Implantation
IESN

Langage de programmation O. O.

Interro formative — 2024

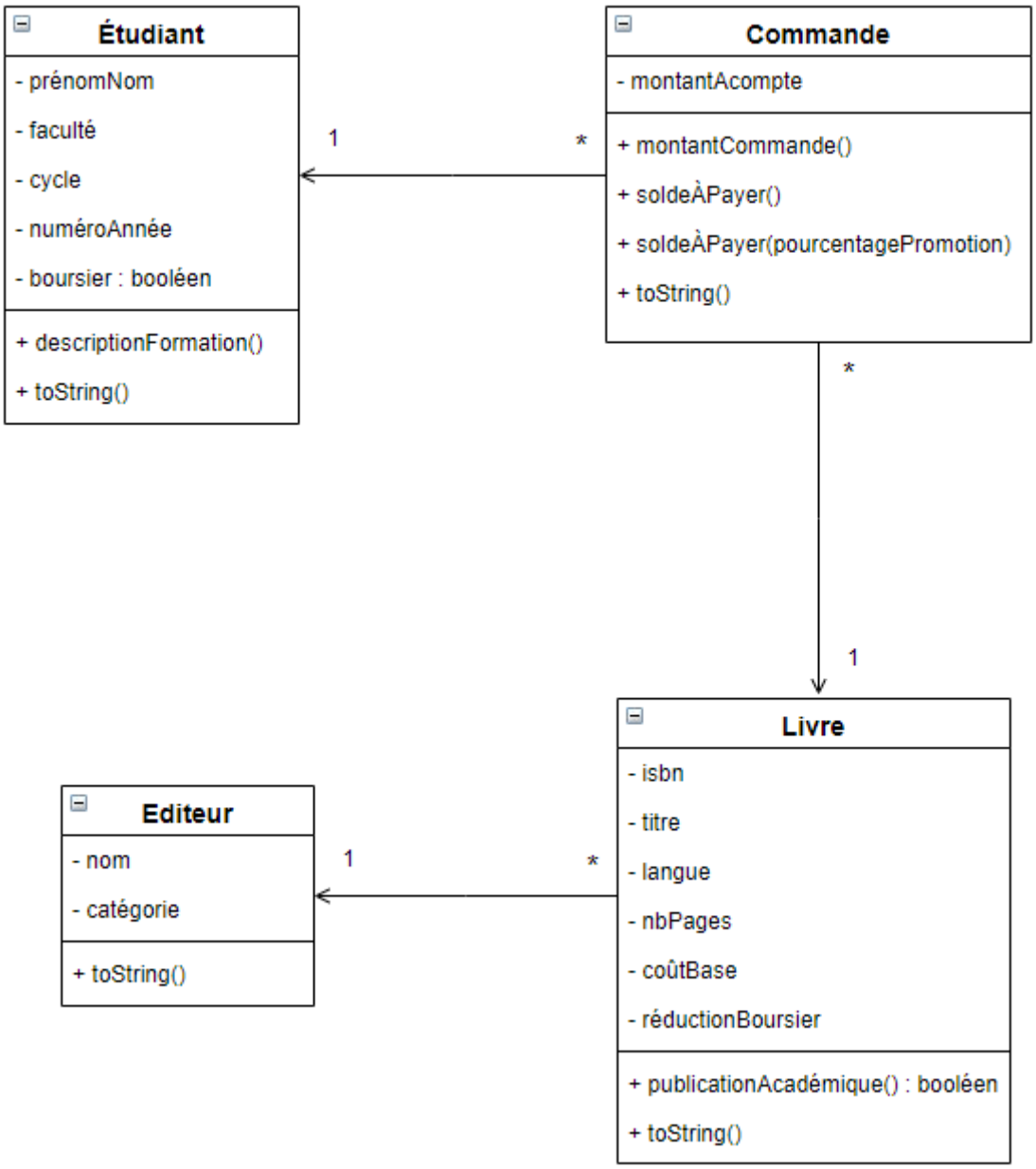
Consignes

- Toutes les **variables d'instance** doivent être déclarées avec une **visibilité privée**.
- Les règles du "**clean code**" s'appliquent !
- Ne créez **que les getters/setters qui sont nécessaires** à ce qui est demandé dans l'exercice, c'est-à-dire afin que la méthode *main* de la classe *Principale* puisse s'exécuter (pas de création automatisée sans réflexion).

Question 1

Vous devez gérer une librairie universitaire qui permet aux étudiants de commander des livres. Un acompte peut être demandé lors de l'établissement de la commande. Les étudiants boursiers bénéficient d'une réduction sur les livres qu'ils commandent.

Le diagramme UML suivant résume la situation.



- Classe **Etudiant**

  Un étudiant est caractérisé par un prénom suivi du nom, la faculté où il est inscrit, le cycle d'années d'études (bac, master ou doctorat) et le numéro de l'année d'études (1, 2 ou 3). On précise si l'étudiant bénéficie ou non d'une bourse. Complétez, s'il y a lieu, les variables d'instance en fonction du diagramme de classe.

  Le seul numéro d'année admis est 1, 2 ou 3. Cette contrainte est valable aussi bien à la création d'un étudiant que si on tente de modifier le numéro d'année d'un étudiant existant. Si on tente d'affecter une valeur < 1, le numéro d'année sera mis à 1 ; si on tente d'affecter une valeur > 3, le numéro d'année sera mis à 3.

  Prévoyez un constructeur permettant d'initialiser toutes les variables d'instance de la classe.

  Prévoyez un second constructeur facilitant la création d'étudiants non boursiers.

  Prévoyez les méthodes suivantes :

  - La méthode *descriptionFormation* qui retourne la chaîne de caractères constituée du cycle suivi de l'année suivi de la faculté, par exemple *: master 1 en médecine.*
  - La méthode *toString* qui retourne la chaîne de caractères décrivant un étudiant en respectant la structure proposée à travers l'exemple suivant :
    Anne Petit inscrit(e) en bac 3 en informatique

  N.B. Ajoutez « *bénéficiant d'une bourse* » si l'étudiant bénéficie d'une bourse
  Ex : Julien Lefort inscrit(e) en bac 2 en médecine bénéficiant d'une bourse

Dans la méthode *main* de la classe *Principale*, écrivez les instructions qui répondent aux demandes ci-dessous.

- Créez un étudiant boursier appelé *premierEtudiant* inscrit en chimie.
- Affichez à la console la description de cet étudiant.
- Modifiez la faculté de cet étudiant : il est désormais inscrit en physique et réaffichez la description de cet étudiant. N.B. vous ne devez pas créer un nouvel objet de type *Etudiant*, mais bien modifier l'objet *premierEtudiant* créé précédemment.
- En utilisant le constructeur le plus adéquat, instanciez ensuite un étudiant **non boursier** appelé *secondEtudiant* et affichez à la console la description de ce second étudiant.

- Classe **Editeur**

  Un éditeur ou maison d'édition est caractérisé par un nom et une catégorie. Complétez, s'il y a lieu, les variables d'instance en fonction du diagramme de classe.

  Prévoyez un constructeur permettant d'initialiser toutes les variables d'instance de la classe.

  - Prévoyez la méthode toString qui retourne la chaîne de caractères décrivant un éditeur en respectant la structure proposée à travers l'exemple suivant :
    la maison d'édition Blue Bird (catégorie : universitaire)

  Dans la méthode *main* de la classe *Principale*, écrivez les instructions qui répondent aux demandes ci-dessous.

  - Instanciez un objet de type Editeur de la catégorie "Nouvelles tendances".
  - Affichez à la console la description de cet éditeur.

- Classe **Livre**

  Un livre est caractérisé par un numéro ISBN, un titre, la langue dans laquelle il est écrit, le nombre de pages, le coût de base et la réduction éventuelle pour les étudiants boursiers (un montant en euros). Complétez, s'il y a lieu, les variables d'instance en fonction du diagramme de classe.

  Faites en sorte que le nombre de pages soit toujours >= à 1. Cette contrainte est valable aussi bien à la création d'un livre que si on tente de modifier le nombre de pages d'un livre existant. Si on tente d'affecter un nombre de pages < 1 et que la variable d'instance contient 0, affectez la valeur 1, sinon, la valeur de la variable d'instance reste inchangée.

  Prévoyez un constructeur permettant d'initialiser toutes les variables d'instance de la classe.

  Prévoyez un second constructeur facilitant la création de livres écrits en français.

  Prévoyez les méthodes suivantes :

  - La méthode *publicationAcademique* qui retourne vrai si la catégorie de l'éditeur est "universitaire" et retourne faux sinon.
  - La méthode *toString* qui retourne la chaîne de caractères décrivant un livre en respectant la structure proposée à travers l'exemple suivant :
    le livre de 986 pages intitulé Mamma mia
    (isbn : 120-29-66-85697-8) édité en italien
    par la maison d'édition Gauthier (catégorie : Découverte)

Dans la méthode *main* de la classe *Principale*, écrivez les instructions qui répondent aux demandes ci-dessous.

- Instanciez un objet de la classe Livre appelé *premierLivre* correspondant à un livre écrit en anglais.
- Affichez à la console la description de ce premier livre.
- Si ce livre est une publication académique, affichez à la console "Ce livre est une publication académique" sinon affichez "Ce livre n'est pas une publication académique".
- En utilisant le constructeur le plus adéquat, instanciez un second objet de la classe Livre appelé *secondLivre* correspondant à un livre écrit en **français**.
- Affichez à la console la description de ce second livre.
- Modifiez ensuite l'éditeur de l'objet *secondLivre* et réaffichez à la console la description de *secondLivre*. N.B. vous ne devez pas créer un nouvel objet de type *Livre*, mais bien modifier l'objet *secondLivre* créé précédemment.

- Classe **Commande**

Une commande est caractérisée par le montant de l'acompte donné par l'étudiant lors de la commande (le montant de l'acompte peut être égal à 0). Complétez, s'il y a lieu, les variables d'instance en fonction du diagramme de classe.

Prévoyez un constructeur permettant d'initialiser toutes les variables d'instance de la classe.

Prévoyez les méthodes suivantes :

- La méthode *montantCommande* qui retourne le prix de la commande ; celui-ci est égal au prix de base du livre commandé duquel on soustrait la réduction pour boursier si l'étudiant qui a effectué la commande bénéficie d'une bourse.
- La méthode *soldeAPayer()* qui retourne le montant qu'il reste à payer sur base du montant de la commande et de l'acompte payé par l'étudiant.
- Surchargez la méthode *soldeAPayer* : celle-ci reçoit en argument un pourcentage de promotion à appliquer sur le solde restant de la commande.
- La méthode *toString* qui retourne la chaîne de caractères décrivant une commande en respectant la structure proposée à travers l'exemple suivant :

Patrick Leben inscrit(e) en bac 2 en pharmacie bénéficiant d'une bourse a commandé
le livre de 104 pages intitulé The small lion
(isbn : 189-99-06-45697-4) édité en anglais
par la maison d'édition Gai lecteur (catégorie : Nouvelles tendances)
avec un acompte de 10.0 euros

Si aucun acompte n'a été donné, supprimez la dernière ligne.

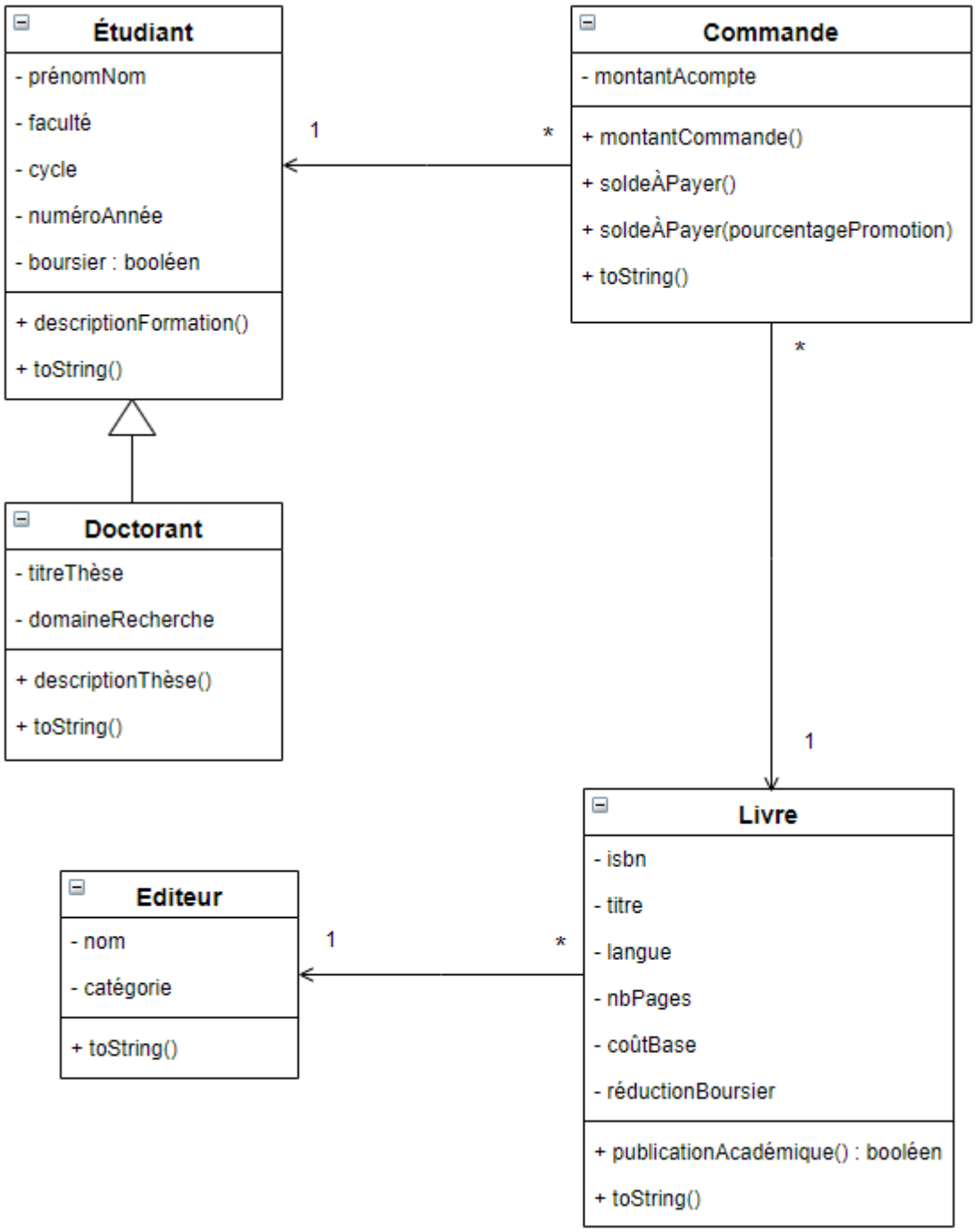
Dans la méthode *main* de la classe *Principale*, écrivez les instructions qui répondent aux demandes ci-dessous.

- Instanciez un objet de la classe *Commande* appelé *commandeEnCours*.
- Affichez à la console le montant de cette commande.
- Affichez à la console le solde restant à payer (sans promotion) de l'objet *commandeEnCours*.
- Affichez à la console le solde restant à payer de l'objet *commandeEnCours* si on applique un pourcentage de promotion de 20%.
- Affichez à la console la description de l'objet *commandeEnCours*.
- Écrivez les instructions suivantes permettant d'afficher à la console des informations à partir de l'objet *commandeEnCours* ; attention, chaque instruction doit impérativement débuter chaque fois par : **System.out.println(commandeEnCours.**
    - Affichez la description de l'étudiant qui a passé la commande *commandeEnCours* ;
    - Affichez le titre du livre de la commande *commandeEnCours*
    - Affichez la catégorie de l'éditeur du livre de la commande *commandeEnCours*.

Question 2

On vous demande d'adapter le programme afin de gérer une catégorie particulière d'étudiants, à savoir, les doctorants.

Le diagramme UML suivant tient compte de cette adaptation.



- Classe **Doctorant**

  Un doctorant **est un étudiant** caractérisé en outre par le titre de sa thèse et le domaine de recherche.

  Prévoyez un constructeur permettant d'initialiser toutes les variables d'instance de la classe. *N.B. le cycle d'études d'un doctorant est forcément "doctorat".*

  Sur tout objet de type *Doctorant*, on doit pouvoir appeler les méthodes suivantes :

  - La méthode *descriptionFormation* qui retourne la chaîne de caractères constituée du cycle suivi de l'année suivi de la faculté ;
  - La méthode *descriptionThese* qui retourne la chaîne de caractères constituée sur base du titre de la thèse, du prénom suivi du nom du doctorant et du domaine de recherche ;

  Exemple :

  Titre de la thèse : Les techniques de défense des châteaux forts
  Auteur de la thèse : Arthur Leroy
  Domaine de recherche : Histoire médiévale

  - La méthode *toString* qui retourne la chaîne de caractères décrivant un doctorant ; basez-vous sur l'exemple ci-dessous afin de déterminer le format :

  Exemple :
  Marie Larritie inscrit(e) en doc 2 en informatique bénéficiant d'une bourse
  Doctorant(e) en intelligence artificielle

Dans la méthode *main* de la classe *Principale*, écrivez les instructions qui répondent aux demandes ci-dessous.

- Instanciez un objet de type *Doctorant* appelé *doc*.
- Modifiez la faculté du doctorant *doc*.
- Affichez à la console le numéro d'année d'études de *doc*.
- Affichez à la console la description de la formation de *doc*.
- Affichez à la console la description de la thèse de *doc*.
- Affichez à la console la description de *doc*.