NOM :

Prénom :

Examen, 1h30, Feuille A4 autorisée, à rendre

Barème indicatif :

exercice	1	2	3	4	5	6	7	8
points	2,5	2	3	2,5	2,5	2,5	2,5	2,5

Exercice 1 : Propriété

Java définit la spécification Java Beans : un objet a une propriété `xxx` de type `X` si sa classe définit une méthode appelée accesseur, `X getXxx()`, qui retourne sa valeur et une méthode appelée modifieur, `void setXxx(X xxx)`, qui modifie sa valeur. Ces méthodes doivent être publiques. Si seul l'accesseur est défini, la propriété est en lecture seule. Les attributs de la classe ne sont pas pris en compte dans la détermination des propriétés. La classe `A` ci-après a trois propriétés `Valeur`, `NbAcces` et `NbModifs`. Les deux dernières sont en lecture seule.

```
1  class A {
2      private String valeur;
3      private int nbAcces, nbModifs;
4      public String getValeur()       { this.nbAcces++; return this.valeur; }
5      public void setValeur(String v) { this.valeur = v; this.nbModifs++; }
6      public int getNbAcces()         { return this.nbAcces; }
7      public int getNbModifs()        { return this.nbModifs; }
8  }
```

1.1. La classe B ci-après a pour propriétés en lecture seule N et C. Expliquer pourquoi.

```
1  class B {
2      private int n;
3      public int  getN()         { return n;  }
4      public int  getB()         { return n / 2;  }
5      public double getC()       { return n * 1.0;  }
6      public void setB(int b)    { n = b * 2; }
7      public void setC(String c) { n = Integer.parseInt(c); }
8  }
```

1.2. Écrire une fonction en Java qui à partir du nom d'une classe (`String`) retourne la liste des noms de ses propriétés en lecture seule.

On pourra utiliser les méthodes `startsWith` et `substring` de `String`. `s1.startsWith(s2)` renvoie vrai si et seulement si `s1` commence par `s2`. `s1.substring(n)` est la chaîne des caractères de `s1` à partir de l'indice `n`. `"getValeur".substring(3)` retourne `"Valeur"`.

On pourra utiliser les éléments suivants (listés par ordre alphabétique) : `Class`, `forName`, `getConstructor`, `getConstructors`, `getDeclaredField`, `getDeclaredFields`, `getDeclaredMethod`, `getDeclaredMethods`, `getField`, `getFields`, `getMethod`, `getMethods`, `getName`, `getParameterTypes`, `getReturnType`, `invoke`, `newInstance`, `NoSuchMethodException`, ...

Exercice 2 Répondre de manière concise et précise aux questions suivantes.

2.1. Acceleo permet de définir des *templates* et des *queries*. Expliquer ces deux notions.

2.2. Expliquer l'objectif de l'outil Pitest et son principe.

Exercice 3 : Test structurel

On considère la fonction `mystere` suivante :

```
1  public int mystere(int[] tab, int elt) {
2          int fin = tab.length;
3          int i = 0;
4          while (i < fin && tab[i] < elt) {
5                  i = i + 1;
6          }
7          return i;
8  }
```

3.1. Dessiner le graphe de contrôle correspondant aux instructions de cette fonction.

3.2. Combien y a-t-il de chemins (en considérant qu'on passe au plus une fois dans la boucle) ?

3.3. Compléter le tableau suivant en indiquant pour chaque variable (en ligne) l'ensemble de ses utilisations : on donnera le numéro de ligne en distinguant les utilisations dans des calculs (*c-use* (*computation*)) et les utilisations dans des conditions (*p-use* (*predicate*)). On donnera aussi l'ensemble des paires *def-use* pour chaque variable.

variable	*c-use*	*p-use*	paires *def-use*
tab	2,	4	(1,2) (1, 4)
elt		4	(1, 4)
fin		4	(2, 4)
i	5. 7	4	(3, 4) (3.5) (3, 7) (5.4) (5,5) (5,

3.4. Est-ce que le test `mystere(new int[] {}, 5)` couvre toutes les décisions ? La réponse doit être argumentée.

Cinémathèque

Après 125 ans d'existence, le cinéma a su offrir toujours plus de spectacles grandioses et touchants, et ce pour toutes les tranches d'âge et toutes les cultures. La quantité effarante de métrages sortis depuis sa naissance fait qu'il est parfois difficile de s'y retrouver.

Nous nous proposons d'apporter une solution à ce problème, à base d'*ingénierie dirigée par les modèles*.

Exercice 4 : Cours

On considère le métamodèle donné à la Figure 1, que l'on appellera CINE.

4.1. Donner le nom des concepts *Ecore* correspondant aux éléments suivants : 1) `Cinémathèque`, 2) `titre` (dans `Métrage`) et 3) `staff` (entre `Métrage` et `Staff`).

4.2. Ajouter les multiplicités (ou arités) des relations sur le le métamodèle (Figure 1).

4.3. Quel est le sens du losange plein (♦) de la flèche entre `Cinémathèque` et `Personne` ? Cette relation est-elle pertinente ? Nécessaire ? Argumenter les réponses.

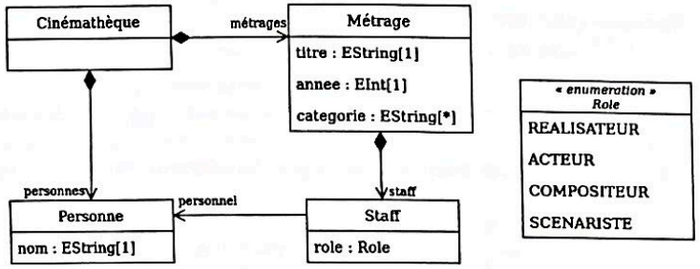

FIGURE 1 – Métamodèle CINE

Exercice 5 : Manipulation du métamodèle

On considère le métamodèle CINE de la Figure 1.

5.1. Représenter, sous la forme d'un **diagramme d'objet** conforme au métamodèle, 1) le film « Le Contrat », sorti en 1986, catégories *Action* et *Thriller*, dont le réalisateur est John Irwin, le scénariste Norman Wexler et Arnold Schwarzenegger l'un des acteurs et 2) le film « Total Recall », sorti en 1990, catégories *Thriller* et *SF* avec Arnold Schwarzenegger comme acteur.

5.2. On voudrait pouvoir ajouter la notion de *société de production* à notre système. Une société de production produit des films. Elle est caractérisée par son **nom** et son **pays**, sous la forme d'un code à deux lettres (FR, US...). Modifier le métamodèle de façon à inclure ce nouveau concept. Ce métamodèle Ecore doit être **valide** et **complet**.

Exercice 6 : OCL

Dans la suite, nous ferons référence au métamodèle de la Figure 1.

6.1. Exprimer, en OCL et en précisant le **contexte**, les contraintes suivantes :

1. Le titre d'un film ne peut pas être vide
2. Un métrage doit avoir (au moins) une personne dans son staff
3. Un métrage doit avoir (au moins) un *réalisateur*

6.2. Dans l'optique de faciliter des traitements ultérieurs, on se propose d'écrire une *requête OCL* qui, pour une **personne** donnée, retourne l'**ensemble des métrages** dans lesquels elle a été impliquée. Écrire en OCL une telle requête en complétant le code donné ci-après :

```
1 context Cinémathèque
2 def: getMétrages(p: Personne): Collection(Métrage) =
3     ...
```

Exercice 7 : Xtext

On associe une grammaire exprimée avec Xtext (Figure 2) pour notre cinémathèque.

7.1. Dessiner le métamodèle Ecore correspondant à la grammaire de la figure 2.

7.2. Écrire un texte, basé sur la syntaxe proposée, pour représenter les données suivantes :

1. « *À la Poursuite d'Octobre Rouge* », sorti en 1990, catégorie *Thriller*, avec John McTiernan (réalisateur), Tom Clancy (scénariste), Sean Connery (acteur).


```
1  grammar fr.n7.gls.cine.Cine
2  generate disco "http://www.n7.fr/gls/cine/Cine"
3
4  Cinémathèque: 'cine' nom=ID '{' métrages+=CinéÉlément* '}' ;
5
6  CinéÉlément: Personne | Métrage ;
7
8  Personne: 'personne' id=ID nom=STRING ';' ;
9
10 Métrage:
11    'métrage' titre=STRING 'sorti' 'en' année=INT '(' catégories+=EString* ')'
12    'avec' staff+=Staff* ;
13
14 Staff: '-' personnel=[Personne] '[' role=Role ']' ';' ;
15
16 enum Role:
17    REALISATEUR='Réalisateur' | ACTEUR='Acteur'
18    | COMPOSITEUR='Compositeur' | SCENARISTE='Scénariste' ;
```

FIGURE 2 – Grammaire Xtext pour le métamodèle DISCO

2. « *Last Action Hero* », sorti en 1993, catégories *Action* et *Fantastique*, avec John McTiernan (réalisateur), Shane Black (scénariste), Arnold Schwarzenegger (acteur).

Exercice 8 : Transformation de modèle

On décide de synchroniser notre collection de métrages avec un site web qui permet de garder trace des films vus. Ce site web utilise un format particulier et classe les films *par acteur*. On désire automatiser l'exportation de notre cinémathèque sous le format requis par le site web, afin de pouvoir synchroniser notre collection régulièrement sans avoir à le faire manuellement.

Le format admis par le site suit le métamodèle REGISTRE donné à la Figure 3. La collection est représentée par un concept `Registre` qui regroupe un ensemble d'`Acteurs`. Chaque `Acteur` est identifié par son nom, et est associé à une liste de `Films`. Chaque `Film` est identifié par son titre et son année de sortie.

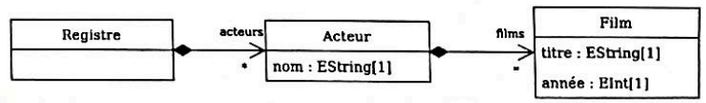

FIGURE 3 – Métamodèle REGISTRE

Par exemple, le registre des films mentionnés dans ce sujet contiendrait :

— *Arnold Schwarzenegger* : Le Contrat (1986), Total Recall (1990), Last Action Hero (1993).

— *Sean Connery* : À la Poursuite d'Octobre Rouge (1990).

8.1. Expliquer comment transformer un modèle conforme au métamodèle CINE (Figure 1) en un autre modèle conforme au métamodèle REGISTRE (Figure 3) en utilisant ATL. On peut donner du pseudo-code ATL dont les principes seront explicités.