

Arbeiten mit Datenstrukturen

Montag, 13. Februar 2023 02:31

| | | | | | | | | | | | | | | | |
|--------------------|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Zahlen und Formeln | <ul style="list-style-type: none">•Verarbeitung von Zahlen<ul style="list-style-type: none">-Zahlen oder Variablen, Konstanten-Operatoren: Arithmetisch oder Vergleich oder Funktionen-Stelligkeit-Präzedenz-Assoziativität-Klammerung | | | | | | | | | | | | | | |
| Strings | <ul style="list-style-type: none">•Verarbeitung von Text <table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>p</td><td>o</td><td>p</td><td>c</td><td>o</td><td>r</td><td>n</td></tr></table> <p>„popcorn“[3] „popcorn“.length „pop“+„corn“</p> <p>x ← „pop“ y ← „corn“ z ← x + y</p> <p>w ← z.substring(4,6)</p> <p>v ← z.indexOf(„c“) v ← z.indexOf(„o“) V ←z.indexOf(„a“)</p> <ul style="list-style-type: none">•Beispiel: elvis.presley@heartbreak.hotel.com <pre>Address ← readAddressFromUser() atIndex ← address.indexOf(„@“) Length ← address.length Username ← address.substring(0,atIndex) Hostsite ← address.substring(atIndex+1, length-4) Extension ← address.substring(length-3, length)</pre> <p>15. Januar 2015</p> <p>15/01/2015 vs. 01/15/2015</p> <pre>Edate ← readDatefromUser() Day ← Edate.substring(0,2) Month ← Edate.substring(3,5) Year ← Edate.substring(6,10) Adate ← Month +"/"+Day +"/"+Year</pre> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | p | o | p | c | o | r | n |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | | | | | | | | | |
| p | o | p | c | o | r | n | | | | | | | | | |
| Reguläre Ausdrücke | <p>-Häufig hilfreich, Muster angeben zu können</p> <ul style="list-style-type: none">•Regeln: -Jedes Zeichen ist ein Ausdruck, der das Zeichen repräsentiert. -Der Punkt ist ein Spezialausdruck, der ein beliebiges einzelnes Zeichen repräsentiert -Wenn A und B Ausdrücke sind, dann auch ->AB: Sequenzen: Ausdruck A gefolgt von Ausdruck B ->A B: Alternativen: Ausdruck A oder Ausdruck B ->(A): Gruppierung•Wichtig: Regeln:<ul style="list-style-type: none">• Jedes Zeichen ist ein Ausdruck, der das Zeichen repräsentiert.• Der Punkt ist ein Spezialausdruck, der ein beliebiges einzelnes Zeichen repräsentiert• Wenn A und B Ausdrücke sind, dann auch<ul style="list-style-type: none">• AB: Sequenzen: Ausdruck A gefolgt von Ausdruck B• A B: Alternativen: Ausdruck A oder Ausdruck B• (A): Gruppierung | | | | | | | | | | | | | | |

Regeln (cont):

- Wenn A ein Ausdruck ist, dann auch
 - A*: Null oder beliebig viele Wiederholungen von A
 - A+: Eine oder mehr Wiederholungen von A
 - A?: Null oder ein Vorkommen von A
 - A{m}: genau m Wiederholungen von A
 - A{m,n}: zwischen m und n Wiederholungen von A
 - A{m,}: mindestens m Wiederholungen von A

Regeln (cont): Zeichenklassen, z.B.

- Ziffern, „Wortzeichen“ (Alphanumerische Zeichen im engeren Sinn)
- \d: einzelne Ziffer
- \D: einzelnes Zeichen, das keine Ziffer ist
- \w: „Wortzeichen“ (a-z,A-Z,0-9)
- \W: einzelnes Zeichen, das kein „Wortzeichen“ ist
- \s: jede Form von Leerzeichen(space, tab, line feed)
- \S: einzelnes Zeichen, das kein Leerzeichen ist
- [a-b]: beliebiges Zeichen einer Klasse von a bis (einschließlich) b
 - Annahme: Zeichenklassen sind geordnet, so dass Ausdrücke der Form [a-b] sinnvoll interpretiert werden können.

• Beispiel:

elvis.presley@heartbreak.hotel.com

Swift.taylor@whitehorse.edu

bobdylan@love.and.theft.edu

`\w+L.?lw*@(lw+L.)+lw{3}`

CAGACTTTCAGGGAACCTTTAGAACTAGCCATTAG
CTATAGGGGAAACCTGGACTTTCAGGGCAGACTTT
CCAGGGAACCTTTAGACTAGCCATTAGCTATAGGGG
AAACCTGG ACTTTCAGGG

AC.*GAA.*AG.