

Compléments de Programmation

Licence 1 UPEC 2023/2024

TM2 : Classes et méthodes en Java

Nous allons continuer la prise en main de Java

Question 1 (Méthodes statiques et méthodes d'instance):

Créez un répertoire TM7. Vous y mettrez tous vos fichiers. Dans une classe `Tm7.java` définissez

- Deux champs entiers et un champ de type `String`.
- Une méthode statique qui prend en entrée un `String` et renvoie le même `String` concaténé avec lui-même. Si on appelle la méthode sur "to" elle renvoie "toto".
- Une méthode d'instance qui renvoie la somme de deux champs entiers si le `String` du champ est "somme" et qui renvoie 0 sinon.

Créez une classe `App` avec la méthode `main` pour tester.

Question 2 (Constructeurs):

- ajoutez à la classe `Tm7.java` un constructeur qui initialise les trois champs
- ajoutez un constructeur qui initialise les deux champs entiers, et qui initialise le champ `String` à "somme".
- ajoutez un constructeur qui initialise les deux champs entiers à une même valeur, et qui initialise le champ `String` à "toto".

Question 3 (ToString):

Ajoutez à la classe `Tm7.java` la méthode `toString` qui affiche joliment les valeurs des champs. Testez dans le `main`.

Question 4 (Utilisation de la librairie standard):

Considérez la classe `StringBuffer` de la librairie standard de Java : <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/StringBuilder.html>. Cette classe permet des opérations très rapides sur les chaînes de caractères. Vous pouvez créer un `StringBuffer` à partir d'un `String s` en faisant `new StringBuffer(s)`. À partir d'un `StringBuffer sb` vous pouvez récupérer un `String` en faisant `sb.toString()`

Étudiez les méthodes :

- `StringBuilder append(String str)`
- `StringBuilder delete(int start, int end)`
- `StringBuilder insert(int offset, String str)`

Pouvez-vous comprendre ce qu'ils font ?

Dans une méthode `main` d'une classe `AppSB.java` cherchez d'utiliser ces méthodes pour manipuler des chaînes de caractères. En particulier concaténez une chaîne de caractère 10000 fois avec elle-même, une première fois en utilisant l'opération "+" entre `String` et une deuxième fois en utilisant la méthode `StringBuilder append(String str)`. Comparez le temps d'exécution.

Question 5 (Un tableau extensible):

Dans cet exercice on va créer une classe `MyTableau.java`. Elle contient

- un champ de type `int[]`, et un champ de type `int` qui dit combien de cases du tableau sont utilisées.
- Un constructeur qui initialise le tableau à une taille donnée, et le champ entier à 0.
- Une méthode d'instance `void add(int val)` qui ajoute la valeur à la fin du tableau (et du coup incrémente aussi le champ entier). Si le tableau est plein, il faudra en créer un plus grand, et recopier les valeurs.
- Une méthode d'instance `void delete (int pos)` qui "efface" la valeur à la position indiquée (et qui déplace toutes les valeurs suivantes vers "la gauche"). Attention au champ entier.
- Une méthode d'instance `int get (int pos)` qui renvoie la valeur qui se trouve à la position indiquée.
- Une méthode d'instance `boolean estTrie()` qui dit si le tableau est trié en ordre croissant

Testez tout ça avec une méthode `main` quelque part.