

Philipps-Universität Marburg

Sommersemester 2005
Informatikpraktikum im Grundstudium



Entwicklung einer Literaturdatenbank

Entwurfsdokument
Gruppe: Anwendungslogik (AL)

Betreuer: Prof. Dr. Bernhard Seeger

Autoren: Manuel Haim, Marc Koch, Jakub Lidke, Sami Madani, Nan Meng, Stefan Sauer, Doru Vormicescu, Evgeni Zinoviev

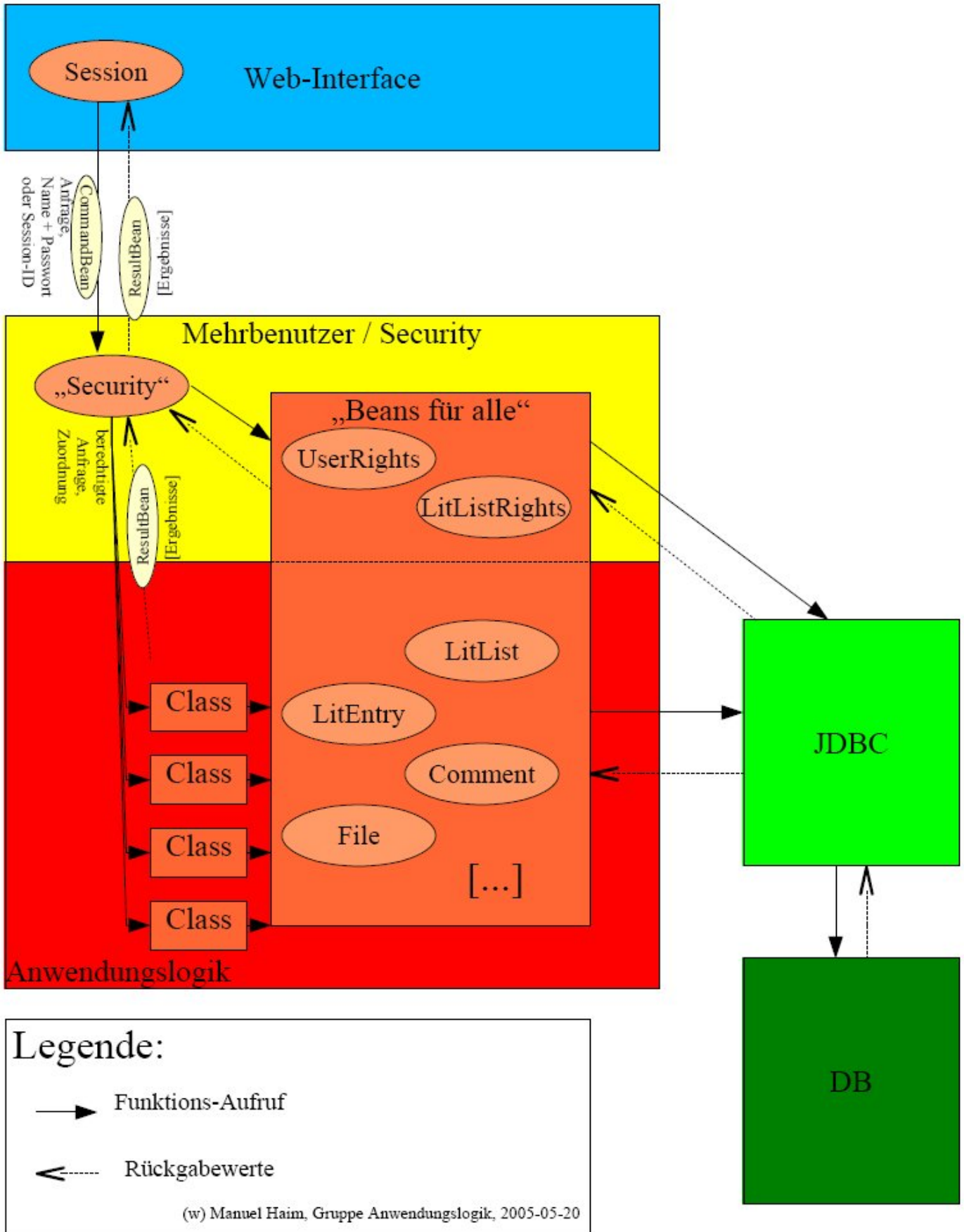
Abgabe: 27.05.2005

Inhalt

1. Ablaufschema	3
2. Bean-Konzept	5
3. Klassendiagramm	6
4. Allgemeines Schnittstellenkonzept	7
4.1. CommandBeans (abstract)	7
4.1.1. CmdSaveBean	7
4.1.2. CmdCreateBean	7
4.1.3. CmdLoadBean	8
4.1.4. CmdDeleteBean	8
4.2. DataBean (abstract)	8
4.2.1. FileBean	8
4.2.2. CommentBean	8
4.2.3. LitEntryBean	8
4.2.4. LitListBean	8
4.2.5. XmlBean	9
4.2.6. BibTexBean	9
4.2.7. TextBean	9
4.2.8. SearchBean	9
4.2.9. ArrayBean	9
4.3. ResultBean	9
4.4. Statische Klassen	9
4.4.1. LitList	9
4.4.2. Comment	10
4.4.3. LitEntry	10
4.4.4. Transfer	10
4.4.5. File	11
4.4.6. Interface Constants	11
5. Weiterführende Links der AL	12
6. Haftungsausschluss	12

1. Ablaufschema

Dieses Ablaufdiagramm verdeutlicht die Zusammenarbeit der einzelnen Arbeitsgruppen.



Insbesondere wird hier die Arbeitsweise mit Hilfe der Beans veranschaulicht. Der Datenaustausch findet mit Hilfe der von der Anwendungslogik erstellten Beans statt. Genauere Ausführungen gibt es in den folgenden Kapiteln.

Aus dem Ablaufschema ist zu erkennen, dass die Multiuser-Gruppe der Anwendungslogik vorgezogen ist. Sie fängt alle Anfragen seitens des Webinterfaces auf und kontrolliert diese auf ihre Zulässigkeit hin (Rechtmanagement) und leitet die Anfragen erst dann an die AL weiter. Die Aufgabe der Anwendungslogik ist es also primär, die Algorithmen zur Verfügung zu stellen, die die Beans brauchen, um als Transportbehälter für die Daten zur Verfügung stehen zu können.

2. Bean-Konzept

Die einzelnen Beans dienen, wie im letzten Kapitel angedeutet, zum Datentransport oder der Aufgabenübertragung zwischen den einzelnen Projektteilen (z.B. WI → AL).

Sowohl für den Datentransport als auch für die Aufgabenübermittlung gibt es verschiedene Beans mit jeweils fest spezifizierten Aufgabenbereichen.

So kann die abstrakte CommandBean zu benötigten Beans erweitert werden, mit deren Hilfe effizient die Aufgaben abgewickelt werden können.

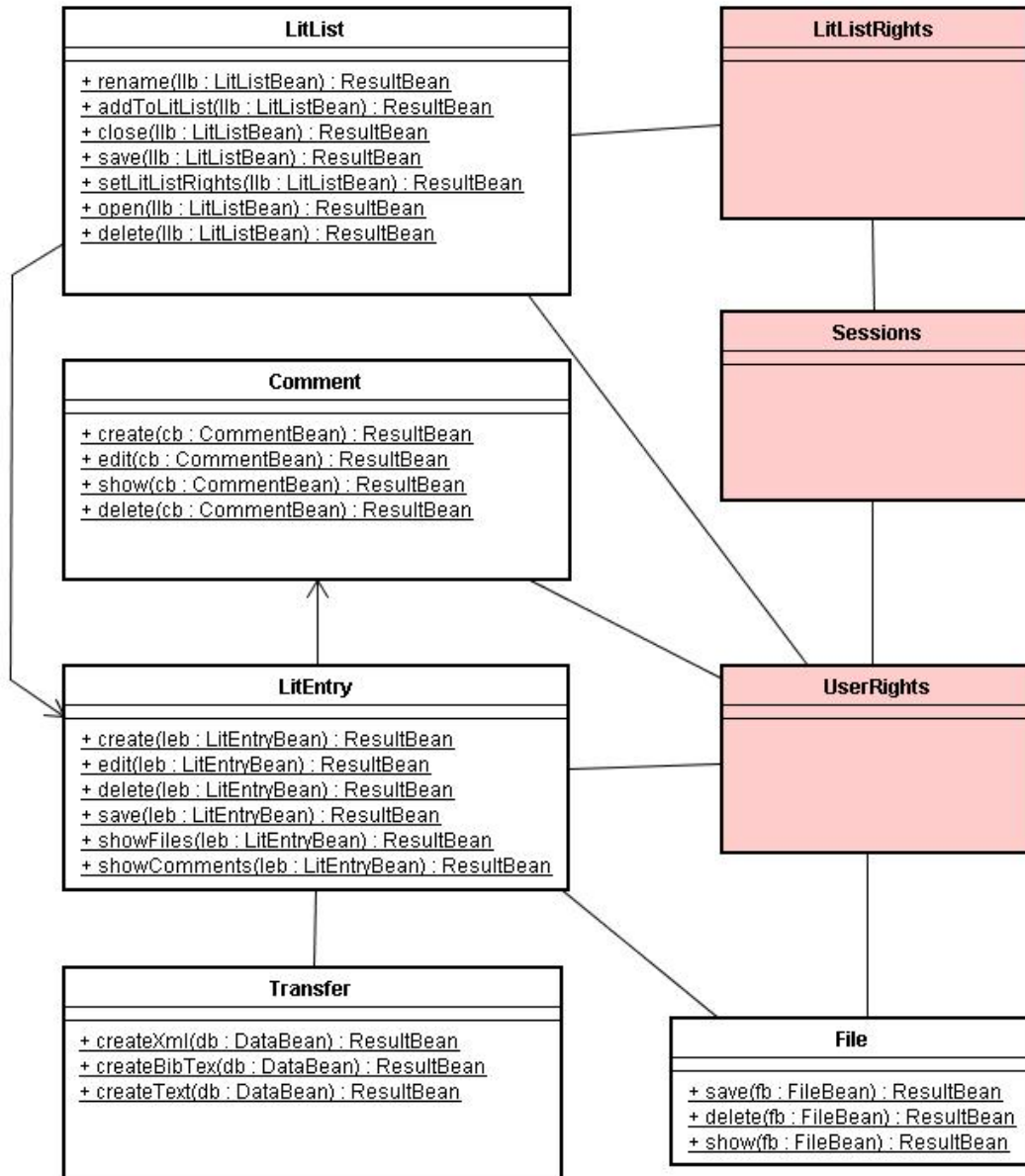
Mit Hilfe der DataBeans werden die Daten zu den entsprechenden CommandBeans mit übergeben. Mit der Kombination der DataBean und der CommandBean kann dann von der AL und den anderen Gruppen der gewünschte Anwendungsfall identifiziert und ausgeführt werden.

In den CommandBeans werden die DataBeans als Parameter übergeben. Zusätzlich enthält die CommandBean die SessionBean, welche die Nutzerdaten des Users, der die Anfrage tätigt, enthält. Diese Nutzerdaten werden zunächst bei der MU ausgewertet. Das Ergebnis dieser Anfrage wird in Form einer ResultBean zurückgegeben, die je nach Ausgang der Anfrage eine DataBean oder eine Exception enthält



Die Typbeschreibung der einzelnen Beans erfolgt im folgenden Kapitel.

3. Klassendiagramm



Constants	
~ARTICLE : int = 101	
~BOOK : int = 102	
~BOOKLET : int = 103	
~COLLECTION : int = 104	
~CONFERENCE : int = 105	
~INBOOK : int = 106	
~INCOLLECTION : int = 107	
~INPROCEEDINGS : int = 108	
~MANUAL : int = 109	
~MASTERSTHESIS : int = 110	
~MISC : int = 111	
~PATENT : int = 112	
~PHDTHESIS : int = 113	
~PROCEEDINGS : int = 114	
~TECHREPORT : int = 115	
~UNPUBLISHED : int = 116	
~ABSTRACT : int = 1	
~ADDRESS : int = 2	
~ANNOTE : int = 3	
~AUTHOR : int = 4	
~BOOKTITLE : int = 5	
~CHAPTER : int = 6	
~CONTENTS : int = 7	
~COPYRIGHT : int = 8	
~CROSSREF : int = 9	
~EDITION : int = 10	
~EDITOR : int = 11	
~HOWPUBLISHED : int = 12	
~INSTITUTION : int = 13	
~ISBN : int = 14	
~ISSN : int = 15	
~JOURNAL : int = 16	
~KEY : int = 17	
~KEYWORDS : int = 18	
~LANGUAGE : int = 19	
~MONTH : int = 20	
~NOTE : int = 21	
~NUMBER : int = 22	
~ORGANIZATION : int = 23	
~PAGES : int = 24	
~PUBLISHER : int = 25	
~SCHOOL : int = 26	
~SERIES : int = 27	
~SHORT_BOOKTITLE : int = 28	
~SHORT_JOURNAL : int = 29	
~SHORT_ORGANIZATION : int = 30	
~SHORT_SERIES : int = 31	
~TITLE : int = 32	
~TYPE : int = 33	
~URL : int = 34	
~VOLUME : int = 35	
~YEAR : int = 36	

4. Allgemeines Schnittstellenkonzept

Sämtliche Daten werden in Beans verpackt, das hat den Vorteil, dass die Daten von der Funktionalität getrennt sind. Dies ermöglicht einem einfachen Datenaustausch mit den anderen Programm-Modulen. Im speziellen haben wir eine Verbindung zum WI mittels CommandBeans und ResultBeans und zur JDBC DataBeans.

4.1. CommandBeans (abstract)

Eine CommandBean muss eine solche Form haben:

```
public abstract class CommandBean {
    public CmdBean cmdBean;
    public SessionBean sessionBean;
    public CommandBean(CmdBean cmdBean, SessionBean sessionBean) {
        this.cmdBean = cmdBean;
        this.sessionBean = sessionBean;
    }
}
```

Es können daraus folgende Beans abgeleitet werden:

4.1.1. CmdSaveBean

Diese ist für die Speicherung der DataBeans zuständig

```
public class CmdSaveBean extends CommandBean {
    public CmdSaveBean(DataBean dataBean1, DataBean dataBean2, DataBean
dataBean3) {
        this.dataBean1 = dataBean1;
        this.dataBean2 = dataBean2;
        this.dataBean3 = dataBean3;
    }
}
```

4.1.2. CmdCreateBean

Diese ist für das erzeugen der DataBeans zuständig

```
public class CmdSaveBean extends CommandBean {
    public CmdSaveBean(DataBean dataBean1, DataBean dataBean2, DataBean
dataBean3) {
        this.dataBean1 = dataBean1;
        this.dataBean2 = dataBean2;
        this.dataBean3 = dataBean3;
    }
}
```

4.1.3. CmdLoadBean

Diese ist für das Laden der DataBeans zuständig

```
public class CmdSaveBean extends CommandBean {
    public CmdSaveBean(DataBean dataBean1, DataBean dataBean2, DataBean
dataBean3) {
        this.dataBean1 = dataBean1;
        this.dataBean2 = dataBean2;
        this.dataBean3 = dataBean3;
    }
}
```

4.1.4. CmdDeleteBean

Diese ist für das Löschen der DataBeans zuständig

```
public class CmdSaveBean extends CommandBean {
    public CmdSaveBean(DataBean dataBean1, DataBean dataBean2, DataBean
dataBean3) {
        this.dataBean1 = dataBean1;
        this.dataBean2 = dataBean2;
        this.dataBean3 = dataBean3;
    }
}
```

4.2. DataBean (abstract)

```
public abstract class DataBean {
}
```

Es können folgende Beans abgeleitet werden:

4.2.1. FileBean

Beinhaltet den Pfad einer Datei, enthaltene Daten:

```
String fileKey
String fileName
String path
```

4.2.2. CommentBean

Beinhaltet einen Kommentar für einen LitEntry, enthaltene Daten:

```
int commentKey
int userKeyfile
String comment
```

4.2.3. LitEntryBean

Beinhaltet den Literatureintrag einer LitList, enthaltene Daten:

```
int litEntryKey
String entryName
int userKey
```

4.2.4. LitListBean

Beinhaltet einen Array mit LitEntryBeans, enthaltene Daten:

```
int litListKey
String listName
```



```
int userKey  
litEntryBean litEntryKey[]
```

4.2.5. XmlBean

Beinhaltet einen StringArray im XML-Format, enthaltene Daten:
String text[]

4.2.6. BibTexBean

Beinhaltet einen StringArray im BitTex-Format, enthaltene Daten:
String text[]

4.2.7. TextBean

Beinhaltet einen StringArray im Txt-Format, enthaltene Daten
String text[]

4.2.8. SearchBean

Beinhaltet ein Suchstring und eine Suchtypvariable, enthaltene Daten:
String searchString

4.2.9. ArrayBean

Beinhaltet einen Array von DataBeans (optional), enthaltene Daten:
DataBean beanArray[]

4.3. ResultBean

Die ResultBean liefert das Ergebnis einer Operation an das WI zurück. Enthalten sein können Exception (boolean) und DataBean.

ResultBean, Daten: DataBean bean;

4.4. Statische Klassen

Die folgenden Klassen enthalten allesamt keine Datenfelder, sondern nur statische Methoden. Diese werden von den Bohnen aufgerufen.

Im Einzelnen sind dies:

4.4.1. LitList

Die Klasse LitList enthält alle Funktionen, die auf einer Literaturliste ausgeführt werden können.

- Liste umbenennen
- Liste löschen
- Liste erzeugen
- Datensatz zur LL hinzufügen
- Die Rechte für Benutzer setzen
- Öffnen ein LL
- Schließen LL
- Speichern

LitList	.rename(LitListBean llb)	returns ResultBean
	.addEntry(LitListBean llb, LitEntryBean leb)	returns ResultBean
	.close(LitListBean llb)	returns ResultBean
	.save(LitListBean llb)	returns ResultBean
	.setRights(LitListBean llb)	returns ResultBean
	.open(LitListBean llb)	returns ResultBean
	.delete(LitListBean llb)	returns ResultBean
	.rename(LitListBean llb)	returns ResultBean
	.create()	returns ResultBean

4.4.2. Comment

Kommentare können von Benutzern zu Datensätzen erstellt werden. Die Klasse kann folgende Operationen durchführen:

- Kommentare erzeugen
- Kommentare editieren
- Kommentare anzeigen
- Kommentare löschen

Comment	.create(CommentBean cb)	returns ResultBean
	.edit(CommentBean cb, LitEntryBean leb)	returns ResultBean
	.show(CommentBean cb)	returns ResultBean
	.delete(CommentBean cb)	returns ResultBean

4.4.3. LitEntry

Ein LitEntry enthält alle Informationen die zu einem Buch gespeichert werden sollen. Die Klasse besitzt folgende Methoden:

- Datensatz anlegen
- Datensatz editieren
- Datensatz speichern
- Datensatz löschen
- Datei zum Datensatz anzeigen
- Kommentare zum Datensatz anzeigen

LitEntry	.create(LitEntryBean leb)	returns ResultBean
	.edit(LitEntryBean leb)	returns ResultBean
	.delete(LitEntryBean leb)	returns ResultBean
	.save(LitEntryBean leb)	returns ResultBean
	.showFiles(LitEntryBean leb)	returns ResultBean
	.showComments(LitEntryBean leb)	returns ResultBean

4.4.4. Transfer

Diese Klasse zeichnet sich zuständig für das Übertragen von Datenbankeinträgen in Txt-,Xml-, oder BibTexdateien. Und umgekehrt.

- Lesen von XML-Datei
- Lesen von BibTex-Datei
- Lesen von Text-Datei
- Erzeugen von XML-Datei
- Erzeugen von BibTex-Datei
- Erzeugen von Text-Datei

Transfer	.read(DataBean db)	returns ResultBean
	.read(DataBean db)	returns ResultBean
	.read(DataBean db)	returns ResultBean

.create(DataBean db)	returns ResultBean
.create(DataBean db)	returns ResultBean
.create(DataBean db)	returns ResultBean

4.4.5. File

Diese Klasse kümmert sich um das Anhängen von Dateien an Literatureinträge und um deren Ausgabe.

- Speichern
- Löschen
- Anzeigen

File	.save (FileBean fb)	returns ResultBean
	.delete(FileBean fb)	returns ResultBean
	.show(FileBean fb)	returns ResultBean

4.4.6. Interface Constants

Hier sind alle nötigen Konstanten zur Speicherung von Daten in der Datenbank enthalten.

Constants
~ARTICLE : int= 101
- BOOK : int= 102
- BOOKLET : int= 103
~ COLLECTION : int= 104
~ CONFERENCE : int= 105
~ INBOOK : int= 106
~ INCOLLECTION : int= 107
~ INPROCEEDINGS : int= 108
~ MANUAL : int= 109
- MASTERSTHESIS : int= 110
~ MISC : int= 111
~ PATENT : int= 112
~ PHDTHESIS : int= 113
~ PROCEEDINGS : int= 114
~ TECHREPORT : int= 115
~ UNPUBLISHED : int= 116
~ ABSTRACT : int= 1
- ADDRESS : int= 2
~ ANNOTE : int= 3
~ AUTHOR : int= 4
~ BOOKTITLE : int= 5
~ CHAPTER : int= 6
~ CONTENTS : int= 7
~ COPYRIGHT : int= 8
- CROSSREF : int= 9
- EDITION : int= 10
~ EDITOR : int= 11
~ HOWPUBLISHED : int= 12
~ INSTITUTION : int= 13
~ ISBN : int= 14
~ ISSN : int= 15
~ JOURNAL : int= 16
- KEY : int= 17
~ KEYWORDS : int= 18
~ LANGUAGE : int= 19
~ MONTH : int= 20
~ NOTE : int= 21
~ NUMBER : int= 22
~ ORGANIZATION : int= 23
- PAGES : int= 24
- PUBLISHER : int= 25
~ SCHOOL : int= 26
~ SERIES : int= 27
~ SHORT BOOKTITLE : int= 28
~ SHORT JOURNAL : int= 29
~ SHORT ORGANIZATION : int= 30
~ SHORT SERIES : int= 31
- TITLE : int= 32
- TYPE : int= 33
~ URL : int= 34
~ VOLUME : int= 35
~ YEAR : int= 36

5. Weiterführende Links der AL

Die AL hat auf ihrer Homepage noch weitere zusätzliche Dokumente zur Verfügung gestellt, wie zum Beispiel eine grobe Umschreibung aller Anwendungsfälle. Die Grafiken, die hier zu sehen sind, sind auch unter <http://www.mathematik.uni-marburg.de/~haim/it/ipg/docs/?C=M;O=A> zu finden.

6. Haftungsausschluss

Die AL-Gruppe übernimmt keine Haftung für die Aktualität, die inhaltliche Richtigkeit sowie für die Vollständigkeit der in ihrem Entwurfsdokument eingestellten Informationen, es sei denn die Fehler wurden vorsätzlich oder grob fahrlässig aufgenommen. Dies bezieht sich auf eventuelle Schäden materieller oder ideeller Art Dritter, die durch die Nutzung dieses Webangebotes verursacht wurden. Änderungen vorbehalten.