

M1 : Ingénierie du Logiciel

UNIVERSITE PIERRE & MARIE CURIE (PARIS VI)

Examen Réparti 2eme partie

10 Janvier 2013 (2 heures avec documents : tous SAUF ANNALES CORRIGÉES). Barème indicatif sur 22 points (max 20/20).

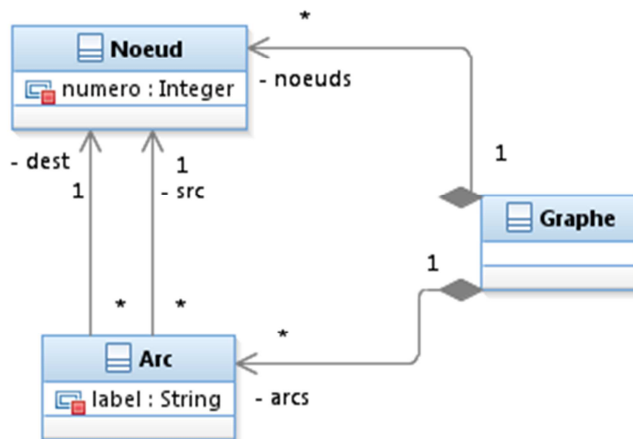
1. Questions de cours

[4 Pts]

Répondez de façon précise et concise aux questions.

Barème : VALABLE sur toutes les questions de cours : -25 à -50% si la réponse inclut la bonne idée, mais qu'elle est noyée dans des infos ou autres réponses fausses/inappropriées.

Q1.1 : Proposez un méta-modèle permettant de représenter des instances de graphes composés de nœuds portant un numéro et d'arcs orientés d'un nœud source vers un nœud destination et portant une étiquette qui est un string. (3 points)



3 méta classes : Graphe, compose * Node et * Edge. Node porte int. Edge porte label. 2 associations 1 de Edge vers Node src et dest. On peut aussi ne pas les orienter et noter * le role des arcs (genre inputEdges, outputEdges).

Barème :

30% compositions de Graphe sur Edge et Node. 20% si juste agregation (pas de losange noir).

20% classe Node + int

20% classe Edge + label

30% les 2 liens entre edge et Node, cardinalité correctes.

Q1.2 : Citez une force et une faiblesse du cycle en V. (1 point)

Force : approche structurée, qualité améliorée, maintenabilité améliorée

Faiblesse : itérations (!!), lourdeur de mise en place

On peut en citer d'autres bien sûr.

50% force

2. Problème: Conception sur AutoLib' [18 Pts]

Suite à l'analyse du système AutoLib', vous avez la charge de l'équipe devant développer le logiciel qui sera déployé dans les stations.

L'équipe qui a la charge du développement du SI central vous propose l'interface suivante en version 0.1 :



Cette version simplifiée ne gère pas encore le suivi de la flotte de véhicules, mais permettra de prototyper les parties abonnement et facturation.

Le principal cas d'utilisation dont vous avez la charge est donc la phase de prise en charge du véhicule, décrite ici par sa fiche détaillée issue de l'analyse.

Titre : Emprunter véhicule

Acteur : un client AutoLib' porteur d'un badge valide.

Description : Un client emprunte une voiture dans une station AutoLib'.

Hypothèse : La borne de station est correctement connectée au système d'information.

Préconditions : aucune. Au repos le système affiche l'écran d'accueil invitant l'utilisateur à passer son badge sur le lecteur. Le lecteur est actif.

Post-conditions : La location d'une voiture par le client est enregistrée. Le client a pris possession de sa BlueCar. La borne est dans son état de repos nominal.

Scénario :

1. Le client passe son badge sur le lecteur
2. Le système demande son mot de passe
3. Le client saisit son mot de passe
4. Le système affiche le formulaire de santé
5. Le client valide
6. Le système propose de louer une voiture et indique la borne de retrait
7. Le client valide
8. Le système affiche une confirmation pendant 10 secondes, puis revient à l'écran d'accueil nominal.
9. Le système notifie le pilier pour préparer son ouverture par le client.
10. Le client badge le pilier
11. Le système déverrouille le pilier
12. Le client l'ouvre et débranche le véhicule.
13. Le système enregistre la date de prise en charge du véhicule comme début de location.

Exception E1 : Echec authentification

En SN4, si l'identifiant et le login ne correspondent pas, la borne affiche un diagnostic et invite l'utilisateur à réessayer. L'interaction est terminée.

Exception E2 : Pas de voiture disponible

En SN4, s'il n'y pas de véhicules disponibles, la borne affiche la carte des stations disponibles.

E2. 1 L'utilisateur lit la carte et valide.

E2. 2 L'interaction est terminée.

Exception E3 : Abandon utilisateur

En SN3, SN5 ou SN7, l'utilisateur peut choisir d'annuler.

L'interaction est terminée.

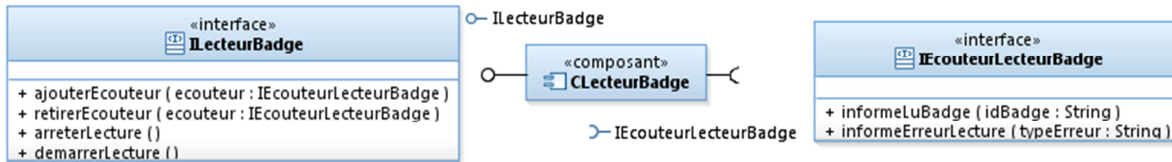
Exception E4 : Timeout borne

En SN9, si l'utilisateur met plus de 5 minutes à activer la borne, la location est annulée.

Exception E5 : mauvais badge

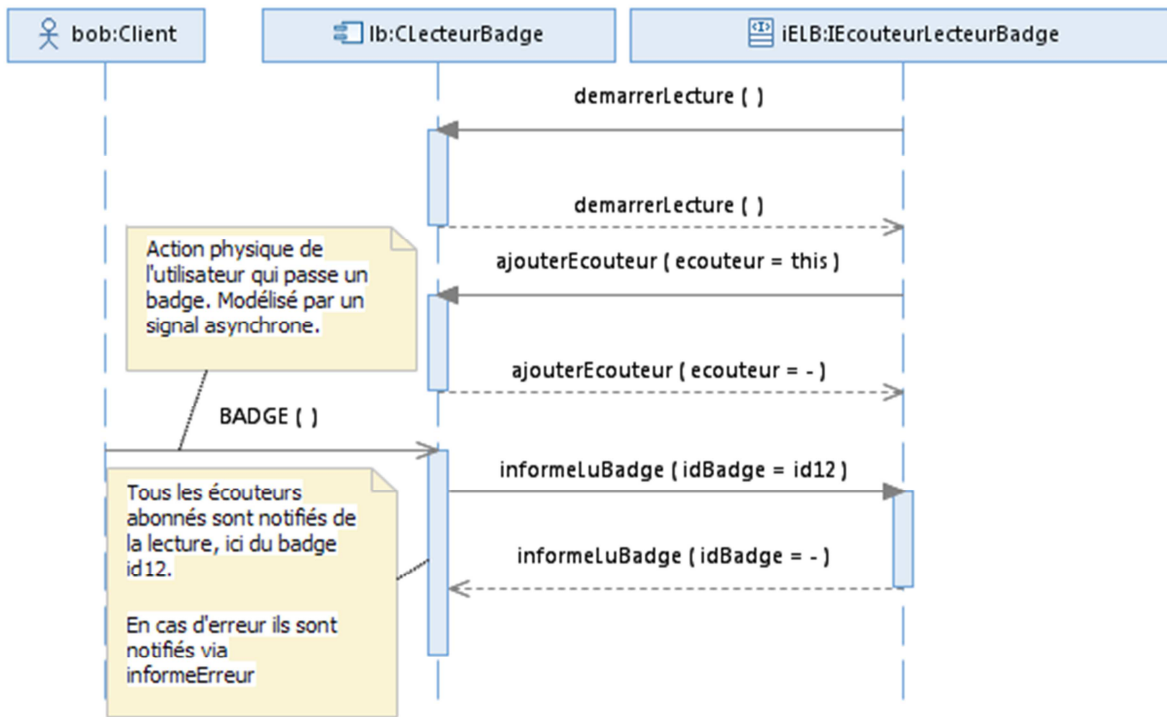
EN SN2, si le badge n'est plus actif ou pas reconnu, le système affiche un diagnostic et l'interaction est terminée.

Question 2.1 : (4 pts) Le composant lecteur de badge est fourni sur étagère. Voici sa spécification :



C'est un composant physique muni d'un capteur RFID, son fonctionnement basé sur le DP Observer est le suivant :

- On peut démarrer ou arrêter la lecture. Quand la lecture est démarrée, le capteur RFID est sensibilisé, tout badge présenté devant le lecteur sera lu. Quand il est arrêté, passer un badge est sans effet.
- Les écouteurs s'abonnent au composant pour être informés quand un badge est passé sur le lecteur actif.
- Quand on passe un badge sur le lecteur actif, il tente de lire son identifiant. Si l'opération est un succès, tous les écouteurs abonnés en sont informés en invoquant « informeLuBadge » avec l'identifiant du badge lu en paramètre. Si l'opération de lecture échoue, c'est l'opération « informeErreurLecture » qui est invoquée, avec un message de diagnostic. Le lecteur reste actif, tant qu'on n'invoque pas arrêterLecture.



- a) **Conception détaillée :** Proposez à l'aide d'un diagramme de classes une réalisation possible en Java de ce composant. On supposera que le signal BADGE() déclenche de façon aléatoire une erreur « désolé » ou une lecture réussie avec l'identifiant « u1 ». On détaillera en pseudo-code ou en Java les opérations (sauf démarrer/arrêter).
- b) A quoi pourrait servir une telle réalisation naïve du composant dans une démarche de développement ?
- c) Ce composant est-il facilement réutilisable dans divers contextes ? Expliquez pourquoi.

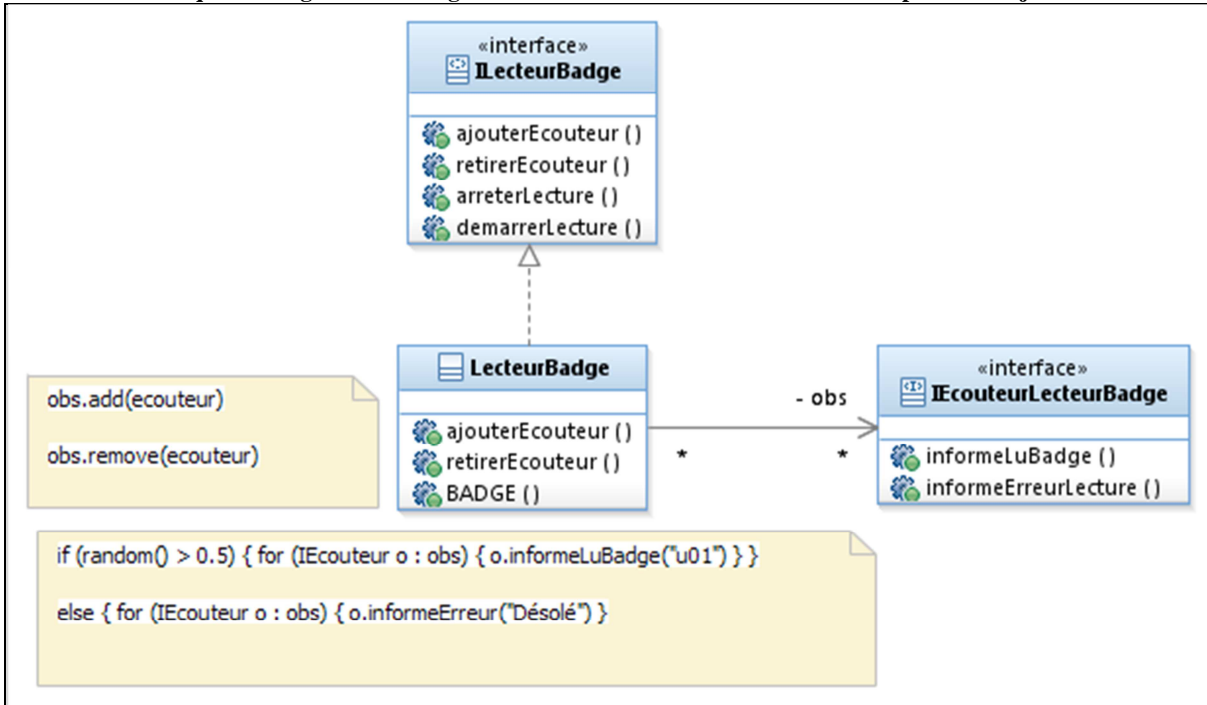
a)

Sur 60% :

20% code de BADGE/ des notifications

15% implémente

25% gestion des écouteurs : 10% pseudo-code add/remove, 15% liste de cardinalité *

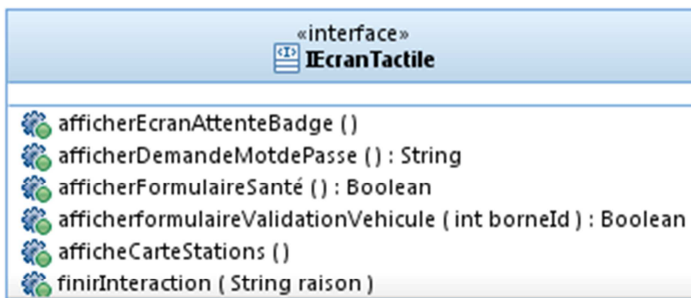


- b) Sur 20% : En tant que composant bouchon, soit pour le dev en parallèle, soit dans le cadre de tests.
- c) Sur 20% : Donc oui, car Observer permet de correctement découpler la source de l'événement des récepteurs. Il n'est peut-être pas « bout de chaîne » car il requiert une itf, mais la dépendance « circulaire » est bien maîtrisée. Donc malgré cette interface requise, le couplage reste faible. On peut parler du fait que dépendance fonctionnelle (sur IELB) != structurel.

Question 2.2 : (1,5 pts)

La borne de la station est constituée d'un écran tactile, d'un lecteur de badge et d'un composant jouant le rôle de contrôleur. Le contrôleur doit interagir avec le système d'information principal pour valider les authentifications et notifier les prises en charge de véhicule.

L'écran tactile et le lecteur de badge sont directement pilotés par le contrôleur de la borne de station. L'écran tactile offre l'interface suivante :



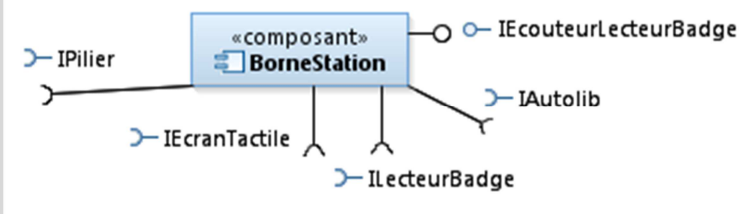
Les écrans sans saisie utilisateur rendent void.

Les écrans avec saisie rendent la valeur saisie, un string pour le mot de passe et un boolean pour les confirmations.

Toutes ces opérations sont susceptibles de lever des TimeoutException si l'utilisateur ne valide pas sa saisie assez vite, ou des AbandonException s'il appuie sur le bouton physique d'annulation.

Les saisies "normales" se font via un clavier tactile.

Représentez le composant ControleurStation (interfaces requises et offertes) à ce stade de la définition des composants.



Barème :25% par interface correcte

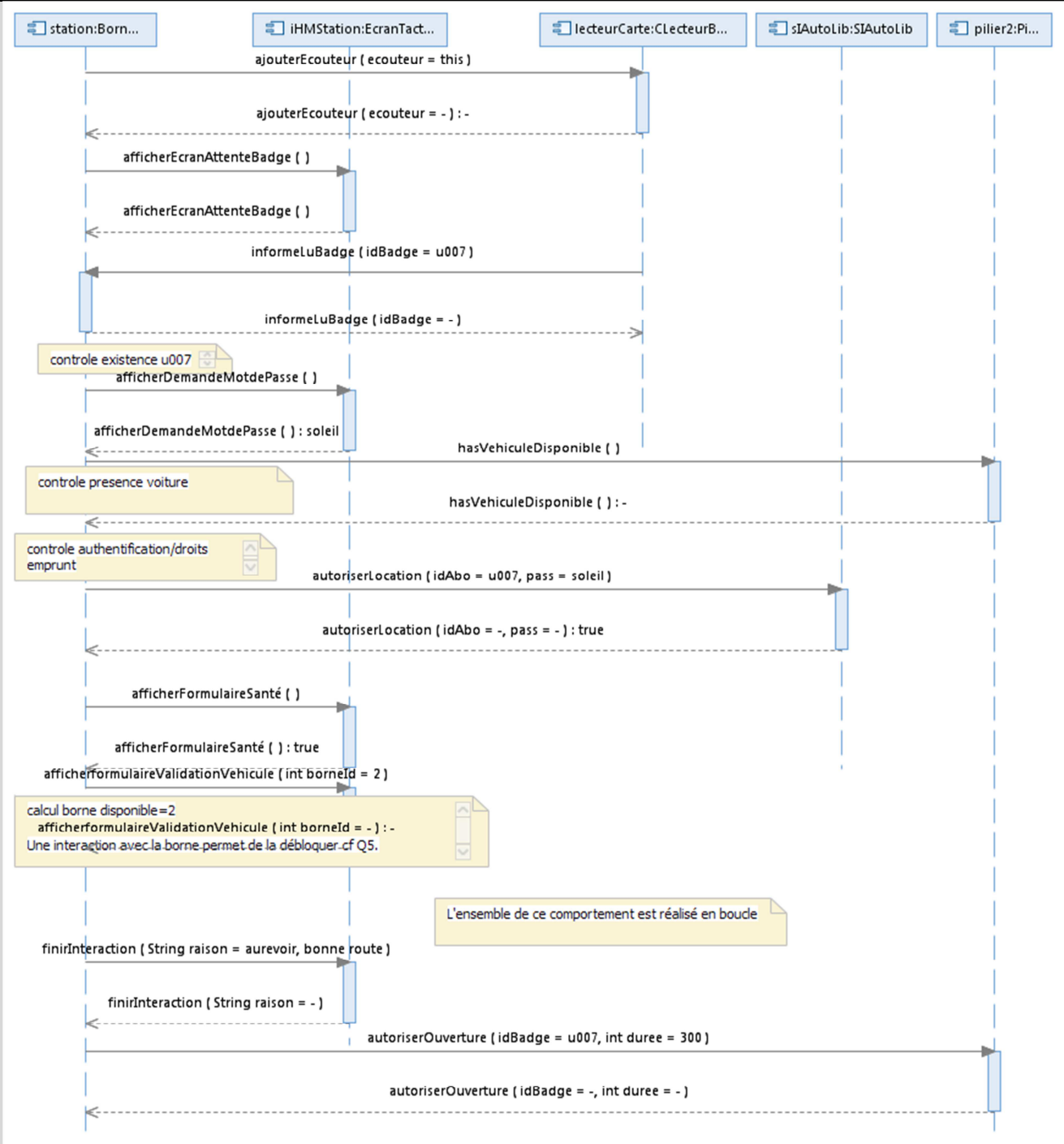
IPilier n'est pas demandé à ce stade.

Question 2.3 : (3,5 pts)

On va se focaliser dans un premier temps sur les étapes 1 à 8 du scenario nominal. L'interaction avec les piliers de chargement sera traitée plus bas. Modélisez sur un diagramme de séquence les échanges entre les 4 composants définis à ce stade pour réaliser les étapes 1 à 8 du scenario nominal. On modélisera également l'initialisation du système (cf. précondition du cas d'utilisation).

On prendra pour les valeurs du scenario le numéro d'abonné « u007 » et le mot de passe « soleil ». On considérera que le pilier numéro 2 est branché à une voiture disponible.

Voici ma correction :



Barème :

Observer : 50% : abonnement/démarrer (15%), demande affichage (10%), luAbonne(u007) correctement orienté du lecteur vers le controleur (25%).

20 % Authentification, au bon moment, avec les bonnes données.

30 % affichages divers dans l'ordre (10% chacun, on peut voir un dernier afficherEcranAccueil ou un commentaire sur la boucle)

** attention le composant pilier n'est pas demandé à cette question.

Question 2.4 (2 points): Les piliers de chargement sont constitués d'un lecteur de badge du même modèle que celui de la borne de station, d'un verrou, et d'une prise de courant uniquement accessible quand le pilier

est déverrouillé. L'ensemble est piloté par un contrôleur de pilier, qui interagit avec le contrôleur de la borne de station.

Les piliers de chargement sont des entités relativement autonomes qui bénéficient d'une connexion directe au SI Autolib. Le composant contrôleur d'un pilier de chargement peut prendre trois états :

1. Aucun véhicule n'est branché, le pilier attend qu'un utilisateur arbitraire utilise son badge pour débloquer le verrou et lui permettre de brancher son véhicule. Il notifie alors le SI Autolib' du branchement en spécifiant la date et le numéro de l'abonné ayant branché le véhicule. Suite à un branchement, le pilier passe à l'état 2.
2. Un véhicule est branché et en charge, le pilier est verrouillé. Il attend de recevoir sur son interface de contrôle un ordre d'ouverture pour un abonné particulier pendant une certaine durée (timeout). Il passe alors à l'état 3.
3. Le pilier attend que l'abonné cité dans l'ordre d'ouverture présente son badge, ce qui déverrouille la borne et lui permet de débrancher le véhicule. Quand le véhicule est effectivement débranché, le pilier notifie le SI Autolib' en lui fournissant le numéro d'abonné et la date. Le pilier revient alors à l'état 1. Si le timeout est rencontré sans que l'utilisateur n'ait badgé le pilier, il revient à l'état 2.

Complétez l'interaction de la question 2.3 jusqu'à l'étape 9 du scenario nominal pour faire apparaître le contrôleur du pilier.

En déduire l'interface (+opérations) qu'offre le pilier de chargement à la borne de station.

Pour le SD, Cf. corrigé 2.3

On doit voir une invocation à débloquerPilier ou quelque chose de ce type.

On peut éventuellement voir le hasVehiculeDispo ou autre pour tester l'état du pilier.

Sur 50% invocation sur le diagramme de séquence

Sur 50% définition de l'interface



80% si bien mais pas d'opération pour tester l'état avant de déverrouiller.

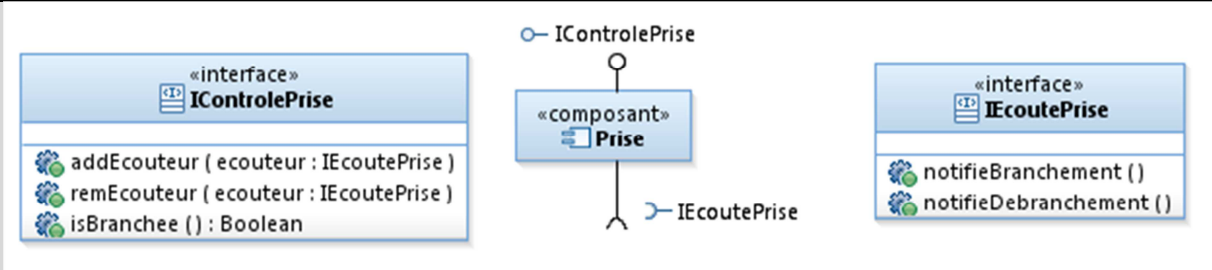
Question 2.5 : (2,5 points)

Le verrou électromécanique du pilier est piloté par son contrôleur. Il est fourni sous la forme d'un composant qui répond à la spécification suivante :



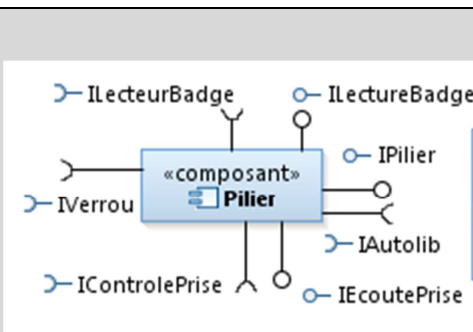
Proposez à l'aide d'un diagramme de composants une spécification du composant représentant la prise de courant, basée sur Observer comme à la Q.2.1. Une prise de courant peut être à un instant donné branchée à un véhicule ou débranchée et l'on peut interroger son état. Un écouteur abonné sur une prise de courant est notifié quand une voiture est branchée ou débranchée.

On veut voir Observer ou une variante dessus.



- 30 % opération d'abonnement
- 20% opération de test de l'état
- 30% interface listener
- 20% le composant et ses deux interfaces offerte/requise.

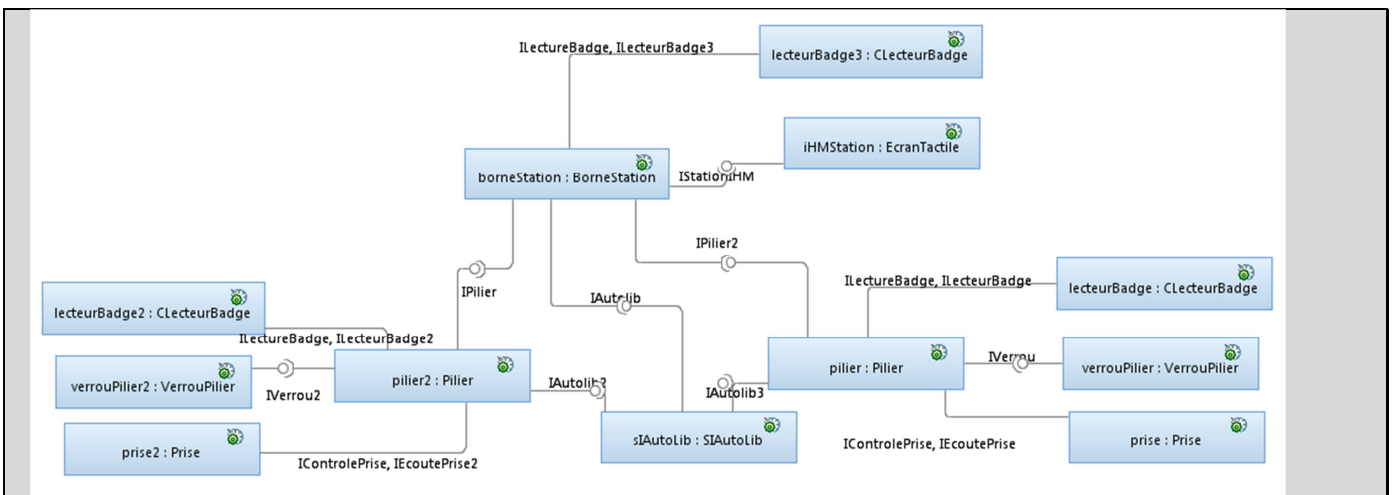
Question 2.6 : (1,5 points) Représenter la spécification du composant ControleurPilier (interfaces requises et offertes).



Barème :

- 25% requis/offert LecteurBadge
- 25% requis/offert sur la prise.
- 15% dépendance sur IVerrou
- 15% offre IPilier
- 15% utilise IAutoLib
- 100% : tout bien (donc la première faute compte au moins -20, vu que mon total fait 95%).

Question 2.7 (3 points): Modélisez à l'aide d'un diagramme de structure interne une topologie d'instanciation nominale où une station qui comporte deux piliers de chargement est connectée au SI d'AutoLib.



Barème : Pour avoir les points sur les liens, l'orientation doit être correcte. / !\ les boucles (Observer) donnent deux liens sur les diagrammes plutôt que le lien avec « IA,IB » que fait l'outil dans ce corrigé.

30% Borne Station connectée à son lecteur (20% bidirectionnel) et son IHM (10%)

10% on a deux instances de piliers symétriques.

10% ces instances sont liées à la borne Controleur de station via IPilier.

30% liens du pilier sur le verrou, un lecteur qui lui est propre, une prise.

20% liens de la borne sur Autolib, des piliers sur Autolib. Il ne doit y avoir qu'une instance du SI.