

TP n°9

Chaînes de caractères

1 Le jeu du pendu

On souhaite écrire le jeu du pendu. Pour rappel, le but du jeu est de trouver un mot caché dont on connaît la taille, en proposant un caractère à chaque essai. Si le caractère proposé existe, on affiche où il se trouve dans le mot (pour toutes ses occurrences) et l'essai n'est pas compté. Si le caractère est incorrect, cela compte pour un essai. L'objectif est de découvrir le mot en un nombre d'essais maximum (5 essais).

Pour écrire ce jeu, on veut utiliser les tableaux de caractères. L'idée est de placer la solution (i.e. le mot à trouver) et l'état du jeu dans deux tableaux de caractères. Le tableau de l'état du jeu est de même longueur que la solution mais rempli par le caractère '-'. Lorsqu'un caractère est trouvé, l'état du jeu est mis à jour : on place le caractère trouvé à la place du '-' pour chacune de ses occurrences.

Dans la classe Scanner, il y a une méthode permettant de lire les chaînes de caractères : `nextLine`. Pour rappel, voici quelques instructions illustrant le fonctionnement des `String` :

```
String s; // Déclaration
s = "Bonjour"; // Initialisation
int taille = s.length(); // Récupération de la taille
char tmp = s.charAt(i); // Récupère le caractère i (i compris dans [0; s.length() - 1])
s = clavier.nextLine(); // Lecture depuis le clavier
```

Questions

- Écrivez une fonction/procédure `estValide` prenant en paramètre un mot sous forme d'un `String` et retournant `true` si le mot est valide. Un mot est valide si :
 - Il possède au moins 4 caractères
 - Toutes ses lettres sont en majuscule
 - Il ne possède pas de lettres accentuées
- Écrivez la fonction/procédure `convertir` qui prend en paramètre une chaîne de caractères (`String`) et qui la convertit en un tableau de caractères. Chaque case du tableau contient un caractère de la chaîne.
- Écrivez la fonction/procédure `initialiserSolution` qui demande à l'utilisateur de saisir le mot à deviner et qui retourne un tableau de caractères contenant ce mot.



Vous utiliserez les deux fonction/procédure précédentes (`estValide` et `convertir`).

4. Écrivez la fonction/procédure `creerMasque` prenant en paramètre un nombre n et retournant un tableau de n cases contenant chacune le caractère '-'.
5. Écrivez la fonction/procédure `afficher` prenant en paramètre un tableau de caractères et l'affichant à l'écran comme une chaîne de caractères.
6. Écrivez la fonction/procédure `jouer` prenant en paramètre la solution, le masque et un caractère, et retournant `true` si le caractère est présent dans la solution. Le cas échéant, le masque est mis à jour : le caractère doit remplacer à la bonne place le caractère '-' (pour toutes ses occurrences).
7. Écrivez la fonction/procédure `afficherEssais` prenant en paramètre le nombre d'essais déjà joués (maximum 5) et le masque, et affichant le nombre de coups restants puis l'état courant du jeu (le masque). Pour afficher une belle potence, vous pouvez utiliser le morceau de code suivant (téléchargeable sur *Moodle*) :

```
switch(nbEssais) {
case 5: // La potence en entier et le pendu
    System.out.println("_/---\\n_|_|_|_|\\n_|_|_|_|O\\n_|_|_|_|/|\\n_|_|_|_|/\\n-^-");
    break;
case 4: // La potence en entier
    System.out.println("_/---\\n_|_|_|_|\\n_|_|\\n_|_|\\n-^-");
    break;
case 3: // La potence sans la corde
    System.out.println("_/---\\n_|_|_|\\n_|_|\\n_|_|\\n-^-");
    break;
case 2: // Le corps de la potence
    System.out.println("\\n_|_|\\n_|_|\\n_|_|\\n-^-");
    break;
case 1: // Le pied de la potence
    System.out.println("\\n\\n\\n\\n\\n-^-");
    break;
}
```

8. Écrivez la fonction/procédure `estFini` prenant en paramètre le masque et qui retourne `true` si tous les caractères du mot ont été trouvés.
9. Écrivez un `main` réalisant les actions suivantes :
 - Demande du mot à deviner à l'utilisateur (fonction/procédure `initialiserSolution`)
 - Création du masque à partir de la taille de la solution (fonction/procédure `creerMasque`)
 - Boucle principale du jeu :
 - Affiche l'état courant du jeu (fonction/procédure `afficherEssais`)
 - Demande à l'utilisateur un caractère
 - Mise-à-jour du masque et du nombre d'essais (suivant si le caractère est présent ou non) (fonction/procédure `jouer`)
 - On s'arrête lorsque le nombre d'essais atteint 5 ou lorsque le jeu est fini (le mot entier a été trouvé)
 - Affichage du résultat : gagné ou perdu !

2 Le jeu du pendu et l'utilisation d'un dictionnaire



Cet exercice est facultatif.

Nous souhaitons maintenant générer des mots aléatoirement à partir d'un "dictionnaire". On considère que le dictionnaire est un tableau de chaînes de caractères, chaque case contenant un mot. Au début du jeu, un mot est choisi aléatoirement dans le dictionnaire.

1. Récupérez le fichier texte contenant des mots du dictionnaire et la classe `Exemple` sur *Moodle*. Compilez la classe et exécutez le programme. Analysez ensuite le code et comprenez son fonctionnement.
2. À l'aide de la classe `Exemple`, écrivez une fonction `lecture` dans votre classe `Pendu` qui charge le dictionnaire en mémoire (vous aurez besoin de la fonction `nbLignes`).
3. Que se passe-t-il avec les mots contenant des accents ?
4. Modifiez la fonction `initialiserSolution` de la classe `Pendu` pour que le mot soit choisi aléatoirement dans le dictionnaire.