



AG80 : Informatique avancée  
pour l'ingénieur  
Examen Final - A2010

Consignes :

- ✓ Ce sujet comporte trois parties : une concernant la programmation orientée objet, une concernant les formats de données et une concernant les bases de données. Il est conseillé de traiter un maximum de questions dans chacune des **trois parties**.
- ✓ Rédiger **impérativement** les **trois parties** sur des **copies différentes**, de façon à simplifier le travail des correcteurs.
- ✓ Le barème total est sur 20 points.
- ✓ Aucun document n'est autorisé.

## Partie Conception et programmation orientée objet

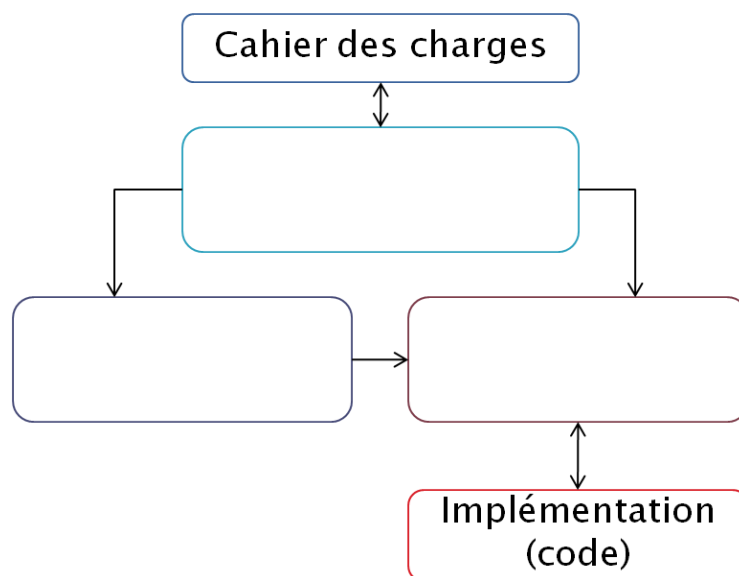
Cette partie est notée sur 11 points.

### Questions de cours :

#### Conception orientée objet

1. Quels sont les avantages de la Conception Orientée Objet par rapport à la programmation « classique » fonctionnelle ? (0,5 pt)
2. Quelle est la différence entre une classe et un objet ? (0,5 pt)
3. Cas d'utilisation : Qu'est ce qu'un acteur ? Est-il forcément un utilisateur du système (1 pt)
4. Diagramme de classes : Donner les 3 types de relations inter classes détaillées en cours. (1 pt)

#### Reproduire sur la copie et compléter (1 pt)



#### Exercice UML :

Nous cherchons à concevoir un programme destiné à la gestion des cours d'une université. Le service de la scolarité crée le catalogue des UVs disponibles en fonction du semestre courant sur notre programme. Une fois ce catalogue créé, il est envoyé sur le serveur dédié de l'université. Lors de chaque fin de semestre, les enseignants choisissent les UVs dont ils souhaitent dispenser les cours en fonction du catalogue des UVs disponibles. L'accès au catalogue des UVs et l'inscription des enseignants à ces UVs se fait par l'intermédiaire de notre programme, qui enverra les informations sur le serveur après validation par l'utilisateur. Une fois les cours associés aux enseignants, les étudiants choisissent leurs UVs par l'intermédiaire de notre programme. Pour chaque semestre, ils devront choisir entre 4 et 6 Uvs. Chaque inscription terminée est envoyée sur le serveur. Enfin, une fois tous les étudiants inscrits, et en fonction des informations disponibles sur le serveur, le service de la scolarité saisit les emplois du temps par l'intermédiaire de notre programme. Ces emplois du temps seront consultables par les étudiants toujours par l'intermédiaire de notre programme.

## Examen Final AG80 – Semestre A2010

---

Précision importante, afin d'éviter les fraudes et erreurs, chaque utilisateur doit s'identifier avant de pouvoir accéder à ses services.

1. Identifier les acteurs du système, et déterminer les cas d'utilisation associés à ce programme de gestion des cours d'une université. (3pts)

Chaque UV du catalogue est associée à certaines informations particulières. Chacune possède une catégorie spécifique : « Scientifique » ou « Culture générale ». Chacune est également associée à un et un seul enseignant responsable, ainsi qu'à un nombre d'élève déterminé. Chaque enseignant et chaque élève sera donc associé à un certain nombre d'UVs du catalogue.

2. En réutilisant les informations fournies dans la première partie de l'exercice, les informations complémentaires de l'énoncé ainsi que les classes listées ci-dessous, déterminer les relations inter-classes, leurs cardinalités, ainsi que les attributs et méthodes (les plus pertinents). (4pts)

Liste des classes :

- Catalogue
- UV
- Étudiant
- Enseignant

## Partie formats de données

Cette partie est notée sur 6 points.

### Questions de cours

1. Quelle est la différence entre le codage ASCII et le codage binaire d'un nombre entier ? (1 pt)
2. Quelles sont les caractéristiques principales des formats d'images matriciel et vectoriel ? Citer quelques formats. (1 pt)
3. Quels sont les avantages de XML par rapport à HTML ? (1 pt)

### Exercice de classement de données

On souhaite ranger dans un fichier XML les données pour un site en ligne de vente de voitures d'occasion. Voici quelques exemples de véhicules à renseigner :

Cat : citadine Marque : Peugeot Modèle : 107 3P Moteur : 1.0 Ess Km : 20000 Année : 2008 Couleur : Jaune Finition : Trendy Rem : Clim Prix : 6800 Euros	Cat : citadine Marque : Citroën Modèle : C1 5P Moteur : Ess 1.0i Km : 29200 Année : 2007 Couleur : Bleu Finition : Confort Rem : Prix : 5990 Euros
Cat : compacte Marque : Peugeot Modèle : 308 5P Moteur : 1.6 HDI90 Km : 52600 Année : 2008 Couleur : Montebello Finition : Premium Rem : 1ere main Prix : 11990 Euros	Cat : compacte Marque : Citroën Modèle : C3 5P Moteur : Ess 1.4 Km : 12500 Année : 2010 Couleur : Rouge métallisé Finition : Confort Rem : Prix : 10790 Euros
Cat : familiale Marque : Peugeot Modèle : 308 SW Moteur : 1.6 HDI FAP 112 Km : 35000 Année : 2008 Couleur : Gris Finition : Navteq Rem : Prix : 19500 Euros	Cat : familiale Marque : Citroën Modèle : C5 Tourer Moteur : HDI FAP 140 Km : 44200 Année : 2009 Couleur : Noir Finition : Exclusive Rem : BV automatique Prix : 14990 Euros
Cat : monospace Marque : Peugeot Modèle : 807 Moteur : 2.2 HDI SV Km : 85500 Année : 2003 Couleur : Gris Alu Finition : Premium Rem : 3 Toits ouvrants, GPS Prix : 11990 Euros	Cat : monospace Marque : Citroën Modèle : C8 Moteur : HDI FAP 135 Km : 145000 Année : 2006 Couleur : Gris fer Finition : Confort Options : Rails de toit Prix : 9500 Euros

4. Proposer les fichiers XML de ces données (donner au minimum 2 véhicules), selon deux modes de classement :
  - classement par catégorie (1,5 pt)
  - classement par marque du véhicule (1,5 pt)

## Partie Base de données

Cette partie est notée sur 3 points.

### Présentation générale

Une PME de déménagement souhaite se doter d'un système d'information afin de pouvoir stocker et préparer ses missions de déménagement. Nous allons concevoir la structure de la base de données relationnelle qui stockera ces informations.

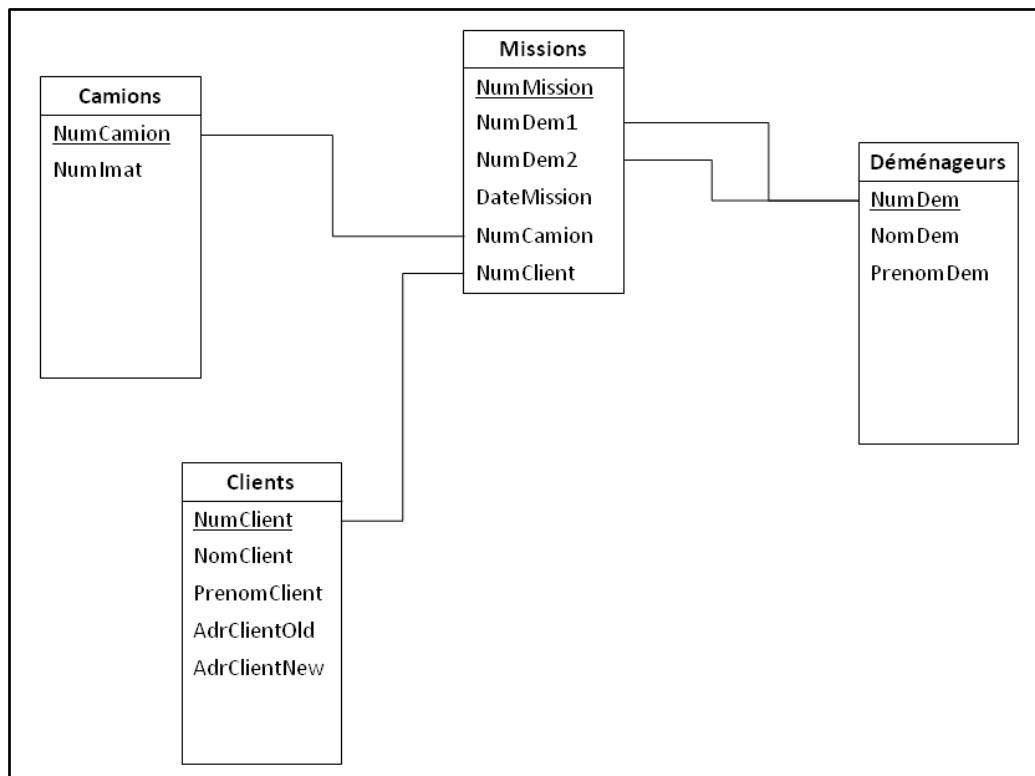
Une analyse du besoin, menée auprès des futurs utilisateurs de ce système, montre qu'il faudra stocker quatre familles d'information : les clients, les déménageurs, les véhicules, ainsi que les missions de déménagement. Ceci nous donnera donc un modèle de base de données constitué de quatre tables. Chaque enregistrement d'une table sera identifié de manière unique par un nombre entier (clé primaire).

Nous souhaitons, de plus, stocker dans la base de données :

- Les immatriculations des camions
- Les prénoms, noms, anciennes et nouvelles adresses des clients
- Les noms et prénoms des déménageurs
- Les deux déménageurs, la date, le camion et le client relatifs à une mission. (Dans un premier temps, on considère qu'il y a exactement deux déménageurs par mission)

### Questions

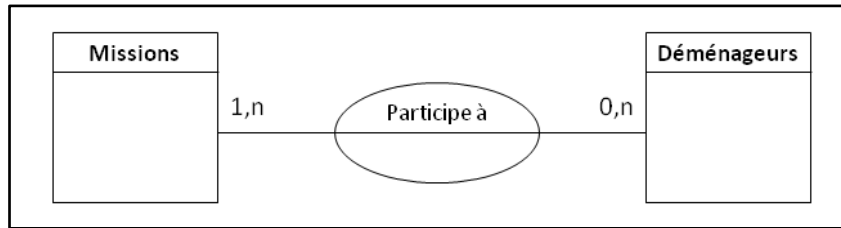
D'après les informations données précédemment, le Modèle Logique de Données (MLD) proposé est le suivant.



## Examen Final AG80 – Semestre A2010

---

1. On souhaite modifier la base de données de telle sorte qu'on puisse assigner un nombre illimité de déménageurs à une mission. La cardinalité entre les tables « Missions » et « Déménageurs » devient comme ci-dessous :



Proposer une nouvelle version du MLD prenant en compte cette modification. (2 pts)

2. Que permet d'obtenir la requête SQL suivante (1 pt) :

```
SELECT NumCamion IN Missions WHERE DateMission=21-01-2011
```