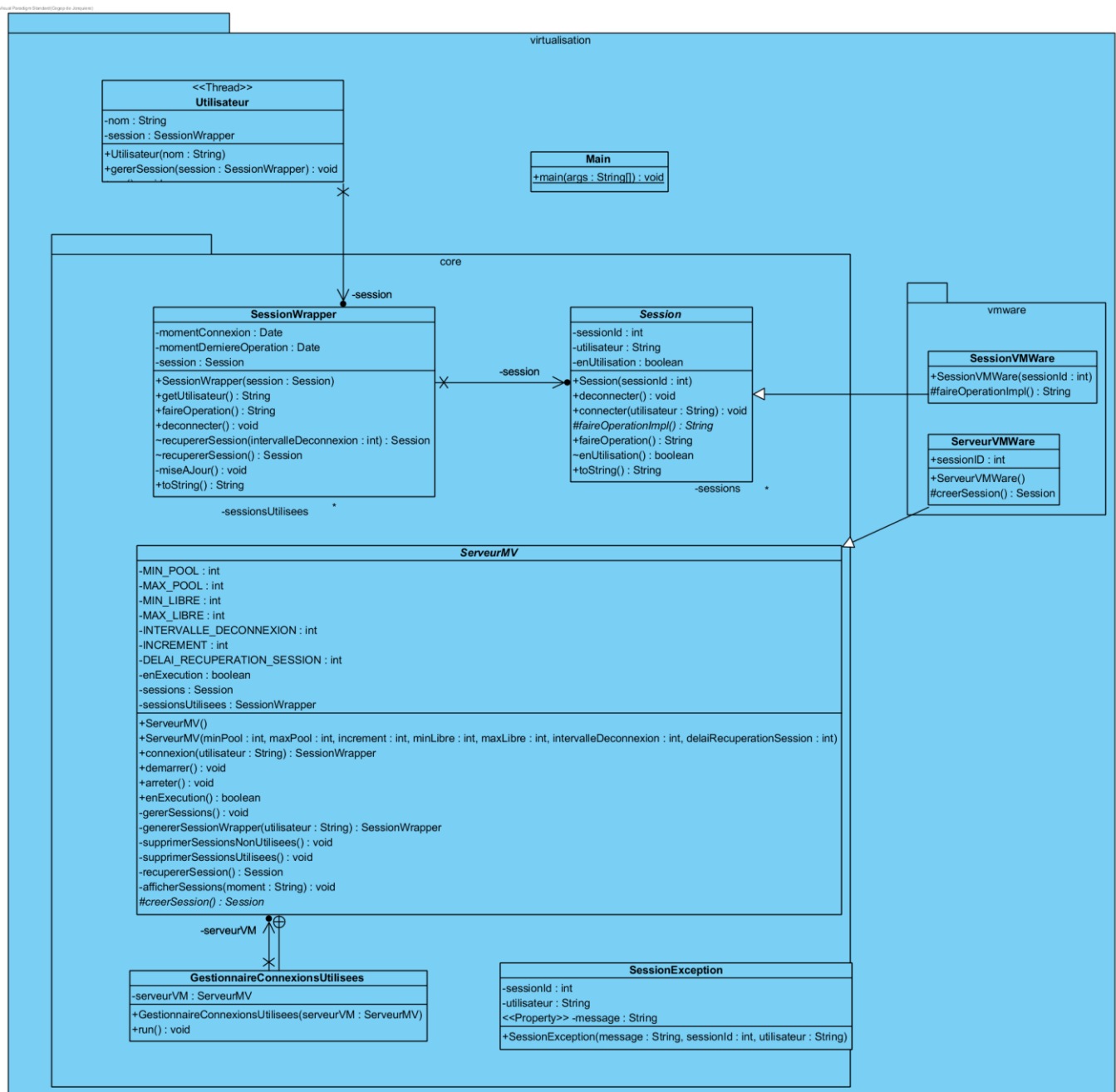


### Patron #10 : Pool



## Mise en contexte

Les ressources sont souvent limitées et ils doivent être partagées. Supposons un serveur qui distribue des machines virtuelles mais qui est limité par le nombre de licences concurrentes. Lorsque plusieurs utilisateurs demandent des sessions de travail, le serveur doit gérer la distribution et la récupération des sessions.

Voici la liste des paramètres permettant cette gestion

- MIN\_POOL : Le nombre minimum de sessions
- MAX\_POOL : Le nombre maximum de sessions
- MIN\_LIBRE : Le nombre minimum de sessions non utilisées (libres)
- MAX\_LIBRE : Le nombre maximum de sessions non utilisées (libres)
- INTERVALLE\_DECONNEXION : Intervalle après laquelle la session est « libérées » automatiquement par le serveur
- INCREMENT : Le nombre de sessions à ouvrir (créer) à chaque nouveau besoin (optionnel)
- DELAI\_RECUPERATION\_SESSION : Intervalle de vérification des sessions utilisées

### Contraintes

À partir du modèle et du projet fourni

- Vous devez programmer la classe « ServeurVM »
- Le modèle UML de la classe ServeurVM (incluant la classe interne) est seulement fourni à titre indicatif. Vous avez l'entière liberté de l'implantation de la classe tant que les autres classes du modèle ne sont pas modifiées.
- Petite exception : Vous pouvez modifier/modifier la journalisation des autres classes afin « d'améliorer » la sortie en console
- Des exemples d'exécution sont fournis