

**UNIVERSITE DE THIES**  
**UFR SCIENCES ET TECHNOLOGIES (SET)**  
**DEPARTEMENT INFORMATIQUE**



---

**MASTER INFOMATIQUE : OPTION GENIE LOGICIEL**

# **PROJET DE SYSTEME DISTRIBUE**

---

**Problème de carrefour**

**Par**

**Habibou DIEDHIOU**  
**El Hadji Code NDIAYE**

**Etudiants en deuxième année de Master informatique**  
**Option : génie logiciel**

**Encadreur : Dr. Cheikhou THIAM**

**ANNEE ACADEMIQUE 2019 - 2020**

## **SOMMAIRE**

<a href="#"><u>Introduction</u></a> .....	3
<a href="#"><u>I. Problématique</u></a> .....	3
<a href="#"><u>II. Résolution</u></a> .....	3
<a href="#"><u>III. Démonstration</u></a> .....	4
<a href="#"><u>Conclusion</u></a> .....	8

## Introduction

Un système distribué est un ensemble d'entités autonomes de calcul (ordinateurs, processeurs, processus, processus léger etc.) interconnectées et qui peuvent communiquer.

Ainsi dans le cadre de la mise en pratique des concepts vus en systèmes distribués et multi-tiers, une tâche nous a été confiée à savoir la création d'une application Java permettant de gérer une situation conflictuelle de circulation de véhicule.

### I. Problématique

Cet exercice consiste à gérer le passage des véhicules dans un tronçon. L'accès de ce dernier est réglementé et circule dans un seul sens selon la direction du véhicule engagé.

Si un véhicule A veut accéder au segment alors qu'il y a au moins un autre véhicule venant du sens inverse sur le tronçon, alors ce véhicule A doit attendre jusqu'à que le tronçon soit libéré.

### II. Résolution

Pour résoudre ce problème et proposer une solution, nous avons fait recours au langage de programmation Java qui est un langage adéquat pour la gestion des flux de message, des processus etc.

Voici les tables de la modélisation de notre scénario :

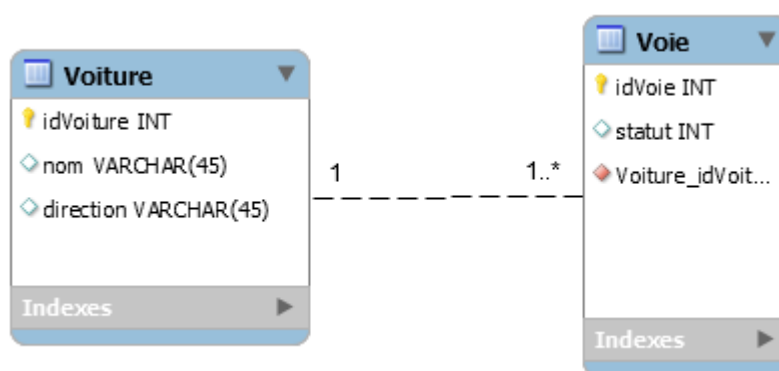


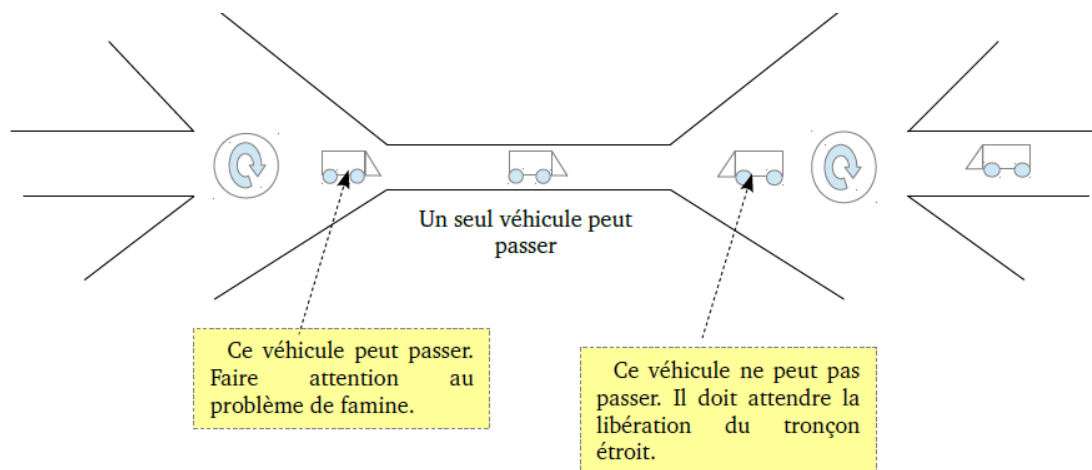
Figure 1: Tables

## Explication

La classe voiture va permettre d'instancier une voiture avec des attributs lui permettant d'identifier son nom et sa direction ou sens de circulation. Elle a le rôle de client dans cet exemple. La voiture envoie un message pour demander l'accès à la voie et attend une réponse pour s'engager ou attendre que sa réponse soit acceptée. Si sa demande est rejetée alors elle renvoie la demande, ce jusqu'à ce que sa demande soit acceptée. Si sa réponse est acceptée, elle s'engage sur la voie selon une vitesse aléatoirement définie et libère la voie.

La classe Voiture est reliée avec la classe Voie qui représente la route. Cette dernière a un ID qui l'identifie, un statut qui désigne si elle est libre ou occupée. La voie joue le rôle de serveur dans notre cas. Elle reçoit les demandes des véhicules et renvoie une réponse pour autoriser ou refuser l'accès la voie selon le sens de la voiture déjà engagée sur la voie.

Ce scenario est décrit dans l'image l'énoncé de l'exercice



## III. Démonstration

Voici le déroulement de la logique du scenario exécuté par la console :

```
[Voiture droite 0 ]: Je débute !
[Voiture gauche 0 ]: Je débute !
[Voiture droite 1 ]: Je débute !
[Voiture gauche 1 ]: Je débute !
[Voiture droite 2 ]: Je débute !
[Voiture gauche 2 ]: Je débute !
[Voiture droite 3 ]: Je débute !
```

[Voiture gauche 3 ]: Je débute !  
[Voiture droite 4 ]: Je débute !  
[Voiture gauche 4 ]: Je débute !  
[Voiture droite 2 ]: Je demande à rentrer  
[Voirie] :Voiture droite 2 accepté  
Voiture engagée -> direction [droite]  
[Voiture droite 2 ]: Je viens d'entrer  
[Voiture droite 0 ]: Je demande à rentrer  
[Voirie] :Voiture droite 0 accepté  
Voiture engagée -> direction [droite]  
[Voiture droite 0 ]: Je viens d'entrer  
[Voiture droite 2 ]: Je demande à sortir  
Voirie :[Voiture droite 2 ] est sortie, -> direction [droite]  
[Voiture gauche 4 ]: Je demande à rentrer  
[Voirie] :Voiture gauche 4 accepté  
Voiture engagée -> direction [gauche]  
[Voiture gauche 4 ]: Je viens d'entrer  
[Voiture droite 0 ]: Je demande à sortir  
Voirie :[Voiture droite 0 ] est sortie, -> direction [droite]  
[Voiture gauche 2 ]: Je demande à rentrer  
[Voirie] :Voiture gauche 2 accepté  
Voiture engagée -> direction [gauche]  
[Voiture gauche 2 ]: Je viens d'entrer  
[Voiture gauche 4 ]: Je demande à sortir  
Voirie :[Voiture gauche 4 ] est sortie, -> direction [gauche]  
[Voiture gauche 3 ]: Je demande à rentrer  
[Voirie] :Voiture gauche 3 accepté

Voiture engagée -> direction [gauche]

[Voiture gauche 3 ]: Je viens d'entrer

[Voiture droite 1 ]: Je demande à rentrer

Voirie : Voiture droite 1 refusée

[Voiture gauche 1 ]: Je demande à rentrer

[Voirie] :Voiture gauche 1 accepté

Voiture engagée -> direction [gauche]

[Voiture gauche 1 ]: Je viens d'entrer

[Voiture gauche 2 ]: Je demande à sortir

Voirie :[Voiture gauche 2 ] est sortie, -> direction [gauche]

[Voiture droite 4 ]: Je demande à rentrer

[Voirie] :Voiture droite 4 accepté

Voiture engagée -> direction [droite]

[Voiture droite 4 ]: Je viens d'entrer

[Voiture droite 0 ]: Je demande à rentrer

[Voirie] :Voiture droite 0 accepté

Voiture engagée -> direction [droite]

[Voiture droite 0 ]: Je viens d'entrer

[Voiture droite 3 ]: Je demande à rentrer

[Voirie] :Voiture droite 3 accepté

Voiture engagée -> direction [droite]

[Voiture droite 3 ]: Je viens d'entrer

[Voiture gauche 3 ]: Je demande à sortir

Voirie :[Voiture gauche 3 ] est sortie, -> direction [gauche]

[Voiture gauche 2 ]: Je demande à rentrer

[Voirie] :Voiture gauche 2 accepté

Voiture engagée -> direction [gauche]

[Voiture gauche 2 ]: Je viens d'entrer  
[Voiture gauche 1 ]: Je demande à sortir  
Voirie :[Voiture gauche 1 ] est sortie, -> direction [gauche]  
[Voiture droite 4 ]: Je demande à sortir  
Voirie :[Voiture droite 4 ] est sortie, -> direction [droite]  
[Voiture droite 0 ]: Je demande à sortir  
Voirie :[Voiture droite 0 ] est sortie, -> direction [droite]  
[Voiture gauche 0 ]: Je demande à rentrer  
[Voirie] :Voiture gauche 0 accepté  
Voiture engagée -> direction [gauche]  
[Voiture gauche 0 ]: Je viens d'entrer  
[Voiture droite 1 ] :direction [droite] -> Je redemande à rentrer  
Voirie : Voiture droite 1 refusée  
[Voiture droite 3 ]: Je demande à sortir  
Voirie :[Voiture droite 3 ] est sortie, -> direction [droite]  
[Voiture gauche 2 ]: Je demande à sortir  
Voirie :[Voiture gauche 2 ] est sortie, -> direction [gauche]  
[Voiture droite 2 ]: Je demande à rentrer  
[Voirie] :Voiture droite 2 accepté  
Voiture engagée -> direction [droite]  
[Voiture droite 2 ]: Je viens d'entrer  
[Voiture gauche 0 ]: Je demande à sortir  
Voirie :[Voiture gauche 0 ] est sortie, -> direction [gauche]  
[Voiture gauche 4 ]: Je demande à rentrer  
[Voirie] :Voiture gauche 4 accepté  
Voiture engagée -> direction [gauche]  
[Voiture gauche 4 ]: Je viens d'entrer

## **Conclusion**

Ceci est une petite démonstration de l'utilisation des flux de message et de la gestion des processus sous le langage Java. Il est possible de d'améliorer afin d'en faire de petite simulation graphique. Java contient d'autres composants permettant de gérer l'échange des messages et les flux d'information qui sont les sockets. Ces derniers sont surtout liés à la programmation réseau en Java.



## **Table des matières**

Introduction .....	3
I. Problématique .....	3
II. Résolution .....	3
III. Démonstration.....	4
Conclusion.....	8