**ACTIVIDAD 1**

**Enunciado**

Realiza un programa en Java que muestre por pantalla los diferentes algoritmos disponibles para procesar resúmenes de mensaje del proveedor SUN en tu máquina virtual Java. Además, se mostrará cierta información del proveedor (nombre y versión).

El nombre de la clase principal será Actividad1.

El resultado de la ejecución del programa mostrará la siguiente información:

\*\* Proveedor SUN, versión xx \*\*

Nombre del algoritmo: "MD2"

Nombre del algoritmo: "MD5"

Nombre del algoritmo: "SHA"

Nombre del algoritmo: "SHA-224"

Nombre del algoritmo: "SHA-256"

Nombre del algoritmo: "SHA-384"

Nombre del algoritmo: "SHA-512"

….

La solución de este programa puede requerir, entre otras clases del paquete java.security, el uso de la clase *Provider.Service*, la cual dispone de los recursos necesarios para obtener el nombre de los algoritmos del proveedor de servicios de encriptación.

**Solución 1**

**// Actividad1.java**

**import** java.security.MessageDigest;

**import** java.security.Provider;

**import** java.security.Provider.Service;

**import** java.security.Security;

**import** java.util.Set;

**public** **class** Actividad1{

**public** **static** **void** main(String[] args) {

Provider p = Security.*getProvider*("SUN"); //instanciar el provider de SUN unicamente

//Mostrar el nombre del proveedor y de la version

System.***out***.println("\*\* Proveedor " + p.getName() + ", versión " + p.getVersion() + " \*\* ");

Set<Service> s = p.getServices(); //Coger todos los servicios de Sun y guardarlos en un set

**for** (Service service : s) { //sacar cada uno de los service

**if**(service.getType().equals(MessageDigest.**class**.getSimpleName())) { //mirar si son del type MessageDigest

System.***out***.println("\tNombre del algoritmo: \"" + service.getAlgorithm() + "\""); //Mostrar los que sean del tipo correcto

}

}

}

}

**Solución 2**

**// Actividad1.java**

**import** java.security.MessageDigest;

**import** java.security.Provider;

**import** java.security.Provider.Service;

**import** java.security.Security;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Set;

**public** **class** Actividad1 {

// Función main

**public** **static** **void** main(String[] args) {

Provider[] providers = Security.*getProviders*();

**for** (Provider provider : providers) {

*showHashAlgorithms*(provider, MessageDigest.**class**);

}

}

// showHashAlgorithms

**private** **static** **final** **void** showHashAlgorithms(Provider prov, Class<?> typeClass) {

String type = typeClass.getSimpleName();

List<Service> algos = **new** ArrayList<>();

Set<Service> services = prov.getServices();

**for** (Service service : services) {

**if** (service.getType().equalsIgnoreCase(type)) {

algos.add(service);

}

}

// Mostrar la información por pantalla

**if** (!algos.isEmpty()) {

// Mostrar información del proveedor

System.***out***.printf(" \*\* Proveedor %s, versión %.2f \*\* %n", prov.getName(), prov.getVersion());

// Bucle para mostrar información de los algoritmos de encriptación del proveedor

**for** (Service service : algos) {

String algo = service.getAlgorithm();

System.***out***.printf("\tNombre del algoritmo: \"%s\"%n", algo);

}

}

}

}

**Solución 3**

**// Actividad1.java**

**import** java.security.MessageDigest;

**import** java.security.Provider;

**import** java.security.Provider.Service;

**import** java.security.Security;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Set;

**public** **class** Actividad1 {

//Main

**public** **static** **void** main(String[] args) {

//Obtener en un array los providers

Provider[] providers=Security.*getProviders*();

//Recorrer el array de Providers

**for**(Provider provider:providers){

*showHashAlgoritms*(provider, MessageDigest.**class**);

}

}

//Método showHashAlgoritms

**private** **static** **final** **void** showHashAlgoritms(Provider prov,Class<?> typeClass){

//Obtener el tipo de la clase

String type=typeClass.getSimpleName();

//Una lista de servicios

List<Service> algos=**new** ArrayList<>();

//Obtener todos los servicios en una estructura de tipo set

Set<Service> services=prov.getServices();

//Recorrer los servicios y filtrar los de la clase messagedigest

**for**(Service service:services){

**if**(service.getType().equalsIgnoreCase(type)){

//Añadir el servicio a la lista

algos.add(service);

}

}

//Mostrar la información por pantalla solo si el array contiene algo

**if**(!algos.isEmpty()){

//Mostrar la información del proveedor

System.***out***.printf("\*\* Proveedor "+prov.getName()+", versión %.2f \*\* %n",prov.getVersion());

//Bucle para mostrar la información de los algoritmos de encriptación del proveedor

**for**(Service service:algos){

//Obtener los algoritmos

String algo=service.getAlgorithm();

//Imprimir el algoritmo

System.***out***.printf("\t Nombre del algoritmo: \""+algo+"\"\n");

}

}

}

}

**Solución 4**

**// Actividad1.java**

**import** java.security.MessageDigest;

**import** java.security.Provider;

**import** java.security.Provider.Service;

**import** java.security.Security;

**import** java.util.Set;

**public** **class** Actividad1 {

**public** **static** **void** main(String[] args) {

//obtenemos y visualizamos los datos del proveedor sun

Provider proveedor =Security.*getProvider*("SUN");

System.***out***.printf("\*\* Proveedor %s, versión %.2f \*\* %n",proveedor.getName(),proveedor.getVersion());

//obtenemos una lista de los servicios del proveedor 'SUN'

Set<Service> serv = proveedor.getServices();

**for**(Service s : serv) {//Recorremos la lista de servicios en busca de los algoritmos hash

**if**(s.getType().equals(MessageDigest.**class**.getSimpleName())) {//una vez filtrado mostramos los datos

System.***out***.println("\tNombre del algoritmo: \""+s.getAlgorithm()+"\"");

}

}

}

}

**RESOLUCIÓN DE PROBLEMAS:**

**REQUISITOS**

1. Entregar cada actividad en un archivo comprimido que contenga los ficheros \*.java empleados.
2. Los archivos comprimidos seguirán la siguiente nomenclatura:

*UD5\_Actividadn\_apellido*

donde:

*n*: valdrá 1, representando el número de la actividad

*apellido*: será el primer apellido del alumno

**CRITERIOS DE CORRECCIÓN**

1. Cumplimiento de los requisitos (15%).
2. Hacer lo que se indica en el enunciado (55%).
3. Claridad del código Java (15%).
4. Documentación del código Java (15%).
5. Explicación detallada de cómo se han resuelto problemas que hayan podido surgir en la realización de la actividad (hasta un 15% adicional sobre la nota obtenida). Dicha explicación se incluirá al principio de los archivos de código fuente Java mediante líneas de comentarios. La primera línea de comentarios sólo incluirá lo siguiente: // Resolución de problemas.
6. Variables no usadas (-15%).
7. Librerías no usadas (-15%).