

FMuD

Termin 6

FMuD

Reguläre Ausdrücke

Übersicht

[abc] einzelner Buchstabe a, b, oder c

[^abc] einzelner Buchstabe außer a, b, oder c

[0-9] alle Zahlen von 0-9

[a-z] alle Buchstaben von a-z

[a-zA-Z] alle Buchstaben von a-z oder A-Z

\s Leerzeichen [\t\n\r\f\v]

\S Nicht-Leerzeichen [^ \t\n\r\f\v]

\d Dezimalziffer [0-9]

\D Nicht-Dezimalziffer [^0-9]

\w beliebiges Wortzeichen [a-zA-Z0-9_]

\W beliebiges Nicht-Wortzeichen [^a-zA-Z0-9_]

\b Position zwischen \w und \W, z.B. Anfang und Ende eines Wortes

\B eine Position nicht zwischen \w und \W

| Trennung mehrerer Möglichkeiten

() Gruppierung für Wiederholungsoperatoren

^ Stringanfang

\$ Stringende

	Quantifizierer
*	0 oder mehr Wiederholungen des vorangegangenen Musters
+	1 oder mehr Wiederholungen des vorangegangenen Musters
?	0 oder 1 Wiederholung des nachfolgenden Musters
{n}	genau n Wiederholungen des vorangegangenen Musters
{n,}	mindestens n Wiederholungen des vorangegangenen Musters
{n,m}	mindestens n und höchstens m Wiederholungen des vorangegangenen Musters

Aufgabe 20

Entwerfen Sie einen regulären Ausdruck für
„Personen-IDs“

Bsp.:

P1234

P3

P924212

Aufgabe 21

Entwerfen Sie einen regulären Ausdruck für
Vornamen

Bsp.:
Hans
Peter
Erika

Aufgabe 22

Entwerfen Sie einen regulären Ausdruck für
Telefonnummern

Bsp.:

080023123123

123123-123123

Aufgabe 23

Entwerfen Sie einen regulären Ausdruck für
Anrede

Herr oder Frau

Aufgabe 24

Entwerfen Sie einen regulären Ausdruck für
PLZ mit Länderkürzel

Bsp.:

D-12345

RUS-54321

Aufgabe 25

Entwerfen Sie einen regulären Ausdruck für
Anschrift

Bsp.:

Waldstrasse 3

Waldstr. 3a

Am Wald 23-1

St.-Georg-Str. 17

Aufgabe 26

Entwerfen Sie einen regulären Ausdruck für
Vornamen

(Auch Doppelnamen und zweite Vornamen erlaubt)

Bsp.:

Hans-Peter

Eva Maria

FMUD

Reguläre Ausdrücke in Java

Übersicht

[abc] einzelner Buchstabe a, b, oder c

[^abc] einzelner Buchstabe außer a, b, oder c

[0-9] alle Zahlen von 0-9

[a-z] alle Buchstaben von a-z

[a-zA-Z] alle Buchstaben von a-z oder A-Z

\s Leerzeichen [\t\n\r\f\v]

\S Nicht-Leerzeichen [^ \t\n\r\f\v]

\d Dezimalziffer [0-9]

\D Nicht-Dezimalziffer [^0-9]

\w beliebiges Wortzeichen [a-zA-Z0-9_]

\W beliebiges Nicht-Wortzeichen [^a-zA-Z0-9_]

\b Position zwischen \w und \W, z.B. Anfang und Ende eines Wortes

\B eine Position nicht zwischen \w und \W

| Trennung mehrerer Möglichkeiten

() Gruppierung für Wiederholungsoperatoren

^ Stringanfang

\$ Stringende

	Quantifizierer
*	0 oder mehr Wiederholungen des vorangegangenen Musters
+	1 oder mehr Wiederholungen des vorangegangenen Musters
?	0 oder 1 Wiederholung des nachfolgenden Musters
{n}	genau n Wiederholungen des vorangegangenen Musters
{n,}	mindestens n Wiederholungen des vorangegangenen Musters
{n,m}	mindestens n und höchstens m Wiederholungen des vorangegangenen Musters

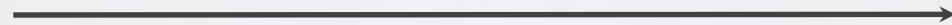
Reguläre Ausdrücke in Java

„\“ muss mit „\“ ‚escaped‘ werden!

„normal“

in Java

`\d`



`\\d`

Reguläre Ausdrücke in Java

Möglichkeit 1:

```
Pattern.matches(regex, s)
```

(„liefert“ boolean-Wert)

Möglichkeit 2:

```
Pattern p = Pattern.compile(regex);  
Matcher m = p.matcher(s);  
m.matches()
```

(„liefert“ boolean-Wert)

Aufgabe 27

Erstellen Sie ein Java-Programm das Tastatureingaben liest und diese auf das Format einer PLZ prüft.

Wenn die Eingabe nicht dem Format entspricht, soll eine entsprechende Meldung ausgegeben werden und erneut nach einer Eingabe gefragt werden. Falls die Eingabe dem Format entspricht, soll eine entsprechende Meldung ausgegeben werden und das Programm beendet werden.