

TP03 – JAVA : LES COLLECTIONS

RÉVISIONS STRUCTURE RÉPÉTITIVES

1 - Choisir un nombre entier compris entre 1 et 100. Effectuer ensuite des tirages aléatoires (de 1 à 100) et compter le nombre de tirages nécessaire pour obtenir le nombre initial. Arrêter les tirages et afficher le nombre de coups

```
int aleatoire = (int) (Math.random() * 100) → détermine un nb aléatoire entre 0 et 100
```

1.

La limite supérieure n'est cependant pas comprise dans l'intervalle. Pour l'inclure, la solution la plus simple consiste à ajouter 1, puis à enlever la partie décimale en convertissant le nombre en entier.

La formule qui en résulte est la suivante :

```
int nombreAleatoire = Min + (int)(Math.random() * ((Max - Min) + 1));
```

Ce qui donne dans notre exercice :

```
int nombreAleatoire = 0 + (int) (Math.random() * ((100 - 0) + 1)
```

```
int nombreAleatoire = (int) (Math.random() * 101 ;
```

Exemple :

```
in chiffre = 40;
```

Nombre trouvé en 24 tirages aléatoires

2 - Effectuer une suite de tirages de nombres aléatoires entre (0, 1000) jusqu'à obtenir une suite composée d'un nombre pair suivi de deux nombres impairs.

Exemple de résultat obtenu :

194,285,494

435,759,162

237,292,768

366,533,397

Résultat obtenu en 4 coups

EXERCICE 1 : DECLARATION DE TABLEAUX

1. Initialiser un tableau (tabInt) de 10 entiers compris entre 0 et 100 choisis au hasard.

```
int aleatoire = (int) (Math.random() * (100 + 1)
```
2. Ecrire le programme qui affiche la somme, le plus petit et le plus grand des éléments de ce tableau. UTILISEZ UNE boucle pour l'initialisation, UNE AUTRE pour l'affichage, UNE TROISIÈME pour le calcul de la somme, du produit, du minimum et du maximum.
3. Ajoutez la déclaration de 2 tableaux supplémentaires tabPair et tabImpair. Ajoutez le code qui copie les éléments pairs de tab dans tabPair, et les éléments impairs dans tabImpair, puis qui affiche ces 2 tableaux.
4. Afficher le maximum et le minimum du tableau
Problème : dans les 10 entiers choisis au hasard, on ne sait pas combien seront pairs ou impairs donc on déclare 2 tableaux de nombres pairs et impairs d'une capacité de 10 (il existe d'autres solutions).

Avec la classe ArrayList, nous n'avons pas ce problème, il suffit de déclarer un ArrayList et ensuite d'y ajouter les éléments.

Refaire cet exercice en utilisant la classe ArrayList :

```
List<Integer> listEntier = new ArrayList<Integer>();  
List<Integer> listEntierPair = new ArrayList<Integer>();  
List<Integer> listEntierImpair = new ArrayList<Integer>();
```

EXERCICE 2 : ArrayList – HashMap

1 - Ecrire un programme qui permet d'initialiser les 2 ArrayList suivants :

pays ==> ("France","Allemagne","Russie","USA","Chine")

population ==>(62,82,288,200,1290)

2 - Ecrire la partie du programme permettant d'afficher le nom du pays ayant la population la plus faible (France : 62 millions)

3 - Ecrire la partie du programme permettant d'afficher le nom des pays ayant une population supérieure à 200 millions.

4 - Remplacer les 2 ArrayList par un HashMap

EXERCICE 3 : Fréquence mot

Afin d'essayer de déterminer automatiquement à quelle catégorie appartient une phrase, on souhaite déterminer le nombre de mots de 1 lettre, 2 lettres, 3 lettres... qu'il contient.

Vous devez donc demander à l'utilisateur de saisir une phrase, Pour chaque longueur de mot possible, et uniquement s'il y avait des mots de cette longueur dans le texte, vous devez afficher sur une ligne la longueur et le nombre de mots de cette longueur, séparés par un deux-points.

CONTRAINTES

Le texte contient un ensemble de mots, séparés par des espaces, sans aucun signe de ponctuation. La longueur maximale d'un mot est de 30 lettres.

Exemple :

phrase → Qui vole un oeuf vole un bœuf une abeille vaut mieux que mille mouches

Sortie :

```
2 : 2   ( 2 mots de 2 lettres)  
3 : 3  
4 : 4  
5 : 3  
7 : 2   (7 mots de 2 lettres)
```

EXERCICE 4 : jeu du pendu

Un mot peut contenir plusieurs fois la même lettre. Trouver une solution à l'aide d'un tableau pour obtenir le résultat suivant :

```
$mot="baobab";
```

```
$lettre='b';
```

- la lettre b se trouve en position 0, 3, 5 dans le mot baobab

Écrire un programme qui affiche une chaîne composée de n #, n correspond au nombre de lettres contenues dans le mot à deviner (Exemple `$mot="info"`).

L'utilisateur saisit une lettre. Si la lettre est présente dans le mot, le programme remplace le # concerné par la lettre. Sinon, le programme indique que la lettre n'est pas présente dans le mot.

On recommencera l'opération tant que le mot n'est pas trouvé.