

Atelier 2 : Déclaration et la Surcharge des constructeurs & les méthodes statiques

Codez les classes suivantes en respectant les annotations UML. Ecrivez un code qui permet de tester les méthodes de chaque classe;

1- La classe Circle ;

Circle
-radius:double = 1.0 -color:String = "red"
+Circle() +Circle(radius:double) +Circle(radius:double,color:String) +getRadius():double +getColor():String +setRadius(radius:double):void +setColor(color:String):void +toString():String +getArea():double

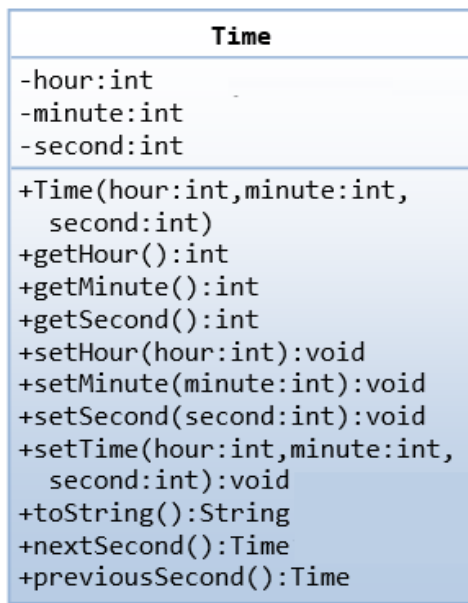
2- La classe Rectangle ;

Rectangle
-length:float = 1.0f -width:float = 1.0f
+Rectangle() +Rectangle(length:float,width:float) +getLength():float +setLength(length:float):void +getWidth():float +setWidth(width:float):void +getArea():double +getPerimeter():double +toString():String

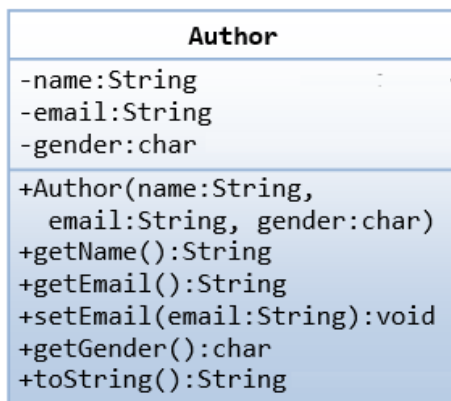
3- La classe Employee ;

Employee
-id:int -firstName:String -lastName:String -salary:int
+Employee(id:int,firstName:String,lastName:String,salary:int) +getID():int +getFirstName():String +getLastName():String +getName():String +getSalary():int +setSalary(salary:int):void +getAnnualSalary():int +raiseSalary(int percent):int +toString():String

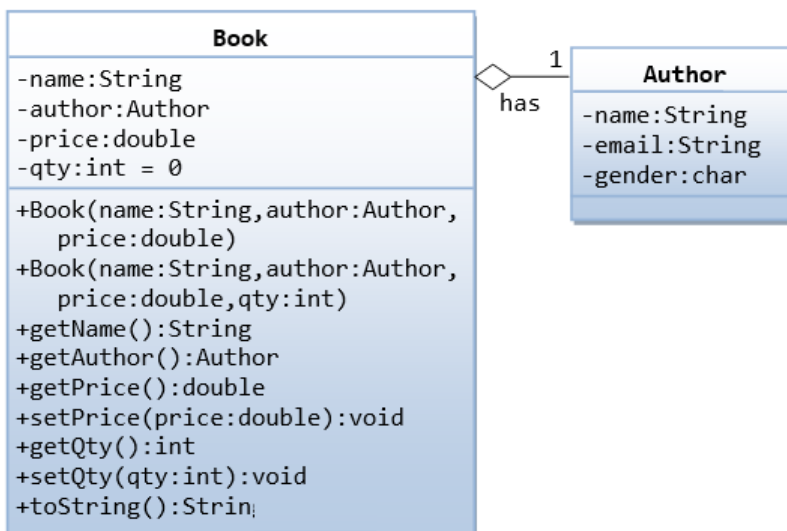
4- La classe **Time** ;



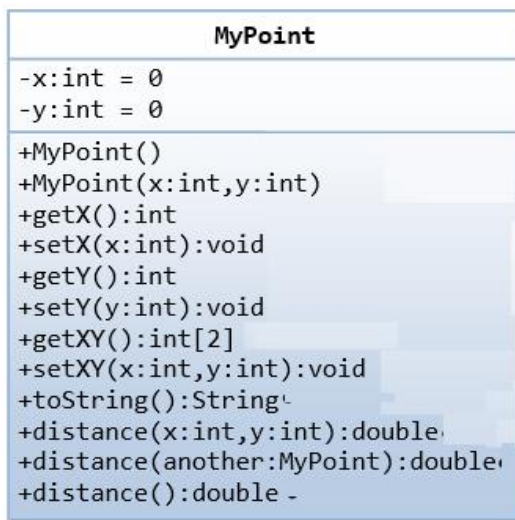
5- la classe **Author** ;



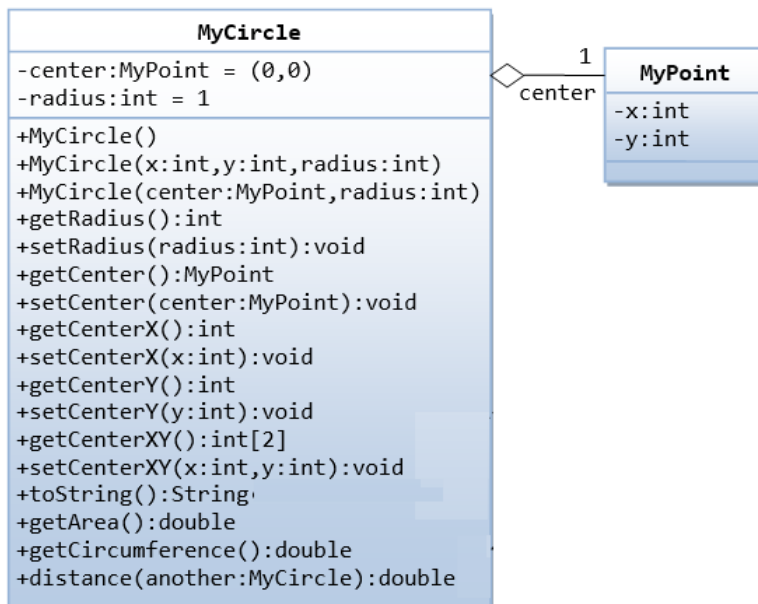
6- La classe **Book** ;



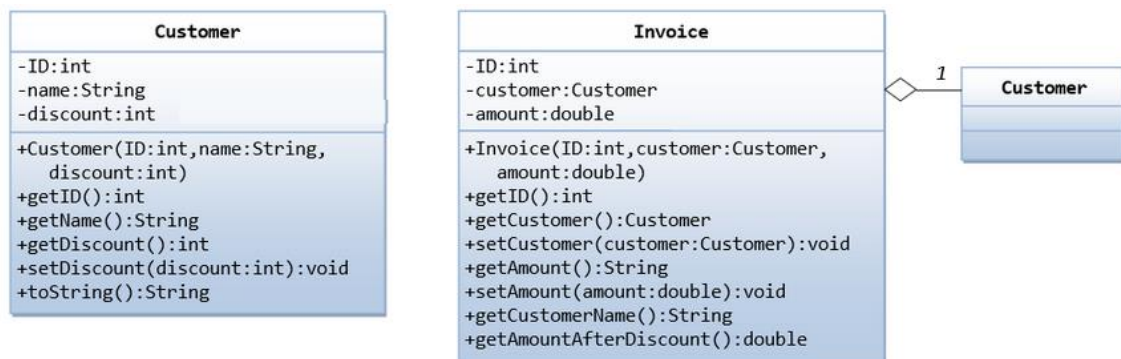
7- La classe **MyPoint** ;



8- La classe **MyCircle**



9- Les deux classe **Customer** et **Invoice**



Exercice 1 :

Un site internet est spécialisé dans la vente des livres, un livre est défini par son titre, son auteur, son prix et son nombre de pages.

- Déclarez la classe **Livre** ;
- On veut afficher les informations d'un livre.
- On veut savoir le nombre total des livres créés.
- On veut savoir le prix max et min de tous les livres ;
- Définissez une liste des livres que vous remplirez de votre choix.
- Afficher la liste des livres.

Exercice 2 :

- 1- Écrivez une méthode statique `max3()` qui prend trois valeurs de type **int** comme arguments et renvoie la plus grande valeur. Ajoutez une fonction surchargée qui fait la même chose avec trois valeurs de types doubles.
- 2- Écrivez une méthode statique `impr()` qui prend trois entrées booléennes et renvoie `true` si un nombre impair d'entrées est vrai, et `false` sinon.
- 3- Écrivez une méthode statique `equi()` qui prend deux tableaux d'entiers comme arguments et renvoie vrai s'ils contiennent le même nombre d'éléments et si toutes les paires d'éléments correspondantes sont égales.