

## Analyse der Anwendungsfälle zu Datensätzen

In Klammern steht die Zuständigkeit für Implementierung,  
ungeb steht für Ungebunden, heißt Methode gehört nicht zu einem Objekt

### AF 16 - DS Import (Jakub)

WI liefert ein Dateiojekt oder Pfad. Erzeuge LitList mit ggf. mehreren BibTex oder Exception fehlende/fehlerhafte Datei o. für unerkannte Datensätze. WI kann LitList dann speichern oder Fehlermeldung ausgeben.

```
-> LitList ungeb.getBibTex(datei) throws FileError,  
      BibTexIFault, BibTexExist, BibTexPossiblyExist
```

#### Benötigt:

Objekte: LitList, BibTex, File, Exception

#### Problem:

Es müssen ggf. mehrere LitList's zurückgegeben werden, da nicht erkannte BibTex noch mal editiert werden können müssen oder BibTex ggf. mit bestehenden BibTex verglichen werden müssen.

#### Lösung:

Erkannte BibTex aufnehmen und Rest als Datei (erkannte ausschneiden) zurückgeben und noch mal in Eigenverantwortung(vom User) bearbeiten. Oder WI gibt dem User die Möglichkeit alle Einträge nochmal zu editieren.

### AF 17 - Manuelle Eingabe (Stefan)

WI liefert Objekt BibTex. Eintragen in DB

```
-> void ungeb.saveBibTex(newBibTex) throws BibTexExist,  
      BibTexPossiblyExist
```

```
-> boolean BibTex.isValidEntry() (Erweiterung)
```

#### Benötigt:

Objekte: BibTex

#### Problem:

Fehlende oder fehlerhafte Eingabe; BibTex bereits vorhanden

#### Lösung:

Eingebaute Methode prüft Gültigkeit -> WI verhindert Übergabe  
Oder prüfen nach Übergabe -> Exception

### AF 18 - editieren (Stefan)

...

```
-> void BibTex.edit()
```

#### Benötigt:

Objekte: BibTex

#### Problem:

Daten bereits vorhanden; BibTex-Eintrag veralten und muss neu aus DB geholt werden

#### Lösung:

Exception; Objektkontrolle überwacht Änderungen bzw. sperrt Objekte

### AF 19 - BibTex-Datei erstellen (Jakub)

Methode liefert zu einem BibTex-Eintrag den entsprechenden BibTex als String an WI. WI lässt noch mal bearbeiten und stellt Datei bereit.

-> String BibTex.outputBibTex()

Benötigt:

Objekte: BibTex

Problem:

Lösung:

### **AF 20 - BibTex-Eintrag löschen (Stefan)**

...

-> void BibTex.delete()

Benötigt:

Objekte: BibTex

Problem:

User ist nicht berechtigt BibTex-Eintrag zu löschen; anderer User hat BibTex-Eintrag offen und speichert ihn später wieder ab

Lösung:

Exception; Objektkontrolle überwacht Änderungen bzw. sperrt Objekte

### **AF 21 - Datei hinzufügen (Evgueni, Doru)**

Hochladen der Datei in entsprechendes Verzeichnis kopieren und umbenennen

-> void ungeb.addFile(BibTex-ID) throws FileError

Benötigt:

eindeutige Referenz auf BibTex-Eintrag

ggf. Dateititel und Pfad in DB speichern

Problem:

User ist nicht berechtigt Dateien hochzuladen; Speichern funktioniert nicht

Lösung:

Exception

### **AF 22 - Datei anzeigen (Evgueni, Doru o. Stefan o. WI)**

Dateien können direkt verlinkt werden -> methode liefert alle Links

-> String[] BibTex.linkFile()

Benötigt:

Dateititel und Pfad aus DB

Problem:

Lösung:

### **AF 23 - Datei löschen (Evgueni, Doru)**

Löschen der DB einträge, Datei kann auch in Tempordner verschoben werden

-> void BibTex.deleteFile(pfad)

Benötigt:

Dateipfad

Problem:

User ist nicht berechtigt Dateien zu löschen;

Lösung:

WI verhindert löschen oder Exception