

Hotelmanager 4.0

Anwendung der linearen Optimierung

Autoren:

Dominique Lebert

Vitaliy Davats

Manuel Falkenstein

Vorzulegen bei: Prof. Dr. Grütz

Vorzulegen am: 24. Januar 2014

Inhalt

1.	Einleitung	2
1.1	Projektziele	2
1.2	Rahmenbedingungen	2
1.3	Projektorganisation	2
1.3.1	Projektverantwortlicher / Auftraggeber:.....	2
1.3.2	Projektmitglieder.....	3
2.	Projektphase.....	3
2.1	Planungsphase	3
2.2	Abprache mit dem Auftraggeber über die Projektziele.....	3
2.3	Entwicklungsphase	3
2.3.1	Verwendete Technologien.....	3
2.3.2	Klassendiagramm	4
2.3.3	Konfiguration für Hotelmanager 4.0	4
2.3.4	Neue Kalender Funktion	5
2.3.5	Saison Einstellung.....	6
2.3.6	Ausgabe von LP Restriktionen und LP Lösung.....	8
3.	Erweiterungen des Hotelmanager	11
4.	Fazit	12
5.	Abbildungsverzeichnis.....	13
6.	Literaturverzeichnis.....	13

1. Einleitung

Im Folgenden wird die Entwicklung des „Hotelmanager 4.0“ dokumentiert, welcher im Rahmen der Lehrveranstaltung Anwendungen der linearen Optimierung an der HTWG Konstanz erfolgte.

1.1 Projektziele

In Absprache mit Herr Prof. Dr. Grütz wurde vereinbart, dass folgende Ziele bis zum 24. Januar 2014 im Hotelmanager implementiert werden:

- Erarbeitung einer Funktion zur Bestimmung von Haupt- und Nebensaison. Diese soll flexibel einstellbar sein.
- Das Anreisedatum soll über eine neue Kalenderfunktion einfacher und übersichtlicher gewählt werden können.
- Ausgabe der LP Restriktionen und LP Lösung soll intern im Programm erfolgen.

Um die oben genannten Punkte implementieren zu können, mussten wir die Lesbarkeit, Verständlichkeit, Wartbarkeit und Erweiterbarkeit des Quellcode verbessern. Das Refactoring dient dazu, um zukünftige Bearbeitung des Hotelmanager 4.0 zu vereinfachen.

1.2 Rahmenbedingungen

- Verbesserung des bestehenden Programm „HotelmanagerNeu“
- Entwicklung der Programmfunktionen in Java 7
- Entwicklungsumgebung ist Eclipse Kepler
- Fertigstellung bis zum 24. Januar 2014

1.3 Projektorganisation

1.3.1 Projektverantwortlicher / Auftraggeber:

Als Auftragsgeber und Abnahmeverantwortlicher wird Herr Professor Dr. Grütz der HTWG Konstanz benannt.

1.3.2 Projektmitglieder

Name	Matrikelnummer	Studiengang
Davats, Vitaliy	279934	Wirtschaftsinformatik B.Sc.
Lebert, Dominique	284623	Wirtschaftsinformatik B.Sc.
Falkenstein, Manuel	284381	Wirtschaftsinformatik B.Sc.

2. Projektphase

2.1 Planungsphase

Zu Beginn der Planungsphase haben wir das Programm „Hotelmanager 3.0“ gründlich getestet und den entsprechenden LP-Ansatz analysiert. Des Weiteren haben wir uns die Dokumentationen der Hotelmanager 1, 2 und 3 der Vorgänger durchgelesen und analysiert. Dabei haben wir die oben genannten Ziele als mögliche Weiterentwicklungen festgelegt. Danach wurden die Java-Programme geöffnet und analysiert und ein Code Refactoring begonnen. Nachdem das Programm verständlich war, wurden sämtliche Verbesserungsmöglichkeiten aufgeschrieben und priorisiert. Dabei waren die Projektziele die höchstpriorisierten. Abschließend zur Planungsphase wurde im Team abgesprochen, wie jedes dieser Ziele erreicht werden soll.

2.2 Absprache mit dem Auftraggeber über die Projektziele

Am 18.12.2013 hat das Projektteam in Absprache mit Herr Prof. Dr. Grütz sich auf die o.g. Projektziele geeinigt.

2.3 Entwicklungsphase

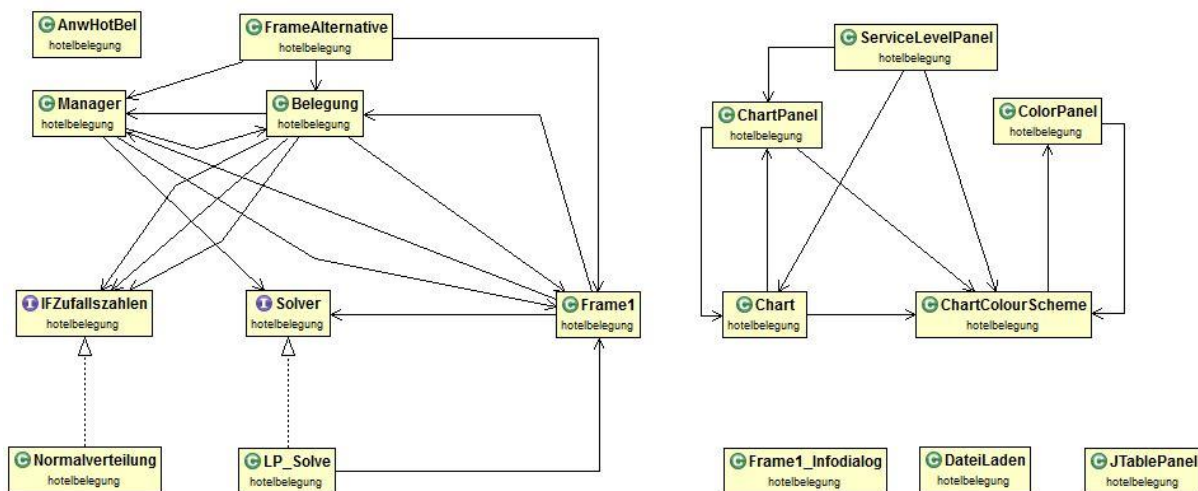
Nachfolgend sollen die einzelnen Technologien und Methoden erläutert werden, die in der Entwicklung verwendet wurden.

2.3.1 Verwendete Technologien

- Java 7 als Programmiersprache
- Eclipse Kepler als Entwicklungsumgebung
- JavaSwing für die Gestaltung der GUI

2.3.2 Klassendiagramm

Das Klassendiagramm soll einen Überblick über die verwendeten Klassen in unserem Programm geben. Dabei sind die von uns verwendeten Variablen und Methoden nicht abgebildet, da diese das Klassendiagramm extrem vergrößern würde.



2.3.3 Konfiguration für Hotelmanager 4.0

Um den Hotelmanager verwenden zu können, benötigt das Programm einen Solver. In der Programmierung wurde von uns der Solver „LP Solve“ verwendet. Der Pfad für den Solver haben wir in der Klasse „Frame1“ mit der Variable „solverPfad“ deklariert. Der von uns deklarierte Default-Solverpfad lautet: „C:\Methodendatenbank\Solver\LP_Solver\Exec“. Sollte sich der Default Pfad in Zukunft ändern bzw. ein anderer Solver verwendet werden, so muss dieser Pfad im Programmcode berücksichtigt werden. Jedoch kann der Pfad auch im Programm verändert werden, unter dem Reiter „Einstellungen“.

Damit das Programm die LP Restriktionen und LP Lösung laden kann, muss ein „Temp“-Verzeichnis auf dem C-Laufwerk existieren. Möchte man ein anderes Arbeitsverzeichnis benutzen, so muss entweder im Programmcode (Klasse „Frame1“ und Variable „arbeitsVerzeichnis“) oder auf der Programmoberfläche (Reiter „Einstellungen“) der Pfad geändert werden.

2.3.4 Neue Kalender Funktion

Wird das Programm gestartet, so sieht man als Startpunkt den Reiter „Reservierung“. Die Auswahl des Anreisedatums wurde in den Versionen 1-3 über ein Dropdown-Menü realisiert. Da wir dies als unübersichtlich und nicht komfortabel ansahen, haben wir die Auswahl für das Anreisedatum etwas ansprechend und übersichtlicher gestaltet. Das Dropdown Menü für Tag, Monat, Jahr haben wir durch einen Monatskalender ersetzt. Durch ein einfaches anklicken einer Zahl kann ein gewünschter Tag ausgewählt werden. In diesem Kalender wird das aktuelle Datum angezeigt und bei einer fehlerhaften Eingabe, wird darauf hingewiesen, dass man nicht in der Vergangenheit buchen kann.

Reservierungswunsch:

Januar

2014

	Mo	Di	Mi	Do	Fr	Sa	So
01			1	2	3	4	5
02	6	7	8	9	10	11	12
03	13	14	15	16	17	18	19
04	20	21	22	23	24	25	26
05	27	28	29	30	31		

Aufenthaltsdauer: Anzahl der Tage

Zimmerkategorie: Zimmerkategorie

Anfrage starten

Abbildung 1: Auswahl von Anreisedatum, Aufenthaltsdauer und Zimmerkategorie

2.3.5 Saison Einstellung

Für die Einstellung der Parameter haben wir eine Erweiterung im Einstellungsfenster programmiert. Es besteht im Gegensatz zur vorherigen Version die Möglichkeit, dass ein Hotel in verschiedene Zeiten (Monate) im Jahr als Saison bzw. Hochsaison einstellen kann. Der flexible saisonale Einfluss wurde von uns damit begründet, dass ein Hotel nicht über das ganze Jahr eine Hochsaison hat (eventuell auch in Teilzeiten geschlossen) und somit über das Jahr verteilt eine unterschiedliche Preisentwicklung hat.

Als Beispiel kann man folgende Szenarien nennen. An einem Strandgebieten ist die Hochsaison von Juni bis August und in einem Ski-Gebiet ist die Hochsaison von Dezember bis Februar. In den anderen Jahreszeiten der jeweiligen Gebiete besteht eine andere Saison. Darüber hinaus kann es sein, dass ein Hotel nicht über das ganze Jahr geöffnet ist.

Für die Programmierung der Saison und Hochsaison haben wir fehlende Variable im Programmcode ergänzt, da die vorherige Version keine Saison definiert hat. Hierfür haben wir die Hochsaison, Nebensaison und die Zeit außerhalb der Saison gesondert betrachtet. Daraufhin haben wir mittels JavaSwing die Oberfläche der Einstellungsparameter programmiert. Dadurch haben wir die Möglichkeit, über Reiter von der Saison zur Hochsaison zu wechseln. In den jeweiligen Reiter kann der Nutzer die Zeiten definieren von wann die Hochsaison bzw. die Saison ist. Ebenso sollen die Werte für Zimmeranzahl, Zimmerpreis, Deckungsbeitrag, Zimmerbelegung und Spontanbuchungen (Hochsaison / Saison) definiert werden. Wird von dem Benutzer kein Parameter festgelegt, so sind standardmäßig Default Werte eingetragen.

In Abbildung 2 sind die Default-Parameter zu sehen. Dabei ist die Saison von Mai bis Oktober definiert. Wird von dem Nutzer des Programms ein Anreisedatum außerhalb dieses Zeitraumes angefragt, so wird eine Fehlermeldung erscheinen. In Abbildung 3 wird die Hochsaison definiert mit veränderten Parameterwerten.

Programmierung:

Die Programmierung der Saison und Hauptsaison wird hier nicht aufgeführt, da der Programmcode für die Darstellung zu groß ist. Diesbezüglich haben wir eine Berichtdatei mitgeliefert, in der der Programmcode zu sehen ist.

Saison Hochsaison

Saison festlegen: von Mai bis (nicht inkl.) Oktober

Parameter

	Kategorie 1	Kategorie 2	Kategorie 3
Zimmeranzahl	50	150	125
Zimmerpreis	100	50	40
Deckungsbeitrag	45	20	12
Zimmerbelegung %	30	40	35
Spontanbuchungen %	20	15	5

Abbildung 2: Parametereinstellung für die Saison

Saison Hochsaison

Saison festlegen: von Juli bis (nicht inkl.) September

Parameter

	Kategorie 1	Kategorie 2	Kategorie 3
Zimmeranzahl	150	300	250
Zimmerpreis	200	100	80
Deckungsbeitrag	90	40	25
Zimmerbelegung %	60	85	70
Spontanbuchungen %	30	20	10

Abbildung 3: Parametereinstellungen für die Hauptsaison

2.3.6 Ausgabe von LP Restriktionen und LP Lösung

In der Hotelmanager Version „Hotelmanager 3“ waren die Ausgaben der LP Restriktionen und die LP Lösung jeweils über eine externe Datei einsehbar die über einen Editor geöffnet wurde. Diese wurde über einen Button in dem Register LP-Solve Ausgabe aktiviert. Jedoch war die Darstellung der LP Restriktionen und die LP Lösung unsauber gelöst, so dass man die Lösung nur schwer lesen, geschweige denn verstehen konnte.

Unsere Überlegung war es, dass man die Ausgabe der LP-Solve Lösung in das Hotelmanager Programm direkt integriert. Motivation dafür war folgende:

- LP-Solve Lösung direkt einsehbar ohne Umwege
- Zeitersparnis, da kein Datei aufruf
- Bessere Darstellung der LP Restriktionen und LP Lösung (lesbar)

Im folgende wollen wir kurz erklären, wie die neue Ansicht von LP Restriktionen und LP Lösung funktioniert und auf die Programmierung der LP-Solve Lösung anschließend eingehen.

Die jeweiligen Ansichten in dem Programm „Hotelmanager“ sind über Reiter einsehbar und können dort auch konfiguriert werden. Für unsere LP-Solve Restriktionen und LP Lösung haben wir jeweils einen weiteren Reiter hinzugefügt. Diese werden erst dann sichtbar, wenn für die Buchung ein Anreisedatum mit Zimmerkategorie und Aufenthaltsdauer ausgewählt wurde. Ist ein Reisedatum, Zimmerkategorie und Aufenthaltsdauer ausgewählt, so wird die jeweilige Lösung in einen Editor eingefügt, welcher in dem Reiter integriert ist. Es besteht dazu die Möglichkeit, dass bei großen Restriktionen der Editor in allen Seiten gescrollt werden kann.

Die Programmierung:

Mit TextArea wird ein neues Panel (Reiter) erstellt, welches ein Objekt von Panel() ist. Dies ist der Reiter für die LP Restriktionen. Das gleiche machen wir ebenfalls mit TextArea2. Dadurch haben wir ein Panel für LP Lösung erstellt. Mit „setLayout()“ definieren wir jeweils ein neues Layout. Als weiterer Schritt wir mit dem JTextArea und JTextArea2 ein Textfeld für das jeweilige Panel erstellt. Dazu wird mit JScrollPane Scroll-Funktion erstellt, damit bei großen Texten nach unten, oben, rechts und links gescrollt werden kann. Abschließend wird dem jeweiligen Textfeld die Scroll-Funktion mitgegeben.

```

802
803         TextArea = new JPanel();
804         TextArea2 = new JPanel();
805         TextArea.setLayout(new BorderLayout(0, 0));
806         TextArea2.setLayout(new BorderLayout(0, 0));
807         jTextArea = new JTextArea(150, 150);
808         jTextArea2 = new JTextArea(150, 150);
809         JScrollPane scrollPane = new JScrollPane(jTextArea);
810         JScrollPane scrollPane2 = new JScrollPane(jTextArea2);
811         TextArea.add(scrollPane);
812         TextArea2.add(scrollPane2);
813

```

Abbildung 4: Objekte anlegen

Um jTextArea und jTextArea2 außerhalb einer Methode verwenden zu können, deklarieren wir sie als „public“-Variablen. Da TextArea und TextArea2 nur innerhalb einer Methode verwendet werden und auch nur dort verwendbar sein sollen, haben wir sie als „private“ deklariert.

```

32     private JPanel TextArea;
33     private JPanel TextArea2;
34     public JTextArea jTextArea;
35     public JTextArea jTextArea2;

```

Abbildung 5: Variablen deklarieren

Als weiteres definieren wir zwei „String“-Variablen, die jeweils den Pfad der Datei tragen, welche wir auslesen möchten. Die Restriktionen befinden sich in der „hotel.lp“-Datei und die LP Lösung in der „hotel.out“-Datei. Sollte in Zukunft eine andere Datei ausgelesen werden, so muss dieser Pfad dem String angepasst werden. Im nächsten Schritt erzeugen wir zwei Objekte „ladeDateiLP“ und „ladeDateiOUT“ aus der Klasse „DateiLaden“. Nun wird ein String „auslesenLP“ definiert, welches aus der Klasse die Funktion „Laden()“ verwendet und diese unseren Pfad beinhaltet. Das gleiche machen wir mit „auslesenOUT“. TextArea und TextArea2 werden mittels RegisterJTabbedPane.add() dem Frame hinzugefügt.

```

1004     String hotelLP = "C:\\Temp\\hotel.lp";
1005     String hotelOUT = "C:\\Temp\\hotel.out";
1006     DateiLaden ladeDateiLP = new DateiLaden();
1007     DateiLaden ladeDateiOUT = new DateiLaden();
1008     String auslesenLP = ladeDateiLP.Laden(hotelLP);
1009     String auslesenOUT = ladeDateiOUT.Laden(hotelOUT);
1010
1011     RegisterJTabbedPane.addTab("LP-Solve: LP Restriktionen", null, TextArea, null);
1012     RegisterJTabbedPane.addTab("LP-Solve: LP Lösung", null, TextArea2, null);
1013

```

Abbildung 6: Laden und anzeigen der Objekte

Im Folgenden werden dem Textfeld „jTextArea“ der String „auslesenLP“ übergeben, welcher den LP-Ansatz beinhaltet. Dieser String (LP-Ansatz) wird nun in unserer „TextArea“ ausgegeben. Das gleiche wird wieder mit „jTextArea2“ gemacht, bloß dass der String „auslesenOUT“ übergeben wird.

```
1025      jTextArea.setText(auslesenLP);
1026      jTextArea2.setText(auslesenOUT);
```

Abbildung 7: Ausgabe von auslesenLP und auslesenOUT

Um sich ein Bild machen zu können, wie die Programmierung in unseren Programm „Hotelmanager“ auswirkt, wollen wir dies hier zeigen. Ist für die Buchung noch keine Reisedatum, Zimmerkategorie und Aufenthaltsdauer ausgewählt, so werden nur die Standardreiter dargestellt.



Abbildung 8: Sichtbare Reiter bei Programmstart

Anders sieht es aus, wenn die jeweiligen Parameter bestimmt worden sind und der Button „Anfrage Starten“ aktiviert worden ist.

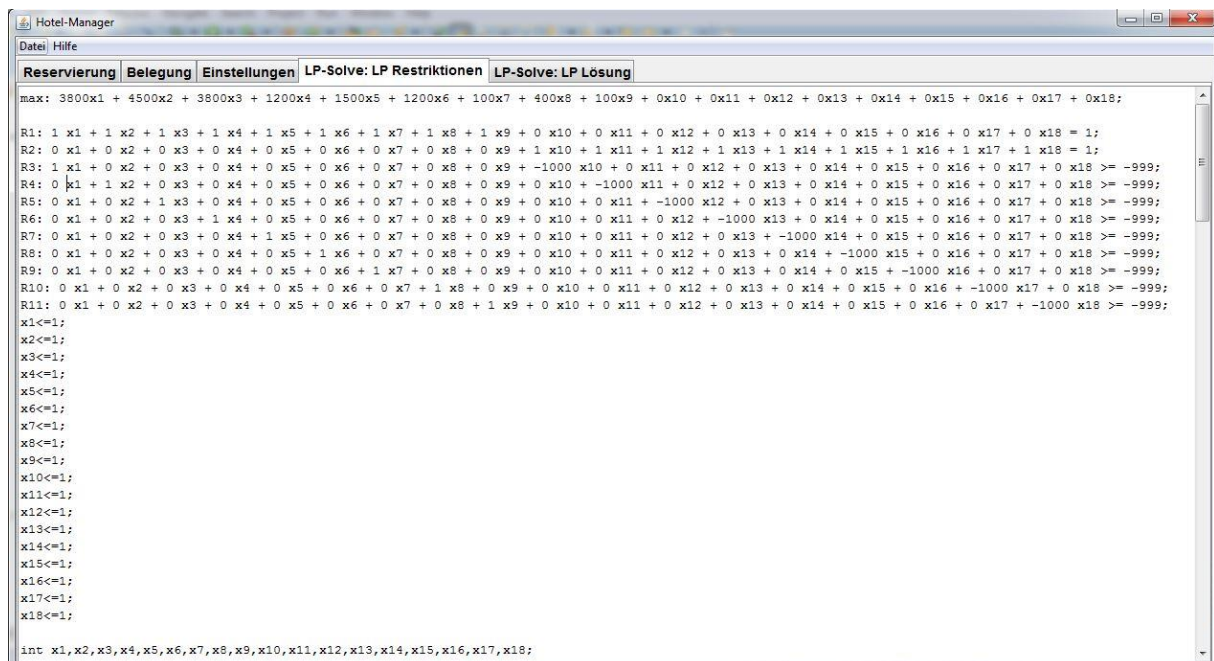


Abbildung 9: Reiter LP Restriktionen

Der Reiter „LP-Solve: LP Lösung“ wird z.B. wie folgendes Dargestellt:

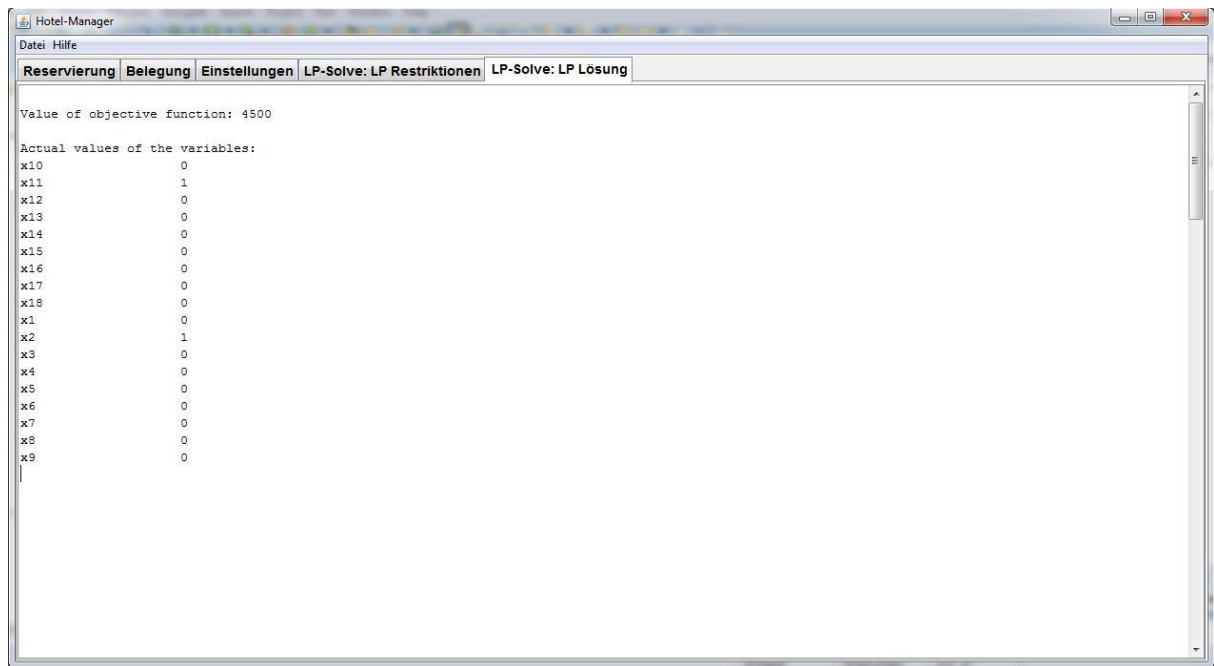


Abbildung 10: Reiter LP Lösung

Die Variablen werden untereinander aufgelistet, damit die Lösung klar zu erkennen ist.

3. Erweiterungen des Hotelmanager

Funktionen	Beschreibung
Zimmerbelegung	Als Erweiterung könnte man sich eine persistente Zimmerbelegung überlegen, die die gebuchten Zimmer speichern und
Personenanzahl	Für die Buchung eines Zimmers könnte man eine Angabe der Person machen, um die Person eindeutig mit dem Zimmer zu identifizieren.
IBM CPLEX	LP-Solve ist für eine hohe Anzahl an Variablen schlecht geeignet, da die Berechnung zu lange dauert. Daher sollte der CPLEX Optimizer von IBM eingebunden werden.

4. Fazit

Bei der Übernahme des Projekts hatten wir früh bemerkt, dass die Einarbeitung in den Quellcode der Vorgängergruppe sehr aufwendig wird. Dabei war es schwer zu verstehen, welche Methode für was verwendet wird, da der Quellcode kaum dokumentiert wurde. Jedoch konnte wir durch intensives einarbeiten den Code verstehen und anwenden. Die neuen Funktionen, die von uns als Ziel definiert wurden, haben wir erfolgreich umgesetzt. Der Drop-Down Kalender wurde durch eine neue Kalenderfunktion ersetzt, die Parametereinstellungen wurden um eine Saison / Hauptsaison erweitert und die Ausgaben der LP Restriktionen / LP Lösung sind nun intern im Programm lesbar bzw. auch verständlich angeordnet. Des Weiteren konnten wir durch die Erarbeitung der graphischen Oberfläche weitere Erfahrung in der Programmierung mit JavaSwing gewinnen. Das Programm könnte durch weitere Funktionen (siehe Kapitel 3), die in einem Teamprojekt erarbeitet werde, verbessert werden.

5. Abbildungsverzeichnis

Abbildung 1: Auswahl von Anreisedatum, Aufenthaltsdauer und Zimmerkategorie	5
Abbildung 2: Parametereinstellung für die Saison	7
Abbildung 3: Parametereinstellungen für die Hauptsaison	7
Abbildung 4: Objekte anlegen	9
Abbildung 5: Variablen deklarieren.....	9
Abbildung 6: Laden und anzeigen der Objekte	9
Abbildung 7: Ausgabe von auslesenLP und auslesenOUT	10
Abbildung 8: Sichtbare Reiter bei Programmstart.....	10
Abbildung 9: Reiter LP Restriktionen	10
Abbildung 10: Reiter LP Lösung.....	11

6. Literaturverzeichnis

Hotelmanager 1 [Autoren: Volker Wohlleber, Oliver Schraag, Florian Raiber]

Hotelmanager2 [Autoren: Kilian Thiel, Oliver Bühler]

HotelmanagerNeu_Doku [Autoren: Michael Gehring, Fabian Griebhaber]