Fichiers : extensions et type MIME[1]

Projet POO & Java 2019-2020 version 1.0 du 05/11/2019

Objectifs

Le projet du module POO - Java permet de mettre en œuvre les principaux éléments du contenu du module dans le cadre de la conception d'une petite application. Le travail est à réaliser en binôme (exceptionnellement seul, pour les situations particulières[2]). Les binômes sont à constituer **dans** les groupes de TD.

Rappel des MCC[3] : le projet compte pour 1/3 de l'évaluation du module « POO / Java », il n'y a pas de seconde session pour le projet et la note obtenue est reportée en session 2.

Contexte du projet

Une croyance répandue est que l'extension d'un fichier permet de déterminer son type : tant que le type effectif d'un fichier et son extension sont cohérents, cette affirmation s'avère vraie, par contre dès que l'extension n'est plus en cohérence avec le contenu, il est nécessaire de s'appuyer sur le type MIME pour évaluer un fichier (c'est le cas en général des échanges de contenus multimédias via le mail ou encore la façon dont les fichiers sont traités par les navigateurs lorsqu'on consulte une page Web). Sous Linux, la commande « file » en ligne de commande permet de connaître le type MIME d'un fichier.

L'objectif du projet est de créer une petite application permettant de détecter des anomalies sur les fichiers de données[4] de l'utilisateur, donc soit de vérifier un fichier spécifique (détection de fichiers vides, détection d'incohérence entre extension et type MIME, analyse approfondie du contenu d'un fichier), soit d'explorer et analyser un ensemble de fichiers à partir d'un dossier.

Les principales actions de votre logiciel sont :

- la vérification d'un fichier de la machine (sur la base de l'extension et du type MIME du fichier)
- l'exploration complète (un dossier et l'ensemble de ses sous-dossiers) d'un répertoire avec vérification de chaque fichier.
- La validation de l'extension et du type MIME sur la base d'une liste de signatures à

1 MIME : Multipurpose Internet Mail Extensions
2 Cas des étudiants en contrôle Terminal, AJAC ou nombre impair d'étudiants sur l'ensemble de la promotion par exemple.
3 cf. https://www.u-cergy.fr/_resources/OFFRE%2520DE%2520FORMATION/MCC_Reglement%2520CC/Licences/UFR%2520Sciences%2520et%2520techniques/4-CFVU-15-05_UFR-ST-29-03_MCC_Definitives-Licences_2018-2019.pdf?download=true
4 On ne considérera pas l'analyse des programmes (de type .exe, .jar, .class, etc.).

créer (exemple : un fichier d'extension « .html » et de type MIME « text/html » devrait commencer[5] par « <!DOCTYPE html> » ; un fichier script d'extension « .sh » et de type MIME « application/x-shellscript » ou « application/x-shell » devrait commencer[6] par « #!/bin/ » (par exemple #!/bin/bash).

- Pour certains type de fichiers, la vérification de leur intégrité : principe : le programme peut lancer la décompression d'un fichier compressé (de type zip par exemple) afin de s'assurer qu'il s'agit bien d'un fichier zip, de même, les fichiers des suites bureautiques (docx, xlsx, pptx / odt, ods, odp) sont en réalité des fichiers zip que l'on peut vérifier de la même manière. Pour les images, on peut récupérer les dimensions afin de s'assurer qu'il s'agit bien d'une image, etc.

Les fichiers de taille 0 (zéro) octet devront être signalés à l'utilisateur, de même, votre programme devra détecter si l'extension d'un fichier ne correspond pas à son type MIME ; enfin, si le contenu d'un fichier ne correspond pas aux informations détectées à priori (« extension et type MIME »), le programme devra le détecter et en informer l'utilisateur.

En mode console (terminal) « CLI[7] » : les paramètres attendus sur la ligne de commande sont : le nom du fichier à traiter ou le nom du dossier à explorer : vous utiliserez un paramètre supplémentaire permettant de spécifier le type d'entrée (« -f » pour un fichier, « -d » dans le cas d'un répertoire). Le programme doit afficher directement dans la console le résultat de son analyse. Si aucun paramètre n'est indiqué (ou option « -h » pour « help »), le programme affiche l'aide et les options possibles. Une option supplémentaire « -s » permet de sauvegarder le résultat d'une analyse dans un fichier à spécifier par l'utilisateur.

En mode graphique « GUI[8] » : l'exploration d'une arborescence quelconque[9] de fichiers permettra de lister dans l'interface graphique tous les fichiers (mais pas les répertoires et sous-répertoires) et leurs emplacements. Pour chaque fichier, le niveau de conformité entre l'extension, le type MIME et le contenu sera indiqué (les fichiers posant problème seront visuellement identifiables).

La base de signatures, i.e. : les extensions, types MIME associés et contenu d'identification si il y a lieu, seront stockés sous la forme d'un fichier CSV.

Quelques scénarios d'exécution (exemples fictifs de lancement de vos 2 programmes[10]) :

```
java -jar cli.jar
java -jar cli.jar -d .
java -jar cli.jar -f test.html
java -jar cli.jar -d . -s analyse
```

5 Après éventuellement des commentaires de type « < !-- … --> »
6 Après éventuellement des lignes de commentaires précédées du caractère « # »
7 CLI : *Command Line Interface*
8 GUI : *Graphical User Interface*
9 Avec exploration de l'ensemble des sous-répertoires.
10 *cli : command line interface / gui : graphical user interface*

```
java -jar gui.jar
```

Explications : les 4 premières commandes concernent le mode console (aussi appelé mode terminal ou fenêtre de commande) ; la dernière commande permet de lancer l'interface graphique.

- la première ligne affiche les modes d'utilisation de votre logiciel en mode console (i.e. les options possibles et leur rôle) ;
- la deuxième ligne liste et analyse tous les fichiers **à partir** du dossier spécifié (*« -d » = directory*) [ici à partir du dossier courant (« . »)] en parcourant l'ensemble de l'arborescence des sous-dossiers, en mode console ;
- la troisième ligne prend en entrée le ficher « test.html » (*« -f » = file*) et affiche à l'écran le résultat de sa vérification, en mode console ;
- la quatrième ligne prend en entrée le dossier courant « . » et sauvegarde le résultat de la vérification dans un fichier « analyse » (*« -s » = save*) en mode console ;
- la dernière ligne correspond au lancement de l'interface graphique.

N.B. : le format de sauvegarde des résultats d'une analyse est laissé à votre initiative mais il est nécessaire de pouvoir exploiter ce résultat en dehors de votre logiciel : une attention particulière est donc à porter à ce choix.

En mode graphique, il est demandé de pouvoir conserver une version sérialisée de l'analyse entre 2 exécutions du programme afin de pouvoir dissocier l'analyse (éventuellement longue), de l'affichage des problèmes repérés. Au lancement du logiciel en mode graphique, c'est donc le dernier résultat obtenu qui est affiché, avec la possibilité de lancer une nouvelle analyse.

Extensions possibles : vous pouvez prévoir des améliorations à votre solution mais **uniquement** si tout le reste est complet. Vous privilégierez donc la qualité de la réalisation à la quantité de fonctionnalités.

Planning et conseils pour la présentation

Planning

- Identifier les principales sous-tâches du projet à réaliser, leur niveau de priorité, la répartition des rôles au sein du binôme ainsi que le planning correspondant pour chaque tache (période et durée). Vous pouvez créer un petit tableau récapitulatif ou (mieux), créer un diagramme de GANTT[11] de votre projet. Dans tous les cas, le tableau ou le diagramme de Gantt est à rendre au format png (capture ou export) avant le samedi 16 novembre 2019 : **1 point**
- points d'avancement en décembre (présence obligatoire) : **3 points** (à titre indicatif, un niveau de réalisation d'environ 50% est attendu lors du dernier point d'avancement)

11 cf. https://www.ganttproject.biz/download/free

- soutenance (5 min) et démonstration (5 min) : **6 points** : lundi 6 ou mercredi 8 janvier 2020.

Soutenance : 5 diapositives maximum, 5 minutes maximum

- la diapositive de titre présentera le binôme, le contexte, le sujet. (i.e. : la page de garde sera compacte),
- les autres diapositives devront présenter les spécificités de réalisation de l'équipe projet, donc aucune information "évidente" (ex. détail du sujet, progression personnelle, ...) ne devra être mentionnée,
- la diapositive de conclusion mettra en évidence le niveau d'achèvement du projet (points traités et non traités du cahier des charges et extensions si il y en a),
- vous devrez prévoir une version pdf de votre diaporama au cas où.

Les informations suivantes devront être présentes : la répartition des tâches, les principaux éléments de conception.

NB : à éviter ABSOLUMENT : les diagrammes de classes UML illisibles (trop chargés, ...), les programmes (code Java), les captures d'écran (puisqu'il y a aussi une démonstration), la liste des outils (ex. Eclipse, etc.), ...

Important : vous devrez avoir votre machine portable allumée, prête avec l'ensemble des logiciels déjà lancés AVANT d'entrer dans la salle. Votre portable devra disposer d'un port VGA ou vous devrez prévoir un adaptateur VGA correspondant à votre situation.

Démonstration : 5 minutes maximum

1. mode console,
2. mode graphique.

Vous veillerez à prévoir un scénario pour la démonstration.

Résultats attendus et critères d'évaluation

Complétude et qualité du projet : 7 **points** (fichier « readme.txt », code java, javadoc, fichiers jar).

Livrables à déposer sur la plate-forme pédagogique avant le **vendredi 27 décembre 2019 (à midi)**.

- fichier « readme.txt » contenant les noms / prénoms / groupe TD / des membres du projets ainsi que les informations spécifiques utiles,
- rapport de projet (minimum 5 pages, maximum 10 pages) : **3 points** (le fond et la forme seront évalués). Les 2 fichiers suivants sont à rendre : 1) le document de traitement de texte (**docx ou odt**), 2) la version **pdf** de votre rapport

- ensembles des fichiers sources du projet (.java),
- la javadoc,
- Les 2 fichiers jar en version compatible java 1.8.

Important : Tous les fichiers et sous-dossiers à remettre doivent être placés dans un répertoire unique portant les 2 noms du binôme (sous la forme NOM1_NOM2) à compresser en un seul fichier au format zip qui sera déposé sur la plate-forme pédagogique de cours *(moodle)*.

Quelques indications pour la réalisation

L'objectif du projet est de vous permettre de mettre en œuvre, dans le cadre d'une réalisation concrète, les notions de POO et Java abordées au cours du module. Il n'est pas nécessaire de vouloir être exhaustif dans le traitement des nombreuses situations présentes dans les fichiers manipulés. De même, il est possible d'utiliser des bibliothèques externes correspondant à vos besoins.

Première partie

Pour démarrer le projet, il est recommandé de traiter dans un premier temps le cas d'un unique fichier en s'appuyant d'abord uniquement sur l'extension, puis sur le type MIME et enfin sur une signature à définir. La vérification de l'intégrité de certains fichiers ne devrait être abordée que dans un second temps.

Seconde partie

Dans un deuxième temps, vous pourrez généraliser votre solution à un ensemble de fichiers placés dans un répertoire et ses sous-répertoires.

Quelques indications pour le rapport

Ce document rédigé à l'aide d'un traitement de texte (MS Word ou libre-office par exemple) doit permettre de fournir un compte-rendu complet et un bilan de votre travail et de son aboutissement. Vous y placerez en particulier le diagramme de classes UML de votre application. Les informations de planning et de répartition des tâches sont attendues. Des explications sur les aspects particuliers de votre solution. Vous pouvez ajouter quelques captures d'écran représentatives mais n'en abusez pas. Vous détaillerez le niveau d'aboutissement de votre réalisation avec également un regard critique sur les points forts et les points faibles que vous aurez identifiés.

Ressources

- https://www.iana.org/assignments/media-types/media-types.xhtml
- https://developer.mozilla.org/fr/docs/Web/HTTP/Basics_of_HTTP/MIME_Types
- https://developer.mozilla.org/fr/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Complete_list_of_MIME_types
- https://tools.ietf.org/html/rfc2045