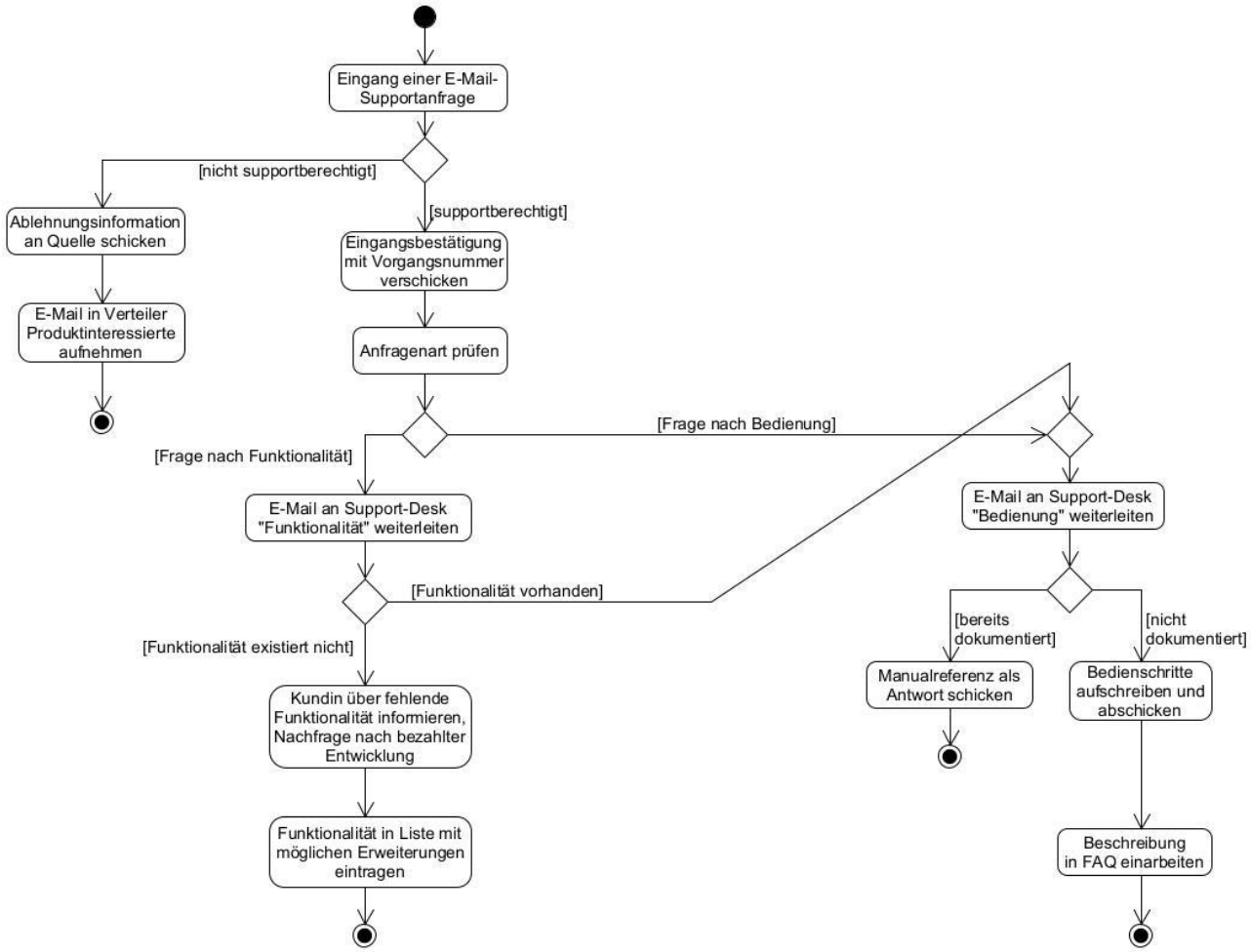
Aufgabe 0.2 (1 Punkt)

Geben Sie das Lösungswort des Quiz aus der Lernnotiz an.

Aufgabe 4 (4 Punkte)

In der folgenden Abbildung ist der Support-Prozess eines IT-Unternehmens für dessen wichtigstes Produkt beschrieben. Das Diagramm kann von der Veranstaltungsseite im Projekt ooadAufgabeAktivitaetsdiagrammerweiterung geladen werden.



Erweitern Sie den Prozess um folgende Abläufe:

a) Wenn eine als Kunde eingetragene Person eine neue Funktionalität sucht, die bereits für das nächste Release geplant ist, soll sie darüber informiert werden.
b) Supportanfragen von nicht supportberechtigten Personen, die Bedienanfragen enthalten, die bisher noch nicht dokumentiert sind, sollen in das zugehörige FAQ-Dokument eingearbeitet werden
c) Anfragen supportberechtigter Personen können auch Hinweise auf potenzielle Fehler enthalten. Dabei ist zunächst zu prüfen, ob es sich um einen echten Fehler oder ein Bedienproblem handelt. Bedienprobleme werden an den Support-Desk „Bedienung“ weitergeleitet. Zu echten Fehlern wird ein Problembericht angelegt, der in die „Next Feature“-Liste integriert wird.
d) Ergänzen Sie für die Aktionen die beteiligten Rollen und Produkte. Dabei sind die Rollen „Supportkoordination“, „Bedienungssupport“ und „Entwicklungssupport“ beteiligt. Überlegen Sie bei den Produkten, wann neue Daten angelegt werden, bzw. Daten in Dokumenten oder Datenbanken verändert werden. Bei Änderungen eines

existierenden Produktes, z. B. beim Eintragen eines Datensatzes in eine Datenbank, wird dieses Produkt mit einer Linie mit Pfeilspitzen an beiden Enden mit der Aktion verbunden. Sie dürfen sich leichte Änderungen der bisherigen Darstellungen überlegen, die die Lesbarkeit des Diagramms etwas erhöhen können.

Aufgabe 5 (2 Punkte)

Neben Aktivitätsdiagrammen gibt es einige andere Notationen für Geschäftsprozesse, die meist weitere graphische Elemente mit zusätzlicher Syntax und Semantik anbieten. Eine Variante ist BPMN (Business Process Model and Notation). Die Abb. 1 zeigt ein BPMN-Diagramm aus der Beispielsammlung zur formalen Referenz von BPMN. Übersetzen Sie das Diagramm möglichst genau mit allen Inhalten in ein Aktivitätsdiagramm. Bei der Übersetzung kann der Standard ([1], Seite 29ff) hilfreich sein.
Von der Veranstaltungsseite kann eine UMLet-Datei mit den Aktionen im Projekt ooadAufgabeBPMNUebersetzung geladen werden.

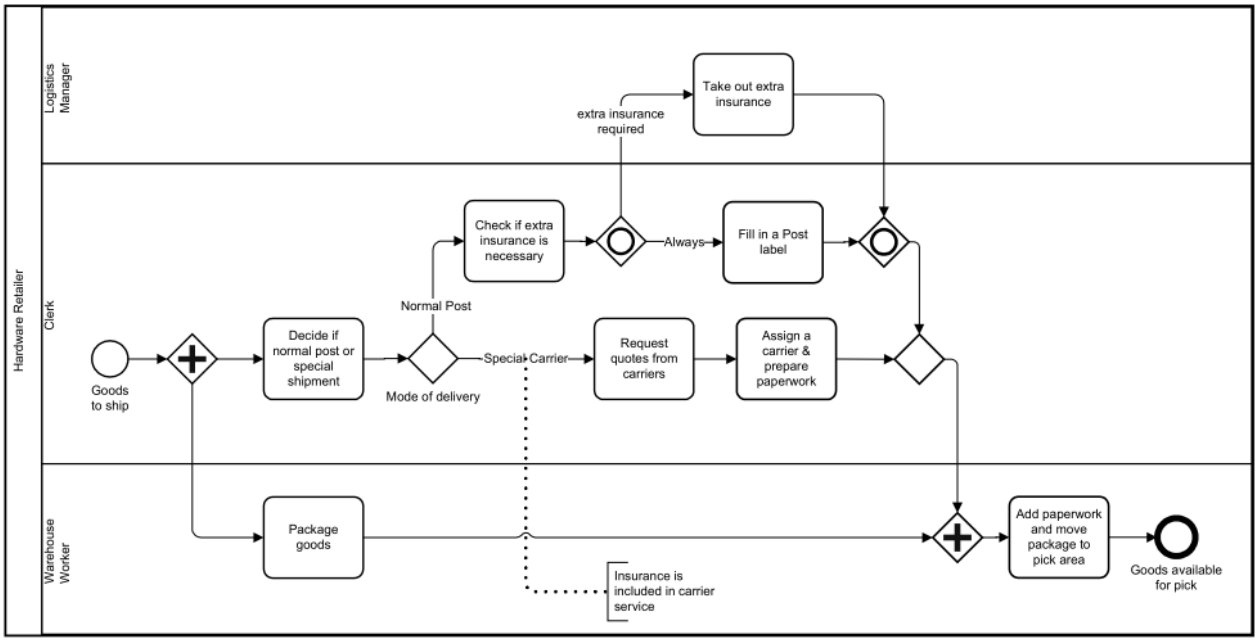


Abb 1.: Beispiel für BPMN-Prozessmodellierung ([2], Seite 3)

[1] OMG, Business Process Model and Notation (BPMN), Version 2.0.2, OMG Document Number: formal/13-12-09, über https://www.omg.org/spec/BPMN/ (letzter Zugriff 23.9.2025), Januar 2014
[2] OMG, BPMN 2.0 by Example, Version 1.0, OMG Document Number: dtc/2010-06-02, http://www.omg.org/cgi-bin/doc?dtc/10-06-02.pdf (letzter Zugriff: 23.9.2025), Juni 2010

Aufgabe 6 (2 Punkte)

Im Projekt ooadAufgabeStaticBeseitigen befindet sich eine zwar korrekte, aber sehr schlechte Lösung für eine minimale Verwaltung von Leistungspunkten. Der folgende Ausschnitt aus dem Nutzungsdialog zeigt die Funktionalität, Eingaben sind umrandet.

```
Verwaltung von Leistungspunkten
0 - Programm beenden
1 - Leistung hinzufuegen
2 - Leistungspunkte fuer Matrikelnummer ausgeben: 1
Matrikelnummer: 42
```

```
Fach: OOAD
Leistungspunkte: 5
0 - Programm beenden
1 - Leistung hinzufuegen
2 - Leistungspunkte fuer Matrikelnummer ausgeben: 1
Matrikelnummer: 42
Fach: SW-Projekt
Leistungspunkte: 10
0 - Programm beenden
1 - Leistung hinzufuegen
2 - Leistungspunkte fuer Matrikelnummer ausgeben: 2
Matrikelnummer: 42
OOAD: 5
SW-Projekt: 10
Hat 15 Leistungspunkte
```

Es gelten u. a. folgende Regeln für gute OO-Programme:

- static-Methoden werden in Objekten nur in Ausnahmefällen genutzt (im Beispiel sind sie nur in der Klasse Eingabe sinnvoll und zum Start als main-Methode notwendig)
- zusammengehörige Daten werden in Objekten einer Klasse verwaltet
- die Verwaltung von mehreren Objekten sollte auf den Standardtypen List, Set und Map basieren
- Objekte, die Ausgaben auf Monitoren machen, sollten keine weitere Funktionalität beinhalten; sie nutzen andere Objekte, um die auszugebenden Daten zu erhalten oder Eingaben zu verarbeiten
- der Start von Java-Programmen findet in einer Klasse Main statt, in deren main-Methode nur das (oder die) wichtige(n) Objekt(e) erstellt und die eigentliche Funktionalität gestartet werden
- generell für Java hilfreich: http://kleuker.iui.hs-osnabrueck.de/querschnittlich/CodingGuidelinesUndGlossar.pdf

a) Schreiben Sie das Programm so um, dass zumindest die genannten Regeln eingehalten werden.
b) Lassen Sie das Programm mit dem Sequencediagrammer laufen, drücken Sie am Anfang auf „Einzelschritte abschalten“, führen Sie die vorher gezeigten Beispieleingaben durch, beenden Sie das Programm mit „0“ und speichern Sie das Sequenzdiagramm ab. Was fällt Ihnen am Sequenzdiagramm auf?

---

zu 0.2:

Quiz Quiz 1 U | Quiz 2 M | Quiz 3 L | Hilfe

Quiz 1: Software-Engineering allgemein

Welche der generellen Aussagen bezüglich des Software-Engineerings sind korrekt?

- [ ] Früher wurde hauptsächlich nach dem Wasserfallmodell gearbeitet.
- [x] Mangelnde Software-Engineering-Kenntnisse führen zu nicht wart- und erweiterbaren Produkten.
- [x] Die UML ist Grundlage des systematischen Entwicklung und Dokumentation von Software.
- [x] Wenn Programmiersprachen den Befehl GTO anbieten, darf er trotzdem nicht genutzt werden.
- [x] Vorgehensmodelle müssen passend zum Projekt und den Kompetenzen der Mitarbeitenden gewählt werden.

Antwort überprüfen

Quiz 2: Stakeholder

Welche der Aussagen bezüglich der Stakeholder sind korrekt?

- [x] Stakeholder sind alle Personen, die Einfluss auf den Projekterfolg haben können.
- [x] Soll ein Produkt von Personen unterschiedlicher Nationalitäten benutzt werden, muss besonders auf die Farbgestaltung und Symbolik geachtet werden.
- [x] Werden personenbezogene Informationen verwaltet, sinn immer für den Datenschutz verantwortliche Personen einzubeziehen.
- [ ] Entscheiden sich die Software-entwickelnden Personen ein Projekt in Java zu realisieren, wird dsas Projekt in Java realisiert.
- [ ] Personen aus der Qualitätssicherung gehören typischerweise nicht zu den relevanten Stakeholdern, da sie das Produkt nicht entwickeln.

Antwort überprüfen

Quiz 3: Ziele

Welche der Aussagen bezüglich der Projektziele sind korrekt?

- [ ] Ziele sollen möglichst generisch formuliert sein, damit sie später an gepasst werden können.
- [ ] Ändern sich nach einem Projektstart die strategischen Ziele eines Unternehmens hat dies keinen Einfluss auf das laufende Projekt.
- [ ] Es ist wichtig in Zielen festzuhalten, was nicht geschehen darf.
- [ ] Ziele sind so zu formulieren, dass festgehalten werden kann, dass das Ziel ganz oder gar nicht erreicht wurde (Messbarkeit).
- [ ] Die Erreichung jedes Ziels muss automatisch überprüfbar sein.

Antwort überprüfen

Zu Aufgabe 4: ooadAufgabeAktivitaetsdiagrammerweiterung

Das nachfolgende Diagramm zeigt nur eine mögliche Lösung. Spannend sind hier eher der Weg und die Diskussionen in Richtung der Lösung. Diskutiert werden könnte z. B.

- statt Bedienanfragen nicht supportberechtigter mit dem vorhandenen Dokumentationszweig zu verknüpfen, könnte man dies getrennt an den gegeben Zweig hängen (Nachteil: duplizierte Aktionen, Vorteil: weniger Entscheidungsrauten); bei einer Verknüpfung muss aufgepasst werden, dass nicht doch eine Mail an nicht supportberichtigte Personen geschickt wird (die eine Aktion mit „aufschreiben“ und „abschicken“ sollte besser aufgeteilt werden)
- alternativ hätte man nicht direkt eine dritte Alternative "Hinweis auf Fehler" nach "Anfragenart prüfen" einbauen und dies z. B. zur Funktionalität zuordnen können

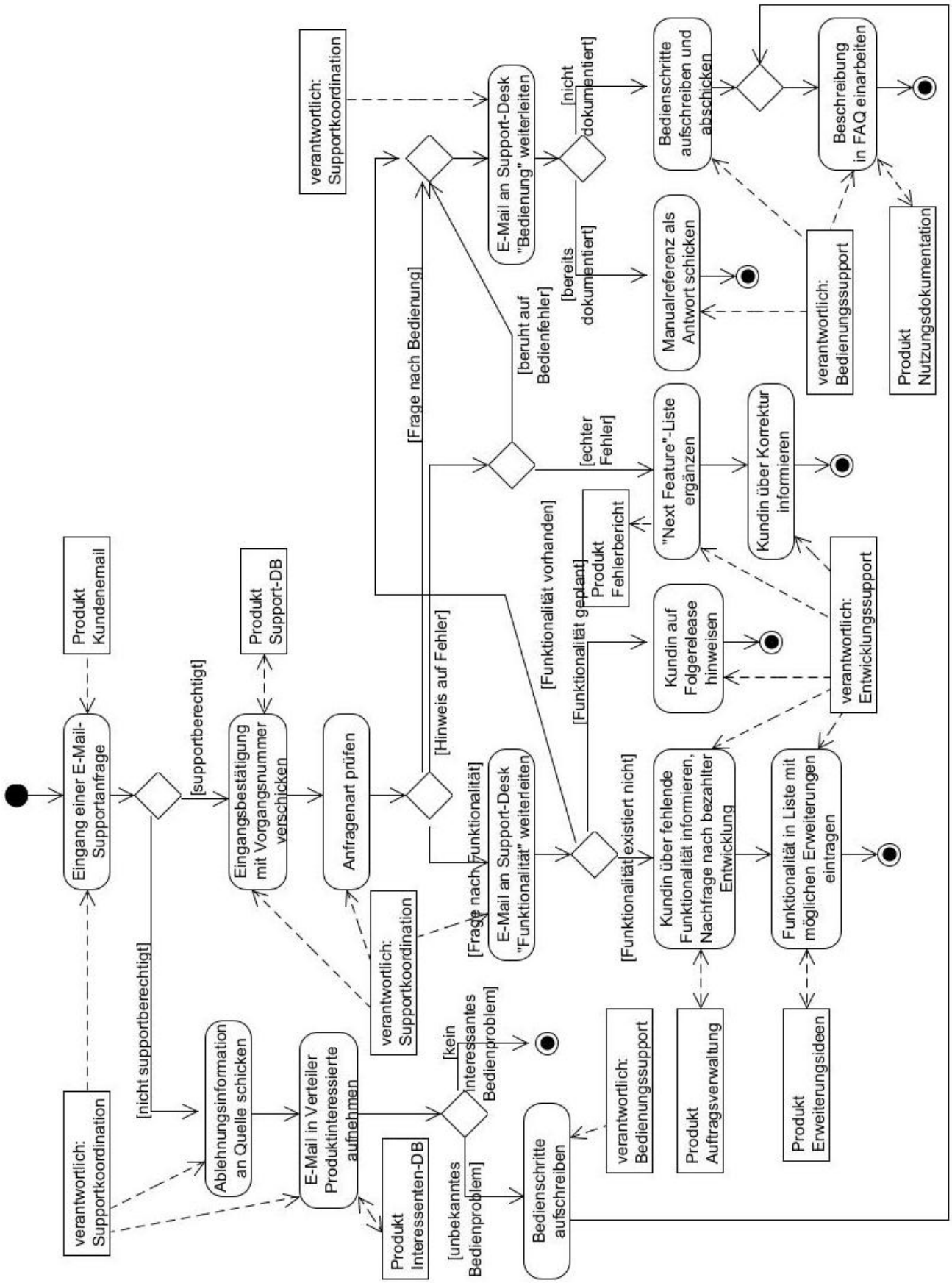
Es sind viele weitere kleinere Varianten möglich, gerade bei den Produkten könnte man mehr ergänzen oder statt mit Doppelpfeilen zu arbeiten nur die Objekte angeben, die erzeugt werden.
Statt alle Aktionen mit Rollen zu markieren, könnte man auch mit farbigen Gebieten pro Rolle arbeiten, Aktionen und Entscheidungen pro Rolle einfärben oder nur Rollen angeben, wenn die Verantwortung wechselt. Es muss dabei eindeutig sein, wer für Entscheidungen verantwortlich ist.

Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing-Wiss. und Informatik
- Software-Entwicklung -

Objektorientierte Analyse und Design

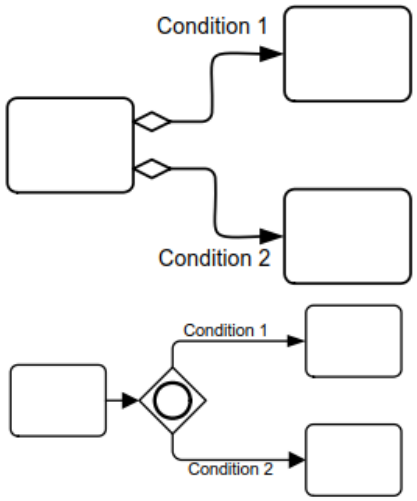
Sommersemester 2026

2. Aufgabenblatt

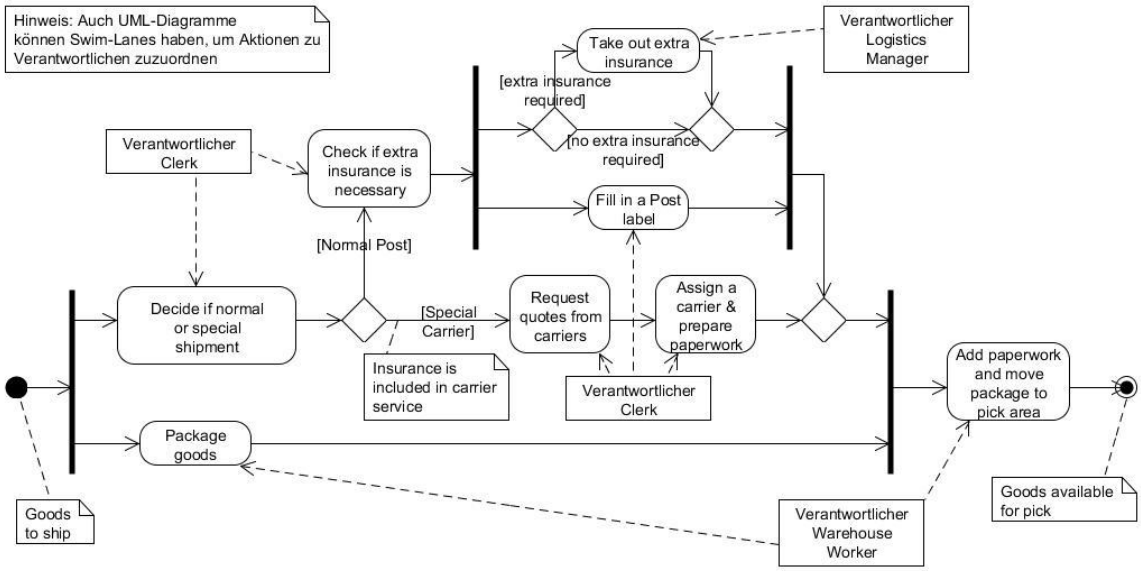


Zu Aufgabe 5: ooadAufgabeBPMNUebersetzung

Interessant ist die Übersetzung der Raute mit einem Kreis drin. Eine Möglichkeit ist es hieraus eine Sequenz zu machen, zunächst den Always-Teil und dann den optionalen Teil als einfache Alternative. Sauberer ist die Übersetzung als Parallelkomposition mit einem Always-Pfad und einem zweiten Ablauf, der optional durchlaufen wird.

Inclusive	This Decision represents a branching point where Alternatives are based on conditional `Expressions` contained within the outgoing Sequence Flows (see page 291). In some sense it is a grouping of related independent Binary (Yes/No) Decisions. Since each path is independent, all combinations of the paths MAY be taken, from zero to all. However, it should be designed so that at least one path is taken. A Default Condition could be used to ensure that at least one path is taken. There are two versions of this type of Decision: • The first uses a collection of conditional Sequence Flows, marked with mini-diamonds (see top-right figure). • The second uses an Inclusive Gateway (see bottom-right picture).	

Business Process Model and Notation (BPMN), v2.0.2 35



Für Entscheidungsrauten wird in diesem Fall festgesetzt, dass die Entscheidung von der Rolle getroffen wird, die verantwortlich für die vorherige Aktion war (ist in diesem Fall eindeutig).

Zu Aufgabe 6

a): ooadAufgabeStaticBeseitigen
Die folgende Lösung ist recht detailliert. Beim Zerlegen sollten zumindest die genannten Regeln eingehalten werden, eine Entitätsklasse mit den Basisdaten (z. B. Leistungspunktzuordnung) und eine Verwaltungsklasse entstehen. Abhängig von der Entitätsklasse bieten sich verschiedene Collections an, dabei ist eine List oft weniger sinnvoll (nicht sinnlos), da doppelte Einträge möglich wären.

```
package main;

import boundary.Boundary;

public class Main {
    public static void main(String[] args) {
        new Boundary().dialog();
    }
}
```

```
package boundary;

import control.Control;
import main.Eingabe;


public class Boundary {
    private Control ctrl = new Control();

    public void dialog(){
        System.out.println("Verwaltung von Leistungspunkten");
        int eingabe = -1;
        while (eingabe != 0) {
            System.out.print(""
                   + "0 - Programm beenden\n"
                   + "1 - Leistung hinzufuegen\n"
                   + "2 - Leistungspunkte fuer Matrikelnummer ausgeben: ");

            eingabe = Eingabe.leseInt();
            if (eingabe == 1) {
                this.leistungspunkteHinzufuegen();
            }
            if (eingabe == 2) {
                this.leistungspunkteBerechnen();
            }
        }
    }

    private void leistungspunkteHinzufuegen() {
        System.out.print("Matrikelnummer: ");
        int matnr = Eingabe.leseInt();
        System.out.print("Fach: ");
```

```
        String fach = Eingabe.leseString();
        System.out.print("Leistungspunkte: ");
        int lp = Eingabe.leseInt();
        this.ctrl.hinzufuegen(matnr, fach, lp);
    }

    public void leistungspunkteBerechnen() {
        System.out.print("Matrikelnummer: ");
        int matnr = Eingabe.leseInt();
        this.ctrl.leistungenVon(matnr).forEach(System.out::println);
        System.out.println("Hat " + this.ctrl.leistunspunkteVon(matnr)
                + " Leistungspunkte");
    }
}
```

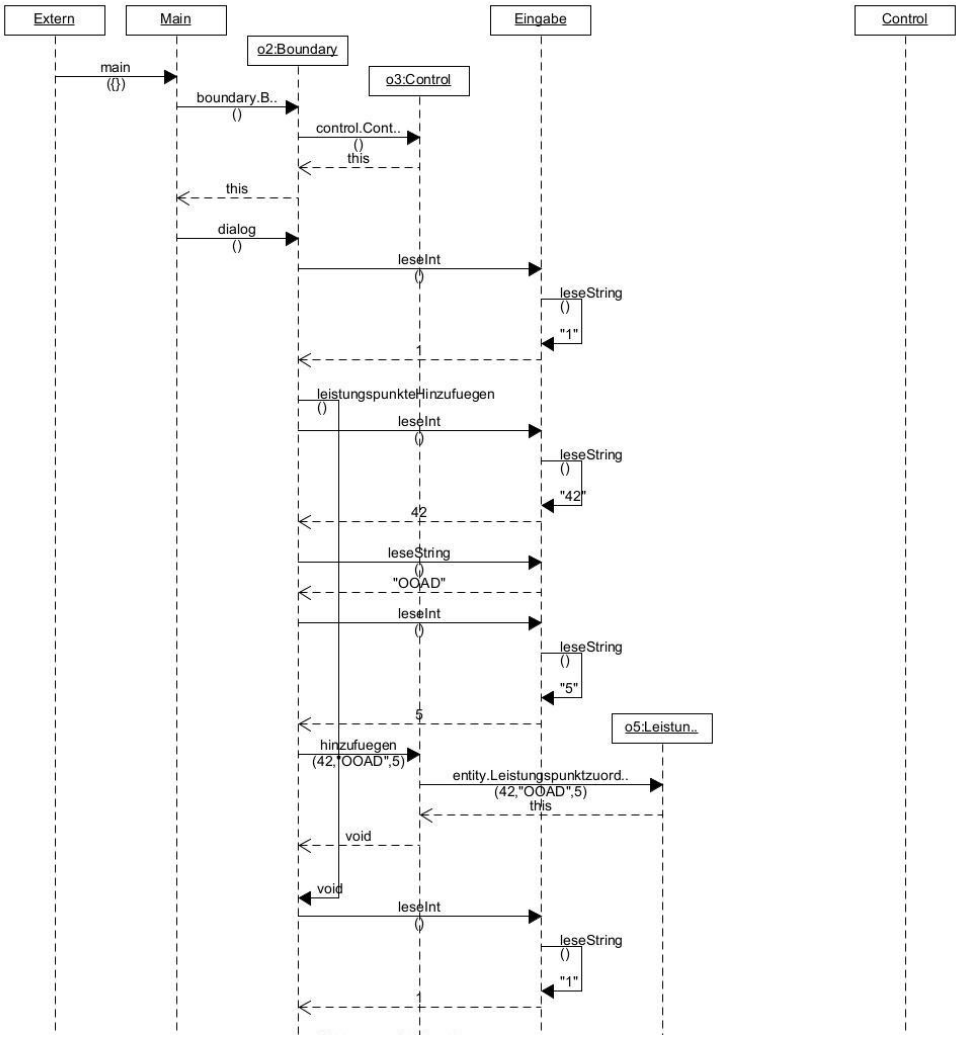
```
package control;

import entity.Leistungspunktzuordnung;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Map;
import java.util.Set;

public class Control {
    private Map<Integer,Set<Leistungspunktzuordnung>> leistungen;

    public Control(){
        this.leistungen = new HashMap<>();
    }

    public void hinzufuegen(int matnr, String fach, int lp){
        Leistungspunktzuordnung neu
                  = new Leistungspunktzuordnung(matnr, fach, lp);
        Set<Leistungspunktzuordnung> tmp = this.leistungen.get(matnr);
        if (tmp == null){
            Set<Leistungspunktzuordnung> set = new HashSet<>();
            set.add(neu);
            this.leistungen.put(matnr,set);
        } else {
            tmp.add(neu);
        }
    }

    public List<String> leistungenVon(int matnr){
        List<String> erg = new ArrayList<>();
        Set<Leistungspunktzuordnung> tmp = this.leistungen.get(matnr);
        if (tmp == null){
            return erg;
        }
        tmp.forEach(l -> erg.add(l.getFach() +": " + l.getLp()));
```

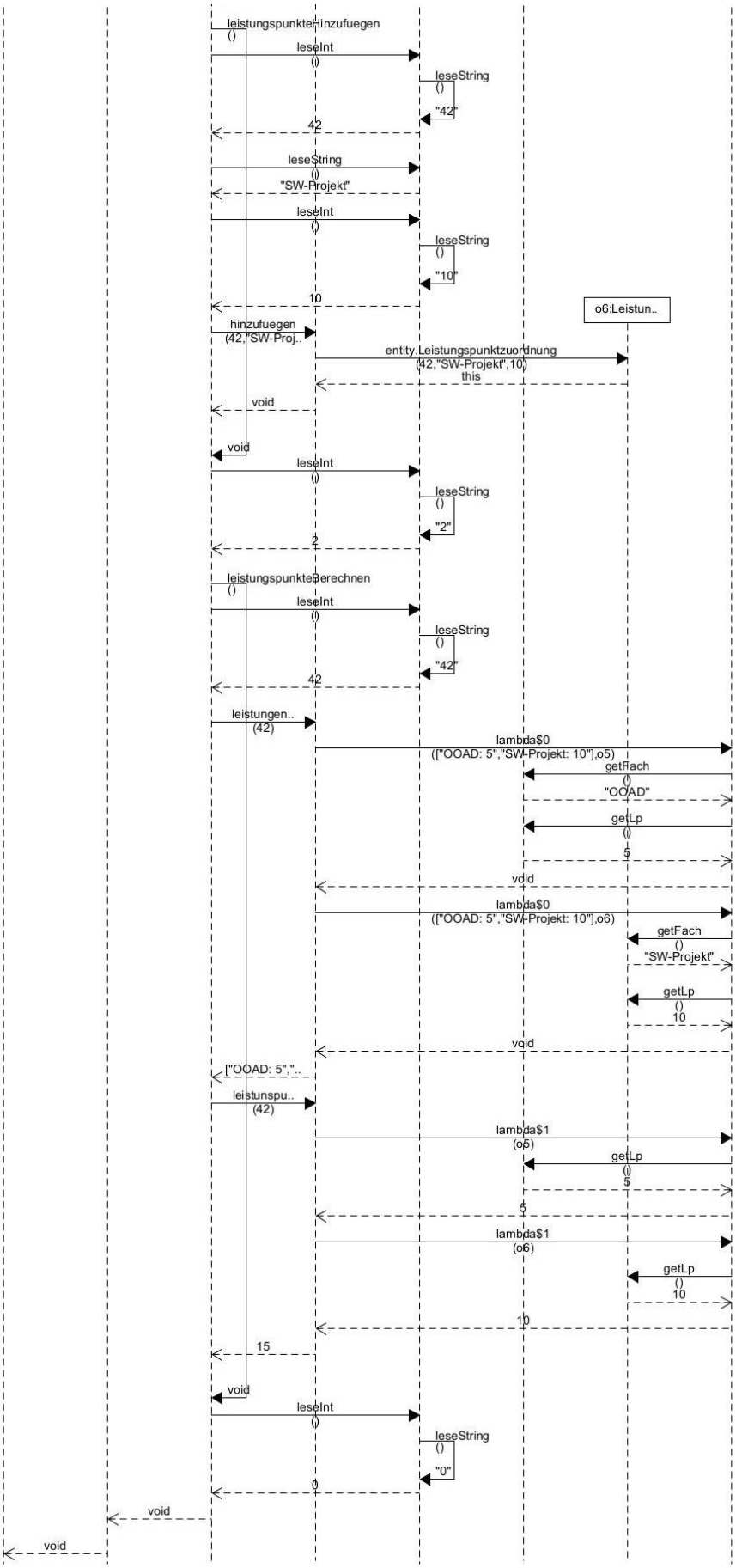
```
        return erg;
    }

    public int leistunspunkteVon(int matnr){
        int erg = 0;
        Set<Leistungspunktzuordnung> tmp = this.leistungen.get(matnr);
        if (tmp == null){
            return 0;
        }
        erg = tmp.stream()
                .map((lp) -> lp.getLp()).reduce(erg, Integer::sum);
        return erg;
    }
}
```

```
package entity;

public class Leistungspunktzuordnung {
    private int matnr;
    private String fach;
    private int lp;

    public Leistungspunktzuordnung(){}

    public Leistungspunktzuordnung(int matnr, String fach, int lp) {
        this.matnr = matnr;
        this.fach = fach;
        this.lp = lp;
    }

    public int getMatnr() {
        return matnr;
    }

    public void setMatnr(int matbr) {
        this.matnr = matbr;
    }

    public String getFach() {
        return fach;
    }

    public void setFach(String fach) {
        this.fach = fach;
    }

    public int getLp() {
        return lp;
    }

    public void setLp(int lp) {
        this.lp = lp;
```

```
    }

    @Override
    public String toString() {
        return "Leistungspunktzuordnung{" + "matbr=" + matnr + ", fach="
                                          + fach + ", lp=" + lp + '}';
    }
}
```

die Lösung zu b) hängt natürlich von a) ab. Die Beobachtungen können sehr unterschiedlich sein und zu Diskussionen über Sequenzdiagramme führen. Es ist zumindest erkennbar, dass die genutzte Collection-Klasse und die Bildschirmausgaben im Diagramm nicht sichtbar sind. Im Beispiel wird eine Lambda-Funktion genutzt, die als Klassenmethode (von Java selbst) umgesetzt wird.




leistungspunkteHinzufuegen
()
leseInt
()
leseString
()
"42"
42
leseString
()
"SW-Projekt"
leseInt
()
leseString
()
"10"
10
o6:Leistun..
hinzufuegen
(42,"SW-Proj..
entity.Leistungspunktzuordnung
(42,"SW-Projekt",10)
this
void
void
leseInt
()
leseString
()
"2"
2
leistungspunkteBerechnen
()
leseInt
()
leseString
()
"42"
42
leistungen..
(42)
lambda$0
(["OOAD: 5","SW-Projekt: 10"],o5)
getFach
()
"OOAD"
getLp
()
5
void
lambda$0
(["OOAD: 5","SW-Projekt: 10"],o6)
getFach
()
"SW-Projekt"
getLp
()
10
void
["OOAD: 5","..
leistunspu..
(42)
lambda$1
(o5)
getLp
()
5
5
lambda$1
(o6)
getLp
()
10
10
15
void
leseInt
()
leseString
()
"0"
0
void
void