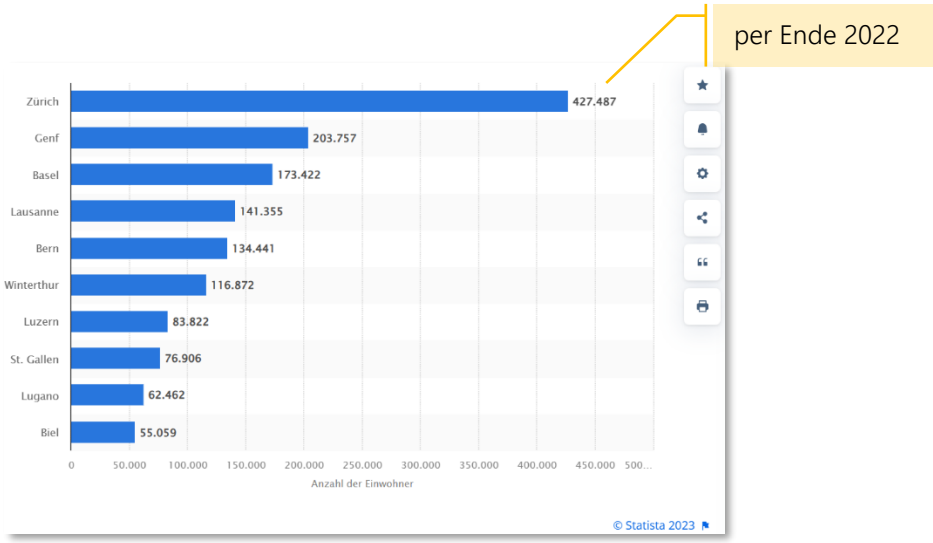
Hashing

Aufgabe 1: Manuelle Hashtabelle

Die zehn nach der Einwohnerzahl grössten Städte der Schweiz werden in einer HashTabelle der Grösse 13 abgelegt. Für die Kollisionsauflösung ist „Quadratic Probing" festgelegt. Die Städte und die für sie berechneten Hash-Codes sind:

5	Zürich
8	Genf
3	Basel
9	Lausanne
0	Bern
12	Winterthur
7	Luzern
7	St. Gallen
7	Lugano
5	Biel



Wie sieht die Hash-Tabelle aus, nachdem alle zehn Städte in alphabetischer Reihenfolge eingefügt wurden.

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Die Aufgabe muss nicht abgegeben werden.

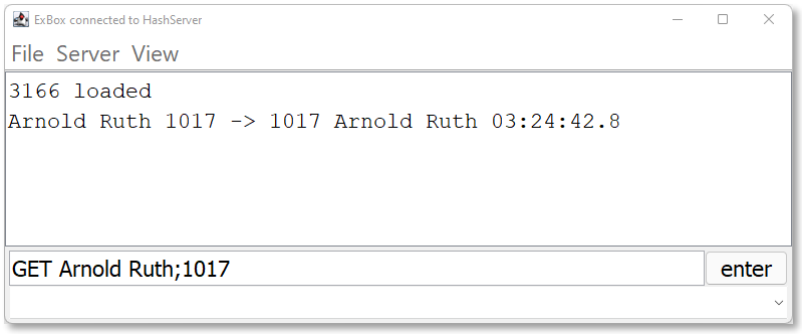
Aufgabe 2: Competitor-Methoden

WICHTIG: Verwenden Sie für dieses Praktikum die ExBox die dieser Übung beigelegt ist!

Die MyCompetitor Klasse soll so angepasst werden, dass der Namen und der Rang der Teilnehmer (siehe beigelegte CSV-Datei) als Kriterien für den Vergleich und den Hashcode verwendet werden. Implementieren Sie die equals-, compareTo- und hashCode-Methode so, dass sie dem Object-Contract gehorchen.

Aufgabe 3: Eine eigene Hashtable

Sie erhalten eine funktionierende HashServer-Klasse, die den CommandoExecutor implementiert und mittels der Sie die Teilnehmerliste in einer Hashtabelle (aktuell mit HaspMap von Java) verwalten können. Als Hashing-Schlüssel wird gemäss der Aufgabe 2 der Name plus Rang verwendet. Sie sollten mit der HashServer-Klasse in der Lage sein, die Daten zu laden (File → Open) und diese mittels des Befehls GET <Name>;<Rang> auszulesen.



Ersetzen Sie in dieser Aufgabe die HashMap durch eine eigene Klasse MyHashtable die das Map-Interface implementiert. Implementieren Sie hierzu hinzufügen (put), suchen (get) und löschen (remove). Überprüfen Sie die korrekte Implementation in der ExBox und mit der Test-KlasseADS9_3_test. Beim Konstruktor soll die maximale Anzahl Elemente so angegeben werden, dass kein Überlauf auftritt, d.h. der Überlauf kann als Fehler betrachtet werden.

Hinweis:

- Verwenden Sie das MyHashtable-Gerüst (siehe Vorlage).
- In der Vorlage werfen alle «überflüssigen» Methoden die UnsupportedOperationException damit die Vorlage compiliert. Diese müssen nicht implementiert werden.
- Implementieren Sie die Methoden mit "to be done" im Kommentar.
- Die Klasse MyCompetitor muss für den Upload in das File der Klasse MyHashtable integriert werden.

Aufgabe 4: Überlauf der Hashtabelle

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