**Тексты**

* [Do realy get classloaders](https://vcfvct.files.wordpress.com/2015/06/do-you-really-get-classloaders.pdf)
* [Martin Lipper. Classloading and Type Visibility in OSGi](http://www.martinlippert.org/events/WJAX2008-ClassloadingTypeVisibilityOSGi.pdf)

**YouTub**

* [Антон Архипов — Загрузчики классов в Java: коллекция граблей](https://www.youtube.com/watch?v=RHVzgXff9jg)
* [Java Classloaders by Simon Maple](https://www.youtube.com/watch?v=W2pvuHy8WtI)

**Для общего развития**

* [Антон Архипов — Байткод для любознательных (StudentDay)](https://www.youtube.com/watch?v=1XV88X3ugiM)
* [Антон Архипов — От HotSwap до Java-агентов: горячая замена классов](https://www.youtube.com/watch?v=wJwCjXcCH90)
* [Андрей Паньгин — Java Runtime: повседневные обязанности виртуальной машины Java](https://www.youtube.com/watch?v=Svc0WtfV63k&t=818s)
* [Никита Липский, Владимир Иванов — JVM: краткий курс общей анатомии](https://www.youtube.com/watch?v=-fcj6EL9rc4)
* <https://javarevisited.blogspot.com/2012/12/how-classloader-works-in-java.html>
* <https://javarevisited.blogspot.com/2012/07/when-class-loading-initialization-java-example.html#axzz4uMIMWleJ>

1)

Ваша задача написать загрузчик плагинов в вашу систему. Допустим вы пишите свой браузер и хотите чтобы люди имели имели возможность писать плагины для него. Соответственно, разные разработчики могут назвать свои классы одинаковым именем, ваш загрузчик должен корректно это обрабатывать.

Задача реализовать следующий класс.

**package** ru.sbt;  
  
**public class** PluginManager {  
 **private final** String **pluginRootDirectory**;  
  
 **public** PluginManager(String pluginRootDirectory) {  
 **this**.**pluginRootDirectory** = pluginRootDirectory;  
 }  
  
 **public** Plugin load(String pluginName, String pluginClassName) {  
 *//****todo*** }  
}

Plugin - это базовый интерфейс для всех плагинов  
**public interface** Plugin {  
 *//methods doesn't matter* **void** doUsefull();  
}

}

PluginManager ищет скомпилированные классы плагина в папке pluginRootDirectory\pluginName\

Усложненная версия задания. Система должна вести себя корректно если в плагине, есть скомпилированные классы с именем, которые есть в вашем браузере(не в плагинах), должны использоваться классы плагина, а не вашего браузера. Для этого придется поменять модель делегирования класслоадера в методе loadClass

2)

Написать EncryptedClassloader

Данный класслоадер умеет загружать классы из файлов дешифрую их. Ваша задача переопределить метод findClass(). В нем лоадер считывает зашифрованный массив байт, дешифрует его и превращает в класс (с помощью метода defineClass).

На вход класслодер принимает ключ шифрования, рутовую папку, в которой будет искать классы, родительский класслодер. Логика шифрования/дешифрования с использованием ключа может быть любой на ваш вкус (например, каждый считаный байт класса увеличить на определение число).

**public class** EncryptedClassLoader **extends** ClassLoader {  
 **private final** String **key**;  
 **private final** File **dir**;  
  
 **public** EncryptedClassLoader(String key, File dir, ClassLoader parent) {  
 **super**(parent);  
 **this**.**key** = key;  
 **this**.**dir** = dir;  
 }

}