

Anpassung der Testdaten:

Vorher:

```
private String testTexte;
```

Neu:

```
private String[] testTexte;
```

Hinzufügen der Testdaten, aufgeteilt in Äquivalenzklassen:

```
void setUp() {
    testTexte = new String[]{
        //initial
        "Test Wort Dings!!! Test",
        //Gültige Äquivalenzklassen
        "Fritz sagt\" die", //Äquivalenzklasse 1
        "Hans ?mEInt ,ER", //Äquivalenzklasse 1
        "JOHN ?,fuegt hinzu", //Äquivalenzklasse 2
        "peter findet\". DaSs", //Äquivalenzklasse 2
        "sUsI :schaut! dort", //Äquivalenzklasse 3
        "Fritz \"HeBt;: das", //Äquivalenzklasse 3
        "NAJiS .F!I?N;D:E,(T) , \" DEN", //Äquivalenzklasse 4
        "frITz ;,z?!e\",g;?t:! dort", //Äquivalenzklasse 4
        ".,?!\":;\"", //Äquivalenzklasse 5
        //Ungültige Äquivalenzklassen
        "", //Äquivalenzklasse 6
        "sagen", //Äquivalenzklasse 6
        "0", //Äquivalenzklasse 6
        "Maja hört zu", //Äquivalenzklasse 6
        "Sunita sagt% dazu", //Äquivalenzklasse 7
        "Lorem %ipsum oder", //Äquivalenzklasse 7
        "John i%st hier", //Äquivalenzklasse 8
        "Jul ge?ht davon", //Äquivalenzklasse 8
    };
}
```

Neu: System.out.Prints der Resultate:

```
@Test
Void testEntferneSatzzeichenAmAnfang
UndAmEnde()
```

Zeichenentfernung aller Wörter in Text Nr. 8:

```
najis          ->    najis
"najis"        ->    najis
".najis,!      ->    najis
"najis?"       ->    najis
"?:najis?"     ->    najis
"najis,"       ->    najis
".najis:."     ->    najis
":najis"       ->    najis
"!najis!."     ->    najis
"!najis;"      ->    najis
",!najis,;"    ->    najis
".najis?"      ->    najis
":najis"?"     ->    najis
",najis:"      ->    najis
"?najis.?"     ->    najis

.fli?n;d:e,(t) -> fli?n;d:e,(t)
?fli?n;d:e,(t)? -> fli?n;d:e,(t)
.f?fli?n;d:e,(t),: -> fli?n;d:e,(t)
,fli?n;d:e,(t); -> fli?n;d:e,(t)
!fli?n;d:e,(t)! -> fli?n;d:e,(t)
```

```
@Test
void testVerarbeiteText()
```

```
Verarbeitung von Text Nr. 1:
test=2
dings=1
wort=1

Verarbeitung von Text Nr. 2:
fritz=1
die=1
test=2
dings=1
wort=1
sagt=1

Verarbeitung von Text Nr. 3:
fritz=1
die=1
test=2
dings=1
meint=1
wort=1
er=1
sagt=1
hans=1

Verarbeitung von Text Nr. 4:
fritz=1
die=1
hans=1
```

```
@Test
void testEntferneSatzzeichenImWort() {
```

Test für Satzzeichen innerhalb eines Wortes:

```
!          ->    test!wort
!          ->    test!!wort
"          ->    test"wort
"          ->    test""wort
:          ->    test:wort
:          ->    test::wort
;          ->    test;wort
;          ->    test;;wort
,          ->    test,wort
,          ->    test,,wort
.          ->    test.wort
.          ->    test..wort
?          ->    test?wort
?          ->    test??wort
```

Anpassung in der Anwendung aufgrund der Testresultate:

1.




```
public class Worthaeufigkeitsanalyse {  
  
    static HashSet<String> satzzeichen = new HashSet<String>();  
    static {  
        satzzeichen.add(".");  
        satzzeichen.add(",");  
        satzzeichen.add("?");  
        satzzeichen.add("!");  
        // satzzeichen.add("'");  
        satzzeichen.add(":");  
        satzzeichen.add(";");  
        satzzeichen.add("\");  
    }  
}
```


2.






```
public void verarbeiteText(String text) {  
    ...  
    if (!satzzeichen.contains(wort) && !wort.isEmpty())  
    ...  
}
```

Test Results:

Finished after 0.141 seconds

Runs: 3/3  Errors: 0  Failures: 0 

 WorthaeufigkeitsanalyseTest [Runner: JUnit 5] (0.088 s) Fail

-  testEntferneSatzzeichenAmAnfangUndAmEnde() (0.069 s) 
-  testVerarbeiteText() (0.011 s) 
-  testEntferneSatzzeichenImWort() (0.008 s) 