

Prof. Dr. Bernhard Seeger
Jochen van den Bercken

Übungen zur Vorlesung
Implementierung von Datenbanksystemen

Sie erhalten in der Übung vom 9.6. nähere Informationen zu den Klassen der XXL-Bibliothek!
Abgabe am 15. 6. in der **VORLESUNG!**

Aufgabe 5.1: Implementierung von linearem Hashing (17)

Implementieren Sie das JDK-Interface *java.util.Map* mit Hilfe linearen Hashings. Nutzen Sie dazu die abstrakte Klasse *java.util.AbstractMap*, und überschreiben Sie deren Methoden für *containsKey*, *put*, *get*, *remove*, *clear* sowie *entrySet*. Ein via *entrySet* gelieferter *Iterator* muß die Operation *remove* nicht unterstützen. Schreiben Sie einen Konstruktor *LinearHashMap* (*Function newBag*, *double alphaExpand*, *double alphaShrink*). Die einzelnen Buckets sollen als *xxl.collections.Bag* modelliert werden; wann immer Sie einen neuen *Bag* benötigen, können Sie die parameterlose Funktion *newBag* rufen. Die beiden restlichen Parameter geben den relativen Füllgrad an, ab dem eine Erweiterung/Verkleinerung der Tabelle stattfinden soll.

Aufgabe 5.2: Lineares Hashing (4 Punkte)

- a) Gegeben sei eine Datei mit 3 Buckets (0 bis 2) mit einer Kapazität von je 3 Sätzen. Nach der Initialisierung seien alle Buckets leer. Für die Folge von Hashfunktionen gelten folgende Funktionen:

$$h_i(K) = K \pmod{3 \cdot 2^i} \quad i \in \{ 1, \dots, n \}$$

Die Sätze mit den Schlüssel 2, 7, 11, 20, 1, 8, 5, 14, 3, 15, 13, 19, 6, 18, 21 und 23 werden in dieser Reihenfolge eingefügt. Der maximale Belegungsfaktor sei 0,7. Zeichnen Sie nach jeweils vier Einfügungen die Hashtabelle.

Aufgabe 5.3: Erweiterbares Hashing (4 Punkte)

- a) Die Sätze mit den Schlüssel K10, J84, K35, A12, E77, K12, B22, K08 sollen in der angegebenen Reihenfolge mit dem Verfahren des Erweiterbaren Hashing in Buckets abgespeichert werden, die jeweils 2 Sätze aufnehmen können. Den Pseudoschlüssel des jeweiligen Satzes erhält man durch Verknüpfung der in ASCII codierten Zeichen mittels XOR. Zeichnen Sie die Belegung der Buckets und das Directory nach jeder zweiten Einfügung.

Aufgabe 5.4: Grid-File (5 Punkte)

Machen Sie sich die dynamischen Reorganisationen im GRID-File bei Einfügungen und Löschungen anhand des folgenden Beispiels klar. Ein Bucket kann jeweils 3 Sätze aufnehmen, das GRID-File ist anfangs leer. Von der Relation AUTO (KFZ-NR, MARKE, FARBE) sollen die beiden Attribute MARKE und FARBE mit der GRID-File-Technik abgespeichert werden.

Zeichnen Sie alle wesentlichen Strukturen (Directory, Suchraum, Buckets) nach jeder zweiten der unten angegebenen Einfügungen.

Löschen Sie danach die ersten 4 Einträge und zeichnen Sie die obigen Strukturen auf.

	<u>KFZ-NR.</u>	MARKE,	FARBE
a)	KL-PP 1	OPEL	GELB
b)	PS-A17	FORD	BLAU
c)	KL-CX 33	OPEL	BLAU
d)	KIB-AM 13	BMW	ROT
e)	SB-F16	AUDI	GRÜN
f)	KL-DZ 12	ALFA	SCHWARZ
g)	ZW-AL 43	VW	WEISS
h)	HOM-C 1	FIAT	VIOLETT
i)	SLS-AF 47	AUDI	AZUR
j)	KL-DM 31	ALFA	BEIGE
k)	VW-FS 40	SAAB	BRAUN