package testjava;

import java.util.HashMap;

import java.util.Random;

/\*

\* URL Shortener

\*/

public class URLShortener {

// storage for generated keys

private HashMap<String, String> keyMap; // key-url map

private HashMap<String, String> valueMap;// url-key map to quickly check

// whether an url is

// already entered in our system

private String domain; // Use this attribute to generate urls for a custom

// domain name defaults to http://fkt.in

private char myChars[]; // This array is used for character to number

// mapping

private Random myRand; // Random object used to generate random integers

private int keyLength; // the key length in URL defaults to 8

// Default Constructor

URLShortener() {

keyMap = new HashMap<String, String>();

valueMap = new HashMap<String, String>();

myRand = new Random();

keyLength = 8;

myChars = new char[62];

for (int i = 0; i < 62; i++) {

int j = 0;

if (i < 10) {

j = i + 48;

} else if (i > 9 && i <= 35) {

j = i + 55;

} else {

j = i + 61;

}

myChars[i] = (char) j;

}

domain = "http://fkt.in";

}

// Constructor which enables you to define tiny URL key length and base URL

// name

URLShortener(int length, String newDomain) {

this();

this.keyLength = length;

if (!newDomain.isEmpty()) {

newDomain = sanitizeURL(newDomain);

domain = newDomain;

}

}

// shortenURL

// the public method which can be called to shorten a given URL

public String shortenURL(String longURL) {

String shortURL = "";

if (validateURL(longURL)) {

longURL = sanitizeURL(longURL);

if (valueMap.containsKey(longURL)) {

shortURL = domain + "/" + valueMap.get(longURL);

} else {

shortURL = domain + "/" + getKey(longURL);

}

}

// add http part

return shortURL;

}

// expandURL

// public method which returns back the original URL given the shortened url

public String expandURL(String shortURL) {

String longURL = "";

String key = shortURL.substring(domain.length() + 1);

longURL = keyMap.get(key);

return longURL;

}

// Validate URL

// not implemented, but should be implemented to check whether the given URL

// is valid or not

boolean validateURL(String url) {

return true;

}

// sanitizeURL

// This method should take care various issues with a valid url

// e.g. www.google.com,www.google.com/, http://www.google.com,

// http://www.google.com/

// all the above URL should point to same shortened URL

// There could be several other cases like these.

String sanitizeURL(String url) {

if (url.substring(0, 7).equals("http://"))

url = url.substring(7);

if (url.substring(0, 8).equals("https://"))

url = url.substring(8);

if (url.charAt(url.length() - 1) == '/')

url = url.substring(0, url.length() - 1);

return url;

}

/\*

\* Get Key method

\*/

private String getKey(String longURL) {

String key;

key = generateKey();

keyMap.put(key, longURL);

valueMap.put(longURL, key);

return key;

}

// generateKey

private String generateKey() {

String key = "";

boolean flag = true;

while (flag) {

key = "";

for (int i = 0; i <= keyLength; i++) {

key += myChars[myRand.nextInt(62)];

}

// System.out.println("Iteration: "+ counter + "Key: "+ key);

if (!keyMap.containsKey(key)) {

flag = false;

}

}

return key;

}

// test the code

public static void main(String args[]) {

URLShortener u = new URLShortener(5, "www.tinyurl.com/");

String urls[] = { "www.google.com/", "www.google.com",

"http://www.yahoo.com", "www.yahoo.com/", "www.amazon.com",

"www.amazon.com/page1.php", "www.amazon.com/page2.php",

"www.flipkart.in", "www.rediff.com", "www.techmeme.com",

"www.techcrunch.com", "www.lifehacker.com", "www.icicibank.com" };

for (int i = 0; i < urls.length; i++) {

System.out.println("URL:" + urls[i] + "\tTiny: "

+ u.shortenURL(urls[i]) + "\tExpanded: "

+ u.expandURL(u.shortenURL(urls[i])));

}

}

}