

Hausaufgaben 3

08./09.04.2019

Abgabe der Lösung am 14.04.2019

Aufgabe 1

Schreiben Sie eine Klasse `GamesHash`, die eine Liste von Spielnamen in einer Hashtabelle speichert. Folgende Methoden sollen implementiert werden:

```
private int hashIndex(String game)           // Berechnet h0
private int hashIncrement(String game)       // Berechnet Inkrement
public void add(String game)                 // Fügt Spiel zu Liste hinzu
public boolean contains(String game)         // Ist Spiel in der Liste?
```

Die Hashtabelle soll eine feste Anzahl von 31 Einträgen haben. Es wird das Verfahren des *doppelten Hashings* angewandt (siehe Vorlesung). Doppeltes Hashing mildert das Häufungsproblem, indem es verschiedene Elemente mit unterschiedlichen Schrittweiten wandern lässt.

Zur Bestimmung der notwendigen Hash-Werte sollten Sie zunächst

```
game.hashCode()
```

aufrufen. Die Methode brauchen Sie nicht zu implementieren, denn `hashCode` liefert Ihnen einen Hashcode für das String-Objekt. Die letzten 2 Dezimalziffern aus dem Ergebnis stellen dann h_0 dar (den ersten Hash-Index). Das Inkrement (für das doppelte Hashing) wird aus der drittletzten Ziffer gebildet. Hierbei müssen Sie noch den Wert 1 addieren, damit kein Inkrement 0 entstehen kann.

Beispiel: Hash-Wert: 123456789876 → $h_0=76, i=8+1=9$
 Hash-Wert: -345543210 → $h_0=10, i=2+1=3$
 Hash-Wert: 15 → $h_0=15, i=0+1=1$

Wenn ein Wert nicht mehr in die Tabelle eingefügt werden kann, wird eine Exception geworfen.

Testen Sie Ihre Hash-Klasse:

- Erzeugen Sie ein `GamesHash`-Objekt
- Lesen Sie die Datei „games20.txt“ ein (liegt auf der Veranstaltungsseite) und speichern Sie alle Spielnamen in dem `GamesHash`-Objekt. Es handelt sich um eine TOP20-Liste der meistverkauften Video-Spiele 2017 (UK), durch Metro UK zur Verfügung gestellt. Hinweis: Die Elemente in der Datei sind durch Tabulatorzeichen („\t“) getrennt.
- Überprüfen Sie nun die Testnamen:
 - "Nioh" ⇒ ok
 - "Cuphead" ⇒ ok
 - "WOW" ⇒ not ok