ENSIAS

Université Mohammed V de Rabat
Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes

Examen

Année Universitaire : 2019 - 2020
Filière : Ingénieur
Semestre : S5
Période : P1 & P2

Date : 02/01/2020
Durée : 1h30min

Module : M.5.1 : Intégration des Applications d'Entreprise
Elément de Module : M.5.1.1 : Intégration des Services et des objets
Professeur : M. Nassar

Consignes aux élèves ingénieurs :

- Uniquement un document A4 recto-verso est autorisé
- Il sera tenu compte de la clarté et de la simplicité des réponses

Exercice 1 : (14 pts)

On considère un parking dont on souhaite automatiser la gestion en l'administrant à distance avec un middleware CORBA. Le parking peut accueillir au maximum 50 véhicules. Une barrière gère les entrées et les sorties de véhicules : elle délivre un ticket à l'entrée d'un véhicule et elle contrôle le ticket fourni par l'automobiliste à la sortie. Un automate de paiement permet sur présentation du ticket, de régler le montant correspondant au temps de stationnement. L'automobiliste règle le montant du stationnement à l'automate avant de franchir la barrière pour sortir. La **barrière**, l'**automate** et le **parking** sont représentés chacun par un objet CORBA.

L'interface ***BarriereItf*** de la barrière fournit deux méthodes :
- ***entrer*** : correspond à l'entrée d'un véhicule dans le parking. Retourne vrai si le véhicule peut rentrer (si place libre), faux sinon (et dans ce cas le véhicule ne peut pas entrer). Cette méthode fournit en paramètre de sortie une structure *Ticket* comprenant un numéro de ticket de type entier et une heure d'entrée sous la forme d'une chaîne de caractères.
- ***sortir*** : correspond à la sortie d'un véhicule. Retourne vrai si l'automobiliste a réglé le montant du stationnement, faux sinon (et dans ce cas le véhicule ne peut pas sortir). Cette méthode prend en paramètre d'entrée une structure Ticket.

L'interface ***AutomateItf*** de l'automate fournit deux méthodes :
- ***payer*** : correspond au paiement en fonction de la durée de stationnement. Retourne vrai si le montant fourni par l'automobiliste est suffisant, faux sinon. Cette méthode prend en paramètre d'entrée une structure Ticket et un montant de type réel double. Cette méthode fournit également en sortie une valeur de type réel double représentant la monnaie à rendre à l'utilisateur.
- ***cestpaye*** : à partir d'une structure Ticket fournie en entrée, retourne vrai si le montant du ticket a été réglé, faux sinon.

L'interface ***ParkingItf*** du parking fournit trois méthodes :
- ***nbPlacesLibres*** : retourne un entier indiquant le nombre de places libres dans le parking.
- ***entreeVehicule*** : signale l'entrée d'un véhicule sur le parking. Ne prend aucun paramètre, ne retourne rien.

- ***sortieVehicule*** : signale la sortie d'un véhicule du parking. Ne prend aucun paramètre, ne retourne rien.

1. Définir la structure Ticket et les interfaces IDL correspondant à ces trois objets CORBA dans un module ParkingPkg. **(2 pts)**
2. Lorsqu'un automobiliste se présente à la Barrière, l'objet CORBA barrière interroge l'objet Parking avant de laisser entrer l'automobiliste. Pourquoi ? Quelle méthode invoque-t-il ? **(3 pts)**
3. A l'issu de cela, il se peut que l'objet Barrière invoque la méthode ***entreeVehicule*** de l'objet Parking. Cette méthode n'a pas de paramètres et a pourtant un rôle essentiel. Lequel ? **(2,5 pts)**
4. Décrire le scénario de la sortie d'un véhicule du parking : quel(s) objet(s) appelle(nt) quelle(s) méthode(s) de quel(s) autre(s) objet(s) et pourquoi ? En particulier, on s'attachera à fournir un scénario de fonctionnement permettant à l'objet Barrière d'autoriser ou non la sortie du véhicule. **(2,5 pts)**
5. Donner le code Java de la classe ParkingImpl implantant l'interface ParkingItf. **(2 pts)**
6. Même question pour l'interface BarriereItf. **(2 pts)**

Exercice 2 (6 points)

Dans cet exercice, il s'agit de créer un service web Axis2 pour un touriste, ***CalcReduceParfum***, qui se base sur deux services : un service qui calcule le nouveau prix de parfum (en Dirham) basé sur le prix initial (en Dirham) selon le sexe de la personne, et un autre service de conversion de Dirham /Dollar (1 Dollar = 9.5 Dh) ou de Dirham /Euro (1 Euro = 11.1 Dh). Ces services proposant deux opérations chacun, et seront déployés sur deux serveurs Axis2 différents.

Le service de conversion ***ConvertisseurWS*** propose les deux opérations suivantes :

- double dh_dollar (double prixdh) ;
- double dh_Euro (double prixdh) ;

Le service de ~~poids idéal~~ ***PrixParfum*** propose les deux opérations suivantes :

- double WFormule (double prix_init) /*Parfum pour Femme : réduction de 40%*/
- double MFormule (double prix_init) /* Parfum pour Homme : réduction de 30%*/

Questions :

1. Proposer l'architecture d'implémentation du service Web ***CalcReduceParfum***.
2. Spécifier l'implémentation des services **ConvertisseurWS**, ***PrixParfum*** et ***CalcReduceParfum*** en donnant pour chacun les déclarations spécifiées dans le fichier services.xml.

Bonne chance

$prix\_init - \left(\frac{prix\_init \times 40}{100}\right)$