

Mode d'emploi

Codage de Huffman

Java



Sommaire

Installation / Lancement.....	P.3
I. Prérequis.....	p. 3
II. Choix du répertoire.....	p. 3
III. Compilation et exécution.....	p. 4
Utilisation.....	P.5
I. Instancier la classe principale.....	p. 5
II. Encoder.....	p. 6
III. Décoder.....	p. 7
IV. Changer les caractères et/ou les fréquences utilisées.....	p. 8
V. Pour aller plus loin.....	p. 9
V.1. Accéder à l'arbre d'Huffman.....	p. 10
V.2. Accéder aux codes des caractères.....	p. 11
V.3. Accéder aux racines.....	p. 12

Installation / Lancement

I. Prérequis

Vous devez installer préalablement le JRE contenant la machine virtuelle java (JVM) dans votre environnement d'exécution.

II. Choix du répertoire

Téléchargez le package contenant les 4 fichiers :

- Arbre.java
- Liste.java
- ListeArbre.java
- CodageHuffman.java

Placez ces fichiers dans le répertoire de votre choix.

Installation / Lancement

III. Compilation et exécution

Deux méthodes possibles :

1. Si vous possédez un IDE (exemple : Eclipse), ouvrez un nouveau projet et incorporez y les fichiers précédents. Appuyez sur la flèche de compilation/exécution (généralement verte) lorsque vous vous trouvez dans la classe principale (CodageHuffman).
2. Sinon avec votre Console/Terminal, placez vous dans le répertoire précédent. Effectuez la compilation manuellement, vous devriez obtenir un .class :

```
~$ javac *.java
```

Exécutez la classe principale (CodageHuffman) :

```
~$ java CodageHuffman
```

Utilisation

I. Instancier la classe principale

Commencez par créer 2 tableaux :

- Un premier contenant tous les caractères que vous souhaitez
- Un deuxième contenant les fréquences d'apparition associées à chacun de ces caractères dans le même ordre.

Puis, créer un objet « Codage Huffman ».

`CodageHuffman(char[] caractères, double[] fréquences)`

Exemple :

```
1. char caracteres[] = {'a', 'b', 'c', 'd', 'e'};
2. double frequencies[] = {9, 5, 7, 2, 12};
3. // ou avec des virgules {20.06, 16.94, 19.1, 10.4, 33.5} (souvent utile dans le cas d'un %)
4. CodageHuffman secret = new CodageHuffman(caracteres, frequencies)
```

Utilisation

II. Encoder

Pour encoder vous devez appeler la méthode de classe « encoder » qui prend en argument un texte.

```
public String encoder(String texte)
```

Exemple :

```
1. String texteInitial = "Hello world !";  
2. String texteEncode;  
3. texteEncode = secret.encoder(texteInitial);  
4. System.out.println(texteEncode);
```

*« secret » est une instance de la classe « CodageHuffman », voir page 5.

Utilisation

III. Décoder

Pour Décoder vous devez appeler la méthode de classe « Decoder » qui prend en argument un texte.

```
public String Decoder(String texte)
```

Exemple :

```
1. String texteInitial = "1010101110001100111001110111010011";  
2. String texteDecode;  
3. texteDecode = secret.decoder(texteInitial);  
4. System.out.println(texteDecode);
```

*« secret » est une instance de la classe « CodageHuffman », voir page 5.

Utilisation

IV. Changer les caractères et/ou les fréquences utilisées

Pour changer les racines (de l'arbre d'Huffman) sans créer une deuxième instance de « CodageHuffman » il est nécessaire d'appeler la méthode de classe « setRacines ». Cette méthode prend comme le constructeur de la classe, un tableau de caractères et un tableau de fréquences.

```
public void setRacines(char[] caractères, double[] fréquences)
```

Exemple :

```
1. char caracteres2[] = {'d', 'e', 'f', 'g', 'h', '1', '2', '3'};  
2. double frequences2[] = {8, 14, 4, 2, 2, 3, 1, 6};  
3. secret.setRacines(caracteres2, frequences2);
```

*« secret » est une instance de la classe « CodageHuffman », voir page 5.

Utilisation

V. Pour aller plus loin

L'encapsulation des classes du package et plus particulièrement la classe « CodageHuffman » est faite de façon à simplifier au mieux son utilisation.

En effet, l'ensemble des opérations (créer la liste de racines, créer l'arbre d'Huffman, créer le tableau de codages des valeurs) est fait en interne de la classe (private).

Ainsi, l'utilisateur n'a besoin que d'écrire le minimum dans le « main » pour faire fonctionner le programme. De plus cela limite les erreurs de l'utilisateur.

Toutefois, si vous voulez accéder à ces éléments, des accesseurs ont été implémentés à cet effet.

Utilisation

V.1. Accéder à l'arbre d'Huffman

Pour accéder à l'arbre vous devez appeler l'accessesseur « getArbre ».

```
public Arbre getArbre()
```

Exemple :

```
1. Arbre arbreHuffman = secret.getArbre();  
2. arbreHuffman.affichePrefixe();
```

« affichePrefixe() » est une méthode de la classe Arbre, elle affiche de façon préfixée les nœuds de l'arbre, « afficheInfixe() » est également implémenté.

*« secret » est une instance de la classe « CodageHuffman », voir page 5.

Utilisation

V.2. Accéder aux codes des caractères

Pour accéder aux codes des caractères vous devez appeler l'accessor « `getCodes` ».

```
public ArrayList<String[]> getCodes()
```

Exemple :

```
1. for(String[] code : secret.getCodes())  
2.     System.out.println(code[0] + " - " + code[1]);
```

Les codes sont stockés dans un tableau à deux dimensions de chaîne de caractères.

*« `secret` » est une instance de la classe « `CodageHuffman` », voir page 5.

Utilisation

V.3. Accéder aux racines

Pour accéder aux racines (arbre n'ayant pas de fils gauche, ni de fils droit) vous devez appeler l'accessor « `getRacines` ».

```
public ListeArbre getRacines()
```

Exemple :

```
1. ListeArbre racines = secret.getRacines();  
2. ListeArbre.affiche(racines);
```

« `affiche()` » est une méthode statique de la classe « `ListeArbre` », qui permet d'afficher une liste d'arbre récursivement.

*« `secret` » est une instance de la classe « `CodageHuffman` », voir page 5.