Files, objects, inheritance, interfaces, aggregation, string manipulation, enumeration, and exceptions

Problem statement:

You will be implementing a project called BookStore. The store has a digital library. You will read scientific publications (papers) information from a text file (publcaitions.txt) and print out the citation of each paper in a specific format. Papers information is stored in a text file. Each paper is stored on a single line. There is two types of publications in the text file: journals and conference proceedings. Journals have 9 fields separated by a semicolon in the following order:

authors; title; venue; publisher; volume; number; starting page; end page; year

Conference proceedings have eight fields separated by a semicolon in the following order:

authors; title; venue; publisher; city; starting page; end page; year

Your program will read the file, and print out the scientific citation of each publication. Journals citations should be printed in the following format:

authors, "title", venue (acronym of the venue), publisher, volume(number): starting page – end page, year

Conference proceedings should be printed in the following format:

authors, "title", venue (acronym), publisher, city, year, pp: starting page – end page

Authors' names of each publication should be printed in the following format:

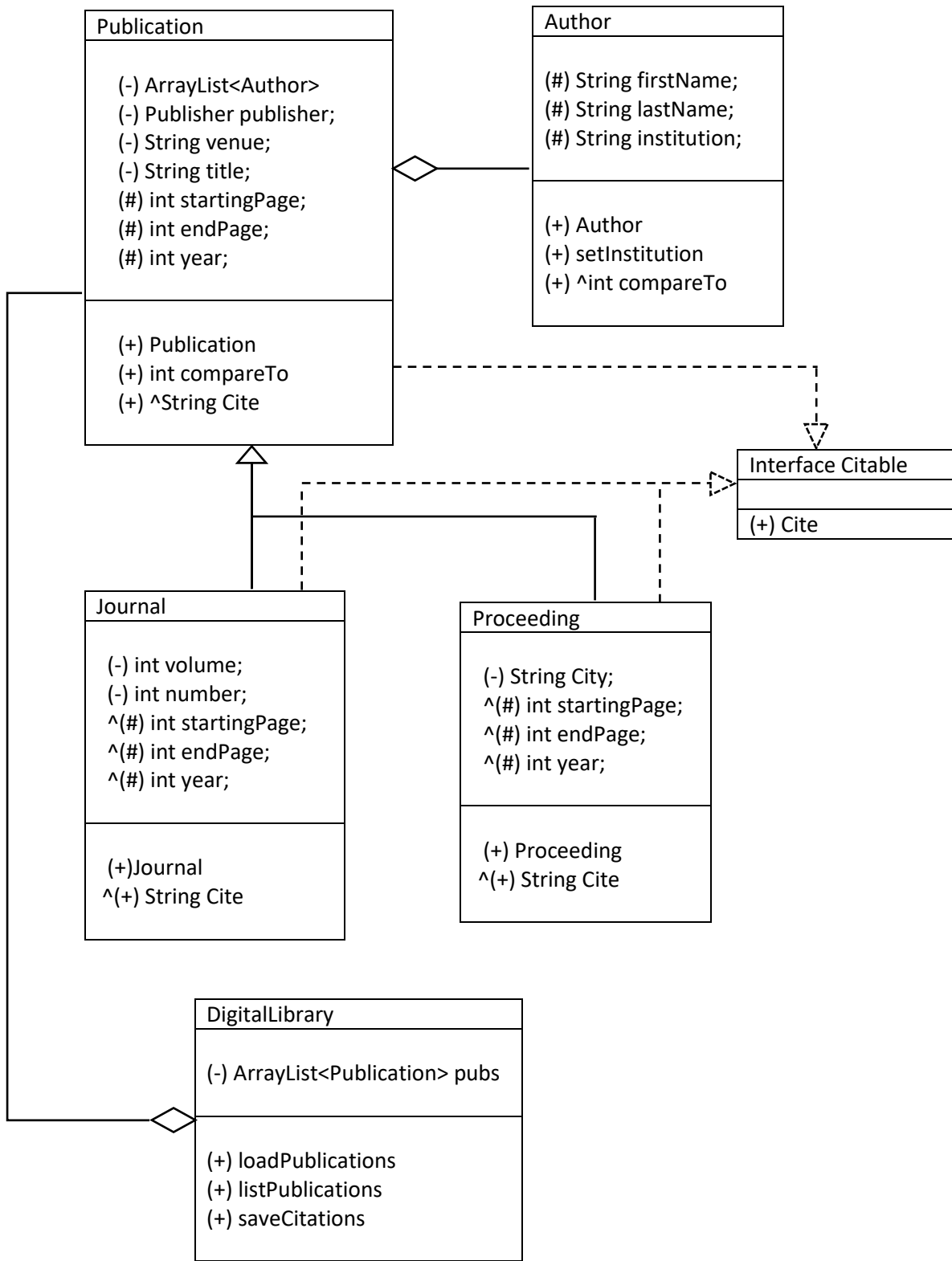**<First letter of first name. Full last name>**

In case of multiple authors', their names should be sorted in an alphabetical order based on last name and then separated by commas. An "and" should separate the last two names. An example of a journal citation:

A. Mahmoud and G. Williams, "Detecting, classifying, and tracing non-functional software requirements", Requirements Engineering Journal (REJ), IEEE, 357(381): 21-3, 2016

An example of a conference proceeding citation:

A. Mahmoud, M. Tushev, and G. Williams, "Mining Twitter feeds for software user requirements", Requirements Engineering (RE), ACM, Lisbon, 2017, pp:1-10

The class diagram of the project is shown below. Feel free to add any other methods or fields as necessry.

Publication

(-) ArrayList<Author>
(-) Publisher publisher;
(-) String venue;
(-) String title;
(#) int startingPage;
(#) int endPage;
(#) int year;

(+) Publication
(+) int compareTo
(+) ^String Cite

Author

(#) String firstName;
(#) String lastName;
(#) String institution;

(+) Author
(+) setInstitution
(+) ^int compareTo

Interface Citable

(+) Cite

Journal

(-) int volume;
(-) int number;
^(#) int startingPage;
^(#) int endPage;
^(#) int year;

(+)Journal
^(+) String Cite

Proceeding

(-) String City;
^(#) int startingPage;
^(#) int endPage;
^(#) int year;

(+) Proceeding
^(+) String Cite

DigitalLibrary

(-) ArrayList<Publication> pubs

(+) loadPublications
(+) listPublications
(+) saveCitations

Instructions:

1- The class DigitalLibrary implement the following public methods
   a- `loadPublications`: loads publications' information from the publications.txt file into the publications `ArrayList`.
   b- `listPublications`: prints all citations in their scientific formats.
   c- `saveCitations`: saves all scientific citations to a text file. The name of the file should be supplied by the user.

2- Before printing out the publications, they should be sorted based on first author's last name, if the name is same, they get sorted based on the venue's name, and if the venue is the same, they get sorted based on the year. Each citation starts with its number [1], [2], etc. as follows:

```
[1] C. Arora, L. Briand, S. Nejati, M. Sabetzadeh, "An active learning approach for improving the
accuracy of automated domain model extraction", Transactions on Software Engineering Methodology
(TOSEM), ACM, 14(34): 28-1, 2019
[2] C. Baker, C. Fillmore, J. Lowe, "The Berkeley Framenet project", International Conference on
Computational Linguistics (ICOCL), IEEE, Washington, 1998, pp:86-90
[3] F. Bert, M. Giacometti, M. Rosaria, R. Siliquini, "Smartphones and health promotion: A review of
the evidence", Journal of Medical Systems (JOMS), SPRINGER, 99(109): 38-1, 2013
[4] D. Blei, M. Jordan, A. Ng, "Latent Dirichlet Allocation", Journal Machine Learning Research
(JMLR), ELSEVIER, 993(1022): 3-1, 2003
[5] N. Jha, A. Mahmoud, "Using frame semantics for classifying and summarizing application store
reviews", Empirical Software Engineering (ESE), IEEE, 3734(3767): 23-6, 2018
[6] A. Mahmoud, G. Williams, "Detecting, classifying, and tracing non-functional software
requirements", Requirements Engineering Journal (REJ), ACM, 357(381): 21-3, 2016
[7] M. Harman, Y. Jia, W. Martin, F. Sarro, Y. Zhang, "The app sampling problem for app store mining",
Mining Software Repositories (MSR), WILEY, Paris, 2015, pp:123-133
[8] A. Mahmoud, G. Williams, "Mining Twitter feeds for software user requirements", Requirements
Engineering (RE), IEEE, Lispon, 2017, pp:1-10
```

3- Publication, Journal, and Proceeding classes implements the `Citable` interface

4- Both Journal and Proceeding classes override the `Cite()` method of their parent class to add the second part of the citation: **<volume(number): starting page – end page, year>** for journals and < **city, year, pp: starting page – end page** > for conference proceedings

5- We want to make sure no venues from outside our list of venues is posted. Therefore, we have the following enumeration of all allowed venues.

```
enum Publisher{ELSEVIER, SPRINGER, IEEE, TAYLORFRANCIS, WILEY, ACM};
```

If the venue name is not in the enumeration, your code should throw an `IllegalArgumentException`

6- The following command will return each field of the publication line in the publication.txt file as a separate element of the String array `fields`.

```
String[] fields = in.nextLine().Split(";");
```

7- The code for the `BookStore.java`, `Citable.Java`, and partial `DigitalLibrary.java` are provided. The file Publcaitions.txt is also provided.