

Génie Logiciel

Année 2004-2005

15 Juin 2005 -Rattrapage-

- durée 3 heures -

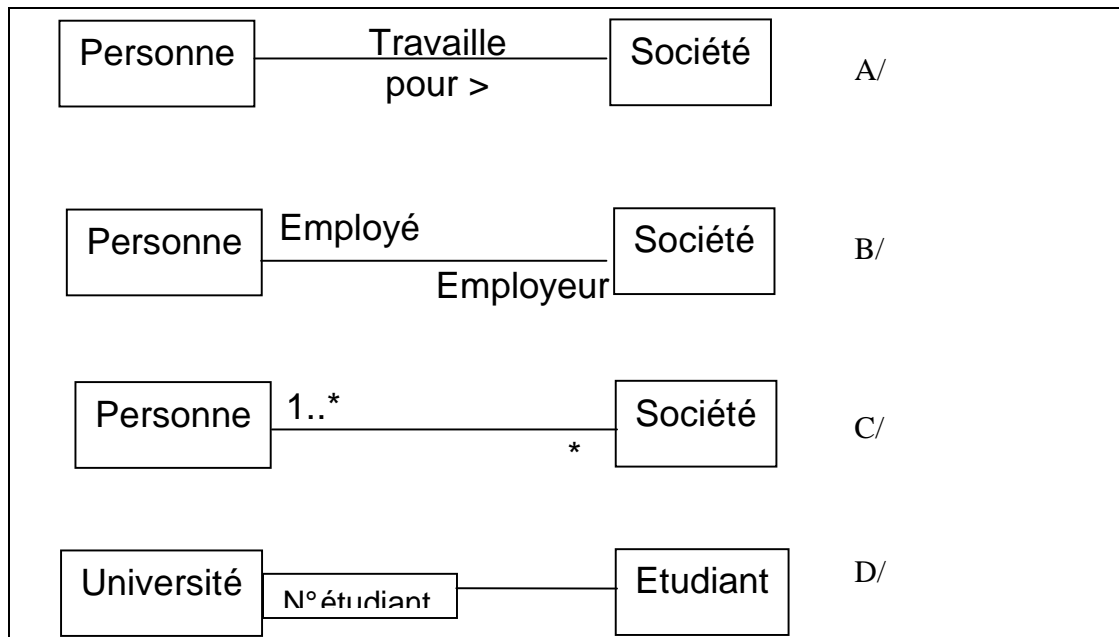
Les documents ne sont pas autorisés.

Les exercices sont indépendants.

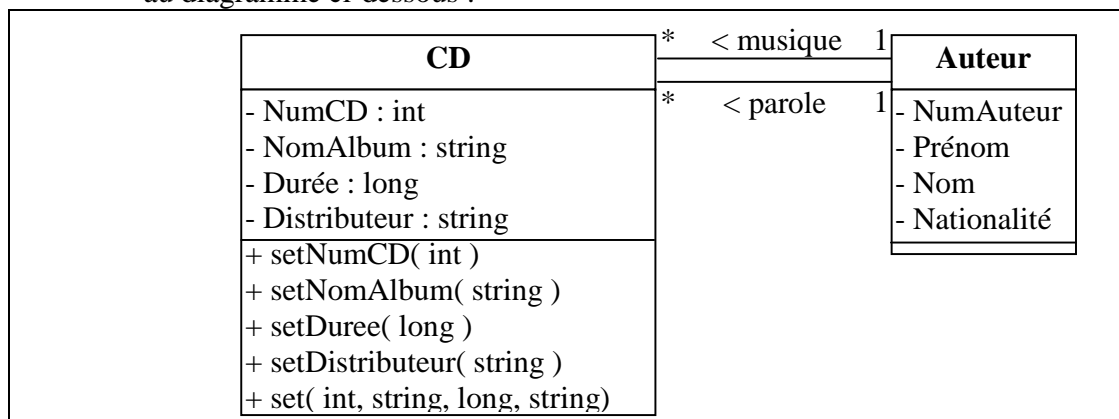
Vous devez justifier toutes vos réponses.

I - Partie académique (durée conseillée : 45 minutes)

1. Comparez l'expressivité des représentations Entités/Associations (MCD Merise) avec le diagramme de classes d'UML.
2. Expliquez chaque diagramme UML suivant :



3. Ecrivez le code C, C++, Pascal ou JAVA (il faut un choisir un seul) correspondant au diagramme ci-dessous :



4. Expliquez la méthode en cascade et citez les projets pour lesquels cette méthode convient et ceux pour lesquels elle ne convient pas.
5. S'il en existe, citez le(s) cas où les méthodes dites agiles ne sont pas applicables.

II - Partie industrielle (durée conseillée : 2 heures)

Vous êtes chef de projets dans la société *Denavelli* spécialisée dans le développement de logiciels spécifiques et vous êtes chargé du projet « Game Of Life » décrit ci-dessous. La réalisation du projet n'est pas à l'ordre du jour mais vous devez préparer et organiser le travail à faire. Pour cela, vous devez faire un dossier comprenant :

- les grandes lignes des documents que vous créerez et utiliserez pour la réalisation de ce projet en expliquant chacun d'eux,
- certains de ces documents seront détaillés (argumentez vos choix),
- la ou les techniques d'évaluation de la qualité de votre prestation.

Le dossier ainsi constitué sera remis à votre direction qui décidera de concrétiser ou pas ce projet.

Le projet :

Le *jeu de la vie* (Game of Life ou Life) appartient à la classe des jeux de simulation. Il s'agit en l'occurrence d'un automate cellulaire dont les règles ont été définies par le mathématicien John Horton Conway en 1970. L'objectif de cet automate est de modéliser l'évolution d'organismes vivants en fonction de règles simples. C'est d'ailleurs l'analogie entre ces règles et certains critères d'évolution de populations de bactéries qui a conduit à donner à cet automate le nom de jeu de la vie. Son intérêt est en effet de montrer que ces règles peuvent permettre de mettre en évidence des fonctionnements non triviaux, rappelant quelque peu la richesse et la diversité de la vie.

Plus concrètement, le jeu de la vie se présente sous la forme d'une tableau (en théorie infini) dont chaque case peut contenir ou non une cellule (une case du tableau possède donc deux états : vide ou contenant une cellule). Une population initiale est fournie au système soit par l'utilisateur, soit à partir d'un pattern contenu dans un fichier d'extension LIF. Lancer le processus consiste donc à demander au système de calculer la ou les génération(s) suivante(s), c'est-à-dire à déterminer pour chaque nouvelle génération, les naissances et les morts.

L'état d'une case, à la génération $n+1$, est décidé en fonction de son état et du nombre des cellules voisines à la génération n . Une cellule A est dite voisine d'une cellule B si elle est contiguë à B en ligne, colonne ou diagonale. Une cellule peut avoir 8 voisines :

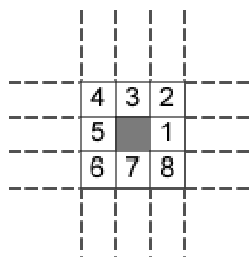


Figure 1 : Voisinage d'une cellule sur une grille rectangulaire

Une cellule vivante à l'étape n ne survivra à l'étape $n+1$ que si elle a 2 ou 3 voisines précisément.

Une cellule vivante à l'étape n possédant moins de 2 voisines mourra par isolement à l'étape $n+1$.

Une cellule vivante à l'étape n possédant plus de 3 voisines mourra par surpopulation à l'étape $n+1$.

Une case vide à l'étape n possédant exactement 3 voisines, donnera naissance à une cellule à l'étape $n+1$.

Note : L'état de l'automate à l'étape $n+1$ est uniquement fonction de son état à l'étape n .


```
#N
#P -20 -52
.....*.*
.....*.*.*
.....*.*
```