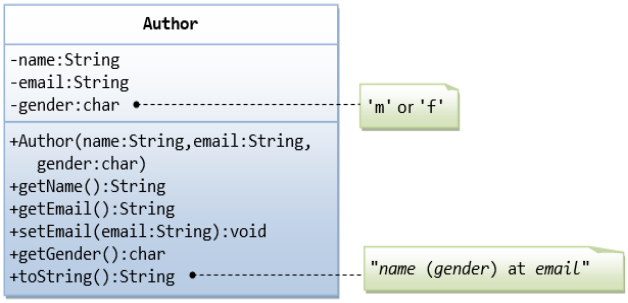
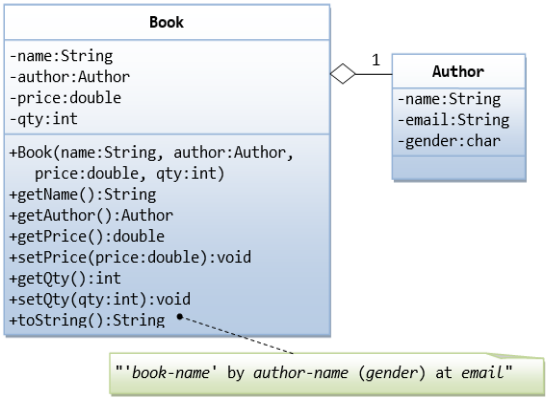
Let's start with the Author class



A class called `Author` is designed as shown in the class diagram. It contains:

- Three `private` member variables: `name` (`String`), `email` (`String`), and `gender` (`char` of either `'m'` or `'f'` - you might also use a `boolean` variable called `isMale` having value of `true` or `false`).
- A constructor to initialize the `name`, `email` and `gender` with the given values.
  (There is no *default constructor*, as there is no default value for `name`, `email` and `gender`.)
- Public getters/setters: `getName()`, `getEmail()`, `setEmail()`, and `getGender()`.
  (There are no setters for `name` and `gender`, as these properties are not designed to be changed.)
- A `toString()` method that returns "*`name (gender) at email`*", e.g., "`Tan Ah Teck (m) at ahTeck@somewhere.com`".

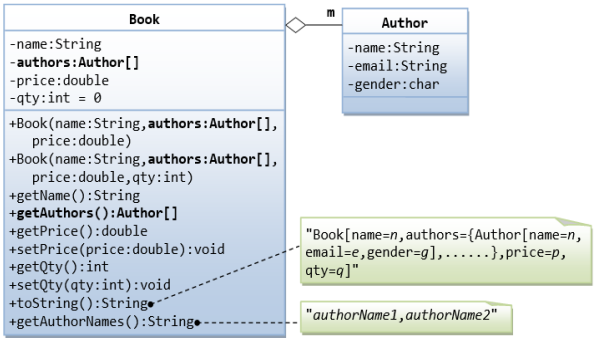
A Book is written by one Author - Using an "Object" Member Variable



Let's design a Book class. Assume that a book is written by one (and exactly one) author. The Book class (as shown in the class diagram) contains the following members:

- Four `private` member variables: `name` (`String`), `author` (an *instance* of the `Author` class we have just created, assuming that each book has exactly one author), `price` (`double`), and `qty` (`int`).
- The `public` getters and setters: `getName()`, `getAuthor()`, `getPrice()`, `setPrice()`, `getQty()`, `setQty()`.
- A `toString()` that returns "`'book-name' by author-name (gender) at email`". You could reuse the `Author`'s `toString()` method, which returns "`author-name (gender) at email`".

2.2 (Advanced) The Author and Book Classes Again - An Array of Objects as an Instance Variable



In the earlier exercise, a book is written by one and only one author. In reality, a book can be written by one or more author. Modify the Book class to support one or more authors by changing the instance variable `authors` to an `Author` array.

Notes:

- The constructors take an array of `Author` (i.e., `Author[]`), instead of an `Author` instance. In this design, once a Book instance is constructor, you cannot add or remove author.
- The `toString()` method shall return "`Book[name=?,authors={Author[name=?,email=?,gender=?],.....},price=?,qty=?]`".

You are required to:

1. Write the code for the Book class. You shall re-use the `Author` class written earlier.
2. Write a test driver (called `TestBook`) to test the Book class.