

Examen Final GL52  
P2012  
Durée: 2h  
Aucun document autorisé  
**1 copie par exercice**

Jeudi 28 juin 2012

## 1 UML et méthodologie orientée-objet (10 points)

### Pompe à essence en libre service

L'objectif de cet exercice vise à modéliser le fonctionnement du système « Pompe à essence automatique en libre service ». Ce système gère la distribution (sortie et débit) du carburant, le paiement par les clients et le niveau de carburant dans les réservoirs de stockage.

La pompe dispose de trois pistolets de distribution. Chaque pistolet est associé à un type spécifique de carburant : *regular*, *plus*, et *premium*, chaque type désignant un niveau d'octane dans l'essence. Le prix varie en fonction du type de carburant.

Un client qui souhaite remplir son réservoir de voiture doit suivre la procédure décrite ci-dessous, le système « Pompe à essence » réagit en fonction des contraintes spécifiées :

- Le client choisit le type de carburant désiré et appuie sur le bouton *Valider*.
- Le système demande au client d'insérer sa carte bancaire.
- Le client insère sa carte bancaire.
- Le système demande au client de taper son code.
- Le client tape son code à 4 chiffres et appuie sur le bouton *Valider*.
- Le système vérifie le code de carte bancaire.
  - Si le code est incorrect la transaction est annulée, le système revient dans son état initial : attente de la sélection du carburant.
  - Si le code est correct, le système affiche sur son écran que la distribution peut débiter, le client obtient l'autorisation d'utiliser le pistolet du carburant sélectionné.
- Le client prend le pistolet du carburant sélectionné et remplit son réservoir jusqu'à ce qu'il soit plein ou que la limite de 200 euros soit atteinte.

- Chaque pompe dispose d'un compteur à débit pour chaque carburant qui envoie un signal au système chaque fois que 1/400ième de litre (0,0025L) de carburant est distribué.
- De manière continue, le volume en Litres et le prix en Euros du carburant déjà distribué sont affichés sur l'écran digital de la pompe.
- Le pistolet dispose d'un capteur qui émet un signal ce qui arrête automatiquement la distribution lorsque le réservoir est plein.
- La distribution est également automatiquement stoppée dès que le prix du carburant déjà distribué atteint 200 euros
- A tout moment le client peut arrêter manuellement la distribution avant que le réservoir soit plein.
- Une fois la distribution terminée, le client retourne le pistolet dans son emplacement initial.
- Le système effectue alors la transaction bancaire du montant correspondant au prix du carburant distribué depuis le compte bancaire du client, puis retourne à son état initial.
- Le niveau d'un carburant dans le réservoir de stockage ne peut pas descendre en dessous d'un seuil défini (5% de la capacité du réservoir). Si cela se produit, la pompe liée à ce réservoir empêche la sélection du carburant en question. Si cela se produit en cours de distribution, la distribution est stoppée et le système affiche un message d'information au client.

**Remarque : les diagrammes UML doivent être complets et clairement présentés.**

### 1.1 Question 1 (4 points)

Fournir le diagramme de classe (attributs, rôles, cardinalités, etc.) du système « Pompe à essence » : pompe, réservoir, client, compte, paiement, etc.

### 1.2 Question 2 (3 points)

Fournir le diagramme d'activité du système « Pompe à essence ».

### 1.3 Question 2 (3 points)

Fournir le diagramme de séquence (niveau objet) pour le cas d'utilisation : *le client fait le plein de carburant.*

## 2 Méthode B (5 points)

### Une machine abstraite

La machine abstraite *Bus* enregistre le nombre de passagers à bord d'un bus et le nombre de tickets vendus. Certains passagers sont dispensés de ticket. L'opération **buy** permet d'acheter plusieurs tickets d'un coup. L'opération **board** enregistre la montée de plusieurs passagers en même temps. L'opération **double** permet de doubler le nombre de passagers et de tickets.

```

MACHINE Bus
SETS
VARIABLES
  tickets, passengers
INVARIANT
   $tickets \in \mathbb{N} \wedge passengers \in \mathbb{N} \wedge tickets \leq passengers + 1$ 
INITIALISATION
   $tickets, passengers := 0, 0;$ 
OPERATIONS
  buy (mm)  $\hat{=}$ 
    PRE  $tickets < passengers \wedge mm \in \mathbb{N}$ 
    THEN  $tickets := tickets + mm$  END ;
  board (nn)  $\hat{=}$ 
    PRE  $nn \in \mathbb{N}$ 
    THEN  $passengers := passengers + nn$  END ;
  double (mm)  $\hat{=}$ 
    PRE TRUE
    THEN  $tickets, passengers := 2 \times tickets, 2 \times passengers$  END ;
END

```

#### 2.1 Question 1 (2.5 points)

Indiquez, en vous basant sur l'obligation de preuve, quelles sont les opérations qui ne sont pas consistantes avec l'invariant (c'est à dire, qui pourraient ne pas toujours le satisfaire). Corrigez cette situation soit en modifiant l'invariant soit en corrigeant les opérations inconsistantes.

#### 2.2 Question 2 (2.5 points)

Ajoutez une nouvelle contrainte à l'invariant :  $passengers \leq tickets + 45$  pour limiter la différence entre passagers avec et sans ticket. Modifier les opérations qui ne seraient plus consistantes avec ce nouvel invariant.

### 3 Questions générales (5 points)

#### 3.1 Question 1 (2.5 points)

Donner les concepts de la méthode SA permettant d'analyser un système, en précisant pour chacun : sa représentation graphique et sa définition.

#### 3.2 Question 2 (2.5 points)

Enumérer les différentes classes de méthodes de développement de logiciels, en indiquant pour chacune, son principe d'analyse.