Projet de Java Licence - JButcher

Le but de ce projet est de réaliser un filtreur de mail. Le logiciel devra tourner au moins sur les environnements Unix, MacOS X et Windows indifféremment.
forax@univ-mlv.fr, pierre.peterlongo@univ-mlv.fr, benoit.olivieri@univ-mlv.fr

Description du projet

Le programme `JButcher` est un filtreur de mail. L'architecture de cet outils devra être assez générique pour permettre à n'importe qui de "customiser" le filtreur pour effectuer différents traitements sur des mails.
Il faudra que vous gardier à l'esprit que `JButcher` devrait permettre de :

- rediriger ou relayer les mail un peu comme la fait le logiciel `procmail`.
- de trouver les spams et de l'indiquer comme tel pour un lecteur de mail comme le ferait un logiciel anti-spam comme par exemple `SpamAssassin`.
- laisser la possibilité à n'importe qui d'ajouter de nouvelles fonctionnalités sous forme de filtre ou d'action.

Modèle et Concept

`JButcher` manipule trois concepts, les règles, les filtres et les actions.
Une règle définie :

- une source de mails
- une filtre à appliquer pour chaque mail
- un seuil (une valeur à virgule flottante)
- un ensemble d'actions

Les mails peuvent être obtenu suivant les protocols POP et IMAP à partir de la source de mails. L'ensemble des actions est effectuées si le filtre (dit filtre de départ ou `startFilter`) est positif, c'est à dire si le filtre dépasse une valeur seuil fixée dans la règle pour chaque mail de la source de mails.
Un filtre de mail analyse un mail en provenance de la source de mails et renvoie une valeur réelle.
Il est possible, de plus, de spécifier des relation entre des filtres et de construire avec cela un arbre de filtre. Par exemple, le filtre `ORFilter` sera en relation avec d'autres filtres pour effectuer un ou logique entre les resultats de ceux-ci.
L'ensemble des actions à effectuer si pour un mail le résultat du filtre `startFilter` est supérieur au seuil. `JButcher` déclenche alors l'ensemble des actions qui s'effectuent sur ce mail.
`JButcher` est initialisé par l'intermédiaire d'un fichier de configuration (`butcher.conf`) définissant un ensemble des règles avec leurs filtres et leurs actions.
La section suivante détaille deux exemples de configurations.

Exemples de fichier de configuration

```
<butcher-conf>
     <rules>
```

```
          <rule src="imap://but.cher:butcher@imap.laposte.net/INBOX"
               onlyRecent="false"
               threshold="20.0"
               start-filter="sum">
          <filters>
           <filter name="sum"
                      class="fr.umlv.jbutcher.example.SumFilter"
                      list="test,test2"/>
           <filter name="test"
                  class="fr.umlv.jbutcher.example.TestFilter"
                  threshold="5.0"/>
           <filter name="test2"
                  class="fr.umlv.jbutcher.example.TestFilter"
                  threshold="15.0"/>
            </filters>
           <actions>
            <action name="print"
class="fr.umlv.jbutcher.example.PrintAction"/>
            <action name="print2"
class="fr.umlv.jbutcher.example.PrintAction"/>
           </actions>
          </rule>
         </rules>
   </butcher-conf>
```

Ici, on récupère par IMAP tous le mails et ceux-ci sont affichés deux fois sur la sortie standard.

```
   <butcher-conf>
          <rules>
           <rule src="pop3://but.cher:butcher@pop.laposte.net/INBOX"
                 onlyRecent="true"
                 threshold="10.0"
                 start-filter="anti-spam">
            <filters>
             <filter name="anti-spam"
                     class="fr.umlv.jbutcher.example.AntiSpamFilter"
                     file="bad-words.txt"/>
            </filters>
            <actions>
             <action name="forward"
class="fr.umlv.jbutcher.example.ForwardAction"
                     email-address="forax@univ-mlv.fr"
                     smtp-host="smtp.laposte.net"
                     smtp-port="25"/>
            </actions>
           </rule>
          </rules>
   </butcher-conf>
```

Ici, on récupère par POP les mails récents et ceux-ci sont transférés (forward) à l'adresse mail "forax@univ-mlv.fr" par le serveur SMTP "smtp.laposte.net" si ceux-ci sont considérés comme du spam.
Exemple de fichier "bad-words.txt"

```
   viagra(5.2)
   xanax(5.0)
   remi(-2.2)
```

Les valeurs mis entre parenthèse indique les valeurs attribuées à chaque mot.

Liste des filtres et actions demandées

Liste des filtres à implanter :

1. `TestFilter` Paramètre : `threshold` : un seuil.
   Renvoie toujours la valeur du seuil passée en paramètre quelque soit le mail.
2. `OrFilter`
   Paramètre : `threshold` : un seuil.
   Paramètre : `list` : une liste de nom de filtre.
   Filtre dont la valeur est égale au seuil si un des filtres de la liste a une valeur supérieure au seuil. Sinon la valeur du filtre est zéro.
3. `AndFilter`
   Paramètre : `threshold` : un seuil.
   Paramètre : `list` : une liste de nom de filtre.
   Filtre dont la valeur est égale au seuil si tous des filtres de la liste ont une valeur supérieure au seuil. Sinon la valeur du filtre est zéro.
4. `SumFilter`
   Paramètre : `list` : une liste de nom de filtre.
   Filtre dont la valeur est égale à la somme des filtres de la liste.
5. `SpamFilter`
   Paramètre : `file` : fichier contenant un mot et sa valeur par ligne.
   Filtre dont la valeur est égale à la somme des valeurs des mots reconnus dans le fichier.

Liste des actions à implanter :

1. `PrintAction`
   Affiche le contenu du mail filtré.
2. `ForwardAction`
   Paramètre : `email-address` adresse mail
   Paramètre : `smtp-host` machine serveur SMTP
   Paramètre : `smtp-port` port SMTP
   Renvoie un mail filtré vers une adresse mail.
3. `MarkAction`
   Paramètre : `flag` nom du drapeau
   Paramètre : `flag-value` valeur du drapeau
   Ajoute à un mail filtré un drapeau (flag) dans son entête, pour indiquer que celui-ci est un spam.

Calendrier

Ce projet est à faire par binôme (cela veut dire deux personnes pas trois ni une). Le rendu du projet est découpé en deux parties :

1. Pré-rapport indiquant l'architecture objet et tests d'utililisation.
   Date de rendu : le 4 avril, avant minuit.
2. Rendu du projet en lui-même.
   Date de rendu : le 21 mai, avant minuit.

Le rendu s'effectuera par mail sous forme d'une pièce jointe au format ZIP contenant l'ensemble des programmes et documents. Le mail devra être envoyé avant minuit aux trois adresses données au début de ce document.

Détail du premier rendu

Le premier rendu est un rapport au format PDF ainsi que des tests sous forme d'un ou plusieurs Jar exécutable. Ce rapport est composé de deux parties, une partie générale sur l'architecture objet du projet et une seconde sur des tests.

L'architecture générale du projet (au moins 7 pages) :

le découpage en différentes classes,
le rôle de chaque classe
l'ensemble des méthodes publiques de chaque classe
les problèmes soulevés par le fichier de configuration.

Un ensemble de tests sur les sujets suivants :

- chargement dynamique d'une classe à partir de son nom, et instantiation d'un objet de cette classe respectant une interface.
- installation de la librairie JavaMail
- envoie d'un mail par SMTP.
- affichage de l'ensemble des mots contenu dans un mail. (i.e entête du mail ainsi que contenu des différentes pièces jointes).
- utilisation d'expression régulière pour reconnaitre des mots dans un texte.
- réception de mails par IMAP.
- réception de mails par POP.
- parsing du fichier de configuration en XML en utilisant JAXP (dispo dans le J2SK1.4).
- création d'un script ANT de compilation.
- création d'un Jar exécutable.

Chaque test doit être accompagné d'un texte expliquant les concepts manipulés, les instructions pour exécuter le test et le code pertinant du test (le plus clair possible).

Essayer de faire un document cohérent et lisible, SVP.

Détail du second rendu

Le second rendu sera vu plus tard.

Références

- JavaMail (`http://java.sun.com/products/javamail/`)
- Java API for XML Processing (JAXP) (`http://java.sun.com/xml/jaxp/index.jsp`)
- Expression régulière (`http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/package-summary.html`)
- Jar exécutable (`http://java.sun.com/j2se/1.4.2/docs/guide/jar/jar.html`)
- Ant (`http://ant.apache.org/`)