

Nom : \_\_\_\_\_  
 Prénom : \_\_\_\_\_  
 Groupe : \_\_\_\_\_  
 Identifiant : \_\_\_\_\_

/ 50

Haute École de Bruxelles  
 École Supérieure d'Informatique  
 Bachelor en Informatique

2014 – 2015

## DEV1 – Développement

### Examen de Janvier

### AEBBCLRS

Vendredi 23 janvier 2015

#### Consignes

L'examen dure 4h. Gérez bien le temps afin de pouvoir répondre à l'ensemble des questions (algorithmique et langage Java)

Le Scrabble est un jeu de lettres, où le but est de former des mots ayant le score le plus élevé possible. Pour calculer le score d'un mot, une valeur est associée à chaque lettre de l'alphabet. Par exemple la lettre A vaut 1, V vaut 4, J vaut 8,...Le score d'un mot est la somme de la valeur de ses lettres<sup>1</sup>.

Par exemple le score du mot JAVA est  $8 + 1 + 4 + 1 = 14$ .

Les joueurs tirent chacun 7 lettres qu'ils placent sur un *chevalet* (voir image ci-dessous). Le but du jeu est de former des mots à partir des lettres du chevalet et de les placer sur le plateau de jeu.



FIGURE 1 – Le plateau de Scrabble et un chevalet

1. dans le vrai jeu c'est un peu plus compliqué.

# I

## Algorithmique

### Consignes

Pour la partie algorithmique,

- Vous travaillez sur papier, vous ne pouvez pas utiliser de notes ni d'ordinateur.
- Vos réponses se feront au bic bleu ou noir sur la feuille de réponses.
- Sauf spécification du contraire, les données lues ou reçues ne comportent pas d'erreurs.
- Les noms en gras (variables et types) doivent être respectés.
- Veillez à travailler de manière modulaire.

1

### Valeur des lettres

(4 points)

Écrire un algorithme, **valeurLettre**, qui reçoit un caractère et retourne sa valeur. Les valeurs des lettres sont les suivantes :

- A,E,I,L,N,O,R,S,T,U : 1 point
- D,G,M : 2 points
- B,C,P : 3 points
- F,H,V : 4 points
- J,Q : 8 points
- K,W,X,Y,Z : 10 points

L'algorithme lance une erreur avec un message adéquat si le caractère passé en paramètre n'est pas une lettre majuscule. Vous aurez besoin de l'algorithme qui teste si un caractère est une majuscule :

**module** *estMajuscule*(lettre : caractère) → booléen

2

### Score d'un mot

(6 points)

Écrire un algorithme, **scoreMot**, qui reçoit une chaîne de caractères, **mot**, et retourne la valeur de ce mot, c'est-à-dire la somme de la valeur de chacune de ses lettres.

Exemple : si l'algorithme reçoit le mot **JAVA**, il retourne 14 (= 8+1+4+1).

Rappel : En pseudo code, les chaînes sont indicées de 1 à n (ou n est la longueur de la chaîne). Vous aurez sans doute besoin de l'algorithme permettant d'obtenir la longueur d'une chaîne et celui permettant d'obtenir le i<sup>e</sup> caractère d'une chaîne :

**module** *longueur*(mot : chaîne) → entier

*nomChaine[i]* désigne le i<sup>e</sup> caractère de la chaîne *nomChaine* (en commençant à 1)

**3****Mot possible**

(8 points)

Écrire un algorithme, `motPossible`, qui reçoit un tableau de caractères, `chevalet`, et une chaîne de caractères, `mot`, et retourne vrai si les lettres du chevalet permettent d'obtenir le mot donné, et retourne faux dans le cas contraire.

Exemple : si l'algorithme reçoit le tableau `[A, O, V, G, G, J, L]` et le mot `"ALGO"`, l'algorithme retourne vrai. Par contre si le mot donné est `"JAVA"`, alors l'algorithme retourne faux car la lettre A n'apparaît pas 2 fois dans le chevalet.

Aide : Une solution possible est de faire une copie du chevalet et pour chaque lettre du mot de vérifier que la lettre s'y trouve, si elle s'y trouve de la remplacer par un symbole (par exemple un `'-'`).

Pour cela on définit un algorithme `copieChevalet` permettant de copier un chevalet et une méthode `indiceLettre` permettant de connaître l'indice d'une lettre dans un chevalet (cet algorithme retourne -1 si la lettre ne s'y trouve pas).

**4****Meilleur mot**

(7 points)

Écrire un algorithme, `meilleurMot`, qui reçoit :

- un tableau de caractères, `chevalet`, qui contient les lettres disponibles pour former un mot ;
- un tableau de chaînes de caractères, `dico`, qui contient par exemple tous les mots du dictionnaire.

L'algorithme retourne le mot du dictionnaire que l'on peut obtenir avec les lettres du chevalet et qui a le score le plus élevé. Si aucun mot n'est possible avec le chevalet, l'algorithme retourne un mot vide `""`.

## II

### Java et laboratoire

#### Consignes

Pour la partie java,

- Vous réaliserez votre travail sur **linux1** et le déposerez dans le casier **linux** de votre professeur par la commande **casier**.
- Vous disposez de toutes vos notes ainsi que de l'aide en ligne.
- Il ne suffit pas que votre code compile. Testez-le pour identifier d'éventuelles erreurs à l'exécution.
- La cotation tiendra compte aussi du style de programmation que vous avez acquis.
- Respectez bien les noms de package, classe, méthodes demandés dans l'énoncé.
- Vous remplacerez bien sûr **g12345** par votre numéro d'étudiant.

5

#### Répertoire de travail et permissions

(0 point)

Créez un répertoire **evaluations/janvier**.

Changez les droits sur votre répertoire **janvier** pour donner les permissions de lecture et d'exécution aux professeurs mais aucun droit aux autres étudiants. Appelez votre professeur pour lui montrer que vos permissions ont bien été changées.

Vous ne continuerez pas l'examen tant que cette question n'a pas été validée par votre professeur.

6

#### Préparation de l'environnement

(2 points)

Dans la suite de l'examen, votre classe fera partie du package **g12345.dev1.janvier**. Votre programme s'appellera **Scrabble.java**. Il sera situé dans **~/evaluations/janvier/sources** et la version compilée sera dans (un sous-dossier de) **~/evaluations/janvier/classes**.

Écrivez ici :

- l'instruction que doit contenir votre classe pour faire partie du package demandé ;
- la commande (complète et précise) que vous allez utiliser pour compiler votre classe ;
- la commande (complète et précise) que vous allez utiliser pour exécuter votre classe ;
- le contenu minimal de votre variable d'environnement **CLASSPATH**.

7

#### La méthode **valeurLettre**

(3 points)

Dans la classe **Scrabble** ajoutez une méthode **valeurLettre** définie à la Question 1 (partie ALM).

La méthode lance une **IllegalArgumentException** avec un message adéquat si le caractère passé en paramètre n'est pas une lettre majuscule.

Pour tester en java si un caractère est une lettre majuscule ou utilise la méthode `isUpperCase` de la classe `Character` qui se trouve dans le package `java.lang`.

8

### La méthode `scoreMot`

(3 points)

Dans la classe `Scrabble` ajoutez une méthode `scoreMot` définie à la Question 2 (partie ALM).

En java pour obtenir la longueur d'un `String` et obtenir le 4<sup>e</sup> caractère, on fait :

```
1 String mot = "laboratoire";  
2 int longueur = mot.length(); // longueur vaut 11.  
3 char lettre = mot.charAt(3); // lettre 'o', car on commence a 0.
```

9

### La javadoc

(3 points)

Écrivez la javadoc de la classe `Scrabble` et de ces méthodes `valeurMot` et `scoreMot` et générez la dans le répertoire `~/evaluations/janvier/doc`

Quelle commande utilisez-vous pour générer cette documentation ?

10

### Les tests

(4 points)

1. Écrivez, dans une classe `ScrabbleTest`, des tests (2 tests) pour la méthode `scoreMot`.
2. Exécutez ces tests en redirigeant les sorties (standard et erreur) vers le fichier `tests.log`.
3. Écrivez ci-dessous la commande que vous avez utilisé.

11

### La classe `ScrabbleUtil`

(1 point)

Nous vous fournissons une classe `ScrabbleUtil` qui se trouve dans le package `esi.dev1.janvier` avec les méthodes suivantes :

- `boolean motPossible(char[] chevalet, String mot)` :  
retourne vrai si les lettres du chevalet passé en paramètre permettent d'obtenir le mot passé en paramètre (Question 3 de la partie ALM) ;
- `char[] tirerChevalet()` :  
retourne un tableau de 7 lettres tirés au hasard (le chevalet) ;
- `String[] lireDico()` :  
retourne un tableau de `String` comprenant 59232 mots du dictionnaire ;
- `void main(String[] args)` :  
méthode principale qui vous permet de tester cette classe ;

Trouvez cette classe qui se trouve dans `/eCours/java/` et exécutez la (sans la copier dans vos répertoires).

- qu'affiche le programme lorsque vous exécutez cette classe ?
- quel est le contenu minimal de votre variable d'environnement `CLASSPATH` pour pouvoir l'exécuter ?

**12****La méthode meilleurMot**

(4 points)

Dans la classe `Scrabble` ajoutez une méthode `meilleurMot` définie à la Question 4 (partie ALM).

Pour cela vous utiliserez la méthode `motPossible` de la classe `ScrabbleUtil` et la méthode `scoreMot` de la classe `Scrabble` que vous avez écrite à la Question 8.

**13****La méthode principale**

(5 points)

Dans la classe `Scrabble` écrivez une méthode principale qui :

1. lit le dictionnaire ;
2. demande un chevalet aléatoire (méthode `tirerChevalet` de `ScrabbleUtil`) ;
3. demande un mot à l'utilisateur ;
4. vérifie que le mot est possible avec le chevalet ;
5. si le mot est possible, affiche le score du mot proposé, et sinon un message adéquat ;
6. cherche le meilleur mot dans le dictionnaire pour ce chevalet ;
7. affiche ce meilleur mot et son score ;
8. demande au joueur s'il veut continuer, si oui retourne en 2, sinon termine.