

TD Concepts objet et Java

2016-17

Exercice 2

Vous allez développer une classe utilitaire en Java permettant de réaliser divers traitements sur des tableaux d'entiers.

1. Créez une classe au nom approprié dans un package pertinent.
2. Ajoutez une méthode statique à votre classe permettant de créer un tableau d'entiers (`int[]`) à partir d'un tableau de chaînes de caractères (`String[]`). Ajoutez également une méthode `main` à cette classe pour tester la méthode précédente en lui transmettant le tableau de paramètres lu depuis la ligne de commandes.
3. Comment la méthode de conversion doit-elle se comporter en cas de problème de conversion, i.e. si une des chaînes du tableau ne peut pas être convertie en entier à l'exécution ?
4. Ajoutez une méthode de classe affichant le contenu d'un tableau d'entiers passé en paramètres.
5. Que se passe-t-il au niveau de la machine virtuelle en cas de dépassement de lecture des éléments d'un tableau, par exemple dans la méthode précédente ? Pourquoi ceci n'a-t-il pas mené à une erreur de compilation ?
6. Ajoutez une méthode à la classe retournant la valeur maximale des valeurs d'un tableau d'entiers passé en paramètre.
7. On souhaite ajouter une méthode de classe qui permettrait d'échanger la valeur minimale d'un premier tableau d'entiers transmis en paramètre avec la valeur maximale d'un second tableau d'entiers. Le passage de paramètres en Java s'effectuant *par copie*, cela est-il possible ?
8. On souhaite maintenant ajouter une méthode de classe qui permettrait d'accepter un nombre quelconque d'entiers directement en paramètres de la méthode. Cela est-il possible en Java ? Si oui, le mettre en œuvre avec une méthode qui retourne la somme des entiers transmis.
9. Créez un fichier `JAR` ne contenant que le code compilé de votre classe, et utilisez-le dans la méthode `main` d'une nouvelle classe d'un package `xxx.utilisateur`. Générez également la Javadoc pour le package dans lequel se situe votre classe. Échangez l'ensemble avec un camarade et mettez-vous en situation de réutiliser son code (attention, aucun nom de méthode n'ayant été imposé, il faudra donc vous référer à sa documentation, éventuellement depuis l'IDE).