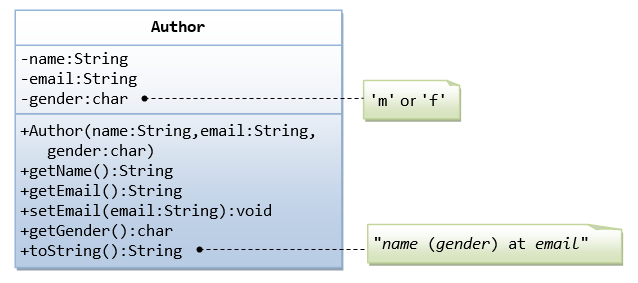
1. **The Author Class**

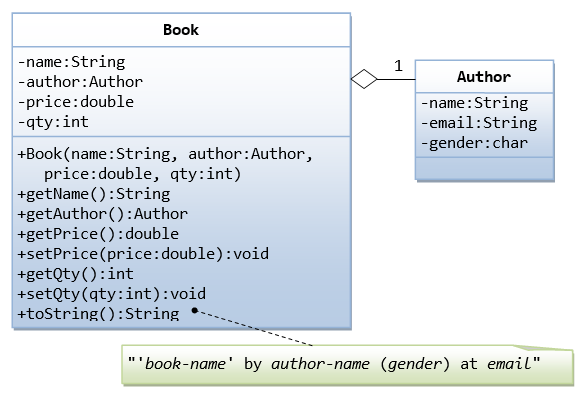
**Create a class called Author by following the pic shown below.**



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'.
* 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., "Prahaash (Male) at [prahaash@outlook.com](mailto:prahaash@outlook.com)".

1. **Create Book class by seeing the following pic**



**Note than now object of Author class is part of Book class.**

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".

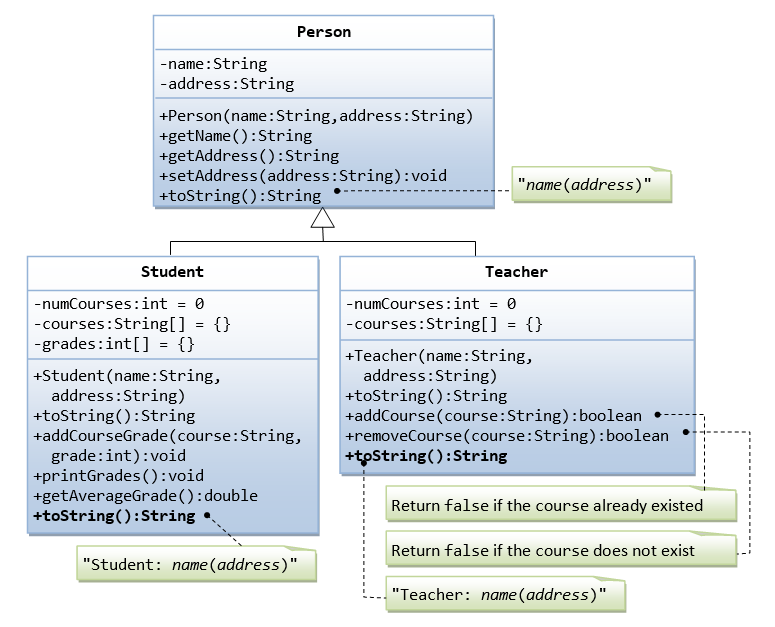
#### **Superclass Person and its Subclasses**

Suppose that we are required to model students and teachers in our application. We can define a superclass called Person to store common properties such as name and address, and subclasses Student and Teacher for their specific properties.

For students, we need to maintain the courses taken and their respective grades; add a course with grade, print all courses taken and the average grade. Assume that a student takes no more than 30 courses for the entire program.

For teachers, we need to maintain the courses taught currently, and able to add or remove a course taught. Assume that a teacher teaches not more than 5 courses concurrently.

Define static fields maximum courses in the two derived classes.



### **Design of the AccountKinds Hierarchy**

In the following section, we will discuss steps used in designing a hierarchy of classes.

#### The Problem Description

Suppose we have been given the task of creating a program that will keep track of all the accounts for a bank. There are a number of different kinds of accounts that the bank supports.

* **Regular Account** - This account charges a fee of which is the smaller of 10 or 10% of the balance at the end of the month. There is no interest. There is a penalty of 10.00 if the balance falls below a minimum of 500.00.
* **Interest Account** - This account charges a fee of which is the smaller of 10 or 10% of the balance at the end of the month. There is interest of 7% paid monthly. There is no minimum balance required.
* **Checking Account** - This account charges a fee of which is the smaller of 10 or 10% of the balance at the end of the month. There is annual interest of 7% paid monthly . There is a penalty of 10.00 if the balance falls below a minimum of 100.00. There is a charge of 0.10 for each transaction.
* **CD Account** - This account charges a fee of which is the smaller of 10 or 10% of the balance at the end of the month. There is interest of 15% paid yearly. There is no minimum balance required, but if there is a withdrawal before 12 months have gone by there will be a penalty of 20% of the current balance.

**(**CD means **certificate of deposit**. What is a certificate of deposit? The definition of certificate of deposit is an account that allows you to save money typically at a fixed interest rate for a fixed amount of time—say, 6 months, 1 year or 5 years.**)**

Each of these accounts has a personal identification number (PIN) with it to provide protection.

#### Attributes and Methods

Our first task is to identify the attributes and methods that each of the classes will need. At first look, we can identify the following attributes for each of the classes:

|  |  |  |
| --- | --- | --- |
| Attributes for Regular Account | Type | Description |
| name | String | the name of the account holder |
| balance | double | balance in the account |
| pin | String | personal identification number |
| minimum balance | double | minimum balance for the account |
| penalty | double | penalty if balance falls below the minimum balance |

|  |  |  |
| --- | --- | --- |
| Attributes for Interest Account | Type | Description |
| name | String | the name of the account holder |
| balance | double | balance in the account |
| pin | String | personal identification number |
| interest | double | yearly interest |

|  |  |  |
| --- | --- | --- |
| Attributes for Checking Account | Type | Description |
| name | String | the name of the account holder |
| balance | double | balance in the account |
| pin | String | personal identification number |
| interest | double | yearly interest |
| minimum balance | double | minimum balance for the account |
| penalty | double | penalty if balance falls below the minimum balance |
| transactions | int | the number of deposits and withdrawals in a month |

|  |  |  |
| --- | --- | --- |
| Attributes for CD Account | Type | Description |
| name | String | the name of the account holder |
| balance | double | balance in the account |
| pin | String | personal identification number |
| interest | double | yearly interest |
| penalty | double | penalty if early withdrawal |
| months | int | number of months since the creation of the account |

All of these classes need to have basically the same methods

* Create the account.
* Deposit an amount.
* Withdraw an amount.
* Access the balance.
* Access the name.
* Check the validity of the PIN.
* Compute the fees.
* Compute the interest.