

Tutorium Woche 3 - Termin 1

Übungsblatt 2

Nils Pukropp

May 16, 2021

INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION

Übersicht

1. Vorstellung
2. Organisatorisches
3. Übungsblatt 2
4. Fragen und Vorrechnen

Über mich

- Nils Pukropp, 21 Jahre
 - Informatik Bachelor im 4. Semester
- E-Mail: nils.pukropp@student.kit.edu

Vorstellung

Regeln/Empfehlungen für MS Teams

- Ton aus/Webcam an (falls, vorhanden, Alternative: droidcam)
- De-Anonymisierung aktiviert
- Fragen stellen
 - einfach los fragen (einfach zwischen rein)
 - oder Hand heben
 - Duzen ist erwünscht



Organisatorisches

Organisatorisches

- Zum Vorrechnen im Ilias anmelden
- Übungsblatt 3
 - **Ausgabe:** 17. Mai, 13 Uhr
 - **Artemis:** 21. Mai, 13 Uhr
 - **Deadline:** 01. Juni, 06 Uhr

Vorrechnen

Ihr werdet ziemlich sicher bestehen, wenn eure Abgabe selbstständig erstellt wurde und das Vorrechnen ist deutlich angenehmer als die Präsenzübung

Aufgabe und Lernziele

Aufgabe und Lernziele

- Utility-Klasse für Strings schreiben

Aufgabe und Lernziele

- Utility-Klasse für Strings schreiben
- Lernziele

Aufgabe und Lernziele

- Utility-Klasse für Strings schreiben
- Lernziele
 - Umgang mit Strings

Aufgabe und Lernziele

- Utility-Klasse für Strings schreiben
- Lernziele
 - Umgang mit Strings
 - Aufbau von Utility-Klassen

Aufgabe A

Privater Konstruktor und final

```
1 public final class StringUtility {  
2  
3     private StringUtility() {  
4         throw new  
5             IllegalStateException(ERROR_UTILITY_CLASS_INSTANTIATION);  
6     }
```

Aufgabe A

Privater Konstruktor und final

```
1 public final class StringUtility {  
2  
3     private StringUtility() {  
4         throw new  
5             IllegalStateException(ERROR_UTILITY_CLASS_INSTANTIATION);  
6     }
```

capitalize

```
1 public static String capitalize(String word) {  
2     return word.substring(0, 1).toUpperCase() + word.substring(1);  
3 }
```

Aufgabe A

countCharacter

```
1 public static int countCharacter(String word, char character) {  
2     int count = 0;  
3     for (char wordCharacter : word.toCharArray()) {  
4         if (wordCharacter == character) {  
5             count += 1;  
6         }  
7     }  
8     return count;  
9 }
```


Aufgabe A

isAnagram

```
1 public static boolean isAnagram(String word1, String word2) {
2     // If the words have a different lengths, they cannot be anagrams
3     if (word1.length() != word2.length()) return false;
4     String anagram = word2;
5     // We do not need the indices so we use foreach
6     for (final char character : word1.toCharArray()) {
7         final int index = anagram.indexOf(character);
8         if (index == -1) return false;
9         anagram = removeCharacter(anagram, index);
10    }
11    return anagram.isEmpty();
12 }
```

Aufgabe A

isPalindrome

```
1 public static boolean isPalindrome(String word) {  
2     return word.equals(reverse(word));  
3 }
```

Aufgabe A

isPalindrome

```
1 public static boolean isPalindrome(String word) {  
2     return word.equals(reverse(word));  
3 }
```

removeCharacter

```
1 public static String removeCharacter(String word, int index) {  
2     return word.substring(0, index) + word.substring(index + 1);  
3 }
```

Aufgabe A

reverse

```
1 public static String reverse(String word) {  
2     String reverse = "";  
3  
4     // We need access to the indices of the characters,  
5     // so we have to use for  
6     // instead of foreach  
7     for (int i = word.length() - 1; i ≥ 0; i--) {  
8         reverse += word.charAt(i);  
9     }  
10  
11     return reverse;  
12 }
```

Aufgabe A

reverse

```
1 public static String reverse(String word) {  
2     String reverse = "";  
3  
4     // We need access to the indices of the characters,  
5     // so we have to use for  
6     // instead of foreach  
7     for (int i = word.length() - 1; i ≥ 0; i--) {  
8         reverse += word.charAt(i);  
9     }  
10  
11     return reverse;  
12 }
```

Alternative zu `String` im Loop?

Aufgabe A

reverse

```
1 public static String reverse(String word) {  
2     String reverse = "";  
3  
4     // We need access to the indices of the characters,  
5     // so we have to use for  
6     // instead of foreach  
7     for (int i = word.length() - 1; i ≥ 0; i--) {  
8         reverse += word.charAt(i);  
9     }  
10  
11     return reverse;  
12 }
```

Alternative zu `String` im Loop? `StringBuilder`!

Aufgabe A

reverse - StringBuilder

```
1 public static String reverse(String word) {  
2     StringBuilder reverse = new StringBuilder;  
3  
4     for (int i = word.length() - 1; i ≥ 0; i--) {  
5         reverse.append(word.charAt(i));  
6     }  
7  
8     return reverse.toString();  
9 }
```

Aufgabe B

Test?

Fragen?

Vorrechnen!