1 Doppelt verkettete Liste

Erstellen Sie eine Klasse `de.hska.iwi.ads.solution.list.DoubleLinkedList`, die die Klasse `de.hska.iwi.ads.dictionary.AbstractDoubleLinkedList` als Basisklasse besitzt. Diese abstrakte Klasse implementiert den abstrakten Datentyp `java.util.List`. Sie basiert auf der abstrakten Klasse `java.util.AbstractMap`.

`AbstractDoubleLinkedList` besitzt direkt oder über Vererbung bereits:

- Eine verschachtelte Member-Klasse `ListLelement` mit Vorgänger- und Nachfolgerverweis sowie einem Objekt-Attribut `Entry entry`, in dem Schlüssel und Wert gespeichert werden können. Verwenden Sie `java.util.AbstractMap.SimpleEntry` für eine Implementierung von `java.util.Entry`.
- Einen Verweis `head` auf den Anfang der Liste.
- Ein `protected int size` Objekt-Attribut.
- Eine `size()`-Methode, die obigen Wert zurückgibt.
- Eine Iterator-Implementierung, mit dessen Hilfe die Methode `entrySet()` implementiert ist. Dies stellt sicher, dass eine Vielzahl von Methoden des abstrakten Datentyps `java.util.List` über die Implementierungen von `java.util.AbstractMap` funktionieren.

Es müssen lediglich noch genau zwei Methoden implementiert werden. Beachten Sie bitte auch die detailliertere Javadoc in der Klasse:

- `public V get(Object o)`: Gibt den Wert zurück, der unter dem angegebenen Schlüssel `o` in der Liste ist. Es wird null zurückgegeben, falls kein Wert mit diesem Schlüssel in der Liste existiert. Casten Sie den Parameter innerhalb Ihrer Implementierung auf den generischen Typ K. Die Warnung können Sie ignorieren oder mit vor der Anweisung vorangestelltem `@SuppressWarnings("unchecked")` unterdrücken. Andere Warnungen sollten nicht unterdrückt werden.
- `public V put(K key, V value)`: Fügt den Wert `value` am Anfang der doppelt verketteten Liste ein und gibt null zurück. Falls jedoch schon ein Wert mit dem angegebenen Schlüssel `key` in der Liste vorhanden ist, so wird beim zugehörigen ListElement-Objekt der Wert mit `value` überschrieben und der alte Wert zurückgegeben.

Ihre Implementierung darf keine Methoden, der Oberklassen aufrufen! Vor allem nicht den Iterator. Achten Sie auch darauf, bei Deklaration mit generischen Klassen die generischen Typen mit anzugeben.

Löschen soll nicht implementiert werden. Die zugehörige Methoden zum Entfernen von Werten werfen bei Aufruf eine `UnsupportedOperationException`. Es empfiehlt sich zur Suche nach einem ListElement-Objekt für einen Schlüssel eine private-Methode zu implementieren. Diese kann für die Suche in get und put aufgerufen werden. Auch wenn von der Rückverkettung kein Gebrauch gemacht wird, sollen die ListElement-Objekte korrekt verkettetet sein.

Testen Sie Ihre Implementierung mit JUnit. `de.hska.iwi.ads.dictionary.MapTest` kann als Basisklasse für Ihre Testklasse verwendet werden.