Partie PC (15 points)

On vous demande de compléter la classe *TableEntiersNonNuls*.

Cette classe possède comme attributs :

private int[] table;  
private int nbEntiers;  
private static final int *MAX* = 10; *// taille physique*

table ne peut contenir plus de *MAX* entiers non nuls , mais pourrait en contenir moins.  
Les entiers occupent les nbEntiers premières cases du tableau.  
L'ordre des entiers n'a pas d'importance.

Voici un exemple :

table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | -6 | 1 | 9 | 3 | -2 |  |  |  |  |

partie de la table inutilisée

nbEntiers = 6

Une des méthodes demandées doit construire une table contenant les trois plus petits entiers.

Vous allez écrire cette méthode en vous basant sur le principe du tri par sélection.

*« On* ***sélectionne*** *chaque fois* ***le plus petit entier*** *parmi les entiers qui n’ont pas encore été sélectionnés. »*

Attention, pour ne pas sélectionner chaque fois le même entier, il faut le remplacer par un très grand entier (Integer.MAX\_VALUE). Pour ne pas modifier la table courante, il est nécessaire de travailler avec une copie.

Voilà ce qui va se passer avec la table donnée en exemple :

Au départ :

table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | -6 | 1 | 9 | 3 | -2 |  |  |  |  |

tableTemp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | -6 | 1 | 9 | 3 | -2 |

table3Min

|  |  |  |
| --- | --- | --- |
|  |  |  |

Après 1 étape :

table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | -6 | 1 | 9 | 3 | -2 |  |  |  |  |

tableTemp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | MAX\_VALUE | 1 | 9 | 3 | -2 |

table3Min

|  |  |  |
| --- | --- | --- |
| -6 |  |  |

Après 2 étapes :

table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | -6 | 1 | 9 | 3 | -2 |  |  |  |  |

tableTemp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | MAX\_VALUE | 1 | 9 | 3 | MAX\_VALUE |

table3Min

|  |  |  |
| --- | --- | --- |
| -6 | -2 |  |

Après 3 étapes :

table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | -6 | 1 | 9 | 3 | -2 |  |  |  |  |

tableTemp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | MAX\_VALUE | MAX\_VALUE | 9 | 3 | MAX\_VALUE |

table3Min

|  |  |  |
| --- | --- | --- |
| -5 | -2 | 1 |

Vous allez compléter un constructeur et 3 méthodes de la classe *TableEntiersNonNuls*. La classe *TestTableEntiersNonNuls* permet de tester les différentes méthodes à compléter.

Suivez bien la *JavaDoc* et les contraintes de programmation imposées.

Il est interdit de passer par des classes *Arrays*, *System* ou autres !

Vous ne pouvez pas ajouter d’autres méthodes.

Répartition des points :

TableEntiersNonNuls() : 4 pts

moyennePositifs() : 3 pts

troisPlusPetitsEntiers() : 4 pts

supprimerDerniereOccurrence() : 4 pts

Conseil :

Ne vous acharnez pas sur une méthode si tous les tests ne passent pas. Passez là. Vous y reviendrez plus tard ! Il n’est pas indispensable que le constructeur soit correct pour tester les autres méthodes.