Teil VIII

Zeichenketten

String, StringBuilder

`String`

- unveränderlich
- stellt viele Funktionen als “Fassade” zur Verfügung

String, StringBuilder

`String`

- unveränderlich
- stellt viele Funktionen als “Fassade” zur Verfügung

`StringBuilder`

- veränderlich
- sollte für Konstruktion und Veränderung von Zeichenketten verwendet werden

StringBuilder: wichtige Methoden

- Veränderung
  - `StringBuilder append(.)`
  - `StringBuilder insert(int offset, .)`
  - `StringBuilder delete(int start, int end)`
  - `StringBuilder replace(int start, int end, String str)`

StringBuilder: wichtige Methoden

- Veränderung
  - `StringBuilder append(.)`
  - `StringBuilder insert(int offset, .)`
  - `StringBuilder delete(int start, int end)`
  - `StringBuilder replace(int start, int end, String str)`
- Länge
  - `int length()`

StringBuilder: wichtige Methoden

- Veränderung
  - `StringBuilder append(.)`
  - `StringBuilder insert(int offset, .)`
  - `StringBuilder delete(int start, int end)`
  - `StringBuilder replace(int start, int end, String str)`
- Länge
  - `int length()`
- Suche
  - `int indexOf( String str )`
  - `int lastIndexOf( String str )`
  - `int indexOf( String str, int fromIndex )`
  - `int lastIndexOf( String str, int fromIndex )`

StringBuilder: Veränderung

Beispiel `StringBuilder.append`:

```
1    StringBuilder ausgabe = new StringBuilder();
2    for (int i = 2;
3         i <= 30;
4         ++i) {
5      if (isPrimeNumber(i)) {
6        ausgabe.append(i)
7                .append(" ");
8      }
9    }
```

Ausgabe:

2 3 5 7 11 13 17 19 23 29

StringBuilder: Veränderung

Beispiel `StringBuilder.insert`, `StringBuilder.indexOf`:

```
1    StringBuilder ausgabe = new StringBuilder("Dies ein Test
       .");
2    ausgabe.insert(ausgabe.indexOf("ein"), "ist ");
```

Ausgabe:

```
Dies ist ein Test.
```

StringBuilder: Veränderung

Beispiel `StringBuilder.delete`, `StringBuilder.indexOf`:

```
1    StringBuilder ausgabe = new StringBuilder();
2    ausgabe.append("Dies ist ein schwerer Test.");
3    ausgabe.delete(ausgabe.indexOf("schwerer"),
4                   ausgabe.indexOf("Test"));
```

Ausgabe:

```
Dies ist ein Test.
```

StringBuilder: Veränderung

Beispiel `StringBuilder.replace`, `StringBuilder.indexOf`, `StringBuilder.lastIndexOf`:

```
1    int fromIndex = 0;
2    StringBuilder ausgabe = new StringBuilder();
3    ausgabe.append("Das Program, das ich geschrieben habe.\n
      ");
4    ausgabe.append("Das Haus, das ich gebaut habe.\n");
5    ausgabe.append("Das Fahrrad, das ich gefahren habe.\n");
6
7    int indexReplace = ausgabe.indexOf("das");
8    while (indexReplace > 0) {
9      ausgabe.replace(indexReplace,
10                     indexReplace + "das".length(),
11                     "welches");
12     indexReplace = ausgabe.lastIndexOf("das");
13   }
```

StringBuilder: Veränderung

Eingabe:

```
Das Program, das ich geschrieben habe.
Das Haus, das ich gebaut habe.
Das Fahrrad, das ich gefahren habe.
```

Ausgabe:

```
Das Program, welches ich geschrieben habe.
Das Haus, welches ich gebaut habe.
Das Fahrrad, welches ich gefahren habe.
```

StringBuilder: wichtige Methoden

- Sonstige
  - `StringBuilder reverse()`
  - `String toString()`
  - `String subString(int start, int end)`
  - `String subString(int start)`

String: wichtige Methoden

- Länge
  - `int length()`
- Suche
  - `int indexOf( String str )`
  - `int lastIndexOf( String str )`
  - `int indexOf( String str, int fromIndex )`
  - `int lastIndexOf( String str, int fromIndex )`
- Sonstige
  - `String subString(int start, int end)`
  - `String subString(int start)`

wie StringBuilder

String: wichtige Methoden

- Vergleich
  - `int compareTo(String str)`
  - `int compareToIgnoreCase(String str)`

String: wichtige Methoden

- Vergleich
  - `int compareTo(String str)`
  - `int compareToIgnoreCase(String str)`
  - `boolean equals(String str)`
  - `boolean equalsIgnoreCase(String str)`

String: wichtige Methoden

- Vergleich
  - `int compareTo(String str)`
  - `int compareToIgnoreCase(String str)`
  - `boolean equals(String str)`
  - `boolean equalsIgnoreCase(String str)`
  - `boolean startsWith(String str)`
  - `boolean endsWith(String str)`

String: wichtige Methoden

- Vergleich
  - `int compareTo(String str)`
  - `int compareToIgnoreCase(String str)`
  - `boolean equals(String str)`
  - `boolean equalsIgnoreCase(String str)`
  - `boolean startsWith(String str)`
  - `boolean endsWith(String str)`
- Weitere Methoden
  - `String toLowerCase()`
  - `String toUpperCase()`
  - `String trim()`

String: reguläre Ausdrücke

- Reguläre Ausdrücke:
  - `String[] split( String regex )`
  - `boolean matches( String regex )`

String: reguläre Ausdrücke

Beispiel `String.split`:

```
1    String[] result = "blue,green;red.yellow".split("[.;,]")
       ;
2
3    for (String color : result) {
4      System.out.println(color);
5    }
```

Ausgabe:

```
blue
green
red
yellow
```