Karlsruher Institut für Technologie

Programmieren – Wintersemester 2016/2017

Software-Design und Qualität (SDQ)
`https://sdqweb.ipd.kit.edu/wiki/Programmieren`
Jun.-Prof. Dr.-Ing. Anne Koziolek

Abschlussaufgabe 1 Literaturverwaltung

Version 1.1

Ausgabe: 10.02.2017 – 13:00
Abgabe: 10.03.2017 – 13:00

- Achten Sie darauf nicht zu lange Zeilen, Methoden und Dateien zu erstellen[1]
- Programmcode muss in englischer Sprache verfasst sein
- Kommentieren Sie Ihren Code angemessen: So viel wie nötig, so wenig wie möglich
- Wählen Sie geeignete Sichtbarkeiten für Ihre Klassen, Methoden und Attribute
- Verwenden Sie keine Klassen der Java-Bibliotheken ausgenommen Klassen der Pakete `java.lang`, `java.io` und `java.util`, es sei denn die Aufgabenstellung erlaubt ausdrücklich weitere Pakete[1]
- Achten Sie auf fehlerfrei kompilierenden Programmcode[1]
- Halten Sie alle Whitespace-Regeln ein[1]
- Halten Sie die Regeln zu Variablen-, Methoden und Paketbenennung ein und wählen Sie aussagekräftige Namen[1]
- Halten Sie die Regeln zu Javadoc-Dokumentation ein[1]
- Nutzen Sie nicht das default-Package[1]
- Halten Sie auch alle anderen Checkstyle-Regeln ein
- System.exit und Runtime.exit dürfen nicht verwendet werden[1]
- Achten Sie darauf, genau die vorgegebenen Ein- und Ausgabeformate einzuhalten.

Abgabemodalitäten

Die Praktomat-Abgabe wird am **Freitag, den 24. Februar 2017, um 13:00 Uhr**, freigeschaltet. Geben Sie die Java-Klassen als `.java`-Dateien ab. Laden Sie die `Terminal`-Klasse nicht mit hoch.

Bitte beachten Sie, dass das erfolgreiche Bestehen der öffentlichen Tests für eine erfolgreiche Abgabe nötig ist. Planen Sie ausreichend Zeit für Abgabeversuche ein, sollte der Praktomat Ihre Abgabe wegen einer Regelverletzung ablehnen.

[1] Der Praktomat wird die Abgabe zurückweisen, falls diese Regel verletzt ist.

A Literatur- und Zitatverwaltung

A.1 Einführung

In dieser Aufgabe implementieren Sie ein System zur Literatur- und Zitatverwaltung. Das System erlaubt die Verwaltung einer Literatursammlung mit Dokumenten unterschiedlicher Art und erlaubt auf Basis der verwalteten Daten einerseits die Suche nach Dokumenten, die bestimmte Kriterien erfüllen, als auch die Berechnung von abgeleiteten Metriken über Autoren oder Dokumente. Zusätzlich kann für eine geplante schriftliche Ausarbeitung ein Literaturverzeichnis mit einem durch den Benutzer gewählten Format generiert werden. Reale Systeme und Software, die teilweise ähnliche Funktionalitäten wie das hier beschriebene zu implementierende Programm anbieten sind bspw. CiteSeer^{X}[2], Google Scholar[3], BibTeX[4], JabRef[5] oder Citavi[6].

Im Folgenden werden zunächst die vorkommenden Konzepte kurz erklärt. Im Anschluss werden die geforderten Befehle, die das von Ihnen implementierte System unterstützen soll, mit beispielhaften Ein- und Ausgaben beschrieben.

A.2 Konzepte

Im Folgenden werden alle Konzepte wie Autoren, Journals, Konferenzen, ... auch als *Entitäten* bezeichnet. *Beziehungen* zwischen Entitäten sind Relationen wie z.B. die *referenziert*-Relation einer Veröffentlichung auf eine andere oder die *hat-geschrieben*-Relation zwischen Autor und Veröffentlichung.

Ist eine Entität im System durch ein Attribut oder eine Kombination von Attributen im Folgenden als „eindeutig (identifiziert)“ bezeichnet, beispielsweise durch ihren Namen, muss ihre Implementierung beim Versuch eine weitere Entität mit dem gleichen eindeutigen Identifier im System anzulegen eine geeignete Fehlermeldung ausgegeben. Hierbei ist die Groß- und Kleinschreibung von Zeichenfolgen **relevant**, also ist `example` ein anderer Identifier als `Example`. Beachten Sie dies insbesondere im Folgenden bei der Implementierung von Befehlen, die dem System Entitäten hinzufügen, wie zum Beispiel `add␣author` oder `add␣journal`.

A.2.1 Veröffentlichung

Eine wissenschaftliche Veröffentlichung hat immer:

- einen eindeutigen Identifier, der nur aus Kleinbuchstaben und Zahlen besteht,
- einen Titel,
- ein Veröffentlichungsjahr und
- mindestens einen Autor.

Es gibt hier nur eine Art von Veröffentlichung: Artikel. Artikel werden für eine Konferenz oder für eine Fachzeitschrift (engl. *journal*) geschrieben.

A.2.2 Konferenzen, Konferenzserien und Fachzeitschriften

Eine **(wissenschaftliche) Konferenz** ist eine Tagung auf der Wissenschaftler zusammenkommen und auf der wissenschaftliche Beiträge, also Artikel, in Vorträgen vorgestellt und diskutiert werden. Die vorgestellten Artikel werden in der Regel in sogenannten **Verhandlungen** (engl. *proceedings*) veröffentlicht.

Konferenzen zu einem Themenbereich finden häufig wiederkehrend unter dem gleichen Titel und Themenbereich an möglicherweise unterschiedlichen Orten statt. Dies wird als **Konferenzserie** bezeichnet. In einem bestimmten Jahr findet maximal eine Konferenz einer Konferenzserie statt. Es gibt auch Konferenzen, die nur einmalig stattfinden. Dann besteht eine Konferenzserie aus nur einer Konferenz. Eine Konferenz ist somit eine konkrete Ausprägung der Konferenzserie in einem bestimmten Jahr an einem festgelegten Ort. Wissenschaftliche Veröffentlichungen auf Konferenzen sind durch ihr Veröffentlichungsjahr immer genau einer konkreten Konferenz

[2] `https://citeseerx.ist.psu.edu/about/site`
[3] `https://scholar.google.de/`
[4] `https://de.wikipedia.org/wiki/BibTeX`
[5] `https://www.jabref.org/`
[6] `https://www.citavi.com/`

zugeordnet.

Eine **Fachzeitschrift** (engl. *journal*) ist eine regelmäßig erscheinende Zeitschrift zu einem bestimmten Fachthema. Ein Journal hat nicht unterschiedliche Ausprägungen mit unterschiedlichen Eigenschaften in einzelnen Jahren, wie dies bei Konferenzen der Fall ist.

Zur einfacheren Bezeichnung werden im Folgenden Konferenzen und Journals unter dem Begriff **Publikationsort** (engl. *venue*) zusammengefasst. Alle Konferenzserien und Journals sind durch ihren Typ (Serie oder Journal) und durch ihren Namen eindeutig identifiziert. Konferenzen sind durch die Konferenzserie und das Jahr eindeutig identifiziert. Ein Artikel kann in genau einem Publikationsort veröffentlicht sein.

A.2.3 Keyword

Um die Suche nach relevanten Artikeln, Konferenzserien, Konferenzen und Journals zu vereinfachen, sind diese oft mit Schlüsselwörtern (*keywords*) versehen. Ein Keyword ist eine Zeichenkette, die nur aus Kleinbuchstaben und keinen Sonderzeichen besteht. Beispiele für Keywords sind: `java`, `softwareengineering`, oder `performance`.

Keywords übertragen sich automatisch von Konferenzserien auf Konferenzen, von Konferenzen auf dort vorgestellte Artikel und von Journals auf Artikel. Möglicherweise doppelt vorkommende Keywords werden in allen relevanten Kontexten nur einmal gezählt oder erwähnt, beispielsweise bei der Auflistung aller Schlüsselworte eines Artikels. Das bedeutet beispielsweise, dass eine Konferenz *zusätzlich zu den eigenen Keywords* auch automatisch die Keywords der Konferenzserie hat.

A.2.4 Autor

Autoren haben einen Vor- und einen Nachnamen. Diese bestehen jeweils aus einem Wort, das nur aus den (Klein- und Groß-)Buchstaben A bis Z ohne Sonderzeichen besteht. Insbesondere enthalten Vor- und Nachname jeweils kein Leerzeichen. Mehrere Autoren können gemeinsam eine Veröffentlichung verfassen und haben dann eine definierte Reihenfolge. Diese Reihenfolge wird in Ihrer Software durch die Reihenfolge, in der Autoren zu einer Veröffentlichung hinzugefügt werden festgelegt. Der erste Autor wird als Erstautor, der zweite als Zweitautor, und so weiter bezeichnet.

Autoren sind im System durch die Kombination aus Vor- und Nachname eindeutig identifiziert. Es kann sein, dass mehrere Autoren den selben Vornamen oder den selben Nachnamen haben, jedoch niemals beides. Die Groß- und Kleinschreibung der Namen ist relevant und wird entsprechend geprüft. Hierbei sind zwei Autoren **unterschiedlich**, wenn sich ihre Namen nur durch die Groß- und Kleinschreibung unterscheiden.

A.2.5 Referenz

Veröffentlichungen können andere Veröffentlichungen referenzieren. Es können nur Veröffentlichungen mit einem echt früheren Veröffentlichungsdatum referenziert werden.

A.2.6 Literaturverzeichnis und Formate

Die von einer Veröffentlichung ausgehenden Referenzen werden in der Regel in einem für die Veröffentlichung spezifischen *Literaturverzeichnis* dargestellt. In der Regel wird hierbei im Text der Veröffentlichung die referenzierende Stelle mit einem Kürzel markiert und für das entsprechende Kürzel werden im Literaturverzeichnis weitere Informationen dargestellt. Die Elemente in einem Literaturverzeichnis haben eine festgelegte *Sortierung* und ein wählbares *Ausgabeformat*. Beide werden im Folgenden näher erläutert.

Sortierung Das Literaturverzeichnis ist immer eindeutig sortiert. Die Sortierung ist durch die folgende Liste beschrieben. Hierbei wird nur zum nächsten Punkt in der Liste fortgeschritten, falls zwei Einträge im vorhergehenden Punkt gleich sind. Das bedeutet, dass für zwei Eintrage gilt: liegt der Nachname eines Autors lexikographisch vor dem anderen, so wird der Eintrag vor diesem angeordnet, bei gleichen Nachnamen wird der Vorname des ersten Autors berücksichtigt, bei gleichen Nach- und Vornamen des Erstautors der Zweitautor, und so weiter. Für Texte (Namen) ist jeweils die aufsteigende lexikographische Ordnung gemeint (also `a` vor `aa` vor `ab` vor `bb`).

1. Nachname des ersten Autors,
2. Vorname des ersten Autors,
3. Nachname des zweiten Autors,
4. Vorname des zweiten Autors,
5. . . . ,
6. Nachname des letzten Autors,
7. Vorname des letzten Autors,
8. Titel,
9. Jahr der Veröffentlichung,
10. der eindeutige Identifier der Veröffentlichung.

Ein nichtvorhandener $n$ter Autor liegt in dieser Sortierung immer vor einem existierenden $n$ten Autor, also liegt hier ebenfalls eine lexikographische Ordnung der Autoren vor. Beispielhaft werden Publikationen wie folgt sortiert (— drückt aus, dass es keinen Zweit- oder Drittautor gibt):

Publikationsidentifier	Autor 1	Autor 2	Autor 3
1	A	—	—
2	A	B	—
3	B	—	—
4	B	C	D
5	B	D	—

Ausgabeformat Es gibt verschiedene Formate, in denen Referenzen dargestellt werden können. Dies umfasst sowohl die Form des Kürzels, als auch die Ausgabe im Literaturverzeichnis.

Im Rahmen dieser Aufgabe sollen die beiden Stile *IEEE Simplified* und *Chicago Simplified* implementiert werden. Eine genaue Beschreibung der Ausgabeformate findet sich in Abschnitt B.

A.3 Hinweise zur Implementierung

Beschreibung der Kommandos

Die ~~Namen aller genannten Entitäten~~für Parameter übergebenen Werte (also Namen von Entitäten, Titel von Artikeln, Keywords, etc.) enthalten nie ein Komma (`,`) oder ein Semikolon (`;`). Wird ein Parameter, der ein solches Symbol enthält übergeben, gibt Ihr Programm eine geeignete Fehlermeldung aus, bspw. mit Bezug auf eine fehlerhafte Parameteranzahl, eine unerwartete Liste als Parameter, oder Ähnliches.

Die spitzen Klammern (`<` und `>`) sind in den folgenden Beschreibungen ausdrücklich nie Teil der Befehle, sondern kennzeichnen Platzhalter für Eingaben durch den Benutzer. Das Zeichen `␣` ist ein Leerzeichen (ASCII-Zeichen 32 bzw. Unicode U+0020).

Listen von Platzhaltern sind immer in der Form `<list of Xs>` angegeben. Eine Liste kann ein oder mehrere Elemente enthalten (also nicht leer sein). Wird nur ein Element angegeben, hat die Eingabe die Form `<X>` . Es können beliebig viele weitere Elemente durch `;<X>` angehängt werden, wodurch sich insgesamt die Form `<X1>;<X2>;` . . . ergibt.

In Beispielen für Ein- und Ausgabe der Kommandos beginnen Zeilen, die Benutzereingaben darstellen mit den Zeichen `>` . Diese sind **nicht** Teil der Eingabe. Alle Zeilen, die nicht mit `>` beginnen, sind Ausgaben des Programms. In manchen Fällen ist die Reihenfolge der Ausgabezeilen nicht relevant, dies ist dann im Text angegeben.

Das Symbol `↩` drückt aus, dass der dieser Stelle folgende Zeilenumbruch **nicht** Teil der Ein- oder Ausgabe ist, sondern der Darstellung auf dem Aufgabenblatt dient.

Ausgabeformat
Achten Sie darauf, genau das in der Aufgabenstellung vorgegebene Ausgabeformat einzuhalten. Dies bezieht sich insbesondere auf Groß- und Kleinschreibung und Interpunktion, sowie Leerzeichen. Gleitkommazahlen werden **immer** mit genau drei Nachkommastellen im Format `X.XXX` ausgegeben, also mit einem Punkt (`.`) als Trennzeichen. Nach der dritten Nachkommastelle wird die Zahl **abgeschnitten**, es wird also **nicht gerundet**. Die Zahl $\frac{2}{3}$ würde beispielsweise als `0.666` ausgegeben.

Terminal-Klasse
Laden Sie für **diese Aufgabe** die `Terminal`-Klasse[a] von unserer Homepage herunter und platzieren Sie diese unbedingt im Paket `edu.kit.informatik`. Die Methode `Terminal.readLine()` liest eine Benutzereingabe von der Konsole und ersetzt `System.in`. Die Methode `Terminal.printLine()` schreibt eine Ausgabe auf die Konsole und ersetzt `System.out`. Verwenden Sie für jegliche Konsoleneingabe oder Konsolenausgabe die `Terminal`-Klasse. Verwenden Sie in keinem Fall `System.in` oder `System.out`. Laden Sie die Terminal-Klasse niemals zusammen mit Ihrer Abgabe hoch.

[a] https://sdqweb.ipd.kit.edu/lehre/WS1617-Programmieren/Terminal.java

Fehlerbehandlung
Ihr Programm soll auf ungültige Benutzereingaben mit einer aussagekräftigen Fehlermeldung reagieren. Aus technischen Gründen muss eine Fehlermeldung unbedingt mit `Error,␣` beginnen. Sie können hierfür die statische Methode `printError(String)` der `Terminal`-Klasse verwenden. Diese versieht die übergebene Nachricht mit dem geforderten Präfix. Eine Fehlermeldung führt nicht dazu, dass das Programm beendet wird; es sei denn, die nachfolgende Aufgabenstellung verlangt dies ausdrücklich. Achten Sie insbesondere auch darauf, dass unbehandelte `RuntimeExceptions`, bzw. Subklassen davon—sogenannte *Unchecked Exceptions*—nicht zum Abbruch des Programms führen.

A.4 Funktionalität

Im Folgenden wird die erforderliche Funktionalität des Systems beschrieben. Geben Sie Fehlermeldungen aus, falls die Benutzereingabe falsch formatiert ist, ein nicht vorhandenes Kommando verwendet, oder nicht-existente Entitäten referenziert. Die Fehlermeldung sollte so geformt sein, dass für den Benutzer erkenntlich ist, warum eine Eingabe abgelehnt wurde.

Beim Befehlen zum Erstellen von Entitäten oder Referenzen wird im Erfolgsfall `Ok` ausgegeben.

Die Beschreibung der Funktionalität des Systems ist in die folgenden Kategorien von Kommandos gegliedert:

- **Datenbestand:** Hinzufügen~~, Ändern und Löschen~~von Entitäten oder der Beziehung zwischen Entitäten.
- **Anfragen:** Informationen über Entitäten, Auflisten von Entitäten die gegebene Eigenschaften erfüllen.
- **Komplexe Anfragen:** Anfragen an die Daten, die mehrere Entitäten in Verbindung setzen.
- **Literaturverzeichnis:** Funktionalität, die die Erstellung eines Literaturverzeichnisses betrifft.

A.4.1 Eingabeformat

Platzhalter – `<venue>`

`<venue>` referenziert im Folgenden immer einen existierenden Publikationsort, also eine Konferenz oder ein Journal. Falls es sich um ein Journal handelt, entspricht `<venue>` dem Format `journal␣<name>`. Falls es sich um eine Konferenz handelt, entspricht `<venue>` dem Format `conference␣<series>,<year>`.

Platzhalter – `<publication>`

`<publication>` referenziert im Folgenden immer eine existierende Publikation durch ihren eindeutigen Identifier. Dieser wird in der Beschreibung der Befehle zum Anlegen der Publikationen durch `<id>` ausgedrückt.

Platzhalter – `<entity>`

`<entity>` referenziert im Folgenden immer eine ~~existierende Entität~~ der folgenden existierenden Entitäten: einen Publikationsort, eine Konferenzserie oder eine Publikation.

Dabei werden Publikationsorte wie in `<venue>` beschrieben ausgedrückt, Konferenzserien durch `series␣<name>`, Publikationen durch `pub␣<id>`.

A.4.2 Kommandos – Datenbestand

C0 – `quit`

Beendet das Programm.

Beispiel

```
> quit
Ok
```

C1 – `add␣author␣<first name>,<last name>`

Fügt dem System einen Autor hinzu.

Beispiel

```
> add␣author␣Eniola,Lowry
Ok
> add␣author␣Richard,Rhinelander
Ok
> add␣author␣Eniola,Lowry
Error, author with same name already added.
> add␣author␣Shashi,Afolabi
Ok
```

C2 – `add␣journal␣<name>,<publisher>`

Legt ein Journal mit dem gegebenen Namen und dem gegebenen Verleger an.

Beispiel

```
> add␣journal␣TSE,IEEE
Ok
```

C3 – `add␣conference␣series␣<name>`

Legt eine Konferenzserie mit dem gegebenen Namen an.

Beispiel

```
> add␣conference␣series␣ICSA
Ok
```

C4 – `add␣conference␣<series>,<year>,<location>`

Legt eine Konferenz an, die zur gegebenen Serie gehört und im gegebene Jahr am gegebenen Ort stattgefunden hat. Falls die Konferenzserie nicht existiert, gebe einen Fehler aus.

Beispiel

```
> add␣conference␣ICSA,2017,Gothenburg
Ok
> add␣conference␣QoSA,2016,Venice
Error, conference series not found.
```

C5 – `add␣article␣to␣<series/journal>:<id>,<year>,<title>`

Fügt einen wissenschaftlichen Artikel zu einem Publikationsort hinzu. `<year>` ist hierbei eine vierstellige Ganzzahl (*integer*) zwischen einschließlich 1000 und 9999. Gibt einen Fehler aus, falls das Journal/die Konferenz im gegebenen Jahr nicht existiert.

Beispiel

```
> add␣article␣to␣series␣ICSA:rr2017,2017,Components␣still␣have␣no␣interfaces
Ok
> add␣article␣to␣journal␣TSE:mvp2015,2015,Model␣Consistency
Ok
> add␣article␣to␣journal␣TSE:mvp2016,2016,Better␣Model␣Consistency
Ok
```

C6 – `written-by␣<publication>,<list of author names>`

Legt fest, dass die angegebenen Autoren an der gegebenen Publikation mitgeschrieben haben. Dieser Befehl kann auch mehrfach für die gleiche Publikation aufgerufen werden, dann werden immer weitere Autoren hinzugefügt. Ein Autor kann nur einmal zur gleichen Publikation hinzugefügt werden, ansonsten wird ein Fehler ausgegeben und keiner der übergebenen Autoren wird hinzugefügt. Das gleiche gilt, falls einer der Autoren nicht im System vorhanden ist. Die Reihenfolge der Autoren ist bei einem Aufruf durch die Reihenfolge in der gegebenen Liste festgelegt. Bei einem weiteren Aufruf des Kommandos werden die weiteren Autoren am Ende der Liste der Autoren hinzugefügt.

Jeder Autor wird als `<firstname>␣<lastname>` angegeben.

Beispiel

```
> written-by␣rr2017,Richard␣Rhinelander
Ok
> written-by␣nngrade,Eniola␣Lowry
Ok
> written-by␣mvp2016,Shashi␣Afolabi;Eniola␣Lowry
Ok
> written-by␣mvp2015,Shashi␣Afolabi;Richard␣Rhinelander
Ok
```

C7 – `cites␣<publication 1>,<publication 2>`

Legt fest, dass die Publikation mit dem Identifier `<publication 1>` die Publikation mit dem Identifier `<publication 2>` zitiert. Diese Beziehung ist nicht im Allgemeinen reflexiv, transitiv oder symmetrisch. Publikationen können nur Publikationen zitieren, die echt vor ihnen veröffentlicht wurden. Damit können Publikationen insbesondere nicht sich selbst zitieren und auch keine Publikationen, die im gleichen Jahr veröffentlicht

wurden. Eine Publikation kann dieselbe Publikation maximal einmal referenzieren. Eine Publikation kann beliebig viele andere Publikationen referenzieren und kann beliebig oft referenziert werden.

Beispiel

```
> cites␣rr2017,mvp2016
Ok
> cites␣rr2017,sommerville2015
Ok
> cites␣rr2017,mvp2015
Ok
> cites␣rr2017,rr2017
Error, publications cannot cite themselves
```

C8 – `add␣keywords␣to␣<entity>:<list of keywords>`

Fügt die gegebenen Keywords zur gegebenen Entität hinzu. Ist eines der Keywords bereits zu der Entität hinzugefügt, wird dieses Keyword ignoriert, die anderen jedoch normal hinzugefügt. `<entity>` muss hierbei eine wie unter Unterunterabschnitt A.4.2 beschriebene existierende Entität sein. Ist dies nicht der Fall (existiert die Entität also nicht), wird eine passende Fehlermeldung ausgegeben.

Ist eines oder mehrere der Keywords falsch formatiert (d.h. es besteht nicht nur aus Kleinbuchstaben), wird entweder eine Fehlermeldung ausgegeben und es wird keines der Keywords hinzugefügt, oder die restlichen Keywords werden normal hinzugefügt.

Beispiel

```
> add␣keywords␣to␣pub␣sommerville2015:swe;reference;java;oop
Ok
> add␣keywords␣to␣pub␣rr2017:mdsd;components;java
Ok
> add␣keywords␣to␣pub␣mvp2016:mdsd;java;oop
Ok
> add␣keywords␣to␣journal␣TSE:swe
Ok
> add␣keywords␣to␣series␣ICPE:swe;performance
Ok
```

A.4.3 Kommandos – Anfragen

Besteht eine Programmausgabe aus mehreren Elementen, wird jedes Element in einer eigenen Zeile ausgegeben, es sei denn, dies ist anders spezifiziert.

C9 – `all␣publications`

Gibt die Identifier aller Publikationen aus. Die Reihenfolge der Ausgabe ist hierbei nicht festgelegt. Gibt es keine Publikationen, wird nichts ausgegeben.

Ist beispielsweise die Menge aller vorhandenen Publikationsidentifier $\{$ `mk2017`, `ky2017a`, `ky2017b` $\}$, so gibt das Kommando `all␣publications` (in irgendeiner Reihenfolge) aus:

Beispiel

```
> all␣publications
ky2017b
mk2017
ky2017a
```

C10 – `list␣invalid␣publications`

Gibt die Identifier der Publikationen aus, die keinen Autor zugewiesen haben. Die Reihenfolge der Ausgabe ist hierbei nicht festgelegt. Gibt es keine solchen Publikationen, wird nichts ausgegeben.

Beispiel

```
> list␣invalid␣publications
mk2017
```

C11 – `publications␣by␣<list of authors>`

Gibt die Identifier aller Publikationen aus, an denen mindestens einer der gegebenen Autoren beteiligt ist. Jeder Autor wird als `<firstname>␣<lastname>` angegeben. Existiert einer der Autoren nicht im System, wird eine passende Fehlermeldung ausgegeben und das Kommando gibt sonst nichts aus.

Jeder Identifier wird in einer separaten Zeile ausgegeben. Die Ausgabe beinhaltet sowohl Publikationen in denen nur einer oder eine Untermenge der gegebenen Autoren beteiligt sind, als auch Publikationen, in denen alle Autoren beteiligt sind. Die Reihenfolge der Ausgabe ist hierbei nicht festgelegt. Jede Publikation wird maximal einmal ausgegeben. Gibt es keine solchen Publikationen, wird nichts ausgegeben.

Beispiel

```
> publications␣by␣Eniola␣Lowry
mvp2015
nngrade
mvp2016
> publications␣by␣unnamed␣author;Eniola␣Lowry
Error, author "unnamed author" not found.
> publications␣by␣unnamed␣author
Error, author "unnamed author" not found.
```

C12 – `in␣proceedings␣<series>,<year>`

Gibt die Identifier aller Publikationen, die in der angegebenen Konferenzserie im angegebenen Jahr veröffentlich wurden, aus. Die Reihenfolge der Ausgabe ist hierbei nicht festgelegt. Gibt es keine solchen Publikationen, wird nichts ausgegeben. Gibt es die angegebene Konferenzserie nicht, oder keine Konferenz dieser Serie im gegebenen Jahr, wird eine passende Fehlermeldung ausgegeben.

Beispiel

```
> in␣proceedings␣ICSA,2017
rr2017
> in␣proceedings␣QoSA,2016
> in␣proceedings␣uksa,2016
Error, series "uksa" not found.
> in␣proceedings␣QoSA,2015
bspblication2015a
bspblication2015b
```

A.4.4 Kommandos – Komplexe Anfragen

C13 – `find␣keywords␣<list of keywords>`

Gibt die Identifier aller Publikationen aus, die **alle** der angegebenen Keywords besitzen. Es müssen nicht alle angegebenen Keywords bereits im System verwendet sein. Falls Keywords in diesem Befehl verwendet werden, die keiner Publikation zugeordnet sind, wird nichts ausgegeben.

Beispiel

```
> find␣keywords␣swe,reference
> find␣keywords␣swe;reference
somerville2015
> find␣keywords␣swe,reference,trivial
> find␣keywords␣swe;reference;trivial
> find␣keywords␣mdsd
mvp2016
rr2016
```

C14 – `jaccard␣<list of words 1>␣<list of words 2>`

Berechnet den sogenannten *Jaccard-Koeffizienten*[7] für zwei Mengen von Schlüsselwörtern. Dieser ist wie folgt definiert:

$$J(A,B) = \begin{cases} 1, & \text{falls } A \text{ und } B \text{ beide leer sind } (A \cup B = \emptyset) \\ \dfrac{|A \cap B|}{|A \cup B|}, & \text{sonst} \end{cases}$$

Hierbei sind A und B zwei Mengen, $A \cap B$ und $A \cup B$ sind die Schnittmenge ($\cap$) beziehungsweise Vereinigung ($\cup$) der beiden Mengen und $|A|$ ist die Anzahl der Elemente der Menge A.

Beispiel

```
> jaccard␣a;b;c␣d;e
0.000
> jaccard␣a;b;c;d;e␣b;c;d;e;f
0.666
```

C15 – `similarity␣<publication 1>,<publication 2>`

Gibt an, wie ähnlich sich zwei Publikationen bezüglich ihrer Keywords sind. Die beiden Publikationen sind hierbei durch ihre Publikationsidentifier gegeben. Existiert eine der beiden Publikationen nicht, wird eine passende Fehlermeldung ausgegeben.

Als Ähnlichkeitsmaß wird der in C14 beschriebene *Jaccard-Koeffizient* der Keywordmengen der beiden Publi-

[7] `https://en.wikipedia.org/w/index.php?title=Jaccard_index&oldid=754661103` (Version vom 13.12.2016)

kationen verwendet. Die Mengen A und B im obengenannten $J(A, B)$ sind also die Mengen der Keywords der jeweiligen Publikation. Beachten Sie bitte, dass sich Keywords von Konferenzserien auf Konferenzen und von Konferenzen und Journals auf Artikel übertragen, wie in Unterunterabschnitt A.2.3 beschrieben.

Geben Sie das Ergebnis als Gleitkommazahl zwischen 0 und 1 aus.

Beispiel

```
> similarity␣rr2017,mvp2016
0.500
```

C16 – `direct␣h-index␣<list of citation counts>`

Gibt den h-Index eines Autors aus, der Publikationen mit den übergebenen Zitationsanzahlen (Anzahl an eingehenden Referenzen) veröffentlicht hat. Die übergebene Liste muss also eine nicht-leere Liste von Ganzzahlen größer gleich null sein.

Der h-Index ist eine Angabe für die Anzahl und den Einfluss von Veröffentlichungen eines Autors. Er ist wie folgt definiert:

> Ein Autor hat den (ganzzahligen) h-Index i, wenn mindestens i Publikationen des Autors i mal zitiert wurden, und dies für keine größere Zahl i gilt.

Hinweis: Die Wikipedia-Seite zum h-Index[a] enthält zahlreiche Beispiele und Erklärungen. Die Ausgabe des Befehls ist aufgrund der Definition des h-Index unabhängig von der Reihenfolge der Listenelemente.

[a] `https://de.wikipedia.org/w/index.php?title=H-Index&oldid=161470685` (Version vom 09.01.2017)

Beispiel

```
> direct␣h-index␣17;3;1;5
3
> direct␣h-index␣8;6;8;4;8;6
5
```

C17 – `h-index␣<firstname>␣<lastname>`

Gibt den h-Index des Autors, gegeben der im System gespeicherten Veröffentlichungen aus. Für nicht definierte Autoren wird eine passende Fehlermeldung ausgegeben. Wurden keine Publikationen des Autors zitiert, wird `0` ausgegeben.

Beispiel

```
> h-index␣Jun␣Martz
3
> h-index␣Noor␣Vilhjalmsson
10
```

C18 – `coauthors␣of␣<firstname>␣<lastname>`

Gibt die Namen aller Mitautoren des gegebenen Autors aus. Für nicht definierte Autoren wird eine passende Fehlermeldung ausgegeben. Jeder Autor wird hierbei im Format `<first name>␣<last name>` in einer eigenen Zeile ausgegeben. Mitautoren sind dabei Autoren, die in irgendeiner Publikation gemeinsam beteiligt sind. Ein Autor ist nicht Mitautor zu sich selbst. Jeder Autor wird maximal einmal ausgegeben. Die Reihenfolge der Ausgabe ist nicht relevant. Gibt es keine Mitautoren zum gegebenen Autor, wird nichts ausgegeben.

Beispiel

```
> coauthors␣of␣Shashi␣Afolabi
Richard␣Rhinelander
Eniola␣Lowry
> coauthors␣of␣Eniola␣Lowry
Shashi␣Afolabi
```

C19 – `foreign␣citations␣of␣<author name>`

Gibt die Identifier aller *Fremdzitate* auf den genannten Autor aus. Für nicht definierte Autoren wird eine passende Fehlermeldung ausgegeben. Fremdzitate sind hierbei definiert als Zitate die nicht aus Publikationen stammen, an denen Autoren mitgeschrieben haben, die bereits (möglicherweise in anderen Publikationen) mit dem genannten Autor zusammen an Publikationen gearbeitet haben. Ein Autor hat im Sinne dieses Kommandos immer bereits mit sich selbst an Publikationen gearbeitet, also kann nie ein Fremdzitat aus einer Publikation kommen, an der der Autor selbst mitgeschrieben hat.

`<author name>` muss hierbei die Form `<first name>␣<last name>` haben.

Beispiel

Publikationsidentifier	Autoren	Referenz auf
p0	a	—
p1	a, b	p0
p2	b, c	p0
p3	c, d	p0

In diesem Fall hat Publikation `p0` nur ein Fremdzitat, da Publikationen, an denen Autor B (hier: `p2`) beteiligt ist nicht gezählt werden.

```
> foreign␣citations␣of␣a
p3
```

A.4.5 Kommandos – Literaturverzeichnis

C20 – `direct␣print␣conference␣<style>:↩`
`<author 1>,<author 2>,<author 3>,<title>,↩`
`<conference series name>,<location>,<year>`

Gibt ein Literaturverzeichnis (siehe Unterunterabschnitt A.2.6 und Abschnitt B) aus, das nur aus einem Eintrag besteht, der einen Konferenzartikel darstellt. Jeder der Autoren muss im Format `<first name>␣<last name>` angegeben sein. `<author 1>` muss immer so angegeben sein, `<author 2>` und `<author 3>` können auch leer sein. Sämtliche Platzhalter enthalten nie ein `,`. Die Konferenzserie, die Konferenzausprägung im gegebenen Jahr und am gegebenen Ort und die gegebenen Autorennamen müssen nicht im Datenbestand vorhanden sein.

Beispiel

```
> direct␣print␣conference␣ieee:Sergey␣Brin,Lawrence␣Page,,The␣Anatomy␣of␣↩
  a␣Large-Scale␣Hypertextual␣Web␣Search␣Engine,WWW,Brisbane Australia,1998.
[1]␣S.␣Brin,␣and␣L.␣Page,␣"The␣Anatomy␣of␣a␣Large-Scale␣Hypertextual␣↩
Web␣Search␣Engine,"␣in␣Proceedings␣of␣WWW,␣Brisbane␣Australia,␣1998.
```

C21 – `direct␣print␣journal␣<style>:↩`

`<author 1>,<author 2>,<author 3>,<title>,<journal title>,<year>`

Gibt wie im vorhergehenden Befehl beschrieben ein Literaturverzeichnis aus, das aus genau einem Eintrag besteht, der einen Journalartikel mit den gegebenen Eigenschaften darstellt.

Beispiel

```
> direct␣print␣journal␣ieee:Edsger␣Dijkstra,,,Go␣To␣Statement␣Considered␣Harmful,↩
  Comm.␣of␣the␣ACM,1968
[1]␣E.␣Dijkstra,␣"Go␣To␣Statement␣Considered␣Harmful,"␣Comm.␣of␣the␣ACM,␣1968.
```

C22 – `print␣bibliography␣<style>:<list of publication ids>`

Gibt ein sortiertes Literaturverzeichnis (siehe Unterunterabschnitt A.2.6 und Abschnitt B) aus.

`<style>` ist entweder `ieee` oder `chicago` und definiert, wie ein Eintrag im Literaturverzeichnis formatiert werden soll. Für jeden Eintrag wird eine Zeile ausgegeben, die wie im Stil definiert formatiert ist.

Ist einer der Publikationsidentifier nicht im System vorhanden wird nur eine entsprechende Fehlermeldung ausgegeben und kein Literaturverzeichnis. Das gleiche gilt, falls für einen Eintrag keine Autoren zugewiesen sind, er also im Sinne von C10 *invalid* ist. Kommt eine Publikation in der gegebenen Liste mehrfach vor, wird sie trotzdem nur einmal im Literaturverzeichnis ausgegeben.

Beispiel

```
> print␣bibliography␣ieee:mvp2016;mvp2015;rr2017
[1]␣S.␣Afolabi␣and␣R.␣Rhinelander,␣"Model␣Consistency,"␣TSE,␣2015.
[2]␣S.␣Afolabi␣and␣E.␣Lowry,␣"Better␣Model␣Consistency,"␣TSE,␣2016.
[1]␣S.␣Afolabi␣and␣E.␣Lowry,␣"Better␣Model␣Consistency,"␣TSE,␣2016.
[2]␣S.␣Afolabi␣and␣R.␣Rhinelander,␣"Model␣Consistency,"␣TSE,␣2015.
[3]␣R.␣Rhinelander,␣"Components␣still␣have␣no␣interfaces,"␣in␣Proceedings␣of␣ICSA,␣↩
Gothenburg,␣2017.
> print␣bibliography␣chicago:rr2017;mvp2016;mvp2015;rr2017
(Afolabi,␣2015)␣Afloabi,␣Shashi,␣and␣Rhinelander,␣Richard.␣"Model␣Consistency."␣↩
TSE␣(2015).
(Afolabi,␣2016)␣Afolabi,␣Shashi,␣and␣Lowry,␣Emiola.␣"Better␣Model␣Consistency."␣↩
TSE␣(2016).
(Afolabi,␣2016)␣Afolabi,␣Shashi,␣and␣Lowry,␣Emiola.␣"Better␣Model␣Consistency."␣↩
TSE␣(2016).
(Afolabi,␣2015)␣Afloabi,␣Shashi,␣and␣Rhinelander,␣Richard.␣"Model␣Consistency."␣↩
TSE␣(2015).
(Rhinelander,␣2017)␣Rhinelander,␣Richard.␣"Components␣still␣have␣no␣interfaces."␣↩
Paper␣presented␣at␣ICSA,␣2017,␣Gothenburg.
```

Hinweis: Im obigen Beispiel wurde ausschließlich die Reihenfolge geändert.

B Literaturverzeichnisstile

Die hier vorgestellten Zitierstile sind vereinfachte und angepasste Versionen von häufig verwendeten Zitierstilen. In Klammern hinter dem Titel ist jeweils ein Identifier angegeben, über den der jeweilige Stil durch den Benutzer Ihres Systems referenziert werden kann (`ieee` und `chicago`). Unterabschnitt B.1 und Unterabschnitt B.2 beschreiben hierbei für jede Art von Publikation, wie diese im jeweiligen Stil im Literaturverzeichnis dargestellt wird.

Hinweis: Beachten Sie insbesondere die unterschiedliche Interpunktion: Punkt oder Komma nach Autoren und nach dem Titel und Hochkommata (") vorhanden / nicht vorhanden.

B.1 IEEE Simplified (`ieee`)

Typ	Formatierung
Konferenzartikel:	`<bibidentifier>␣<author list>,␣"<title>,"␣in␣Proceedings␣of␣↩` `<conference series name>,␣<location of conference>,␣<year of conference>.`
Journalartikel:	`<bibidentifier>␣<author list>,␣"<title>,"␣<journal title>,␣<year>.`

`<bibidentifier>` ist für diesen Stil so definiert: `[<order>]`, wobei `<order>` die Position des referenzierten Werks im wie oben beschrieben sortierten Literaturverzeichnis ist. Somit ist `<bibidentifier>` für das an erster Stelle stehende Werk `[1]`, für das zweite `[2]`, und so weiter.

`<author list>` ist für diesen Stil so definiert:

Fall	`<author list>`
Genau ein Autor:	`<author 1 name>`
Genau zwei Autoren:	`<author 1 name>␣and␣<author 2 name>`
Drei oder mehr Autoren:	`<author 1 name>␣et␣al.`

Hierbei ist `<author i name>` wie folgt definiert: `<uppercase first letter of first name>.␣<last name>`

B.2 Chicago Simplified (`chicago`)

Typ	Formatierung
Konferenzartikel:	`<bibidentifier>␣<author list>.␣"<title>."␣Paper␣presented␣at␣↩` `<conference series name>,␣<year of conference>,␣↩` `<location of conference>.`
Journalartikel:	`<bibidentifier>␣<author list>.␣"<title>."␣<journal title>␣(<year>).`

`<bibidentifier>` ist für diesen Stil so definiert: `(<last name of first author>,␣<year>)`

`<author list>` ist für diesen Stil so definiert:

Fall	`<author list>`
Genau ein Autor:	`<author 1 name>`
Genau zwei Autoren:	`<author 1 name>,␣and␣<author 2 name>`
Mehr als zwei Autoren:	`<author 1 name>,␣<author 2 name>,␣<author 3 name>,` ⋯ `,␣and␣<last author name>`

Die drei Punkte (⋯) bedeuten hierbei, dass alle Autoren nacheinander genannt werden. Die Autorenliste nennt also explizit alle Autoren und kürzt die Auflistung nicht mit `et al.` oder Punkten ab. Hierbei ist `<author i name>`: `<last name>,␣<first name>`