

Pecha-Kucha- Session

20 Jahre JUGS, 20 x JFS (Oliver Böhm)

REST API Dokumentation mit Swagger und AsciiDoc (Jan-Paul Buchwald)

Philosophy screws it all up! (Markus Harrer)

Mit Mutationstesten die Tests testen (Johannes Dienst)

Schall und Rauch (Michael Wiedeking)



20 Jahre JUGS,
20 x JFS

(Oliver Böhm)

20 Jahre JUGS

JAVAFORUM **20.** Jubiläum
1997-2017 *stuttgart*

Oliver Böhm

WRITE ONCE RUN ANYWHERE

2016



Service

2014

λ



mobileForum

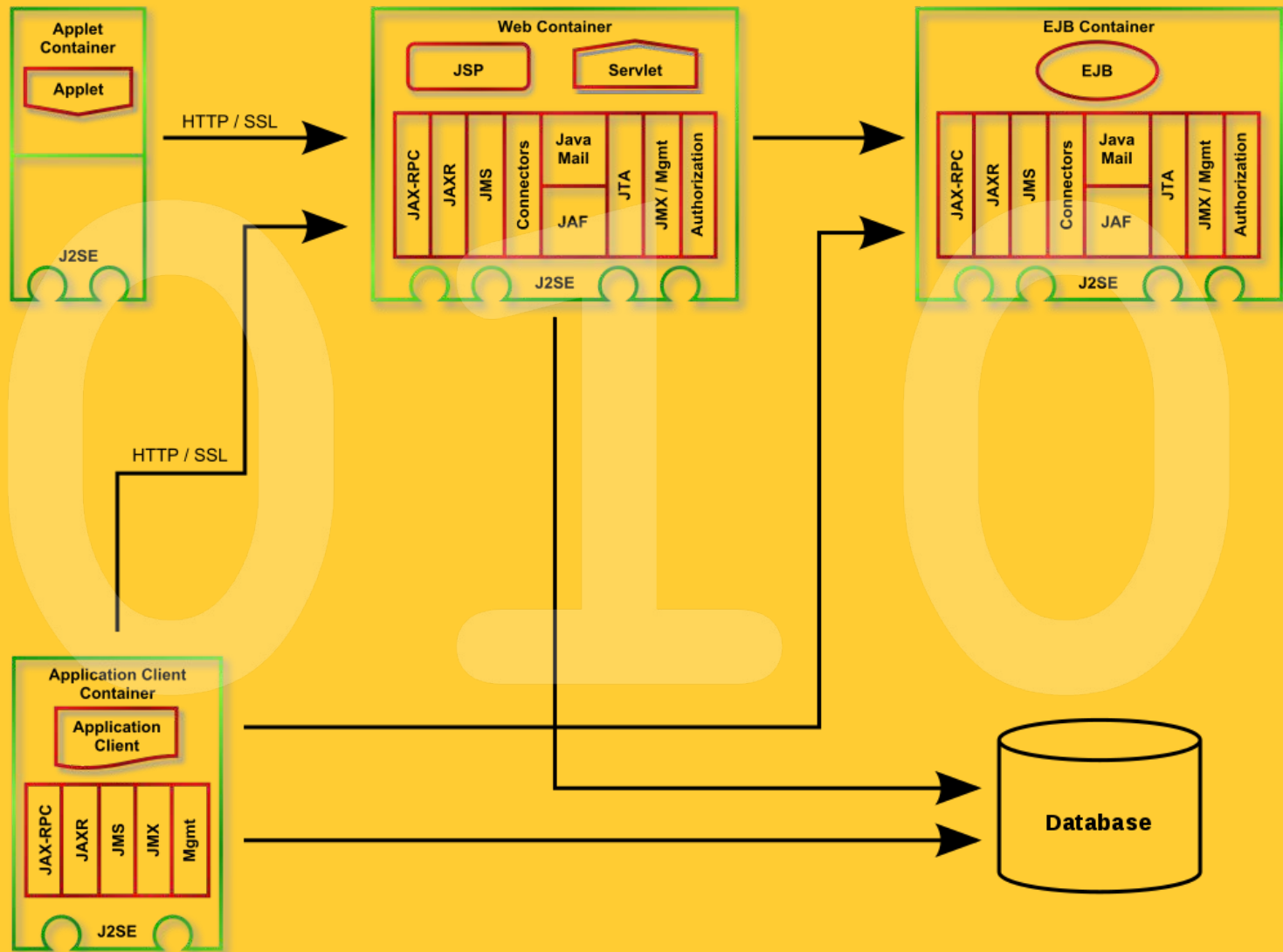
STUTTGART 2013

- 3 Tracks parallel
- ca. 200 TN



2017







web.archive.org/web/20100124044749/http://blogs.sun.com:80/jg

James Gosling: on the Java Road

« This Modern Age is... | Main | OK, it's time to... »

THURSDAY JANUARY 21, 2010

So long, old friend...



8 Fallacies bio Standards

BOOKS WORTH READING

- The Omnivore's Dilemma: A Natural History of Four Meals* by Michael Pollan
- The Botany of Desire: A Plant's-Eye View of the World* by Michael Pollan
- Newton's Clock: Chaos in the Solar System* by Ivars Peterson
- Guns, Germs, and Steel: The Fates of Human Societies* by Jared Diamond
- Collapse: How Societies Choose to Fail or Succeed* by Jared Diamond
- Freakonomics : A Rogue Economist Explores the Hidden Side of Everything* by Steven D. Levitt, Stephen J. Dubner
- The Discoverers: A History of Man's Search to Know His World and Himself* by Daniel J. Boorstin
- The Creators: A History of Heroes of the Imagination* by Daniel J. Boorstin

2



Java
inside

Kultur- & Kongresszentrum Liederhalle

10 Jahre JAVA FORUM 2007
STUTTGART

10 Jahre JUGS
VEREINIGTE JUGENDLICHE
STUTTGART

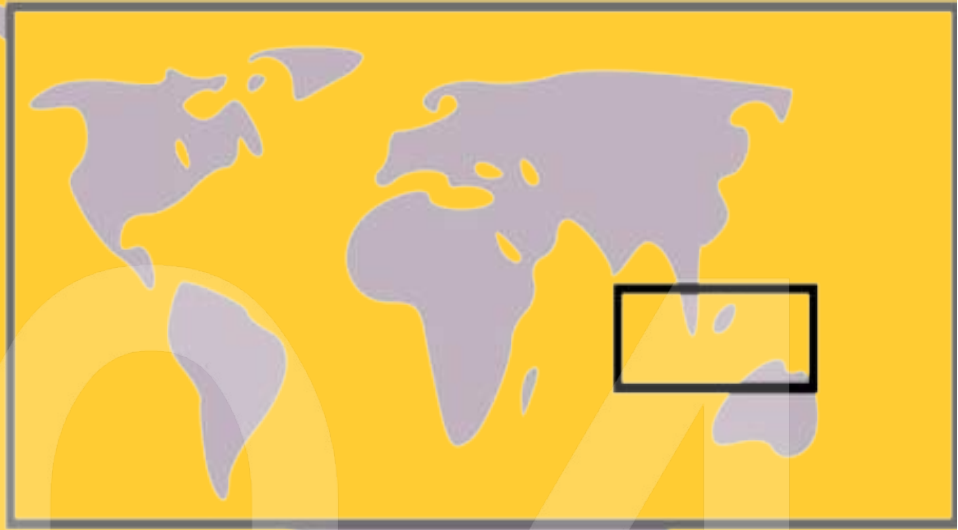
MEYER SAAL
SCHILLER SAAL
TAGUNGSRÄUME

Diese Bedienungsanweisung soll Ihnen zeigen, wie Sie mit dieser neuen, hervorragenden Maschine die höchstmögliche Leistung erzielen können. Machen Sie sich mit ihren Funktionen vertraut . . . und Sie werden schneller und leichter schreiben. Lesen Sie nachfolgende Aufstellung, und Sie werden bereits mit jedem einzelnen Bedienungselement vertrauter sein und die restlichen Seiten dieser Bedienungsanweisung leicht verstehen.



AspectJ Winter Camp





5



Java

o o o
spring

2002

mySAP

Weblogic

Websphere

Eclipse

.NET

20€1



Y2K
TESTED

OSGi



EJB

JAVA FORUM STUTTGART

300%



JAVA™



REST API Dokumentation mit Swagger und AsciiDoc

(Jan-Paul Buchwald)

REST API Dokumentation mit Swagger und AsciiDoc

Jan Paul Buchwald

YOPITER GmbH

Java Forum Stuttgart 2017, 06.07.2017

REST API Dokumentation



Source Code



Maschinenlesbar



Menschenlesbar

Swagger und AsciiDoc

JAX-RS + Swagger Annotationen



Docker run

/swagger.json



Zusätzliche **AsciiDoc** Dokumentation

swagger2markup und asciidoctor

JAX-RS + **Swagger** Annotationen



/swagger.json

Zusätzliche **AsciiDoc** Dokumentation



swagger2markup **Gradle** Plugin



Asciidoc-Dateistruktur



asciidoctor **Gradle** Plugin

HTML-API-Dokumentation



JAX-RS

```
@Path("/products")
public class ProductsResource {

    @Inject
    private DataHandler dataHandler;

    @GET
    public Response getAllProducts() {
        final List<Product> allProducts = dataHandler.loadProducts();
        return Response.ok(allProducts).build();
    }
}
```


JAX-RS Swagger Annotationen

```
dependencies {  
    ...  
    compile 'io.swagger:swagger-jaxrs:1.5.14'  
}
```

build.gradle

```
@GET  
@Produces(MediaType.APPLICATION_JSON)  
@ApiOperation(value = "Get a list of all available products", response =  
Product.class, responseContainer = "List")  
public Response getAllProducts(@ApiParam(value = "Cookie holding the JWT  
authentication token", required = true) @CookieParam("ACCESSTOKEN") final  
String accessToken) {  
    ...  
}
```

API Model Annotationen

```
@ApiModelProperty(description = "Attributes of a particular product.")
public class Product {
    ...
    @ApiModelProperty(value = "The unique product number", required = true)
    public int getNumber() {
        return number;
    }

    @ApiModelProperty(value = "The product code", required = true)
    public String getCode() {
        return name;
    }
}
```

Swagger Annotations Bean Config

```
@ApplicationPath("/")
public class ProductManagementApplication extends Application {
    ...
    public ProductManagementApplication() {
        final BeanConfig beanConfig = new BeanConfig();
        beanConfig.setVersion("1.0");
        beanConfig.setTitle("Product Management API");
        beanConfig.setHost("yopiter.com");
        beanConfig.setSchemes(new String[] {"https"});
        beanConfig.setBasePath("/pms");
        beanConfig.setResourcePackage("com.yopiter.sample.resources");
        beanConfig.setPrettyPrint(true);
        beanConfig.setScan();
    }
}
```

Gradle

- Build System für die JVM
- Groovy DSL
- Vorteile gegenüber Maven
 - Performance
 - Dependency Management
 - ...



```
apply plugin: 'war'

repositories {
    mavenCentral()
}

dependencies {
    providedCompile 'javax:javaee-api:7.0'
}
```

build.gradle

Docker Build

```
FROM jboss/wildfly
```

```
EXPOSE 8080
```

```
ADD build/libs/pms.war /opt/jboss/wildfly/standalone/deployments
```

Dockerfile

```
$ gradle build
```

```
$ docker build --tag=pms-wildfly .
```

```
$ docker run --name pms -p 8100:8080 -d pms-wildfly
```


Swagger JSON

```
localhost:8100/pms/swagger.json

{"swagger":"2.0","info":{"version":"1.0","title":"Product Management API"},"host":"yopiter.com","basePath":"/pms","schemes":["https"],"paths":{"products":{"get":{"summary":"Get a list of all available products","description":"","operationId":"getAllProducts","produces":["application/json"],"parameters":[{"name":"ACCESS_TOKEN","in":"cookie","description":"Cookie holding the JWT authentication token","required":true,"type":"string"}],"responses":{"200":{"description":"successful operation","schema":{"type":"array","items":{"$ref":"#/definitions/Product"}}},"500":{"description":"An unexpected internal error occurred"}}}}},"definitions":{"Product":{"type":"object","required":["code","number"],"properties":{"number":{"type":"integer","format":"int32","description":"The unique product number"},"code":{"type":"string","description":"The product code"},"description":{"description":"Attributes of a particular product."}}}}}
```

AsciiDoc

```
== Products REST API
This document describes the
*JSF2017* sample products API:
```

```
.Content
```

- * Data Model
- * Resources
- * Error Handling

```
.Code Sample
```

```
----
GET /products
----
```

Products REST API

This document describes the **JSF2017** sample products API:

Content

- Data Model
- Resources
- Error Handling

Code Sample

```
GET /products
```

Atom Editor AsciiDoc Preview

Gradle Plugin Setup

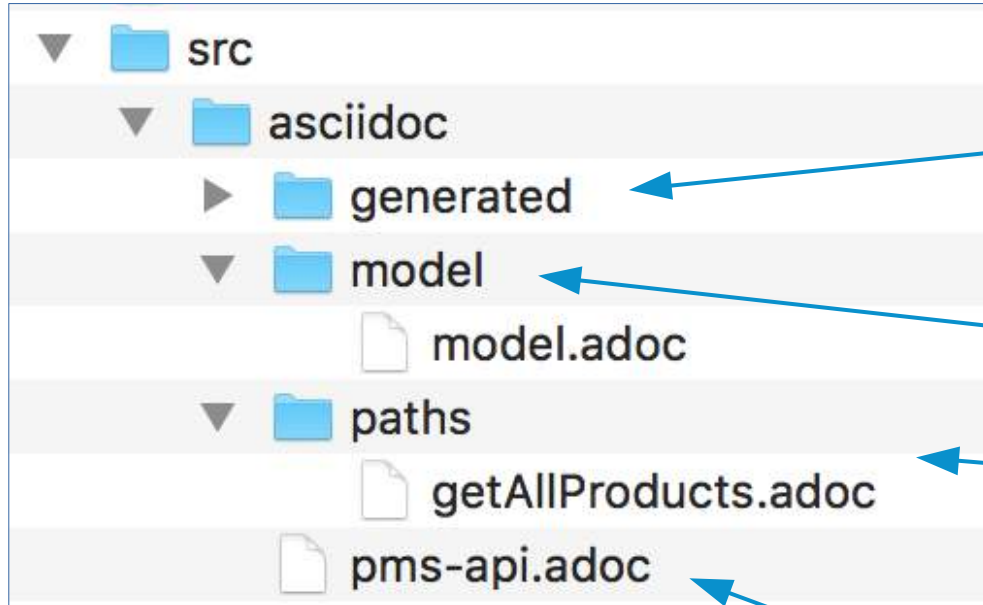
```
buildscript {
    repositories {
        mavenCentral()
        jcenter()
    }

    dependencies {
        classpath 'io.github.swagger2markup:swagger2markup-gradle-plugin:1.3.1'
        classpath 'io.github.swagger2markup:swagger2markup-import-files-ext:1.3.1'
        classpath 'org.asciidoctor:asciidoctor-gradle-plugin:1.5.3'
    }
}

apply plugin: 'io.github.swagger2markup'
apply plugin: 'org.asciidoctor.convert'
```

build.gradle
e

AsciiDoc-Ordnerstruktur



Später aus Swagger generiert

Zusätzliche, freie Dokumentation

Beschreibung pro Pfad

Basis-Dokument

AsciiDoc: paths

```
Loads all available products from the product management
system's database.
```

```
*Note:* This is a very expensive operation that should only be
called for administrative purposes.
```

```
getAllProducts.adoc
```



Dateiname = Swagger Operation Id
(default: Java-Methodenname)

Alternativ: feingranulare Dokumente in Unterverzeichnis

```
paths/getAllProducts/operation-begin-description.adoc
```

```
paths/getAllProducts/operation-end-example1.adoc
```

```
paths/getAllProducts/operation-end-example2.adoc
```


AsciiDoc: Basis-Dokument

```
include::generated/overview.adoc[]  
include::model/model.adoc[]  
include::generated/paths.adoc[]  
include::generated/definitions.adoc[]  
pms-api.adoc
```

Reihenfolge und Includes frei gestaltbar,
Namen unter `generated` fest vorgegeben

Gradle Tasks

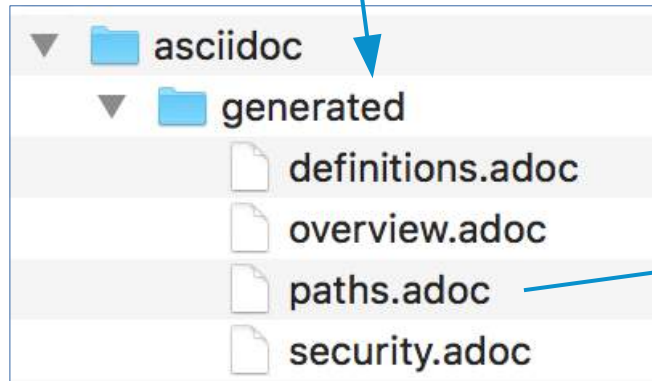
```
convertSwagger2markup {  
    swaggerInput 'http://localhost:8100/pms/swagger.json'  
    outputDir file('src/asciidoc/generated')  
    config = ['swagger2markup.markupLanguage' : 'ASCIIDOC',  
            'swagger2markup.extensions.dynamicPaths.contentPath' :  
            'src/asciidoc/paths' ]  
}
```

```
asciidocctor {  
    sourceDir 'src/asciidoc'  
    Sources { include 'pms-api.adoc' }  
    backends = ['html5']  
    attributes = [ doctype: 'book',  
                toc: 'left',  
                generated: 'src/asciidoc/generated' ]  
}
```

build.gradle

Generierter AsciiDoc Output

```
$ gradle convertSwagger2Markup
```



```
[[_paths]]
== Paths

[[_getallproducts]]
=== Get a list of all available products
....
GET /products
....

==== Responses

[options="header", cols=".^2, .^14, .^4"]
|===
|HTTP Code|Description|Schema
|**200**|successful operation|< <<_product,Product>> > array
|**500**|An unexpected internal error occurred|No Content
|===

==== Produces

* `application/json`
```

Ergebnis

```
$ gradle asciidoctor
```

The screenshot shows a web browser displaying the Swagger API documentation for 'Product Management API'. The browser's address bar shows the file path: `file:///Users/jpb/Documents/Projekte/SwaggerAsciiDocSample/build/asciidoc/html5/pms-api.html`. The page layout includes a left-hand navigation menu and a main content area.

Table of Contents

- Overview
- Version information
- URI scheme
- Background Information
- Example
- Paths
- Get a list of all available products
- Definitions
- Product

Product Management API

Overview

Version information

Version : 1.0

URI scheme

Host : yopiter.com
BasePath : /pms
Schemes : HTTPS

Background Information

The input and output objects of the Product Management API ...

Example

```
{  
  "number": 42,  
  "code": "KNO-42"  
}
```

Paths

Get a list of all available products

GET /products

Responses

HTTP Code	Description	Schema
200	successful operation	< Product > array
500	An unexpected internal error occurred	No Content

Produces

- application/json

Definitions

Product

Vielen Dank!

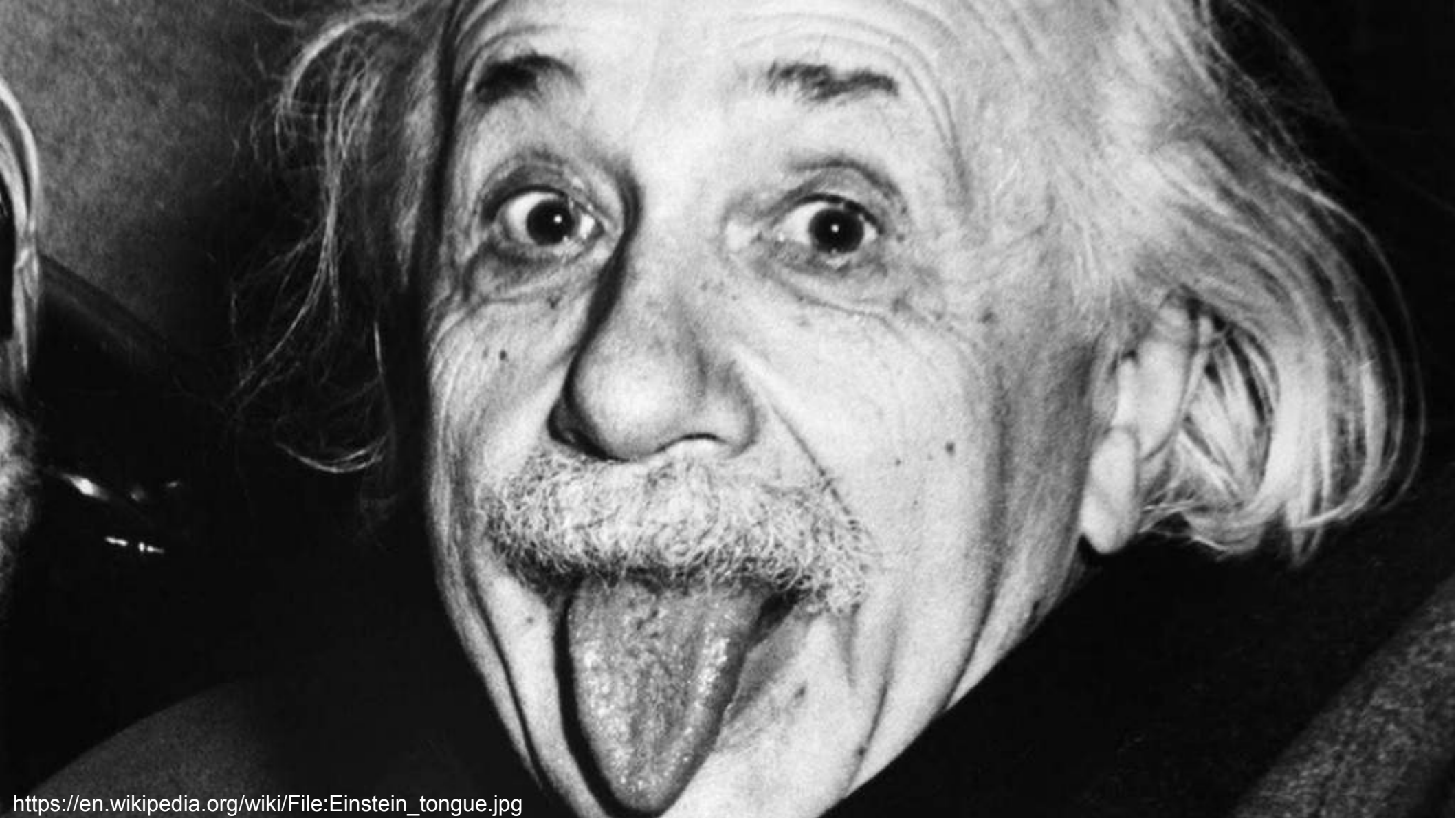
<http://swagger2markup.github.io/swagger2markup/1.3.1/>

<http://asciidoctor.org/docs/asciidoctor-gradle-plugin/>



Philosophy screws
it all up!

(Markus Harrer)



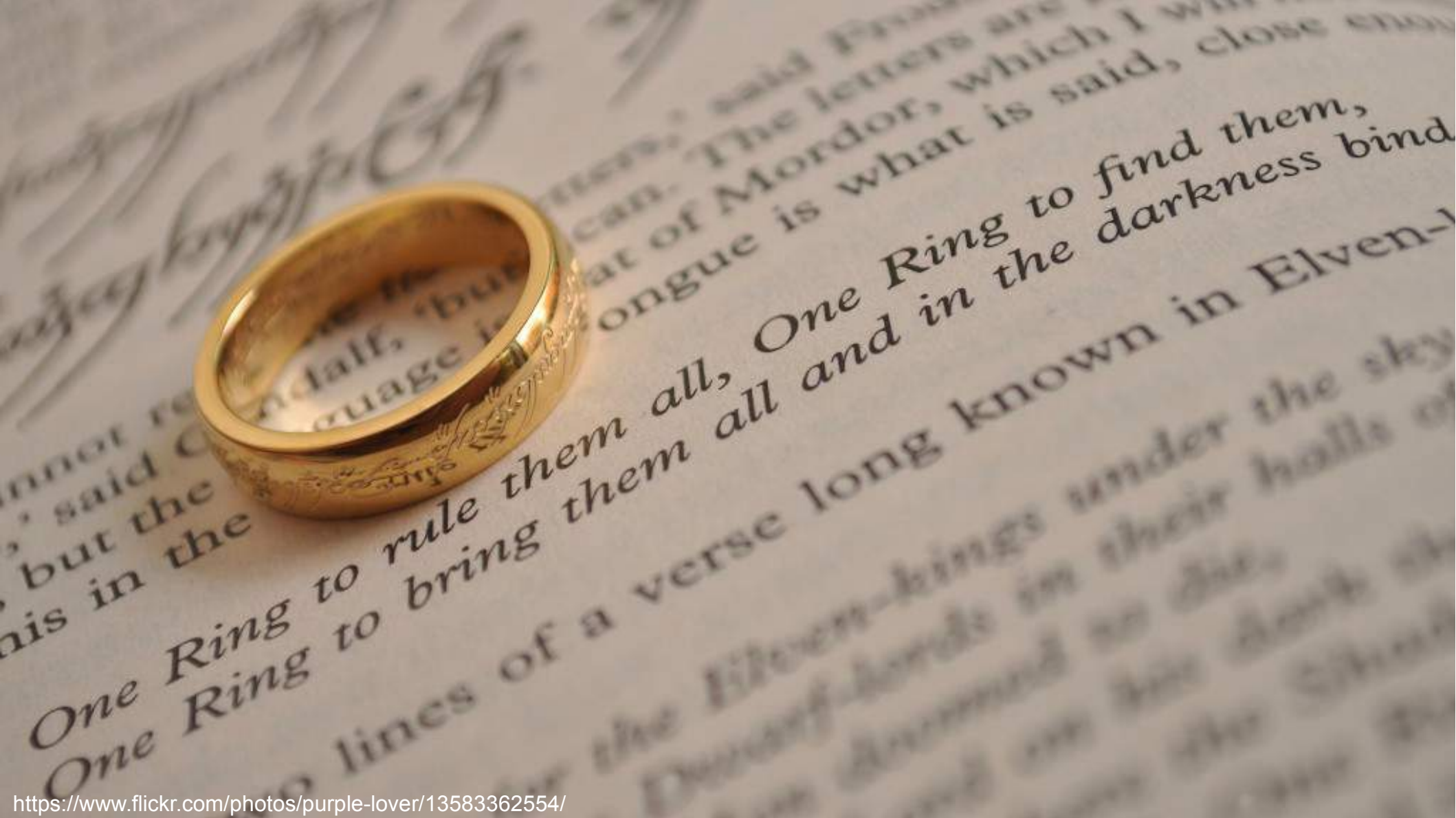
















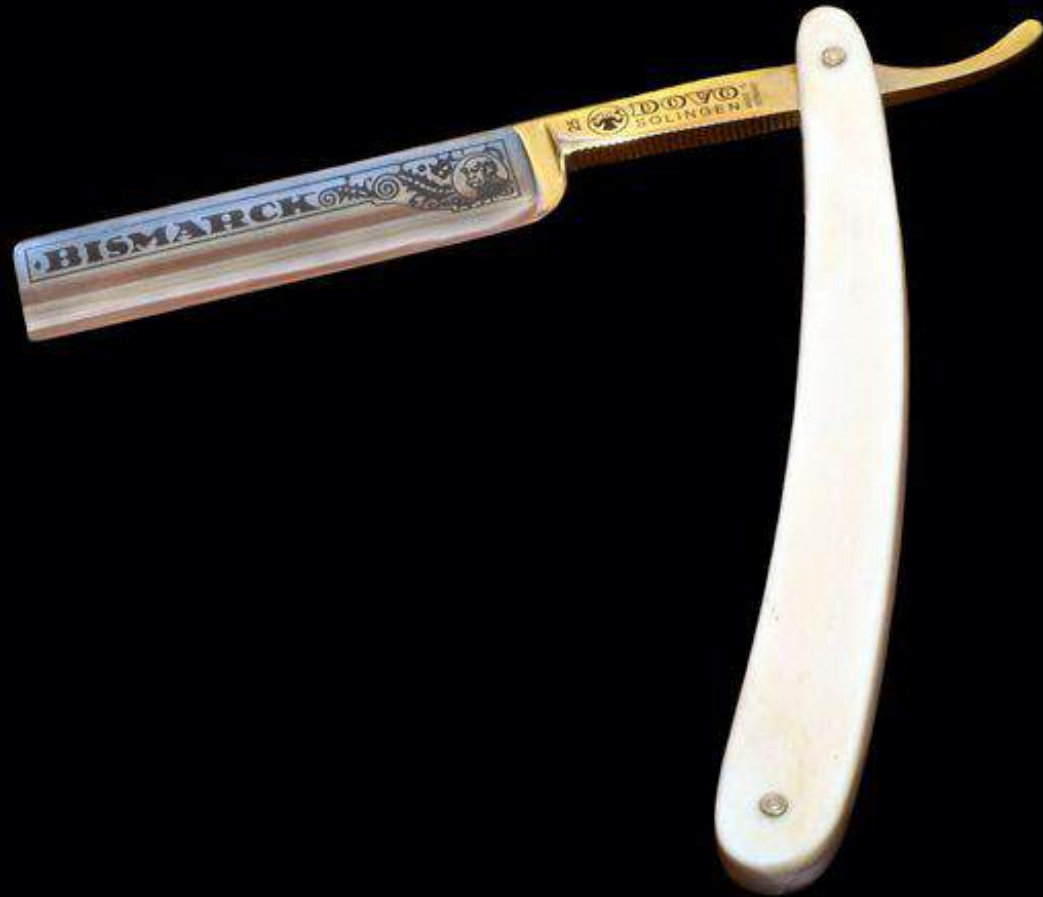
















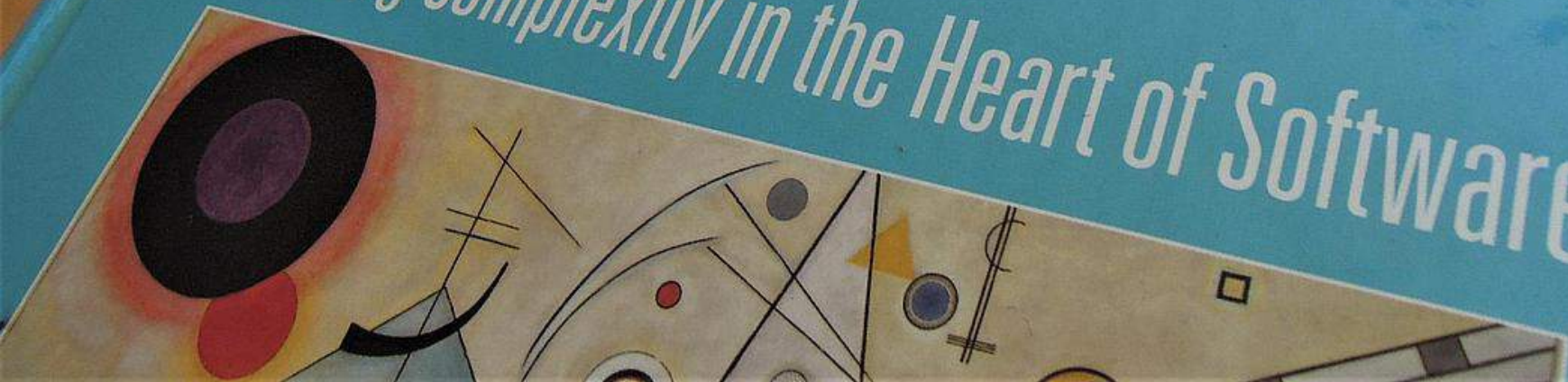
e*xtreme
Programming
e*xp*lain*ed

EMBRACE CHANGE

Domain-Driven

DESIGN

Tackling Complexity in the Heart of Software







Mit Mutationstesten die Tests testen

(Johannes Dienst)

MIT MUTATIONSTESTEN DIE TESTS TESTEN



@JohannesDienst



TESTSUITE! QUALITÄT?

CODE-REVIEWS...



TEUER

FEHLERANFÄLLIG

A close-up photograph of a man's head and neck being shaved by a barber. The barber's hand is visible, holding a safety razor against the man's skin. Another hand, wearing a black watch, rests on the man's shoulder. The scene is dimly lit, focusing on the action of shaving.

SEHR

EFFEKTIV

A photograph of a railway station platform with tracks and overhead power lines, shrouded in a thick fog. The tracks lead into the distance, and the fog obscures the background. The scene is dimly lit, with some light reflecting off the wet tracks.

AUTOMATISIERUNG?

MUTATIONS TESTEN





i++ → i--

== → ! =



GETÖTET? ÜBERLEBT?



GETÖTET!



ÜBERLEBT!





VIELE...



SEHR VIELE...



MUTANTEN

ANALYSE



**Wer will fleißige
Unit-Tests sehen,
der muss ihnen
Mutanten geben...**

A photograph of a gravestone in a cemetery. The gravestone is a simple, rectangular, light-colored stone, partially covered with fallen yellow and orange leaves. The surrounding area is a lush green lawn, also heavily scattered with fallen autumn leaves. The overall scene is a somber and quiet representation of the end of life.

RIP

CODE



**UNGETESTETE
SEITENEFFEKTE**

ECHTE TESTLÜCKEN

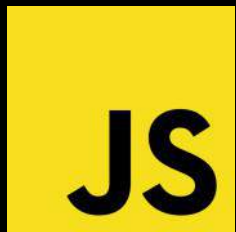


TESTSUITE



TESTSUITE-REVIEWS





Major

Stryker

Humbug

Heckle

MuCheck

Pitest

Mutant





Schall und Rauch

(Michael Wiedeking)

$$\vec{v} := \vec{v}_0 + g \cdot \Delta t^2 \text{ where } g := [-9.81 \ 0 \ 0] \text{ m/s}^2$$

$$\binom{n}{k} := \frac{n!}{(n-k)! k!}$$

isStraight := (hand : Set [[Card]]) ↦

∃(card ∈ hand)

∀(c ∈ hand, c ≠ card)

∃(p ∈ hand)

(succ(p.rank) = c.rank)

isStraight := (hand : Set [[Card]]) ↦

∃(set ∈ $\wp_4(\text{hand})$) sameRank(set) where

sameRank: (s) ↦ ∀(c ∈ s where r := arb(s)) (c.rank = r)

Schall und Rauch

Über die Unmöglichkeit, den richtigen Namen zu finden

$M := (7F\dots7F)_{16}$

$r \leftarrow \text{bitwiseAnd}(x, M)$

$r \leftarrow \text{addUnsignedUnchecked}(r, M)$

$r \leftarrow \text{bitwiseOr}(r, x)$

$r \leftarrow \text{bitwiseOr}(r, M)$

$r \leftarrow \text{bitwiseNot}(r)$

$k \leftarrow \text{numberOfTrailingZeroBits}(r)$

$k \leftarrow \text{shiftRight}(k, 3)$

return k

```
static int abs(int x) {  
    int y = x >> 31;  
    return (x ^ y) - y;  
}
```

function $\text{abs}(x : \mathbb{I}_k) \rightarrow \mathbb{I}_k$ **is**

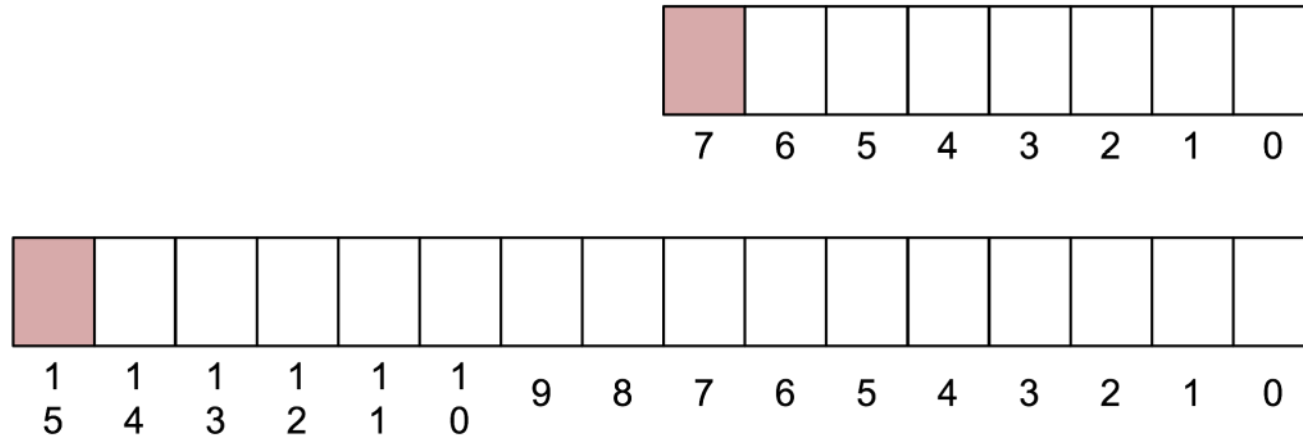
$s \leftarrow \text{signMask}(x)$

$x \leftarrow \text{bitwiseXor}(x, s)$

$y \leftarrow \text{subtractUnchecked}(x, s)$

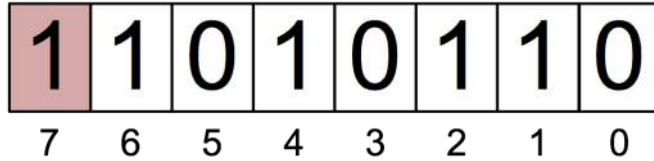
return y

end



Eine Bibliothek soll [...] unabhängig von der Breite der zur Verfügung stehenden Maschinenwörter den Zugriff auf einzelne Bits oder Bit-Sequenzen ermöglichen.

Welche Funktionsnamen passen am besten, wenn man auf die Bits von Wörtern zugreifen will?

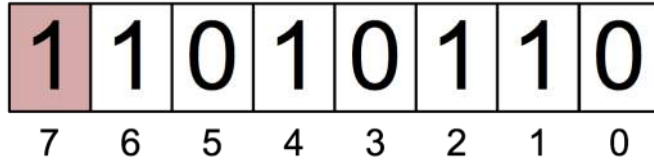


Mit Hilfe einer Funktion möchte man Zugriff auf das Vorzeichen – also das höchstwertigste Bit – eines ganzzahligen (vorzeichenbehafteten) Werts erhalten

byte extractSignBit(**byte** x)

Welches Bit-Muster wird diese Funktion vermutlich für das Byte mit dem Bit-Muster $(1101\ 0110)_2$, bei dem ja das Vorzeichen gesetzt ist, liefern?

- $(1000\ 0000)_2$
- $(0000\ 0001)_2$
- Weder noch; das liefert ...



Mit Hilfe einer Funktion möchte man Zugriff auf das Vorzeichen – also das höchstwertigste Bit – eines ganzzahligen (vorzeichenbehafteten) Werts erhalten

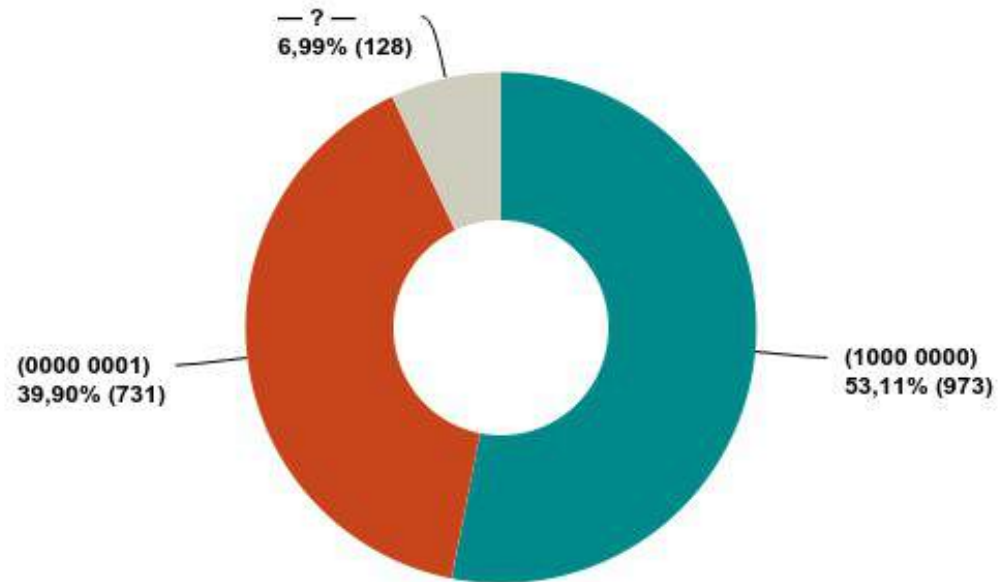
byte extractSignBit(**byte** x)

Welches Bit-Muster wird diese Funktion vermutlich für das Byte mit dem Bit-Muster $(1101\ 0110)_2$, bei dem ja das Vorzeichen gesetzt ist, liefern?

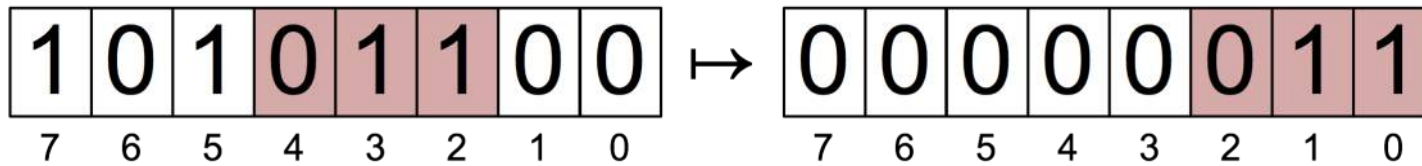
- $(1000\ 0000)_2$
- $(0000\ 0001)_2$
- Weder noch; das liefert ...

Q1 extractSign(1101 0110)

Beantwortet: 1.832 Übersprungen: 0



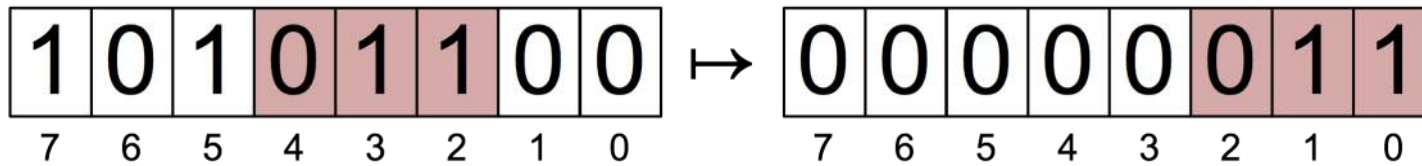
Ungleich Null Zurück Hängt Seiteneffekt MSB Endianness
Wäre 0xff Wert mask SignBit Bool
Beide Varianten Byte Abhängig True Boolean
Extract Handelt Gesetz somit Möglich
ob Little



Aus dem Byte mit dem Bit-Muster $(1010\ 1100)_2$ soll die aus den drei zusammenhängenden Bits bestehende Sequenz von der Position 2 bis Position 4 gelesen werden. Hat das niederwertigste Bit die Position 0 und das höchstwertigste die Position 7, so soll das Ergebnis $(0000\ 0011)_2$, also dezimal 3, ergeben.

Welche Funktion wird man vermutlich dazu benutzen müssen?

- `extractBitField(...)`
- `isolateBitField(...)`
- `getBitField(...)`
- Noch besser wäre aber ...



Aus dem Byte mit dem Bit-Muster $(1010\ 1100)_2$ soll die aus den drei zusammenhängenden Bits bestehende Sequenz von der Position 2 bis Position 4 gelesen werden. Hat das niederwertigste Bit die Position 0 und das höchstwertigste die Position 7, so soll das Ergebnis $(0000\ 0011)_2$, also dezimal 3, ergeben.

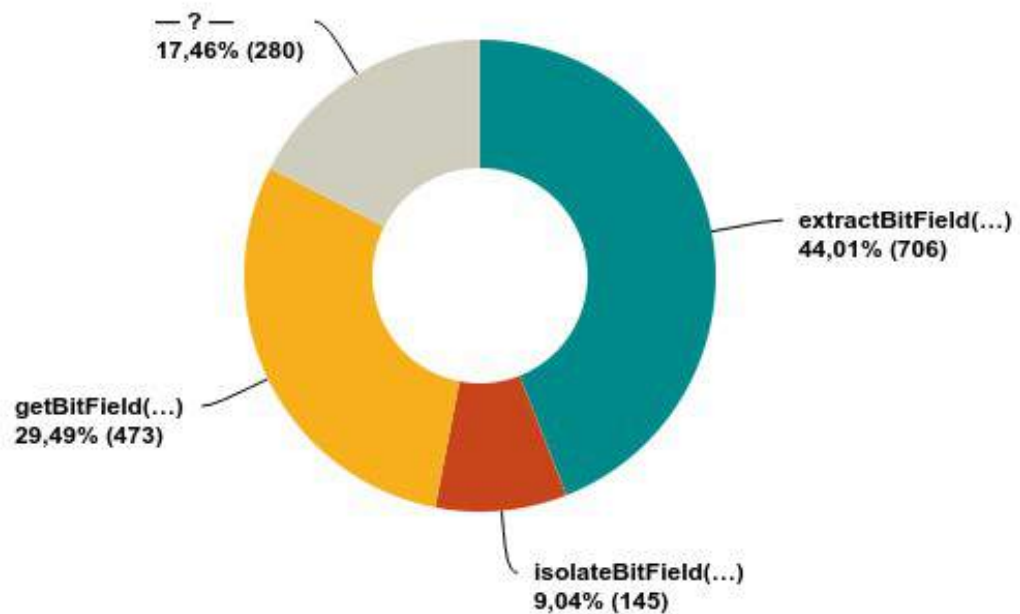
Welche Funktion wird man vermutlich dazu benutzen müssen?

- `extractBitField(...)`
- `isolateBitField(...)`
- `getBitField(...)`
- Noch besser wäre aber ...

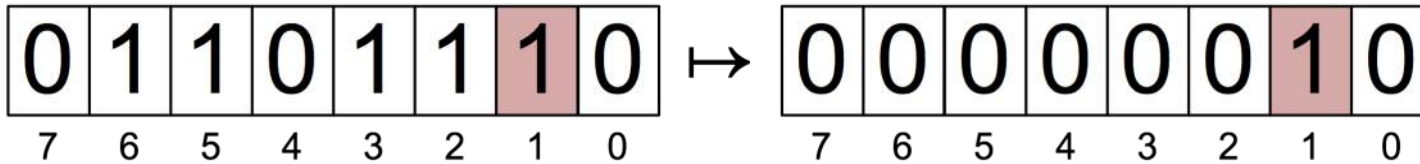
Q2

`xxxBitField(1010 1100, 2, 4) = (0000 0011)`

Beantwortet: 1.604 Übersprungen: 228



getSubBitField Mask subBits startPos
extractBitRange Getbits2to4 Int Besser
getBits bitSlice
extractBitSequence subBitField
Byte Start Funktion getBitSlice
getBitRange getBitFieldValue
getBitSequence Unklar sliceBitField
Substring extractBits Denn readBitField



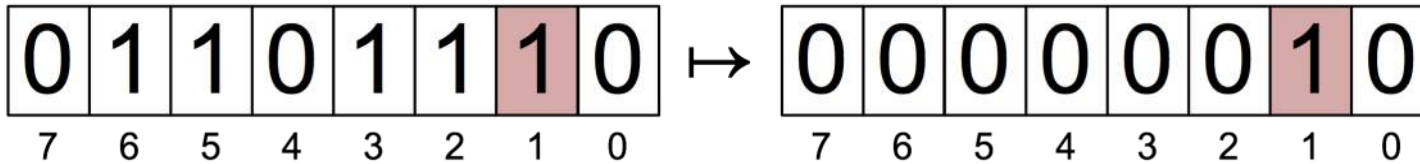
Mit Hilfe der Funktion

byte xxxLowestOneBit(**byte** x)

sollen alle bis auf das niederwertigste, gesetzte Bit „gelöscht“ werden. So soll etwa die Funktion bei Eingabe des Bytes mit dem Bit-Muster $(0110\ 1110)_2$ das Byte mit dem Bit-Muster $(0000\ 0010)_2$ liefern.

Wie sollte diese Funktion korrekt heißen?

- extractLowestOneBit
- isolateLowestOneBit
- getLowestOneBit
- Am besten wäre aber ...



Mit Hilfe der Funktion

byte xxxLowestOneBit(**byte** x)

sollen alle bis auf das niederwertigste, gesetzte Bit „gelöscht“ werden. So soll etwa die Funktion bei Eingabe des Bytes mit dem Bit-Muster $(0110\ 1110)_2$ das Byte mit dem Bit-Muster $(0000\ 0010)_2$ liefern.

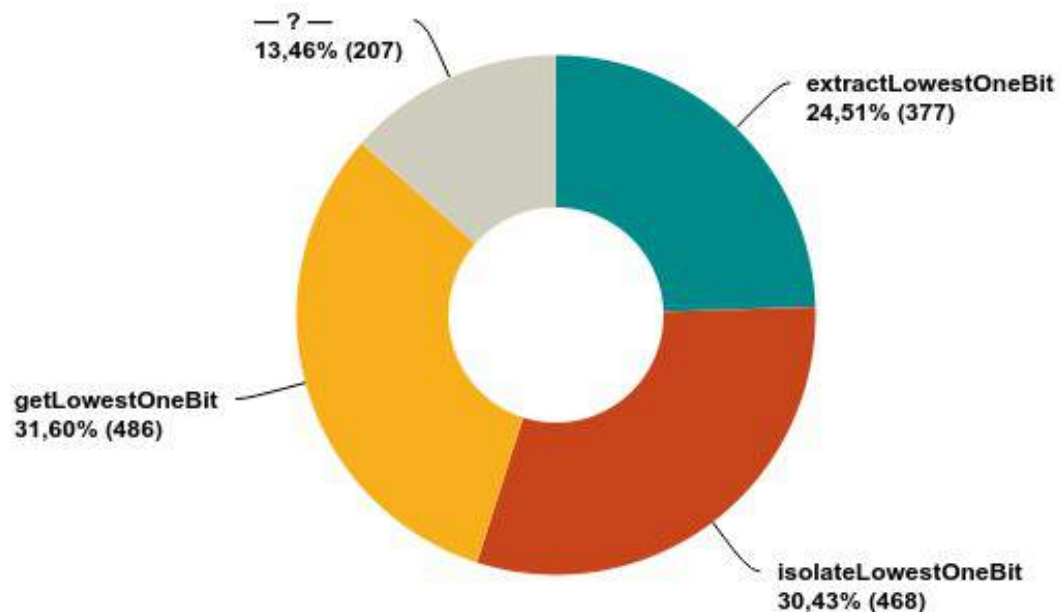
Wie sollte diese Funktion korrekt heißen?

- `extractLowestOneBit`
- `isolateLowestOneBit`
- `getLowestOneBit`
- Am besten wäre aber ...

Q3

`xxxLowestOneBit(0110 1110) = (0000 0010)`

Beantwortet: 1.538 Übersprungen: 294



Liefert Nummer **keepOnlyLowestOneBit**

reduceToLowestOneBit **extractLowestSetBit**

extractLowestBit **isolateLowestSetBit**

MaskLowestSetBit **findLowestOneBit**

filterLowestOneBit **Funktion**

eraseLowestOneBit

maskLowestOneBit

deleteLowestOneBit **Byte**

extractLeastSignificantBit

keepLowestOneBit keepLowestSetBit

getLowestSetBit resetToLowestOneBit

retainLowestOneBit getLowestOneBitMask

clearAllButLowestOneBit Gegensatz

getLeastSignificantBit Nur **lowestOneBit**

Set

1 1 0 1 0 1 1 0

1 1 0 1 0 1 1 0

1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1

0 1 1 0 1 1 1 0

1 0 1 0 1 1 0 0

0 0 0 0 0 0 1 0

0 0 0 0 0 0 1 1

55.88%

Sue runs around the track twice, and John runs around the track four times more than Sue.

How many times does John run around the track?

Sue runs around the track **twice**, and John runs around the track **four times more** than Sue.

How many times does John run around the track?

- $2 \times 4 = 8$
- $2 + 4 = 6$
- $2 + (2 \times 4) = 10$

Fazit

im Grunde kann man doch alles benennen wie man will
(allerdings sollte man dabei konsistent bleiben)

Leider: Keine Zeit für Fragen!

Vielen Dank!

Aber so viel Zeit muss sein:
Das ... Bit

niedrigwertige	niederwertige	höherwertige		
<i>niedrigwertigere</i>	<i>niederwertigere</i>	<i>höherwertigere</i>		
niedrigstwertige	niederwertigste	<i>höherwertigste</i>	höchstwertige	
	<i>niedrigstwertigste</i>	<i>höchstwertigste</i>		
	<i>einzigste</i>	<i>alleinste</i>	<i>bestbezahlteste</i>	<i>aktuellste</i>

???



20 Jahre JUGS, 20 x JFS
(Oliver Böhm)



REST API Dokumentation mit Swagger und AsciiDoc
(Jan-Paul Buchwald)



Philosophy screws it all up!
(Markus Harrer)



Mit Mutationstesten die Tests testen
(Johannes Dienst)



Schall und Rauch
(Michael Wiedeking)