**Lernjournal**

**Java Zertifizierung**

****

|  |  |
| --- | --- |
| Verfasser | Severin Müller |
| Dozent | Michael Reiser |
| Modul | Programmieren |
| Kurs | “Java Zertifizierung” |
| Erscheinungsjahr | 2014 |

**inhalt**

# **1. Einleitung**

Die Programmiersprache Java erlangte in den letzten Jahren eine grosse Bedeutung in der Software-Entwicklung. Der riesige Bibliothek sowie die Tatsache, dass Java den Programmierer zwingt, objektorientiert zu denken und zu entwickeln trug sicherlich dazu bei.

Im Kurs Java-Zertifizierung wurden wir intensiv auf eine allfällige Java-Zertifizierungsprüfung vorbereitet. Es wurden alle wichtige Themen für die Programmiersprache

Im Lernjournal ging es darum, das gelernte in die Praxis umzusetzen. Da ich als Software-Entwickler tätig bin, bot es sich an, eine Komponente aus meinem Berufsalltag ins Lernjournal einfliessen zu lassen.

Alle wichtigen Inhalte die in diesem Kurs vermittelt oder repetiert wurden kamen hier zum Einsatz.

# **2. MESPAS AG**

## **2.1 Vorstellung MESPAS AG**

Mein Arbeitgeber ist die Firme MESPAS AG. Wir entwickeln Software für Reedereien und Schiffe. Wir haben dafür eine Cloud Lösung geschaffen, die es unseren Kunden erlaubt, die Daten zentral bei uns zu lagern und eine Synchronisation zwischen Reederei (Office) und den Schiffen (Vessel) zu ermöglichen.

## **2.2 Genereller Use Case**

## **2.3 Use Case Einkauf**

Einer der wenn nicht der wichtigste Use Case in unserer Software ist der Einkauf. Mit unserer Lösung ist es möglich, den gesamten Bestellvorgang vom Schiff, das Ersatzteile benötigt bis hin zum Lieferanten, der diese liefert online und mit unseren Clients abzuwickeln. Der Prozess sieht überlicherweise so aus (vereinfacht):

1. Das Schiff benötigt Ersatzteile

2. Der zuständige Mitarbeiter auf dem Schiff erfasst im Vessel-Client einen Purchase Request (PR).

3. Der Zuständige Vorgesetzte gibt diesen PR frei.

4. Das Schiff synrchonisiert über Satellit mit unserer Cloud

5. Die Redderei erhält den PR im Office Client und bewilligt diesen.

6. Der zuständige Einkäufer erstellt eine Offertanfrage (Request for Quotation, RFQ) und schickt diese an Lieferanten, die das benötigte Teil liefern.

7. Der Lieferant (Supplier) erhält den RFQ im Supplier Client und platziert sein Angebot.

8. Das Angebot wird angenommen und eine Bestellung (Order) wird versendet.

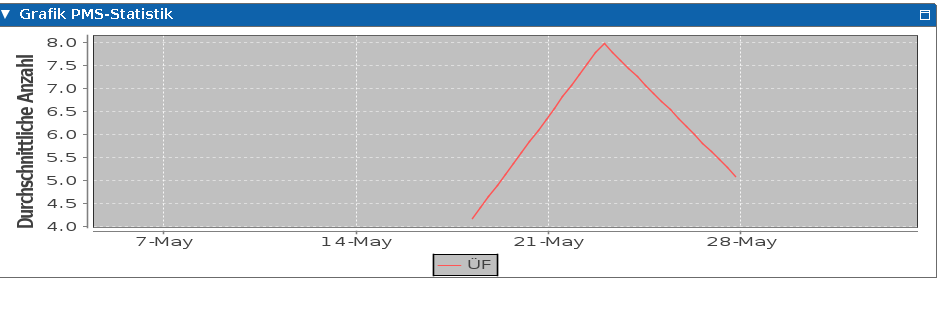
9. Der Lieferant liefert die Ware und das Schiff betätigt den Erhalt.

Der gesamte Vorgang stellt einen sogenannten **Procurement Case** dar.

## **2.4 Dashboard und Gadgets**

Weil Manager und leitende Angestellt sich einen schnellen Überblick über bestimmte Situationen machen möchten haben wir ein Dashboard entwickelt, das verschiedene Gadget anzeigen kann.

Beispiele können so Wartungsdaten von der Flotte angezeigt werden:

Dieses Gadget zeigt die offenen Wartungsarbeiten an die ausgeführt werden müssen. 

Für dieses Dashboard habe ich nun ein weiteres Gadget entwickelt, dass sic für das Lernjournal als geeignet herausstellte.

# **3. Lernjournal**

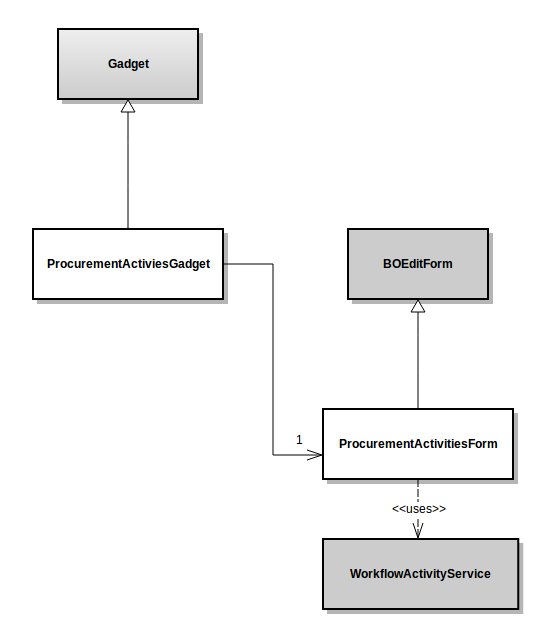
## 3.1 Auftrag

Ich erhielt den Auftrag, ein neues Gadget zu schreiben, das die Einkaufsaktivitäten über die Flotte tabellarisch anzeigt.

Dabei sollen für jeden Einkaufsfall die Objekte in den einzelnen Phasen gezählt werden, also z.B. wieviele Offerten, wieviele Offeranfragen, Wieviele Bestellungen, etc. das flottenweit derzeit aktiv sind.

## 3.2 Requirement Engineering

In der Phase Requirement Engineering hatte ich folgendes Objektmodell erstellt:



## 3.3 Implementierung

Die grau hinterlegten Objekte mussten neu erstellt werden. Ich werde nun im Sinne des Lernjournals erläutern, welche Lerninhalte dass Codeseitig zum tragen kamen.

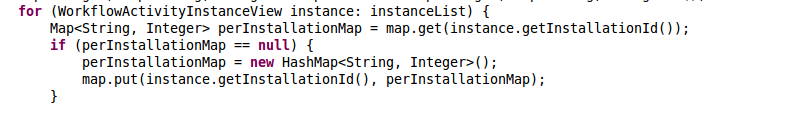
### 3.3.1 Java Basic, Data Types

### 

Wie in jeder Klasse wurden hier verschiedene Datentypen, Konstruktoren und Methoden gebraucht. Da ein Interface implementiert wurden mussten zusätzlich die Methoden überschrieben werden.

### 3.3.2 Operators, Descision Constructs, Arrays

Jede Aktivität in einem Einkaufsvorgang ist eine sogenannte WorkflowActivity. Diese liegt in der Datenbank ab und kann als Business Objekt von dieser geholt werden. Sämtliche Daten die wir benötigen werden über ein Persistance Layer persistiert und abstrahiert. Die perstistierung erfolgt mittels dem Framework **Andromda.**



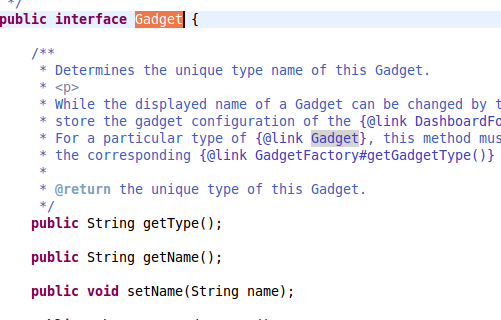
In diesem Code Snippet gehe ich durch eine Liste von Objekten des Typs WorkActivitsInstanceView durch und lege für jedes Element dieser Liste ein neues Element an.

Ich hole für jedes dieser Elemente die Installation Id (die Schiffs Id, für die diese Aktivität gilt) und lege diese in einerm Map ab.

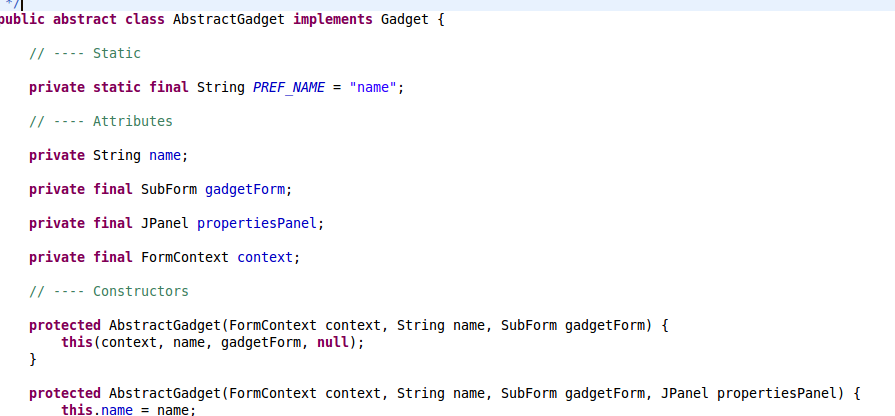
### 3.3.3 Inheritance, Exceptions, GUI

In den vorherigen Unterkapiteln waren Beispiel Schnipsel zu sehen, die mit der Aufbereitung der Daten zu tun haben.

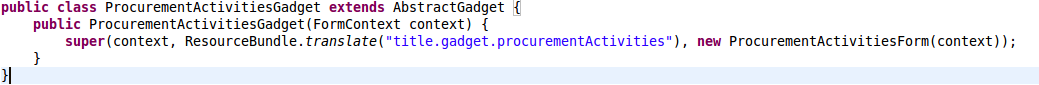
Für das Gadget selber waren andere Klassen notwendig.



Das Interface Gadget muss für jedes neue Gadget implementiert werden. Direkt darunter in der Hierarchy haben wir eine abstrakte Klasse erstellt, um einige Gemeinsamkeiten, die alle Gadgets haben festzuhalten:

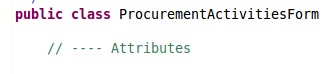


Jedes neue Gadget impleentiert also nicht direkt das Interface sondern erweitert diese abstrakte Klasse:



Dies ist nun die Klasse die ich erstellt habe. Sie macht nichts anders als den Konstruktor der Superklasse mit meinen Optionen aufzurufen.

Die wichtigste Option ist das GUI Element, welches ich dem Kontruktor übergebe. Ich habe für diesen Zweck ein neues Form erstellt:

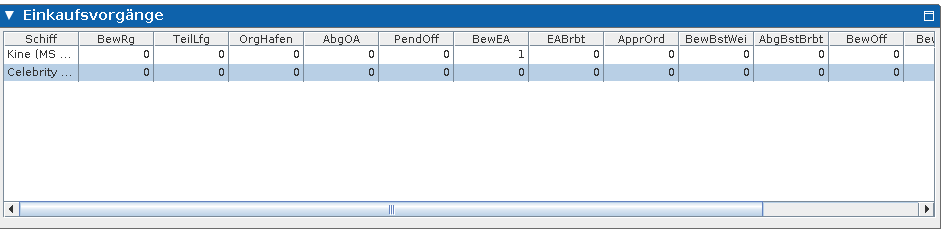


Diese Klasse erweitert unsere abstrakte GUI Klase BOEditForm, welche wiederum AbstractForm erweitert. Diese wiederum erweitert Jpanel und Implementiert Form (Standard Java GUI Klassen/Interface)

In dieses Form kommt eine Tabelle mit entsprechendem Modell:

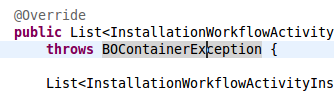
Das Modell habe ich so gestaltet, dass die Aktivitäten nicht statisch als Tabelle definiert sind, sondern dynamisch aus der Datenbank geladen werden für den Fall, dass neue Aktivitäten dazukommen in der Zukunft (Stichwort Wiederverwendbarkeit).

Das fertige GUI sieht dann so aus:



Wenn man über die Spalten mit der Maus “hovert” wird zudem ein Tooltip Text angezeigt in dem der volle Spaltenname angezeigt wird.

Exceptions kommen immer wieder zum tragen:



Das nötige Exception wurde also berücksichtigt.

# **4. Fazit**

Viele der Inhalte des Kurse konnte ich direkt anwenden und ich habe vieles mitgenommen wovon ich in meinem Berufsleben nun profitieren kann. Ich würde den Kurs also wieder machen.