

Feuille 1 - Programmation Impérative

1 Fizz-Buzz

- Écrire un programme qui affiche les entiers jusqu'à 100 en remplaçant les multiples de 5 par fizz, les multiples de 7 par buzz et les multiples des deux par fizzbuzz. Utiliser le modulo (%).
- Modifier le programme ci-dessus pour qu'il demande à l'utilisateur un entier positif n à la place de 100, et exécute le même code jusqu'à n .
- Modifier le programme ci-dessus pour qu'il prenne l'entier n comme paramètre sur la ligne de commande.

2 La Récursivité

- Écrire un programme qui, étant donné un entier n sur la ligne de commande, calcule $n!$, la factorielle de n . Donnez deux versions de votre programme : une avec une boucle, et une avec une fonction `static` récursive.
- Écrire un programme qui, étant donné un entier n sur la ligne de commande, calcule le n -ième terme de la suite de Fibonacci. On rappelle que la suite de Fibonacci est 1, 1, 2, 3, 5, 8, 13, ..., où chaque terme est la somme des deux termes qui les précèdent. Donnez deux versions de votre programme : une avec une boucle, et une avec une fonction `static` récursive.
- Quelle version est préférable pour ces deux problèmes, la solution récursive ou la solution itérative? Justifier votre réponse en termes de temps d'exécution et en termes d'utilisation de mémoire.

3 Les Matrices

Écrire un programme qui produit deux matrices (tableaux à 2 dimensions) des entiers et calcule leur produit. Plus précisément :

1. Le programme est donné trois entiers a, b, c sur la ligne de commande.
2. Le programme construit une matrice A de dimensions $a \times b$, et une matrice B de dimensions $b \times c$.
3. Le programme remplit les deux matrices des valeurs aléatoires. Utiliser ici la classe `Random`, de `java.util.Random`.
4. Le programme calcule le produit $A \times B$, qui est une matrice de dimensions $a \times c$.
5. Le programme affiche les trois matrices.