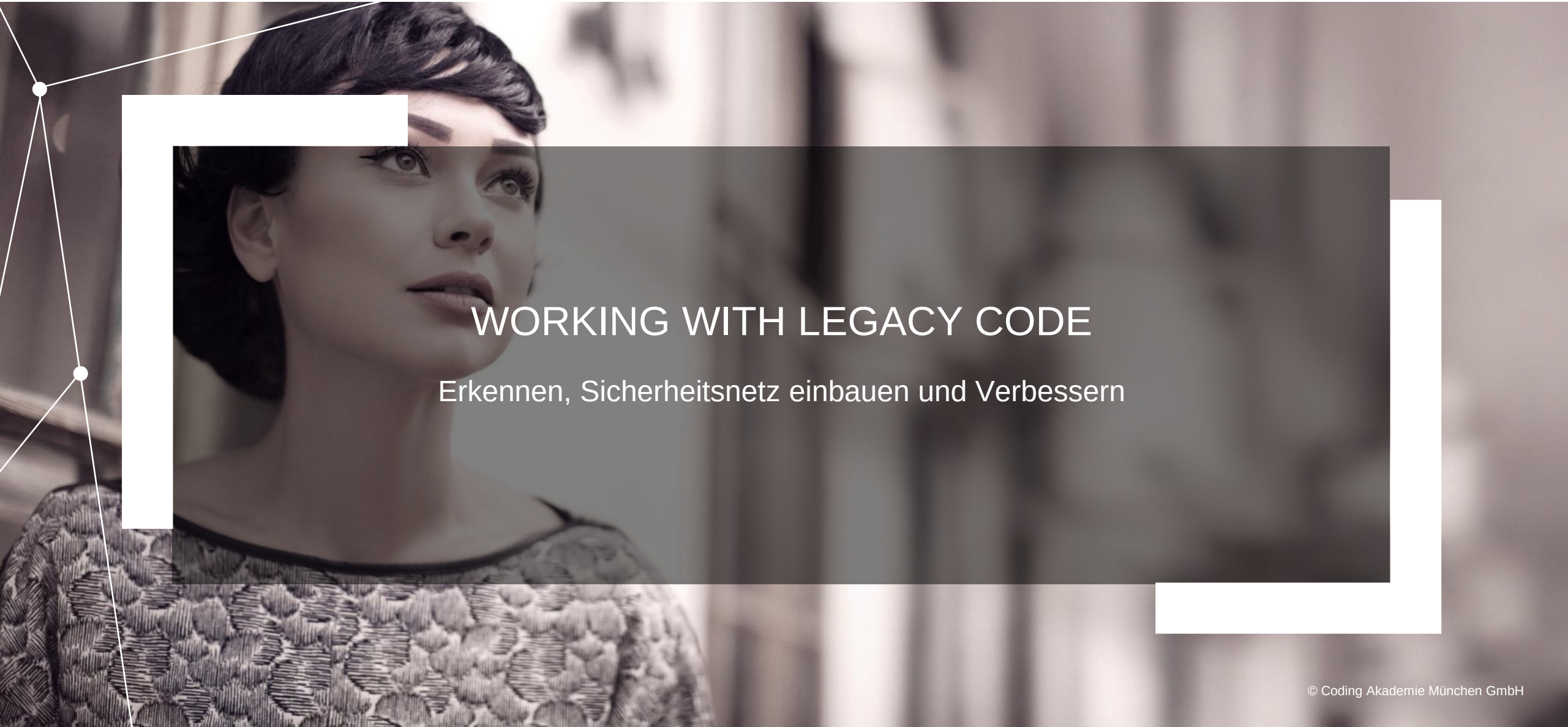


WORKING WITH LEGACY CODE

Erkennen, Sicherheitsnetz einbauen und Verbessern

© Coding Akademie München GmbH

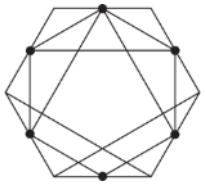
Übung: Pair-Share

WAS BRAUCHE ICH DAMIT ICH CODE VERBESSERN KANN?

- Denke zurück als du in einem Code eine Erweiterung implementieren solltest
  - Was hat die Aufgabe erleichtert?
  - Was hat die Aufgabe erschwert?

- Sprich mit deinem Partner über diese konkrete Situation

© Coding Akademie München GmbH



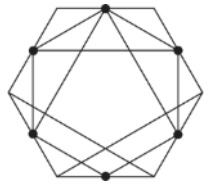
SICHERHEITSNETZ

- Ein Sicherheitsnetz verbessert die Fähigkeit Veränderungen durchzuführen
- Ein Sicherheitsnetz verbessert die Beurteilungsfähigkeit ob etwas besser oder schlechter geworden ist

SICHERHEITSNETZ IN JAVA → UNIT-TEST

- Wie muss ein Sicherheitsnetz aussehen, welches bei Legacy Code „Sicherheit“ bieten kann?

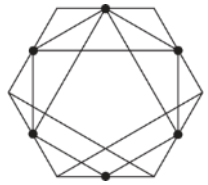
© Coding Akademie München GmbH



ZUR ERINNERUNG – EIN EINFACHER UNIT-TEST

```
public class TicketMachineTest {
    @Test
    public void noDiscountLessThan10Visits() {
        // given
        MembershipCard card = new MembershipCard(LocalDate.of(1970, 5, 22));
        card.setNumberOfVisits(3);
        TicketMachine ticketMachine = createTicketMachine();

        // when
        String display = ticketMachine.displayTicketPrice("DISCOUNT", card);

        // then
        assertEquals("please pay 10,00", display);
    }
}
```

© Coding Akademie München GmbH



SICHERHEITSNETZ IN KOMPLEXEN LEGACY SYSTEMEN

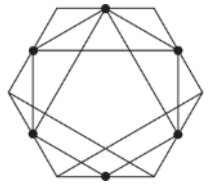
Legacy System kennzeichnen sich häufig durch fehlende oder zu wenig Tests aus.

Gleichzeitig eine hohe Komplexität.

Gleichzeitig ein komplexes Softwaredesign, welches das Erstellen von Tests nicht begünstigt.

Der Golden Master Ansatz bietet eine Möglichkeit ein Sicherheitsnetz einzuziehen.

© Coding Akademie München GmbH



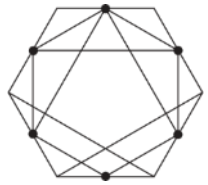
GOLDEN MASTER

1. Jede Interaktion mit externen System wird sehr detailliert protokolliert.
2. Gegen dieses Protokoll (=Golden Master) wird kontinuierlich getestet.
3. Falls der Test okay ist, ist man sicher, dass die Applikation immer noch korrekt ist.
   Relativ zu vor den Änderungen

→ = Sicherheitsnetz

- **Konsolenausgabe**
- **Aufrufe über das Netzwerk**
- **Aufrufe (und Änderungen) auf der Datenbank**
- **Systemprotokolldateien**
- **Geldtransferanweisungen**
- **…**

© Coding Akademie München GmbH



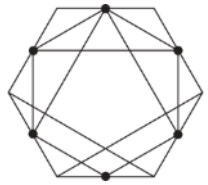
EINFACHER GOLDEN MASTER – FIZZ BUZZ

```
.\excercise\golden-master-fizzbuzz
```

```
public class FizzBuzz {
    public static void main(String[] args) {
        for (int i = 0; i < 100; i++, System.out.println(
                i % 3 == 0 || i % 5 == 0 ? ((i % 3) == 0 ? "fizz" : "") + ((i % 5) == 0 ? "buzz" : "") : i))
            ;
    }
}
```

```
    @Test
    public void goldenMasterTest() throws Exception {
        // TODO create a golden master from FizzBuzz's System.out.println
    }
```

© Coding Akademie München GmbH



WAS SIND SEAMS?

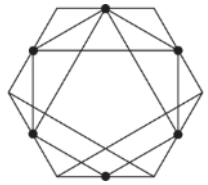
Ein „**seam**“ ist eine Nahtstelle, die aufgetrennt werden kann.

Im Code ist sind es Stellen, um z.B. Informationen abzugreifen oder ein besonderes Verhalten einzuführen.

Häufig bei Golden Master müssen wir non-invasive Codeänderungen einführen, damit wir an interessante Informationen kommen.

Speziell in komplexen Fällen führen wir seams ein, damit wir einen besseren Golden Master erstellen können. **Später dienen diese seams als Ausgangspunkt um das Software Design zu verbessern**.

© Coding Akademie München GmbH

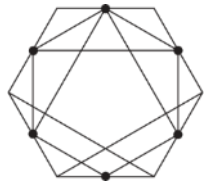


EIN EINFACHER SEAM

```
public static void main(String[] args) {
    for (int i = 0; i < 100; i++, println_seam(
            i % 3 == 0 || i % 5 == 0 ? ((i % 3) == 0 ? "fizz" : "") + ((i % 5) == 0 ? "buzz" : "") : i))
        ;
}

protected static void println_seam(Object message) {
    System.out.println(message);
}
```

- Seams sind häufig **`public`** oder **`protected`**
- Seams sind ein erster Schritt in Richtung **dependency breaking**
- Seams sind ein erster Schritt in Richtung **redesign**

© Coding Akademie München GmbH



SEAMS UND GOLDEN MASTER

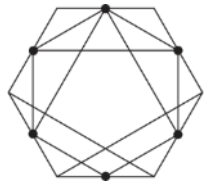
./excercise/golden-master-seam/

```
private void main() throws IOException {
       System.out.println("Hello User");

       URL url = new URL("https://www.timeanddate.com/");
       HttpURLConnection con = (HttpURLConnection) url.openConnection();
       con.setRequestMethod("GET");
       con.setDoOutput(true);

       DataOutputStream out = new DataOutputStream(con.getOutputStream());
       out.flush();
       out.close();

       BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
       String inputLine;
       StringBuffer content = new StringBuffer();
       while ((inputLine = in.readLine()) != null) {
           content.append(inputLine);
       }
       in.close();
       con.disconnect();

       final String searchFor = "<span id=\"clk_hm\">";
       int pos = content.indexOf(searchFor);
       System.out.println(content.substring(pos + searchFor.length(), pos + searchFor.length() + 5));
       System.out.println("Bye");
   }
```

```
@Test
    public void goldenMasterTest() throws Exception {
        // TODO create a golden master from BuildSeamsInto's
System.out.println AND network communication
    }
```

- Führe seams ein,
  - damit an diesen seams Informationen für den Golden Master bereit gestellt werden können
  - damit „stabile“ Informationen von außen bereitgestellt werden

© Coding Akademie München GmbH



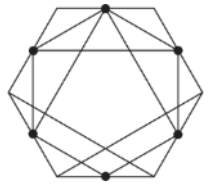
KEY POINTS

- Jede Änderung braucht ein Sicherheitsnetz
- Mit der Qualität und Zuverlässlichkeit des Sicherheitsnetz wächst die Fähigkeit tiefgreifende Veränderungen zum Bessern durchzuführen
- Mit einem guten Sicherheitsnetz sind Änderungen und Auslieferung mit vertretbaren Risiko im vollen Umfang jederzeit möglich

- **Golden Master und Characterization Test**
- **Seam**
- **Unit Test**
- **Continuous Integration Practice**
- **Continuous Deployment**

© Coding Akademie München GmbH


PATTERN

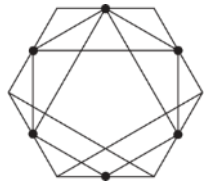


PATTERN: LONG METHOD

- Lange Methoden „wissen“ nicht was sie tun
- Die Kohäsion innerhalb der Methode fehlt
- Weil eine lange Methode „viel“ macht, ist es schwierige diese Methode korrekt aufzurufen und mit Fehlerfällen umzugehen

- **Finde kohäsive Abschnitte in langen Methoden**
- **Benenne eine Methode passend zu dem was sie macht**

© Coding Akademie München GmbH

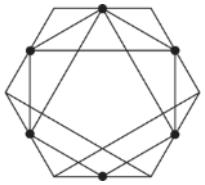


REFACTORING: EXTRACT METHOD

- Beim Aufräumen innerhalb einer Methode wird ein kohäsive Teilabschnitt gefunden
- Extrahiere diesen in eine eigene Methode oder Klasse
- Bringe sinnvolle Fehlerbehandlung in die Methode hinein
- Ggf. müssen Codezeilen neu sortiert werden

- **Erkenne zusammenhängende Codezeilen**
- **Sortiere zusammenhängende Codezeilen nahe zusammen**
- **Extrahiere diese Codezeilen in eine neue Methode**

© Coding Akademie München GmbH



EXTRACT METHOD

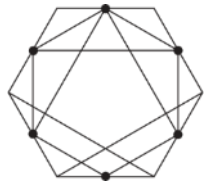
```
public class BuildSeamsInto {
…
    private void main() throws IOException {
…
```

`./excercise/golden-master-seam/`

- Suche zusammenhänge Codezeilen
- Versuche diese mit der IDE zu extrahieren
- Sortiere Codezeilen neue und versuche erneut eine Methode zu extrahieren

- Was fällt Euch auf?

© Coding Akademie München GmbH

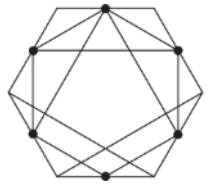


PATTERN: EXTRACT CLASS

- Gibt es einen Ansammlung von Methoden, die eng miteinander in der Klasse arbeiten, aber mit vielen anderen Methoden dieser Klasse nicht
  → überlege, ob diese Methoden in einer eigenen Klasse extrahiert werden können
- Die neue Klasse kann unabhängige Zustände und Lebenszyklen haben
- Das Verhalten kann durch gezieltes Testen der Klasse gesichert und ausgetauscht werden

- **Methoden, die in enger Kollaboration arbeiten**
- **Zustände, die nur für diese und zwischen diesen Methoden Sinn ergeben**
- **Vereinfacht das Testen und Austauschen von Verhalten**

© Coding Akademie München GmbH



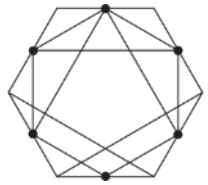
EXTRACT CLASSES

./excercise/extract-class/

```
public class User {
…
 public static User createUser(String name, String mobil, String fax, String address) {
        System.out.println("create user " + name);
…
```

- Was sind Konzept und kann das Konzept explizit statt implizit abgebildet werden?
- Wo sind zu viele Details?

© Coding Akademie München GmbH



WAS IST FEATURE ENVY?

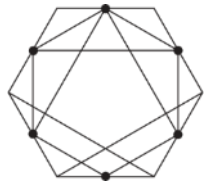
Kollaboration zwischen Objekten ist gewollt, schließlich gestalten wir Klassen entsprechend (abstrakten) Konzepten zur Abstraktion und besseren Wiederverwendung.

Wenn nun eine Klasse/Methode eine andere ständig Fragen muss, so wirft dies die Frage auf, ob diese korrekt designed wurde:

- Relevante Informationen zur Erledigung einer Aufgabe ist woanders beheimatet als dort wo diese gebraucht wird.
- Internas sind weithin sichtbar und implizit umfangreiche Änderungen

Dies nennt man **feature envy und macht Code sehr rigide und schwierig zu verändern**.

© Coding Akademie München GmbH



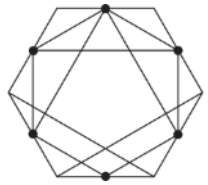
FEATURE ENVY

./excercise/feature-envy/

```
public class CinemaDoorman {
    public void main(String[] args) {
        System.out.println("Welcome to the cinema of Horror movies");

        User anton = new User(18);
        Movie movie = new Movie("Horror of the Code");

        if (anton.getWallet()
                .getAmount() >= movie.getEntryFee()) {
            if (anton.getAge() >= movie.getAgeRate()) {
                System.out.println(anton.getName() + " visits " + movie.getName());
            }
        }
    }
}
```

- Wo sieht man feature-envy?
- Wie kann das gelöst werden?
  - Im Code?
  - Im Prozess?
    Bedenke:„Conway-Law“ bzgl. Kommunikation zwischen Abteilungen und Abbildung im Code.

© Coding Akademie München GmbH

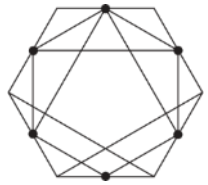


FEATURE ENVY AUFLÖSEN

```
$ git log --oneline --graph
* a2ffc7c (HEAD -> master, origin/master) Working on feature-envy: Remove unused method from User
* 3572316 Working on feature-envy: Remove unused method from Money
* 5e05737 Working on feature-envy: Remove feature-envies from Wallet
* f562690 Working on feature-envy: Refactor - inline variable
* f9fe06c Working on feature-envy: Refactor - inline variable
* 5d37389 Working on feature-envy: Rename method in Movie
* ad849e8 Working on feature-envy: Remove ununsed method from Movie
* e96d0b1 Working on feature-envy: Remove unused method from User
* 16adb81 Working on feature-envy: Remove unused method from Wallet
* c63767c Working on feature-envy: Implement permit_entry_if_visitor_does_have_sufficient_funds
* f937456 Working on feature-envy: Improve the safety-net by adding unit test
* 3d252ba Working on feature-envy: Added ability to User to have a non-empty Wallet
* ac59e49 Working on feature-envy: Added ability to Wallet to fund with more than no money
* 02523b0 Working on feature-envy: Added ability to copy Money
* c098730 Working on feature-envy: Switch logic to use Money in isUserAllowedToVisitTheMovie
* 6d2138c Working on feature-envy: Introduce class Money for concept money
* 36e7d95 Working on feature-envy: Remove feature-envy statement
* 8530350 Working on feature-envy: Improve unit test
* 57b8c79 Working on feature-envy: Create seam with extract method
* cb18ac8 Working on feature-envy: Create seam with extract method
* 41b2466 Working on feature-envy: Introduce pre-cursor to extract method.
```

Eine Lösungsvariante – Schritt für Schritt in 21 commits

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

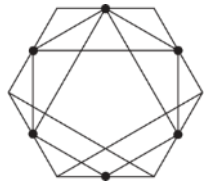
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
index 517fe91..b51679a 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
@@ -7,11 +7,17 @@
         User anton = new User(18);
         Movie movie = new Movie("Horror of the Code");

+        boolean userIsAllowToVisitTheMovie = false;
         if (anton.getWallet()
                 .getAmount() >= movie.getEntryFee()) {
             if (anton.getAge() >= movie.getAgeRate()) {
+                // userIsAllowToVisitTheMovie = true;
                 System.out.println(anton.getName() + " visits " + movie.getName());
             }
         }
+
+        if (userIsAllowToVisitTheMovie) {
+            System.out.println(anton.getName() + " visits " + movie.getName());
+        }
     }
 }
```

Sicherheitsnetz verbessern

- Seam Einführung vorbereiten

```
Working on feature-envy: Introduce pre-
cursor to extract method.

Introduced code is for now not-used, for
now we want to have those
changes non-invasive.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

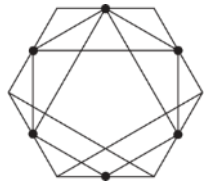
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
index b51679a..556f9fb 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
@@ -8,16 +8,20 @@
         Movie movie = new Movie("Horror of the Code");

         boolean userIsAllowToVisitTheMovie = false;
-        if (anton.getWallet()
-                .getAmount() >= movie.getEntryFee()) {
-            if (anton.getAge() >= movie.getAgeRate()) {
-                // userIsAllowToVisitTheMovie = true;
-                System.out.println(anton.getName() + " visits " + movie.getName());
-            }
-        }
+        userIsAllowToVisitTheMovie = isUserAllowedToVisitTheMovie(anton, movie);

         if (userIsAllowToVisitTheMovie) {
             System.out.println(anton.getName() + " visits " + movie.getName());
         }
     }
+
+    public boolean isUserAllowedToVisitTheMovie(User anton, Movie movie) {
+        if (anton.getWallet()
+                .getAmount() >= movie.getEntryFee()) {
+            if (anton.getAge() >= movie.getAgeRate()) {
+                return true;
+            }
+        }
+        return false;
+    }
 }
```

Sicherheitsnetz verbessern

- Seam einführen

```
Working on feature-envy: Create seam with
extract method

This way, unit tests in the same package
can query the core business
logic.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

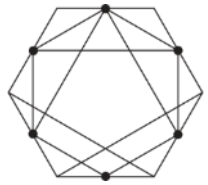
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
index 556f9fb..11541a9 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
@@ -15,7 +15,7 @@
         }
     }

-    public boolean isUserAllowedToVisitTheMovie(User anton, Movie movie) {
+    boolean isUserAllowedToVisitTheMovie(User anton, Movie movie) {
         if (anton.getWallet()
                 .getAmount() >= movie.getEntryFee()) {
             if (anton.getAge() >= movie.getAgeRate()) {

diff --git a/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java b/feature-
envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
index 1f1e54b..fdec1db 100644
--- a/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
+++ b/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
@@ -7,6 +7,7 @@
 public class CinemaDoormanShould {
     @Test
     void permit_access_to_old_people() {
+        new CinemaDoorman().main(null);
         assertEquals(0, 1);
     }
 }
```

Sicherheitsnetz verbessern

- Sichtbarkeit anpassen

```
Working on feature-envy: Create seam with
extract method

This way, unit tests in the same package
can query the core business
logic directly and verify proper behavior.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

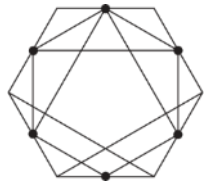
```
diff --git a/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java b/feature-
envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
index fdec1db..d3a99a8 100644
--- a/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
+++ b/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
@@ -1,13 +1,16 @@
 package de.codingakademie.featureenvy;

-import static org.junit.jupiter.api.Assertions.assertEquals;
+import static org.junit.jupiter.api.Assertions.assertFalse;

 import org.junit.jupiter.api.Test;

 public class CinemaDoormanShould {
     @Test
     void permit_access_to_old_people() {
-        new CinemaDoorman().main(null);
-        assertEquals(0, 1);
+        CinemaDoorman doorman = new CinemaDoorman();
+        User anton = new User(18);
+        Movie movie = new Movie("Horror of the Code");
+
+        assertFalse(doorman.isUserAllowedToVisitTheMovie(anton, movie));
     }
 }
```

Sicherheitsnetz verbessern

- Unittest

```
Working on feature-envy: Improve unit test

A minimal-viable unit test. Normally, more
tests are needed for a better
safety-net - for the purpose of this
lecture unit, this single unit test
is barely-sufficient.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

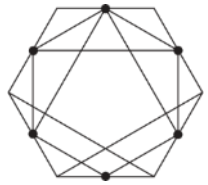
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
index 11541a9..389248e 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
@@ -16,8 +16,7 @@
     }

     boolean isUserAllowedToVisitTheMovie(User anton, Movie movie) {
-        if (anton.getWallet()
-                .getAmount() >= movie.getEntryFee()) {
+        if (anton.canPay(movie.getEntryFee())) {
             if (anton.getAge() >= movie.getAgeRate()) {
                 return true;
             }

diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/User.java
index 7f8a9d8..6c0e95e 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
@@ -19,4 +19,9 @@
     public String getName() {
         return "anton";
     }
+
+    public boolean canPay(int entryFee) {
+        // TODO Auto-generated method stub
+        return false;
+    }
 }
```

Code verbessern

- Code sprechend machen

```
Working on feature-envy: Remove feature-
envy statement

State the actual intent of the statement
and implement stubs.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Money.java
new file mode 100644
index 0000000..a9a355d
--- /dev/null
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java
@@ -0,0 +1,13 @@
+package de.codingakademie.featureenvy;
+
+import java.math.BigDecimal;
+
+public class Money {
+    private final BigDecimal value;
+    private final String currency;
+
+    public Money(String value, String currency) {
+        this.value = new BigDecimal(value);
+        this.currency = currency;
+    }
+}
```

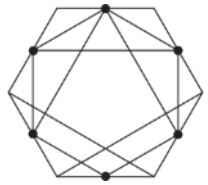
Code verbessern

- Code sprechend machen
- Konzept „Money“ einführen

1. Teil des commit

```
Working on feature-envy: Introduce class
Money for concept money

Realize,
CinemaDoorman.isUserAllowedToVisitTheMovie
is talking about
money, but up until this point the code
uses Integer. The concept of
money is better represented by the class
Money.
For now, introduce the class Money without
any connection to actual
active code. This way, we "stay green" and
don't break any currently
deployed code.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Movie.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Movie.java
index 3ce792c..079ac8f 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Movie.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Movie.java
@@ -17,6 +17,10 @@
          return 12;
      }

+     public Money getEntryFeeMoney() {
+         return new Money("12.00", "EUR");
+     }
+
      public String getName() {
          return name;
      }
```

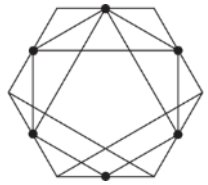
Code verbessern

- Code sprechend machen
- Konzept „Money“ einführen

2. Teil des commit

```
Working on feature-envy: Introduce class
Money for concept money

Realize,
CinemaDoorman.isUserAllowedToVisitTheMovie
is talking about
money, but up until this point the code
uses Integer. The concept of
money is better represented by the class
Money.
For now, introduce the class Money without
any connection to actual
active code. This way, we "stay green" and
don't break any currently
deployed code.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
index 35d63a9..4646776 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
@@ -1,6 +1,7 @@
 package de.codingakademie.featureenvy;

 public class Wallet {
+    private Money value = new Money("0.00", "EUR");

     public int getAmount() {
         // TODO Auto-generated method stub
```

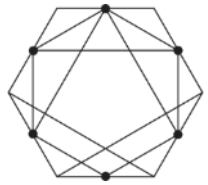
Code verbessern

- Code sprechend machen
- Konzept „Money“ einführen

3. Teil des commit

```
Working on feature-envy: Introduce class
Money for concept money

Realize,
CinemaDoorman.isUserAllowedToVisitTheMovie
is talking about
money, but up until this point the code
uses Integer. The concept of
money is better represented by the class
Money.
For now, introduce the class Money without
any connection to actual
active code. This way, we "stay green" and
don't break any currently
deployed code.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

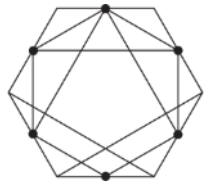
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
index 389248e..d72318b 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
@@ -16,7 +16,7 @@
     }

     boolean isUserAllowedToVisitTheMovie(User anton, Movie movie) {
-        if (anton.canPay(movie.getEntryFee())) {
+        if (anton.canPay(movie.getEntryFeeMoney())) {
             if (anton.getAge() >= movie.getAgeRate()) {
                 return true;
             }

diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/User.java
index 6c0e95e..008e0dd 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
@@ -24,4 +24,9 @@
         // TODO Auto-generated method stub
         return false;
     }
+
+    public boolean canPay(Money price) {
+        // TODO Auto-generated method stub
+        return false;
+    }
 }
```

Code verbessern

- Konzept Money verwenden

```
Working on feature-envy: Switch logic to
use Money in isUserAllowedToVisitTheMovie
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

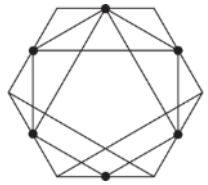
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Money.java
index a9a355d..4d98e8f 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java
@@ -10,4 +10,11 @@
          this.value = new BigDecimal(value);
          this.currency = currency;
      }
+
+    public Money(Money money) {
+        // for immutable classes: rule construct your own copy
+        // here: BigDecimal and String is immutable and the assignment operator creates an independent copy
+        this.value = money.value;
+        this.currency = money.currency;
+    }
 }
```

Code verbessern

- Klasse Money für Verwendung vorbereiten

```
Working on feature-envy: Added ability to
copy Money
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

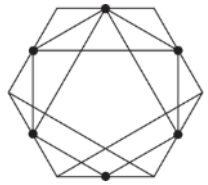
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
index 4646776..3358064 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
@@ -3,6 +3,13 @@
 public class Wallet {
     private Money value = new Money("0.00", "EUR");

+    public Wallet() {
+    }
+
+    public Wallet(Money initialWalletFund) {
+        value = new Money(initialWalletFund);
+    }
+
     public int getAmount() {
         // TODO Auto-generated method stub
         return 0;
```

Code verbessern

- Money in Wallet integrieren

```
Working on feature-envy: Added ability to
Wallet to fund with more than no money
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

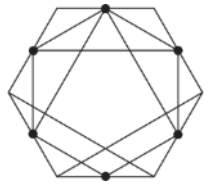
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/User.java
index 008e0dd..505e0cd 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
@@ -1,13 +1,18 @@
 package de.codingakademie.featureenvy;

 public class User {
-    private final Wallet wallet = new Wallet();
+    private Wallet wallet = new Wallet();
     private int age;

     User(int age) {
         this.age = age;
     }

+    User(int age, Money initialWalletFund) {
+        this.age = age;
+        this.wallet = new Wallet(initialWalletFund);
+    }
+
     public int getAge() {
         return age;
     }
```

Sicherheitsnetz verbessern

- Nutzern die Möglichkeit geben eine unterschiedlich gefüllt Brieftasche zu geben
- Dies dient der Vorbereitung der Verbesserung der Unittest

```
Working on feature-envy: Added ability to
User to have a non-empty Wallet
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

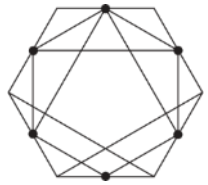
```
diff --git a/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java b/feature-
envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
index d3a99a8..9bb7a02 100644
--- a/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
+++ b/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
@@ -13,4 +13,24 @@

         assertFalse(doorman.isUserAllowedToVisitTheMovie(anton, movie));
     }
+
+    @Test
+    void deny_entry_if_visitor_does_have_insufficient_funds() {
+        CinemaDoorman doorman = new CinemaDoorman();
+        User anton = new User(18);
+        Movie movie = new Movie("Horror of the Code");
+
+        assertFalse(doorman.isUserAllowedToVisitTheMovie(anton, movie));
+    }
+
+    // @Test
+    // void permit_entry_if_visitor_does_have_sufficient_funds() {
+    // CinemaDoorman doorman = new CinemaDoorman();
+    // User anton = new User(18, new Money("20.00", "EUR"));
+    // Movie movie = new Movie("Horror of the Code");
+    //
+    // TODO missing logic prevents from setting this test to active
+    //
+    // assertTrue(doorman.isUserAllowedToVisitTheMovie(anton, movie));
+    // }
 }
```

Sicherheitsnetz verbessern

- Zwei Unittest hinzufügen
- Der zweite auskommentiert, weil die Codeunterstüzung noch fehlt – dieser ist nun Design-Richtlinie für den Code

```
Working on feature-envy: Improve the
safety-net by adding unit test
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Money.java
index 4d98e8f..d95ea60 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java
@@ -17,4 +17,10 @@
         this.value = money.value;
         this.currency = money.currency;
     }
+
+    public BigDecimal getValue() {
+        // for immutable classes: rule construct a copy of a value before returning
+        // here: since BigDecial is already immutable, returning plainly is okay.
+        return value;
+    }
 }
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/User.java
index 505e0cd..3dfbde0 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
@@ -31,7 +31,6 @@
     }

     public boolean canPay(Money price) {
-        // TODO Auto-generated method stub
-        return false;
+        return wallet.contains(price);
     }
 }
```

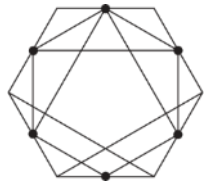
Sicherheitsnetz verbessern

- Unittest mit Wallet
- Aktivieren des Unittests nachdem die notwendige Codeunterstützung existiert

1. Teil des commit

```
Working on feature-envy: Implement
permit_entry_if_visitor_does_have_sufficie
nt_funds

In order to have this test green, we
implement minimal sufficient logic,
and for training-purpose we introduce
another feature-envy situation in
Wallet.contains(Money price).
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
index 3358064..05961be 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
@@ -15,4 +15,9 @@
          return 0;
      }

+     public boolean contains(Money price) {
+         return value.getValue()
+                     .compareTo(price.getValue()) >= 0;
+     }
+
 }
```

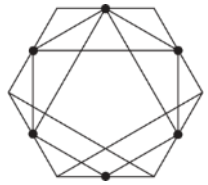
Sicherheitsnetz verbessern

- Unittest mit Wallet
- Aktivieren des Unittests nachdem die notwendige Codeunterstützung existiert

2. Teil des commit

```
Working on feature-envy: Implement
permit_entry_if_visitor_does_have_sufficie
nt_funds

In order to have this test green, we
implement minimal sufficient logic,
and for training-purpose we introduce
another feature-envy situation in
Wallet.contains(Money price).
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

```
diff --git a/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java b/feature-
envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
index 9bb7a02..aeb6685 100644
--- a/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
+++ b/feature-envy/src/test/java/de/codingakademie/featureenvy/CinemaDoormanShould.java
@@ -1,6 +1,7 @@
 package de.codingakademie.featureenvy;

 import static org.junit.jupiter.api.Assertions.assertFalse;
+import static org.junit.jupiter.api.Assertions.assertTrue;

 import org.junit.jupiter.api.Test;

@@ -23,14 +24,12 @@
         assertFalse(doorman.isUserAllowedToVisitTheMovie(anton, movie));
     }

-    // @Test
-    // void permit_entry_if_visitor_does_have_sufficient_funds() {
-    // CinemaDoorman doorman = new CinemaDoorman();
-    // User anton = new User(18, new Money("20.00", "EUR"));
-    // Movie movie = new Movie("Horror of the Code");
-    //
-    // TODO missing logic prevents from setting this test to active
-    //
-    // assertTrue(doorman.isUserAllowedToVisitTheMovie(anton, movie));
-    // }
+    @Test
+    void permit_entry_if_visitor_does_have_sufficient_funds() {
+        CinemaDoorman doorman = new CinemaDoorman();
+        User anton = new User(18, new Money("20.00", "EUR"));
+        Movie movie = new Movie("Horror of the Code");
+
+        assertTrue(doorman.isUserAllowedToVisitTheMovie(anton, movie));
+    }
 }
```

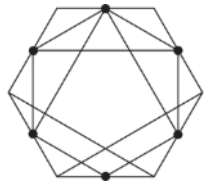
Sicherheitsnetz verbessern

- Unittest mit Wallet
- Aktivieren des Unittests nachdem die notwendige Codeunterstützung existiert

3. Teil des commit

```
Working on feature-envy: Implement
permit_entry_if_visitor_does_have_sufficie
nt_funds

In order to have this test green, we
implement minimal sufficient logic,
and for training-purpose we introduce
another feature-envy situation in
Wallet.contains(Money price).
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

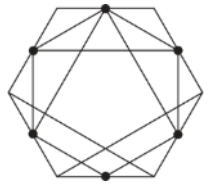
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
index 05961be..f16ec27 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
@@ -10,14 +10,8 @@
         value = new Money(initialWalletFund);
     }

-    public int getAmount() {
-        // TODO Auto-generated method stub
-        return 0;
-    }
-
     public boolean contains(Money price) {
         return value.getValue()
                     .compareTo(price.getValue()) >= 0;
     }
-
 }
```

Code verbessern

- Entferne nicht mehr verwendeten Code

```
Working on feature-envy: Remove unused
method from Wallet
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

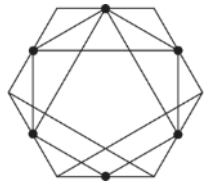
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/User.java
index 3dfbde0..0667b0d 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
@@ -25,11 +25,6 @@
         return "anton";
     }

-    public boolean canPay(int entryFee) {
-        // TODO Auto-generated method stub
-        return false;
-    }
-
     public boolean canPay(Money price) {
         return wallet.contains(price);
     }
```

Code verbessern

- Entferne nicht mehr verwendeten Code

```
Working on feature-envy: Remove unused
method from User
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

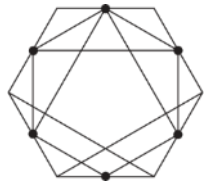
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Movie.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Movie.java
index 079ac8f..612f307 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Movie.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Movie.java
@@ -13,10 +13,6 @@
         return minimumAge;
     }

-    public int getEntryFee() {
-        return 12;
-    }
-
     public Money getEntryFeeMoney() {
         return new Money("12.00", "EUR");
     }
```

Code verbessern

- Entferne nicht mehr verwendeten Code

```
Working on feature-envy: Remove ununsed
method from Movie
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

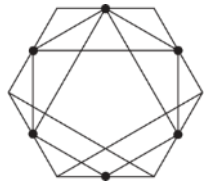
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
index d72318b..f4c68fc 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
@@ -16,7 +16,7 @@
     }

     boolean isUserAllowedToVisitTheMovie(User anton, Movie movie) {
-        if (anton.canPay(movie.getEntryFeeMoney())) {
+        if (anton.canPay(movie.entryFee())) {
             if (anton.getAge() >= movie.getAgeRate()) {
                 return true;
             }
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Movie.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Movie.java
index 612f307..5d0099b 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Movie.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Movie.java
@@ -13,7 +13,7 @@
         return minimumAge;
     }

-    public Money getEntryFeeMoney() {
+    public Money entryFee() {
         return new Money("12.00", "EUR");
     }
```

Code verbessern

- Methodennamen im Code verbessern und nicht mehr verwendeten Code entfernen

```
Working on feature-envy: Rename method in
Movie

Note: As the "other" fee method in Movie
has been removed, we can rename the
renaming method. Now, the condition in
CinemaDoorman.isUserAllowedToVisitTheMovie
() sounds more natural.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
index f4c68fc..b340d36 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
@@ -7,8 +7,7 @@
         User anton = new User(18);
         Movie movie = new Movie("Horror of the Code");

-        boolean userIsAllowToVisitTheMovie = false;
-        userIsAllowToVisitTheMovie = isUserAllowedToVisitTheMovie(anton, movie);
+        boolean userIsAllowToVisitTheMovie = isUserAllowedToVisitTheMovie(anton, movie);

         if (userIsAllowToVisitTheMovie) {
             System.out.println(anton.getName() + " visits " + movie.getName());
```

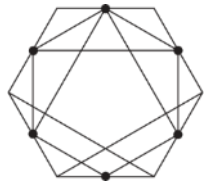
Code verbessern

- Refactoring: eine explizite Variable bringt hier keine Übersichtlichkeit und keinen anderen Vorteil → inline (Teil 1)

Erster Teil des Refactoring

```
Working on feature-envy: Refactor - inline
variable
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
index b340d36..13394cf 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/CinemaDoorman.java
@@ -7,9 +7,7 @@
         User anton = new User(18);
         Movie movie = new Movie("Horror of the Code");

-        boolean userIsAllowToVisitTheMovie = isUserAllowedToVisitTheMovie(anton, movie);
-
-        if (userIsAllowToVisitTheMovie) {
+        if (isUserAllowedToVisitTheMovie(anton, movie)) {
             System.out.println(anton.getName() + " visits " + movie.getName());
         }
     }
```

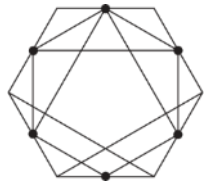
Code verbessern

- Refactoring: eine explizite Variable bringt hier keine Übersichtlichkeit und keinen anderen Vorteil → inline (Teil 2)

Zweiter Teil des Refactoring

```
Working on feature-envy: Refactor - inline
variable
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

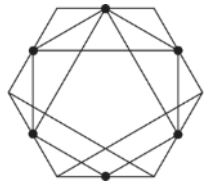
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Money.java
index d95ea60..2ee7bb8 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java
@@ -23,4 +23,8 @@
         // here: since BigDecial is already immutable, returning plainly is okay.
         return value;
     }
+
+    public boolean isGreaterOrEqual(Money otherMoney) {
+        return value.compareTo(otherMoney.value) >= 0;
+    }
 }

diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
index f16ec27..2e8ead8 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Wallet.java
@@ -11,7 +11,6 @@
     }

     public boolean contains(Money price) {
-        return value.getValue()
-                    .compareTo(price.getValue()) >= 0;
+        return value.isGreaterOrEqual(price);
     }
 }
```

Code verbessern

- Die Klasse Money verbessern, damit wird den Code im nächsten Schritt verbessern können
- Vermeidung von feature-envy

```
Working on feature-envy: Remove feature-
envies from Wallet

Adding a comparing function to Money
solves the feature-envy-problem in
Wallet.
To validate: the concept of money supports
comparability of more or less
money by nature, therefore the
introduction of this function is
justified.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

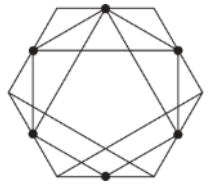
```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/Money.java
index 2ee7bb8..a3d19e3 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/Money.java
@@ -18,12 +18,6 @@
         this.currency = money.currency;
     }

-    public BigDecimal getValue() {
-        // for immutable classes: rule construct a copy of a value before returning
-        // here: since BigDecial is already immutable, returning plainly is okay.
-        return value;
-    }
-
     public boolean isGreaterOrEqual(Money otherMoney) {
         return value.compareTo(otherMoney.value) >= 0;
     }
```

Code verbessern

- Nicht mehr verwendete Methode entfernen
- Damit werden zukünftigen feature-envy Situationen vorgebeugt

```
Working on feature-envy: Remove unused
method from Money

By removal of this method, we effectively
prevent future (accidental)
feature-envy situations.
```

© Coding Akademie München GmbH



SOLUTION IDEA – STEP-BY-STEP

FEATURE ENVY AUFLÖSEN

```
diff --git a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java b/feature-
envy/src/main/java/de/codingakademie/featureenvy/User.java
index 0667b0d..780a618 100644
--- a/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
+++ b/feature-envy/src/main/java/de/codingakademie/featureenvy/User.java
@@ -17,10 +17,6 @@
         return age;
     }

-    public Wallet getWallet() {
-        return wallet;
-    }
-
     public String getName() {
         return "anton";
     }
```

Code verbessern

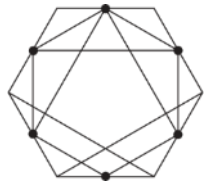
- Nicht mehr verwendete Methode entfernen
- Damit werden zukünftigen feature-envy Situationen vorgebeugt

Letzter Commit dieses Refactorings
Dieses Refactoring hat sich über 21 Commits erstreckt

```
Working on feature-envy: Remove unused
method from User

Reducing the risk of future feature-
envyness towards User.
```

© Coding Akademie München GmbH

Photo by Ariel on Unsplash



WAS IST CODE COVERAGE?

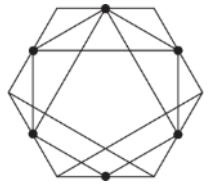
Eine Code Coverage Analyse/Darstellung zeigt, welcher Code zur Ausführung kommt.

- Welcher Code wurde überhaupt nicht ausgeführt?
- Welcher wurde sehr häufig ausgeführt?

Mit Hilfe von Code Coverage ist es möglich **Aufwände** für das Erstellen von Unittest **zu lenken**.

Mit Hilfe von Code Coverage ist es möglich **gezielt** refactoring **zu planen und durchzuführen**.

© Coding Akademie München GmbH

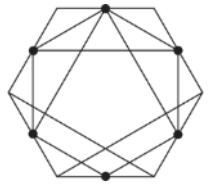


EINFACHE CODE COVERAGE

```
public class Simple {
    public int doMagic(int a, int b) {
        if (a > 100) {
            return b * 9;
        } else {
            return a * 3 + b * 5;
        }
    }
}
```

- Welche branches gibt es?
- Diesen Code in die IDE abtippen
- Den Code zur Ausführung bringen
- Code Coverage Tool (ggf. in der IDE) aktivieren und die branches anschauen
- Mehrere Unittest schreiben, welche alle branches abgedeckt sind

© Coding Akademie München GmbH



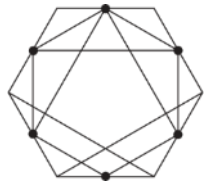
./excercise/trip-service-kata/

SCHREIBE EINEN UNITTEST MITHILFE VON CODE COVERAGE

```
public class TripService {

  public List<Trip> getTripsByUser(User user) throws UserNotLoggedInException {
    List<Trip> tripList = new ArrayList<Trip>();
    User loggedUser = UserSession.getInstance().getLoggedUser();
    boolean isFriend = false;
    if (loggedUser != null) {

…
```

1. Finde den **kürzesten** branch
2. Schreibe hierfür einen Test

© Coding Akademie München GmbH



./excercise/trip-service-kata/

REFACTORING MITHILFE VON CODE COVERAGE

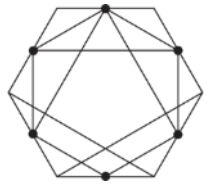
```
public class TripService {

  public List<Trip> getTripsByUser(User user) throws UserNotLoggedInException {
    List<Trip> tripList = new ArrayList<Trip>();
    User loggedUser = UserSession.getInstance().getLoggedUser();
    boolean isFriend = false;
    if (loggedUser != null) {

…
```

1. Finde den **tiefsten** branch
2. Betrachte die Funktionsweise und die Absicht
3. Ist ein Refactoring auf dieser Ebene sinnvoll oder auf einer der Ebenen darüber?

Hinweis

- In der tiefsten Ebene sind häufig feature-envies anzutreffen

© Coding Akademie München GmbH



VORGEHEN, VORTEILE UND NACHTEILE

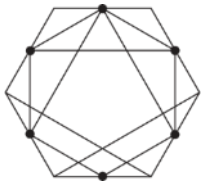
Vorteile

- Test: **Kürzester** Branch zuerst → (normalerweise) einfachster Fall zuerst
- Refactoring: **Tiefster** Branch zuerst → Verbesserung der Klassen und Methoden, Fall-nach-Fall
- Durch schreitweise Behandlung von Fällen vereinfacht sich der Code und verbessert sich das Sicherheitsnetz

Nachteile

- Gegebenenfalls „zu“ komplex weil Implementierungsfehler und Implementierungsfreiheitsgrad übernommen werden, die fachlich nicht korrekt sind
- Gefahr einer zu engen Kopplung von Tests an aktuellen Code
  → Erfordert Mut **Codeverbesserungen** den **Vorrang** vor Menge an Tests zu geben

© Coding Akademie München GmbH



WIE GUT IST UNSERE TEST SUITE?

Eine hohe Code Coverage ist kein Garant dafür, dass alle Fälle abgetest worden sind.

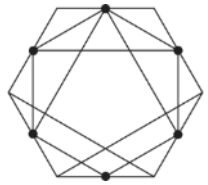
Fachliche Tests inkl. menschlicher Überprüfung sind unabdingbar → Mit **Specification by Example** können menschliche Experten effektiv in die Entwicklung und Überprüfung eingebunden werden.

Ein **mutation testing** Tool modifiziert den zu testenden Code und prüft, ob diese Modifikation (=Mutant) durch die Tests entdeckt worden sind. Falls nicht, überlebt der Mutant und wird gemeldet.

Mit mutation testing wird der Effektivität der Unittests überprüft.

Wird üblicherweise bei **sicherheitskritischen new code** angewandt, weniger bei legacy code.

© Coding Akademie München GmbH



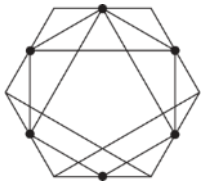
MUTATION TESTING

```
<build>
  <plugins>
    <plugin>
      <groupId>org.pitest</groupId>
      <artifactId>pitest-maven</artifactId>
      <version>1.6.7</version>

      <dependencies>
        <dependency>
          <groupId>org.pitest</groupId>
          <artifactId>pitest-junit5-plugin</artifactId>
          <version>0.14</version>
        </dependency>
      </dependencies>
    </plugin>
  </plugins>
</build>
```

- `mvn clean test org.pitest:pitest-maven:mutationCoverage`
- Ergebnisse in `target\pit-reports`
- Implementiere Tests, welche alle Mutanten entdecken

© Coding Akademie München GmbH

Photo by Javier Allegue Barros on Unsplash



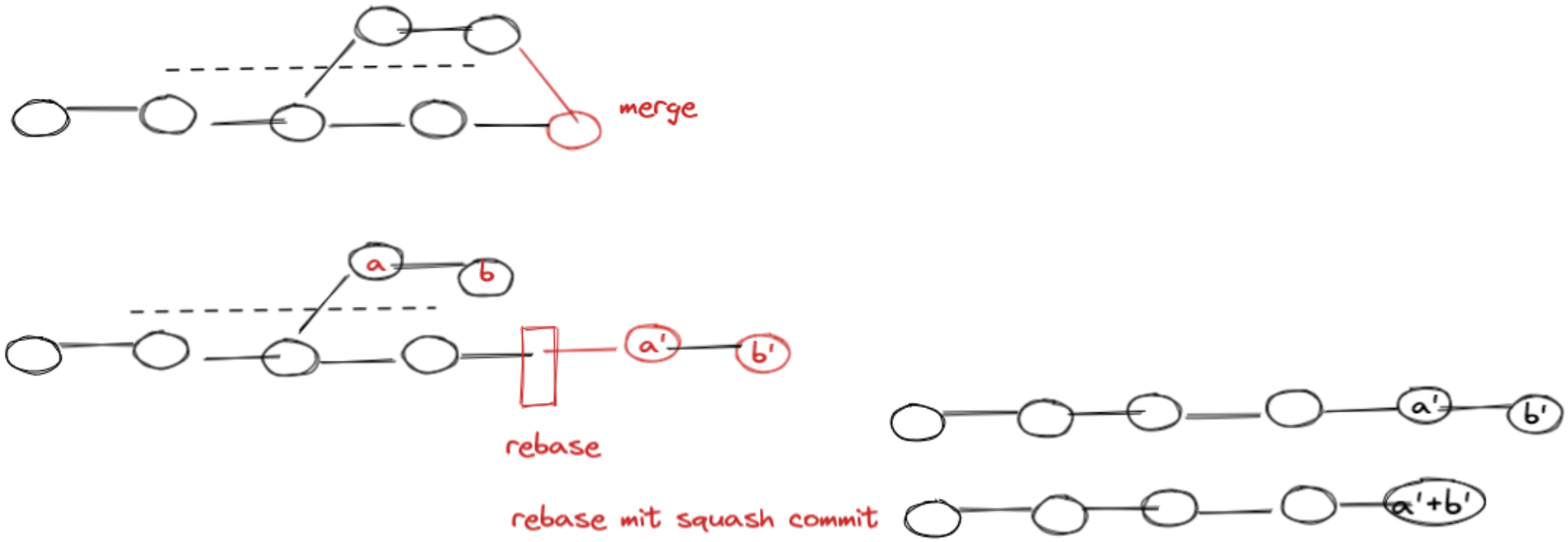
GILDED ROSE

```
public class BuildSeamsInto {
…
    private void main() throws IOException {
…
```

© Coding Akademie München GmbH

© Coding Akademie München GmbH

GIT: MERGE, REBASE UND SQUASH COMMIT



Rebase

- Rebase erzeugt eine einfach (linear) lesbare Projekt/git history

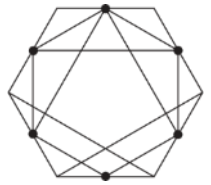
Merge

- Merge erhält branches und deren Historie
- Die Projekt/git history wird damit nicht-linear und komplexer

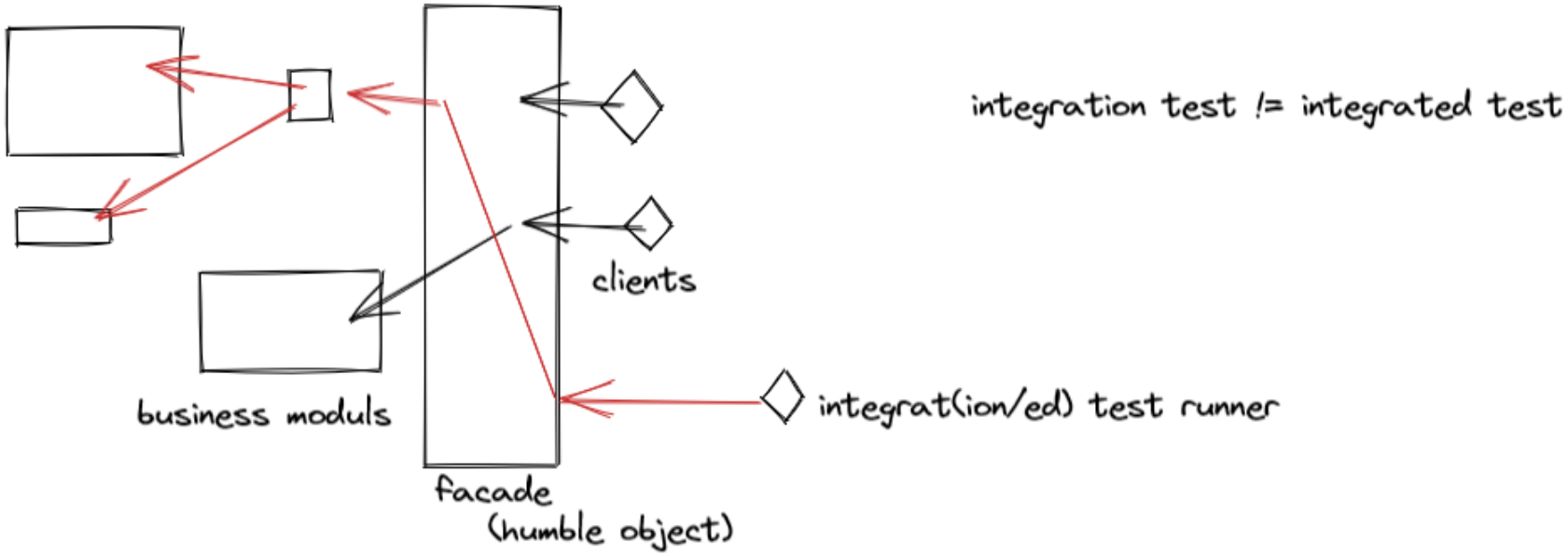
Squash commit

- Mit squash commit können nicht relevante Zwischen-schritte zusammen gezogen werden

© Coding Akademie München GmbH



SICHERHEITSNETZ MIT EINER FASSADE (ZIEL: REFACTORING/REDESIGN)



Ideen für Refactoring und Redesign

- „Extract Class“ → Zuviel Funktionalität aus der Fassade entfernen und in eine eigene Klasse extrahieren. Diese Detail-Klasse kann gezielt und qualitätsgesichert getestet werden.
- Sobald die Fassade keine spezifische Funktionalität mehr enthält, durch eine „dumme“ Bibliothek ersetzten. Diese soll eine einfache Transformation von links nach rechts durchführen.

Sicherheitsnetz aufbauen

- Erstelle Arbeitshypothese: Ich will einen Test, der einen bestimmten Pfad in Komponente A berührt
- Erstelle einen integrated test, welche diesen Pfad berührt und bestätige dies durch CodeCoverage, Logs oder ähnliches
- Reichere den integrated test hinreichend an (ggf. mit mutation test prüfen)

Refactoring und Redesign

- … bei ständigem grün des integrated test

Annahme

- Fassade ist ein HumbleObject und muss als solches nicht besonderes beachtet werden

© Coding Akademie München GmbH

© Coding Akademie München GmbH