

Title: Seidl: EOE1 (09.02.2012)

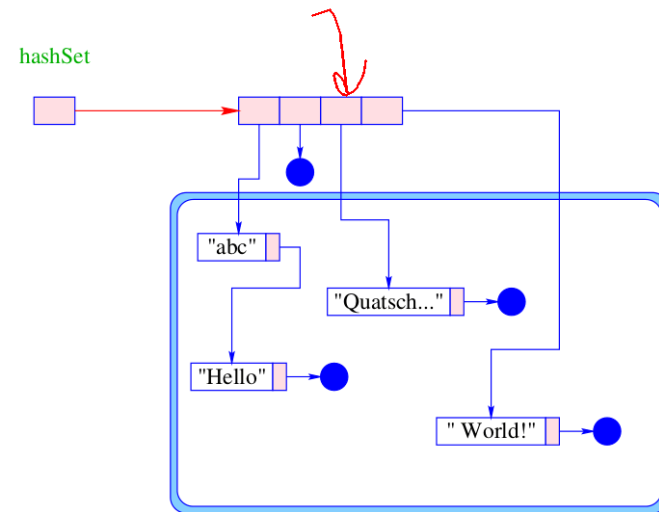
Date: Thu Feb 09 12:30:56 CET 2012

Duration: 48:45 min

Pages: 9

Auffinden der richtigen Liste:

- Benutze eine (leicht zu berechnende) Funktion `hash: String -> int;`
- Eine solche Funktion heißt **Hash-Funktion**.
- Eine Hash-Funktion ist gut, wenn sie die Wörter (einigermaßen) gleichmäßig verteilt.
- Hat das Feld `hashSet` die Größe m , und gibt es n Wörter im Pool, dann müssen pro Aufruf von `intern()`; nur Listen einer Länge ca. n/m durchsucht werden !!!



Sei s das Wort $s_0s_1 \dots s_{k-1}$.

Beispiele für Hash-Funktionen:

- $h_0(s) = s_0 + s_{k-1};$
- $h_1(s) = s_0 + s_1 + \dots + s_{k-1};$
- $h_2(s) = (\dots((s_0 * p) + s_1) * p + \dots) * p + s_{k-1}$ für eine krumme Zahl p .

(Die `String`-Objekt-Methode `hashCode()` entspricht der Funktion h_2 mit $p = 31$.)

String	h_0	h_1	$h_2 (p = 7)$
alloc	196	523	276109
add	197	297	5553
and	197	307	5623
const	215	551	282083
div	218	323	5753
eq	214	214	820
fjump	214	546	287868
false	203	523	284371
halt	220	425	41297
jump	218	444	42966
less	223	439	42913
leq	221	322	6112
...

String	h_0	h_1	h_2
...
load	208	416	43262
mod	209	320	6218
mul	217	334	6268
neq	223	324	6210
neg	213	314	6200
not	226	337	6283
or	225	225	891
read	214	412	44830
store	216	557	322241
sub	213	330	6552
true	217	448	46294
write	220	555	330879

439

Mögliche Implementierung von intern():

```
public class String {
    private static int n = 1024;
    private static List<String>[] hashSet = new List<String>[n];
    public String intern() {
        int i = (Math.abs(hashCode())%n);
        for (List<String> t=hashSet[i]; t!=null; t=t.next)
            if (equals(t.info)) return t.info;
        hashSet[i] = new List<String>(this, hashSet[i]);
        return this;
    } // end of intern()
    ...
} // end of class String
```

440

Sei s das Wort $s_0s_1 \dots s_{k-1}$.

Beispiele für Hash-Funktionen:

- $h_0(s) = s_0 + s_{k-1}$;
- $h_1(s) = s_0 + s_1 + \dots + s_{k-1}$;
- $h_2(s) = (\dots((s_0 * p) + s_1) * p + \dots) * p + s_{k-1}$ für eine krumme Zahl p .

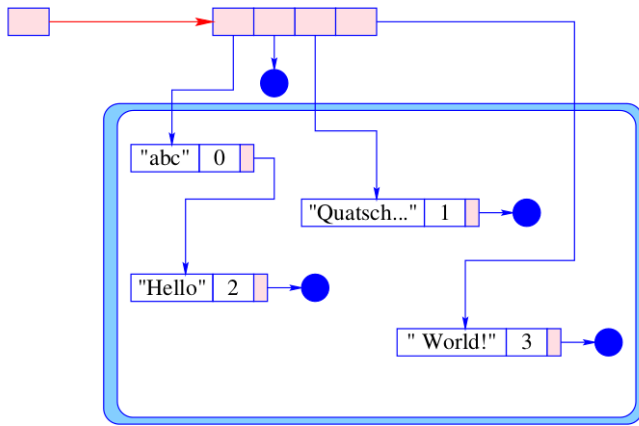
(Die String-Objekt-Methode `hashCode()` entspricht der Funktion h_2 mit $p = 31$.)

438

- Die Methode `hashCode()` existiert für sämtliche Objekte.
- Folglich können wir (wenn wir Lust haben) ähnliche Pools auch für andere Klassen implementieren.
- **Vorsicht!** In den Pool eingetragene Objekte können vom Garbage-Collector nicht eingesammelt werden ...
- Statt nur nachzusehen, ob ein Wort `str` (bzw. ein Objekt `obj`) im Pool enthalten ist, könnten wir im Pool auch noch einen Wert hinterlegen
 \implies Implementierung von beliebigen Funktionen `String` \rightarrow `type` (bzw. `Object` \rightarrow `type`)

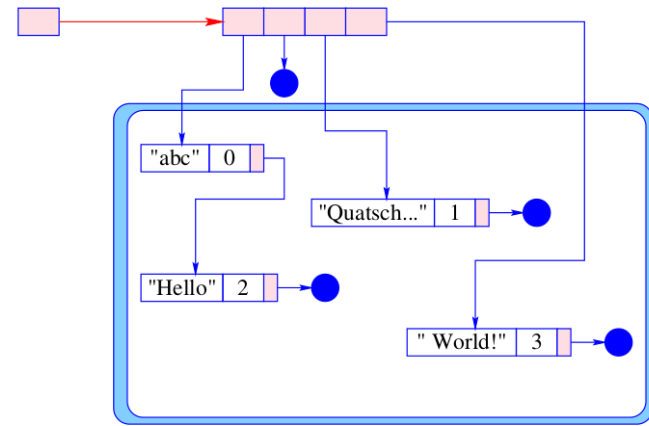
441

hashTable



442

hashTable



442