

Arbeitsblatt: Hash Tables

Dieses Arbeitsblatt soll Ihnen den Umgang mit *Hashing* und *Hash Tabellen* anhand mehrerer kleinerer Aufgaben näherbringen.

1. Erarbeiten Sie im Script das Kapitel 6.7 – *Wahrscheinlichkeit von Kollisionen* selbständig.
2. Lesen Sie das Kapitel 6.8 – *Berechnung von hashCode*. Berechnen Sie die Hash-Werte in der ersten Tabelle. Stellen Sie sicher, dass die oberhalb der Tabelle beschriebene Problematik ersichtlich ist.
3. Füllen Sie im Kapitel 6.8 die zweite Tabelle aus. Verwenden Sie nun die abgebildete Methode `hashCode()` der Klasse `String` zur Berechnung des Hash-Werts und prüfen Sie, ob die Problematik der vorhergehenden Aufgabe nun gelöst ist.
4. Betrachten Sie die im Kapitel 6.8 abgebildete `hashCode()` Methode der Klasse `String` ein wenig genauer. Was fällt Ihnen auf?

5. Gegeben sei die folgende, einfache Klasse `MyInteger`:

```
public class MyInteger {  
    private int i;  
  
    public MyInteger(int i) {  
        this.i = i;  
    }  
}
```

Bei eigenen Datentypen müssen die Methoden `hashCode()` und `equals(Object o)` überschrieben werden. Beschreiben Sie, welche Fehler im Zusammenhang mit Hashing auftreten können, wenn eine der beiden Methoden nicht überschrieben oder falsch implementiert wird:

`equals(Object o)` nicht (korrekt) überschrieben:

`hashCode()` nicht (korrekt) überschrieben:

Machen Sie einen Vorschlag zur Implementation der beiden Methoden für die Klasse MyInteger:

<pre>@Override public boolean equals(Object obj) { } </pre>	<pre>@Override public int hashCode() { } </pre>
--------------------------------------------------------------	--------------------------------------------------

6. Die FHNW möchte von Eventoweb auf ein neues System umsteigen. Dazu müssen für alle Studierende die Gesamtzahl ECTS exportiert werden. Da sich für Zuordnungen Map-Strukturen gut eignen, schlagen Sie die Verwendung einer HashMap vor. (Auf einen Export ECTS / Note pro Modul verzichten wir einfachheitshalber einmal.)

Definieren Sie in Java eine Klasse Person, so dass diese korrekt funktionierend in einer HashMap als Key verwendet werden kann. Die Klasse Person soll die folgenden Variablen beinhalten, welche als Kombination eindeutig sind:

```
String firstname, lastname;
int birthyear, birthmonth, birthday, age;
```

7. Sie haben in der Klasse String gesehen, dass der Hashwert zwischengespeichert wird, so dass dieser nur einmal berechnet werden muss. Passen Sie Ihre Klasse Person an, so dass auch dort der Hashwert nur einmalig berechnet werden muss. Sehen Sie mögliche Schwierigkeiten, die bei zwischengespeicherten Hashwerten auftreten können? Falls ja, machen Sie einen Lösungsvorschlag.
8. Sie sehen folgende HashMap-Implementation vor sich. Die Methode indexOf(K key) wird verwendet, um die Position innerhalb des Arrays zu berechnen:

```
public class HashMap<K,V> implements Map<K,V> {

    Entry<K,V>[] table;

    private int indexOf(K key){
        return key.hashCode() % table.length;
    }

    static class Entry<K,V> implements Map.Entry<K,V> {
        final K key;
        V value;
        Entry<K,V> next;
        ...
    }
}
```

Welche Probleme können mit dieser Implementation auftreten? Machen Sie einen Korrekturvorschlag und testen sie diesen mit dem String „polygenelubricants“.

9. Unten finden Sie zwei Auszüge aus der Java HashMap (Version aus Java 1.7). Was können Sie anhand dieser Code-Stücke über Grösse des Arrays aussagen?

Die Methode indexFor wird mit dem von der Methode hashCode() eines Elements gelieferten Wertes als Parameter h aufgerufen, um den Index in das Array zu berechnen. Was genau ist der Sinn der Berechnung in der return-Anweisung und warum funktioniert das?

<pre>public HashMap (int initialCapacity) { int capacity = 1; while (capacity < initialCapacity) capacity <= 1; table = new Entry[capacity]; } </pre>	<pre>private int indexFor(int h) { return h & (table.length - 1); } </pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------