

Pierre Lefebvre

Objectifs du TP

- Comprendre la différence entre une analyse et une conception UML sur un projet
- Générer du code Java à partir d'un modèle UML (forward)
- Remonter le code Java dans le modèle UML (reverse)
- Placer les méthodes grâce au diagramme de séquence
- Générer un diagramme de séquence à l'aide de l'outil

SUJET

La direction d'une société souhaite améliorer la sécurité dans leur immeuble et sur leur site, sans contrarier leurs employés. Elle voudrait aussi empêcher les personnes qui ne font pas parti de la compagnie, d'utiliser leur parking.

Il a été décidé de délivrer des cartes d'identité à tous leurs employés, qu'ils devront porter sur le site. Les cartes indiquent le nom, le département, et numéro de membre du personnel, et permettent l'accès au parking de la société.

Une barrière et un lecteur de carte sont placés à l'entrée du parking de voitures. Un capteur est positionné en entrée et en sortie du parking pour déterminer si une voiture est aperçue. Le conducteur d'une voiture s'approchant du parking introduit sa carte numérotée dans le lecteur de carte, celui-ci vérifiant que le numéro de la carte est bien connu du système. Si la carte est reconnue, le lecteur envoie un signal pour lever la barrière et la voiture peut entrer dans le parking.

A la sortie, il y a aussi une barrière, qui est levée quand une voiture souhaite quitter le parking. Quand il n'y a plus de places dans le parking, un indicateur à l'entrée affiche « plein » et est remis en position « off » quand une voiture sort du parking.

Des cartes spéciales « visiteur », qui indiquent un numéro et la date courante, permettent aussi d'accéder au parking de voitures.

Analyse du projet

- 1) Créer un diagramme de cas d'utilisation représentant les besoins du système (c'est très simple !)
- 2) Développer des scénarios pour chacun des cas d'utilisation.
- 3) A partir des scénarios, identifier les classes candidates et construire un modèle objet de l'étude de cas. Il est conseillé de créer un paquetage racine de l'application (ex: parking) et d'y mettre les classes que vous allez définir. Pour l'instant, ne vous attardez pas sur les méthodes qui seront rajoutées par l'étude du modèle dynamique.
- 4) Développer un diagramme de séquence pour représenter qu'un conducteur entre dans le parking.
Pour cela, éditer un diagramme de séquence « entrer dans le parking » à partir du cas d'utilisation « entrer dans le parking »
- 5) Développer un diagramme de séquence pour représenter qu'une carte d'un conducteur n'est pas reconnue. Pour cela, éditer un diagramme de séquence « la carte n'est pas reconnue » à partir du cas d'utilisation « entrer dans le parking »
- 6) Faire un diagramme d'état pour la classe « Barriere »
- 7) Faire un diagramme d'état pour la classe « Parking »

Conception du projet

1) Une classe « Simulateur » va permettre de simuler une voiture qui arrive ou une voiture qui part en tirant aléatoirement une valeur par le biais de méthodes statiques :

- si la valeur est inférieure ou égale à 0.7 => une voiture arrive
- si la valeur est supérieure à 0.7 => une voiture part
- si la valeur est inférieure à 0.5 => une voiture est présente

De même, cette classe permet de récupérer aléatoirement un numéro de carte pour permettre de simuler si une carte est valide ou non. Il faudra bien entendu développer un ensemble de cartes valides dont le numéro tiré aléatoirement sera comparé à celui-ci.

- Faire un diagramme de conception permettant de gérer le scénario « entrer dans le parking ». Il faudra intégrer la classe « Simulateur » dans le diagramme. Chaque message reçu dans une classe sera converti en une méthode dans la classe réceptrice du message.
- Faire un diagramme de conception permettant de gérer le scénario « sortir du parking »
- Finaliser le diagramme de classes en plaçant les attributs et les méthodes manquantes.
- Développer les classes une par une en s'appuyant sur le code généré. Le résultat de la simulation doit aboutir à une exécution dont voici un exemple :

simulation d'un parking (nb de places maximum : 3)

place occupe = 0

la carte n'est pas reconnue

place occupe = 0

la carte est reconnue

la barriere d'entree se lève

la barriere d'entree se baisse

place occupe = 1

il reste de la place

place occupe = 1

la carte est reconnue

la barriere d'entree se lève

la barriere d'entree se baisse

place occupe = 2

il reste de la place

une voiture part

la barriere de sortie se lève

la barriere de sortie se baisse

il reste de la place

place occupe = 1

la carte est reconnue

la barriere d'entree se lève

la barriere d'entree se baisse