

Évaluation POO & Java

avril 2020
responsable : M.Munier

Projet - Références Bibliographiques

Remarque : Étant donné la période de confinement pour lutter contre l'épidémie de Covid-19, l'évaluation du module M2207 se fera cette année uniquement au travers d'un mini projet. La description des classes vous est fournie ; vous n'avez qu'à coder ces classes en Java.

★ ★ ★

1 Présentation du projet

Le thème de cet exercice est la conception d'une base de données bibliographique permettant de répertorier des publications de différents types. Chaque publication sera décrite par un mot de référence (la clé) ainsi qu'un certain nombre de champs (noms des auteurs, titre du document,...) qui dépendent du type de la publication. La classification standard des publications prévoit un nombre de catégories important s'étendant du plus simple article jusqu'aux actes de conférences. Voici quelques-unes de ces catégories :

- **article** : un article d'un journal ou d'une revue
Champs : `author`, `title`, `journal`, `year`, `volume`, `number`, `pages`, `month`, `note`
- **book** : un livre avec un éditeur explicite
Champs : `author`, `title`, `publisher`, `year`, `volume`, `series`, `address`, `edition`, `month`, `note`
- **inproceedings** : un article dans les actes d'une conférence
Champs : `author`, `title`, `booktitle`, `publisher`, `year`, `editor`, `pages`, `volume`, `series`, `organization`, `address`, `month`, `note`
- **phdthesis** : une thèse de doctorat
Champs : `author`, `title`, `school`, `year`, `type`, `address`, `month`, `note`

Afin d'éviter tout mouvement de panique parmi ceux d'entre vous qui ne parlent pas anglais et/ou qui sont dépourvus de toute notion bibliographique, voici une description de l'utilisation que l'on peut faire de ces différents champs :

- **address** : l'adresse de l'éditeur. Pour certaines grandes maisons d'édition, seule la ville est donnée
- **author** : le nom du ou des auteur(s)
- **booktitle** : le titre d'un livre, le nom de la partie citée
- **edition** : le numéro d'édition d'un livre
- **editor** : le ou les nom(s) du ou des rédacteur(s) ou éditeur(s)
- **journal** : le nom d'un journal
- **month** : le mois où l'ouvrage a été publié ou, pour un ouvrage non publié, celui où il a été achevé
- **note** : des informations supplémentaires pour aider le lecteur

- **number** : le numéro d'un journal, d'une revue ou d'un rapport technique
- **organization** : l'organisation ayant parrainé une conférence
- **pages** : un ou plusieurs numéros de pages ou intervalles de numéros (i.e. deux nombres séparés par un tiret court)
- **publisher** : le nom de la maison d'édition
- **school** : le nom de l'école où une thèse a été soutenue
- **series** : le nom d'une collection de livres
- **title** : le titre de l'ouvrage
- **type** : le type d'un rapport technique. Par exemple "Notes de recherche".
- **volume** : le numéro de volume d'un journal ou d'un livre en plusieurs parties
- **year** : l'année de publication ou, pour un ouvrage non publié, l'année où il a été achevé

2 Travail à réaliser

On se propose de spécifier puis d'implémenter les classes qui vont nous permettre de gérer une liste de références bibliographiques de ce style.

1. En ce qui concerne les **publications**, nous aurons bien évidemment la classe **Publication** (la classe mère de toutes les catégories de publications) avec :
 - les attributs communs tels que **reference**, **author**, **title** et **year**
 - les méthodes d'accès correspondantes
 - une méthode **afficher()** permettant... d'afficher le détail de cette publication
2. Nous aurons ensuite les différentes **sous-classes de publications** : **Article**, **Book**, **InProceedings** et **PhDThesis**.
 - ajouter les attributs nécessaires à chacune des sous-classes
 - ajouter les méthodes d'accès correspondantes
 - écrire les constructeurs adéquats (en faisant appel aux constructeurs de la classe mère)
 - redéfinir la méthode **afficher()**
3. Nous définirons ensuite la classe **Bibliography** qui contiendra une **liste de références bibliographiques** (i.e. un tableau de références vers des instances de **Publication**) et fournira un certain nombre de méthodes pour :
 - ajouter une référence bibliographique
 - afficher la liste de toutes les références bibliographiques
 - rechercher des références bibliographiques à partir d'un mot-clé, du nom d'un des auteurs, etc...
 - calculer le nombre de publications pour une année donnée

Un exemple de ce que pourrait donner l'exécution d'un programme de test élémentaire (que vous devrez écrire...) vous est donné, à titre indicatif, en annexe.
4. Écrire un **programme principal** permettant de gérer une bibliographie :
 - gestion d'un petit menu
 - saisie des informations au clavier
 - appel des méthodes sur les objets créés
 - affichage des résultats

3 Restitution de vos travaux

- Vous développerez toutes les classes de ce projet dans un seul et unique répertoire dont le nom est **votre nom de famille en minuscules**. Pour la restitution de votre travail, vous créez une archive (tar ou zip) de ce répertoire, le nom de cette archive étant lui aussi **votre nom de famille en minuscules**. Vous enverrez ensuite cette archive par mail à l'adresse manuel.munier@gmail.com.
- Il s'agit d'un travail individuel. Vous pouvez bien évidemment vous entraider, vous documenter sur Internet, etc. Mais en aucun cas je n'accepterai d'avoir le même projet en x exemplaires!
- La date limite pour rendre votre projet est fixée au **mardi 14 avril 2020 à 12h**.

★ ★ ★

Informations complémentaires :

- Afin de vous aider dans votre tâche, vous trouverez la documentation en ligne du Java SE 10 à l'URL suivante : <https://docs.oracle.com/javase/10/docs/api/index.html?overview-summary.html>.
- Pour mémoire, `String` est une classe. Pour comparer deux chaînes de caractères vous devrez donc utiliser la méthode `equals` et non pas faire une comparaison des références. Par exemple, si vous avez deux chaînes `chA` et `chB`, la comparaison se fait via l'instruction `chA.equals(chB)` qui retourne `true` ou `false`.
- De la même façon, pour savoir si une chaîne de caractères `chB` est contenue dans une chaîne `chA`, nous utiliserons l'instruction `chA.indexOf(chB)` qui retourne un entier indiquant la position (1^{er} caractère) de `chB` à l'intérieur de `chA`. Si la chaîne `chB` n'est pas trouvée dans `chA`, alors la méthode `indexOf` renvoie la valeur `-1`.

Bibliography testing program

*** Bibliography ***

PhDThesis

- Reference : munier99a
- Title : Une architecture pour intégrer des composants de contrôle de la coopération dans
- Authors : Manuel Munier
- Year : 1999
- Month : janvier
- Keywords : coopération, distribution, transaction, schéma de coopération, négociation
- School : Université Henry Poincaré - Nancy I, Loria
- Type : Thèse en Informatique

Article

- Reference : munier00a
- Title : DisCOO, a really distributed system for co-operation
- Authors : M. Munier and K. Benali and C. Godart
- Year : 2000
- Journal : Networking and Information Systems Journal

InProceedings

- Reference : gallon98b
- Title : Analyse Qualitative et Quantitative basées sur des Réseaux de Petri Temporisés S
- Authors : G. Juanole and L. Gallon
- Year : 1998
- Month : juillet
- Keywords : STPN, qualitative analysis, quantitative analysis
- BookTitle : MOVEP'98, Ecole d'été Informatique et Automatique
- Address : Nantes, France

InProceedings

- Reference : munier98a
- Title : Cooperation services for widely distributed applications
- Authors : M. Munier and C. Godart
- Year : 1998
- Month : juin
- BookTitle : Tenth International Conference on Software Engineering and Knowledge Engineering
- Address : San Francisco Bay, USA

Nombre de publications par année depuis 1997:

- 1997 : 0 publication
- 1998 : 2 publications
- 1999 : 1 publication
- 2000 : 1 publication

Références des publications de "munier": munier99a, munier00a, munier98a

Références des publications de "gallon": gallon98b