Technische Universität Berlin **SS99**
Fachbereich Mathematik
Sekr. MA 6-1 / Frau Marcus

Dr. Michael Joswig

Olaf Jahn
Matthias Müller, André Opitz

6. Programmieraufgabe

Abnahme: spätestens 06.07.99 zur Rechnervorrangzeit

Wie in der großen Übung erklärt, kann jede Website eine Datei `/robots.txt` anlegen, in der steht, welche Suchmaschine sich von welchem Teil des Dateibaumes bitte fernzuhalten hat. Eure Suchmaschine soll diese Angaben natürlich auch berücksichtigen. Dazu benötigt Ihr eine Klasse, die den Inhalt von `robots.txt` verstehen kann und so abspeichert, daß sich später sehr schnell feststellen läßt, ob eine gewünschte URL erlaubt oder verboten ist. Das Format von `robots.txt` ist in

`http://info.webcrawler.com/mak/projects/robots/norobots.html`

erklärt. Eure Klasse sollte etwa die folgende Struktur haben

```
class DisallowedPages
{
 /**
  * Build data structure of forbidden (partial) filenames from
  * robots.txt.
  *
  * @param userAgent when interpreting robots.txt assume that we are this agent.
  * @param robots_txt the content of robots.txt as retrieved from a web server
  */
 public DisallowedPages(String robots_txt, String userAgent) { ... }

 /**
  * Check if the given (absolute!) URL is allowed with respect to this
  * robots.txt.
  * Note: This method has to strip the hostname from the URL to compare
  * it with the stored (partial) paths.
  *
  * @param url page URL.
  * @return true if and only if the URL is allowed to be retrieved.
  */
 public boolean isAllowed(String url) { ... }
}
```

Dabei speichert Ihr natürlich nicht das komplette `robots.txt` als String(s) und sucht bei jeder Anfrage darin. Das ist uns zu aufwendig, würde es doch $O(n)$ Zeit dauern mit n als Summe aller Pfadlängen. Überlegt Euch eine Datenstruktur, die jede Anfrage mit `isAllowed` in $O(\ell_{\max})$ Zeit beantwortet, wobei $\ell_{\max}$ die Länge des längsten in `robots.txt` vorkommenden Pfades ist. Ihr dürft dabei voraussetzen, daß alle Pfade mit `/` anfangen und nur die ASCII-Zeichen mit den Codes 32 bis 126 enthalten dürfen.

Schreibt für diese Klasse ein kleines `main`, das eine angegebene Datei von der Festplatte als `robots.txt` einliest und dann bei per Tastatur eingegebene URLs (von denen wir ungeprüft annehmen, daß sie sich auf den gleichen Server beziehen) ausgibt, ob sie erlaubt sind oder nicht.

Tip: Beim Auswerten des `robots.txt`, das ja als ein mehrzeiliger String vorliegt, teilt Ihr es am besten mit einem `StringTokenizer` in einzelne Zeilen (wie geht das?), die dann einzeln betrachtet werden. Beachtet, daß vor `#` gar kein Leerzeichen stehen muß. Die Methoden der Klasse `String` sind Eure Freunde.