

Développement orientée objet - Java

TP 05

Lionnel Conoir, Isabelle Delignières, Wajdi Elleuch,
Rémi Synave et Franck Vandewiele

Travail préliminaire

Dans votre répertoire `java`, créez un nouveau répertoire `Tp5`. Vous y placerez tous les fichiers utilisés pour la réalisation du TP. Vous avez 2 séances pour faire ce TP.

Les tableaux statiques et type primitif

Attention ! Nous n'utilisons pas d'`ArrayList` ici.

1. Dans une classe `Exercice1`, écrivez un programme dans lequel vous créez un tableau de 20 entiers, le remplissez aléatoirement avec des chiffres de 0 à 9 et l'affichez.
2. Dans une classe `Exercice2`, écrivez un programme dans lequel vous créez un tableau de 20 entiers, le remplissez aléatoirement avec des chiffres de 0 à 9, l'affichez, le trie avec l'algorithme de votre choix (voir tp de python du semestre 1) et l'affichez de nouveau.
3. Dans une classe `Exercice3`, écrivez un programme dans lequel vous créez un tableau de 20 entiers, le remplissez aléatoirement avec des chiffres de 0 à 9, l'affichez puis demandez à l'utilisateur quel chiffre doit être supprimé. Le programme doit supprimer **la première occurrence de ce chiffre**. Affichez finalement le **nouveau** tableau résultant.
4. Dans une classe `Exercice4`, écrivez un programme dans lequel vous créez un tableau de 20 entiers, le remplissez aléatoirement avec des chiffres de 0 à 9, l'affichez puis demandez à l'utilisateur quel chiffre doit être supprimé. Le programme doit supprimer **la dernière occurrence de ce chiffre**. Affichez finalement le **nouveau** tableau résultant.
5. Dans une classe `Exercice5`, écrivez un programme dans lequel vous créez un tableau de 20 entiers, le remplissez aléatoirement avec des chiffres de 0 à 9, l'affichez puis demandez à l'utilisateur quel chiffre doit être supprimé. Le programme doit supprimer **toutes les occurrences de ce chiffre**. Affichez finalement le **nouveau** tableau résultant.

6. **Astuce :** Vous pouvez déclarer et initialiser vos tableaux directement avec la syntaxe suivante :

```
int tab[] = {5, 4, 2, 8, 9};
```

Dans une classe **Exercice6**, écrivez un programme dans lequel vous créez un tableau de N entiers que vous remplirez avec l’astuce ci-dessus. Suite à cette déclaration et initialisation du tableau, écrivez un algorithme permettant de vérifier si le tableau est un palindromme. Vous testerez votre programme en modifiant les valeurs du tableau pour tester les différents cas.

Les tableaux statiques et type classe

Attention ! Nous n’utilisons toujours pas d’`ArrayList` ici.

En utilisant la classe **Voiture** fournie, reprenez les exercices 1 à 5 précédent (que vous nommerez en exercice 7, 8, 9, 10 et 11). Vous remplirez le tableau aléatoirement en tirant aléatoirement la marque entre Renault et Toyota, puis le kilométrage entre 0 et 9. L’utilisateur devra entrer la marque et le kilométrage. Le tri se fera sur le kilométrage seulement.

Les tableaux dynamiques

Attention ! Voilà. Ici vous utiliserez des `ArrayList`.

Dans une classe **Exercice12** :

- créez un tableau dynamique de 10 voitures Renault dont le kilométrage ira de 100 à 1000 kilomètres (par pas de 100 évidemment).
- Affichez les voitures dans le terminal.
- Grâce à la méthode **add**, ajoutez une voiture Renault ayant 250km de façon à ce que le tri des voitures par kilométrage soit respecté.
- Affichez de nouveau les voitures dans le terminal.
- En utilisant la méthode **remove** prenant un objet en paramètre, supprimez la voiture ayant 500 kilomètres au compteur. Pour ce faire, vous devez, bien entendu créer une nouvelle voiture Renault ayant 500 kilomètres.

Est ce que la suppression a été réalisée ?

Si oui, revérifiez ! Vraiment ? Bon ok, c’est parfait, vous avez fait le nécessaire.

Si non, que faut-il faire et... faites le !

Ça ne fonctionne toujours pas ? Si tu as bien écouté en cours, il a été dit que cette méthode n’avait la bonne signature. Cherchez sur Internet la bonne signature de la méthode et comment la développer correctement. Et... faites le !