



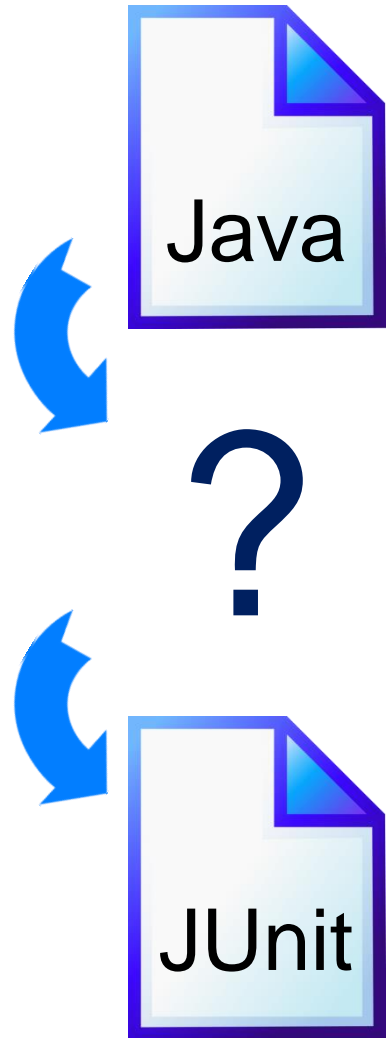
**BERNER & MATTNER**  
AN ASSYSTEM COMPANY

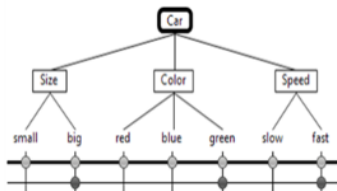
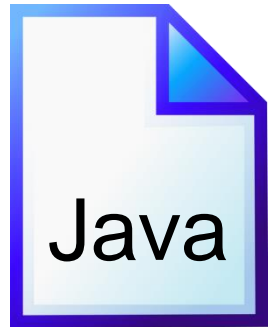
# Klassifikationsbäume als Testrahmen für den Modultest automatisch generieren

**Andreas Schäfer**

## Problem: Generierung von JUnit-Tests

---



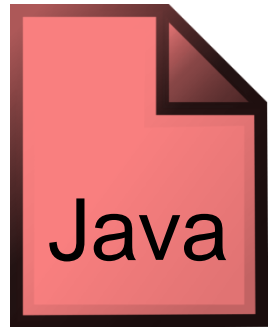


- Generierung eines Klassifikationsbaums als Zwischenprodukt



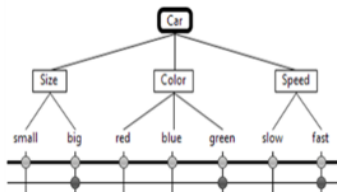
## Beispiel: Unit-Tests mit Klassifikationsbaum-Methode

---



**String** getLastArticleDate(**Author** author)

- Methodensignaturen analysieren
- Zentrale Token identifizieren
  - Modifier, Rückgabebetyp , Bezeichner, Parameter

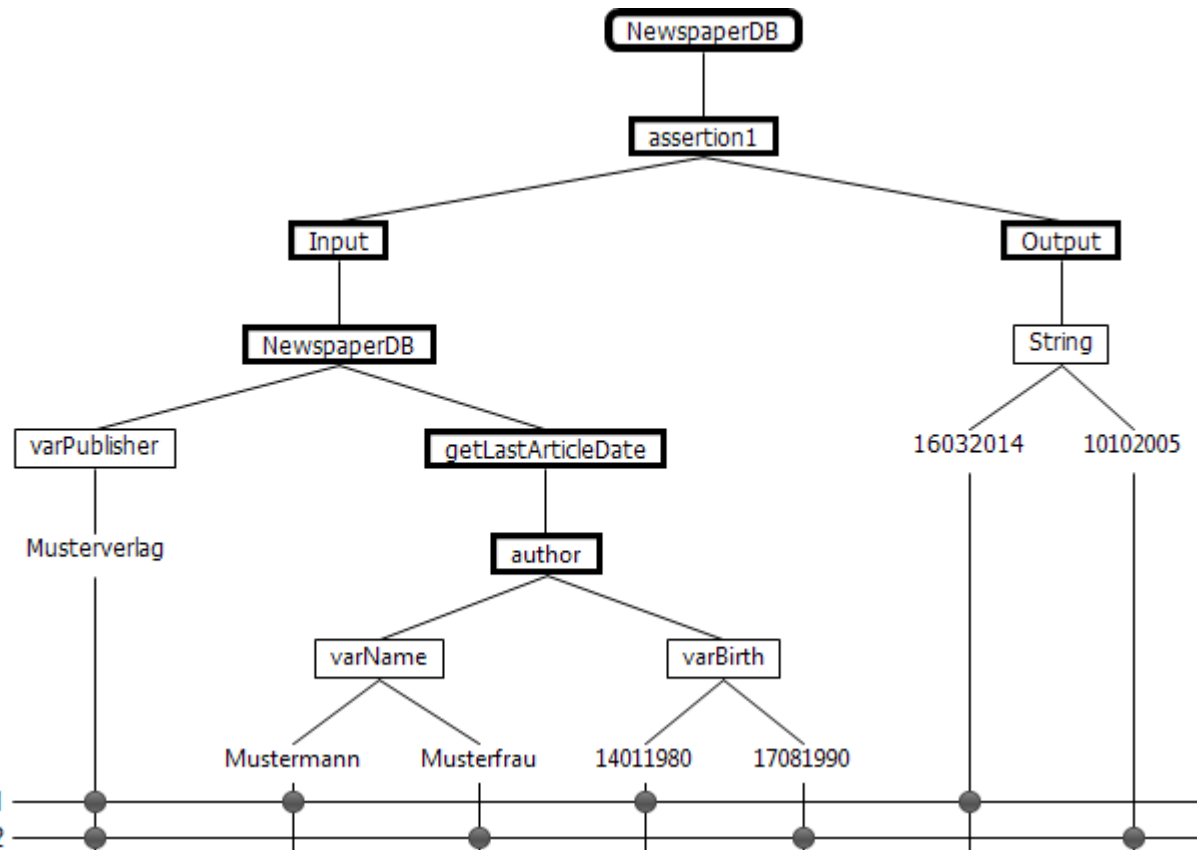
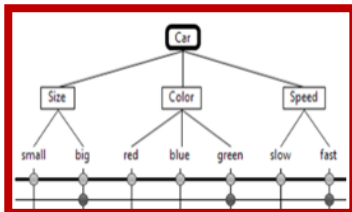
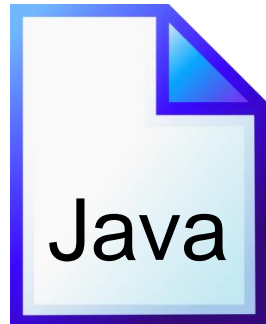


- Weiterverarbeitung als Syntaxbaum
- Ergebnis ist I/O-Klassifikationsbaum

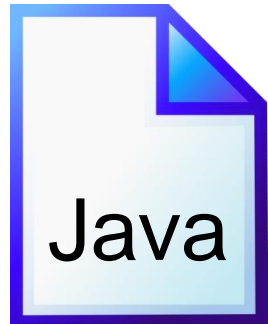


# Der resultierende I/O-Klassifikationsbaum

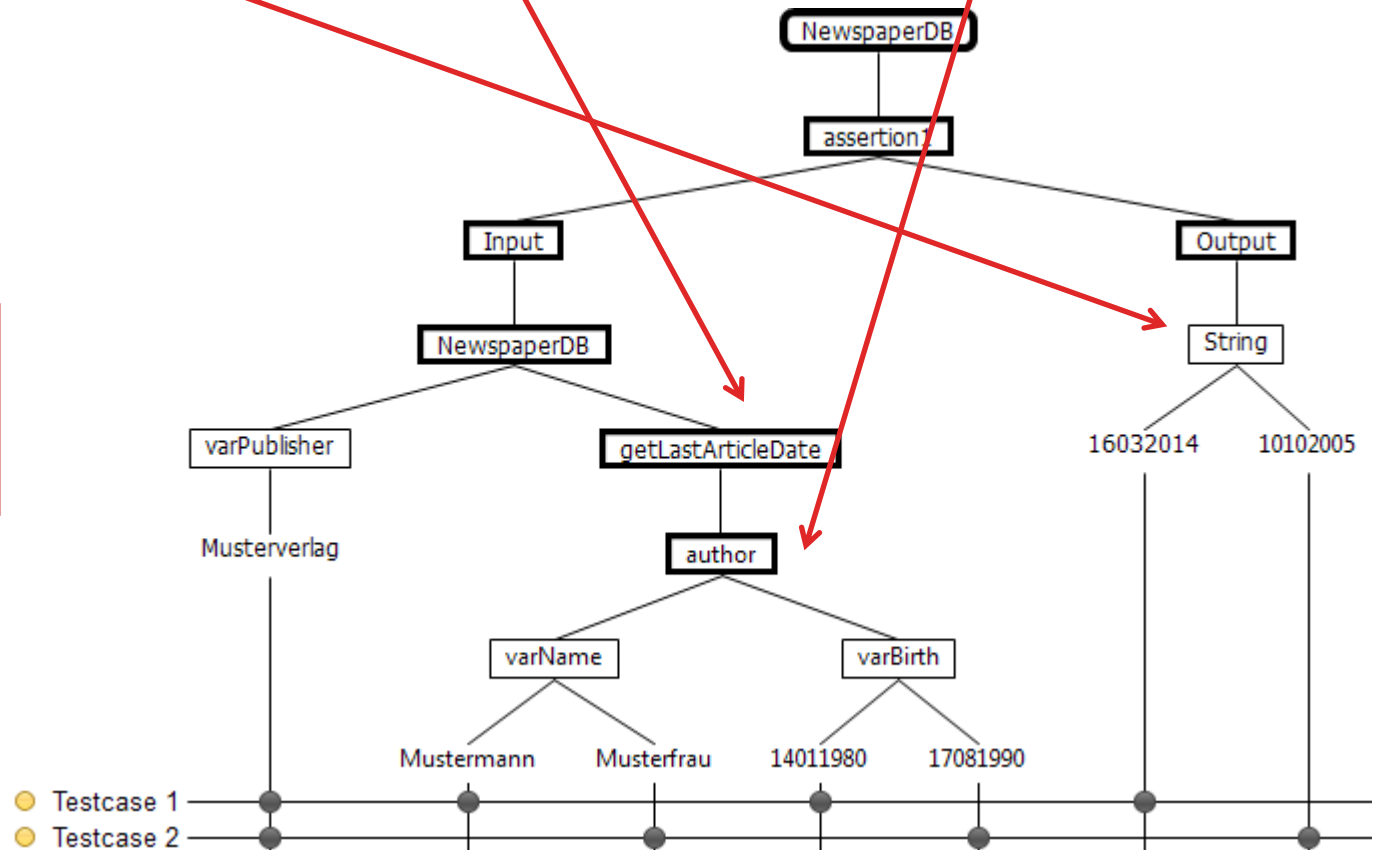
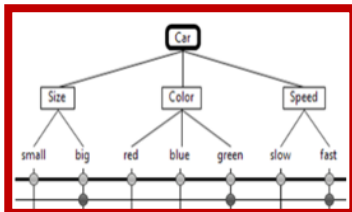
**String** getLastArticleDate(**Author** author)



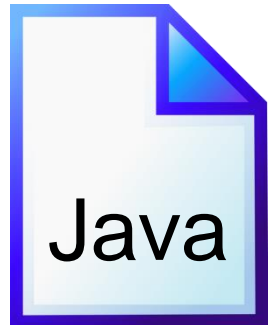
# Der resultierende I/O-Klassifikationsbaum



**String getLastArticleDate(Author author)**



# Beispiel für die Generierung



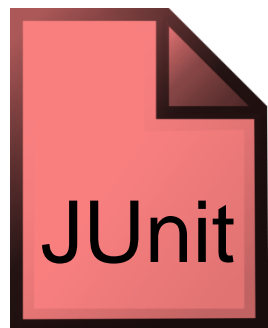
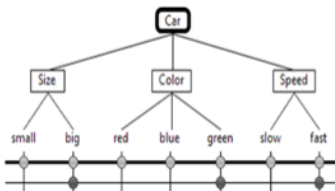
## Der generierte JUnit Test (nur oberer Testfall)

```
package newspaper;
import static org.junit.Assert.assertEquals;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;

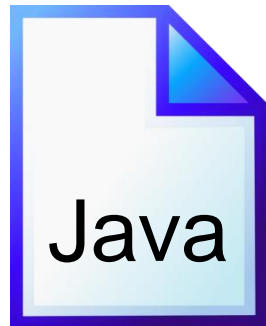
public class NewspaperDBTest{

    @Rule
    public ExpectedException thrown =
        ExpectedException.none();

    @Test
    public void tc_1(){
        NewspaperDB NewspaperDB_id129 = new
            NewspaperDB("Musterverlag");
        String getLastArticleDate_id134 =
            NewspaperDB_id129.getLastArticleDate(
                new Author("Mustermann", "14011980"));
        String String_id60 = "16032014";
        assertEquals("The same?", String_id60, (
            getLastArticleDate_id134));
    }
}
```



# Beispiel für die Generierung



## Der generierte JUnit Test (nur oberer Testfall)

```
package newspaper;
import static org.junit.Assert.assertEquals;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
```

```
public class NewspaperDBTest{
```

```
    @Rule
```

```
    public ExpectedException thrown =
        ExpectedException.none();
```

```
    @Test
```

```
    public void tc_1(){
```

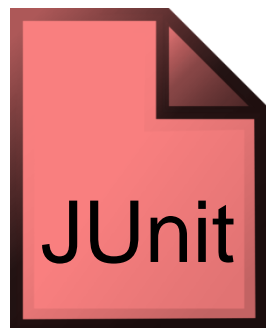
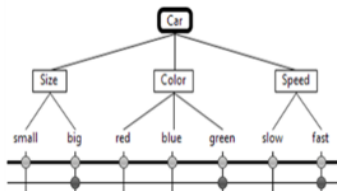
```
        NewspaperDB NewspaperDB_id129 = new
            NewspaperDB("Musterverlag");
```

```
        String getLastArticleDate_id134 =
            NewspaperDB_id129.getLastArticleDate(
                new Author("Mustermann", "14011980"));
```

```
        String String_id60 = "16032014";
        assertEquals("The same?", String_id60, (
            getLastArticleDate_id134));
```

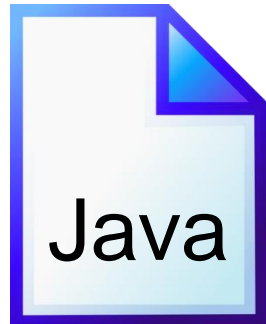
```
    }
```

```
}
```





# Beispiel für die Generierung



## Der generierte JUnit Test (nur oberer Testfall)

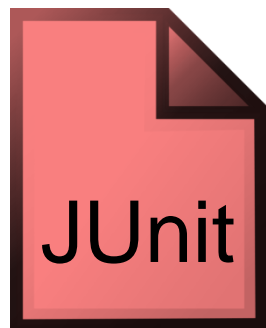
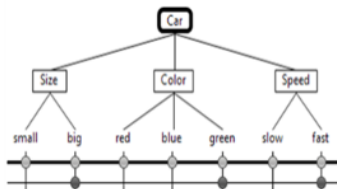
```
package newspaper;
import static org.junit.Assert.assertEquals;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
```

```
public class NewspaperDBTest{
```

```
@Rule
public ExpectedException thrown =
    ExpectedException.none();
```

```
@Test
public void tc_1(){
    NewspaperDB NewspaperDB_id129 = new
        NewspaperDB("Musterverlag");
    String getLastArticleDate_id134 =
        NewspaperDB_id129.getLastArticleDate(
            new Author("Mustermann", "14011980"));
    String String_id60 = "16032014";
    assertEquals("The same?", String_id60, (
        getLastArticleDate_id134));
```

```
}
}
```



varPublisher

Musterverlag

author

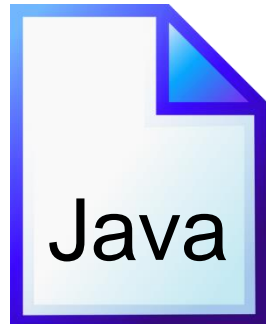
varName

Mustermann

varBirth

14011980

# Beispiel für die Generierung



## Der generierte JUnit Test (nur oberer Testfall)

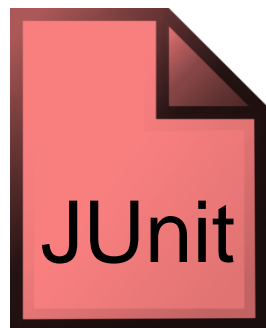
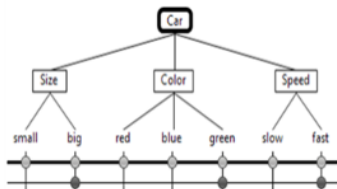
```
package newspaper;
import static org.junit.Assert.assertEquals;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
```

```
public class NewspaperDBTest{
```

```
@Rule
public ExpectedException thrown =
    ExpectedException.none();
```

```
@Test
public void tc_1(){
    NewspaperDB NewspaperDB_id129 = new
        NewspaperDB("Musterverlag");
    String getLastArticleDate_id134 =
        NewspaperDB_id129.getLastArticleDate(
            new Author("Mustermann", "14011980"));
    String String_id60 = "16032014";
    assertEquals("The same?", String_id60, (
        getLastArticleDate_id134));
```

```
}
}
```



varPublisher

Musterverlag

author

varName

Mustermann

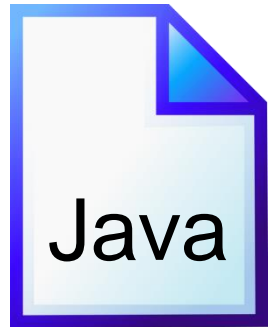
varBirth

14011980

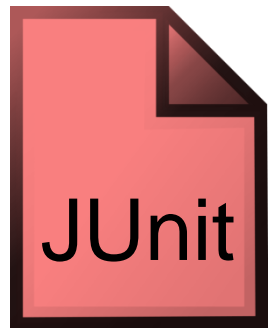
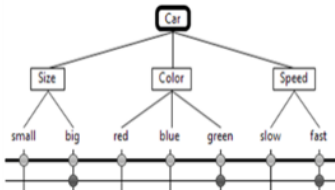
String

16032014

## Weiterverarbeitung des I/O-Klassifikationsbaums

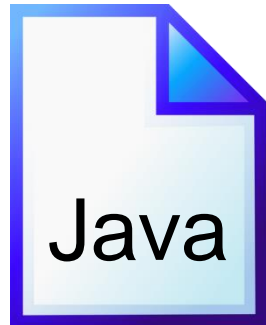


- Generierungsalgorithmus wandelt I/O-Klassifikationsbaum in JUnit-Test um
- Bezeichner der Klassen werden zu Elementen im Unit-Test
- Attribute liefern Informationen über Datentypen, Container-Klassen...
- Tester kann Testfallgenerierung nutzen oder Baum händisch nachbearbeiten

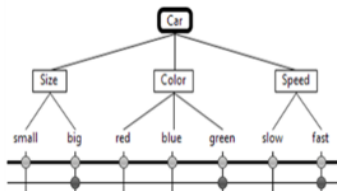


## Fazit & Ausblick

---



- Klassifikationsbaum-Methode geeignet zur Unit-Test-Erstellung
- Werkzeugunterstützung (TESTONA)
- Klassifikationsbaum als Meta-Objekt für Generierung von Unit-Tests aus beliebigen Programmiersprachen



### Ausblick

- Automatisches Einlesen von Output-Daten (Stichwort: Regressionstests)
- Implementierung weiterer spezieller Typen und Szenarien (Wildcard-Typen, Closures...)



Andreas Schäfer [andreas.schaefer@berner-mattner.com](mailto:andreas.schaefer@berner-mattner.com)