

La gare

Objectifs

Vous allez simuler le fonctionnement d'une gare. Les comportements à simuler ne sont pas toujours réalistes mais permettent de simplifier la modélisation afin de résoudre différents problèmes de synchronisation. La modélisation se fera en trois étapes, chacune étant l'objet de l'une des sections suivantes. Dans les trois étapes, la modélisation comprend les éléments suivants :

- Un **espace-quai**, composé d'un nombre limité de voies, chaque voie pouvant accueillir un seul train. La gare ne comporte qu'un seul espace-quai.
- Des **trains**, au nombre limité par `NB_TRAINS`, qui arrivent en gare (sur une voie de l'espace-quai), restent un temps limité en gare pour laisser monter les voyageurs, et repartent.
- Des **voyageurs**, au nombre limité par `NB_VOYAGEURS`, qui doivent dans l'ordre : 1) acheter un billet et 2) monter dans un train.
- Un **espace-vente**, composé de guichets qui vendent les billets. Chaque billet peut être vendu par n'importe quel guichet.

Première étape de modélisation

Dans cette première partie, on fera les hypothèses suivantes :

- L'espace-quai a un nombre de voies fixé à `NB_VOIES`. Si une voie est libre, alors un train peut entrer en gare.
- Il n'y a qu'une seule destination. Tous les trains y vont et tous les clients veulent y aller.
- L'espace-vente n'a qu'un seul guichet, et le nombre de billets n'est pas borné. Imprimer un ticket prend un temps `IMPRESSION_TICKET`.
- Un train : 1) arrive en gare en `10000 / VITESSE_TRAIN` (ms) où `VITESSE_TRAIN` est compris entre 50 et 300 (km/h), 2) cherche à se garer sur une voie libre, 3) s'arrête un temps `ARRET_TRAIN`, permettant à des voyageurs munis de billet de monter, 4) repart. Il a un nombre de places libres qui sera fixé avant son arrivée en gare, et qui sera compris entre 0 et `CAPACITE_TRAIN`.
- Un voyageur munis d'un billet montent dans un train non plein dès que possible.

Seconde étape de modélisation

Afin de rendre la modélisation plus réaliste, nous allons multiplier les guichets. Ensuite, nous allons limiter le nombre de billets mis en vente en prenant en compte le nombre de places disponibles dans les trains qui arrivent en gare (au moment où ils arrivent). Ainsi, les comportements à simuler sont les suivants :

- Une fois garée sur sa voie, un train annonce son nombre de places disponibles. Ce nombre est ajoutée au nombre de billets mis en vente.
- Lorsqu'un client obtient un billet, ce billet correspond a une place dans un train en particulier, bien identifié, qui a encore des places disponibles.

- A l'expiration du temps `ARRET_TRAIN`, un train ne part plus tout de suite, mais stoppe la vente des billets pour ce train (s'il en restait). Avant de partir, le train doit attendre que tous les voyageurs ayant acheté un billet pour ce train soient effectivement montés à son bord.

Troisième étape de modélisation

Nous considérons maintenant trois gares, A , B , et C . Ces trois gares vendent des billets pour tous les trajets possibles, à savoir les trajets $A - B$, $B - A$, $B - C$, $C - B$, $A - C$ et $C - A$. Chaque billet vendu est associé à l'un de ces six trajets. Chaque client commence par acheter un billet dans n'importe quelle gare, puis se rend dans la gare de départ correspondant à son billet. Chaque train effectue aussi un trajet choisi parmi les six possibles. Une fois le trajet choisi, il se comporte comme à la deuxième étape de simulation.

Un serveur de *billetterie* comptant le nombre de billets disponibles pour chacun de ces trajets est accédé par les trois gares de façon concurrente pour la vente des billets. Chaque vente décrémentant le nombre de billets disponibles, chaque arrivée de train l'incrémentant.

La *billetterie* exposera une API REST permettant :

- de lister tous les trajets ainsi que les places disponibles au moment de la requête,
- à un guichet de prendre un billet pour un trajet donné,
- à un train d'ajouter un nombre de place disponible à son trajet.

Instructions

Le TP doit être rendu au plus tard le vendredi 28 novembre 2014 (23h59, CET) adressé à votre chargé de TP, et dont l'objet sera [TP-CSR Gr<xx>] <Nom1> <Nom2>¹. Ce courrier devra contenir :

1. Les sources Java des trois étapes de simulation (commentées et indentées correctement).
2. Des traces d'exécution explicites
3. Un rapport au format PDF contenant :
 - (a) la décomposition en classes de votre modélisation, avec l'identification des threads et des objets partagés
 - (b) la liste des problèmes de synchronisation rencontrés ainsi que leur résolution
 - (c) la description de l'API de la *billetterie*,
 - (d) toutes les informations que vous jugerez nécessaires à l'exécution et à la compréhension de votre modélisation et de votre code.

1. Merci de respecter ce format, permettant de retrouver votre travail dans le flot de mails.