(Collection<? extends E> c)`
- `boolean addAll(int index, Collection<? extends E> c)`
- `boolean containsAll(Collection<?> c)`
- `boolean removeAll(Collection<?> c)`
- `boolean retainAll(Collection<?> c)`
- `List<E> subList(int fromIndex, int toIndex)`
- `Object[] toArray()`
- `<T> T[] toArray(T[] a)`

per i quali si deve fornire una implementazione che semplicemente lancia una `UnsupportedOperationException`. La classe deve ridefinire i metodi `boolean equals(Object o)`, `int hashcode()` e `String toString()`.

In particolare, la classe deve fornire una implementazione (in una classe privata interna) di un `Iterator<E>` e di un `ListIterator<E>`. Per approfondire il comportamento di un oggetto `ListIterator<E>` si faccia riferimento alla documentazione delle API su `https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html`

2 Test della classe definita

Fornire una classe di test che permetta di controllare (con almeno un esempio) se tutti i metodi implementati funzionano.

In particolare, per quanto riguarda il test dell'oggetto di tipo `ListIterator<E>` che viene fornito dalla classe, è richiesto di implementare una classe di test `BracketInsideEater` che:

1. crea una `List<Character>` (in pratica una stringa, ma non della classe `String`) utilizzando la classe implementata `MyLinkedList<Character>`
2. richiede alla lista creata l'oggetto `ListIterator<Character>` corrispondente
3. utilizzando **solamente** i metodi del `ListIterator<Character>` deve implementare un algoritmo che depuri la lista dai caratteri che sono compresi tra due parentesi aperta e chiusa *al primo livello*[1], lasciando inalterati gli altri caratteri. Ad esempio, se la lista è
   `A X G F G ( D E ? ) J ; * D ) D D G ( D G 3 4 5 ) D ( D ( D ) ( ) G ) 5 6 S ( R T T`
   dopo la procedura dovranno rimanere solo i caratteri che sono al di fuori di una coppia di partentesi aperte e chiuse, cioè la lista:
   `A X G F G ( ) J ; * D ) D D G ( ) D ( ) ( ) G ) 5 6 S ( R T T`
   in quanto sono state trovate le seguenti coppie di parentesi:
   `A X G F G ( D E ? ) J ; * D ) D D G ( D G 3 4 5 ) D ( D ( D ) ( ) G ) 5 6 S ( R T T`

Per il punto (3) si può procedere come segue: si utilizza l'iteratore in avanti fino a quando si trova una parentesi aperta, dopodiché si continua a procedere in avanti fino a quando si trova una parentesi chiusa. A quel punto si torna indietro alla ricerca della parentesi aperta precedente eliminando lungo la strada i caratteri che sono nel mezzo. Ritrovata la parentesi aperta, si riparte in avanti rileggendo subito la parentesi chiusa e poi ricominciando il ciclo.

Modalità di Consegna

I file `.java` per le classi, senza indicazione di package (cioè appartenenti al package di default), devono essere caricati entro la data di scadenza in una cartella Google Drive dal nome

`ASDL1617MP2-CognomeStudente-NomeStudente`

che deve essere condivisa, in sola lettura, tramite l'account

`nome-studente.cognome-studente@studenti.unicam.it`

con gli account:

[1] Non si devono cioè considerare ulteriori parentesi aperte se una è già stata aperta e si deve considerare come chiusura la prima parentesi chiusa che si incontra dopo una che è stata aperta.

- `luca.tesei@unicam.it` (docente) e
- `matteo.micheletti@unicam.it` (tutor).

Per la scadenza, farà fede la data dei file su Google Drive.