Übung 9: Hashing

Abgabetermin: 28.05.2013

Name: _____ **Matrikelnummer:** _____

Gruppe: □ **G1** Di 10:15 □ **G2** Di 11:00 □ **G3** Di 12:45

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	24	□	Java-Programm, Testfälle und Ergebnisse	Java-Programm	□	

Aufgabe 1: Hashtabelle mit Überlauflisten (24 Punkte)

Implementieren Sie ein assoziatives Datenfeld (Map) für beliebige Objekte mit Hilfe einer Streuwerttabelle (Hashtable). Verwenden Sie dabei Strings als Schlüssel (Key), um die enthaltenen Elemente (Value) zu adressieren. Bei Kollisionen verwenden Sie Überlauflisten. Die Schnittstelle ist durch die abstrakte Klasse *Map* gegeben (für Methodenbeschreibungen siehe Java-Dokumentation in der Vorgabedatei).

```
package at.jku.ssw;

public abstract class Map {
  public abstract boolean containsKey(String key);
  public abstract Object get(String key);
  public abstract void put(String key, Object value);
  public abstract Object remove(String key);
}
```

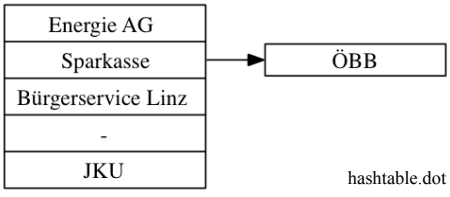
Implementieren Sie die Klasse *HashMap* im Paket *at.jku.students* wie folgt mit der Vorgabeklasse *Element* aus dem Paket *at.jku.ssw*.

```
class HashMap extends Map {
  Element[] table;
  HashMap(float fillFactor) { ... }
  String makeDot() { ... }
  String makeDot(float width) { ... }
  ...
}
class Element {
  final String key;
  final Object value;
  Element next;
  Element(String key, Object value) {...}
}
```

```
HashMap m = new HashMap(0.9f);
m.put("JKU", "0732/2468");
m.put("Bürgerservice Linz", "0732/7070");
m.put("ÖBB", "05-1717");
m.put("Energie AG", "0800-81 8000");
m.put("Sparkasse", "50-100-10100");
Out.open("hashtable.dot");
Out.println(m.makeDot());
Out.close();
Out.println(m.get("ÖBB"));
// Ausgabe: 05-1717
```

Implementierungshinweise:

a) Initialisieren Sie die Hashtabelle mit Länge 1 und vergrößern Sie automatisch, wenn der im Konstruktor definierte maximale Füllfaktor überschritten wird. Vergrößern Sie um den Faktor 2, setzen Sie die Länge auf die nächstgrößere Primzahl und führen Sie ein *rehashing* durch. Der maximale Füllfaktor muss größer 0.0 und kleiner oder gleich 3.0 (entspricht 300%) sein.



hashtable.dot

b) Berechnen Sie den Hashwert der Schlüssel mit der Hashfunktion *hashCode*() in der Klasse *String*.

c) Die Methode *PrimeNumbers.nextPrime(int number)* aus der Vorgabe liefert zu einer gegebenen Zahl die nächstgrößere Primzahl.

d) Definieren Sie für alle Klassen, Methoden und Felder die geeignete Sichtbarkeit (private, protected, package, public).

e) Verwenden Sie die Methoden *DotMaker.makeDotForHashtable(Element[] table)* und *DotMaker.makeDotForHashtable(Element[] table, float width)*, um GraphViz-Bilder Ihrer Streuwerttabelle zu erzeugen.

Abzugeben ist: Java-Programm, Testergebnisse