TP Java n^{o}1

Dans ce TP nous allons écrire des applications Java. Les entrées/sorties mettront en œuvre la classe JOptionPane du paquetage swing. On peut lire une chaîne à l'aide de la méthode JOptionPane.showInputDialog. Ensuite, si nécessaire, on peut convertir une chaîne de caractères en valeur numérique ou booléen à l'aide des méthodes :

public static int Integer.parseInt(String s)
public static double Double.parseDouble(String s)
public static boolean Boolean.parseBoolean(String s)
public static short Short.parseShort(String s)

Note : A la fin de ce TP, vous devez penser à rendre vos sources à l'enseignant.

1 Exercices sur les types élémentaires

1. **Base deux**
   Écrire une application qui demande un entier et l'écrit en base deux. Utilisez de préférence les opérations logiques sur les bits (et, ou, décalages...).
   – A l'aide de Netbeans créez un projet de type "application Java", et écrivez le code correspondant.
   – Utilisez le débuggeur pour suivre pas à pas le déroulement de plusieurs exécutions de votre programme.
   – Une fois que votre application fonctionne correctement, fermez Netbeans, et exécutez votre programme sans son aide.
2. **Formule de Zeller**
   La formule de Zeller donne le jour de la semaine correspondant à une date (dans le calendrier grégorien) :

   ```
   w = t + [2,6 m - 0,2] + j + [j/4] + [h/4] - 2h
   ```

   h désigne la centaine d'années ou bien le siècle diminué d'une unité (nous sommes au 21ème siècle).

   j désigne l'année dans le siècle considéré (nous sommes en 09).

   m est le mois selon la numérotation romaine : mars correspond à 1, avril à 2 et janvier et février à 11 et 12 (mais de l'année précédente).

   t est le jour du mois.

   w modulo 7 donne le jour de la semaine avec la convention que dimanche correspond à 0.

La fonction désignée par [ ] dans la formule est la fonction partie entière qui agit sur un nombre réel.

Pour vérification le 1 er janvier 1901 était un mardi.

Écrire une application qui demande une date à l'utilisateur et qui écrit (en claire) le jour de la semaine qui lui correspond.

2 Tableaux de lettres aléatoires

1. Écrire une application qui demande un nombre entier représentant une taille, génère (à l'aide de Math.random()) un tableau des lettres minuscules aléatoires de la taille entrée et l'affiche. L'affichage se fera à l'aide d'une méthode (fonction) statique a part.
2. Écrire une méthode (fonction) de tri du tableau par l'algorithme de tri à bulles. L'utiliser pour trier le tableau, puis afficher le tableau trié. *Rappels :*L'algorithme du tri à bulles consiste à effectuer des parcours successifs d'un tableau pendant lesquels les paires de valeurs adjacentes sont comparées et échangées si elles ne sont pas dans le bon ordre. Ainsi, les entrées du tableau remontent comme des bulles jusqu'à ce qu'elles rencontrent une valeur avec laquelle elles sont bien rangées. Les parcours continuent jusqu'à ce qu'il n'y ai pas de valeurs échangés, et donc, que le tableau soit trié.
3. Écrire une méthode qui fusionne deux tableaux que l'on suppose triés dans un seul, en conservant l'ordre de manière à produire un tableau trié. On évitera d'appeler une fonction de tri dans cette méthode. Compléter l'application précédente pour générer deux tableaux, les trier à part, puis les fusionner dans un seul avant de l'afficher.
4. Écrire une méthode qu'enlève les éléments dupliqués d'un tableau trié, en gardant qu'un seul exemplaire. Appliquer cette fonction au tableau du point précédent.
5. Tri Boustrophédon. Au lieu de faire les parcours toujours dans le même sens, on alterne : on balaye de la première à la dernière case puis on balaye de la dernière case à la première. Optimisez pour ne pas repasser par les éléments déjà triés. Faire une méthode qui trie un tableau par cette algorithme, puis trouvez dans la documentation de la classe System une méthode pour mesurer le temps en nanosecondes. Comparez le temps de tri pour un même tableau pour les deux algorithmes.