

IFT 1020

Programmation II

Etienne Bergeron

8 juillet 2004

Travail pratique #4

Technique de recherche, Entrées/Sorties et Mécanismes d'exception

1 Explications générales

Équipe : Le travail est à faire en binôme ou en monôme, mais pas plus de deux.

Remise : Deux remises à effectuer : Électronique le **26 juillet à 22h30** au plus tard, sans possibilité de retard et papier au début du cours le **27 juillet 2004**.

Conseil : N'attendez pas à la dernière minute pour faire votre travail, on ne sait jamais les surprises qui nous attendent ! Aussi, lisez **attentivement** l'énoncé et prenez bien conscience du problème avant de vous lancer dans la programmation.

2 But

Le but du travail est de réaliser un décodeur de code Morse. Il vous permettra de vous familiariser avec les techniques de recherche, le modèle d'entrées/sorties et les mécanismes d'exception de Java.

3 Énoncé

3.1 Mise en situation

Lors de la première guerre mondiale, de nombreux télégrammes ont été interceptés et conservés dans les archives du département de la défense nationale. Le gouvernement a décidé de rendre publique tous les messages en les publiant. Le travail étant trop ardu pour être réalisé à la main, vous avez été engagé pour écrire un programme qui décode automatiquement les messages Morses et les formate correctement, prêts pour la publication.

3.2 Exécution de l'application

Votre programme doit décoder les messages selon les paramètres donnés à votre application sur la ligne de commande. Si l'élément est un fichier, il est décodé. Si l'élément est un

répertoire, votre programme parcourt récursivement le répertoire et décode les fichiers. Si aucun élément n'est passé en paramètre, le répertoire courant est assumé comme paramètre.

Lorsque le traitement d'un répertoire ou d'un fichier génère une erreur (exception) vous devez écrire sur la sortie standard le nom du fichier en erreur et continuer **normalement** le traitement sur le prochain fichier. Un fonctionnaire du gouvernement s'occupera de régler les problèmes avec les messages qui comportent des erreurs. Votre application doit continuer le traitement pour convertir le maximum de fichier possible.

```
% java Tp4
% java Tp4 .
% java Tp4 test message1.in
```

3.3 Table de conversion Morse

Un fichier nommé "Morse.txt", disponible sur la page WEB du cours, contient une table de conversion des caractères Morses. Vous devez charger cette table à partir du fichier et vous construire une table de conversion en mémoire. Le fichier sera présent dans le répertoire courant. En l'absence de ce fichier, l'application se termine en donnant un message d'erreur compréhensible.

Vu le grand nombre de messages à traduire, la performance de votre algorithme de conversion est importante. On vous demande donc d'implanter un algorithme de tri et un algorithme de recherche pour trouver rapidement un patron Morse et son caractère équivalent.

Chaque ligne du fichier contient un patron d'encodage. Le premier caractère est le caractère ASCII décodé et est suivi par le patron Morse.

Contenu du fichier "Morse.txt" :

```
A .-
B -...
C -.-.
D -..
[...]
```

3.4 Conversion Morse

Les messages Morses sont contenus dans des fichiers portant l'extension ".in". Votre programme devra convertir ce fichier et produire le résultat dans un fichier portant l'extension ".out".

La conversion de message Morse consiste à découper le contenu du fichier en patron Morse et à trouver le caractère ASCII équivalent. Dans un message Morse, tous les patrons sont séparés par un espace. Les mots sont séparés par plusieurs espaces. Il n'y a pas de conversion à faire pour les retours de chariot, ils sont simplement remis tel quel dans le fichier produit.

Voici le contenu du fichier "Exemple.in" :

```
...- --- .. -.-. .. ..- -. . -.-. . -- .--. .-.. .
```

et voici le résultat du fichier "Exemple.out" produit par votre application :

```
VOICI UN EXEMPLE
```

Si une erreur survient, la conversion du fichier est interrompu.

3.5 Formattage

Comme les messages Morses étaient imprimés sur une vieille imprimante qui ne possède que 40 colonnes, on vous demande de formater le message de manière à ce qu'aucune ligne ne soit plus longue que 40 caractères. Vous devez couper la ligne au dernier mot qui ne rentrait pas sur la ligne. Un retour de chariot est ainsi introduit pour mieux formater le texte.

Pour réaliser cette tâche, on vous demande d'écrire une classe qui étend (extends) la classe abstraite `Writer`. Ainsi, vous pourrez utiliser votre formatteur de la façon suivante :

```
PrintWriter pw = new PrintWriter(  
    new WordWrapWriter(  
        new FileWriter( fichier ) ));  
pw.println("Une longue ligne qui sera découpée " +  
    "automatiquement par notre formatteur.");
```

4 Rapport

Le présent travail est destiné à un milieu gouvernemental. Donc, il est important de bien documenter votre programme et de fournir un bon rapport. Votre salaire dépend de votre capacité à expliquer ce que vous avez réalisé.

Votre programme **doit** être documenté avec les commentaires à la javadoc. La remise de la javadoc n'est pas obligatoire. De plus, vous devez fournir un document en deux sections. La première est un manuel de l'utilisateur et la seconde un manuel du programmeur.

Le manuel de l'utilisateur explique aux fonctionnaires qui vont utiliser votre programme comment fonctionne votre programme et ce qu'il fait. Vous ne devez pas prendre pour acquis qu'il a lu l'énoncé. Le manuel du programmeur consiste à expliquer la structure de votre programme. Un programmeur ayant de bonnes connaissances en Java devrait pouvoir comprendre rapidement la structure de votre programme et les algorithmes utilisés. De plus, on vous demande d'analyser la performance de vos algorithmes de tri et de recherche.

5 Remise

Votre programme principal **doit** porter le nom `Tp4` et **doit** être contenu dans le fichier `Tp4.java`. Vous pouvez remettre plusieurs fichiers.

Il est important de noter que votre TP sera exécuté par les démonstrateurs sur une machine Linux avec le compilateur `jdk (1.4.2_01)`. Si par choix vous décidez d'utiliser un autre compilateur, vérifiez que le code que vous avez produit (qui normalement fonctionne correctement chez vous) fonctionne bien sur les ordinateurs du DIRO. Avant de faire cela, assurez-vous d'abord que vous avez la bonne version de `jdk`, en utilisant pour cela la commande : `"java - version"` devra donner le numéro de version `"1.4.2_01"`.

La remise comprend 3 (TROIS) choses :

1. Envoyez vos fichiers par la procédure de remise électronique habituelle (Pour obtenir de l'aide sur cette commande, lancer la commande : `man remise`).

Respectez les noms des fichiers. Cette remise est due pour le **26 juillet avant 22h30**

remise diff1020 Tp4.java AutreFichier.java Foo.java Bidon.java

2. Remettez une copie papier d'un rapport qui devrait décrire votre programme. Cette remise doit être faite le **29 juillet 2004**, au début du cours.

3. N'oubliez pas de remettre avec votre rapport une copie papier de votre programme Java.

6 Barème

Ce TP4 est noté sur 14 points. Le barème est le suivant :

1. Code

- (a) 2 points sur le fonctionnement de la conversion.
- (b) 4 points sur l'algorithme de tri et de recherche.
- (c) 2 points sur la recherche récursive des fichiers.

2. Rapport

- (a) 2 points sur le manuel de l'utilisateur.
- (b) 2 points sur le manuel du programmeur.
- (c) 2 points sur la qualité de présentation.

À noter que :

1. La non remise électronique (volontaire ou par erreur) est sanctionnée par la note 0.
2. Un programme qui ne compile pas : 0/14
3. Un programme qui compile mais ne fait pas les choses prévues dans la spécification : 0/14
4. La non remise papier vous pénalise de 6 point.
5. Les programmes ne contenant pas d'en-tête, -4 points.
6. Le code doit être lisible, bien indenté et commenté.

7 Des questions à propos de ce TP ?

L'adresse email : dift1020@iro.umontreal.ca

Pour faciliter le traitement de votre requête, inclure dans le sujet de votre email, au moins la chaîne :
"[IFT1020]"

8 Mise à jour

08-07-2004 diffusion