

Advanced ArchUnit: Tests auf ByteCode- Analysen aufbauen



Kai Schmidt
Selbständig
Software-Architekt und -Entwickler

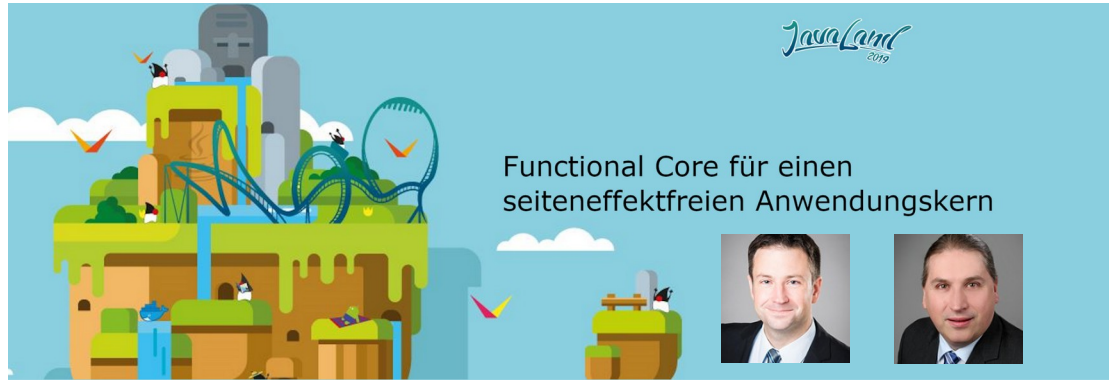


Thomas Ruhroth
msg systems ag - Travel & Logistics
Software-Architekt und Entwickler
Lead IT Consultant

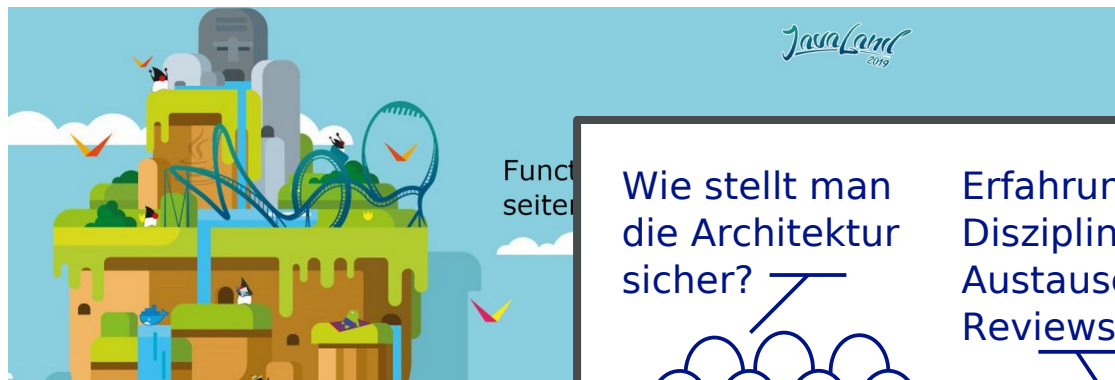


Die Geschichte zu unserem Thema

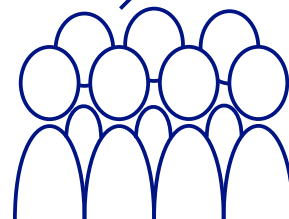
1	6.10.2022	JFN 2022
Advanced ArchUnit: Tests auf ByteCode-Analysen aufbauen		
Thomas Ruhroth und Kai Schmidt		



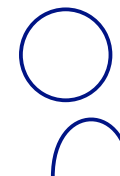
Die Geschichte zu unserem Thema



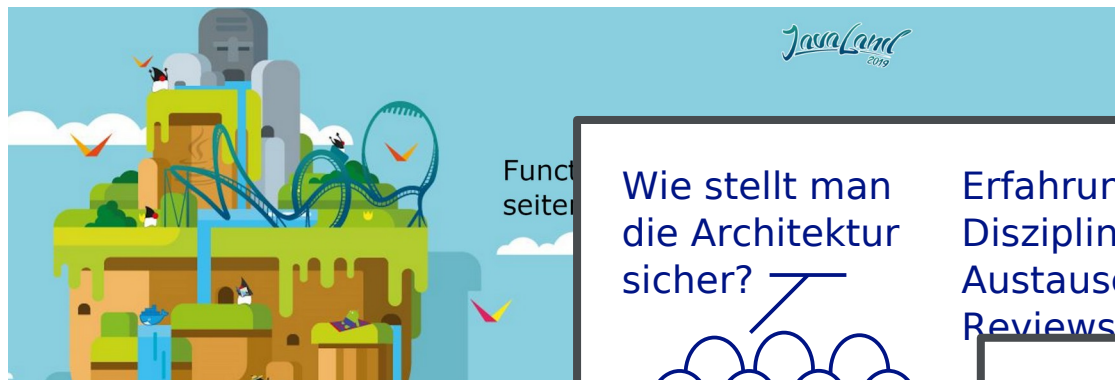
Wie stellt man
die Architektur
sicher?



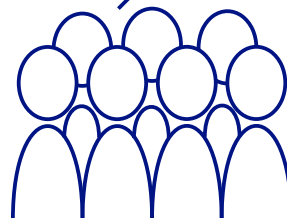
Erfahrung,
Disziplin,
Austausch und
Reviews



Die Geschichte zu unserem Thema



Wie stellt man
die Architektur
sicher?



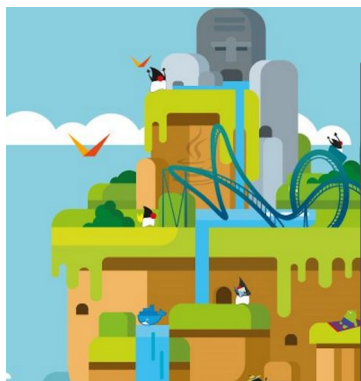
Erfahrung,
Disziplin,
Austausch und
Reviews

Wie funktioniert ArchUnit im
Inneren

Wie baut man eigene
Prüfungen?

Mit der IDE in den Code

Die Geschichte zu unserem Thema

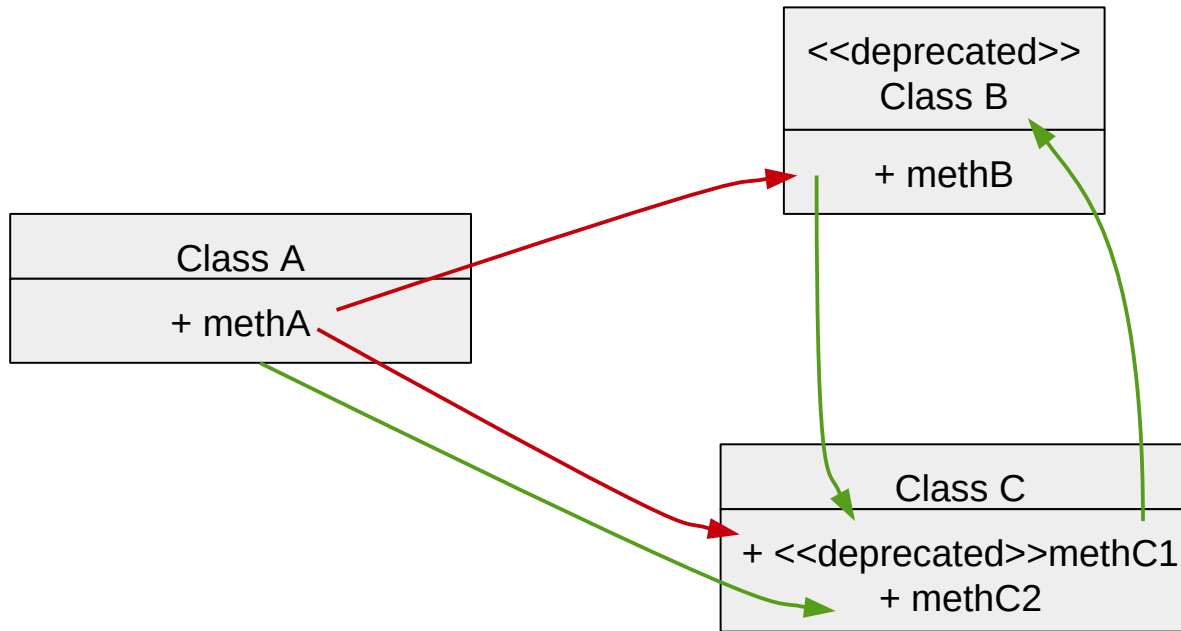


Achtung:
Dies ist keine
ArchUnit-Einführung

im

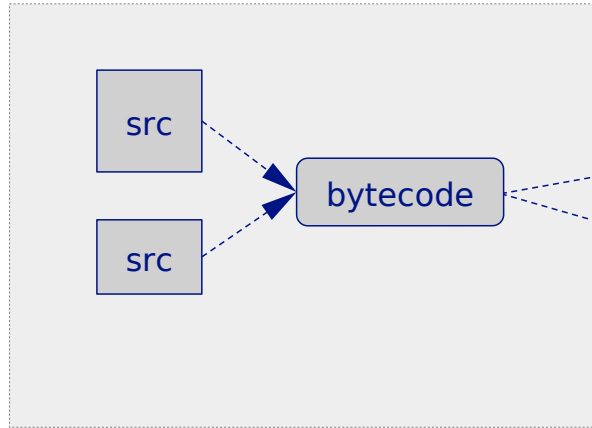
n

Erlaubte und nicht erlaubte Beziehungen



Wie arbeitet ArchUnit im Inneren

Java JDK

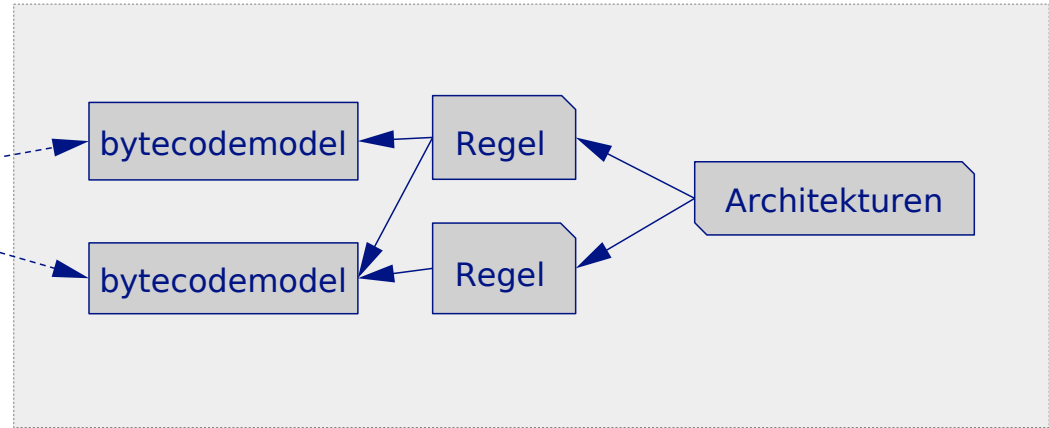


ArchUnit

Core API

Lang API

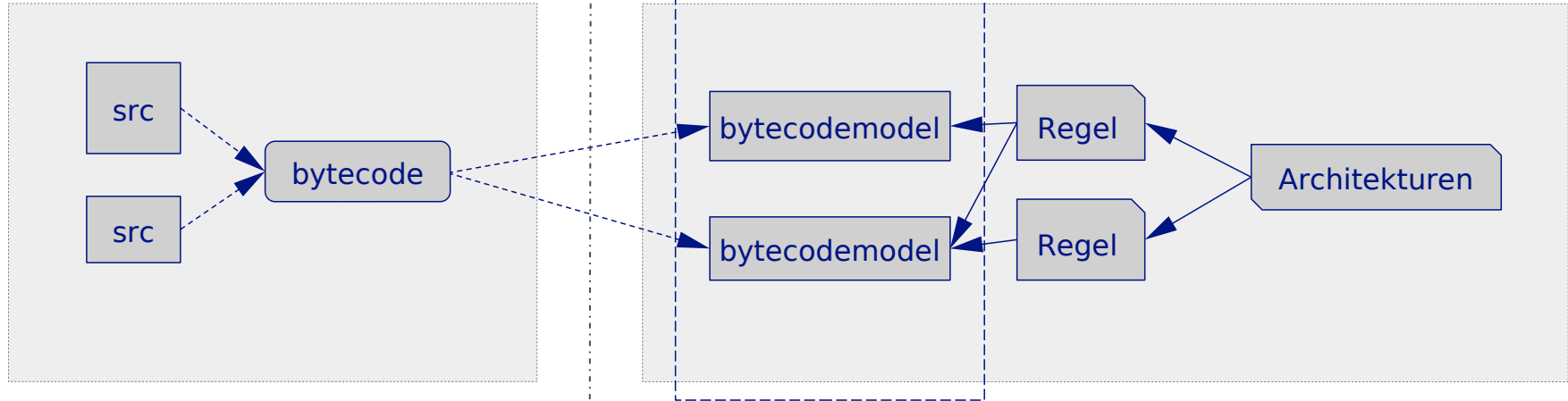
Library API



Wie arbeitet ArchUnit im Inneren

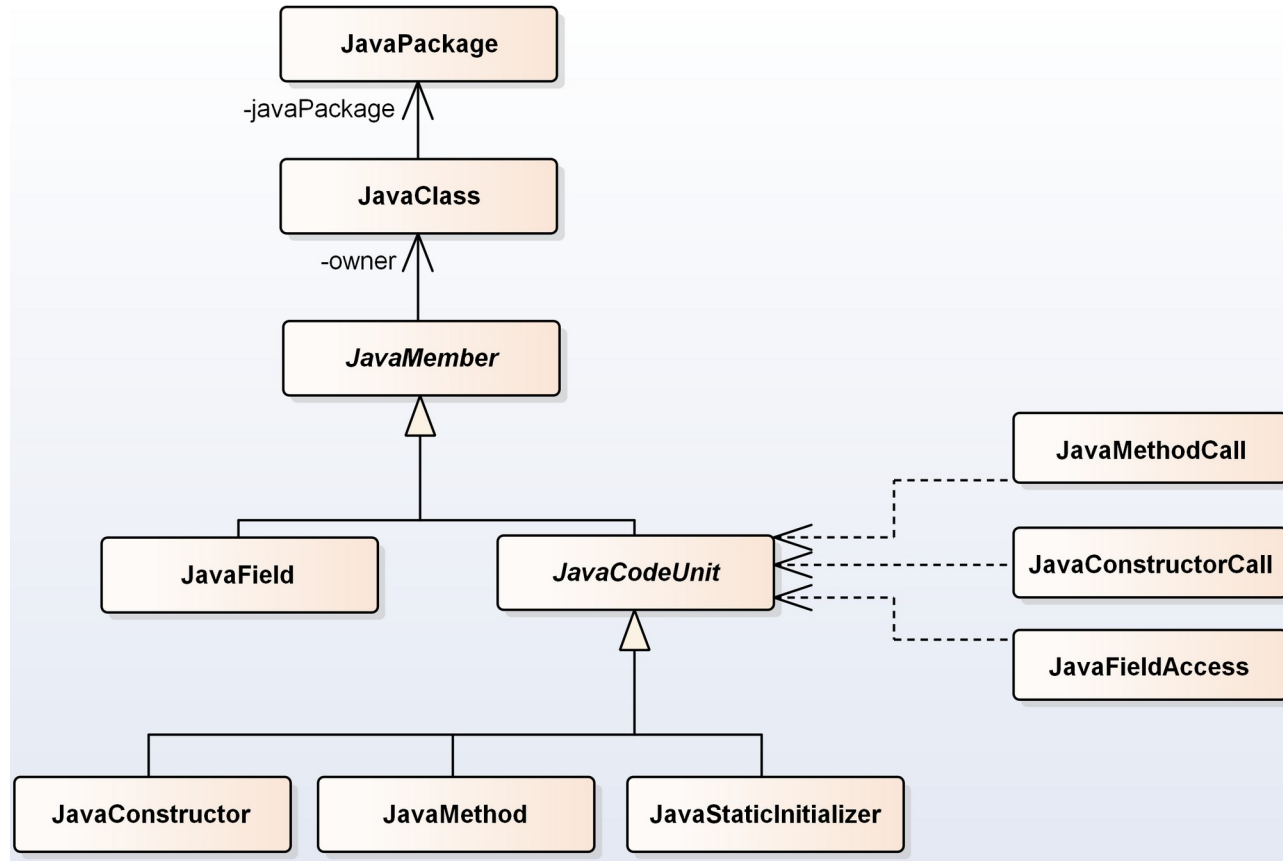
Java JDK

ArchUnit

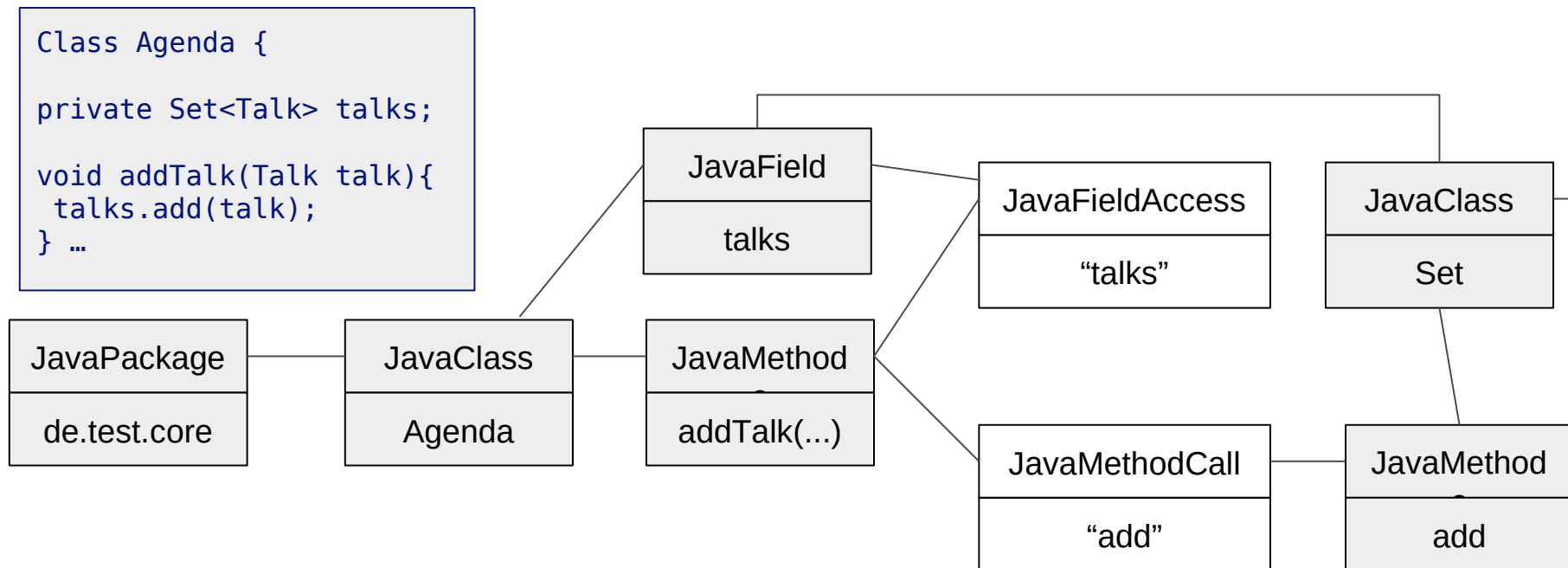


Datenmodell der Core API

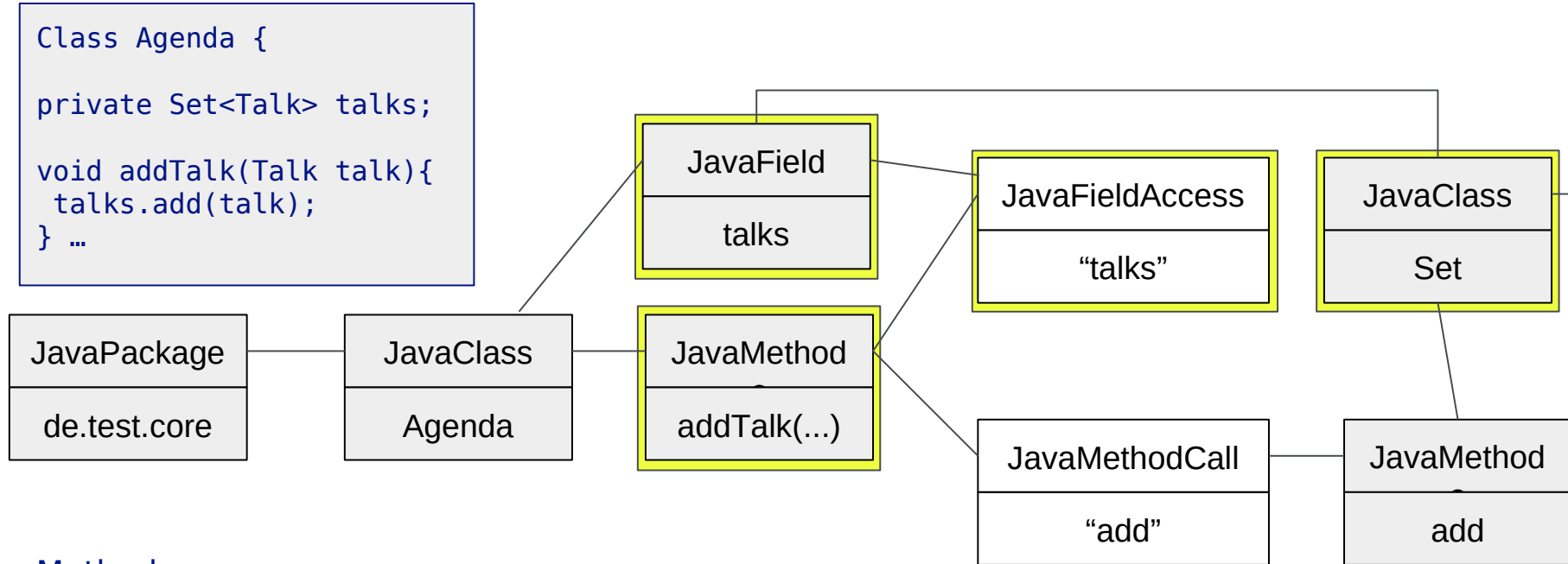
4	6.10.2022	JFN 2022
Advanced ArchUnit: Tests auf ByteCode-Analysen aufbauen		
Thomas Ruhroth und Kai Schmidt		



Object-Diagram als Beispiel



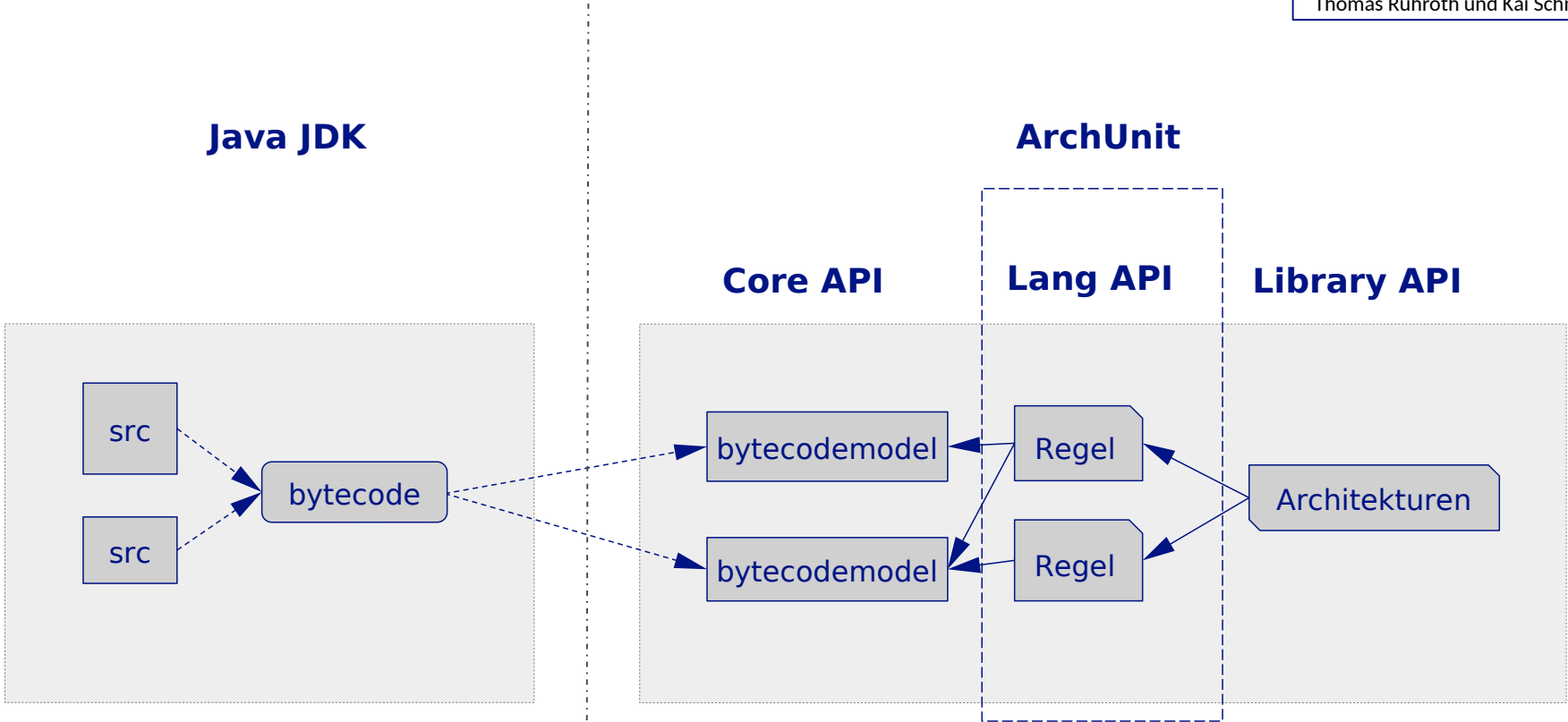
Object-Diagramm als Beispiel



Methoden,
deren Name mit add beginnt
sollen auf Variablen vom Typ Set zugreifen

Ein Blick in den Code

Wie arbeitet ArchUnit im Inneren



Lang API

```
classes()  
  .that().resideInAnyPackage("..test..")  
  .should().haveSimpleNameEndingWith("Test")
```

```
$STRUCTURE  
  that $PREDICATE  
  should $CONDITION
```

STRUCTURE

Einstiegspunkt: Füllung mit Annotation oder ClassFileImporter
Legt den zu prüfenden Typen fest <T>

PREDICATE

Filter

```
public interface Predicate<T> {  
    boolean test(T input);  
}
```

CONDITION

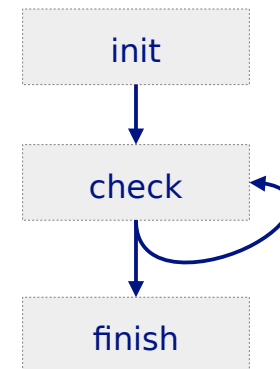
Implementierung der abstrakten Klasse ArchCondition, um
Prüfungen durchzuführen.

ArchCondition

Zentraler Anlaufpunkt für Erweiterungen

Wichtige Methoden:

```
public void init(java.lang.Iterable<T> allObjectsToTest)
public abstract void check(T item, ConditionEvents events)
public void finish(ConditionEvents events)
```



ConditionEvents sammelt die Analyse-Ergebnisse:

```
conditionEvents
.add(SimpleConditionEvent.violated(call.getOrigin(),
"CodeUnit " + call.getOrigin() + " calls deprecated " + target.getFullName()));
```

ArchCondition

10	6.10.2022	JFN 2022
Advanced ArchUnit: Tests auf ByteCode-Analysen aufbauen		
Thomas Ruhroth und Kai Schmidt		

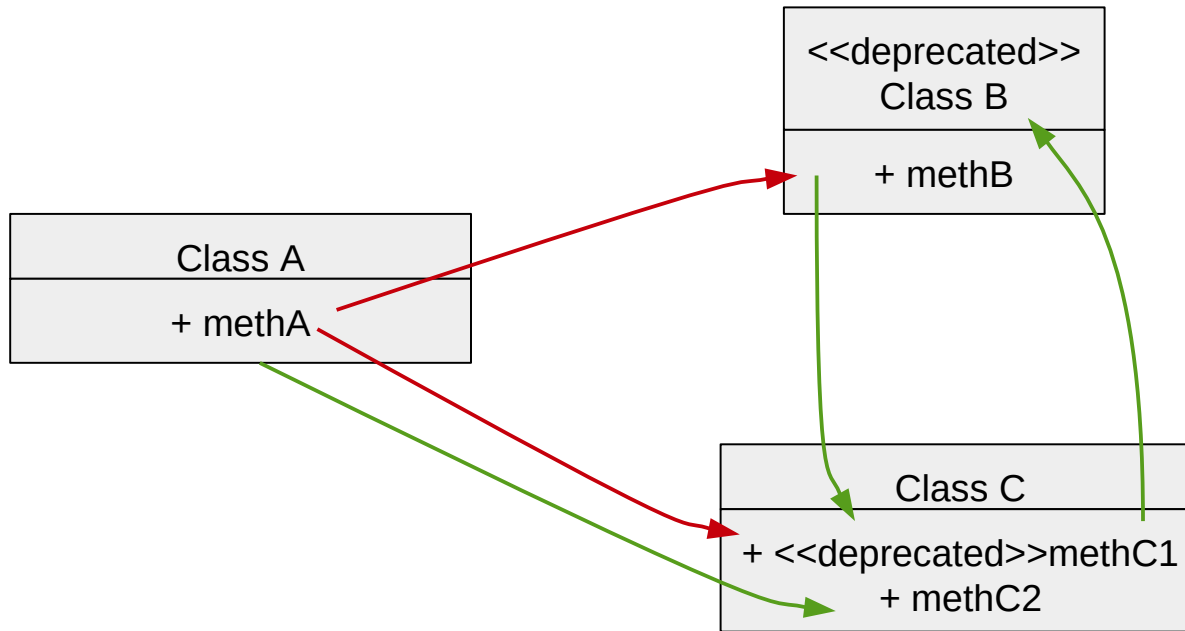
Implementation von ArchCondition erlaubt die Anwendung von eigenen Regeln in Should-Klauseln:

```
ArchCondition complyToOwnCondition = new OwnCondition();  
...should(complyToOwnCondition)...
```

Hinweis: Syntax-Unterschied beachten

```
...should().haveName(...)...
```

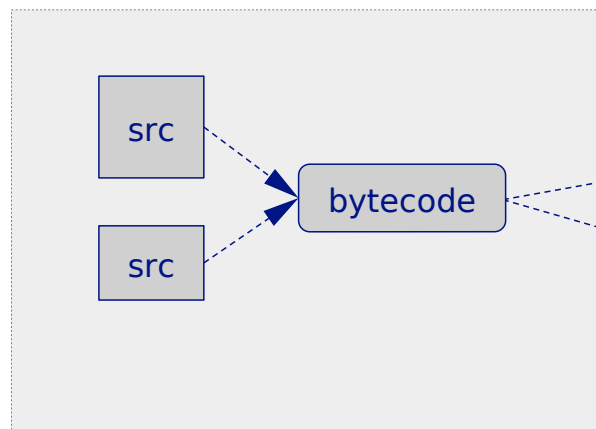

Erlaubte und nicht erlaubte Beziehungen



Ein Blick in den Code

Wie arbeitet ArchUnit im Inneren

Java JDK

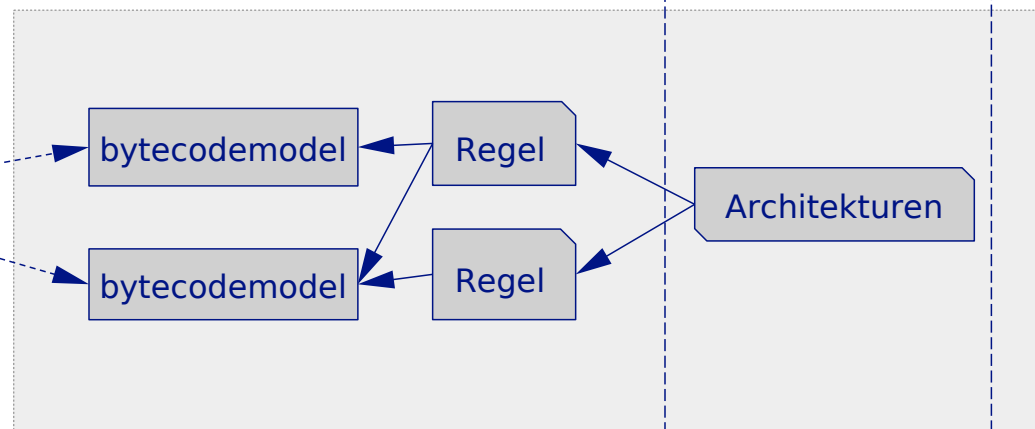


ArchUnit

Core API

Lang API

Library API



Library API

```
@ArchTest
public static final ArchRule configureDeprecatedOpCalls =
    MyArchTestLibrary.deprecationAwareArchitecture()
        .packages("deprecated.callees").areAllowedToBeCalled()
        .packages("deprecated.callers").areAllowedToUseDeprecated();
```

<<interface>> ArchRule

```
void check(JavaClasses classes);
ArchRule because(String reason);
EvaluationResult evaluate(JavaClasses classes);
```

DeprecationAwareArchitecture

PackageClassification packages(..)

PackageClassification

```
DeprecationAwareArchitecture areAllowedToBeCalled()
DeprecationAwareArchitecture areAllowedToUseDeprecated()
```

Ein Blick in den Code

Ausblick & Erfahrungen

16	6.10.2022	JFN 2022
Advanced ArchUnit: Tests auf ByteCode-Analysen aufbauen		
Thomas Ruhroth und Kai Schmidt		

Große Analysen recht performant,
wenn man weiß wie

Schnell abstrakte Prüfungen
(Beispiel Seiteneffektfreiheit)

Wissensquelle ArchUnit-Code

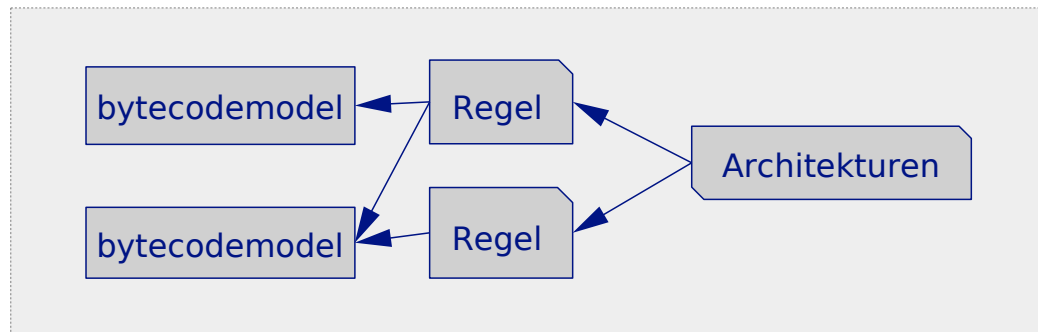
Bytecode hat Vor- und Nachteile

Zusammenfassung und Fragen

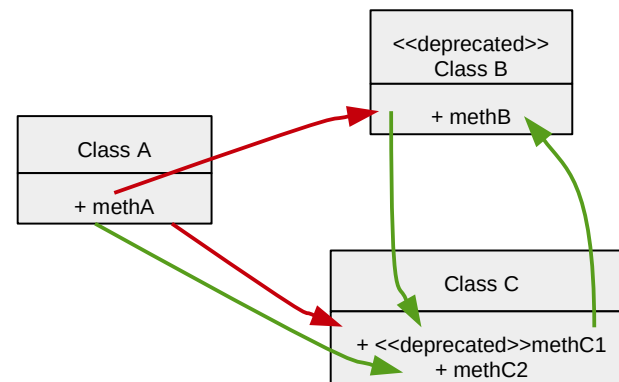
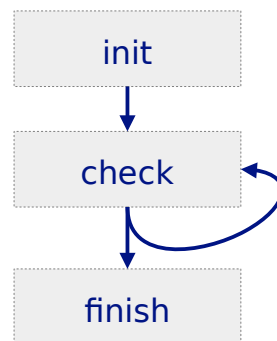
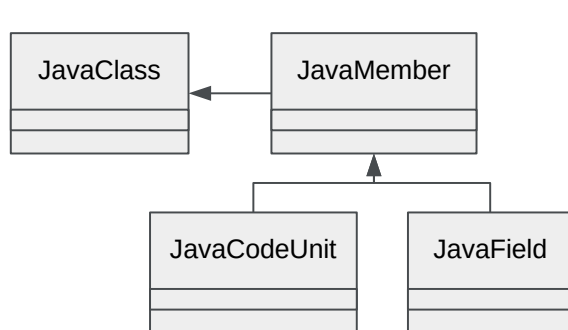
Core API

Lang API

Library API



Repository:
<https://github.com/electronickai/ArchUnit-Layers-Example>



Vielen Dank!
Bitte nicht weiterblättern