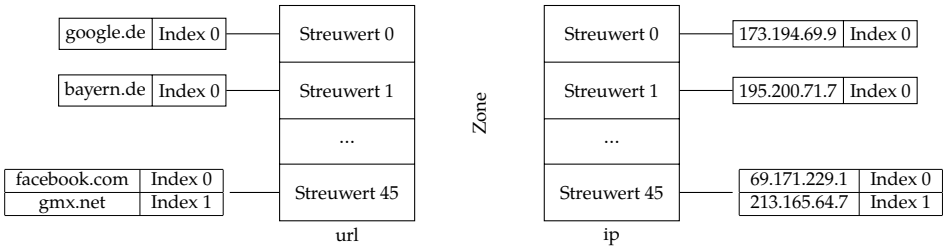
Aufgabe 6 Streutabellen (Hash-Tables)

Um die URL (zum Beispiel google.de) und die zugehörige IP des Servers (hier 173.194.69.9) zu verwalten, werden Streutabellen verwendet, die eine bestimmte Zone von Adressen abbilden. Die Streutabellen werden als zwei dynamische Arrays (in Java: ArrayLists) realisiert. Kollisionen innerhalb einer Zone werden ebenfalls in dynamischen Arrays verwaltet.



Um zu einer URL die IP zu finden, berechnet man zunächst mittels der Funktion `hash()` den entsprechenden Streuwert, entnimmt dann den Index der Tabelle URL und sucht schließlich an entsprechender Stelle in der Tabelle IP die IP-Adresse.

(a) Erläutern Sie am vorgestellten Beispiel, wie ein Hash-Verfahren zum Speichern großer Datenmengen prinzipiell funktioniert und welche Voraussetzungen und Bedingungen daran geknüpft sind.

(b) Nun implementieren Sie Teile dieser IP- und URL-Verwaltung in einer objektorientierten Sprache Ihrer Wahl. Verwenden Sie dabei die folgende Klasse (die Vorgaben sind in der Sprache Java gehalten):

```
1  class Zone {
2  private ArrayList<ArrayList<String>> urlList =
3    new ArrayList<ArrayList<String>>();
4  private ArrayList<ArrayList<String>> ipList =
5    new ArrayList<ArrayList<String>>();
6  public int hash(String url) {/* calculates hash-value h, >=0 */}
7  }
```

(i) Prüfen Sie in einer Methode `boolean exists(int h)` der Klasse `zone`, ob bereits mindestens ein Eintrag für einen gegebenen Streuwert vorhanden ist. Falls `h` größer ist als die derzeitige Größe der Streutabelle, existiert der Eintrag nicht.

```
1  package org.bschlangaul.examen.examen_66115_2013_03;
2
3  import java.util.ArrayList;
4
5  class Zone {
6    private ArrayList<ArrayList<String>> urlList = new
     ↪  ArrayList<ArrayList<String>>();
7    private ArrayList<ArrayList<String>> ipList = new
     ↪  ArrayList<ArrayList<String>>();
8
9    public int hash(String url) {
```

```
10      return 1;
11      /* calculates hash-value h, >=0 */
12
13    }
14
15    /**
16     * Prüfe, ob bereits mindestens ein Eintrag für einen gegebenen
      ↪  Streuwert
17     * vorhanden ist. Falls h größer ist als die derzeitige Größe
      ↪  der Streutabelle,
18     * existiert der Eintrag nicht.
19     *
20     * @param h
21     * @return
22     */
23    boolean exists(int h) {
24      return true;
25    }
26
27    /**
28     * Berechne den Index einer URL in der Kollisionsliste. Ist die
      ↪  URL in der
29     * Kollisionsliste nicht vorhanden, soll -1 zurückgeliefert
      ↪  werden.
30     */
31    int getIndex(String url, ArrayList<String> urlList) {
32      return 0;
33
34    }
35
36    /**
37     * Gib in der Streutabelle die IP-Adresse zurück. Wird eine
      ↪  nicht vorhandene
38     * Adresse abgerufen, wird eine Fehlermeldung zurückgegeben.
39     *
40     * @param url
41     * @return
42     */
43    String lookup(String url) {
44      return "lol";
45    }
46
47  }
```

(ii) Die Methode `int getIndex (string url, ArrayList<String> urlList)` soll den Index einer URL in der Kollisionsliste berechnen. Ist die URL in der Kollisionsliste nicht vorhanden, soll `-1` zurückgeliefert werden.

```
1   package org.bschlangaul.examen.examen_66115_2013_03;
2
3   import java.util.ArrayList;
4
5   class Zone {
6     private ArrayList<ArrayList<String>> urlList = new
      ↪  ArrayList<ArrayList<String>>();
7     private ArrayList<ArrayList<String>> ipList = new
      ↪  ArrayList<ArrayList<String>>();
8
9     public int hash(String url) {
10      return 1;
```

```
11      /* calculates hash-value h, >=0 */
12
13    }
14
15    /**
16     * Prüfe, ob bereits mindestens ein Eintrag für einen gegebenen
      ↪  Streuwert
17     * vorhanden ist. Falls h größer ist als die derzeitige Größe
      ↪  der Streutabelle,
18     * existiert der Eintrag nicht.
19     *
20     * @param h
21     * @return
22     */
23    boolean exists(int h) {
24      return true;
25    }
26
27    /**
28     * Berechne den Index einer URL in der Kollisionsliste. Ist die
      ↪  URL in der
29     * Kollisionsliste nicht vorhanden, soll -1 zurückgeliefert
      ↪  werden.
30     */
31    int getIndex(String url, ArrayList<String> urlList) {
32      return 0;
33
34    }
35
36    /**
37     * Gib in der Streutabelle die IP-Adresse zurück. Wird eine
      ↪  nicht vorhandene
38     * Adresse abgerufen, wird eine Fehlermeldung zurückgegeben.
39     *
40     * @param url
41     * @return
42     */
43    String lookup(String url) {
44      return "lol";
45    }
46
47  }
```

(iii) Ergänzen Sie die Klasse Zone um eine Methode `String lookup (String url)`, die in der Streutabelle die IP-Adresse zur `url` zurückgibt. Wird eine nicht vorhandene Adresse abgerufen, wird eine Fehlermeldung zurückgegeben.