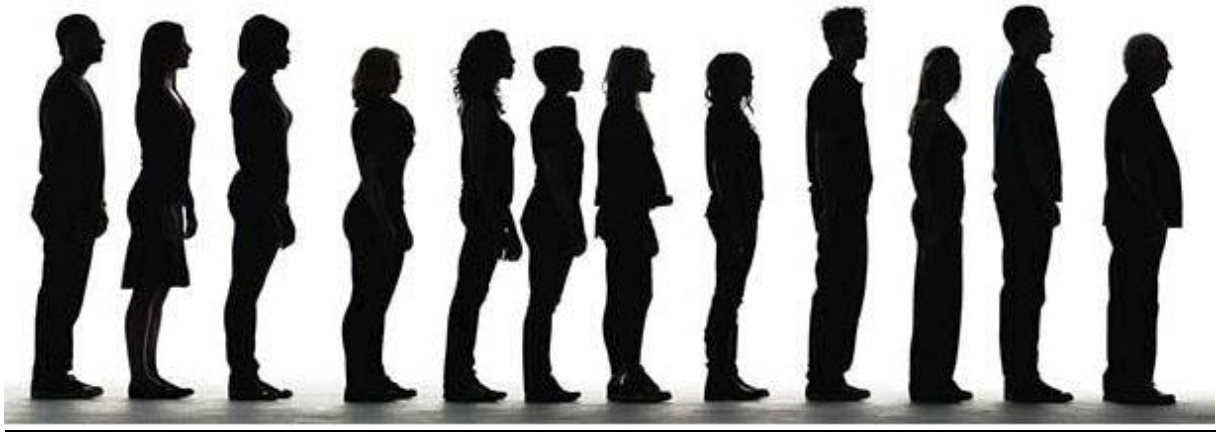


# **Dossier de recherche opérationnelle : les files d'attente**



**Par Nicolas Bauthier**

# **Introduction**

Le but du programme est de réaliser de manière informatique une simulation de files d'attentes, dont les règles sont définies dans l'énoncé joint au dossier.

J'ai décidé de programmer cette simulation en java (et donc, en orienté objet). Les diagrammes d'action ne permettant pas de représenter certaines subtilités de l'orienté objet, beaucoup de divergences peuvent être relevées entre ceux-ci et le code du programme. Dans la première partie de ce dossier, vous trouverez les entrées et sorties de la situation telle que modélisée sur le diagramme d'action. Puis, une autre section expliquera quels sont les objets que j'ai dégagés pour passer d'un diagramme d'action (modélisation procédurale) à une approche orienté objet.

Afin de me fondre dans la communauté java internationale, les variables et les noms de méthodes de mon programme sont en anglais.

Ces précisions indiquées, je vous souhaite une bonne lecture !

# Les entrées et les sorties

## Les entrées

L'utilisateur n'a rien à entrer pour utiliser le programme. Les valeurs utilisées pour la simulation sont déjà initialisées dans le programme. Le nombre minimum de stations imaginé est de 15, et le nombre maximum est de 30.

## Les sorties

-Un fichier « **sortiesDebut.txt** » contenant, pour la première valeur de stations envisagée (15) et pour les 20 premières minutes de simulation les renseignements suivants :

- **nbArrivées1520** : le nombre d'arrivées dans le système observé durant cette période

- pour chaque minute écoulée (jusqu'à 20)

- un bilan de l'état du tableau des stations en début et en fin de minute, exprimé sous forme d'un tableau dans le fichier : par station, Le type de priorité du client servi (**Client.typePriorité**) et la durée qu'il lui reste à passer dans la station (**Client.duréeService**) avant que sa requête ne soit traitée doivent y figurer.

- un bilan de l'état des files en début et en fin de minute : pour chacune des 3 files, un tableau reprenant les renseignements suivants, pour chaque client composant la file : **Client.typePriorité** et **Client.duréeService**.

-Un fichier « **resultatSimulations.txt** » contenant les renseignements suivants :

- Dans un tableau de 16 cellules (reprenant les tests pour un **nbStations** variant de 15 à 30 :

- nbStations** : Le nombre de stations ouvertes pour cette partie de la simulation.

- coutTotal** : Le coût total qu'a coûté un tel système après 960 minutes de simulation.

## Les structures

### La structure Client

Dans ce programme, les clients seront des structures représentés par les attributs suivants :

-**tempsSyst** correspond au temps en minutes que le client a déjà passé dans le système, c'est-à-dire le temps qu'il a passé dans les files d'attente et dans les stations. A la création du client (au moment où il rejoint le système), ce temps est initialisé à 0.

-**duréeService** correspond à la durée en minutes que le client doit encore passer dans une station avant que sa requête ne soit considérée comme traitée et qu'il puisse quitter le système. Cette durée est générée aléatoirement pour chaque client en respectant la loi de services fournie dans l'énoncé.

-**typePriorité** correspond au niveau de priorité du client, et peut prendre les valeurs (dans le programme : énumération) suivantes : ABSOLU, RELATIF, ORDINAIRE

### La structure Station

Chaque station est représentée par une structure qui aura pour attributs :

-**clientServi** l'actuel client servi par la station. Initialisé à NULL lors de la création de la station.

-**coutFonctionnement** : initialisé à 0 lors de la création de la station, ce cout représente le cout que cette station représente dans le système pour le moment. Il sera ainsi incrémenté au fur-et-à-mesure que le temps passera.

### tabStations

Il s'agira d'un tableau de stations. Il sera trié par ordre de priorité du client servi (dans l'ordre suivant : ABSOLU, RELATIF, ORDINAIRE). Les cellules ayant pour valeur NIL seront placées en fin de tableau.

### Files d'attente

Il s'agira de tableaux de clients. Il y en aura trois : **fileAbsolu**, **fileRelatif**, **fileOrdinaire**.

Ces tableaux seront triés sur la durée de service restante des clients qui les composent.

## Conversion en orienté objet

L'architecture de mon programme est la suivante : Les clients (**Customer**) et les stations (**Station**) sont modélisés en tant qu'objets, et au lieu d'en faire des tableaux dans la classe principale, j'ai créé des objets représentant l'ensemble des stations actives (**Stations**) et des clients. Pour modéliser les files, j'ai créé une classe générique qui correspond aux files relatives et ordinaires (**Queue**), et une classe qui hérite de **Queue**, **AbsoluteQueue**, représentant la file des prioritaires absolus. J'ai fait ce choix car les files relatives et ordinaires ont un comportement similaire (le gros changement est que la file des prioritaires relatifs est défilée en premier), mais la file de prioritaires absolus, même si elle partage des points communs avec les deux autres files, présente beaucoup de singularités par rapport à celles-ci.

Enfin, la génération aléatoire étant au cœur du programme, j'ai créé une classe de génération aléatoire (**RandomGenerator**). Cette classe est instanciée une fois au début de la méthode principale et se charge de faire les différentes générations aléatoires utiles au programme.

Voici les détails des attributs de ces classes :

### **Customer :**

**systTime** : Le temps que le client a déjà passé dans le système

**serviceDuration** : La durée de service restante du client

**priorityType** : Le type de priorité du client, représenté par une énumération.

### **Station :**

**currentCustomer** : Le client actuellement en train d'être servi par la station

**stationCost** : Le coût total que représente l'occupation/l'inoccupation de la station

### **Stations :**

**stationsArray** : Le tableau (ArrayList) représentant l'ensemble des stations

**numberOfCurrentServices** : Il s'agit du nombre de stations occupées par un client

### **Queue :**

**queue** : représente une file de clients.

### **AbsoluteQueue (extends Queue) :**

**numberOfCustomers** : Le nombre de clients composant la file de prioritaires absolus.

### **RandomGenerator :**

Cette classe a pour attributs les différents éléments de la formule de génération aléatoire.