

Projet :  
Projet systèmes concurrents/intergiciels

## Table des matières

---

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Membres de groupe . . . . .	2
1.2	Présentation . . . . .	2
1.3	Gradle . . . . .	2
<b>2</b>	<b>Version <i>mémoire partagée</i></b>	<b>2</b>
2.1	Version naïve . . . . .	2
2.2	Version multithreadée . . . . .	2

Remarque : Ceci est le document réponse, il ne reprend pas l'intégralité du sujet.

## 1. Introduction

### 1.1. Membres de groupe

Nous sommes trois étudiants du groupe M2 :

- ❑ Ying LIU
- ❑ Philippe NEGREL-JERZY
- ❑ Sébastien PONT

### 1.2. Présentation

Nous ne représentons pas tout le sujet (que vous pouvez retrouver [ici](#)<sup>1</sup>), mais voici un résumé.

Linda est un service permettant de partager des données sous formes de `Tuple`. Dans ce projet nous allons implémenter deux manières de partager et gérer ces ressources à travers plusieurs clients :

- une version dite "locale" à base de mémoire partagée (`shm` package Figure 1)
- une version dite "distante" à base de clients / monoserveur (`server` package Figure 1)

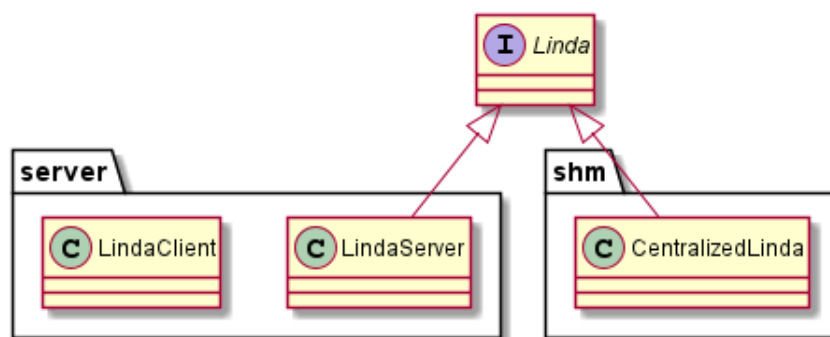


FIGURE 1 – Diagramme de classe général du projet

- ❑ Version mémoire partagée (version locale) : Chaque client est un nouveau *thread*.
- ❑ Version client / monoserveur (version distante) : Un serveur tourne et communique via *RMI* aux différents clients.

### 1.3. Gradle

Pour simplifier la gestion des dépendances et faciliter le développement nous avons choisi d'utiliser l'outil *Gradle*<sup>2</sup>. Ainsi, pour compiler le programme, il suffit par exemple de taper dans un terminal à la racine du projet :

```
</> Lancer l'application java via Gradle
```

```
./gradlew run
```

## 2. Version mémoire partagée

### 2.1. Version naïve

### 2.2. Version multithreadée

1. <https://spont.me/mjxoog>

2. [gradle.org](https://gradle.org)