### MessageDigest类（java.security.MessageDigest）

MessageDigest类为应用程序提供信息摘要算法的功能，如MD5或SHA算法。信息摘要是安全的单向哈希函数，它接收任意大小的数据，输出固定长度的哈希值。简单点说就是用于生成散列码。关于信息摘要和散列码请参照《[数字证书简介](http://hubingforever.blog.163.com/blog/static/17104057920118981219705/)》。

MessageDigest

* 通过getInstance()系列静态函数来进行实例化和初始化。
* 该对象通过使用update方法处理数据。任何时候都可以调用reset方法重置摘要。
* 一旦所有需要更新的数据都已经被更新了，应该调用digest方法之一完成哈希计算并返回结果。

注：对于给定数量的更新数据，digest方法只能被调用一次。digest被调用后

，MessageDigest对象呗重新设置成其初始状态。

MessageDigest的实现可随意选择是否实现Cloneable接口。客户端应用程序可以通过尝试复制和捕获CloneNotSupportedException测试可复制性：

注意1：

即时给定MessageDigest的实现是不可复制的，则仍然能够通过getInstance方法实例化几个实例计算来同时进行摘要信息的计算。

注意2：

由于历史原因，此类是抽象的，是从 MessageDigestSpi 扩展的。应用程序开发人员只应该注意在此 MessageDigest 类中定义的方法；超类中的所有方法是供希望提供自己的信息摘要算法实现的加密服务提供者使用的。

注意3：

MessageDigest并不是单实例的。如下代码所示：

try {  
 MessageDigest md1 = MessageDigest.*getInstance*("MD5");  
 MessageDigest md2 = MessageDigest.*getInstance*("MD5");  
 MessageDigest md3 = MessageDigest.*getInstance*("MD5");  
  
 System.*out*.println("md1==md2?:" + (md1 == md2));  
 System.*out*.println("md2==md3?:" + (md2 == md3));  
} catch (NoSuchAlgorithmException e) {  
 e.printStackTrace();  
}

### 实践

#### 1、创建MessageDigest对象

实例化MessageDigest时不区分算法的大小写：

MessageDigest.getInstance("SHA");

MessageDigest.getInstance("sha");

MessageDigest.getInstance("sHa");

调用程序可选择指定提供者名称，以保证所要求的算法是由已命名提供者实现的：

public static MessageDigest getInstance(String algorithm, String provider);

调用 getInstance 将返回已初始化过的MessageDigest对象。因此，它不需要进一步的初始化。

#### 2、向MessageDigest传送要计算的数据

计算数据的摘要的第二步是向已初始化的MessageDigest对象提供传送要计算的数据。这将通过一次或多次调用以下某个update（更新）方法来完成。

public void update(byte input);

public void update(byte[] input);

public void update(byte[] input, int offset, int len);

#### 3、计算摘要

通过调用update方法向MessageDigest对象提传送要计算的数据后，你就可以调用以下某个digest（摘要）方法来计算摘要（即生成散列码）：

public byte[] digest();

public byte[] digest(byte[] input);

public int digest(byte[] buf, int offset, int len);

前两个方法返回计算出的摘要。后一个方法把计算出的摘要储存在所提供的 buf 缓冲区中，起点是 offset。len 是 buf 中分配给该摘要的字节数。该方法返回实际存储在 buf 中的字节数。

对第二个接受输入字节数组变量的digest方法的调用等价于用指定的输入调用：

public void update(byte[] input)

接着调用不带参数的digest方法。

### 编程思路

java.security包中的MessageDigest类提供了计算消息摘要（即生成散列码）的方法，首先生成对象，执行其update()方法可以将原始数据传递给该对象，然后执行其digest()方法即可得到消息摘要。具体步骤如下：

public class MessageDigestAlogrithm {

*/\*\*  
 \* java.security包中的MessageDigest类提供了计算消息摘要（即生成散列码）的方法  
 \*/* public static void main(String[] args) throws Exception {  
  
 String x = "abc";  
 byte[] bytes = *messageDigest*(x);  
 String hexString = *convertToHexString*(bytes);  
 System.*out*.println(hexString);  
 }  
  
 public static byte[] messageDigest(String x) throws Exception {  
  
 // 生成MessageDigest对象  
 MessageDigest md5 = MessageDigest.*getInstance*("MD5");  
 // 传入需要计算的字符串  
 md5.update(x.getBytes("UTF8"));  
 // 计算信息摘要  
 byte[] digest = md5.digest();  
 return digest;  
 }  
  
 */\*\*  
 \* 处理计算结果 转换成十六进制字符串  
 \*  
 \** ***@param*** *data  
 \** ***@return*** *\*/* public static String convertToHexString(byte data[]) {  
 StringBuffer strBuffer = new StringBuffer();  
 for (int i = 0; i < data.length; i++) {  
 strBuffer.append(Integer.*toHexString*(0xff & data[i]));  
 }  
 return strBuffer.toString();  
 }  
}