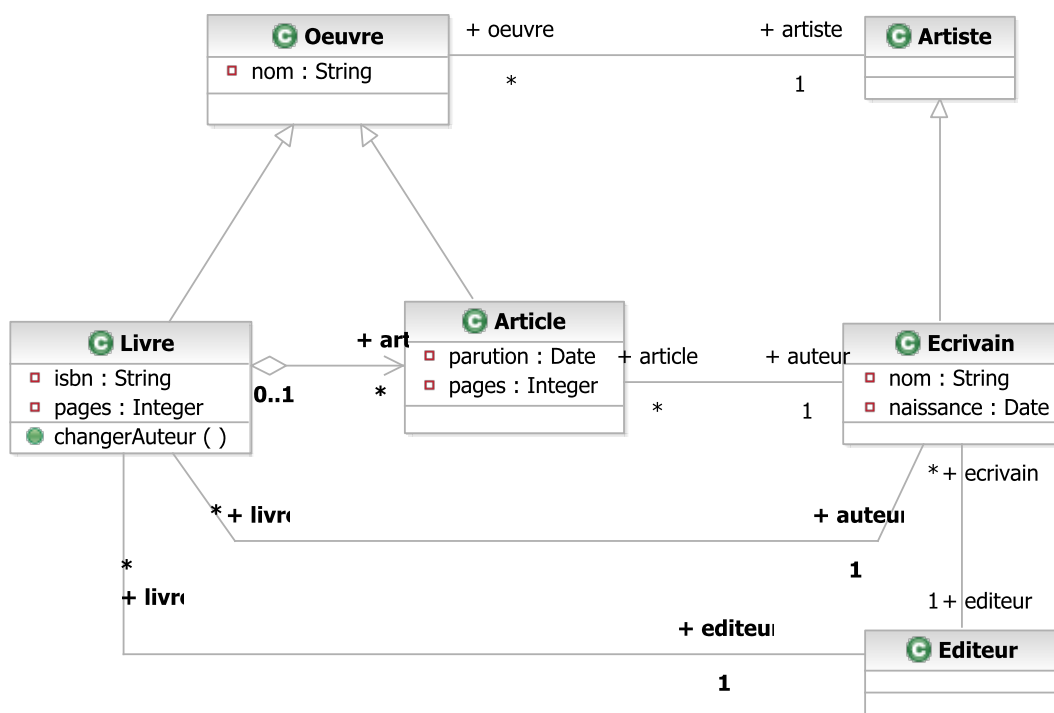


**Contrôle continu – Première session**  
– Documents autorisés –

Répondez aux questions sur cette feuille. Si vous n'avez pas assez de place pour finir votre réponse, utilisez une feuille additionnelle.  
Ce contrôle a 2 questions, pour un total de 20 points.

Question:	1	2	Total
Points:	8	12	20
Score:			

1. Lire attentivement le diagramme suivant et spécifier en OCL les invariants et les opérations décrits ci-dessous:



- (a) Utilisez OCL pour spécifier les invariants suivants:

(1½)

- i. Les oeuvres d'un écrivain font partie soit de ses livres, soit de ses articles.

**Solution:**

```

context Ecrivain
inv:
    self.oeuvres = self.livres → union(self.articles)
    
```

(1½)

- ii. La date de parution d'un article ne peut pas être antérieure à la date de naissance de son auteur.

**Solution:**

```
context Article
inv:
    self.parution > self.auteur.naissance
```

(1½)

iii. L'éditeur d'un livre est le même que celui de l'écrivain de ce même livre.

**Solution:**

```
context Livre
inv:
    self.editeur = self.auteur.editeur
```

(1½)

iv. Lorsqu'un livre est composé d'articles, son nombre de pages est supérieur à celui de la somme des pages de ses articles.

**Solution:**

```
context Livre
inv:
    self.pages > self.art→collect(pages)→sum()
```

(b) Spécifier en OCL la signature et le comportement de l'opération suivante:

(2)

i. L'opération `changerAuteur(e:Ecrivain)` change l'auteur d'un livre. Lorsqu'un livre est composé d'articles, cette opération change aussi l'auteur des articles dont l'auteur est le même que celui du livre.

**Solution:**

```
context Livre::changerAuteur(e : Ecrivain)
post:
    self.auteur = e and self.art@pre→forAll(each : Article |
        each.auteur@pre = self.auteur@pre implies each.auteur = self.auteur)
```

2. Analyse et conception

Il s'agit ici de modéliser le domaine des agendas. Un agenda est composé d'un ensemble d'événements. Chaque événement possède un titre, une heure et une date de début et une durée. Il existe deux types d'événement, les événements simples et les événements récurrents.

Un événement récurrent peut se répéter tous les jours, toutes les semaines, tous les mois et toutes les années. La récurrence se finit après un nombre prédéfini de répétitions ou après une date d'échéance.

Deux événements simples sont en conflit lorsque leurs intervalles de temps se chevauchent. Un intervalle est le temps qui se déroule entre le début et la fin d'un événement. Deux événements récurrents sont en conflit lorsque deux occurrences des ces événements sont en conflit.

- (5) (a) Utilisez le diagramme de classes UML pour modéliser le domaine, avec des classes, des attributs et des associations entre ces classes. Ne pas oublier que tout attribut a un type, qui est un type de base de UML, une classe ou un *datatype*.

**Solution:**

- (7) (b) Utilisez les diagrammes UML de classes et de séquences pour modéliser la conception de la solution. L'objectif de la conception est de spécifier l'opération booléenne `enConflit()`, qui retourne Vrai lorsque deux événements sont en conflit et Faux dans le cas contraire. La conception doit être guidée par la performance de cette opération.

**Solution:**

**Solution:**