

Duale Hochschule Baden-Württemberg
Mannheim

Seminararbeit

Star Greg
Das Unternehmensplanspiel

Studiengang Wirtschaftsinformatik

Vertiefungsrichtung Softwaremethodik

Verfasser: Britta Jochum
Julia Lakatos
Philipp Mail
Jan Schlenker
Marcel Steinleitner
Fredrik Teschke

Kurs: WWI 10 SWM A
Studiengangsleiter: Prof. Dr.-Ing. Jörg Baumgart

Modul: Umsetzung von Methoden der Wirtschaftsinformatik

Lehrveranstaltung: Fallstudie Systemanalyse

Dozent: Gregor Tielsch

Kurzfassung

Verfasser: Britta Jochum

Julia Lakatos

Philipp Mail

Jan Schlenker

Marcel Steinleitner

Fredrik Teschke

Kurs: WWI 10 SWM A

Firma:

Thema: Star Greg - Das Unternehmensplanspiel

Vorwort

Diese L^AT_EX-Vorlage soll es ermöglichen, ohne tiefergehende L^AT_EX-Kenntnisse eine Seminararbeit erstellen zu können.

Das vorliegende Dokument besitzt die Version x, da die zugehörigen sty-Dateien nach und nach ergänzt werden, um für Seminar-, Projekt- und Bachelorarbeiten neue Kommandos für eine effiziente Dokumenterstellung anbieten zu können. Neue Versionen werden abwärtskompatibel sein.

Inhaltsverzeichnis

Verzeichnisse	vi
Abkürzungsverzeichnis	vi
Symbolverzeichnis	vi
Abbildungsverzeichnis	vi
Tabellenverzeichnis	vi
Listingverzeichnis	vi
1 Einleitung	1
2 Spielwelt	2
2.1 Abschnitt	2
2.2 Abschnitt	2
2.3 Abschnitt	2
2.4 Abschnitt	2
2.5 Spiellogik	2
2.5.1 Einleitung	2
2.5.2 Bauteilpreise	3
2.5.3 Absatzmengen	7
2.5.4 Fehlerkosten	7
2.5.5 Bewertung	9
3 Fachkonzept	10
3.1 Abschnitt	10
3.2 Entwurfsentscheidungen	10
3.2.1 Einleitung	10
3.2.2 Die Typen	11
3.2.3 Die Märkte	11
3.2.4 Das Unternehmen	12
3.2.5 Spielrunde und Transaktionen	12
3.3 Implementierung	12
4 UI Mockups	13
5 junit Tests	14

6	Fazit und Ausblick	15
A	Einige wichtige \LaTeX-Kommandos	17
A.1	Kommandos für die Erzeugung von Literaturverweisen	17
A.2	Kommandos für die Erzeugung von Abkürzungen, Symbolen und Glos- sareinträgen	18
A.3	Abbildungen, Tabellen und Programmlistings	19
A.4	Die Definition und Anwendung von zwei neuen Listenumgebungen . .	20
A.4.1	Das Layout der Standardlistenumgebung von \LaTeX	20
A.4.2	Die neue Listenumgebung <code>seList</code> für Stichpunktlisten	21
A.4.3	Die neue Listenumgebung <code>seToplist</code> für Listen mit einem Label und Aufzählungslisten	23
B	Hinweise zur Installation und Übersetzung	25
B.1	Verwendung von TeXShop (Apple-Welt)	25
B.2	Verwendung von MiKTeX (Windows-Welt)	25
	Glossar	27
	Literaturverzeichnis	28

Verzeichnisse

Abkürzungsverzeichnis

USB Universal Serial Bus

Symbolverzeichnis

π Die Kreiszahl
ND Nutzungsdauer einer Maschine

Abbildungsverzeichnis

1:	Die Beziehung zwischen Angebot und Nachfrage	4
2:	Legende zum Algorithmus	4
3:	Die Berechnung des Bauteilpreises	5
5:	Die Darstellung eines Rechtecks	19

Tabellenverzeichnis

1:	Beispielwerte für die Berechnung neuer Bauteilpreise	6
2:	Beispiel für die Berechnung der Fehlerkosten	9
3:	Ergebnisse der Klausur Programmierung I	19

Listingverzeichnis

1:	Die Klasse HelloDHBW	20
----	--------------------------------	----

1 Einleitung

2 Spielwelt

2.1 Abschnitt

Test

2.2 Abschnitt

2.3 Abschnitt

2.4 Abschnitt

2.5 Spiellogik

Beginn: Fredrik

2.5.1 Einleitung

Bereits sehr früh in der Planungsphase des Projekts kam der Wunsch auf, das zu erstellende Planspiel durch den Einsatz flexibler Spielteile möglichst realitätsnah zu gestalten. Hierbei sollten nicht alle Werte wie z.B. Preise für die Dauer des Spiels fest vorgegeben sein, sondern anhand von dynamischen Einflüssen zur Spielzeit schwanken. Dies macht das Spiel für den Spieler einerseits interessanter, da er die Auswirkungen seiner Entscheidungen und derer seiner Mitspieler auf die Spielwelt beobachten kann. Andererseits erhöht dies auch den Lerneffekt, da Spieler die

möglichen Auswirkungen mit in ihre Entscheidungsfindung einbauen und somit verantwortungsvoller handeln müssen.

Um das Planspiel in der vorgegebenen Zeit realisieren zu können, war es nötig, einige Bereiche für diese dynamische Umsetzung auszuwählen und andere außen vor zu lassen. Dabei wurde die Entscheidung gefällt, sowohl die Bauteilpreise, die Absatzmengen auf dem Raumschiffmarkt und die Fehlerkosten flexibel zu gestalten. Nicht in die Betrachtung einbezogen wurden beispielsweise die Personalkosten und die Grundnachfrage auf dem Raumschiffmarkt, die sich stattdessen abhängig von der Spiel-Geschichte entwickeln, um so die Spannung für die Spieler zu erhöhen.

Die folgenden Abschnitte geben einen Überblick über die Berechnungsvorschriften und Abläufe bei der flexiblen Berechnung der oben erwähnten Werte, woran sich zudem die Beschreibung der Bewertung der Unternehmen am Ende des Spiels anschließt.

2.5.2 Bauteilpreise

Die Bauteile dienen den Unternehmen in dem Planspiel zur Herstellung von Raumschiffen. Für die Preisfindung der Unternehmen sind vor allem die variablen Stückkosten maßgeblich, die sich in der vereinfachten Darstellung des Planspiels lediglich aus der Summe der Bauteilpreise zusammensetzen. Hierbei schwanken die Bauteilpreise, was zunächst der Dynamik des Spiels zu Gute kommt, weil erreicht wird, dass Spieler auf die geänderten Bedingungen reagieren müssen, indem sie erkennen, dass gewissen Bauteilpreise gesunken sind.

Die Grundidee zur Veränderung der Bauteilpreise stammt aus der Betrachtung von Angebot und Nachfrage in der Makroökonomie. Wie in Abbildung 1 zu erkennen ist, führt die Erhöhung der Nachfrage bei gleich bleibendem Angebot zu einem neuen Marktgleichgewicht bei höheren Preisen. Ausgehend hiervon wurde der Algorithmus entwickelt, der durch Gleichung 1, Gleichung 2 und Gleichung 3 beschrieben wird. Die Bedeutung der Bezeichner ist Abbildung 2 zu entnehmen.

$$\overline{N} = \frac{1}{n} \cdot \sum_{i=1}^n x_i \cdot p_i \quad (1)$$

Mit Gleichung 1 wird für jede einzelne Bauteilgruppe (Standard- und Sonderbauteile) zunächst die durchschnittliche Nachfrage in der zurückliegenden Periode berechnet. Die einzelnen Bauteilgruppen müssen hierbei getrennt betrachtet werden,

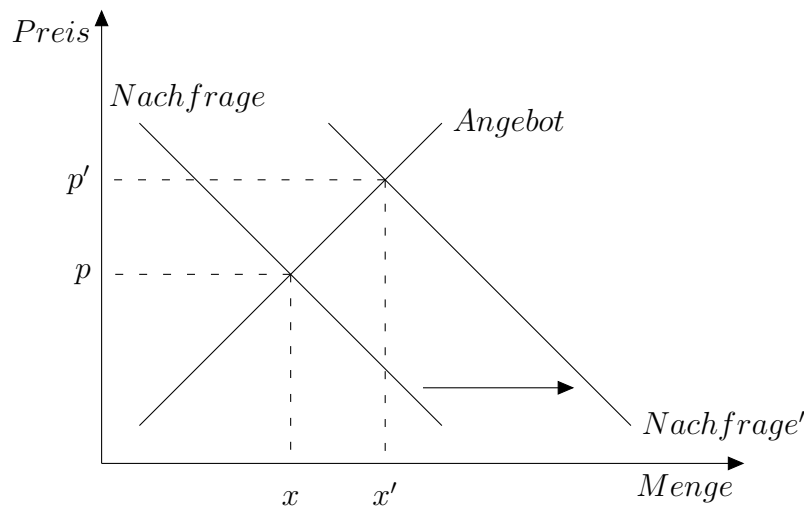


Abbildung 1: Die Beziehung zwischen Angebot und Nachfrage

\bar{N} : durchschnittliche Nachfrage

x_i : von Bauteil i in zurückliegender Runde abgesetzte Menge

p_i : Preis des Bauteils i in der zurückliegenden Runde

ΔN_{rel} : relative Nachfrageabweichung eines Produkts von \bar{N}

p_{neu} : neuer Preis des Bauteils

p_{basis} : Grundpreis des Bauteils

Δp_{max} : maximal erlaubte Abweichung vom Grundpreis

Abbildung 2: Legende zum Algorithmus

da die Sonderbauteile nur einen relativ geringen Anteil am Wert eines Raumschiffes haben, und die durchschnittliche Nachfrage zwischen den Gruppen somit nicht vergleichbar ist. Die Nachfrage wird am Umsatz gemessen, da beispielsweise von den teuren Triebwerken nur wenig, von den günstigen Rumpfbauteilen aber viele für ein Raumschiff benötigt werden.

$$\Delta N_{rel} = \frac{x_i \cdot p_i - \bar{N}}{\bar{N}} \quad (2)$$

In einem zweiten Schritt wird dann für jedes einzelne Bauteil der betrachteten Gruppe mit Gleichung 2 die relative Abweichung der Nachfrage für dieses Bauteil von der durchschnittlichen Nachfrage gebildet. Diese Abweichung kommt zu Stande, da

sich die Mengen der benötigten Bauteile auf die verschiedenen Raumschiffotypen in unterschiedlichen Verhältnissen verteilen. So führt eine überdurchschnittlich hohe Produktion von Millennium Falken beispielsweise zu einer überdurchschnittlich hohen Nachfrage nach Hitzeschilden, wie im Kapitel 2.3 ersichtlich wird.

$$p_{neu} = p_{basis} - \Delta p_{max} + \frac{2 \cdot \Delta p_{max}^2}{\Delta p_{max} + \frac{\Delta p_{max}}{2 \cdot \Delta p_{max} \cdot \Delta N_{rel}} \left(\frac{2}{\Delta p_{max}} + 1 \right)} \quad (3)$$

Zu guter letzt wird dann, ausgehend von der errechneten relative Abweichung zur durchschnittlichen Nachfrage, der neue Preis des Bauteils anhand von Gleichung 3 berechnet. Lag eine überdurchschnittlich hohe Nachfrage vor, ist also die relative Abweichung positiv, so steigt der Bauteilpreis; entsprechendes gilt auch umgekehrt. Wie sich der Preis genau verhält, ist abhängig von dem Grundpreis des Bauteils und der maximal erlaubten Abweichung von diesem Preis nach oben und unten. Entwickelt wurde diese kompliziert anmutende Funktionsvorschrift aus der Funktion des logistischen Wachstums, die dann dahingehend parametrisiert wurde, dass der Schnittpunkt mit der x -Achse als Basispreis und die maximale Preisabweichung frei gewählt werden können. Hierdurch entsteht für den einfachen Sachverhalt eine sehr verschachtelte Funktion, die jedoch nur eine Abwandlung einer einfachen Funktion ist.

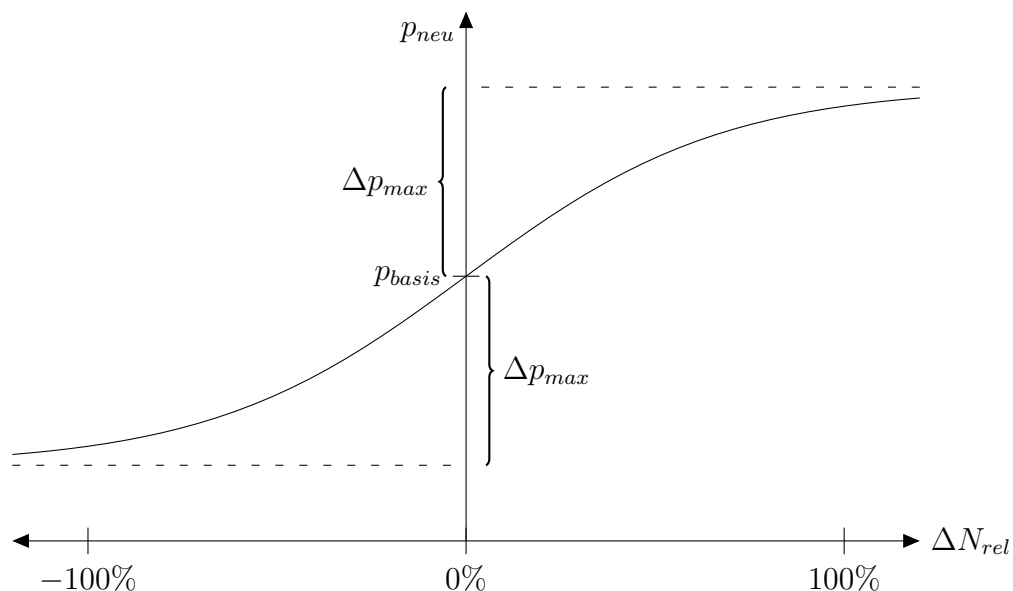


Abbildung 3: Die Berechnung des Bauteilpreises

Zur Verdeutlichung der Berechnung des letzten Schritts ist in Abbildung 3 der Funktionsgraph der Gleichung dargestellt. In diesem wird ersichtlich, wie sich der Preis eines Bauteils bei einer gegebenen Abweichung von der durchschnittlichen Nachfrage verhält. Zu erkennen sind ebenfalls die oben erwähnten Parameter, nämlich der Grundpreis p_{basis} und die maximale Abweichung Δp_{max} .

Bauteiltyp	Absatzmenge	Preis
Rumpfbauteil	130 Stück	100,00€
Hitzeschild	60 Stück	200,00€
Triebwerk	40 Stück	500,00€

Tabelle 1: Beispielwerte für die Berechnung neuer Bauteilpreise

Anhand eines Beispiels soll die Berechnung der Bauteilpreise abschließend betrachtet werden. Zu diesem Zweck nehme man die Werte in Tabelle 1 als die Werte für die zurückliegende Spielrunde an. Folgt man nun der Berechnungsvorschrift, so muss zunächst mit Hilfe von Gleichung 1 die durchschnittliche Nachfrage berechnet werden.

$$\bar{N} = \frac{1}{3} \cdot (130 \cdot 100\text{€} + 60 \cdot 200\text{€} + 40 \cdot 500\text{€}) = 15000\text{€}$$

Im Durchschnitt wurde in der vergangenen Spielrunde also für jedes Standardbauteil insgesamt 15000€ ausgegeben. Diese durchschnittliche Nachfrage, gemessen am Umsatz, wird dann verglichen mit der Nachfrage nach einem bestimmten Bauteil, beispielsweise dem Hitzeschild. Hierzu wird Gleichung 2 angewendet.

$$\Delta N_{rel} = \frac{60 \cdot 200\text{€} - 15000\text{€}}{15000\text{€}} = -0.2 \hat{=} -20\%$$

Die negative relative Abweichung sagt aus, dass im Vergleich zu den anderen Bauteilen, weniger Hitzeschilder eingekauft wurden, die Nachfrage nach diesen war also vergleichsweise gering. Wie sich dies auf den Preis für Hitzeschilder in der nächsten Runde auswirkt, ergibt sich aus Gleichung 3. Dazu müssen der Datenbasis in Kapitel 2.3 noch die Werte für den Basispreis $p_{basis} = 200,00\text{€}$ und die maximal erlaubte Abweichung von diesem $\Delta p_{max} = 80,00\text{€}$ entnommen werden.

$$p_{neu} = 200\text{€} - 80\text{€} + \frac{2 \cdot 80\text{€}^2}{80\text{€} + \frac{80\text{€}}{\left(\frac{2}{80\text{€}} + 1\right) \cdot 2 \cdot 80\text{€} \cdot (-0.2)}} = 169.94\text{€}$$

Aufgrund der vergleichsweise niedrigen Nachfrage ergibt sich also in der nächsten Spielrunde ein niedrigerer Preis für Hitzeschilder, nämlich statt 200,00€ nur noch 169,94€, was die Simulation des Zusammenhangs zwischen Angebot und Nachfrage in Abbildung 1 ausdrückt. Geschickte Spieler können sogar auf besonders niedrige Bauteilpreise reagieren, indem sie diese in Massen auf Vorrat einkaufen und so einerseits selbst Geld sparen, andererseits den Preis der Bauteile in der Folgerunde für die anderen Spieler in die Höhe treiben.

Ende: Fredrik

Beginn: Philipp

2.5.3 Absatzmengen

2.5.4 Fehlerkosten

Da es, wie in den vorherigen Kapitel genauer beschrieben, mehrere Personaltypen gibt, die sich in ihrer Qualität unterscheiden und je nach Qualitätsstufe unterschiedliche Kosten bei der Produktion verursachen, musste ein Algorithmus definiert werden, der diese Mehrkosten in jeder Spielrunde ermittelt.

Zur Ermittlung der Mehrkosten werden Reparaturkosten, die für die Reparatur eines fehlerhaften Raumschiffs anfallen, herangezogen. Das heißt also, wenn ein Unternehmen ein fehlerhaftes Raumschiff produziert, wird das Konto des Unternehmens mit Mehrkosten, die für die Reparatur oder Behebung des Fehlers anfallen, belastet.

Andere Überlegungen waren beispielsweise der Ausschuss ganzer Raumschiffe oder der Ausfall einzelner Bauteile. Vergleicht man den Ausschuss bei der Raumschiffproduktion beispielsweise mit der der Automobilproduktion, so stellt man fest, dass es beim Zusammenbau einzelner Fertigbauteile nicht zum Ausschuss eines kompletten Endprodukts kommen kann. Daher erschien uns dieser Ansatz zum einen äußerst

unrealistisch und zum anderen würde der Ausschuss eines ganzen Raumschiffs aufgrund des hohen Produktionspreises einen zu hohen wirtschaftlichen Nachteil für das jeweilige Unternehmen darstellen. Um das Planspiel nicht zu sehr von Zufallsergebnissen abhängig zu machen wurde daher auf diese Variante verzichtet.

Auch wenn der Ausfall einzelner Bauteile sehr realistisch erscheint und auch finanziell für die Unternehmen zu verkraften wäre, würden sich bei dieser Variante andere Probleme ergeben. So ist es einem Unternehmen möglich Fertigbauteile einzukaufen um daraus Raumschiffe zu produzieren. Fällt nun ein Teil aus, sind möglicherweise nicht genügend Bauteile auf Lager, um das Raumschiff trotz des Ausfalls eines Bauteils produzieren zu können. In der Realität hätte das Unternehmen die Chance, Teile kurzfristig dazuzukaufen, wohingegen beim Planspiel das Unternehmen nach dem Einchecken der Spielrunde keine Möglichkeit hat etwas am Einkauf oder der Produktion für die jeweilige Runde zu ändern. Aus diesem Grund wurde auch diese Variante als mangelhaft eingestuft und man hat sich auf die am besten ins Planspiel eingliederbare und im nachfolgenden Abschnitt beschriebene Variante der Reparaturkosten entschieden.

Aufgrund der Tatsache, dass sich das Personal eines Unternehmens aus allen drei Personaltypen zusammensetzen kann, ist zunächst die durchschnittliche Personalqualität, des im Unternehmen eingesetzten Personals zu bestimmen:

$$\bar{q} = \frac{1}{n} \cdot \sum_{i=1}^n q_i \cdot n_i \quad (4)$$

Mit dieser Formel wird die Durchschnittsqualität des Personals als gewichtetes arithmetisches Mittel berechnet. Die Durchschnittsqualität eines Unternehmens liegt in jedem Fall im Bereich 0,69 - 0,99.

Mit Hilfe dieser Information ist man nun in der Lage, die durch das Personal verursachte fehlerhaft produzierte Menge zu berechnen. Zur Berechnung wird die in Java vorgefertigte `Math.random()`-Funktion der Klasse `random` verwendet. Diese Funktion generiert für jedes in Produktion gegebene Raumschiff eine Zufallszahl zwischen 0,0 und 1,0. Sollte die erzeugte Zufallszahl den Wert der Durchschnittsqualität des Personals überbieten, so wird dieses Raumschiff als fehlerhaft eingestuft. Ist die Qualität des Personals also höher, so erhöht sich die Wahrscheinlichkeit fehlerfreie Raumschiffe zu produzieren. Wiederholt man diesen Vorgang für jedes produzierte Raumschiff, so lassen sich auf diese Weise die anfallenden Mehrkosten errechnen.

ACHTUNG!!!

TEXT EINFÜGEN

ACHTUNG!!!

Raumschiff	Zufallszahl	Fehlerhaft?
X-Wing	0,3	Nein
Corellian Corvette	0,9	Ja
Corellian Corvette	0,7	Nein
Millenium Falke	0,1	Nein

Tabelle 2: Beispiel für die Berechnung der Fehlerkosten

Im Beispiel aus Tabelle 2 möchte ich dies genauer aufzeigen. Beim ersten Raumschiff erzeugt die Funktion `Math.random()` die Zufallszahl 0,3. Da diese die Personalqualität von 0,772068 nicht überbietet, liegt kein Fehler vor. Dagegen weist eine der Corellian Corvettes einen Fehler auf. Die zweite Corellian Corvette und der Millenium Falke wurden wieder fehlerfrei produziert. Aufgrund der Qualität des Personals liegen bei einem von vier Raumschiffen Fehler vor. Hätte das Unternehmen verstärkt in ihr Personal investiert und in Form von Schulungen oder Neueinstellungen die Qualität verbessert, so hätte dieser Fehler möglicherweise vermieden werden können.

ACHTUNG!!!

TEXT EINFÜGEN

ACHTUNG!!!

2.5.5 Bewertung

.

Ende: Philipp

3 Fachkonzept

3.1 Abschnitt

Beginn: Fredrik

3.2 Entwurfsentscheidungen

3.2.1 Einleitung

Zur Erstellung eines Unternehmensplanspiels bedarf es - wie in jedem Prozess der Software-Erstellung - nach einer anfänglichen Analyse der benötigten Funktionalität, deren Ergebnisse in Kapitel 3.1 anhand eines Use Case Diagramms verkürzt dargestellt wurden, Entscheidungen zum grundlegenden Entwurf der Software. Diese Phase grenzt sich klar ab von den Entscheidungen zur Implementierung des Projekts, die in Kapitel 3.3 erläutert werden, da sie sich auf einem höheren Abstraktionsniveau bewegt. Es geht hierbei nicht darum, die Lösung auf eine bestimmte Umsetzung hin zu optimieren, sondern die erkannten Anforderungen weitestgehend zu ermöglichen. Diese Entscheidungen sollen strukturiert werden mit Hilfe der Notation des UML Klassendiagramms.

Konkret bedeutet der höhere Abstraktionsgrad, dass Eigenheiten wie die System-Architektur (z.B. Client-Server-Architektur), die Persistierung von Daten oder die Benutzeroberfläche an dieser Stelle weder behandelt noch berücksichtigt werden sollen. Es wird vielmehr die Spielwelt, die eingefasst ist in einen Rahmen, der den Zugriff durch Spielleiter und Spieler von Außen ermöglicht, modelliert.

Abbildung 4: Klassendiagramm der Entwurfsphase

In ?? ist das Klassendiagramm der Entwurfsphase dargestellt. Die Erstellung dieses Modell orientierte sich dabei zunächst an der erdachten Spielwelt.

Spiel: zentrale Klasse keine trivialen getter und setter

Im Folgenden sollen einige Bereiche aus diesem Modell getrennt voneinander betrachtet werden. Zunächst werden die sogenannten Typen, die Beschreibungen von konkreten Gegenständen der Spielwelt wie z.B. den Raumschifftypen darstellen, betrachtet. Daran schließt sich eine Erläuterung der Märkte an, gefolgt von einer Betrachtung der Klasse Unternehmen. Zuletzt werden die zur Erfassung der Vorgänge in der Spielwelt und deren Speicherung dienenden Transaktions-Klassen und die SpielRunde erörtert.

3.2.2 Die Typen

Der Begriff *Typen* bezeichnet, wie oben bereits erwähnt, eine Beschreibung von einer Gruppe von Gegenständen der Spielwelt. Es wurde bewusst auf eine Modellierung jedes einzelnen Objekts in der Spielwelt verzichtet, um die Anzahl der benötigten Objekte nicht Überhand nehmen zu lassen. Obgleich diese Entscheidung vielleicht wie ein Vorgriff auf den Abstraktionsgrad der Implementierung wirkt, wurde hierdurch versucht, die Schere zwischen dem Modell der Entwurfs- und dem der Implementierungsphase möglichst gering zu halten, vor allem da sich die gewählte Möglichkeit nicht negativ auf die Funktionalität auswirkt.

Erkannt und umgesetzt wurde dabei die folgenden Typen, die ihrerseits von der Klassen *Typ* erben:

- **PersonalTyp**
- **BauteilTyp**
- **RaumschiffTyp**

3.2.3 Die Märkte

Aus der Definition des Spielszenarios lassen sich folgende Märkte ableiten:

- **Raumschiffmarkt:** Dieser Markt ist vermutlich derjenige, der sich bei der Betrachtung von Raumschiffproduktions-Unternehmen in den Vordergrund drängt.

Es handelt sich für die Spieler um den Absatzmarkt, an dem sie Verkaufsangebote machen können, die sich durch eine Menge und einen Preis auszeichnen. Alle Angebote, die im Laufe der Runde gemacht werden, müssen zwischengespeichert und zum Schluss der Runde ausgewertet werden, um die Absatzwerte berechnen zu können. Zusätzlich verwaltet der Raumschiffmarkt alle Raumschifftypen der Spielwelt.

Referenz auf die Felder / Methoden.

- **Bauteilmarkt:** Ähnlich

Synergien!?

3.2.4 Das Unternehmen

3.2.5 Spielrunde und Transaktionen

Ende: Fredrik

3.3 Implementierung

System-Architektur (Client-Server) Daten-Persistierung Benutzeroberfläche

BauteilTypArt als enum Assoziationsklassen als HashMaps

4 UI Mockups

5 jUnit Tests

6 Fazit und Ausblick

Beginn: Julia, Philipp

Während der Entwicklung des Planspiels wurden einige Ideen und Möglichkeiten diskutiert, welche den vorgegebenen Zeitrahmen gesprengt hätten.

Eine Überlegung bezüglich des Finanzwesens war es, anstelle des automatisch eingeräumten Dispositionskredits mit festem Zinssatz dem Spieler eine manuelle Kreditaufnahme anzubieten. Auf diese Weise wäre es dem Spieler möglich vorausschauend den Kreditrahmen zu bestimmen und diesen zu einem angegebenen Zinssatz aufzunehmen. Dieser Zinssatz würde dadurch von der Geldnachfrage abhängig sein und sich von Runde zu Runde verändern. Somit würde dem Spieler mehr Verantwortung und Entscheidungskraft übertragen werden.

Mit der Einführung von zusätzlichen Maschinen, welche der Spieler während einer Spielrunde erwerben könnte, wäre das Planspiel realistischer aufgebaut. Durch diese Maschinen könnte die Produktionskapazität weiter erhöht, neue Funktionen in das Raumschiff eingebaut oder der Ausschuss verringert werden. Das zuvor aufgenommene Darlehen könnte beispielsweise zur Finanzierung einer neuen Maschine verwendet werden.

Im Spielverlauf könnte die revolutionäre Erfindung des Warp-Antriebs die Nachfrage erheblich beeinträchtigen. Dieser Antrieb ermöglicht den Raumschiffen mit Warp-Geschwindigkeit das Weltall zu durchqueren. Nach Einführung des Warp-Antriebs würde die Nachfrage nach Raumschiffen mit herkömmlichen Antrieben stark zurück gehen, was den Spieler dazu zwingen würde, die alten Modelle schnellstmöglich abzusetzen. Desweiteren müsste der Spieler seinen Maschinenpark aufrüsten um der Nachfrage nach den modernen Raumschiffen gerecht zu werden.

Um noch mehr Dynamik in das Spiel zu bringen, könnte man in bestimmten Runden die Nachfrage nach Raumschiffen durch Zusatzaufträge erhöhen. Diese kann der Spieler nach Prüfung der Rentabilität annehmen oder ablehnen. Auf diese Weise könnte der Spieler freie Kapazitäten nutzen, seinen Gewinn steigern sowie das Image verbessern.

Bemerkt der Spieler einen Rückgang seiner Absatzzahlen so wäre es ihm mit dem so genannten “Macht”-Button möglich, die Entscheidungen der Abnehmer zu seinen Gunsten zu manipulieren. Allerdings würde dann das Risiko bestehen, durch die galaktischen Ordnungshüter für eine Runde auf den Planeten Despayre verbannt zu werden. Während dem Aufenthalt auf Despayre wäre es dem Spieler nicht möglich im Namen seines Unternehmens zu handeln.

Einen Blick auf die technischen Aspekte des Planspiels lässt weitere Punkte offen, welche noch realisiert werden könnten.

Um das Planspiel benutzerfreundlich zu gestalten, wäre eine intuitive Benutzeroberfläche von Nöten. Diese könnte sich an den bereits vorgestellten Mockups orientieren. Darauf aufbauend könnte sich jeder Spieler selbstständig am System anmelden und zur Startseite gelangen. Würde man zudem die Funktionen des Spieleiters in Java-Code umsetzen, so würde das Spiel diese Aufgaben automatisch übernehmen. Auf diese Weise könnten auch unerfahrene Spieler das Planspiel eigenständig durchführen.

In der aktuellen Version des Planspiels werden die entstehenden Daten in Java mithilfe von Hashmaps und Vektoren realisiert. Um die Performance und die Übersichtlichkeit des Planspiels zu erhöhen wäre eine Datenbankbindung sinnvoll. Die zentrale Speicherung auf die Daten würde zudem einen einfacheren Zugriff auf abgespeicherte Daten ermöglichen. Der Hauptvorteil einer Datenbankbindung würde in der dauerhaften Speicherung der Daten liegen. Auf diese Weise wären die gespeicherten Informationen auch nach dem Schließen des Planspiels abrufbar.

Abschließend ist zu sagen, dass sowohl die Planung als auch die Durchführung des Projekts reibungslos von statten ging. Zum ständigen Austausch der Teilnehmer fanden regelmäßig Besprechungen statt, welche zur optimalen Abstimmung untereinander führte. Um die Arbeit an einem Projekt mit mehreren Personen zu koordinieren wurde das verteilte Versionsverwaltungssystem Git genutzt, das den Austausch von Dateien vereinfacht.

Ende: Julia, Philipp

A Einige wichtige L^AT_EX-Kommandos

A.1 Kommandos für die Erzeugung von Literaturverweisen

Das Kommando `\seCite{par1}{par2}{par3}` erzeugt einen Literaturverweis im Text.

par1: Der erste Parameter definiert einen optionalen Text, der vor dem eigentlichen Literaturverweis ausgegeben wird, typischerweise Vgl. oder vgl.

par2: Der zweite Parameter wird verwendet, um (z. B.) zusätzliche Seitenangaben für den Literaturverweis vorzunehmen.

par2: Der dritte Parameter ist der entsprechende Schlüssel in der .bib-Datei, in der die Literaturquellen beschrieben sind (vgl. `wa.bib`).

Als Beispiel für die Verwendung des `\seCite`-Befehls dient folgendes Zitat: „Die **Funktion** eines Anhangs einer wissenschaftlichen Arbeit wird sehr häufig **missdeutet**, der Anhang selbst nicht selten **mißbraucht**.“ (*Theisen*, Wissenschaftliches Arbeiten, 2009, S. 170).

Bei der von Theisen vorgeschlagenen Zitierweise erfolgt die Angabe der Literaturverweise in der Regel innerhalb einer Fußnote. Hierfür kann das Kommando `\seFootcite` verwendet werden, das dieselben Parameter wie `\seCite` besitzt.

Als Beispiel für ein indirektes Zitat lässt sich die Aussage von Theisen anführen, dass Hauptinhalte eines (berechtigten) Anhangs ergänzende Materialien und Dokumente sind, die weitere themenbezogene Informationen liefern können.¹

Weder das `\seFootcite`- noch das `\footnote`-Kommando können bei Gleitobjekten (Verwendung der `figure`-, `table`- oder `programm`-Umgebung) verwendet werden.

¹ Vgl. *Theisen*, Wissenschaftliches Arbeiten, 2009, S. 171.

Ein kleiner Workaround, um L^AT_EX doch dazu zu bringen, Fußnoten bei Gleitobjekten zu akzeptieren, ist in Kapitel A.3 auf S. 19 zu finden.

A.2 Kommandos für die Erzeugung von Abkürzungen, Symbolen und Glossareinträgen

Für Abkürzungen, Symbole und Glossareinträge wird das Kommando `\gls{par1}` verwendet. `par1` stellt einen Schlüssel dar, der die entsprechende Definition identifiziert (vgl. den Inhalt der Datei `pa1-abkuerzungen.tex`).

Eine Abkürzung: Universal Serial Bus (USB); das zweite Auftreten der Abkürzung: USB. Und jetzt kommt ein Symbol: π ; das zweite Symbol ist *ND*.

Und auch ein Eintrag im Glossar muss sein: Active Directory^{GL}; das zweite Auftreten des Eintrags ist Active Directory.

Und auf der nächsten Seite: Active Directory. Im Glossar ist jeweils angegeben, auf welchen Seiten der Begriff verwendet wurde.

A.3 Abbildungen, Tabellen und Programmlistings

Ein Rechteck besitzt die in Abbildung 5 auf S. 19 dargestellte Struktur.

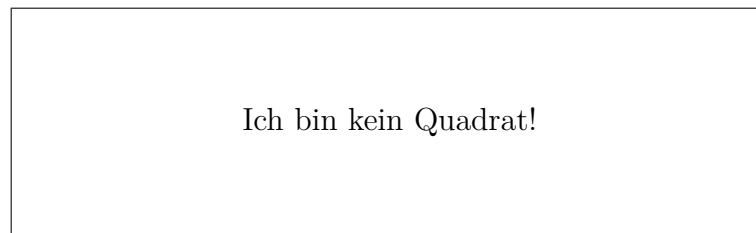


Abbildung 5: Die Darstellung eines Rechtecks²

Der optionale Parameter im folgenden `\caption`-Kommando

```
\caption[Die Darstellung eines Rechtecks]%  
{Die Darstellung eines Rechtecks\label{abb1}\footnotemark}
```

definiert den Eintrag für das Abbildungsverzeichnis. Dort sollte die Fußnotennummer nicht auftauchen. Nutzt man den optionalen Parameter nicht, ist es notwendig, vor `\footnotemark` noch ein `\protect` einzufügen, da L^AT_EX andernfalls die Übersetzung mit einer Fehlermeldung abbricht.

Eine Notentabelle kann wie in Tabelle 3 auf S. 19 dargestellt aussehen.

Matrikelnummer	Note
1234567	2,7
2323456	3,5
9865783	1,0

Tabelle 3: Ergebnisse der Klausur Programmierung I

Eines der wichtigsten Java-Programme überhaupt ist in Listing 1 auf S. 20 zu sehen.

² Vgl. *Theisen*, Wissenschaftliches Arbeiten, 2009, S. 400. Achtung: Dieser Literaturverweis ist rein fiktiver Natur, die Seite 400 existiert in *Theisen*, Wissenschaftliches Arbeiten, 2009 nicht!

```
public class HelloDHBW {  
    public static void main ( String[] args ) {  
        System.out.println ( "Hello_DHBW" );  
    } // main  
} // HelloDHBW
```

Listing 1: Die Klasse HelloDHBW

A.4 Die Definition und Anwendung von zwei neuen Listenumgebungen

A.4.1 Das Layout der Standardlistenumgebung von L^AT_EX

Stichpunktlisten werden in L^AT_EX mit der `itemize`-Umgebung erzeugt. Die Stichpunktliste

- 1. Stichpunkt der ersten Ebene
 - 1. Stichpunkt der zweiten Ebene
 - 2. Stichpunkt der zweiten Ebene
 - * 1. Stichpunkt der dritten Ebene
 - * 2. Stichpunkt der dritten Ebene
 - 1. Stichpunkt der vierten Ebene
 - 2. Stichpunkt der vierten Ebene
- 2. Stichpunkt der ersten Ebene
- 3. Stichpunkt der ersten Ebene

wird durch die folgenden Anweisungen erreicht:

```
\begin{itemize}  
\item 1. Stichpunkt der ersten Ebene  
\begin{itemize}
```

```

\item 1. Stichpunkt der zweiten Ebene
\item 2. Stichpunkt der zweiten Ebene
\begin{itemize}
\item 1. Stichpunkt der dritten Ebene
\item 2. Stichpunkt der dritten Ebene
\begin{itemize}
\item 1. Stichpunkt der vierten Ebene
\item 2. Stichpunkt der vierten Ebene
\end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}
\item 2. Stichpunkt der ersten Ebene
\item 3. Stichpunkt der ersten Ebene
\end{itemize}

```

A.4.2 Die neue Listenumgebung `seList` für Stichpunktlisten

Weder die Einrückung der einzelnen Ebenen noch die großen Abstände zwischen den einzelnen Stichpunkten sind bei der `itemize`-Umgebung bezüglich des Layouts sonderlich überzeugend.

Daher wird eine neue `seList`-Umgebung zur Verfügung gestellt.

- 1. Stichpunkt der ersten Ebene
 - 1. Stichpunkt der zweiten Ebene
 - 2. Stichpunkt der zweiten Ebene
 - ◊ 1. Stichpunkt der dritten Ebene
 - ◊ 2. Stichpunkt der dritten Ebene
 - ▷ 1. Stichpunkt der vierten Ebene
 - ▷ 2. Stichpunkt der vierten Ebene
 - 1. Stichpunkt der fünften Ebene
 - 2. Stichpunkt der fünften Ebene
- 2. Stichpunkt der ersten Ebene
- 3. Stichpunkt der ersten Ebene

Der L^AT_EX-Quelltext für diese Liste ist:

```

\begin{seList}
\item 1. Stichpunkt der ersten Ebene
\begin{seList}
\item 1. Stichpunkt der zweiten Ebene
\item 2. Stichpunkt der zweiten Ebene
\begin{seList}
\item 1. Stichpunkt der dritten Ebene
\item 2. Stichpunkt der dritten Ebene
\begin{seList}
\item 1. Stichpunkt der vierten Ebene
\item 2. Stichpunkt der vierten Ebene
\begin{seList}
\item 1. Stichpunkt der f\"unften Ebene
\item 2. Stichpunkt der f\"unften Ebene
\end{seList}
\end{seList}
\end{seList}
\end{seList}
\item 2. Stichpunkt der ersten Ebene
\item 3. Stichpunkt der ersten Ebene
\end{seList}

```

Neben der Eigenschaft, im Gegensatz zur `itemize`-Umgebung fünf Verschachtelungsebenen angeben zu können, ist es möglich, die Zeilenabstände für die einzelnen Ebenen zu konfigurieren.

Mit dem Kommando

```
\seSetlistbaselineskip{b1}{b2}{b3}{b4}{b5}
```

kann für die Verschachtelungsebene i der Grundlinienabstand b_i festgelegt werden. Als Einheit wird der Wert von `\baselineskip` (Grundlinienabstand des Dokuments) verwendet. Die folgenden Werte sind für ein Dokument voreingestellt:

```
\seSetlistbaselineskip{1}{0.75}{0.75}{0.75}{0.75}
```

Mit dem Kommando

```
\seResetlistbaselineskip{}
```

wird die letzte Änderung der Werte rückgängig gemacht.

A.4.3 Die neue Listenumgebung `seToplist` für Listen mit einem Label und Aufzählungslisten

Die neue Listenumgebung `seToplist` erlaubt es, jeden Stichpunkt mit einem Label zu versehen. Die Liste³

```
Audi:          400000 Gesamtverkäufe
               A4:          200000 Verkäufe
               A5:          50000 Verkäufe
               A6:          150000 Verkäufe

Mercedes Benz: 500000 Gesamtverkäufe

BMW:           650000 Gesamtverkäufe
               1er Reihe: 100000 Verkäufe
               3er Reihe: 300000 Verkäufe
               5er Reihe: 250000 Verkäufe
```

wird durch die folgenden L^AT_EX-Anweisungen erzeugt:

```
\begin{seToplist}{Mercedes Benz:}
\item[Audi:] 400000 Gesamtverk\"aufe
\begin{seToplist}{3er Reihe:}
\item[A4:] 200000 Verk\"aufe
\item[A5:] 50000 Verk\"aufe
\item[A6:] 150000 Verk\"aufe
\end{seToplist}
\item[Mercedes Benz:] 500000 Gesamtverk\"aufe
\item[BMW:] 650000 Gesamtverk\"aufe
\begin{seToplist}{3er Reihe:}
\item[1er Reihe:] 100000 Verk\"aufe
\item[3er Reihe:] 300000 Verk\"aufe
\item[5er Reihe:] 250000 Verk\"aufe
\end{seToplist}
\end{seToplist}
```

Der Parameter `par` von `\begin{seToplist}{par}` definiert die Breite des Labels für die zugehörige Liste.

³ Die folgenden Werte sind frei erfunden.

Für die `seToplist`-Umgebung können ebenfalls fünf Verschachtelungsebenen definiert werden. Über die Kommandos

`\seSettoplistbaselineskip{b1}{b2}{b3}{b4}{b5}`

bzw.

`\seResettolistbaselineskip{}`

lassen sich analog zur `seList`-Umgebung die Grundlinienabstände der einzelnen Verschachtelungsebenen verändern bzw. zurücksetzen. Die folgenden Werte sind für ein Dokument voreingestellt:

`\seSettoplistbaselineskip{1}{0.75}{0.75}{0.75}{0.75}`

Durch eine entsprechende Wahl der Labels können Aufzählungslisten erzeugt werden:

a) Deutsche Automarken

- 1) Mercedes Benz
- 2) Audi
- 3) VW
- 4) BMW

b) Japanische Automarken

- 1) Toyota
- 2) Honda
- 3) Mazda

B Hinweise zur Installation und Übersetzung

B.1 Verwendung von TeXShop (Apple-Welt)

Unter den ausgelieferten Dateien befinden sich zwei **engine**-Dateien:

- `dhbw-projektarbeit.engine`
- `dhbw-projektarbeit-remove-all.engine` (löscht alle erzeugten *Hilfsdateien*)

Mit jeder dieser beiden Dateien kann man die Vorlage `se-sem-vorlage.tex` übersetzen. Alle Verzeichnisse (insbesondere Abkürzungs- und Symbolverzeichnis) sowie das Glossar werden (hoffentlich) korrekt erstellt.

In den engine-Dateien ist beschrieben, an welcher Stelle sie im Mac OS X Dateisystem installiert werden müssen, damit man sie direkt von TeXShop aus aufrufen kann.

B.2 Verwendung von MiKTeX (Windows-Welt)

Für die Übersetzung wird eine batch-Datei `make-projektarbeit.bat` zur Verfügung gestellt, mit der man in der Windows-*Eingabeaufforderung* (cmd) die Vorlage übersetzen kann. Der Aufruf lautet: `make-projektarbeit.bat se-sem-vorlage`

Da MiKTeX eine andere Version von `jurabib` verwendet, mit der sich die Vorlage nicht korrekt übersetzen lässt, werden die beiden Dateien

- `jurabib.sty` und
- `jurabib.bst`

aus der TeX Live Version von Mac OS X mitgeliefert. Damit sollte die Übersetzung problemlos funktionieren.

Glossar

Active Directory

Active Directory ist in einem Windows 2000/Windows Server 2003-Netzwerk der Verzeichnisdienst, der die zentrale Organisation und Verwaltung aller Netzwerkressourcen erlaubt. Es ermöglicht den Benutzern über eine einzige zentrale Anmeldung den Zugriff auf alle Ressourcen und den Administratoren die zentral organisierte Verwaltung, transparent von der Netzwerktopologie und den eingesetzten Netzwerkprotokollen. Das dafür benötigte Betriebssystem ist entweder Windows 2000 Server oder Windows Server 2003, welches auf dem zentralen Domänencontroller installiert wird. Dieser hält alle Daten des Active Directory vor, wie z. B. Benutzernamen und Kennwörter.⁴ 18, 19

⁴ Vgl. *Dudenredaktion*, Duden, 2009, S. 200.

Literaturverzeichnis

- Andermann, Ulrich/Drees, Martin/Grätz, Frank* [Verfassen wissenschaftlicher Arbeiten, 2006]: Duden. Wie verfasst man wissenschaftliche Arbeiten? Ein Leitfaden für das Studium und die Promotion. 4. Auflage. Dudenverlag, 2006, ISBN 978-3-411-05113-7
- Balzert, Helmut et al.* [Wissenschaftliches Arbeiten, 2008]: Wissenschaftliches Arbeiten: Wissenschaft, Quellen, Artefakte, Organisation, Präsentation. Herdecke, Witten: W3L-Verlag, 2008, ISBN 978-3-937137-59-9
- Bänsch, Axel/Alewell, Dorothea* [Wissenschaftliches Arbeiten, 2009]: Wissenschaftliches Arbeiten. 10. Auflage. München: Oldenbourg Verlag, 2009, ISBN 978-3-4865-9090-6
- Bonsiepen, Lena/Coy, Wolfgang* [Curriculardebatte, 1992]: Eine Curriculardebatte. Informatik-Spektrum, 15 1992, Nr. 6, 323–325
- Brink, Alfred* [Anfertigung wissenschaftlicher Arbeiten, 2007]: Anfertigung wissenschaftlicher Arbeiten – Ein prozessorientierter Leitfaden zur Erstellung von Bachelor-, Master- und Diplomarbeiten in acht Lerneinheiten. 3. Auflage. München Wien: Oldenbourg Verlag, 2007, ISBN 978-3-486-58512-4
- Brooks, Frederick P.* [No Silver Bullet-Essence, 1987]: No Silver Bullet-Essence and Accidents of Software Engineering. IEEE Computer, 20 1987, Nr. 4, 10–19
- Denning, Peter J. et al.* [Computing as a Disciplin, 1989]: Computing as a Disciplin. Communications of the ACM (CACM), Januar 1989, 9–23
- Dijkstra, Edsger W.* [Teaching Computer Science, 1989]: On the Cruelty of Really Teaching Computer Science. Communications of the ACM (CACM), 32 1989, Nr. 12, 1398–1404
- Duale Hochschule Baden-Württemberg* [Corporate Design, 2009]: Corporate De-

- sign. Moodle Kursraum 10SWMA – Wiss. Arbeiten, DHBW Mannheim, 2009, styleguide-DHBW_CDM_090610_kap1-1.3.pdf
- Dudenredaktion (Hrsg.)* [Duden, 2006]: DUDEN – Die deutsche Rechtschreibung. 24. Auflage. Mannheim: Bibliographisches Institut & F. A. Brockhaus AG, 2006, ISBN 978-3-411-04014-9
- Dudenredaktion (Hrsg.)* [Duden, 2009]: DUDEN – Die deutsche Rechtschreibung. 25. Auflage. Mannheim: Bibliographisches Institut & F. A. Brockhaus AG, 2009, ISBN 978-3-411-04015-5
- Freytag, Jürgen* [Informatikstudium Fachhochschulen, 1993]: Das Studium der Informatik an Fachhochschulen. In *Troitzsch, Klaus G. (Hrsg.): Informatik als Schlüssel zur Qualifikation*. Springer Verlag, 1993, Informatik Aktuell, 55–63
- Gesellschaft für Informatik* [Informatik-Spektrum 1985, 1985]: Ausbildung von Diplominformatikern an wissenschaftlichen Hochschulen – Empfehlungen der Gesellschaft für Informatik. Informatik-Spektrum, Juni 1985, 164–165
- Gesellschaft für Informatik* [Aufgaben und Ziele der Informatik, 1987]: Aufgaben und Ziele der Informatik. Arbeitspapier der Gesellschaft für Informatik, Oktober 1987, Bonn
- Karmasin, Matthias/Ribing, Rainer* [Gestaltung wissenschaftlicher Arbeiten, 2009]: Die Gestaltung wissenschaftlicher Arbeiten: Ein Leitfaden für Seminararbeiten, Bachelor-, Master- und Magisterarbeiten, Diplomarbeiten und Dissertationen. 4. Auflage. UTB, 2009, ISBN 978-3-8252-2774-6
- Kohm, Markus/Morawski, Jens-Uwe* [KOMA-Script, 2009]: KOMA-Script: Eine Sammlung von Klassen und Paketen für L^AT_EX 2_ε. 3. Auflage. Berlin: Lehmanns Media, 2009, ISBN 978-3-86541-291-1
- Kornmeier, Martin* [Wissenschaftstheorie und wissenschaftliches Arbeiten, 2007]: Wissenschaftstheorie und wissenschaftliches Arbeiten: Eine Einführung für Wirtschaftswissenschaftler. Heidelberg: Physica-Verlag, 2007, ISBN 978-3-7908-1918-2
- Langenheder, Werner/Müller, Günter/Schinzel, Britta (Hrsg.)* [Informatik – cui bono?, 1992]: Informatik cui bono? GI-FB 8 Fachtagung, Freiburg, 23.-26. September 1992. Springer, 1992, Informatik Aktuell, ISBN 3-540-55957-4

- Parnas, David L.* [Education, 1990]: Education for Computer Professionals. IEEE Computer, 23 1990, Nr. 1, 17–22
- Rossig, Wolfram E./Prätsch, Joachim* [Wissenschaftliches Arbeiten, 2006]: Wissenschaftliche Arbeiten – Leitfaden für Haus- und Seminararbeiten, Bachelor- und Masterthesis, Diplom- und Magisterarbeiten, Dissertationen. 6. Auflage. Weyhe: TEAMDRUCK, 2006, ISBN-10: 3-9810012-2-2
- Schlosser, Joachim* [Wissenschaftliche Arbeiten mit L^AT_EX, 2009]: Wissenschaftliche Arbeiten schreiben mit L^AT_EX: Leitfaden für Einsteiger. 3. Auflage. Heidelberg: mitp, 2009, ISBN 978-3-8266-5892-1
- Theisen, Manuel René* [Wissenschaftliches Arbeiten, 2009]: Wissenschaftliches Arbeiten: Technik – Methodik – Form. Vahlen, 2009, Limitierte Jubiläumsausgabe, ISBN 978-3-8006-3669-3

Ehrenwörtliche Erklärung

„Ich erkläre hiermit ehrenwörtlich:

1. dass ich meine Seminararbeit mit dem Thema

Star Greg
Das Unternehmensplanspiel

ohne fremde Hilfe angefertigt habe;

2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Seminararbeit gekennzeichnet habe;
3. dass ich meine Seminararbeit bei keiner anderen Prüfung vorgelegt habe;
4. dass die eingereichte elektronische Fassung exakt mit der eingereichten schriftlichen Fassung übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.“

Ort, Datum

Unterschrift