

NFA031: projet

Année 2018-2019

1 Enoncé

Le but du projet est de réaliser un jeu inspiré du *mot le plus long* de l'émission télévisée *des chiffres et des lettres* (France 3). Dix lettres sont tirées au sort, ce qui donne un **tirage**.

L'utilisateur du programme doit former le mot le plus long possible comportant certaines ou toutes les lettres du tirage. Ce mot doit exister dans la langue française.

Le programme lit au clavier le mot trouvé par l'utilisateur, il vérifie que le mot est bien composé de lettres du tirage et affiche la correspondance des lettres du mot et du tirage. Il vérifie si le mot est bien présent dans le dictionnaire. Si les deux conditions sont réunies, il affiche la longueur du mot. Les lettres ne comportent ni accent, ni cédille, ni trémas. Elles sont écrites en lettres capitales.

Voyons un exemple :

Tirage: BCEEHIMOSS

Proposez un mot (ou une ligne vide pour passer): BOISEE

```
BOISEE
075823
0123456789
BCEEHIMOSS
```

Votre mot fait 6 lettres

La correspondance des lettres du mot avec celle du tirage est donnée en affichant sous chaque lettre du mot le numéro qu'a cette lettre dans le tirage. Chaque lettre du tirage ne doit être utilisée qu'une seule fois dans le mot. Dans notre exemple, le B est la lettre numéro 0 du tirage, le O la lettre numéro 7, etc.

Comme il y a deux fois la lettre E dans le tirage, on peut proposer des mots avec deux E mais pas de mots avec trois E.

2 Code fourni

Pour vous aider dans votre tâche, on vous fournit deux méthodes et un fichier texte contenant le dictionnaire. La première méthode, `donnerTirage`, renvoie un tirage de 10 lettres dans un tableau de caractères. Le tirage au sort tient compte de la fréquence des lettres dans la langue. Il y a ainsi beaucoup plus de chances de tirer un E qu'un Y.

La seconde méthode, `getDictionnaire`, renvoie le dictionnaire sous forme d'un tableau de caractères à deux dimensions (type `char[][]`). Cette méthode lit les mots dans le fichier **dico_nfa031.txt**. Si ce fichier n'est pas présent, une exception est levée à l'exécution. Le tableau comporte un peu plus

de 99 000 mots classés par ordre alphabétique. Il respecte les catégories acceptées par le jeu (les mots composés, les féminins et pluriels sont acceptés. La plupart des formes conjuguées des verbes et les noms propres sont refusés). Ce dictionnaire est construit à partir du dictionnaire de correction orthographique de Open Office qui a une licence libre. Il ne comporte pas les mots de plus de 10 lettres. Le jeu télévisé utilise deux dictionnaires commerciaux qui ne sont pas utilisables librement.

Nous vous conseillons de recopier le code de ces deux méthodes dans la classe que vous devez écrire pour votre projet. Notez bien qu'il n'est pas nécessaire de comprendre le code de ces deux méthodes pour les utiliser.

```
public static char[] donnerTirage(){
    double[] frequences = {
        0.084641116045612,
        0.097550346533667,
        0.134264256131715,
        0.175931501299866,
        0.339787563705305,
        0.352412789645191,
        0.366356662429924,
        0.37898710353533,
        0.454338255487703,
        0.458247915903501,
        0.461526843221907,
        0.517821090574744,
        0.547529673593674,
        0.620002807215,
        0.677444355180383,
        0.70566493929644,
        0.71299279223788,
        0.781799811286431,
        0.855650833407438,
        0.922800098125308,
        0.974199670376239,
        0.986838818540602,
        0.98881203071521,
        0.993095137557963,
        0.998288765757996,
        1};
    char[] res=new char[10];
    for (int i=0; i<res.length; i++){
        double ausort = Math.random();
        int j=0;
        while(frequences[j]<ausort)
            j++;
        res[i] = (char)('A'+j);
    }
    java.util.Arrays.sort(res);
    return res;
}
```

```

}

public static char[][] getDictionnaire(){
    char[][] res;
    char[][] temp = new char[500000][];
    int nbMots = 0;
    try{
        java.io.FileReader fr =
            new java.io.FileReader(" dico_nfa031 . txt ");
        java.io.BufferedReader br =
            new java.io.BufferedReader(fr);
        String ligne = br.readLine();
        while(ligne != null){
            temp[nbMots] = ligne.trim().toCharArray();
            nbMots++;
            ligne = br.readLine();
        }
        br.close();
    }
    catch(java.io.FileNotFoundException e){
        String str = " dico_nfa031 . txt non trouvé ";
        throw new RuntimeException(str);
    }
    catch(java.io.IOException e){
        throw new RuntimeException("Problème lecture fichier");
    }
    res = new char[nbMots][];
    for (int i=0; i<nbMots; i=i+1){
        res[i]=temp[i];
    }
    return res;
}

```

Pour lire au clavier le mot de l'utilisateur, vous pouvez utiliser `Terminal.lireString` puis transformer la chaîne de caractères obtenue en tableau du char avec la méthode `toCharArray`.

```

char[] str;
str = Terminal.lireString().toCharArray();

```

3 Fonctionnalités optionnelles

Votre programme peut chercher dans le dictionnaire le ou les mots les plus longs pour le tirage donné.

Vous pouvez travailler pour mettre en place des méthodes de recherche efficaces dans le dictionnaire (recherche dichotomique ou mise en place d'un index).

4 Contraintes

Vos structures de données ne devront comporter que des **tableaux déclarés dans la méthode main**. Les structures devront respecter le type des données en évitant de faire des codages et décodages. Par exemple, représenter un nombre entier ou un booléen par une chaîne de caractère est totalement déconseillé. Votre programme devra comporter des méthodes **recevant les tableaux en paramètres**. Les `ArrayList` et l'utilisation de la classe `Arrays` sont **interdits**. Vous pouvez utiliser les méthodes de la classe `String`.

Vous présenterez votre programme avec une **indentation correcte**. Vous n'écrirez aucune méthode de plus de 30 lignes (y compris la méthode `main`). Vous mettrez peu ou pas de commentaires dans le corps des méthodes, mais vous pourrez mettre un commentaire pour chaque méthode avant son code.

Vous essayerez d'utiliser des exceptions dans votre programme.

5 Consignes

Le projet est individuel : chaque élève doit le réaliser et doit en écrire seul chacune des lignes de code. Le projet donne lieu à une soutenance au cours de laquelle l'élève doit justifier les choix qu'il a fait.

Pendant la période de préparation du projet, vous pouvez demander l'aide des enseignants de TP et du tuteur pour les élèves à distance. Vous pouvez lui poser toutes les questions et lui soumettre tous vos problèmes.

6 Remise et soutenance du projet

Le projet donnera lieu à deux rendez-vous avec votre enseignant (chargé de TP ou tuteur pour les auditeurs à distance). Le premier concernera l'analyse du projet et la structure de donnée choisie pour représenter le questionnaire. Le second évaluera le programme terminé et fonctionnel rendu sous la forme d'un fichier source `.java`.

Dates importantes :

- Présentation de l'analyse du projet : semaine du 7 janvier. Pour les élèves en présentiel, le jour de votre TP. Pour les élèves à distance : avant le mercredi 9 janvier à minuit.
- Remise du projet terminé : semaine du 21 janvier. Pour les élèves en présentiel, le jour de votre TP. Pour les élèves à distance : avant le mercredi 23 janvier à minuit.

Pour les élèves en présentiel, les modalités précises des deux étapes seront données par l'enseignant de TP. Pour les élèves à distances, il y aura remise de documents via un formulaire Moodle et la deuxième étape donnera lieu à une soutenance sur rendez-vous soit en présentiel à Paris, soit en visio-conférence de chez vous.