

Arbeitsblatt: Hash Tables

Aufgabe 1 TreeMap vs HashMap

Studieren Sie die Java-Dokumentation von TreeMap und HashMap und beantworten Sie die folgenden Fragen. Worin unterscheiden sie sich hauptsächlich? Warum ist eine HashMap für gewisse Elemente einfacher zu nutzen als eine TreeMap? Was ist der asymptotische Aufwand der Operation put() im best und im worst case?

Aufgabe 2 MyInteger

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:

```
@Override  
public boolean equals(Object obj) {  
  
  
  
  
  
  
  
  
  
}
```

```
@Override  
public int hashCode() {  
  
  
  
  
  
  
  
  
  
}
```

Aufgabe 3 Class Person I

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 einfachheits- halber 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 Kombina- tion eindeutig sind:

```
public class Person {
    public String firstname, lastname;
    public int birthyear, birthmonth, birthday;
    public int age;

    public Person(String first, String last, int year, int month, int day) {
        this.firstname = first;
        this.lastname = last;
        this.birthyear = year;
        this.birthmonth = month;
        this.birthday = day;
        age = (int)ChronoUnit.YEARS.between(LocalDate.of(year, month, day), Local-
Date.now());
    }

    @Override
    public int hashCode() {
        int result = 0;

        return result;
    }

    @Override
    public boolean equals(Object other) {
        if (other != null && getClass() == other.getClass()) {

        } else return false;
    }
}
```

Aufgabe 4 Class Person II

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.

Aufgabe 5 polygenelubricants

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;
        ...
    }
}
```

Für negative Werte von a ist $a \% b$ möglicherweise ebenfalls negativ¹. Negative Werte sind aber keine gültigen Array-Indizes. Die Methode `indexOf()` muss deshalb für beliebige `int`-Werte ein Ergebnis r im Bereich $0 \leq r < \text{table.length}$ berechnen.

Machen Sie einen Korrekturvorschlag und testen sie diesen mit dem String „polygenelubricants“.

¹ Es gilt für alle `int a, b`: $0 \leq a \Rightarrow 0 \leq a \% b < |b|$ und $0 \geq a \Rightarrow 0 \geq a \% b > -|b|$, wobei $|b|$ für den Betrag von b steht. Grund ist, dass dann immer gilt: $a = (a / b) * b + a \% b$ und $|(a / b) * b| \leq |a|$ – in Worten: Die Ganzzahldivision rundet gegen 0 und der Modulo-Operator berechnet den entsprechenden Divisionsrest.