



# ***Vertretungsplan mit Java***

***StringBuilder,  
Daten lesen/schreiben***

# Die Java-Klasse StringBuilder

- String text;  
text = text + "neuer Text";

**Sehr ineffizient  
für große Texte.**



- Besser: StringBuilder

```
StringBuilder sb = new StringBuilder();  
sb.append("neuer Text1");  
sb.append("neuer Text2");  
String erg = sb.toString();
```

- Siehe Dokumentation für weitere Befehle:

```
java StringBuilder suchen
```

## HTML mit Java erzeugen

```
StringBuilder sb = new StringBuilder();  
sb.append("<!DOCTYPE html>\n");  
sb.append("<html>\n");  
sb.append("<head>\n");  
sb.append("  <meta charset=\"utf-8\">\n");  
...  
String erg = sb.toString();
```

# Generieren von HTML-Code



- Schreiben Sie ein Programm, dass eine HTML-Datei erzeugt, die Hello World ausgibt: `<p>Hello World</p>`
- Ergänzen Sie Ihr Programm, so dass diese Zeile 1000 mal ausgegeben wird.
- Speichern Sie Ihre HTML-Datei in eine eigene Datei (s. ueberblick-java.pdf)

- Erzeugen Sie eine andere HTML-Datei, die eine Tabelle mit  $n \times m$  Zellen, die Zufallszahlen enthalten.

3	78	92	63	19	37	45	69	83	70	56	9
22	43	39	100	54	10	49	63	62	96	41	20
44	78	88	51	13	98	20	100	90	66	4	13
28	13	33	81	87	9	25	79	27	10	80	20
84	22	31	23	2	47	80	3	76	45	95	97
88	48	42	37	30	53	16	4	80	68	55	86
82	6	92	2	29	75	24	73	56	17	21	66
32	23	33	7	68	17	94	73	3	35	52	39
62	56	78	88	49	25	62	22	85	58	43	43

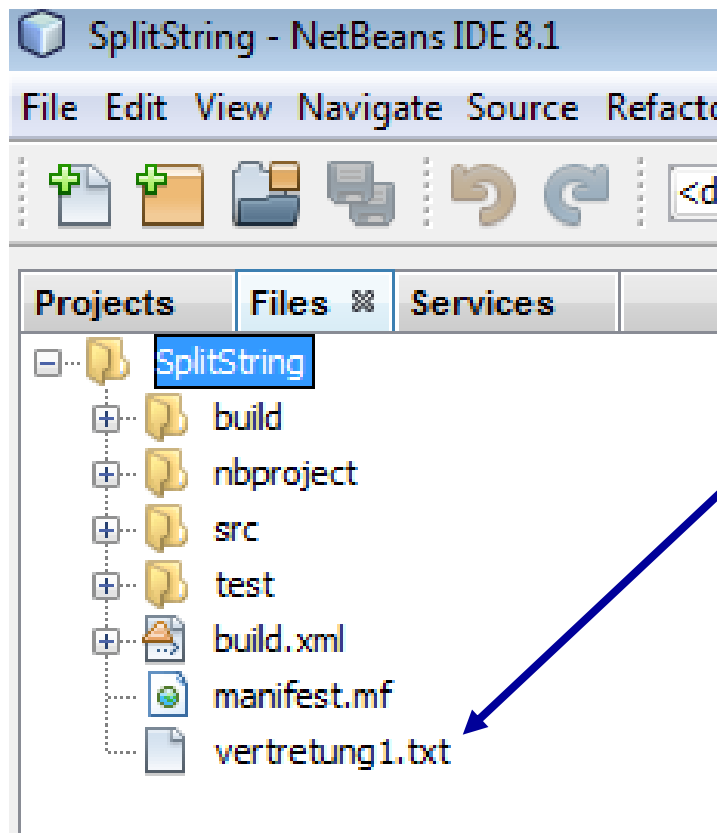
- Färben Sie die Zellen mit verschiedenen Farben:  
`<td style = "background-color: rgb(255,0,0) " > ... </td>`

# Daten lesen / schreiben

- Siehe Datei **ueberblick-java.pdf**

Textdatei einlesen	<pre>//import java.io.*; über der Klasse angeben String filename="test.txt"; try {     FileReader file = new <b>FileReader</b>(filename);     BufferedReader buff = new <b>BufferedReader</b>(file);     String line;     <b>while</b>((line=buff.readLine())!=null) {         System.out.println(line);     }     buff.close(); } catch (IOException e) {System.out.println("Error:"+e);}</pre>
Textdatei schreiben	<pre>//import java.io.*; über der Klasse angeben String filename="test.txt"; boolean append=true; try {     FileWriter file = new <b>FileWriter</b>(filename,append);     BufferedWriter buff = new <b>BufferedWriter</b>(file);     PrintWriter out = new <b>PrintWriter</b>(buff);     <b>out.println</b>("Neue Zeile");     out.flush();     out.close(); } catch (IOException e) {System.out.println("Error:"+e);}</pre>
Alternativ ohne PrintWriter	<pre>... buff.append(text); buff.newLine(); ...</pre>

# Ort der Textdatei im Netbeans-Projekt



```
public static void main(String[] args) {  
    String filename = "vertretung1.txt";  
  
    try {  
        FileReader file = new FileReader(filename);  
        BufferedReader buff = new BufferedReader(file);  
        String line;  
        while ((line = buff.readLine()) != null) {  
            System.out.println(line);  
        }  
        buff.close();  
    } catch (IOException e) {  
        System.out.println("Error:" + e);  
    }  
}
```

## Generierte Datei im Browser anzeigen lassen

Windows:

```
new ProcessBuilder( "cmd", "/c", "start", "/B", filename ).start();
```

Linux:

```
new ProcessBuilder( "firefox", filename ).start();
```

## Datenzeile von String in Array aufteilen

```
String text = "a;b;c";  
String[] textArray = text.split(";");
```



**Trennzeichen**

```
System.out.println(textArray[0]); //a  
System.out.println(textArray[1]); //b  
System.out.println(textArray[2]); //c
```