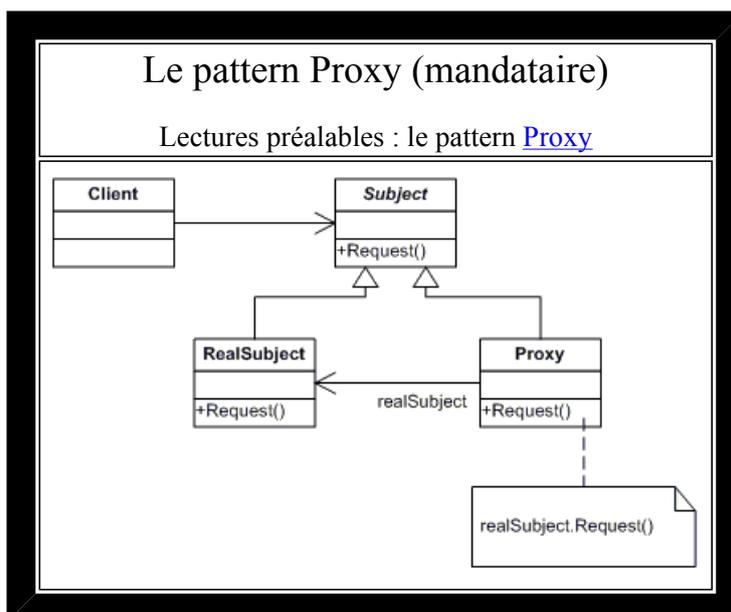


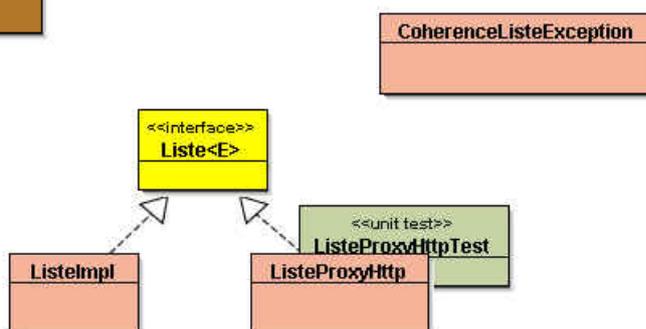
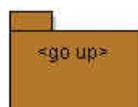
# tp6



- Visualisez dans un navigateur le sujet en ouvrant `index.html` du répertoire qui a été créé à l'ouverture de `tp6.jar` par BlueJ; vous aurez ainsi accès aux différents liens qui sont proposés pour vous aider, et aux applettes.
- Soumettez chaque question à l'outil d'évaluation JNews.



Un mandataire "HTTP"



## Question :

Selon le patron "Procuration/Proxy", installez un mandataire (classe `ListeProxyHttp`) qui pour chaque opération sur la liste locale (classe `ListeImpl`) engendre cette même opération sur la liste distante associée.

Si une incohérence entre la liste locale et la liste distante est détectée, une exception sera levée.

- Le service "web liste distante" est fourni.
- La liste distante est **persistante** et nécessite un **identifiant** de la liste,
- les opérations (*ajouter, supprimer, ...*) ont la même sémantique.

Le mandataire demandé : la classe *ListeProxyHttp*,

- **utilise** une liste locale (instance de la classe *ListeImpl*),
- **effectue** les requêtes HTTP,
- **vérifie** la cohérence entre la liste locale et la liste distante pour les opérations : ajouter, retirer, vider
- **propose** la méthode restaurer non implémentée dans la classe *ListeImpl*,
- **lève** une exception *CoherenceListeException* à chaque incohérence détectée.
  - note : Les **erreurs** de réseau engendrent de fait une levée de cette exception.

**Pour cette question vous disposez :**

- D'une interface *Liste* (*Liste<E>*) → lire la [Javadoc](#) pour prendre connaissance des spécifications de chaque méthode,
- et d'une implémentation concrète (*ListeImpl*), dont les éléments de la liste sont des "String".
- D'une gestion de listes accessible depuis le Web à cette l'URL [http://pc5357c.esiee.fr:8788/tests/tp\\_proxy.html](http://pc5357c.esiee.fr:8788/tests/tp_proxy.html)

Exemple 1 :

```
Liste<String> liste = new ListeProxyHttp("liste_777");
liste.ajouter("essai");
```

déclenche la méthode *ajouter* sur la liste locale et effectue une requête avec l'URL ci dessous  
[http://pc5357c.esiee.fr:8788/tests/tp\\_proxy.html?nom=liste\\_777&commande=ajouter&elt=essai](http://pc5357c.esiee.fr:8788/tests/tp_proxy.html?nom=liste_777&commande=ajouter&elt=essai)

dans laquelle le paramètre *nom* représente un identifiant de cette liste, le paramètre *commande* reflète l'action choisie et *elt* l'élément

Exemple 2 :

*Une ébauche possible de la méthode ajouter*

```
public boolean ajouter(String elt){

    // requête au protocole HTTP, liste distante
    Connexion connect = new Connexion(this.url,"nom=" + this.nom
                                     + "&commande=ajouter&elt="+ elt);
    Boolean resultatLocal = liste.ajouter(elt);

    String resultatDistant = // le résultat de la connexion
    //si le resultatLocal est différent du resultatDistant
    alors levée de CoherenceListeException("ajouter");
    return resultatLocal;
}
```

**AIDE :** On peut comparer les 2 résultats sous forme de booléens ou sous forme de String.

**Attention !** `==` compare les références alors que `equals()` compare bien le contenu des objets.

Récapitulatif des associations méthodes Java / requêtes HTTP :

- Le site gestionnaire est en [http://pc5357c.esiee.fr:8788/tests/tp\\_proxy.html](http://pc5357c.esiee.fr:8788/tests/tp_proxy.html)

méthodes "java"	<a href="http://pc5357c.esiee.fr:8788/tests/tp_proxy.html">http://pc5357c.esiee.fr:8788/tests/tp_proxy.html</a>	Formulaires HTML
public boolean ajouter(E elt);	<a href="http://pc5357c.esiee.fr:8788/tests/tp_proxy.html?nom=liste_777&amp;commande=ajouter&amp;elt=essai">nom=liste_777&amp;commande=ajouter&amp;elt=essai</a>	nom : <input type="text" value="liste_777"/> elt : <input type="text" value="essai"/> <input type="button" value="ajouter"/>

public boolean retirer(E elt);	<a href="#">nom=liste_777&amp;commande=retirer&amp;elt=essai</a>	nom : <input type="text" value="liste_777"/> elt : <input type="text" value="essai"/> <input type="button" value="retirer"/>
public boolean vider();	<a href="#">nom=liste_777&amp;commande=vider</a>	nom : <input type="text" value="liste_777"/> <input type="button" value="vider"/>
public boolean estPresent(E elt);	<a href="#">nom=liste_777&amp;commande=estPresent&amp;elt=essai</a>	nom : <input type="text" value="liste_777"/> elt : <input type="text" value="essai"/> <input type="button" value="estPresent"/>
public int taille ();	<a href="#">nom=liste_777&amp;commande=taille</a>	nom : <input type="text" value="liste_777"/> <input type="button" value="taille"/>
public Iterator<E> iterator();	non implémentée	
public String toString();	<a href="#">nom=liste_777&amp;commande=toString</a>	nom : <input type="text" value="liste_777"/> <input type="button" value="toString"/>

notez que

1. Les données manipulées sont des "String", limitées à cette expression régulière : [A-Za-z0-9]+
2. Toute requête engendre la création de la liste distante, si celle-ci n'existait pas encore sur le serveur
3. L'identifiant associé à votre liste doit être unique, choisissez plutôt un nom comme : "liste\_XXXXXX", où XXXXXX est un nombre à 6 chiffres ( les listes sont placées dans la même table de la base de données côté serveur).

La présence d'un identifiant dans cette table peut être vérifiée avec ce test

identifiant :

Soumettre cette question à JNews.



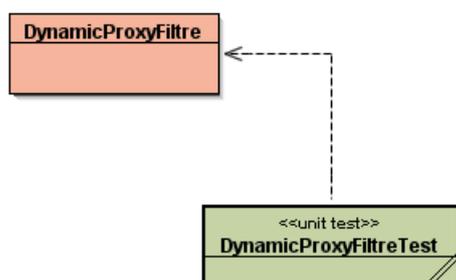
question2

## Un nouveau mandataire

Toujours selon, l'architecture du patron "Procurator/Proxy", proposer un nouveau mandataire qui filtre le nom des méthodes. Les noms de ces méthodes sont définis à l'exécution, ce mandataire utilise les classes "[DynamicProxy](#)".

Les noms des méthodes autorisés sont transmis lors de la création du mandataire, celui-ci lève une exception de type **IllegalAccessException** si une tentative d'accès est effectuée sur une méthode inhibée.

[Lire attentivement la classe de tests](#)



Exemple d'utilisation d'une liste avec un accès restreint : seules les méthodes **"ajouter,estPresent,taille,toString"** sont permises

```

public void testAccesRestreint1() throws Exception{

    String methodesPermises[] = new String[]
{"ajouter", "estPresent", "taille", "toString"};

    Liste<String> liste = DynamicProxyFiltre.getProxy(new ListeImpl
("listeA"), methodesPermises);

    try{

        liste.ajouter("test");

        liste.ajouter("test2");

    }catch(Exception e){

        fail("une exception pour une méthode autorisée ???");

    }

    assertTrue(liste.taille()==2);

    try{

        liste.vider();

        fail("une exception est attendue pour une méthode inhibée !");

    }catch(Throwable e){

        assertTrue( "IllegalAccessException est attendue ?",
e.getCause() instanceof IllegalAccessException);

    }

}
}

```

Document conseillé pour la réalisation :

- <http://java.sun.com/javase/6/docs/technotes/guides/reflection/index.html>

Soumettre cette question à JNews.