

Exercice 1 : Structure irrégulière

On souhaite pouvoir disposer d'une structure basée sur un tableau à deux dimensions, mais dont toutes les lignes ne contiennent pas le même nombre d'éléments. Dans l'exemple suivant, les lignes ont successivement des tailles qui varient : la ligne d'indice 0 contient 3 éléments, la ligne d'indice 1 ne porte qu'un seul élément. . .

3	9	4
6		
2	9	
7	2	1

Ce type de structure est définie sur-mesure et ne doit donc avoir aucune case vide. L'invocation de la méthode `length` donne un résultat différent selon la ligne sur laquelle est invoquée la méthode.

Q1. Écrivez une classe `Irregular` permettant de créer une telle structure, ainsi que son constructeur répondant à la signature suivante. Le constructeur doit donc réserver l'espace mémoire ligne par ligne.

```
Irregular(int[] lineSize)
```

La taille de chacune des lignes est spécifiée dans le paramètre `lineSize`. Un paramètre valant `[1, 3, 2]` mènera à une structure dont la première ligne n'a qu'un élément, la seconde ligne 3 éléments alors qu'il y aura 2 éléments sur la dernière ligne.

Q2. Écrivez une méthode de remplissage aléatoire au sein de cette classe, de signature :

```
void randomFilling()
```

Q3. Écrivez une méthode générant la représentation textuelle d'une telle structure, afin d'en préparer l'affichage, de signature :

```
String display()
```

Q4. Écrivez une classe `UseIrregular` permettant la vérification du code par l'exécution d'un scénario exemple.

Q5. Ajoutez une méthode `isCommon` permettant de déterminer si un élément donné en paramètre est présent dans chacune des lignes la structure :

```
boolean isCommun(int element)
```

Q6. Écrivez une méthode `existCommon` permettant de déterminer l'existence ou non d'un élément commun à toutes les lignes :

```
boolean existCommon()
```

Q7. Modifiez le scénario de la classe `UseIrregular` pour permettre la validation du code.