

TP 1 2ème partie Durée : 2h

Les exercices 6 et 7 sont facultatifs. Ils utilisent des ArrayList qui seront vus au TP2.

Exercice 5 (gestion des dates)

But de cet exercice: essayer diverses manières d'obtenir et afficher la date courante (encore un prétexte pour vous faire chercher dans la documentation...).

A.Première manière : la méthode `java.lang.System.currentTimeMillis()` donne la date courante, exprimée comme le nombre de millisecondes qui se sont écoulées depuis le 1^{er} janvier 1970 à 0 heures GMT. C'est précis, mais pas très pratique pour organiser sa semaine! (Retenez quand même l'existence de cette méthode, car elle est bien utile pour mesurer et comparer les performances des programmes).

Écrivez un programme qui affiche le nombre de secondes écoulées depuis le 1^{er} janvier 1970. Exécutez deux fois ce programme, à une minute d'intervalle, et voyez si les valeurs obtenues correspondent à peu près à l'explication.

B.Deuxième manière : ajoutez au programme précédent des instructions qui créent un objet de type `java.util.Date` (par une expression comme «`Date d = new Date();`») et le donnent à afficher à `System.out`. Comme vous le voyez, c'est mieux, car on obtient bien une date, mais écrite à la manière des anglo-saxons. Ne cherchez pas la correction de ce défaut parmi les nombreuses méthodes de la classe `Date`, elles sont presque toutes deprecated (obsolètes, désapprouvées) et ne méritent pas qu'on s'y investisse.

C.Troisième manière : créer un objet de type `java.util.Calendar` (par une expression comme «`Calendar c = Calendar.getInstance();`») et obtenir séparément les éléments de la date (le jour de la semaine, le jour du mois, le mois, l'année) pour les afficher comme bon vous semble. En étudiant la documentation de la classe `Calendar` vous découvrirez qu'on obtient les divers composants par des expressions de la forme «`c.get(Calendar.MONTH);`», «`c.get(Calendar.YEAR);`», etc. Ecrivez un programme pour afficher la date d'aujourd'hui sous la forme : le jour de la semaine (son numéro), le jour dans le mois, le numéro du mois, l'année. Cette présentation n'est pas satisfaisante. On veut afficher la date du jour sous la forme : jour de la semaine en toutes lettres, puis jour du mois en chiffres, puis mois en lettres, puis année en chiffres (ex : mardi 23 janvier 2018).

Des tableaux de chaînes déclarés comme ceci peuvent vous aider à obtenir une présentation adéquate:

```
String[] mois = { "janvier", "février", ... "décembre" };
```

```
String[] jSem = { "lundi", "mardi", ... "dimanche" };
```

Au lieu de ces tableaux, vous pouvez utiliser des `ArrayList<String>`. Voir la classe `java.util.ArrayList` dans la documentation.

```
Ex : ArrayList<String> listeJourSemaine = new ArrayList<String> ();listeJourSemaine.add("lundi");
```

D.Quatrième manière (la meilleure): construire un objet `d` de type `java.util.Date` comme dans la deuxième manière et un objet `f` de type `java.text.SimpleDateFormat` (qui est une variété de `java.text.DateFormat`) et afficher le résultat du formatage du premier par le second, par une expression comme `f.format(d)`.

Ne vous laissez pas effrayer par la documentation de `SimpleDateFormat`. Si vous ne voyez pas comment cela marche, essayez ceci et vous comprendrez:

```
...
Date d = new Date();
SimpleDateFormat f = new SimpleDateFormat("dd MMMMM yyyy HH:mm");
System.out.println("maintenant: " + f.format(d));
....
```

Ecrivez un programme pour afficher la date d'aujourd'hui (et éventuellement l'heure) sous la forme :

- jj/mois en lettres/année hh:mm

- jj/mm/aa à hh:mm:ss

- jour de la semaine en toutes lettres, puis jour du mois en chiffres, puis mois en lettres, puis année en chiffres, puis hh:mm:ss (ex : mardi 23 janvier 2018 15:47:29).

Exercice 6 (ArrayList)

1) On veut enregistrer dans un objet de la classe `ArrayList` les notes obtenues par un élève à une matière. Le programme devra comporter les opérations suivantes, chacune réalisée par une méthode prenant en paramètre un objet de la classe `ArrayList` contenant les notes : ajouter une nouvelle note, afficher toutes les notes, calculer la moyenne. Le programme principal devra réaliser un test des méthodes avec les notes 12, 14, 9, un calcul de la moyenne, puis ajout de la note 13 et nouveau calcul de la moyenne et affichage des notes.

2) Une nouvelle méthode permettra de changer une note du tableau en indiquant son rang dans

l'ArrayList. Cette méthode peut être utile en cas d'erreur dans la saisie ou dans la correction de la copie.

3) Modifiez le programme principal pour que les différentes opérations soient proposées au moyen d'un menu.

Remarque : on ne gère ici que les notes d'un seul élève.

Exercice 7 (ArrayList)

Ecrivez un petit programme qui simule une caisse enregistreuse. On peut éditer le ticket de caisse d'un client en y ajoutant des produits ou en les supprimant.

Le menu proposera à l'utilisateur de

- saisir un produit (son libellé et son prix)
- afficher le contenu de la liste sous la forme :numéro d'ordre du produit, son libellé, son prix (ces 3 informations sur une seule ligne par produit
- supprimer un produit
- d'afficher le total des produits achetés avec la somme des prix des produits
- de quitter le programme.

Si l'utilisateur choisit de supprimer un produit de la liste, le programme :

- affichera le contenu
- demandera à l'utilisateur de choisir le numéro du produit à enlever
- supprimera le produit choisi
- affichera le contenu de la liste après suppression du produit.

Vous utiliserez la classe ArrayList et ses méthodes. Au vu de l'état de vos connaissances actuelles, vous créerez 2 ArrayList différentes : l'une pour les libellés et l'autre pour les prix, l'indice d'un produit dans les ArrayList étant le même. Cette façon de faire ne devra plus être utilisée dès le TP2.