



Extrait du Environnement iSeries

<http://xdocs400.com/spip.php?article272>

# Petit exercice algorithmique exploitant les listes mémoires

- Les articles -



Date de mise en ligne : vendredi 30 décembre 2005

## **Description :**

Cet article résulte du développement d'une édition un peu particulière, produite récemment pour un client, et qui a donné lieu à la mise au point d'un algorithme s'appuyant sur le concept de listes mémoires.

---

**Environnement iSeries**

---

**Cet article résulte du développement d'une édition un peu particulière, produite récemment pour un client, et qui a donné lieu à la mise au point d'un algorithme s'appuyant sur le concept de listes mémoires, dont j'ai parlé dans un précédent article.**

Je vous ai parlé des listes mémoires dans un précédent article ([Du bon usage des listes mémoires](#)). Dans Ce nouvel article, je vais vous présenter un autre exemple d'utilisation de ces listes mémoires. A noter que la méthode proposée ici pourrait tout aussi bien s'appliquer à un fichier de travail temporaire, mais il me semblait tout à fait opportun d'exploiter le principe des listes mémoires dans ce cas.

Le cas en question consistait à éditer un bon de préparation (BP) sur 2 colonnes par page, selon les exigences de notre client, qui souhaitait retrouver ainsi un type de présentation qu'il avait apprécié dans un autre logiciel. On a dû se gratter le crâne, accepter le fait que l'on devrait sortir de nos éditions standards exploitant les sauts de page automatiques (tels que le gère Adelia), mais ça pouvait donner lieu à un algo sympa à mettre au point, alors on a retroussé nos manches.

Pour bien comprendre ce que l'on devait produire, je vous propose une illustration au travers d'un petit tableau. Dans notre exemple fictif ci-dessous, on supposera que nos pages ne contiennent que 5 lignes ce qui ne correspond pas à la réalité) :

### Page 1

Code art.	Libellé article	Code art.	Libellé article
1	article 1	6	article 6
2	article 2	7	article 7
3	article 3	8	article 8
4	article 4	9	article 9
5	article 5	10	article 10

### Page 2

Code art.	Libellé article	Code art.	Libellé article
11	article 11	16	article 16
12	article 12	17	article 17
13	article 13	18	article 18
14	article 14	19	article 19
15	article 15	20	article 20

## Petit exercice algorithmique exploitant les listes mémoires

---

Bien sûr, j'ai simplifié au maximum l'état que je souhaite présenter dans cet article, car en réalité l'édition de BP est composée d'autres colonnes ("quantité", etc...), qui sont de peu d'intérêt pour ce que je veux vous expliquer ici.

Attention : ça a l'air simple en apparence, mais si vous avez déjà eu à traiter ce type de problème, vous savez comme moi qu'il y a quelques pièges à éviter, pour répondre correctement à la problématique posée.

Si vous n'avez jamais eu à traiter ce type de cas, je vous recommande d'essayer d'écrire l'algorithme, avant de lire la suite de cet article.

### Passons à la résolution du problème.

Puisque nous voulons traiter ce problème au moyen d'une liste mémoire, nous avons besoin d'une liste mémoire composée de plusieurs éléments, tels que code article, libellé article, etc..., et surtout de variables servant à conserver le numéro de page, le numéro de ligne et le numéro de colonne.

On va donc procéder en plusieurs étapes :

1. création de la liste mémoire
2. chargement de la liste mémoire avec les articles de la commande à éditer
3. tri de la liste mémoire selon les critères exigés par le client (par exemple, tri des articles par ordre alphabétique)
4. relecture de la liste mémoire selon le tri indiqué au point 3, et affectation à chaque élément de la liste mémoire d'un numéro de page, d'un numéro de ligne et d'un numéro de colonne
5. tri de la liste mémoire sur les critères suivants : numéro de page, numéro de ligne, numéro de colonne
6. relecture de la liste mémoire selon le tri indiqué au point 5, et édition de l'état en respectant les ruptures sur numéro de page et numéro de ligne, et en s'appuyant sur le numéro de colonne pour dispatcher les données au bon endroit dans la ligne.

Les différentes étapes ci-dessus sont décrites avec force commentaires dans le source Adelia que je vous propose dans un fichier texte zippé joint à cet article (algo\_BP2col.zip). Ce source est bien sûr donné ici à titre d'exemple, et il peut sans aucun doute être amélioré. Si vous deviez développer un état similaire à celui présenté ici, je pense que vous pourriez adapter assez facilement l'algorithme présenté dans ce source à vos besoins.

Bonne lecture.

*Post-scriptum :*

*Comme je l'ai dit précédemment, on aurait pu écrire ce programme en utilisant un fichier de travail temporaire plutôt qu'une liste mémoire. La méthode présentée ici n'a pas la prétention d'être la meilleure, on aurait sans doute pu trouver d'autres manières encore d'éditer cet état. D'ailleurs, si vous avez des idées d'algorithmes à nous proposer, n'hésitez pas à prendre contact avec David Malle, notre cher webmaster.*