# Utilisation d’un fichier pour ne devoir retaper toute la suite de commandes

La brocante offrant beaucoup de fonctionnalités, il va falloir entrer beaucoup de commandes lors de l’exécution de GestionBrocante pour les tester toutes !

Or la moindre faute de frappe peut faire planter l’exécution et nous obliger à retaper toute la séquence de commandes ! Ceci est très embêtant et peut faire perdre beaucoup de temps !

Pour éviter cela, le plus simple est d’écrire (via Bloc-notes) la séquence de commandes dans un fichier que je vais appeler **lancement.txt**

Dans le tableau ci-dessous, dans la colonne de gauche, on voit l’exécution de GestionBrocante dans lequel on interagit en entrant les commandes (en vert) les unes après les autres.

Dans la colonne de droite se trouve le contenu du fichier **lancement.txt** dans lequel j’ai retranscrit les commandes les unes en-dessous des autres.

|  |  |
| --- | --- |
| **Données entrées au clavier** | **Fichier lancement.txt** |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Gestion d'une brocante  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Configuration de la brocante  ----------------------------  Entrez le nombre d'emplacements : 8  Entrez le nombre de riverains : 3  Entrez le nom du riverain 1: riv1  Entrez le nom du riverain 2: riv2  Entrez le nom du riverain 3: riv3  Phase 1  -------  1 -> reserver un emplacement  2 -> afficher la brocante  Votre choix : 1  Entrez le nom : riv1  Entrez le numero de l'emplacement : 0  1 -> reserver un emplacement  2 -> afficher la brocante  Votre choix : 2  tableEmplacements[riv1, null, null, null, null, null, null, null]  mapRiverains{riv3=0, riv2=0, riv1=1}  pileEmplacementsLibres[] | 8  3  riv1  riv2  riv3  1  riv1  0  2 |

Normalement, dans la classe GestionBrocante, on utilise le scanner sur l’entrée standard grâce à l’instruction *scanner* = **new** Scanner(System.***in***). De cette manière, on lit les données entrées au clavier.

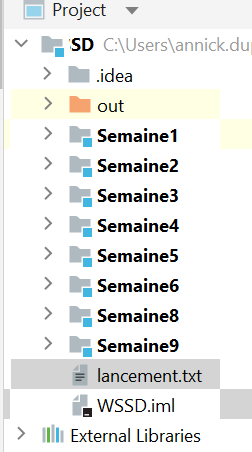
Afin de lire les données dans le fichier **lancement.txt** plutôt qu’au clavier, on va d’une part utiliser un objet de la classe **MonScanner** qui vous est fournie et d’autre part indiquer le fichier dans lequel se trouve les commandes.

*scanner* = **new** MonScanner("lancement.txt") ;

Mettez le fichier à la racine du projet d’IntelliJ.

Le fichier **lancement.txt** apparait dans IntelliJ et on peut le modifier via IntelliJ.

Si le fichier est bien placé, voilà où il devrait se trouver :



Quand le scanner aura lu toutes les commandes du fichier, il se remettra automatiquement sur l’entrée standard et on pourra ainsi de nouveau entrer des commandes au clavier. C’est la classe **MonScanner** qui permet cela. On vous fournit cette classe car la classe Scanner de Java ne permet pas de revenir sur l’entrée standard après avoir lu un fichier ! De plus, le Scanner de Java n’affiche pas à l’écran les commandes présentes dans un fichier mais un objet de type **MonScanner** le fait !

On retrouve la même exécution que si on avait entré toutes les commandes à la main mais en beaucoup plus rapide !

Les fonctionnalités offertes par un objet de type **MonScanner** sont les mêmes que celles d’un objet de type Scanner, à savoir :

* next()
* nextLine()
* nextInt()
* nextDouble()