

## 5.1 Iterable

Die Klassen *Kette*, *ListNode* und das Interface *Liste* wurden aus Aufgabe 2 übernommen. *Kette* basiert hierbei auf „*LinkedList*“. Außerdem wurde die Klasse *MyIterator* hinzugefügt, in der die zum Iterieren benötigten Methoden *hasNext()* und *next()* definiert werden. Die Methode *hasNext()* überprüft ob in der Liste nach der aktuellen Position noch ein weiterer Wert vorhanden ist. Die Methode *next()* setzt die aktuelle Position auf das nächste Element und gibt dessen Wert zurück. Die Klasse *Kette* hat eine Methode *iterator()* erhalten, die einen dem Typ entsprechenden Iterator initialisiert und zurückgibt.

Der Iterator speichert intern immer die aktuelle Position und die Methode *next()* gibt die nächste zurück, d.h. bei der Initialisierung steht der Iterator noch auf keinem Element, erst durch Aufruf von *next()* wird das erste Element der Liste zurückgegeben.

Zum Test wird in der *main()* eine neue Kette angelegt, mit einer for-each Schleife durchlaufen und der Wert jedes Elementes ausgegeben.