Venue Recommendation

Pre-processing
This project requires an additional external library in order to enable access and processing of JSON-objects:

- `javax.json-1.1.2`

RetrieveData.java

Program description
Retrieves the `dblp` data about conference editions and venues related to the authors contained in `similarauthors.txt`.

Required files:
- `authorkeys.txt`
- `similarauthors.txt`
- `dblp_papers/`
- `dblp_confs/`

Output:
- `authorconfs.txt`
- `venues.txt`
- `conferences.txt`

Data structure
`static String authorkeyfile` Path to authors-key-file
`static String simfile` Path to authors-similarity-file
`static String ntdirpath_p` Path to directory with filtered `DBLP` papers data
`static ArrayList<String> ntfiles_p` Names of `.nt`-files with `DBLP` papers data
`static String ntdirpath_c` Path to directory with filtered `DBLP` conferences data
`static ArrayList<String> ntfiles_c` Names of `.nt`-files with `DBLP` conferences data

`Author (String id, String uri)`
Represents an author with the corresponding ID and URI. The URI is obtained using the authors-key-file.

`Conference (String id, String uri, String venue, String title, String series, int year)`
Represents a conference edition where `venue` refers to the corresponding conference series and is assigned later (after the instance was created).

`Venue (String id, String uri, String name, ArrayList<Conference> conferences)`
Represents a conference series where `conferences` refers to the conference editions corresponding to this venue; the list is assigned later (after the instance was created).
The venue's name is obtained by calling the method `setName()`.

> **`String setName(int count)`**
> Sends a request to the `DBLP API` in order to obtain the venue's name where `count` refers to the maximum number of result data sets to be retrieved.

Methods

`ArrayList<Author> getAuthors()`
Returns a list of authors included in the authors-similarity-file.

`String[] getConfURIs(Author author)`
Returns a list of URIs referring to conference editions in which the author corresponding to `author` published.

`ArrayList<Venue> retrieveVenues(ArrayList<Conference> conferences)`
Retrieves a list of venues by grouping the given list of conferences by their venues and assigns an ID to each entry in the list. Furthermore, a list of conferences is assigned to each entry in the venue list and the corresponding venue ID is assigned to each entry in the list `conferences`.

`void writeToNewFile(String path, String content)`
Creates a new file using the given `path` and stores the text given as `String content`.

Program Flow (main method)

- The input files are processed
- The list with names of the `.nt`-files containing `DBLP` papers data is obtained
- The list with names of the `.nt`-files containing `DBLP` conferences data is obtained
- Names of the output files are defined
- A list of authors and the associations of authors and URIs of conference editions are retrieved
- A list of conferences is retrieved and an ID is assigned to each entry in this list
- The authors-conference editions-file is created
- The venues-file is created
- The conferences-file is created

VenueRecommendation.java

Program description

Produces a `.txt`-file containing venue recommendations and recommendation weights corresponding to the data given in the similar-authors-file.
Furthermore, a `.js`-file is produced containing `JSON`-objects each containing the venue recommendations for an author.

Input:
- `conferences.txt`
- `venues.txt`
- `similarauthors.txt`
- `authorkeys.txt`
- `authorconfs.txt`
- *`suffix`* `(i.e., s95)`

Output:
- `v_recommendations_`*`suffix`*`.txt`
- `v_recommendations_`*`suffix`*`.js`

Data structure

`static String authorkeyspath` Path to authors-keys-file
`static String authorconfspath` Path to authors-conferences-file
`static String suffix` Suffix for output-files and produced `JSON`-object-names
`static String recommendationspath` Path to `.txt`-output-file
`static String recommendationsfile` Content of `.txt`-output-file
`static String recommendationsjspath` Path to `.js`-output-file
`static String recommendationsjsfile` Content of `.js`-output-file
`static HashMap<String, Conference> confslist` Reference list with all conferences
`static HashMap<String, Venue> venueslist` Reference list with all venues
`static HashMap<String, Author> authorslist` Reference list with all authors

`Conference (String id, String name, String venue, int year)`
Represents a conference edition. When the reference list `conferences` is created, its objects are added by assigning all attributes right away. Once the list exists, an object of this class can be instantiated by only assigning the attribute `id`; the remaining attributes are assigned automatically by retrieving the corresponding item of the list `conferences`.

`Venue (String id, String name, ArrayList<Conference> conferences)`
Represents a venue. Conference editions which are related to each other are grouped to a venue.

`Author (String id, String name, ArrayList<Conference> conferences), ArrayList<String[]> simauthors`
Represents an author. The list `conferences` contains conference editions in which this author published. Each entry in the list `simauthors` refers to a similar author and is represented by an array with 2 entries; the first entry contains the similar author's ID, the second entry contains the similarity weight of both authors.

Methods

`boolean publishedInVenue(String authorid, String venueid)`
Checks if the author corresponding to `authorid` published in any conference edition associated with the venue corresponding to `venueid`.

`ArrayList<String[]> publishedInConference(ArrayList<String[]> simauthors, String conferenceid)`
Returns a subset of `simauthors` containing entries corresponding to authors who published in the conference edition corresponding to `conferenceid`. Each entry of the returned list consists of an array with 2 entries; the first entry contains the similar author's ID, the second entry contains the similarity weight.

`void retrieveRecommendations(ArrayList<String> authors)`
Computes venue recommendations for each author represented in the list `authors` and sorts the recommendations by similarity value (`sortRecommendations()`).
The content of the output-text-file is assigned containing the recommended venues and corresponding similarity values for each author.
Furthermore, the output-js-file is assigned containing a `JSON`-object for each of the authors in `authors`. Each `JSON`-object is named according to the represented author and the suffix given as input and contains the following information:

```
VRecAxsuffix = { "id": "Ax", "name": "…",
  "recommendations": [
      { "venueid": "…", "venuename": "…",
        "conferences": [
            { "id": "…", "name": "…", "year": xxyy,
              "simauthors": [{ "id": "…", "name": "…", "weight": x.yz } …]
            } …
        ]
      } …
  ]
};
```

`ArrayList<String[]> sortRecommendations(ArrayList<String[]> unsorted)`
Sorts the list `unsorted` given as input by similarity value and returns the sorted list.
The list `unsorted` contains venue recommendations; each item of the list consists of an array with two entries: a venue-ID and a similarity value.

`void writeToNewFile(String path, String content)`
Creates a new file using the given `path` and stores the text given as `String content`.

Program Flow (main method)

- Input-files and the suffix given as input are processed
- The conferences-reference-list and venues-reference-list are retrieved
- The authors whose similar authors are listed in the similar-authors-file are retrieved and stored in the authors-reference-list
- The similar authors for other authors occurring in the similar-authors-file are retrieved and those other authors are then also added to the authors-reference-list.
- The venue recommendations are retrieved and assigned to the output-files (`retrieveRecommendations()`)
- The `.txt`-recommendations-file is stored (`writeToNewFile()`)
- The `.js`-recommendations-file is stored (`writeToNewFile()`)