

Übung 10: Stringsuche

Abgabetermin: 30.05.2017

Name:

Matrikelnummer:

Gruppe:

G1 Di 10:15-11:00

G2 Di 11:00-11:45

G3 Di 12:45-13:30

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	6	<input type="checkbox"/>	Skip-Tabelle, Simulation	Skip-Tabelle, Simulation	<input type="checkbox"/>	
Aufgabe 2	18	<input type="checkbox"/>	Java-Programm	Projekt Archiv	<input type="checkbox"/>	

Aufgabe 1: Stringsuche mit Boyer-Moore simulieren (6 Punkte)

Wenden Sie den Boyer-Moore-Algorithmus an. Bilden Sie zunächst die Skip-Tabelle für die Zeichenfolge "**FRISCH**". Suchen Sie dann diese Folge im abgebildeten Text (bis zum Ende!), und tragen Sie in den leeren Zellen die verglichenen Zeichen ein, wie in grau angedeutet. Verwenden Sie eine neue Zeile immer dann wenn die Sprungdistanz weniger als die Musterlänge beträgt, damit Sie genügend Platz für mögliche weitere Vergleiche haben. Markieren Sie gefundene Vorkommen der Zeichenkette durch einkreisen des Startbuchstabens, zum Beispiel: **F**

Skip-Tabelle

F	
R	
I	
S	
C	
H	
sonst	

Simulation

F	I	S	C	H	E	R	S		F	R	I	T	Z		F	I	S	C	H	
					H															

T		F	R	I	S	C	H	E		F	I	S	C	H	E	,		F	R	

I	S	C	H	E		F	I	S	C	H	E		F	I	S	C	H	T		

F	I	S	C	H	E	R	S		F	R	I	T	Z

Abzugeben ist: Skip-Tabelle, Simulation

Aufgabe 2: Stringsuche mit Brute-Force und mit Boyer-Moore (18 Punkte)

Implementieren Sie die Stringsuche-Algorithmen mit Brute-Force und nach Boyer-Moore. Implementieren Sie die Klassen *BruteForceSearcher* und *BoyerMooreSearcher* im Paket *at.jku.students.textsearch*.

Die *findMatches* Methode durchsucht den Text und erzeugt eine List mit den Positionen aller Vorkommnisse des Musters. Wenn der Parameter *matchCase* den Wert *true* hat, dann soll die Groß-/Kleinschreibung berücksichtigt werden. Wenn der Parameter *wholeWordOnly* den Wert *true* hat, zählt ein Muster nur dann als gefunden, wenn es als eigenes Wort vorkommt.

Testen Sie Ihre Implementierung mit dem vorgegebenen Programm *TextSearchMain*. Dieses Programm führt verschiedene Test aus. Die Methode *markMatches* markiert alle Vorkommnisse in einem Text. Beispiel:

```
*** Whiskeymixer: Mark all matches of 'Whiskey' ***

Der Whiskeymixer mixt frischen Whiskey.
  ^                   ^
Frischen Whiskey mixt der Whiskeymixer
  ^                   ^

Occurrences of 'Whiskey' (matchCase=true wholeWord=false): 4
```

Die Methode *benchmark* misst die Ausführungsdauer der beiden Methoden. (Um die Messgenauigkeit zu erhöhen, wird jede Suche 100-mal wiederholt.) Die Ausgabe sieht beispielsweise so aus:

```
*** Benchmark ***

Pattern                #  Brute ms      #  Boyer ms
-----
a                      634   152 ms      634   95 ms
affectionately         1    142 ms       1    44 ms
alarm                  2    120 ms       2    23 ms
...
...
to                     736   129 ms      736   65 ms
uncomfortably         1    134 ms       1    13 ms

Pattern                #  Brute ms      #  Boyer ms
-----
                        6786  5549 ms     6786  1315 ms
```

Abzugeben ist: Projekt Archiv

Implementierungshinweise:

- Verwenden Sie das Vorgabeprojekt **PI2_UE10.zip**.
- Fügen Sie Ihre Implementierung in den mit **TODO** markierten Teilen den Klassen *BruteForceSearcher* und *BoyerMooreSearcher* ein.
- Verwenden Sie als Liste die in der Vorgabe enthaltene Implementierung von *LinkedList*, die im Paket *at.jku.ssw.list.impl* zu finden ist.
- Verwenden Sie die Methode *Character.isLetterOrDigit(char c)* um zu prüfen, ob ein Zeichen ein Buchstabe oder ein Leerzeichen ist.
- Verwenden Sie die statische Methoden *Character.toLowerCase(char c)* oder *Character.toUpperCase(char c)* um einen Buchstaben in einen Kleinbuchstaben oder Großbuchstaben zu verwandeln.
- Verwenden Sie *Arrays.fill(array, value)* um ein Array mit einem konstanten Wert zu befüllen.
- Die Methode *toLowerCase(Locale.ROOT)* der Klasse *String* wandelt alle Buchstaben in einem *String* in Kleinbuchstaben um. Sie können diese Methode auf das Pattern anwenden da dieses meistens kurz ist. Der Eingabetext ist allerdings im Allgemeinen sehr lang weshalb Umwandeln nicht zu empfehlen ist.
- Ändern sie **keine public Interfaces** vorgegebener Skeleton Klassen.
- Halten Sie sich an die **Codierungsrichtlinien** auf der Kurs Website.