



Legacy Code meistern in x einfachen Schritten

und testgetrieben weiterentwickeln

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0



Abstract (Teil 1)



In einer idealen Welt würden wir nur "neuen" Code schreiben, der natürlich perfekt und wunderschön ist. Wir müssten nie wieder unseren Code anschauen, geschweige denn 10 Jahre alte Projekte warten. Ende des Tagtraums ... Leider ist unsere Welt nicht so ideal, unser Code von gestern ist heute schon Legacy. Diesen im Nachhinein zu verstehen, zu erweitern oder darin Fehler zu beheben, ist immer eine Herausforderung, insbesondere wenn Tests fehlen.

Trotzdem gibt es einfache Möglichkeiten, wie man die Qualität von Legacy-Code verbessern kann. Das Wichtigste ist das Einziehen von Fangnetzen, sodass man trotz fehlender Tests guten Gewissens Änderungen durchführen kann. Wer Golden Master, Subclass to Test und Extract Pure Functions an konkreten Beispielen kennenlernen möchte, ist in dieser Session genau richtig.

Abstract (Teil 2)



Orientation in Objects

Bestandsanwendungen müssen gewartet und ggf. weiterentwickelt werden, bergen aber meist viele Defekte. Als Entwickler fürchten wir uns zudem, mehr Schaden anzurichten, weil das Verständnis für den Legacy Code fehlt. Refactoring kann zum Verstehen beitragen, endet aber aufgrund der typischerweise fehlenden automatisierten Tests in einem Blindflug.

Mit testgetriebener Entwicklung scheint es eine Allzweckwaffe für gutes Design und eine geringe Fehlrate zu geben. Aber TDD und Legacy Code scheinen sich eigentlich auszuschließen. Anhand von Live Coding schauen wir, wie die testgetriebene Entwicklung trotzdem helfen kann, den Code ohne allzu große Bauchschmerzen anzupassen. Schöner Nebeneffekt wird das Entstehen eines automatisierten Testbetts sein, welches zukünftige Refactorings leichter machen wird.

Über mich



Falk Sippach (@sppsack)

Trainer, Berater, Entwickler



Co-Organisator

Schwerpunkte
Architektur
Agile Softwareentwicklung
Codequalität



Java, XML und Open Source seit 1998



Java und XML

) Software Factory)

- Schlüsselfertige Realisierung von Java Software
- Individualsoftware
- Pilot- und Migrationsprojekte
- Sanierung von Software
- Software Wartung

) Object Rangers)

- Unterstützung laufender Java Projekte
- Perfect Match
- Rent-a-team
- Coaching on the project
- Inhouse Outsourcing

) Competence Center)

- Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme
- Methoden, Standards und Tools für die Entwicklung von offenen, unternehmensweiten Systemen

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 5 |



Unser Thema heute:

Refactoring Legacy Code + TDLCR

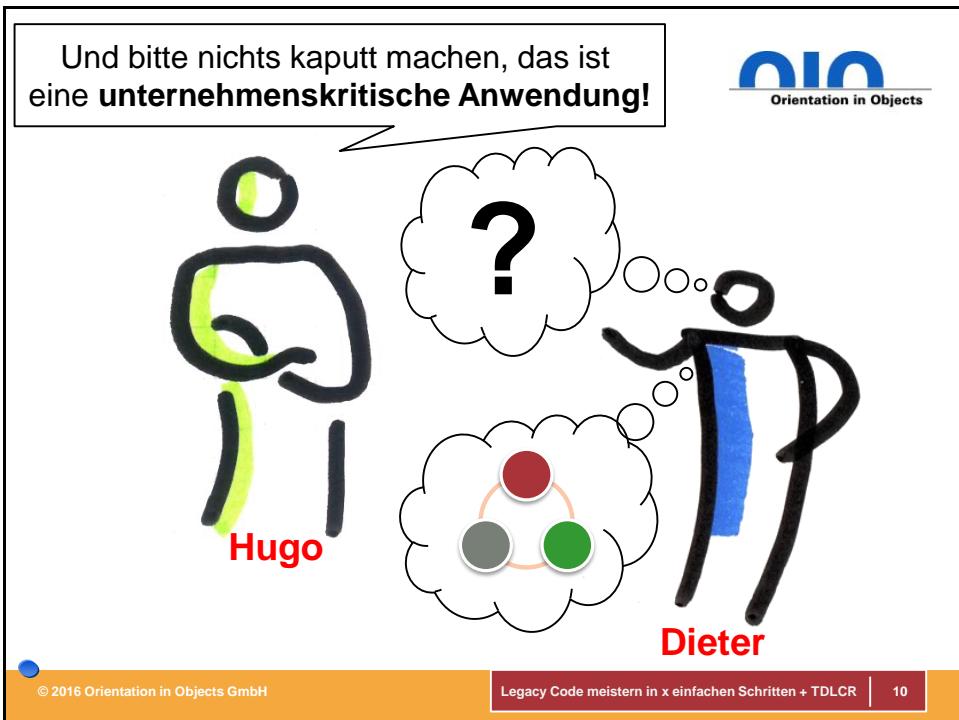
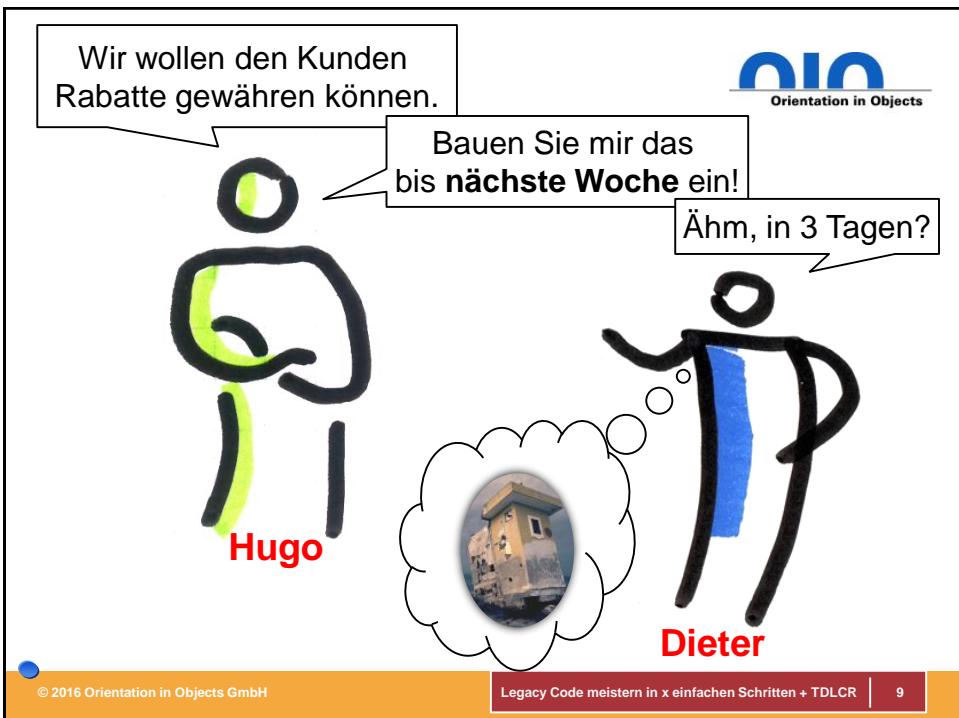
Disclaimer: KEIN Projekterfahrungsbericht

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 6 |





TD LCR

Test Driven Development

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 11

TDD hilft beim Verbessern des Codes



TDD Quotes
@TDDQuotes



Folgen

TDD is not a design tool. It's a software development workflow that has prompts for code improvement in its lifecycle #TDD
#programming

Übersetzung anzeigen



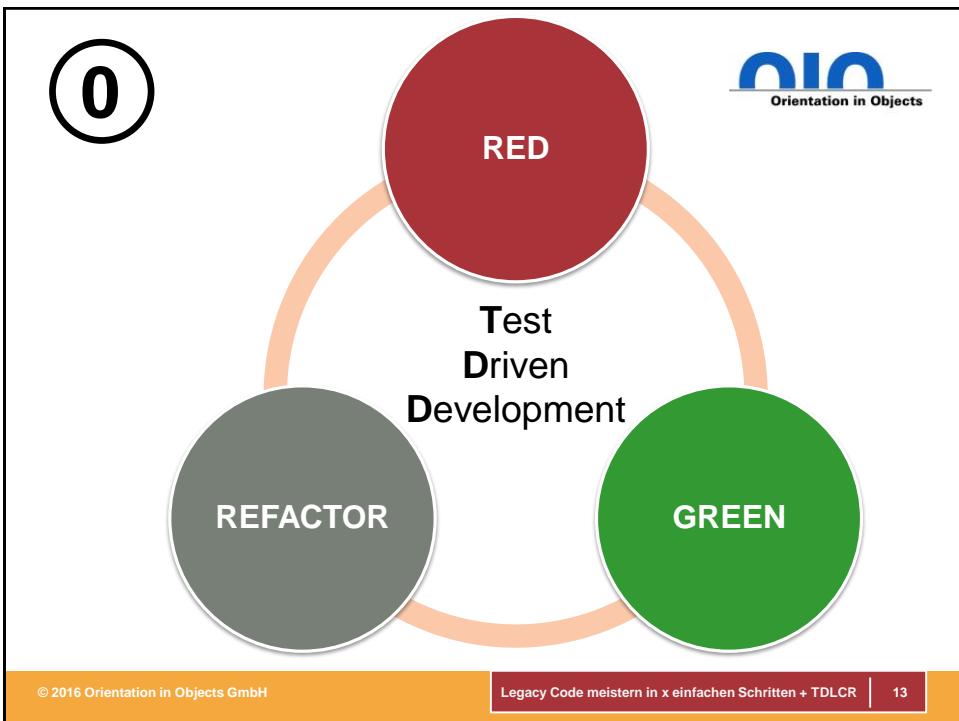
Sandro Mancuso

<https://mobile.twitter.com/TDDQuotes/status/783787123237265408>

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 12



- ### "Gesetze" von TDD
- Schreibe einen fehlschlagenden Test, bevor du Code für das Produktivsystem verfasst.
 - Schreibe nur so viel Code für den Test, damit er kompiliert (**Rot**)
 - Schreibe nur so viel Code, um den Test zu bestehen (**Grün**)
 - Refactor: Duplikation entfernen
- © 2016 Orientation in Objects GmbH
- Legacy Code meistern in x einfachen Schritten + TDLCR | 14

Live-Coding

Rabatt-Klasse mit Prozentsatz
10 % Rabatt => berechnePreis liefert 90 % vom Preis zurück
kein Rabatt => berechnePreis liefert gleichen Wert zurück
ignoriere negative Rabatte => berechnePreis liefert gleichen Wert zurück

Foto von StockSnap: <https://pixabay.com/de/codierung-gesch%C3%A4ft-arbeiten-macbook-699318/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 15 |

Tipps TDD

- kleine Schritte
- Workflow (Red – Green - Refactor) einhalten
- Ideen für weitere Testfälle in eine Liste schreiben und nach und nach abarbeiten (kleine Schritte!)

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 16 |

Vorteile TDD



- kein ungetesteter Code => sehr hohe Testabdeckung
- saubere/testbare Architektur durch TDD als Designstrategie
 - Anforderungen werden durch Tests dokumentiert
 - Software weniger fehleranfällig und wartungsintensiv
- hohe Qualität und verständlicher Code durch ständiges Aufräumen
 - Entfernen von Duplikationen
 - Abstrahieren wo sinnvoll
 - Einhalten der Code-Konventionen
- kein unnötiger Code auf Vorrat (YAGNI)
 - Konzentration auf das Wesentliche



TDLCR

Legacy Code

Legacy Code meistern in **8** einfachen Schritten

Nur heute,
nicht 1,
nicht 2,
...

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 19



Greenfield, Agile, TDD, ...

Foto von Antranias, CC0 Public Domain Lizenz, <https://pixabay.com/de/bauarbeiten-ge%C3%BCst-baustelle-670278/>

// TODO: refactor this

Foto von John Hult, CC0 1.0 Universal, <https://unsplash.com/search/uin?photo=uho8OH4zuE>

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR | 21

Was ist Legacy Code?

- Alter Code, geerbter Code, Code den niemand mehr versteht ...
- "Some crap made by someone else"
- Code entstanden unter Zeitdruck und ohne automatisierte Tests
- Technische Schulden

Was ist mit unserem eigenen Code?

Foto von John Hult, CC0 1.0 Universal, <https://unsplash.com/search/uin?photo=uho8OH4zuE>

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR | 22

Warum Legacy Code anfassen?

Verstehen

Erweitern

Bugfixing

Optimierung

© 2016 Orientation in Objects GmbH



Orientierung in Objekten

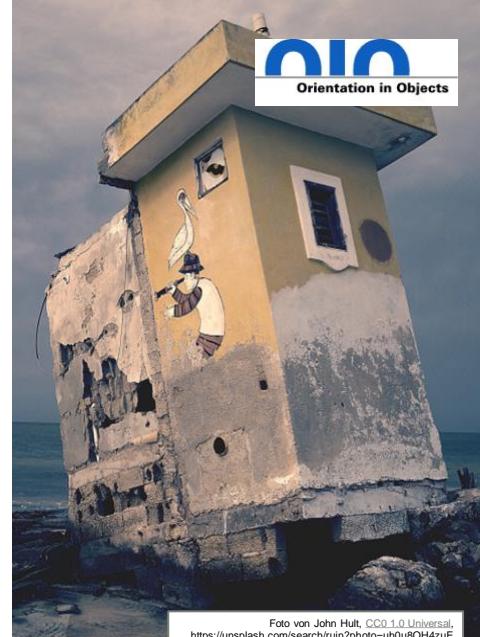


Foto von John Hult, CC0 1.0 Universal,
<https://unsplash.com/search/uin?photo=uhou8OH4zuE>

Legacy Code meistern in x einfachen Schritten + TDLCR

| 23

Wo kann TDD bei Legacy Code helfen?

Verstehen

Erweitern

Bugfixing

Optimierung

© 2016 Orientation in Objects GmbH



Orientierung in Objekten

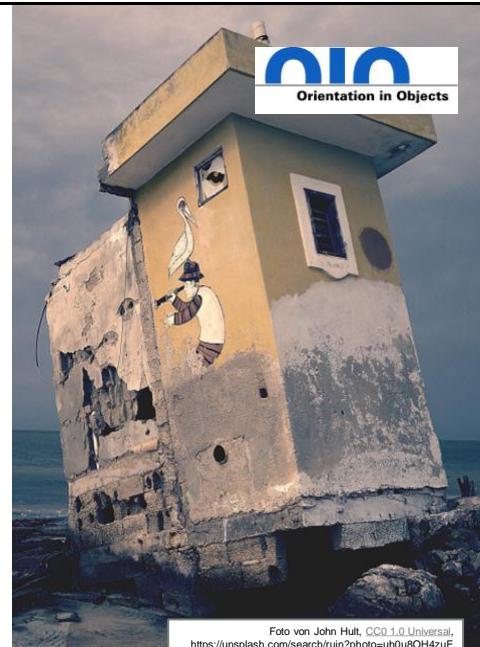


Foto von John Hult, CC0 1.0 Universal,
<https://unsplash.com/search/uin?photo=uhou8OH4zuE>

Legacy Code meistern in x einfachen Schritten + TDLCR

| 24

TDD vs. Legacy Code

Test vor dem Code vs.
Code schon da

.

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 25

TDLCR

Test Driven Legacy Code Refactoring

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 26



Somebody else's code

Jeder kennt ihn ...

Keiner mag ihn ...

Was ist mit unserem
eigenen Code?

Refactoring Legacy Code

Warum?

Verstehen

Erweitern

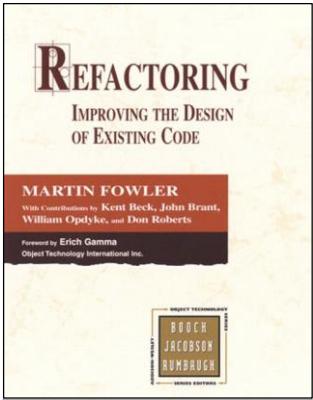
Bugfixing

Optimierung



Refactoring Legacy Code

OIO
Orientation in Objects



Annahmen

Es gibt automatisierte Tests ...

Quellcode ist schon testbar ...

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 31

Refactoring Legacy Code

OIO
Orientation in Objects

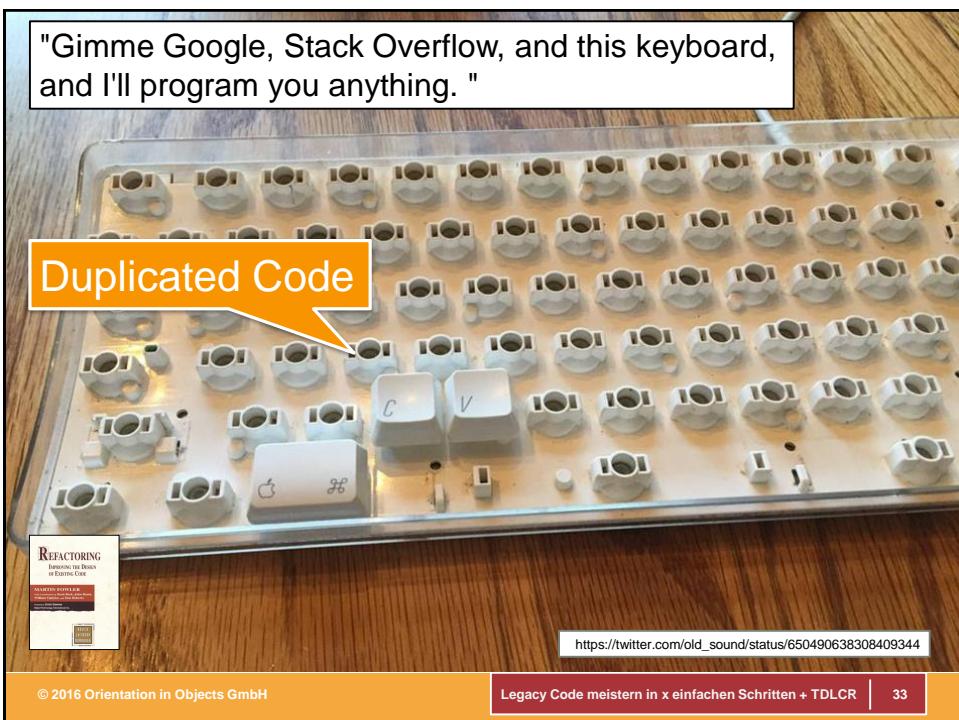
Code Smells 

- Temporary Field
- Long Method
- Feature Envy
- ...
- Code Comments**



<https://twitter.com/petecheslock/status/646507209413775360/photo/1>

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 32



Refactoring Legacy Code

“Code
without
tests

Orientation in Objects

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 34



**“ Code
without tests
is bad code.**

Michael Feathers



J. B. Rainsberger

**“ Legacy code is
valuable code
that we feel afraid
to change.**



Es ist egal, wie ...



... gut geschrieben der Code ist

... schön der Code ist

... objektorientiert der Code ist

... entkoppelt der Code ist

Tests

lassen unser Verhalten schnell und verifizierbar ändern

Ohne Tests

wissen wir nicht, ob der Code besser oder schlechter wird

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 37

Die gute Nachricht ...

Keine Wissenschaft

Gesunder Menschenverstand

Foto von Engel62: <https://pixabay.com/de/daniel-d%C3%BCsentrieb-helferlein-123206/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 38

Was macht es dann schwierig?



Hello World vs. 50.000++ LOC

Disziplin (kleine Schritte, ...)

Aussagekräftige Testabdeckung

Legacy zu Clean Code?



Clean Code ist NICHT das Ziel

Hauptfokus: testbarer Code

Dann schreiben wir halt Tests ...



viel zu teuer

Code meist kaum/nicht testbar

starke Kopplung, geringe Kohäsion

Legacy Code meistern in x einfachen Schritten + TDLCR

41

Legacy Code meistern in x einfachen Schritten + TPLCR

42



Henne-Ei-Problem

Legacy Code meistern in x einfachen Schritten + TDLCR

43

Vorgehen



1. Identify what to change
 2. Identify what to test
 3. Break dependencies
 4. Write the tests
 5. Modify and refactoring

Legacy Code meistern in x einfachen Schritten + TDD GB

44

Dann mal her mit den
X einfachen Schritten!

No Silver Bullet

Jedes Projekt individuell

Vorsicht beim Beheben von offensichtlichen Fehlern

A



Sicherheitsnetz
+ Tests

Foto von bella67: <https://pixabay.com/de/spinnennetz-mit-wasserperlen-netz-921039/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 47

B



Sanierung

Foto von KlausHausmann: <https://pixabay.com/de/bauarbeiter-bau-bauen-bohrhammer-921224/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 48

1 Golden Master



gegenwärtiges Verhalten dokumentieren und erhalten



Foto von istara: <https://pixabay.com/de/gold-bar-goldbarren-reich-geld-296115/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 49

```
# suppose that our legacy code is this program called 'game'
$ game > GOLDEN_MASTER

# after some changes we can check to see if behaviour has changed
$ game > OUT-01
$ diff GOLDEN_MASTER OUT-01
# GOLDEN_MASTER and OUT-01 are the same

# after some other changes we check again and...
$ game > OUT-02
$ diff GOLDEN_MASTER OUT-02
# GOLDEN_MASTER and OUT-02 are different -> behaviour changed
```

Golden Master (aka characterization tests)

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 50

Konsolenausgaben abprüfen

```
@Before
public void init() {
    originalSysOut = System.out;
    consoleStream = new ByteArrayOutputStream();
    PrintStream printStream = new PrintStream(consoleStream);
    System.setOut(printStream);
}

@Test
public void testSimpleOutput() {
    System.out.println("Hallo Publikum!");
    System.out.print("Hallo Falk!");
    assertEquals("Hallo Publikum!\r\nHallo Falk!", consoleStream.toString());
}

@After
public void teardown() {
    System.setOut(originalSysOut);
}
```

Live-Coding



Foto von StockSnap: <https://pixabay.com/de/codierung-gesch%C3%A4ft-arbeiten-macbook-699318/> (CC0 Public Domain Lizenz)

1

Golden Master



Vorsicht bei Zufallsgeneratoren

```
1.9.3 > g = Random.new  
1.9.3 > (1..10).map{g.rand(1000)}  
=> [691, 362, 997, 692, 236, 532, 687, 616, 218, 702]  
1.9.3 > g = Random.new  
1.9.3 > (1..10).map{g.rand(1000)}  
=> [865, 186, 89, 382, 894, 708, 769, 850, 452, 85]  
1.9.3 > g = Random.new(1)  
1.9.3 > (1..10).map{g.rand(1000)}  
=> [37, 235, 908, 72, 767, 905, 715, 645, 847, 960]  
1.9.3 > g = Random.new(1)  
1.9.3 > (1..10).map{g.rand(1000)}  
=> [37, 235, 908, 72, 767, 905, 715, 645, 847, 960]
```

Festlegen von Seeds (Pseudo-Random)

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 53

2

Subclass To Test

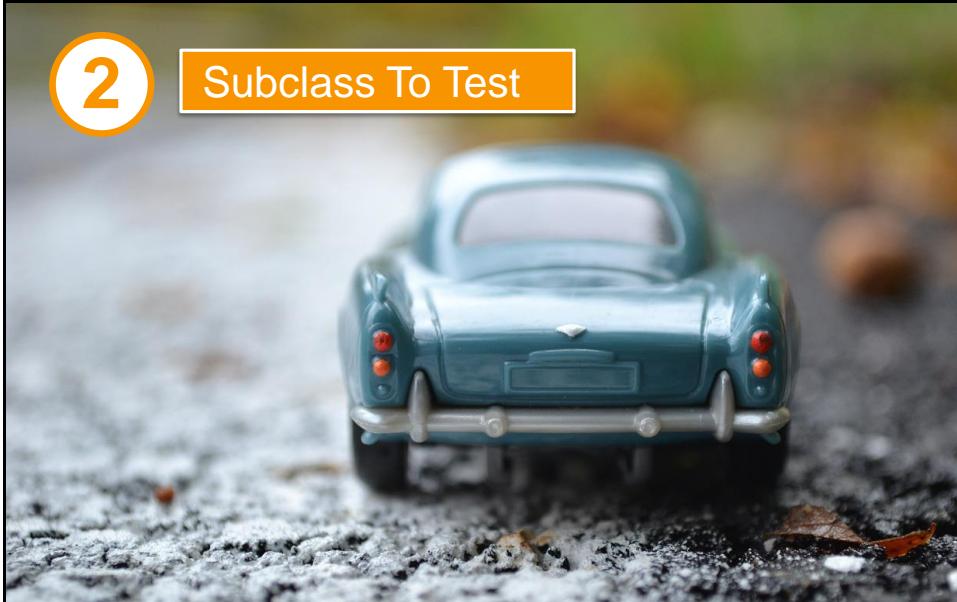


Foto von chrisli8020: <https://pixabay.com/de/auto-maschine-spielzeug-786315/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 54

2

Subclass To Test



Seam (Nahtstelle)

Ein Seam ist eine Stelle, an der man das Verhalten editieren kann, ohne direkt an dieser Stelle zu ändern.

Aufbrechen stark gekoppelter Abhängigkeiten

aka Extract and Override

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 55

Live-Coding



Foto von StockSnap: <https://pixabay.com/de/codierung-gesch%C3%A4ft-arbeiten-macbook-699318/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 56

3 Extract Pure Functions



Foto von PublicDomainPictures: <https://pixabay.com/de/wasser-tropfen-tr%C3%BCpfchen-grau-72879/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 57

3 Extract Pure Functions



seiteneffektfrei

keine Statusänderung



"It's a classic,
we call it a Klassiker"

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 58

3

Extract Pure Functions



```
"pure function".substring(5);  
UrlEncoder.encode("pure function");  
Math.max(x, y);
```



```
System.out.println("unpure");  
list.add(3);  
Collections.sort(list);
```



3

Extract Pure Functions



Codestellen isolieren

Ziele

Separat testbar

Duplikation reduzieren



Live-Coding



Foto von StockSnap: <https://pixabay.com/de/codierung-gesch%C3%A4ft-arbeiten-macbook-699318/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 63

5 Extract Class



Large class
SRP verletzt

Ziele

Unabhängiges Testen einzelner Teile
Sauberes OO-Design

Foto von blickpixel: <https://pixabay.com/de/weihnachtsdekoration-pakete-geschenk-570797/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 64

Live-Coding



Foto von StockSnap: <https://pixabay.com/de/codierung-gesch%C3%A4ft-arbeiten-macbook-699318/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH Legacy Code meistern in x einfachen Schritten + TDLCR | 65

~~Hätten wir mehr Zeit ...~~

- ⑥ Dependency Inversion
- ⑦ Test non-public member
- ⑧ Mocking Framework

© 2016 Orientation in Objects GmbH Legacy Code meistern in x einfachen Schritten + TDLCR | 66

6

Dependency Inversion

Entkoppeln
durch **explizites**
Setzen der
Abhängigkeiten



Foto von tatlin: <https://pixabay.com/de/zigarette-rauch-tabak-106610/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 67

Live-Coding



Foto von StockSnap: <https://pixabay.com/de/codierung-gesch%C3%A4ft-arbeiten-macbook-699318/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 68

7 Test non-public member

Wirklich?

Gute Alternative bei Legacy Code!

Foto von Guian Bolisay: <https://www.flickr.com/photos/instantvantage/5151841152/> (CC BY-SA 2.0)

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 69

7 Test non-public member

Tools

Reflection

Spring Reflection(Test)Utils

dp4j

BoundBox

PowerMock Whitebox

Foto von Guian Bolisay: <https://www.flickr.com/photos/instantvantage/5151841152/> (CC BY-SA 2.0)

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 70



8 Mocking

- Subclass To Test on Steroids!
- Interaktion mit Umgebung testen
- Erwartete Parameter und Aufrufreihenfolge

Foto von tatlin: <https://pixabay.com/de/zigarette-rauch-tabak-106610/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 72



OIO
Orientation in Objects

Zusammenfassung

© 2016 Orientation in Objects GmbH Legacy Code meistern in x einfachen Schritten + TDLCR | 74

Sicherheitsnetz

- ① Golden Master
- ② Subclass To Test
- ③ Extract Pure Functions
- ④ Remove Duplication
- ⑤ Extract Class
- ⑥ Dependency Inversion
- ⑦ Test non-public member
- ⑧ Mocking Framework

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 75

OIO
Orientation in Objects

Sicherheitsnetz

- ① Golden Master

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 76

Was mache ich bei GUI-Anwendungen?

③

④

public static void main(String... args) throws Exception {
 WebDriver driver = new FirefoxDriver();
 driver.get("http://www.retest.de");
 while (true) {
 List<WebElement> links = driver.findElements(By.tagName("a"));
 links.get(random.nextInt(links.size())).click();
 Thread.sleep(500);
 List<WebElement> fields =
 driver.findElements(By.xpath("//input[@type='text']"));
 WebElement field = fields.get(random.nextInt(fields.size()));
 field.sendKeys(randomString());
 Thread.sleep(500);
 }
}

https://entwicklertag.de/frankfurt/2016/sites/entwicklertag.de.frankfurt.2016/files/slides/Bei%20uns%20testen%20lauter%20Affen_0.pdf

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 77

Leichtere Testbarkeit

② Subclass To Test

③ Extract Pure Functions

⑤ Extract Class

⑥ Dependency Inversion

⑦ Test non-public member

⑧ Mocking Framework

© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 78

Besseres Verständnis



- ③ Extract Pure Functions
- ④ Remove Duplication
- ⑤ Extract Class



© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 79

T
D
L
C
R



Neue
Features
ausliefern ...

Foto von PublicDomainPictures: <https://pixabay.com/en/adorable-baby-basket-beautiful-boy-21259/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 80

Aufhören Legacy Code zu schreiben!



- sonst wird die Codebasis nur schlimmer und man entfernt sich immer mehr davon, jemals Tests hinzuzufügen
- **Keine neuen Features mehr ohne Unit-Testing!**
- Sprout Method + Wrap Method als Hilfen

Sprout Method



```
public int berechne(int a, int b) {  
    int c = a + b;  
    // weitere wichtige Aufgaben  
  
    // neues Verhalten  
    c = verdoppeleResult(c);  
  
    // weitere wichtige Aufgaben  
    return c;  
}
```

Aufruf aus
Bestandscode

neue Methode,
testgetrieben
entwickeln

```
protected int verdoppeleResult(int result) {...}
```

Sprout Method und TDD (1)



```
// nicht kompilierenden Test schreiben

@Test
public void testVerdoppeleResult() {
    assertEquals(2, rechner.verdoppeleResult(1));
}

// Kompilerfehler beheben

protected int verdoppeleResult(int result) {
    return 0;
}

// Test ist rot
```

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 83

Sprout Method und TDD (2)



```
// einfachste Lösung, damit Test grün wird

protected int verdoppeleResult(int result) {
    return 2;
}

// nichts zu refactoren, von vorn beginnen

@Test
public void testVerdoppeleResult() {
    assertEquals(2, rechner.verdoppeleResult(1));
    assertEquals(4, rechner.verdoppeleResult(2));
}

// Test wieder rot, Lösung generalisieren
```

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 84

Wrap Method



- ähnlich zu Sprout Method, neues Verhalten aber vor oder am Ende der zu ändernden Methode

```
public int berechne(int a, int b) {  
    logResult(c);  
    return berechnePrivate(a, b);  
}  
  
private int berechnePrivate(int a, int b) {  
    int c = a + b;  
    // weitere wichtige Aufgaben  
    return c;  
}  
  
protected void logResult(int result) {....}
```

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 85 |

Live-Coding



Foto von StockSnap: <https://pixabay.com/de/codierung-gesch%C3%A4ft-arbeiten-macbook-699318/> (CC0 Public Domain Lizenz)

© 2016 Orientation in Objects GmbH

Legacy Code meistern in x einfachen Schritten + TDLCR

| 86 |

Zusammenfassung: Sprout Method und TDD



- Test schreiben vor der Integration der neuen Funktionalität
 - Red – Green – Refactor
- alte Funktionalität wird gleich mitgetestet
 - Testbett wird bei jeder Erweiterung besser
- uninteressante Abhängigkeiten durch Mocks ersetzen

Zusammenfassung (2. Teil)



- Sicherheitsnetz spannen: Golden Master
- Legacy Code refactoren => Ziel testbarer Code
 - Dependencies aufbrechen
 - Codestellen separieren
- Tests immer hinzufügen bei
 - Bugfixing
 - Erweitern
- TDD konsequent beim Schreiben von neuem Code und auch bei der Aufruf-Integration von neuem Code
- nach und nach alten Code ebenfalls mit Unit-Tests versehen

Tools

Eclipse Metrics

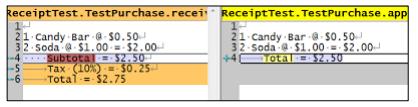




MoreUnit

+ MoreUnit

Approval Tests



Infinitest

No related tests found for last change.

1 test cases ran at 10:24:01

Code Coverage



© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 89

Legacy Code Retreat



<https://github.com/jbrains/trivia>

Foto von Jmabel: https://commons.wikimedia.org/wiki/File:Seattle_-_Budokan_Dojidojo_demo_04.jpg?uselang=de (CC BY-SA 3.0 Lizenz)

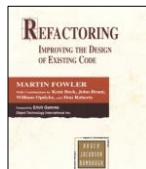
© 2016 Orientation in Objects GmbH | Legacy Code meistern in x einfachen Schritten + TDLCR | 90

Links

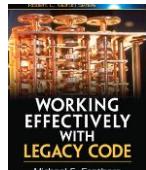


- Code-Beispiel der Live-Demo (**Legacy Code meistern in x ...**)
 - <https://github.com/sippsack/BadTelefon-Refactoring-Legacy-Code>
- Code-Beispiel der Live-Demo (**TDLCR**)
 - <https://github.com/sippsack/BadTelefon-Test-Driven-Legacy-Code-Refactoring>
- anderes Code-Beispiel für Legacy Code
 - <https://github.com/jbrains/trivia>
- Blog: Techniken zu Legacy Code-Retreat
 - <http://blog.adrianbolboaca.ro/2014/04/legacy-coderetreat/>

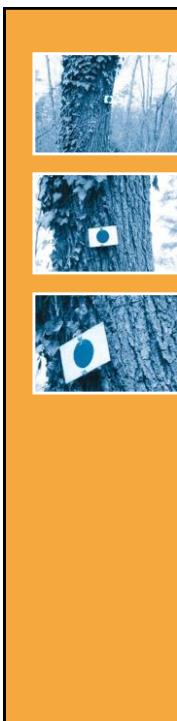
Literaturhinweise



- Refactoring
 - Sprache: Englisch
 - Gebunden - 464 Seiten - Addison Wesley
 - Erscheinungsdatum: 1. Juni 1999
 - ISBN: 0201485672



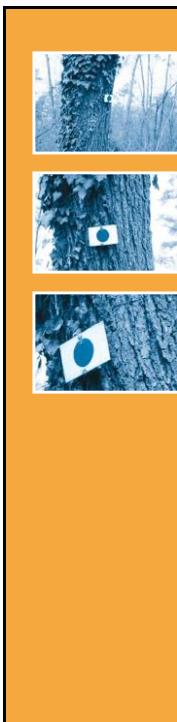
- Working Effectively with Legacy Code
 - Sprache: Englisch
 - Gebunden



Fragen ?

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de
info@oio.de

oio
Orientation in Objects



**Vielen Dank für ihre
Aufmerksamkeit !**

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de
info@oio.de

oio
Orientation in Objects