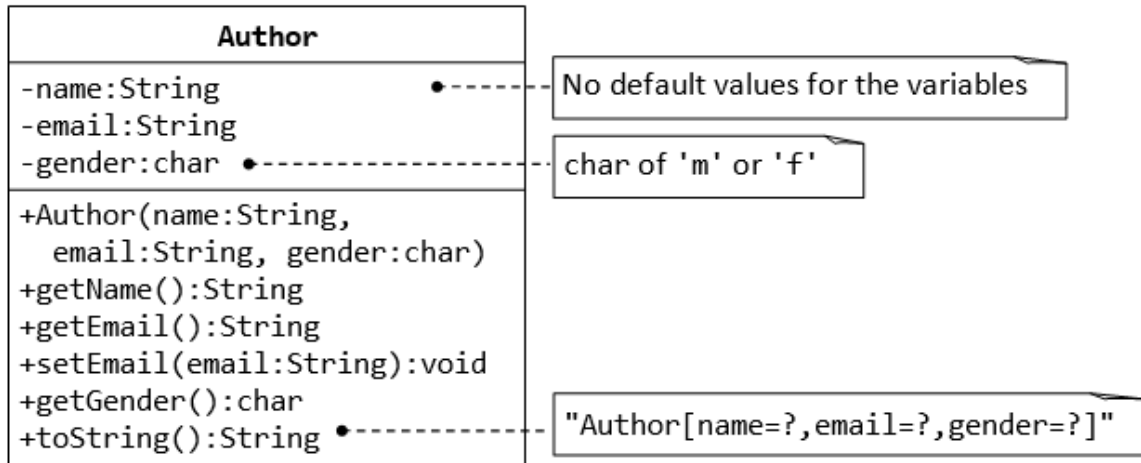


# Assignment 1:

## Exercise 1: The Author and Book Classes

(Author:40 points, TestAuthor: 20 points, Book:50 points, TestBook :20 points )

This first exercise shall lead you through all the concepts involved in OOP Composition.



A class called Author (as shown in the class diagram) is designed to model a book's author. It contains:

- Three private instance variables: `name` (String), `email` (String), and `gender` (char of either 'm' or 'f');
- One constructor to initialize the name, email and gender with the given values;

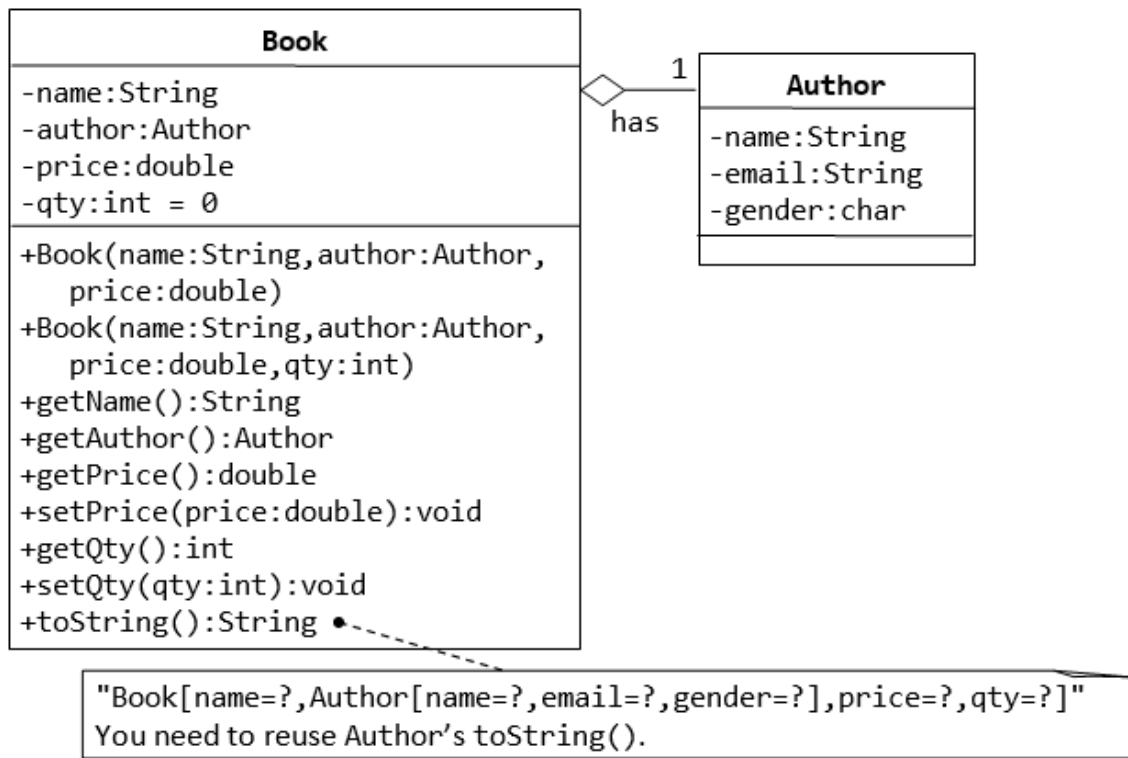
(There is no default constructor for Author, as there are no defaults for name, email and gender.)

- public getters/setters: `getName()`, `getEmail()`, `setEmail()`, and `getGender()`; (There are no setters for name and gender, as these attributes cannot be changed.)
- A `toString()` method that returns "Author[name=?,email=?,gender=?]", g., "Author[name=Tan Ah Teck,email=[ahTeck@somewhere.com](mailto:ahTeck@somewhere.com),gender=m];".

Write the Author class. Also write a *test driver* called `TestAuthor` to test all the public methods, e.g.,

```
Author ahTeck = new Author("Tan Ah Teck", "ahteck@nowhere.com", 'm'); // Test
the constructor

System.out.println("email is: " + ahTeck.getEmail()); // Test getter
```



A class called Book is designed (as shown in the class diagram) to model a book written by *one* author. It contains:

- Four private instance variables: name (String), author (of the class Author you have just created, assume that a book has one and only one author), price (double), and qty (int);
- Two constructors:

```
public Book (String name, Author author, double price) { ..... }
```

```
public Book (String name, Author author, double price, int qty) { ..... }
```

- public methods `getName()`, `getAuthor()`, `getPrice()`, `setPrice()`, `getQty()`, `setQty()`.
- `getAuthorName()`, `getAuthorEmail()`, `getAuthorGender()` in the Book class to return the name, email and gender of the author of the book. For example

```
public String getAuthorName() {
    return author.getName();
    // cannot use author.name as name is private in Author class
}
```

- A `toString()` that returns `"Book[name=?, Author[name=?, email=?, gender=?], price=?, qty=?]"`. You should reuse Author's `toString()`.

Write the Book class (which uses the Author class written earlier). Also write a test driver called `TestBook`, It contains:

1. `main()` method to test all the public methods in the class Book. Take Note that you have to construct an instance of Author before you can construct an instance of Book.
2. Printing the name and email of the author from a Book instance. (Hint: `getAuthor().getName()`, `getAuthor().getEmail()`).

**Requirements:** class `Book` and `Author` are in different package.

## Exercise 2: Date Class (50 points)

---

1. Define a Date class with three member variables of year, month and day, and the following functions:

1. define a constructor with parameters to initialize the member variable.
2. define a `public void setData (int year, int month, int day)` method, change the value of the member variables, The three parameters of the method represent year, month and day.

Notice: Reject changing three member variables to illegal values, if wrong value is passed, display an error message.

3. define a `public void addOneDay ()` method to add one day to the original date.
4. define a `public void display ()` method to display the date in the format of day / month / year.

### Requirements:

- **Initialize date, set date and date plus one day to ensure that date change is valid(some months are 30 or 31 days, leap year is 28 or 29 days)**
2. Create a `TestDate` class that contains a `main ()` method, Method:
1. create a Date class object .
  2. correctly modify the date and plus one day to the date and display the running results
  3. modify the date with wrong data and display error message.
  4. run and display the results for the last day of the month and the last day of the year plus one day.

## Exercise 3: String and StringBuffer (10 points)

---

Write a Java program to test if a given string is a symmetric string.

for example:

"asdfgasdf" is not a symmetric string.

"asdfgfdsa" is a symmetric string.

Tip: implement with the methods provided by String and StringBuffer.