

# Hochregallager

Modellierung und Simulation

Berner Fachhochschule - Technik und Informatik

Marc Schärer scham36@bfh.ch  
Arthur van Ommen vano1@bfh.ch  
Fabian Affolter affof1@bfh.ch

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Rahmenbedingungen . . . . .	1
1.2	Abgrenzung . . . . .	1
<b>2</b>	<b>Grundlagen</b>	<b>1</b>
2.1	Allgemeine Grundlagen . . . . .	2
2.1.1	Koordinaten . . . . .	2
2.1.2	Klappung . . . . .	3
2.2	Mathematische Grundlagen . . . . .	4
<b>3</b>	<b>Modellierung</b>	<b>5</b>
3.1	Einheiten . . . . .	5
3.2	Hochregal . . . . .	5
3.3	Regalbediengerät . . . . .	5
3.4	Ereignisse . . . . .	6
3.5	Architektur . . . . .	7
<b>4</b>	<b>Simulation</b>	<b>8</b>
4.1	Allgemein . . . . .	8
4.2	Spezialfälle . . . . .	8
4.3	Berechnung Simulationszeit . . . . .	9
4.4	Fehler durch Rechenzeit und dessen Kompensation . . . . .	9
4.5	Mögliche Strategien . . . . .	9
4.6	Ausführen der Simulation . . . . .	10
4.7	Erzeugung der Eingabedaten . . . . .	10
<b>5</b>	<b>Visualisierung</b>	<b>11</b>
5.1	Trennung . . . . .	11
<b>6</b>	<b>Resultate</b>	<b>11</b>
<b>A</b>	<b>Abbildungsverzeichnis</b>	<b>15</b>
<b>B</b>	<b>UML-Diagramme</b>	<b>15</b>
<b>C</b>	<b>Projekt-Beteiligte</b>	<b>16</b>
<b>D</b>	<b>Sonstiges</b>	<b>17</b>

# 1 Einleitung

Ein Hochregallager (HRL) beschreibt ein Lagersystem mit Plätzen in sogenannten Regalen. Hochregallager gibt es in den unterschiedlichsten Ausprägungen. Die grössten Ausführungen besitzen Höhen bis etwa 50 m und können mehreren hunderttausend Plätze besitzen. Oftmals werden direkt Euro-Paletten als Träger für das Lagergut verwendet, ist das Lagergut zu klein, werden häufig spezielle Kunststoff-Behälter benutzt.

Grobgesagt besteht ein Hochregallager aus einer bestimmten Anzahl von Gassen. Eine Gasse wiederum hat links und rechts Lagerplätze und im Freiraum bewegt sich ein Bediengerät. In einem manuellen Hochregallager ist dieser Raum so gross, dass mit einem Gabelstapler zwischen den Regalwänden manövriert werden kann. Bei automatischen Lagern fährt ein Bediengerät, welches von einem Lagerverwaltungssystem seine Befehle bekommt, ohne manuelle Interventionen in der Gasse und liefert das Lagergut zur Entnahmestelle.

Die Hochregallager haben eine hohe Raumnutzung und bei der Erstellung sind hohe Investitionen nötig, da bei kleineren Ausführungen eine Halle um das Hochregallager gebaut werden muss. Bei grossen Varianten wird das Hochregal als Tragstruktur für das Gebäude mitbenutzt.

## 1.1 Rahmenbedingungen

## 1.2 Abgrenzung

Die Schnittstelle liegt an der Stirnseite des Hochregallagers zur Vorzone. Die gestrichelte Linie in 1 auf Seite 2 stellt diese Grenze dar. Das Hochregal hat keine fest definierte Abmessungen und auch die Lagergüter haben keine definierten Masse oder Gewicht. Es wird davon ausgegangen, dass die Lagergüter auf einem Träger (z. B. Euro-Palette) platziert sind oder sich in einem Behälter befinden und so zum Hochregal kommen. Alle Lagergüter werden mit den gleichen physikalischen Bedingungen befördert.

# 2 Grundlagen

Im Sinne eines Hochregallagers besteht eine Gasse aus einer rechten und einer linken Regalwand während sich in der Mitte der beiden Wände ein Korridor für das Regalbediengerät (RBG) befindet. Die Regalwände sind in Lagerplätze unterteilt, die von Regalbediengerät be- und entladen werden. Hochregallager können aus einer beliebigen Anzahl Gassen bestehen. Im Normalfall befindet sich an einer Stirnseite der Gassen die sogenannte Vorzone, welche die Aufgabe hat, die Lagergüter auf die zugewiesenen Gassen zu verteilen.

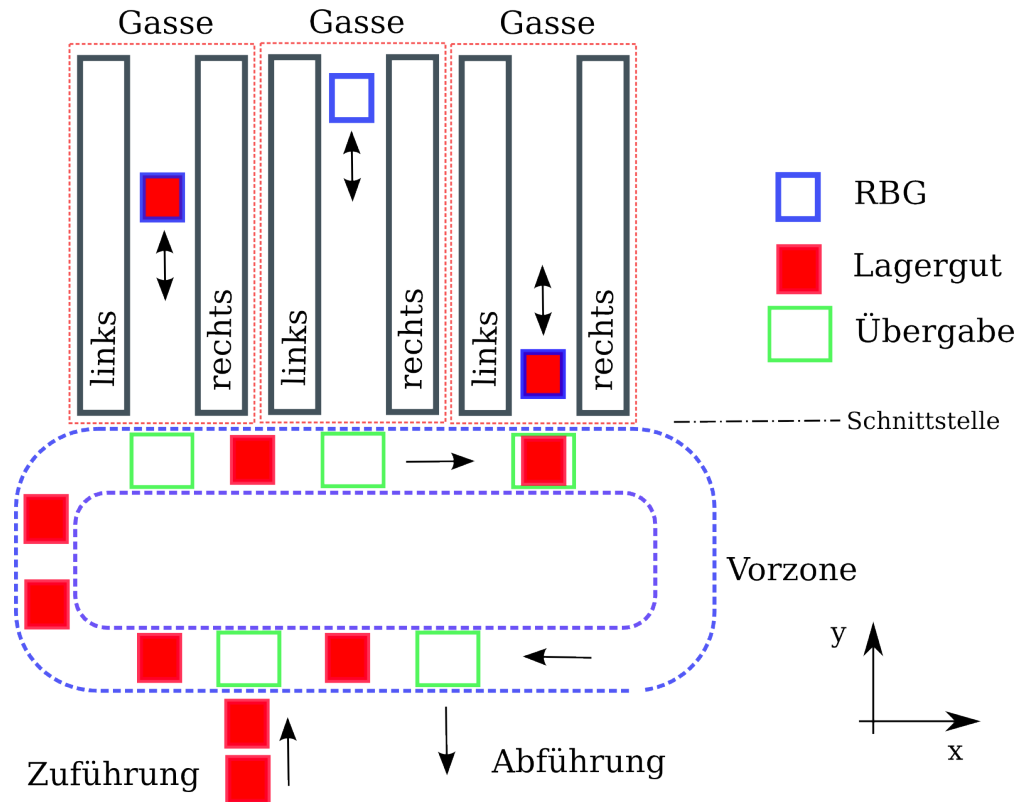


Abbildung 1: Übersicht

## 2.1 Allgemeine Grundlagen

Dieser Abschnitt enthält Details allgemeiner Natur, damit im weiteren Verlauf des Dokuments Unklarheiten vermieden werden können. Die Sprache innerhalb des Source Code ist Englisch, während die Dokumentationssprache Deutsch ist. Die wichtigsten Benennungen sind in Tabelle 1 ersichtlich.

Tabelle 1: Benennungen

Location	Lager
Gap	Gasse
Grid	Regalwand
Column	Spalten im Regal
Row	Zeilen im Regal
Bin	Lagerfach
Rack feeder	Regalbediengerät

### 2.1.1 Koordinaten

Der Koordinatenursprung befindet sich in der linken unteren Ecke der Regalwand (Grid). Es wird yz-Koordinatensystem aufgespannt und die Koordinaten der Lagerplätze (Bins) sind in die linke untere Ecke gesetzt. Das Regalbediengerät kann sich auf der y- und der z-Achse bewegen. Der

Tabelle 2: Farbzuordnung Lagerplätze

grün	leer
gelb	belegt
rot	reserviert/defekt/spezial (wird nicht implementiert)

Übergabebereich befindet sich ausserhalb des Koordinatensystems auf dem negativen Abschnitt der y-Achse. In Abbildung 2 sind ebenfalls die verwendete Farbzuordnung für die Lagerplätze (Bins)

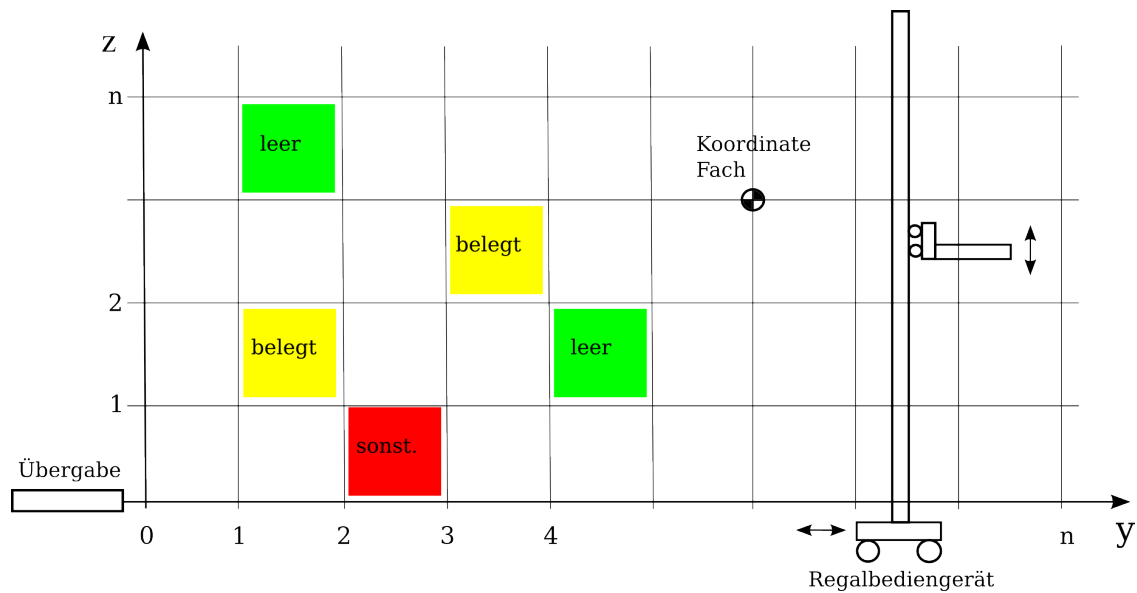


Abbildung 2: Lagerwand

ersichtlich.

### 2.1.2 Klappung

Für die zweidimensionale Darstellung wird eine Lagergasse gemäss Abbildung 3 aufgefaltet, respektive aufgeklappt. Dies erlaubt die komplette Darstellung einer Lagergasse. In der Visualisierung ist die Gasse so dargestellt. Dies ist deshalb wichtig, da sich die Achsen des Koordinatensystems entsprechend der Klappung ändern.

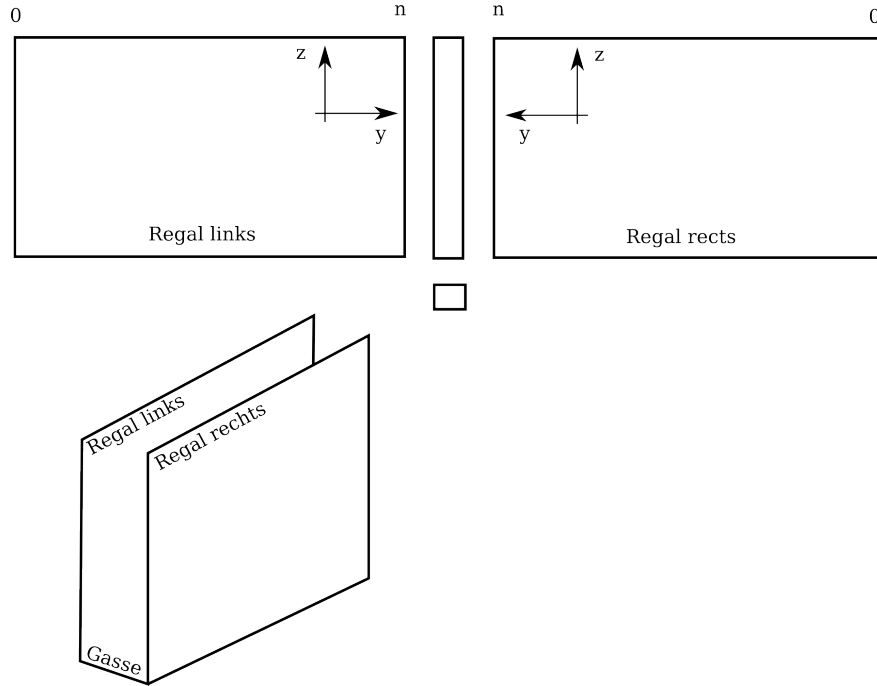


Abbildung 3: Klappung

## 2.2 Mathematische Grundlagen

Das Regalbediengerät bewegt sich in der Gasse zwischen den Lagerwänden auf der y- und der z-Achse. Der Arm des Regalbediengerät verfährt auf der y-Achse und der Ausleger auf der z-Achse. Die Bewegungen auf beiden Achsen lassen somit entsprechend der elektromechanischen Steuerung beliebige Verfahrwege auf der yz-Ebene zu. Die Graduierung der Bewegungsbahn wird durch die Grösse der Inkremente bestimmt. Die Grundgleichung für die Geschwindigkeit in der Ebene lautet:

$$v = \frac{s}{t} \quad (1)$$

Die Fahrt des Regalbediengerät besteht aus einer Phase für die Beschleunigung, einer Phase der gleichförmigen Bewegung mit möglichst maximaler Geschwindigkeit und einem Bremsabschnitt, resp. einer Verzögerungsphase, am Ende. In allen Berechnung wird die Masseträgheit des Regalbediengerätes vernachlässigt. Die Einflüsse durch die Masse der Lagergüter auf die Dynamik des Regalbedientgerätes wird nicht untersucht oder in den Berechnungen berücksichtigt. Die Annahme ist, dass die Lagergüter masslos sind und bei Beschleunigung und Verzögerung weder kippen noch verrutschen können. Weiter führt die Vernachlässigung der Ergbeschleunigung bei Bewegungen auf der z-Achse zu Fehler. Die Erdbeschleunigung ( $g$ ) summiert sich bei einer Bewegung in Richtung der negativen z-Achse zur vorhandenen Beschleunigung hinzu und bei Bewegungen in positiver Richtung (nach oben) müsste sie zusätzlich überwunden werden. Das Regalbediengerät wird als reibungsfrei angenommen. Die kürzeste Fahrzeit ergibt sich auch mehreren Faktoren. Die Synchronisationsgerade ist ein in diesem Zusammenhang oft verwendeter Begriff, welcher die optimale Fahrbahn des Regalbediengerät beschreibt. Dieser Fahrbahn führt jedoch nicht zur optimalen Fahrzeit, da unter Umständen auf beide Achsen gebremst werden müssten. Wir haben den Ansatz gewählt, dass, wenn möglich, nur auf der schnelleren Achse die Geschwindigkeit reduziert wird, was dazu führt, dass sich die zweite Achse entsprechend ihrem Maximum fortbewegen kann. Ein Grenzfall tritt auf, wenn die zu bewältigende Strecke kleiner wird als die Summe der Wege von der

Beschleunigung und Verzögerung bei konstanten Beschleunigungen/Verzögerungen und Geschwindigkeiten.

$$v = \sqrt{\frac{2 \cdot s \cdot a \cdot d}{(a + d)}} \quad (2)$$

Anhand dieser Formel lässt sich der optimale Verlauf der Bewegung innerhalb der Regalwand bestimmen.

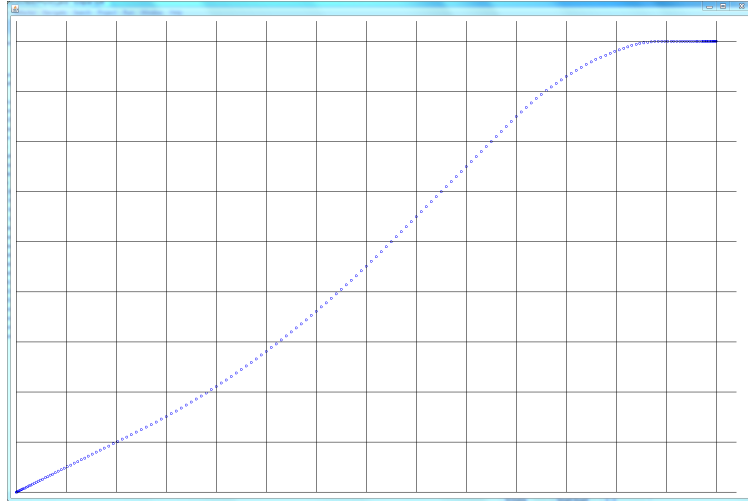


Abbildung 4: Bewegung

### 3 Modellierung

Der Aufbau des Hochregallagers wurde in diverse Bereiche unterteilt. Die wichtigsten sind das Hochregal selber, auch das Regalbediengerät. Weitere sind die entsprechenden Zustände/Ereignisse, die innerhalb des Hochregallagers auftreten können.

#### 3.1 Einheiten

Die Basiseinheit ist Milimeter. Es ist jedoch möglich mit anderen SI-Einheiten (m, dm oder cm) zu arbeiten.

#### 3.2 Hochregal

Das gesamte Hochregallager, oder auch der Lagerort, wird durch ein einzelnes Objekt dargestellt. Dieses Objekt enthält alle benötigten Informationen und ist als Singleton implementiert. Das heisst, dass auf die Gassen, das Regal und die Lagerfächer direkt zugegriffen werden kann. Dies bietet eine komfortable Arbeit bei der Simulation und soll auch eine einfache Implementierung der Visualisierung. Weiter bietet es den Vorteil, dass alle Informationen über die Elemente des Lagerortes immer verfügbar sind und abgefragt werden können. `GetInstance()` und den entsprechenden Get-Methoden des Objekts auf jedes einzelne Element des Lagerorts zugegriffen werden.

#### 3.3 Regalbediengerät

Innerhalb der Gasse bewegt sich nur das Regalbediengerät. Die Klasse `RackFeeder` bildet dies ab. In vorherigen Abschnitt wurde auf die zulässigen Bewegungen eingegangen. Die maximalen Be-

schleunigungen, Verzögerungen und die Geschwindigkeit sind innerhalb dieser Klasse als Standardwerte definiert. Bei Bedarf lassen sie sich überschreiben.

Das Regalbediengerät (Rackfeeder) kann nicht beliebige Bewegungen ausführen. Die einzig gültigen Bewegungen sind durch ein vollständiges Objekt definiert, es sind nur drei Bewegungen zulässig:

- Bewegung in yz-Achse
- Bewegung in x-Achse (dies ist vorhanden, jedoch wird es nicht verwendet, da in der Realität die Operationen in der Vorzone von einem zusätzlichen System wahrgenommen werden)
- Bewegung in u-Achse (in das Regalfach hinein zum Be- und Entladen des Lagergütes)

Dies verhindert, dass der RackFeeder physikalisch unmögliche Bewegungen ausführen kann und stellt sicher, dass in jedem Zustand nur die darin gültigen Bewegungen ausgeführt werden können.

### 3.4 Ereignisse

Innerhalb des Systems wurden die Ereignisse (Events) in die Operationen Einlagern und Auslagern zerlegt. Die Verzweigungen sind in den nachfolgenden Abbildungen in gelb dargestellt. Der Startpunkt ist blau.

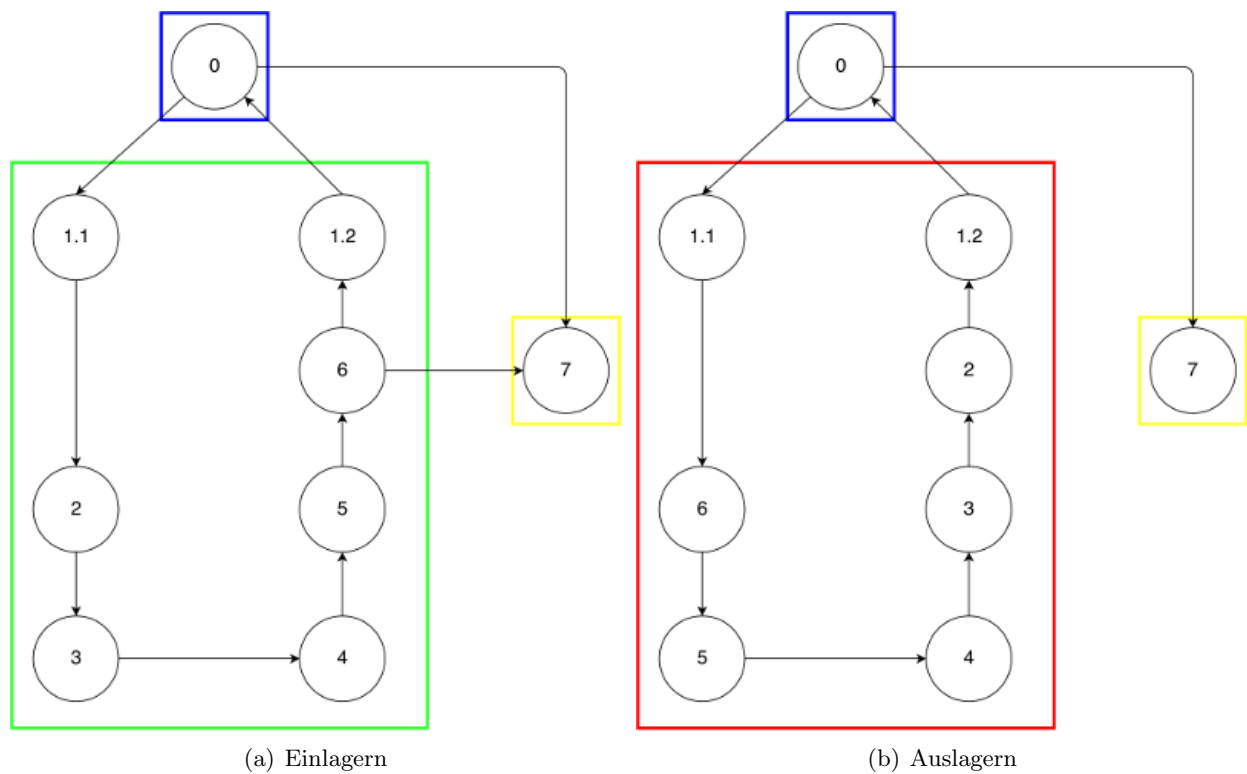


Abbildung 5: Verhalten

Die nachfolgende Tabelle zeigt auf der linken Seite die Einlagerung und auf der rechten Seite die Auslagerung.



Eventnr.	Koord.	Zustand	Eventnr.	Koord.	Zustand
1	0/0	empty	1	0/0	empty
2	0/0	loaded	6	y/z	empty
3	y/z	loaded	5	x	empty
4	x	loaded	4	x	loaded
5	x	empty	3	y/z	loaded
6	y/z	empty	2	0/0	loaded
1	0/0	empty	1	0/0	empty
7		sleep	7		sleep

### 3.5 Architektur

Die Unterteilung der Anwendung erfolgte in einen Teil für die Modellierung, Simulation und die Visualisierung. Diese Trennung macht die Anwendung in Bezug auf geänderte Lagerorte und Simulationensszenarien, oder auch für Optimierungen flexibel.

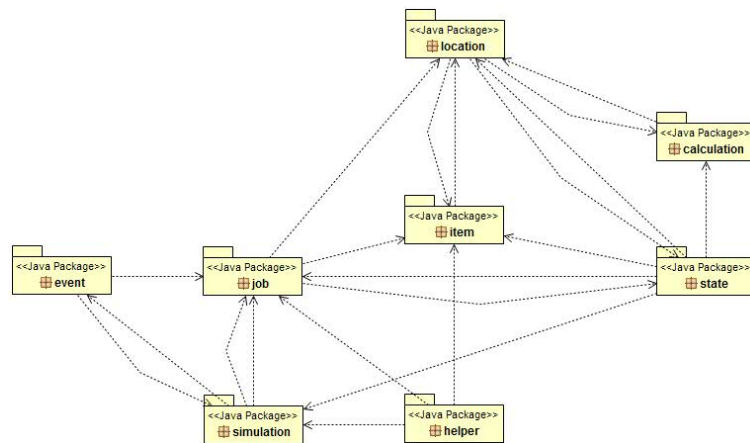


Abbildung 6: Package-Diagramm

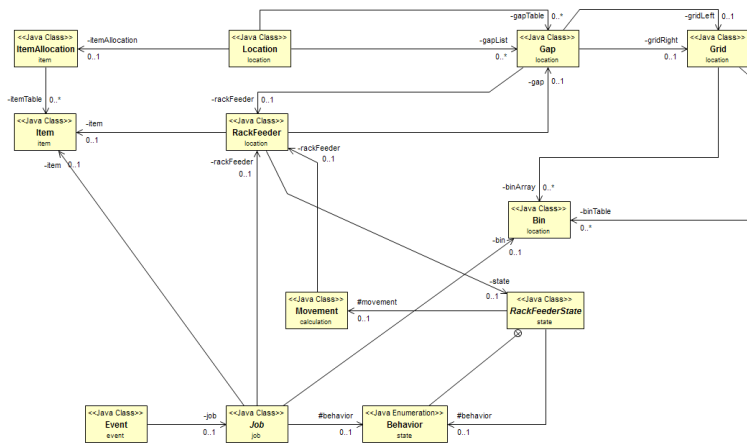


Abbildung 7: Class-Diagramm

Die weiteren Diagramme befinden sich im Anhang.

## 4 Simulation

## 4.1 Allgemein

Die Simulation wurde als ereignisgesteuerte Echtzeitsimulation implementiert. Die Basis sind Lageraufträge wie Einlagerungs-, Auslagerungs- und Umlagerungsaufträge. Zu Beginn der Simulation muss (momentan) mindestens 1 Auftrag vorhanden sein, da die leere Auftragsliste, bzw. die daraus resultierende leere Ereignisliste, das Abbruchkriterium ist, um die Simulation zu beenden. Dies könnte noch verbessert werden, indem die gewünschte Simulationsendzeit als Erinnerungsereignis in die Ereignisliste eingetragen würde. Die Simulation würde dann nach Abarbeitung dieses letzten Erinnerungsereignisses gestoppt, sofern mittlerweile keine noch späteren Ereignisse eingetragen wurden.

Es könnten während der laufenden Simulation weitere Aufträge hinzugefügt werden. Die Startzeit des Auftrags muss allerdings die aktuelle Simulationszeit übersteigen. Bei der Eingabe von zusätzlichen Aufträgen ist das Zeitformat zu berücksichtigen. Diese Funktion ist momentan nicht implementiert.

Für jeden zukünftigen Auftrag wird ein Erinnerungsereignis in die Ereignisliste eingetragen mit Ereigniszeit = Startzeit des Auftrags. Dies dient dazu, dass die Simulation selbst nur die Ereignisliste abarbeiten kann, indem die Simulation jeweils das früheste Ereignis aus der Liste entfernt, anhand der Simulationszeit die benötigte Wartezeit berechnet und dann exakt zur richtigen Simulationszeit das Ereignis ausführt. In der Regel führt jedes Ereignis zu einem Nachfolgeereignis, welches wiederum in die Ereignisliste eingetragen wird. Dieser Vorgang wird solange wiederholt, bis die Ereignisliste leer ist, wodurch die Simulation dadurch beendet wird.

## 4.2 Spezialfälle

Einzig das letzte Ereignis eines Auftrags sowie unter Umständen ein Erinnerungsereignis können dazuführen, dass kein Nachfolgeereignis mehr erstellt wird. Dies ist der Fall, wenn für die aktuelle Gasse keine weiteren Aufträge mehr vorhanden sind oder überhaupt keine Aufträge mehr zur Abarbeitung bereitstehen.

Würden keine Erinnerungsereignisse generiert, sondern direkt das erste Ereignis des entsprechenden Auftrags, wäre es schwierig zusätzliche Aufträge, die vor dem anderen Ereignis startet, in die

Ereignisliste einzufügen und diese Liste entsprechend zu bereinigen.

Dies wäre aber notwendig, da ansonsten zwei unterschiedliche Aufträge in derselben Gasse parallel ausgeführt würden. Dies ist aber physikalisch unmöglich und darf in der Simulation nicht geschehen! Bei jedem Erinnerungsereignis wird deshalb nur die Auftragsliste auf fällige Aufträge geprüft und bei Bedarf das erste Ereignis für diese Aufträge generiert. Es werden aber nur Ereignisse generiert für Aufträge von Gassen, welche nicht bereits durch andere, momentan ausgeführte Aufträge blockiert sind. Für diese Prüfung wird wiederum die Ereignisliste herangezogen.

### 4.3 Berechnung Simulationszeit

Die aktuelle Simulationszeit wird anhand folgender Formel berechnet:

$$\begin{aligned} \text{AktuelleSimulationszeit}[ms] &= \text{Simulationsstartzeit}[ms] \\ &+ \text{Faktor} * (\text{AktuelleSystemzeit}[ms] - \text{Startsystemzeit}[ms]) \end{aligned} \quad (3)$$

Da im Modus *As Fast As Possible* nicht auf die exakte Ausführungszeit gewartet werden soll, sondern die Ereignisse sofort ausgeführt werden sollen, wird die Formel mit einem Korrekturwert ergänzt, welcher der Summe der Wartezeiten entspricht:

$$\begin{aligned} \text{AktuelleSimulationszeit}[ms] &= \text{Simulationsstartzeit}[ms] \\ &+ \text{Faktor} * (\text{AktuelleSystemzeit}[ms] - \text{Startsystemzeit}[ms]) + \text{Korrekturwert}[ms] \end{aligned} \quad (4)$$

Der Korrekturwert bleibt bei der Echtzeitsimulation immer 0, weshalb für beide Simulationsmodi dieselbe Formel verwendet werden kann.

### 4.4 Fehler durch Rechenzeit und dessen Kompensation

Es gibt nun aber Fälle, in denen die Ereignisse anhand der aktuellen Simulationszeit zu spät ausgeführt werden. Dies dadurch, dass mehrere Ereignisse die gleiche oder zumindest eine kurz aufeinanderfolgende Startzeit haben. Der Grund für die Verzögerung ist, dass ja das *ausführen* der Ereignisse Rechenzeit benötigt und die Berechnung der aktuellen Simulationszeit auf der Systemzeit beruht. Diese Verzögerungen würden sich summieren und müssen unbedingt kompensiert werden. Also wird für die Nachfolgeereignisse nicht die aktuelle Simulationszeit als Basis zur Berechnung der Startzeit des Nachfolgeereignisses verwendet, sondern die Startzeit seines Vorgängerereignisses. Die aktuelle Simulationszeit wird also nur verwendet, um die Wartezeit bis zur Ausführung des nächsten Ereignisses zu berechnen und um zu jedem Zeitpunkt der Simulation den entsprechenden Zustand bzw. die aktuellen Koordinaten der Regalbediengeräte zu berechnen

### 4.5 Mögliche Strategien

Für die Simulation gibt es unterschiedliche Strategien, welche das Verhalten des Lagers beeinflussen. Neben den Teilen für das Hochregallager selber, gäbe es auch Punkte, die nur das Regalbediengerät berücksichtigen. Im momentan Stand wird das Regalbediengerät, als Ruhepositionsstrategien, immer wieder auf die Warteposition bei der Vorzone gesetzt. Ohne statistische Daten einer Lagerort-Konfiguration kann die bestmögliche Strategie für die Ruheposition nicht bestimmt werden.

Das Hochregallager wird mit den Artikel aus der Artikelliste (siehe 4.7 auf Seite 11) per Zufall gefüllt. Eine Überprüfung stellt sicher, dass die ersten Aufträge erledigt werden können. Das heisst, dass für die Auslagerung die Fächer belegt sind und bei der Einlagerung die Fächer frei.

## 4.6 Ausführen der Simulation

Die Simulation lässt sich durch Kommandozeilen-Argumente steuern.

Arguments definition:

```
-m (afap | f) // as fast as possible, factor
-f (double)   // factor
-l (int)      // location number, joblist number, itemlist number
-p (int)      // print type, 0 for file, any other for console
```

As fast as possible, Faktor 1 (wird überschrieben), Lagerort 2 laden, Ausgabe in Datei

```
-m afap -f 2.5 -l 2 -p 0
```

As fast as possible, Faktor 1, Lagerort 3 laden, Ausgabe in Konsole

```
-m afap -f 1 -l 3 -p 15
```

Mit Faktor, Faktor 2.5, Lagerort 2 laden, Ausgabe in Konsole

```
-m f -f 2.5 -l 2 -p 14
```

Mit Faktor, Faktor 1, Lagerort 3 laden, Ausgabe in Datei

```
-m f -f 1 -l 3 -p 0
```

## 4.7 Erzeugung der Eingabedaten

Für die Simulation werden flache Text-Dateien verwendet, welche die relevanten Daten in einer Strichpunkt-getrennten Schreibweise enthalten. Es werden eine Definition des Lagerorts, einer Liste mit Artikeln und einer Auftragsliste.

### Lagerort-Liste (locationX.txt)

```
Location;my location;3;MM
Gap;Gasse1;1000;Grid1;Grid2;7;4
Grid;Grid1;1000
Grid;Grid2;1000
Column;A;1200
...
Row;1;600
...
Gap;Gasse2;1000;Grid3;Grid4;5;3
Grid;Grid3;1000
...
Column;A;1000
...
Row;1;500
...
Gap;Gasse3;600;Grid5;Grid6;4;4
Grid;Grid5;600
Grid;Grid6;600
Column;A;1000
...
Row;1;500
...
...
```

### Artikel-Liste (item\_listX.txt)

```
00001;Article 1
00002;Article 2
00003;Article 3
00004;Article 4
00005;Article 5
00006;Article 6
...
```

### Auftragsliste (job\_listX.txt)

```
2000.01.01 00:00:00.300;0;Gasse1-1-Grid2-E-3
2000.01.01 00:00:00.200.123;I;Gasse3-1-Grid6-C-2;00123
2000.01.01 00:00:00.100;I;Gasse3-1-Grid6-C-4;00001
2000.01.01 00:00:00.400;0;Gasse2-0-Grid3-E-2
...
```

## 5 Visualisierung

Auf Grund der hohen Dichte von möglichen Visualisierungselementen wird das Hochregallager aufgeklappt und vereinfacht dargestellt.

### 5.1 Trennung

Der Lagerwand und den involvierten Akteure wurde je eine Ebene (Layer) zugewiesen. Dies stellt sicher, dass bei der Darstellung nur die Elemente, welche sich seit dem letzten Schritt verändert haben, neu geladen werden müssten.

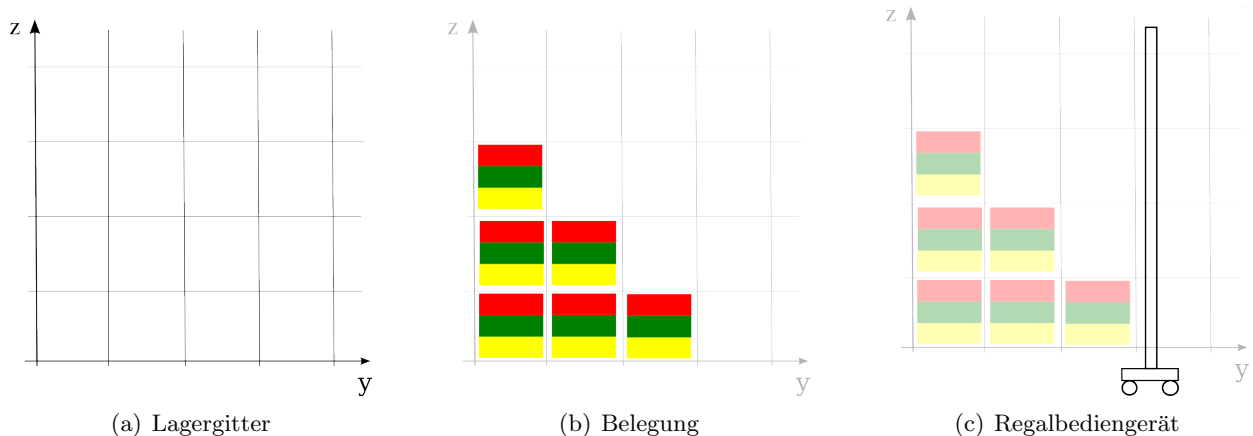
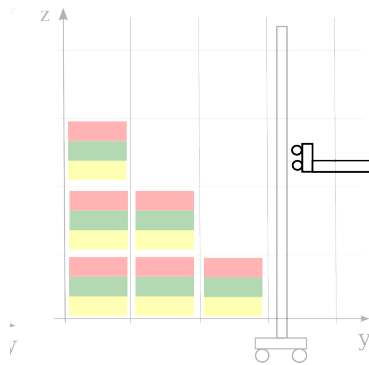


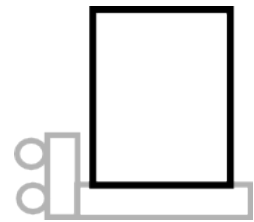
Abbildung 8: Trennungen

## 6 Resultate

Die Ausgabe des Durchlaufs wird in der Konsole ausgegeben. Es sind die folgenden Informationen vorhanden:



(a) Regalbediengerät-Ausleger



(b) Lagergut

Abbildung 9: Weitere Trennungen

- Kompletter Lagerort
- Lagerplätze und deren Belegung
- Bekannte Job beim Simulationsstart
- Gewählte Parameter
- Neue Zustand des gesamten Lagerorts

Lagerplatzzuweisung vor Start der Simulation:

Lagerort:

Name = my location 2

Anzahl Gassen: 5

Name: Gasse1, X-Koordinate: 1000, Breite: 1000

Grid L: Name: Grid1, Tiefe: 1000, Laenge: 20400, Hoehe: 11250

Grid R: Name: Grid2, Tiefe: 1000, Laenge: 20400, Hoehe: 11250

Name: Gasse2, X-Koordinate: 4000, Breite: 1000

Grid L: Name: Grid3, Tiefe: 1000, Laenge: 4800, Hoehe: 1600

Grid R: Name: Grid4, Tiefe: 1000, Laenge: 4800, Hoehe: 1600

[snip]

Anzahl Bin: 710

- 1 - ID: Gasse1-0-Grid1-A-1, Artikel:  
    , Koordinaten: (X/Y/Z/U) : 1000/0/0/-1000
- 2 - ID: Gasse1-0-Grid1-A-2, Artikel:  
    , Koordinaten: (X/Y/Z/U) : 1000/0/600/-1000
- 3 - ID: Gasse1-0-Grid1-A-3, Artikel:  
    , Koordinaten: (X/Y/Z/U) : 1000/0/1400/-1000
- 4 - ID: Gasse1-0-Grid1-A-4, Artikel:  
    , Koordinaten: (X/Y/Z/U) : 1000/0/1900/-1000
- 5 - ID: Gasse1-0-Grid1-A-5, Artikel:  
    , Koordinaten: (X/Y/Z/U) : 1000/0/2500/-1000

```

6 - ID: Gasse1-0-Grid1-A-6, Artikel: Article 828
    , Koordinaten: (X/Y/Z/U) : 1000/0/3250/-1000
7 - ID: Gasse1-0-Grid1-A-7, Artikel:
    , Koordinaten: (X/Y/Z/U) : 1000/0/4050/-1000
8 - ID: Gasse1-0-Grid1-A-8, Artikel: Article 447
    , Koordinaten: (X/Y/Z/U) : 1000/0/4550/-1000
[snip]
709 - ID: Gasse5-1-Grid10-D-3, Artikel:
    , Koordinaten: (X/Y/Z/U) : 14000/3200/1000/1000
710 - ID: Gasse5-1-Grid10-D-4, Artikel: Article 1382
    , Koordinaten: (X/Y/Z/U) : 14000/3200/1600/1000

Bereits bekannte Jobs beim Start der Simulation:
-----
Job: ID = ' 4', RackFeeder = 'Gasse3', Startzeit = 2000.01.01 00:00:00.049
Job: ID = ' 2', RackFeeder = 'Gasse2', Startzeit = 2000.01.01 00:00:00.050
Job: ID = ' 3', RackFeeder = 'Gasse3', Startzeit = 2000.01.01 00:00:02.321
[snip]

Erinnerungsevent hinzugefuegt; Eventzeit: 2000.01.01 00:00:00.049
Erinnerungsevent hinzugefuegt; Eventzeit: 2000.01.01 00:00:00.050
[snip]

Simulation wird nun gestartet:
Aktuelle Systemzeit: 2014.01.17 12:48:50.657
Aktueller Faktor: 1.0
Aktueller Modus: AS_FAST_AS_POSSIBLE

-----
1 -----
Simulationszeit: 2000:01:01 00:00:00.003
Erinnerungsevent gefunden, Startzeit: 2000.01.01 00:00:00.049

Wartezeit in ms (simuliert): 45

Simulationszeit (echt) nach Wartezeit: 2000:01:01 00:00:00.050
Simulationszeit (soll) nach Wartezeit: 2000.01.01 00:00:00.049

Kein Nachfolgeevent. Eventuell Erinnerungsevents oder Startevents anlegen?
Event hinzugefuegt fuer Job '4', RackFeeder 'Gasse3';
    Eventzeit: 2000.01.01 00:00:00.049
-----
2 -----
Simulationszeit: 2000:01:01 00:00:00.079
[snip]
Wartezeit in ms (simuliert): 0

Simulationszeit (echt) nach Wartezeit: 2000:01:01 00:10:07.744
Simulationszeit (soll) nach Wartezeit: 2000.01.01 00:10:07.740

```

Kein Nachfolgeevent. Eventuell Erinnerungsevents oder Startevents anlegen?

-----

Simulation wird nun beendet, Start-Systemzeit: 2014.01.17 12:48:50.657

aktuelle Systemzeit: 2014.01.17 12:48:51.140

Vergangene Systemzeit in Millis: 483

Die komplette Ausgabe kann auch in eine Datei geschrieben werden, was eine später Vergleich der Durchgänge ermöglicht.

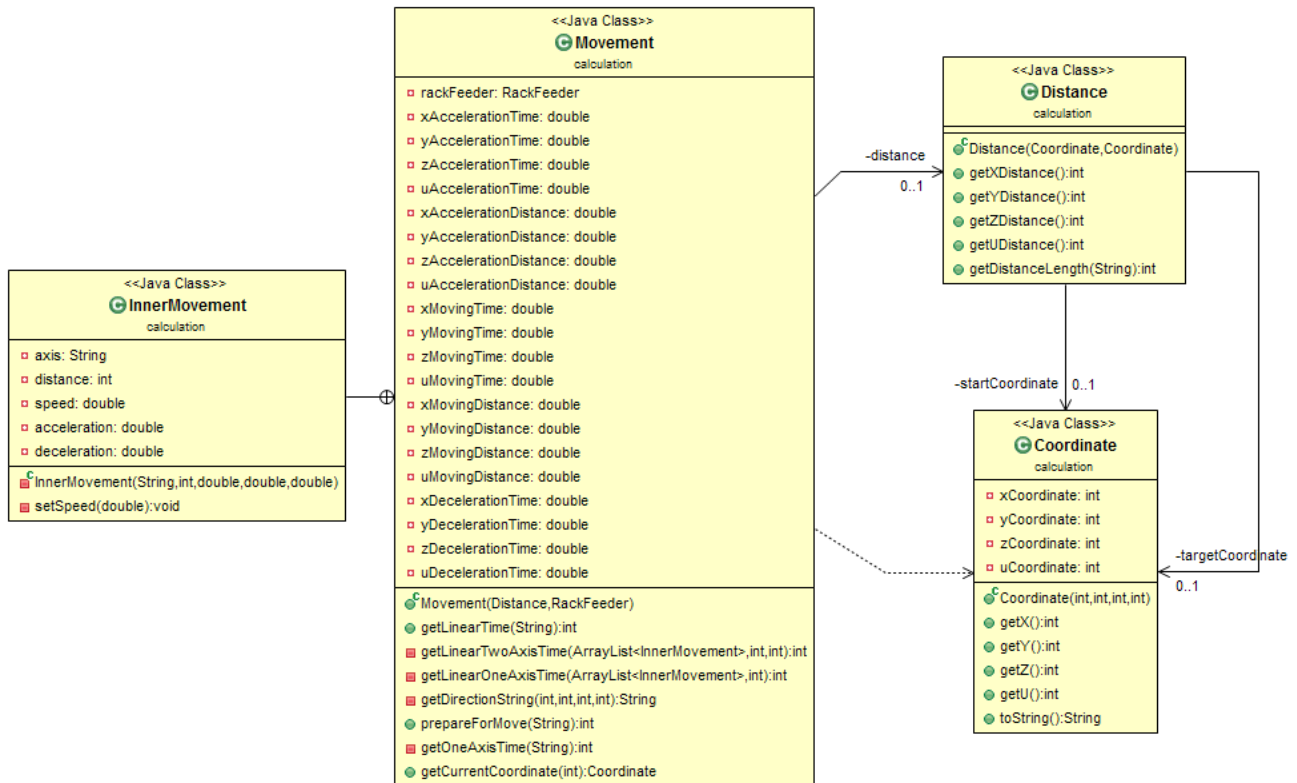


## A Abbildungsverzeichnis

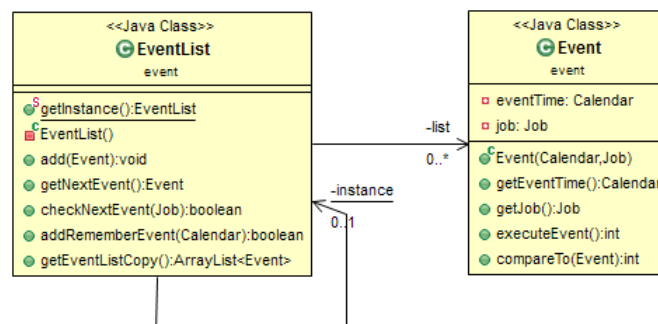
1	Übersicht . . . . .	2
2	Lagerwand . . . . .	3
3	Klappung . . . . .	4
4	Bewegung . . . . .	5
5	Verhalten . . . . .	6
6	Package-Diagramm . . . . .	7
7	Class-Diagramm . . . . .	8
8	Trennungen . . . . .	11
9	Weitere Trennungen . . . . .	12

## B UML-Diagramme

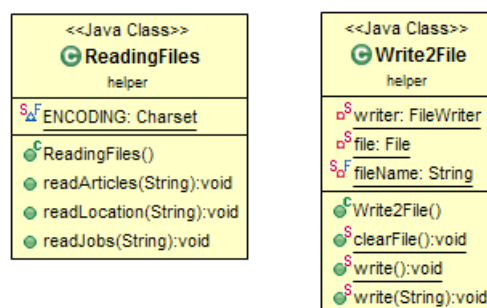
## Calculation



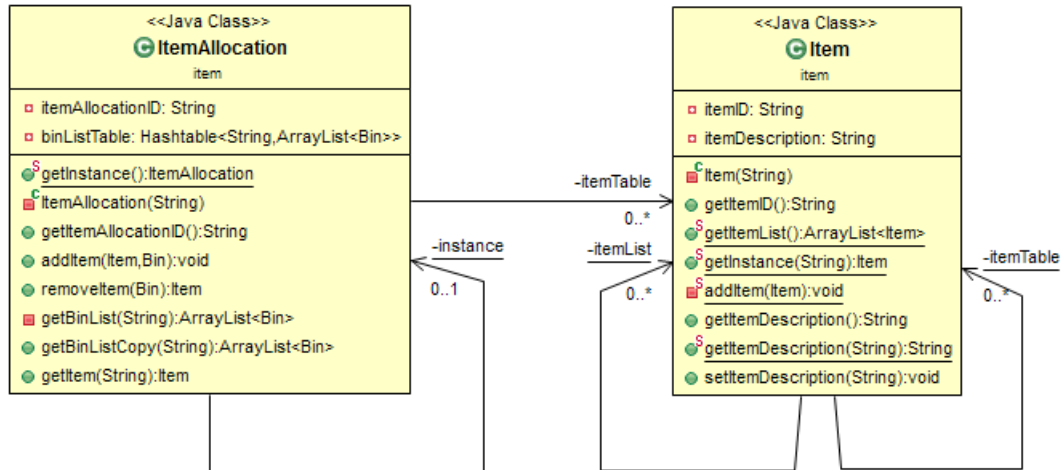
## Event



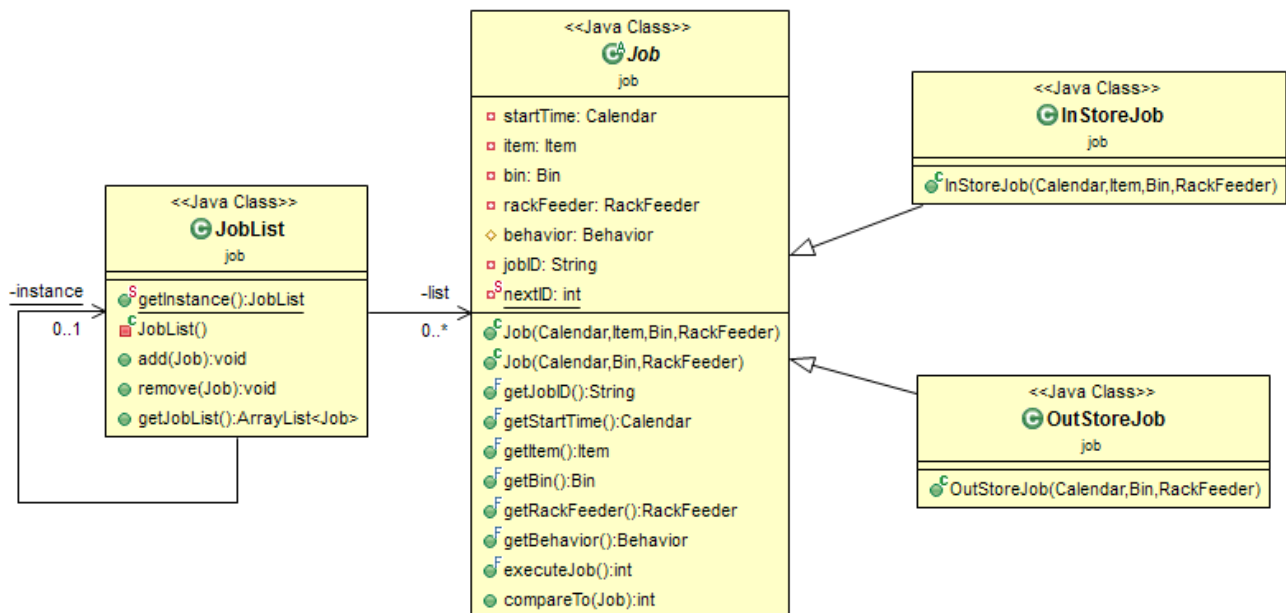
## Helper



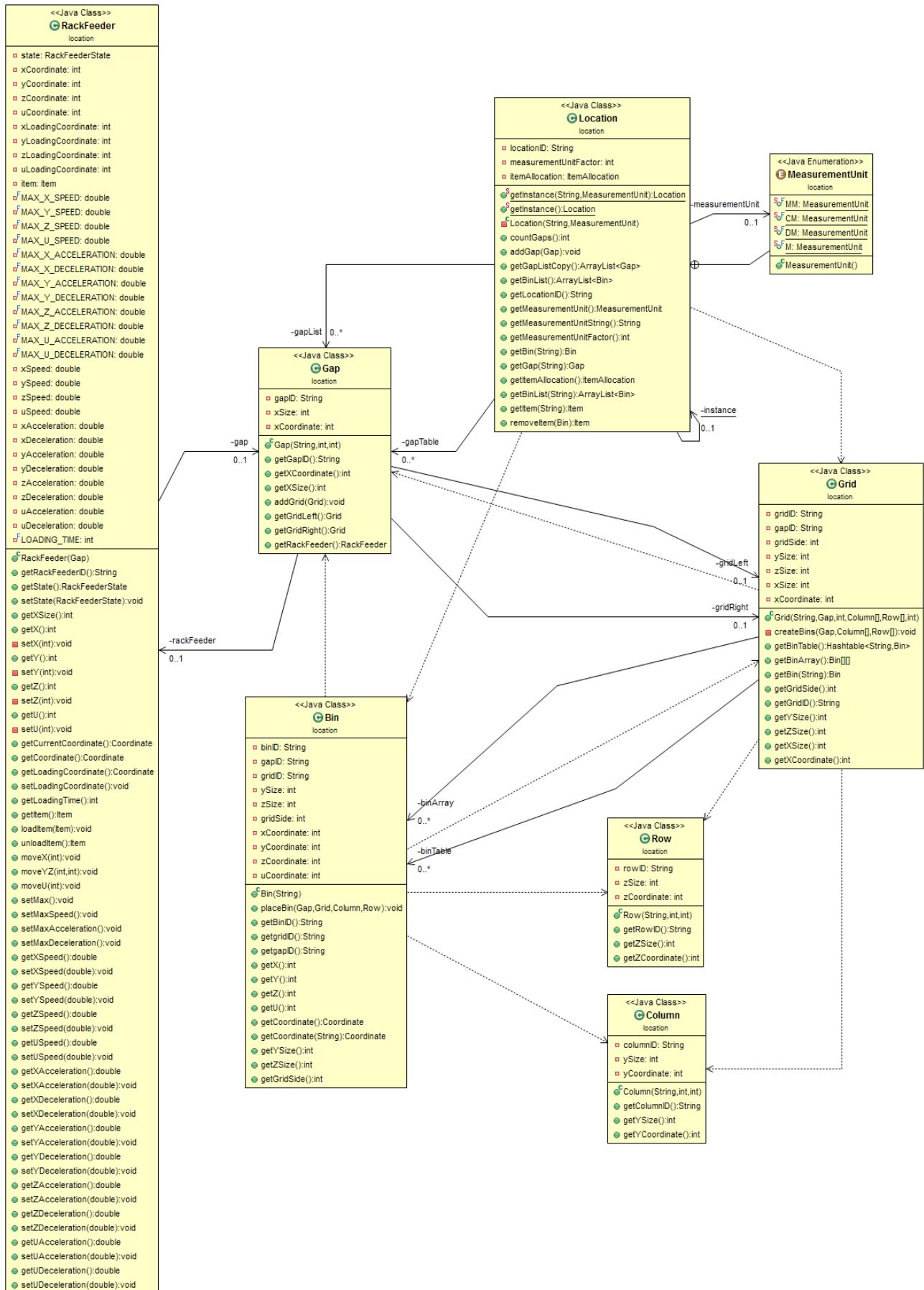
## Item



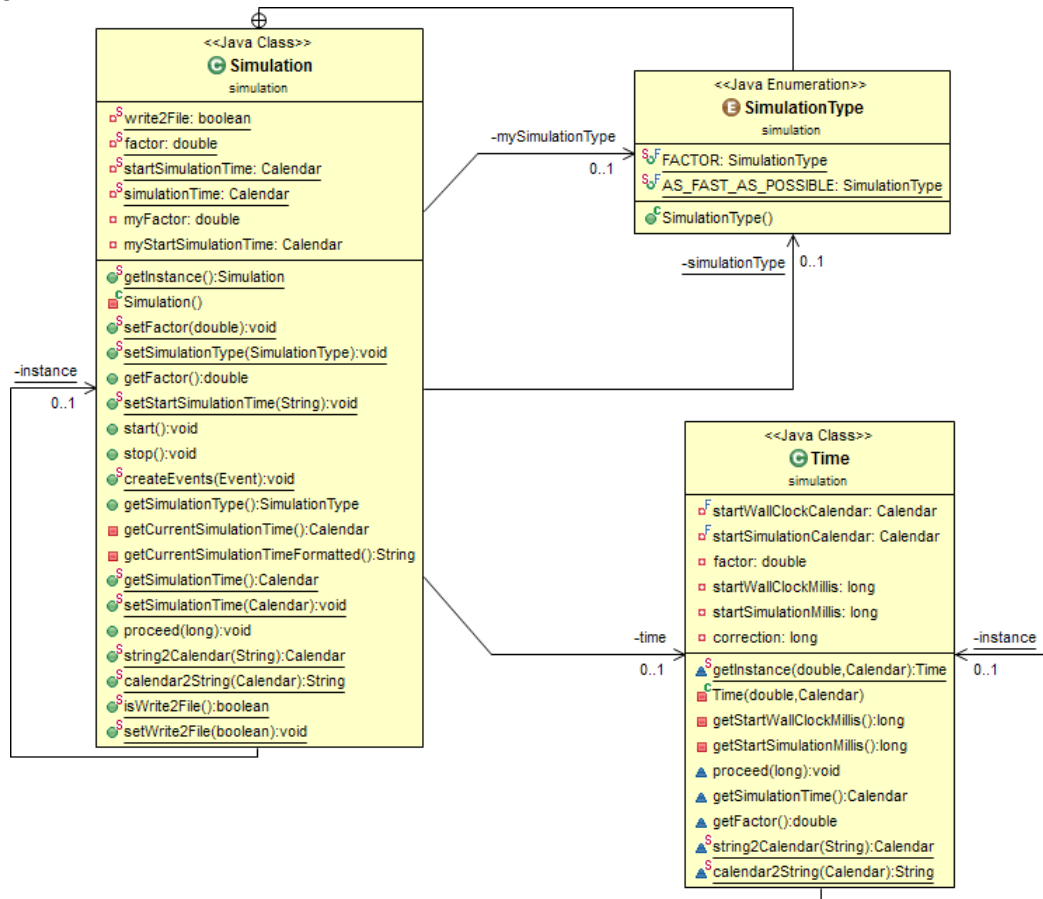
## Job



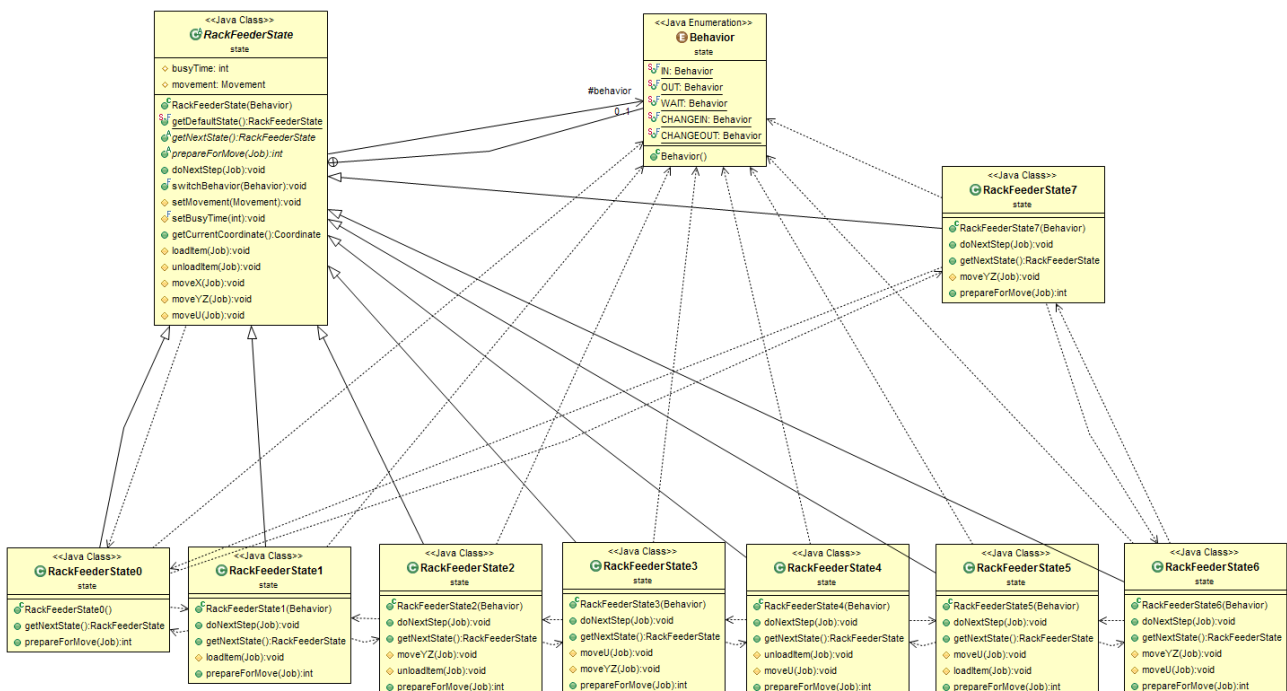
# Location



## Simulation



## State



## C Projekt-Beteiligte

### Verfasser

Marc Schärer `scha36@bfh.ch`

Arthur van Ommen `vanoa1@bfh.ch`

Fabian Affolter `affof1@bfh.ch`

### Betreuer

Berner Fachhochschule  
Technik und Informatik  
Wankdorffeldstrasse 102  
3014 Bern

Jürgen Eckerle

`erj1@bfh.ch`

## **D   Sonstiges**

### **Differenzierung zwischen Mann und Frau**

Für eine bessere Lesbarkeit bei allgemeinen Aussagen wird nur die männliche Form des Substantivs verwendet. Die Leserinnen bitten die Autoren um Verständnis für diese Vereinfachung.

### **Markennamen und Warenzeichen**

Alle Markennamen, Warenzeichen und eingetragenen Warenzeichen, die in diesem Dokument verwendet werden, sind Eigentum ihrer rechtmäßigen Eigentümer. Sie dienen hier nur der Beschreibung beziehungsweise der Identifikation der jeweiligen Firmen, Produkte und Dienstleistungen.