

JAVA-ÜBUNGS-AUFGABEN KLASSEN 2

Bemerkungen (siehe auch):

<https://www.java-blog-buch.de/0308-stringbuffer-und-stringbuilder/>

Bemerkung zu Strings:

Strings sind unveränderbare Objekte (Strings are immutable).

1)

Legen Sie ein nichtleeres StringBufferobjekt sb1 mit beliebigem Inhalt an. Erzeugen sie eine Referenz sb2, die auf dasselbe Objekt zeigt.

Legen Sie ein neues StringBufferobjekt sb3 an, das den gleichen (nicht denselben !) Inhalt wie sb1 hat.

Welche Ergebnisse haben die Vergleiche sb1 == sb2 , sb1 == sb3 , sb2 == sb3 ?

2)

Wie kann man in der obigen Situation die Inhalte von zwei StringBuffer-Objekten vergleichen ?

3)

Sie wollen zwei StringBuffer-Objekte mit + verketteten, das Ergebnis soll ein String sein. Mit welchem kleinen Trick geht das ?

4)

Sie wollen zwei StringBuffer-Objekte mit + verketteten, das Ergebnis soll ein StringBuffer sein. Wie geht das ?

5)

Schreiben Sie eine Methode cleanString(...), die aus einem String alle Sonderzeichen entfernt. Die Methode bekommt einen String übergeben und gibt einen neuen String ohne Sonderzeichen zurück.

Als Sonderzeichen gelten alle Zeichen der oberen Hälfte des ASCII-Codes und alle Interpunktionszeichen.

Ziffern und Leerzeichen gelten nicht als Sonderzeichen.

6)

Schreiben Sie eine Methode deleteDigits(...), die alle Ziffern aus einem String entfernt.

7)

Schreiben Sie eine Methode deleteAllBlanks(...), die alle Leerzeichen aus einem String entfernt.

8)

Schreiben Sie eine Methode deleteBlanks(...), die doppelte Blanks aus einem String entfernen soll.

9)

Schreiben Sie eine Methode `deleteChar(...)`, die bestimmte Zeichen aus einem String entfernt.

10)

Schreiben Sie eine Methode `charCount(...)`, die zählt, wie oft ein bestimmtes Zeichen in einem String vorkommt.

Die Methode bekommt einen String `s` und ein `char ch` übergeben und zählt, wie oft `ch` in `s` vorkommt. Kommt `ch` nicht in `s` vor, so wird 0 zurückgegeben.

11)

Schreiben Sie eine Methode `wordCount(...)`, die zählt, wie oft ein bestimmtes Wort (Teilstring) in einem Text (String) vorkommt. Die Methode bekommt zwei Parameter vom Typ String übergeben. Der erste Parameter ist der Textstring, der zweite String stellt das Wort dar, dessen Vorkommen gezählt werden soll. Kommt das Wort nicht, so wird 0 zurückgegeben.

12)

Legen Sie eine Klasse `StringUtil` an, die obige Methoden als statische Methoden enthält.

13)

Lösen Sie die Punkte 5) bis 10) analog für `StringBuffer` als Parameter und Returntyp.

14)

Ergänzen Sie die Klasse `StringUtil` um die neuen Methoden.

Lösungen:

```
// ----- StringUtil ----- \\

import java.util.*;

public class StringUtil
{
    // ----- cleanString ----- \\
    // entfernt alle Sonderzeichen, bzw. übernimmt nur
    // Buchstaben (klein und groß), Ziffern und leerzeichen
    public static String cleanString(String st)
    {
        StringBuffer tmp = new StringBuffer();

        for(int i=0; i<st.length() ; i++)
        {
            char ch = st.charAt(i);
            if (ch==' ' || '0'<=ch && ch<='9' || 'a'<=ch && ch<='z'
                || 'A'<=ch && ch<='Z')
                tmp.append(ch);
        }
        return tmp.toString();
    }

    // ----- deleteDigits ----- \\
    // entfernt alle Ziffern aus einem String
    public static String deleteDigits(String st)
    {
        StringBuffer tmp = new StringBuffer();

        for(int i=0; i<st.length() ; i++)
        {
            char ch = st.charAt(i);
            if (ch<'0' || ch>'9')
                tmp.append(ch);
        }
        return tmp.toString();
    }

    // ----- deleteAllBlanks ----- \\
    // entfernt alle Ziffern aus einem String
    public static String deleteAllBlanks(String st)
    {
        StringBuffer tmp = new StringBuffer();

        for(int i=0; i<st.length() ; i++)
        {
            char ch = st.charAt(i);

            if (ch!=' ')
                tmp.append(ch);
        }

        return tmp.toString();
    }
}
```

```

// ----- deleteBlanks ----- \\
// entfernt alle mehrfachen Blanks aus einem String
public static String deleteBlanks(String st)
{
    StringBuffer tmp = new StringBuffer();
    StringTokenizer stok = new StringTokenizer(st, " " ) ;

    while (stok.hasMoreTokens())
        tmp.append( stok.nextToken() + ' ');

    return tmp.toString().trim();
}

// ----- deleteChar ----- \\
// entfernt alle mehrfachen Blanks aus einem String
public static String deleteChar(String st, char weg)
{
    StringBuffer tmp = new StringBuffer();
    for(int i=0; i<st.length() ; i++)
    {
        char ch = st.charAt(i);
        if (ch!=weg)
            tmp.append(ch);
    }
    return tmp.toString();
}

// ----- countChar ----- \\
// zählt, wie oft ein bestimmtes zeichen in einem string vorkommt
public static int countChar(String st, char toCount)
{
    int counter=0;
    for(int i=0; i<st.length() ; i++)
    {
        if ( st.charAt(i) == toCount)
            counter++ ;
    }
    return counter;
}
}

```