

T9

Il n'y a pas si longtemps, taper un SMS était un vrai cauchemar : seulement 9 touches pour taper 26 lettres. Il fallait appuyer plusieurs fois sur une même touche pour entrer un caractère : par exemple, pour écrire "hello", il fallait appuyer 2 fois sur 4, 2 fois sur 3, 3 fois sur 5, puis encore 3 fois sur 5 et enfin 3 fois sur 6.

Ceci a amené les constructeurs de téléphones à trouver une manière plus simple d'entrer du texte : le T9. Le 9 signifie que vous pouvez rentrer à peu près n'importe quel mot avec seulement 9 touches en appuyant sur seulement une touche par caractère.

Le principe : on tape les touches sans répétition, et le programme utilise un dictionnaire pour trouver le mot le plus probable correspondant à ce que vous avez tapé. Par exemple, pour écrire "hello", on tape les touches 4, 3, 5, 5 et 6 une seule fois chacune. Bien sûr, cela aurait pu correspondre au mot "gdjjm", mais c'est assez peu probable..

Plus précisément, pour chaque caractère tapé, le téléphone va afficher le mot le plus probable à chaque instant. Imaginons que le téléphone connaisse le mot "idee" et "hello", avec "idee" apparaissant plus souvent que "hello", si on appuie sur les touches 4,3,5,5, et 6, l'une après l'autre, le téléphone proposera "idee", "idee", puis changera pour "hello", "hello", et finalement "hello".



L'objectif de ce TP est d'écrire une implémentation du T9 qui affiche le mot le plus probable après chaque appui sur une touche.

Dans la suite, on vous donne les fichiers suivants :

- un fichier `liste_mots.txt` qui associe à chaque mot du français un nombre qui est d'autant plus élevé que le mot est fréquent.
- un fichier `ClavierTel.java` que vous n'avez pas à modifier et qui gère l'affichage d'un clavier de téléphone avec un "écran" pour afficher des résultats.

- un fichier `ActionsBouton.java` que vous n'avez pas à modifier, et qui gère les appuis sur les touches.
- un fichier `Executable.java` que vous n'avez pas à modifier.
- un fichier `T9.java` qui est à compléter.

Premiers pas

1. Récupérez et décompressez [T9.zip](#). Vous devez retrouver les fichiers cités ci-dessus. Compilez et exécutez. Lorsque vous appuyez sur les touches, le symbole de la touche (pour le moment "P" pour chaque touche) est ajouté à l'affichage.
2. Modifiez le code de T9 pour que le symbole de chaque touche ne soit pas "P", mais le numéro de la touche.
3. Modifiez le code de T9 pour que le symbole corresponde à celui d'un téléphone portable. Que se passe-t-il lorsqu'on appuie sur la touche '*' ?
4. Pour le moment sur chaque touche, on voit le texte 'abc'. Ce texte est attribué par la méthode `getTexte`. Modifiez cette méthode pour que le texte affiché corresponde à celui d'un téléphone portable.
5. Modifiez le constructeur de T9 pour qu'il initialise et remplisse la map "dicoFreq" qui à chaque mot associe sa (pseudo) fréquence.

L'algorithme du T9

La question qui se pose à présent est de définir une méthode `getMot(String prefixe)` qui prend en entrée une suite de symboles correspondant chacun à une touche (par exemple '8686') et renvoyant le mot le plus probable pour ce préfixe.

L'idée pour cela va être de créer une map qui à un préfixe associe un mot. On l'appellera *prefixeMot* et la définira comme attribut privée de la classe T9.

1. Définissez une tel map. Initialisez la dans le constructeur (pour le moment, on ne la remplit pas).

Pour remplir cette map, on va parcourir les mots de `dicoFreq`. Par exemple, on a le mot "autrui". Pour chaque préfixe de ce mot, ici "a", "au", "aut", "autr", "autru" et "autrui", on va calculer la suite de touches correspondante. Par exemple, ici : "2", "28", "288", "2887", "28878", "288784". Pour chaque suite de touche, on va regarder dans *prefixeMot* s'il y a déjà un mot associé. Si ce n'est pas le cas, on y associe le mot courant (ici "autrui"), si c'est le cas, on choisit le mot le plus fréquent entre celui courant et celui qu'il y a déjà.

2. Pour implémenter cet algorithme, on va utiliser une méthode `toTouche` qui prend en entrée une chaîne et renvoie la suite de symboles à taper pour obtenir cette chaîne. Implémentez cette méthode.
3. Implémentez le remplissage de la map *prefixeMot* dans le constructeur de T9.

4. Modifiez la méthode `getMot` pour qu'elle renvoie le mot le plus fréquent pour une suite de symboles donnée. Vous devez obtenir un programme fonctionnel.