

Arbre

1. BUT DE L'EXERCICE

Le but de cet exercice est d'utiliser la structure arbre.

2. SPECIFICATION

Votre programme devra implémenter un arbre de recherche. Le cas de figure est celui d'un IDE qui doit afficher les prédictions de noms de fonctions (d'un langage de programmation) lorsque l'utilisateur commence à taper le nom d'une fonction.

Attention, le but n'est pas développer un IDE, juste les parties pour la prédiction.

Les exigences sont les suivantes :

❑ 1. Modéliser l'arbre

Dans cette partie, vous allez modéliser les classes `WordSearchTree` et `Node`.

Afin de simplifier, les noms des fonctions qui seront stockés dans l'arbre ne se composent que de lettres minuscules exclusivement.

❑ 2. Remplir l'arbre

Dans le programme principal, créez un objet `WordSearchTree` et remplissez-le avec le contenu du fichier `functions.txt` qui vous est donné.

❑ 3. Recherche dans l'arbre

Dans la suite du programme principal, demandez à l'utilisateur (en mode console) de saisir le nom d'une fonction, puis affichez (en parcourant votre arbre) si ce nom existe dans l'arbre (oui ou non).

❑ 4. Mesures de performances

Il s'agit ici de réaliser un autre programme principal qui va utiliser votre arbre avec le même contenu que dans l'exigence 2.

De plus, le contenu du fichier `functions.txt` sera également chargé dans un tableau (standard de votre langage de programmation).

Ensuite, le programme demandera à l'utilisateur de saisir un nom de fonction à rechercher. Votre programme utilisera et mesurera le temps passé tantôt dans l'arbre et tantôt dans le tableau, il affichera les résultats. (Attention : réfléchissez à la pertinence de ceux-ci)

❑ 5. BONUS - Prédictions

Pour les plus téméraires, modifiez l'exigence 3 (et aussi votre implémentation de l'arbre) afin que l'on demande à l'utilisateur de saisir les 3 premières lettres d'une fonction, pour ensuite afficher (en parcourant votre arbre) les noms des fonctions correspondant à ce préfixe.

L'utilisateur pourra continuer à saisir la suite du nom, lettre par lettre, avec à chaque fois le programme qui affichera les noms correspondants (à chaque lettre il doit y avoir moins de résultats).