**Programme de gestion de files d’attente**

L’énoncé donné est disponible dans le dossier. Je vais ici présenter les différents objets et structures utilisés pour ce programme (le langage employé pour le codage est le Java), ainsi que le diagramme d’action.

**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**

L’objet Client

Dans ce programme, les clients seront des objets 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.

-**durééService** 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

L’objet Station

Chaque station sera représentée par un objet 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) et, pour une même priorité, par temps de service restant du client servi (en ordre croissant).

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.