Reflection, proxy et modification de byte-code

Exercice 1 - Launcher

Ecrire une classe `Launcher` qui prend en paramètre un nom de classe et des paramètres et qui exécute la méthode `main` de cette classe avec les paramétres.

Exercice 2 - Couleurs

Ecrire un programme qui affiche l'ensemble des couleurs (`java.awt.Color`) prédéfinie en Java.
L'affichage d'une couleur correspond à l'affichage de son nom suivi de ses valeurs RGB (red/green/blue).
Essayer d'écrire le programme le plus générique possible.

Exercice 3 - Proxy et Trace d'appel

Soit l'interface `SimiliCollection` suivante :

```
interface SimiliCollection {
  void add(Object value);
  void remove(Object value);
}
```

A l'aide de la classe `java.util.ArrayList`, ecrire une classe `SimiliList` implantant l'interface `SimiliCollection`
Ecrire ensuite une méthode statique prenant en paramètre une `SimiliCollection` et retournant une `SimiliCollection` qui lors d'un appel à add() ou remove() affiche l'entrée dans la méthode et la sortie de la méthode.
Modifier le code de la méthode statique pour qu'il marche de façon générique quelque soit l'interface utilisée (regarder la classe `java.lang.reflect.Proxy`).

Exercice 4 - Modification de byte-code

Downloader le fichier zip qui contient la librairie ASM qui permet de modifier le byte-code lors du chargement de celui-ci. Exemple d'utilisation d'ASM (demo-asm.zip) (`demo-asm.zip`)
PS: vous pouvez l'importer comme un nouveau projet eclipse.
L'APÏ d'ASM est disponible à cette adresse : http://asm.objectweb.org/current/doc/javadoc/user/index.html
(`http://asm.objectweb.org/current/doc/javadoc/user/index.html`)
Executer et commenter l'exemple (cf fichier `ModifyClassLoader`).
Modifier l'exemple pour que celui-ci afficher une trace que le fil d'exécution rentre ou sort des méthodes de `HelloWorld`.