## *Java SE 8 Programming Language*

# Training Assignments

| | |
|---|---|
| **Document Code** | **25e-BM/HR/HDCV/FSOFT** |
| **Version** | **1.1** |
| **Effective Date** | **20/11/2012** |

**Hanoi, 04/2019**

## RECORD OF CHANGES

| No | Effective Date | Change Description | Reason | Reviewer | Approver |
|---|---|---|---|---|---|
| 1. | 01/Mar/2019 | Add the new assignment | Create new | DieuNT1 | VinhNV |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

# Contents

Long Assignment 1 – Option 1: Advanced OOP, Generic and Collection, Stream and Filter ...........4

Objectives: ...........4

Working Environments ...........4

Assignment Descriptions ...........4

Validation Rules ...........5

Functional Requirements ...........5

User Interface Requirements ...........5

| | |
|---|---|
| CODE: | JPL.L.A101 |
| TYPE: | LONG |
| LOC: | |
| DURATION: | 180 MINUTES |

# Long Assignment 1 – Option 1: Advanced OOP, Generic and Collection, Stream and Filter

## Objectives:

After finishing the following exercises, trainees will:

- Understand and practice with Classes, Object, Inheritance, Encapsulation, Abstraction and Polymorphism.
- Understand and practice with Control-of-flow statements.
- Understand and practice with method overloading, method overriding.
- Understand and practice with Java Collection, Streams and Filters.

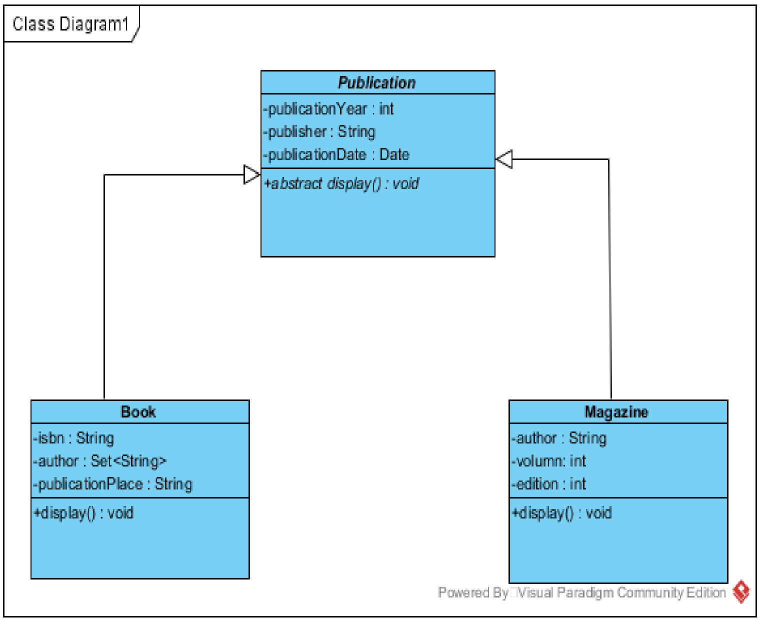
## Working Environments

- JDK 1.8
- Eclipse

## Assignment Descriptions

Create a Java consoled based application to manage publications in a library. In the library, there are two kinds of publication: book and magazine.

Refer to the class hierarchy which is decribed below, trainees need to create Java classes in order to implement the entities and the relationship between them.



**Publication** is an abstract superclass that contains the general information of book and magazine which will be extended. It includes *publication year*, *publisher and publication date*.

**Book** inherits from Publication class and has more attributes: *isbn* (international standard book number), *author* (a set of author name*)*, *publication place*.

**Magazine** is similar to Book class. It inherits from Publication class and has more attributes: *author* (author name of magazine), *volume* and *edition*.

All of these classes must have getter/setter and constructor if needed.

You must override **display()** method in Book/Magazine class to print all of the information related to Book/Magazine.

## Validation Rules

- The *isbn* is a unique numeric commercial book identifier. The length of ‘isbn’ is in the 10-17 digit number and the ‘-‘ quote. Ex: 678-3-16-1486.

## Functional Requirements

1. The program must have a function to add a new book.
2. The program must have a function to add a new magazine.
3. The program must have a function to display the list of all books and magazines that have the same publication year and publisher.
4. The program must have a function to add an author to a specific book, if the author existed, the program should print a message “Author existed”, otherwise print “Add successfully”.
5. The program must have a function to display the list of top 10 magazines which have the largest volume.
6. The program must provide functions to:
   - search book by isbn
   - search book by author
   - search book by publisher

   Search results should be sorted by isbn, publication date.

## User Interface Requirements

Create a new class named **LibraryManagement** that contains a main() method to display user interface.

The main screen allows user to select the following functions:

Menu

```
====== LIBRARY MANAGEMENT SYSTEM ======
1.  Add a book
2.  Add a magazine
3.  Display books and magazines
4.  Add author to book
5.  Display top 10 of magazines by volume
6.  Search book by (isbn, author, publisher)

Please choose function you'd like to do:
```

### Storage Data

- The user inputs data from the keyboard.
- Data is stored in the collection (List, Set…).
- Output data is displayed on the console.

**Guildelines**:

- Create a project named **JPL.L.A101**, create package ***fa.training.entities*** that contains classes/interfaces: **Publication**, **Book**, **Magazine**.
- Create other package ***fa.training.services*** that contains classes to implement Functional Requirements. The package ***fa.training.utils*** to implement Validation Rules requirements.

  *Note, the functional requirements related to any entity, you have to create the service class corresponding to that entity. Ex: BookService, …*

- Create a package named ***fa.training.main*** contains **LibraryManagement** class.

**-- THE END --**