Aufgabe 2 Competitor Methoden [2 Punkte]

Die Competitor Klasse soll so angepasst werden, dass der *Namen* und *Jahrgang* als Kriterien für den Vergleich und den Hashcode verwendet werden. Implementieren Sie equals, compareTo und hashCode Methode so, dass sie dem *Object-Contract* gehorchen. Überprüfen Sie die korrekte Implementation mit der Test-Klasse.

Aufgabe 3 Eigene Hashtable [4 Punkte]

Ersetzen Sie die HashMap durch eine eigene Klasse MyHashtable. die das Map-Interface implementiert. Implementieren Sie hinzufügen (put), suchen (get) und löschen (remove). Überprüfen Sie die korrekte Implementation mit der Test-Klasse. Beim Konstruktor soll die maximale Anzahl Elemente angegeben werden, d.h. der Überlauf kann als Fehler betrachtet werden.

Hinweise:

- Verwenden Sie das MyHashtable Gerüst (siehe Vorlage):

```
public class MyHashtable<K,V> implements java.util.Map<K,V> {
    private Object[] keys; private Object[] values;
    public MyHashtable(int size) { // to be done }
    public V put(Object key, Object value) { // to be done }
    public V get(Object key) { // to be done }
    public V remove(Object key) { // to be done }
    ...
}
```

- In der Vorlage werfen alle Methoden die UnsupportedOperationException damit die Vorlage compiliert.

Aufgabe 4 Überlauf [2 Punkte]

Wenn die Hashtabelle überläuft, müssen die internen Tabellen erweitert und die Werte umkodiert werden. Implementieren Sie das korrekte Verhalten beim Überlauf.