

# Reguläre Ausdrücke

## Zeichenklassen

.	(Punkt)	Beliebiges Zeichen
\d		Ziffern
\D		Nicht Ziffern
\w		Wortzeichen (0-9a-zA-Z)
\W		Nicht Wortzeichen
\s		Whitespaces (Leerzeichen, Zeilenumbruch, Tab, Linefeed)
\S		Nicht-Whitespaces
[]		Selbstdefinierte Liste an Zeichen (nicht mit Komma trennen!) Beispiel 1: [a-zA-Z] (alle kleinen und großen Buchstaben) Beispiel 2: [aeiou] (Alle Vokale)
()		Alternative Zeichenketten. Beispiel (de com org)

## Spezialzeichen

- \* Vorstehendes Teilmuster ist 0 oder mehrmals enthalten
  - + Vorstehendes Teilmuster ist 1 oder mehrmals enthalten
  - ? Vorstehendes Teilmuster ist 0 oder einmal enthalten
  - {n,m} Vorstehendes Teilmuster ist mindestens n und maximal m Mal enthalten
- Beispiel 1: Vorwahl besteht aus 3-6 Zahlen `\d{3-6}`
- Beispiel 2: `{3,}` (mindestens 3 Mal)
- Beispiel 3: `{,6}` (maximal 6 Mal)
- \ Escape-Zeichen (Maskierung)

//3. Ersetzen

```
String telefonnummer = "49-7531-12345";  
telefonnummer = telefonnummer.replaceAll("(49)-(7531)-(\d*)", "$1-1337-$3");  
System.out.println("Die neue Telefonnummer lautet: " + telefonnummer);
```

Element 1

Element 2

Element 3

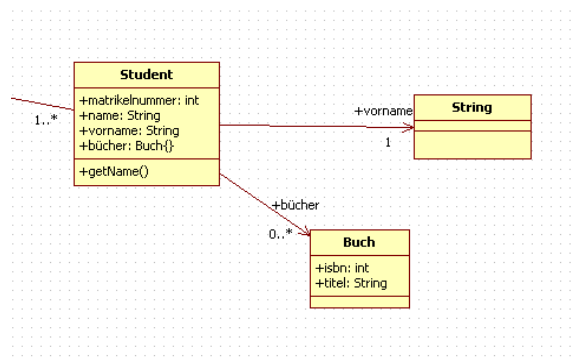
Mit \$-Zeichen kann man auf die durch Klammer gekennzeichneten Elemente zugreifen.

Das Ergebnis dieses Beispiels ist 49-1337-12345

## Hinweise zur Übungsaufgabe

1. Neue Testklasse anlegen: rechter Mausklick -> New -> JUnitTestCase
2. Klasse EMailCheck implementieren (aus Beispielcode übernehmen)
3. JUnitTestCase ausführen: rechter Mausklick -> Run As -> JUnitTest

# Abbilden von 1:n Beziehungen



1. Variante: Linked List verwenden  
siehe nächste Seite  
Nachteil: Zugriff wächst mit Größe  
Vorteil: Dynamisch (bzgl. Anzahl der Elemente)
2. Variante: Array verwenden  
Nachteil: Fest Länge  
Vorteil: Schneller Zugriff auf Element

```
class ArrayList{
    public void add(Object o) {
        //neues Array, das eins größer ist
        //alle Elemente einfügen
    }

    public void delete(int position){...}
    public void delete(Object o){...}
    public void replace(Object o, int position){...}
}
```

