

Cinéma

1 Simulation parallèle

On vous commande une application de simulation du fonctionnement d'une salle de cinéma. Ce cinéma est composé d'une seule billetterie et d'une seule salle de projection. L'objectif principal de cette simulation est de cerner les problèmes de synchronisation inhérent à un cinéma, dont nous allons simplifier le fonctionnement pour éviter de développer un programme trop complexe. Vous allez devoir considérer les éléments suivants:

- **Le cinéma** représente l'entité globale. Il constituera votre programme principal. Il est composé d'une billetterie et d'une salle de projection, qui projette le même film, à la suite.
- **La billetterie:** les clients se présentent à la billetterie et achètent un ticket. L'ordre de vente aux clients est anarchique, mais la billetterie ne peut délivrer qu'un seul ticket à la fois. Le nombre de billets est limité. S'il y a plus de clients que de billets, les clients qui n'arrivent pas à se procurer de tickets rentrent chez eux, et la simulation se termine pour eux. Le nombre de billets est typiquement un multiple du nombre de sièges dans la salle. En d'autres termes, le client n'achète pas de billet pour une séance en particulier. Une fois muni d'un billet, il essaiera de rentrer dès que possible dans la salle. S'il n'y a plus de places pour la séance à venir, il devra simplement attendre la séance suivante.
- **La salle de projection** est initialement dans l'état **CLOSED**. Ensuite, elle passe par plusieurs états, dans cet ordre:
 1. **OPEN.** Cet état peut être simulé par une durée de 3 secondes. Lorsque la salle est dans cet état, les spectateurs munis d'un billet peuvent y rentrer (et seulement à ce moment là.), dans la limite du nombre de sièges de la salle. Les spectateurs munis d'un billet qui n'auront pas trouvé de siège, attendront la séance suivante.
 2. **PROJECTING.** Cet état simule la projection du film, et peut être simulée par une durée de 5 secondes. Pendant ce temps, aucun spectateur ne peut ni rentrer ni sortir de la salle (Ce n'est pas réaliste, mais cela permet de simplifier les choses.)
 3. **EXITING.** A la fin de la séance, et seulement à ce moment là, les spectateurs peuvent sortir de la salle.
 4. **CLEANING.** Une fois tous les spectateurs sortis de la salle (et uniquement à ce moment là), l'employé du cinéma peut à son tour rentrer dans la salle pour la nettoyer et la remettre en ordre, en vue de la séance suivante. Cet état peut être simulé par une durée de 2 secondes. Une fois le nettoyage effectué, la salle peut repasser dans l'état **OPEN**.

- **Le client:** lorsqu'il arrive au cinéma, le client commence par aller à la billetterie où il achète un ticket pour la séance. Soit il reste un billet et il essaie de rentrer dans la salle (lorsqu'elle se trouve dans l'état **OPEN**), soit il rentre chez lui. Un client restera dans la salle jusqu'à la fin de la projection, puis sortira de la salle. Si un client n'arrive pas à rentrer pour la première séance, il devra attendre que la salle repasse dans l'état **OPEN** et réessaiera ainsi jusqu'à voir le film.
- **L'employé.** L'employé du cinéma est responsable de la mise en œuvre des états de la salle de projection. Il devra rester actif jusqu'à ce que tous les clients aient pu voir le film et effectuera en boucle l'algorithme suivant:

1. ouvrir la salle
2. lancer la projection
3. attendre la sortie des spectateurs
4. nettoyer la salle

Une fois tous les clients partis, il pourra *se terminer*.

2 Code et rapport

Avant de partir sur le code, vous veillerez à prendre votre temps pour concevoir votre solution. Mettez en place des affichages sur la sortie standard pour vous permettre de suivre en détail ce qu'il se passe dans le cinéma. N'oubliez pas de commenter votre code. Vous utiliserez les outils de synchronisation de votre choix.

En plus des sources de ce TP, vous devrez rédiger un rapport (max 8 pages) décrivant vos choix de conception. Vous pourrez inclure un diagramme des classes. Vous devrez aussi indiquer quels éléments sont des threads, lesquels sont des objets partagés. Vous devrez expliquer tous les problèmes de synchronisation que vous aurez rencontré, et expliquer la solution que vous avez mise en place pour les résoudre.

Rendu

Le rendu est à envoyer par mail à votre chargé de TP, dans une archive nommée `NOM1_NOM2.zip` dont la décompression créera un dossier respectant lui aussi cette convention de nommage. Ce dossier comportera votre rapport et vos fichiers sources Java.

Vous devez rendre votre travail pour le 15 Décembre 2023 23h59. A partir d'une heure de retard, vous encourez des points de pénalités proportionnels au retard.