

DS OBJET

2 novembre 2015 - Durée 2h - Tous documents autorisés

1 PRÉSENTATION DU SUJET

1.1 PRÉCISIONS SUR LE DEGRÉ DE RÉALISME DU SUJET

En préambule, veuillez noter que les marques, situations et les procédés de fabrication décrits dans ce sujet sont fictifs et ne cherchent pas à représenter fidèlement ce qui se passe dans la réalité. Ainsi, certaines de nos affirmations ou descriptions ne sont pas réalistes, mais vous devez les considérer exactes dans le cadre de ce sujet d'examen.

Les approximations faites ici le sont dans le but de pouvoir illustrer les notions abordées au cours des cours magistraux et lors des travaux pratiques du module de Programmation Orientée Objet.

1.2 RECOMMANDATIONS

Il vous est vivement recommandé de lire l'intégralité du sujet avant de commencer à répondre aux questions !

Pour les implémentations en Java, il n'est pas nécessaire d'écrire intégralement les accesseurs et modificateurs.

1.3 SYSTÈME D'INFORMATION D'UNE LIGNE DE PRODUCTION DE VÉHICULES

La marque *Citronault* s'est adressée à vous dans le cadre de la modernisation du système d'information de sa ligne de production. À partir de l'expression du besoin de cette société

(voir plus bas), vous devrez répondre aux questions de la Section 2. Voici l'expression du besoin telle que décrite par *Citronault*.

La société *Citronault* est une marque bien connue produisant trois types de véhicules :

- des voitures,
- des motos,
- des camions.

Ces véhicules sont produits sur une ligne de production, à partir d'une collection de modèles disponibles au bureau d'étude (BE). Chacun des **modèles** est soit une **voiture**, soit une **moto**, soit un **camion** et possède un **panel d'options** potentielles ainsi qu'un **panel de couleurs** disponibles, un modèle est **identifié** par un nom (chaîne de caractères) qui est supposé unique et est caractérisé par un **type de moteur** (essence, diesel, etc.).

Le BE peut vous fournir un modèle particulier à partir de son nom, si celui-ci est bien dans sa collection.

Lorsqu'un client passe une commande, celui-ci choisit donc un modèle de véhicule ainsi qu'une couleur et une liste d'options disponibles pour ce modèle. À partir de ces informations, un nouveau véhicule sera donc rajouté à la ligne de production de *Citronault*. Un client pourra par exemple commander la moto Pipo de la marque *Citronault* en modèle essence, de couleur bleu nuit avec les options suivantes :

- "bulle racing",
- "silencieux de pot d'échappement",
- "protection réservoir",
- "capot de selle".

Il est bien évident que plusieurs clients peuvent passer commande et dès alors, les véhicules commandés sont rajoutés dans la ligne de production en fonction de leur date de commande (on suppose qu'il ne peut y avoir eu plusieurs commandes passées exactement au même instant). La ligne de production de *Citronault* contient donc à un instant donné tous les véhicules en cours de production

La marque *Citronault* compte aussi la Police Nationale comme client. Toutefois, cette dernière possède quelques particularités :

- elle commande uniquement des motos et des voitures pour l'instant ;
- elle possède des modèles particuliers à elle ;
- les véhicules de police nécessitant de l'équipement particulier (gyrophares et radar de contrôle de vitesse embarqué), ces derniers ont besoin d'être séparés des véhicules "classiques" de la ligne de production de *Citronault* à chaque fin de journée.

1.4 PARTICULARITÉS DES VÉHICULES

Afin de nous éviter des erreurs et pour contraindre notre proposition de système d'information, *Citronault* vous fournit les informations suivantes concernant les véhicules.

Pour chacun des véhicules, *Citronault* veut être capable de connaître à tout instant :

- sa couleur,
- son nombre de chevaux fiscaux (un nombre entier),
- sa consommation moyenne en litres d'essence sur 100 km (un nombre flottant),
- son type de moteur.

En plus des informations précédentes, les **voitures** possèdent des informations qui leurs sont propres et que l'on veut pouvoir connaître, à savoir :

- le type de voiture (4x4/cabriolet/berline),
- son type de boîte de vitesse (automatique ou manuelle),
- le nombre de portes (3 ou 5).

De la même manière, les camions doivent pouvoir nous spécifier (en plus des informations communes à tous les véhicules) :

- le nombre de roues (un nombre entier),
- son tonnage, c'est à dire le poids maximum du camion (un nombre flottant).

Enfin, en plus des informations communes à tous les véhicules, on souhaite connaître des motos :

- leur type (routière/trail/tout-terrain),
- leur cylindrée en cm^3 (un nombre entier).

Les véhicules de police ont quant à eux des méthodes particulières nous permettant de connaître :

- la portée de leur radar embarqué (nombre flottant),
- le nombre de gyrophares installés sur le véhicule (nombre entier).

Bien évidemment, **ces informations viennent en complément des informations disponibles sur les véhicules "classiques"**. Par exemple, sur une moto de police on pourra connaître :

- sa couleur,
- son nombre de chevaux fiscaux,
- sa consommation moyenne,
- son type de moteur,
- son type,
- sa cylindrée,
- la portée de son radar embarqué,
- le nombre de gyrophares installés.

1.5 INFORMATIONS FOURNIES PAR *Citronault*

Pour simplifier votre tâche de modélisation et de proposition de son nouveau système d'information, *Citronault* vous fournit les diagrammes de classes UML suivants :

- Couleur (voir Figure 1.1 gauche),
- Option (voir Figure 1.1 milieu),
- Moteur (voir Figure 1.1 droite).

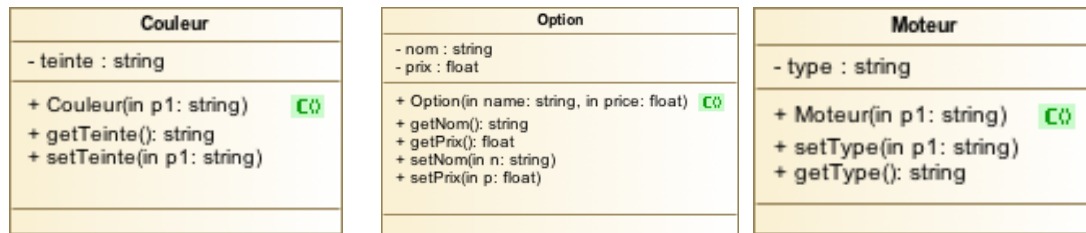


FIGURE 1.1 – Diagramme de classe UML des classes `Couleur`, `Option` et `Moteur`.

Pour rappel, le logiciel Modelio utilisé ici remplace la classe `String` de Java par “string”.

2 QUESTIONS

2.1 QUESTION 1 :

Proposez le diagramme de classes UML représentant les différentes classes du nouveau système d’information de *Citronault*.

A minima, votre schéma UML devra contenir les éléments suivants :

- `Vehicule`,
- `Camion`,
- `Moto`,
- `Voiture`,
- `LigneProduction`,
- `BE`.

Veillez noter que les éléments donnés ci-dessus sont probablement insuffisants pour modéliser le SI de *Citronault*;

Pour chaque classe de ce diagramme, veillez à bien faire figurer :

- les attributs (droit d’accès, type, nom) ;
- les principales méthodes (celles qui présentent un intérêt à vos yeux, typiquement pas les accesseurs/modificateurs) ;
- le constructeur de copie de chaque classe ;
- les relations éventuelles d’héritage et/ou d’implémentation d’interfaces ;
- n’hésitez pas à nous indiquer les éléments UML devant être écrits en italique (par exemple en les soulignant, en utilisant une autre couleur, etc.) de manière claire : en écrivant par exemple que les éléments soulignés sont en italique.

2.2 QUESTION 2 :

Implémentez `Vehicule` en Java. De plus, nous souhaitons que `Vehicule` propose une méthode `copie` qui ne prend pas de paramètre d’entrée et qui retourne un objet de type `Vehicule` qui est une copie exacte à l’instant de la copie.

On veut par exemple pouvoir écrire le code suivant :

```
// on suppose l'appel au constructeur correct !
Vehicule monVec = new Camion(...);

Vehicule vehi2 = monVec.copie();
// ou encore
Camion monCam = (Camion)monVec.copie();
```

2.3 QUESTION 3 :

Implémentez les méthodes suivantes de `Camion` :

- `copie()`;
- le constructeur de recopie `public Camion(Camion c) ...`.

2.4 QUESTION 4 :

Implémenter la méthode `ajoutVehicule` de `LigneProduction`, qui prendra les paramètres suivants :

- `modele` : de type `String`;
- `options` : une liste de noms d'options (et donc de type `String`);
- `couleur` : le nom d'une couleur (de type `String`).

Pour ce faire, vous supposerez écrits (et corrects) les constructeurs de recopie de `Couleur` et d'`Option`.

Pour rappel, cette méthode est censée rajouter un nouveau `Vehicule` à la ligne de production.

2.5 QUESTION 5 :

Comme décrit précédemment, *Citronault* souhaite pouvoir isoler sur un parking séparé tous les véhicules de police présents sur la ligne de production à certains moments.

Proposez une méthode `getVehiculePolice()` de `LigneProduction` permettant :

- d'obtenir la liste des véhicules de police;
- en les retirant de la liste des véhicules de la ligne de production.

2.6 QUESTION 6 :

La police vient de passer commande de camions *Citronault*! La direction est affolée, vous devez les rassurer sur la robustesse de leur nouveau système d'information. Indiquez :

1. les changements à apporter à votre schéma UML de la Question 1 ;
2. les modifications à apporter au code de votre méthode `getVehiculePolice()` (si besoin ré-écrivez la).

2.7 QUESTION 7 :

Implémentez `CamionPolice`.