

Judith Koch - Übung 5

5 Ergänze die Erbschaft

Entsprechend der Vorgaben aus der Vorlesung habe ich zuerst die Node Klasse um die Methode iterator() ergänzt:

```
public java.util.Iterator<T> iterator(){
    class NodeIterator<T> implements java.util.Iterator<T> {
        private Node<T> node;
        NodeIterator(Node<T> node){
            this.node = node;
        }
        public T next() {
            T result = node.getElement();
            node = node.getNext();
            return result;
        }
        public boolean hasNext(){
            return node != null;
        }
        public void remove(){
            throw new UnsupportedOperationException();
        }
    }
    return new NodeIterator<T>(this);
}
```

Anschliessend habe ich das Listen-Interface als Erweiterung des Iterable-Interface definiert. Dann hab ich erst mal geschaut, was überhaupt passiert. Da ich den Code jetzt zum Laufen bekommen habe, weiss ich das nicht mehr so ganz im Detail.

Daher hier meine Anmerkungen:

Die Methode insert() in der Klasse Chain hat nicht ganz korrekt funktioniert, tut es jetzt aber selbstverständlich.

Durch den Zusatz :

```
class Chain<T> implements List<T>
```

musste noch eine iterator() Funktion eingefügt werden:

```
@Override
public Iterator<T> iterator() {
    return first.iterator();
}
```

Wird nun ein Iterator aus einer Chain generiert, kann man über den Chain-Iterator iterieren.

```
Chain<String> chain = new Chain<String>();
chain.insert("Hallo", 0);
chain.insert("Welt", 1);
chain.insert("!", 2);
Iterator<String> iterator = chain.iterator();
while(iterator.hasNext()){
    System.out.println(iterator.next());
}
}
```

Unter gegebenen Umständen wird bei einer leeren Kette eine Ausnahme geworfen (try/catch in der main() Methode).