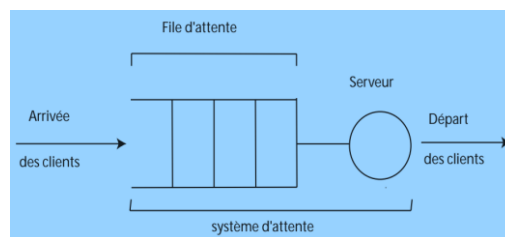


## Simulation d'une file d'attente

### 1. Présentation du système

Un système de file d'attente comprend : (Figure1)

- Un espace de service avec une ou plusieurs stations de service,
- Un espace d'attente dans lequel se forme une éventuelle queue d'attente.



**Figure 1.** Un système de File d'attente

- On parle de phénomène d'attente chaque fois que certaines unités appelées “clients” se présentent d'une manière aléatoire à des “stations” afin de recevoir un service dont la durée est généralement aléatoire.
- Si un poste de service est libre, le client qui arrive se dirige immédiatement vers ce poste où il est servi, sinon, il prend sa place dans une queue d'attente dans laquelle les clients se rangent suivant leur ordre d'arrivée.

### 2. Enoncé

La file d'attente à simuler dans ce TP est une file **FIFO** d'un **seul serveur** et d'une **capacité illimitée** (le nombre de places dans la queue est illimité), avec un flux d'arrivée de clients suivant un processus aléatoire du **taux d'arrivée  $\lambda$**  (autrement dit les intervalles d'inter-arrivées entre les clients sont aléatoires), et un temps de service aléatoire pour chaque client du **taux  $\mu$** . C'est ce qu'on appelle **M/M/1** dans le jargon de la théorie des files d'attente.

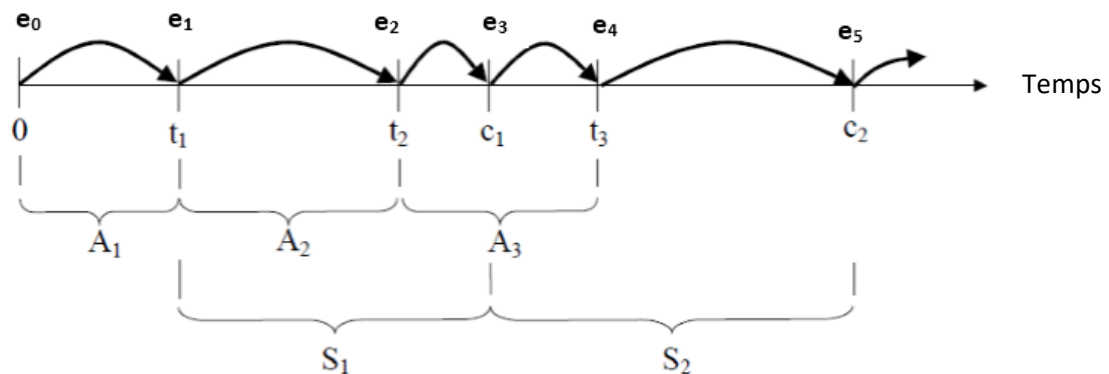
En suivant le document « **Simulation en Java de M/M/1** », simuler une file M/M/1 en JAVA :

- Présentant le fonctionnement d'un système de file d'attente avec **un seul serveur** pour **60 clients**

- Comme résultat : pour chaque client, afficher :
  - Son numéro ou son identifiant : 1, 2, 3, ...60
  - Son temps d'arrivée
  - Son service commence à quel instant
  - La durée de son service
  - Son service termine à quel instant
- Tels que :
  - $\lambda$  est le taux d'arrivée des clients par unité de temps
  - $\mu$  est le taux de départ des clients par unité de temps
  - Les valeurs de  $\lambda$  et  $\mu$  sont données au début du programme, par exemple :  
 $\lambda = 0,5$  clients/s et  $\mu = 1$  client/s
  - Le temps d'arrivée du premier client est « **0.00** »
  - Ensuite, pour chaque client on lance un temps d'inter arrivée =  
 $(-Math.log(Math.random()))/\lambda$
  - Pour chaque client : la durée de service =  $-Math.log(Math.random())/\mu$
  - Deux évènements possibles pour la classe « Event » sont : l'arrivée et le départ des clients.

### 3. Principe de simulation

La figure suivante (Figure 2) donne une illustration de ce principe de simulation.



**Figure 2.** Une trajectoire des événements

Soit :

- $t_i$  = l'instant d'arrivée du  $i^{\text{ème}}$  client ( $t_0 = 0$ )
- $A_i = t_i - t_{i-1}$  = temps d'inter-arrivées entre  $(i-1)^{\text{ème}}$  et  $i^{\text{ème}}$  arrivée de clients
- $S_i$  = temps de service du  $i^{\text{ème}}$  client (temps d'attente exclus)
- $D_i$  = temps d'attente dans la queue du  $i^{\text{ème}}$  client
- $c_i = t_i + D_i + S_i$  = l'instant où le  $i^{\text{ème}}$  client finit son service et quitte le système
- $e_i$  = l'instant d'occurrence du  $i^{\text{ème}}$  événement du type quelconque