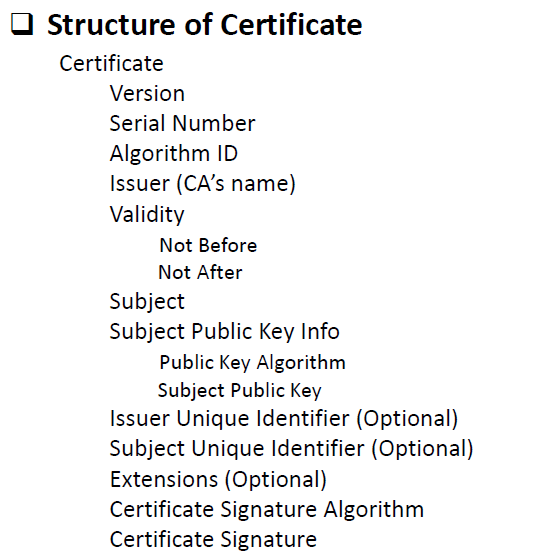
**X.509数字证书的读取**

**X.509证书结构描述：**

X.509 是密码学里公钥证书的格式标准。 X.509证书己应用在包括TLS/SSL在内的众多 Intenet协议里.同时它也用在很多非在线应用场景里，比如电子签名服务。X.509证书里含有公钥、身份信息（比如网络主机名，组织的名称或个体名称等）和签名信息（可以是证书签发机构CA的签名，也可以是自签名）。对于一份经由可信的证书签发机构签名或者可以通过其它方式验证的证书，证书的拥有者就可以用证书及相应的私钥来创建安全的通信，对文档进行数字签名。

Ppt上给出了X.509的组成结果（其实就是维基百科里的结构定义）：



结构解释：

* 证书
  + 版本号：标识证书的版本（版本1、版本2、或是版本3）
  + 序列号：标识证书的唯一整数，由证书颁发者分配的本证书的唯一标识符
  + 签名算法：用于说明证书所用的数字签名算法
  + 颁发者：证书颁发者的可识别名
  + 证书有效期 ：证书有效的时间段
    - 此日期前无效
    - 此日期后无效
  + 主体：证书拥有者的可识别名，这个字段必须是非空的，除非你在证书扩展中有别名
  + 主体公钥信息：主体的密钥以及算法标识符
    - 公钥算法
    - 主体公钥
  + 颁发者唯一身份信息（可选项）：证书颁发者的唯一标识符，仅在版本2和版本3中有要求，属于可选项
  + 主体唯一身份信息（可选项）：证书拥有者的唯一标识符，仅在版本2和版本3中有要求，属于可选项。
  + 扩展信息（可选项）
    - ...
* 证书签名算法
* 数字签名

那么如何获取相应的证书？

打开Chrome浏览器，点击网址左边的感叹号，可以看到证书（这里采用csdn的证书）：



进入证书的详细信息，点击复制到文件：



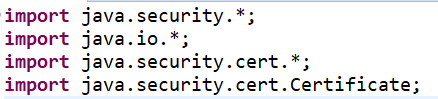
选择相应的格式，点击下一步，选择保存路径就可以完成导出：

这里我保存下来的路径如下：C:\Users\lcccy\Desktop\test.cer

我们可以看到，通过这种方式保存下来的X.509证书的扩展名是.cer，实际上证书还有其他的扩展名类型，比如.crt，.der，.p7b，等等。cer的格式有DER二进制编码和Base64编码。

**数据结构：**

相比于C语言/C++，Java语言中有现成的对X.509证书进行操作的很多函数，要使用这些函数，只需要导入相应的包即可：



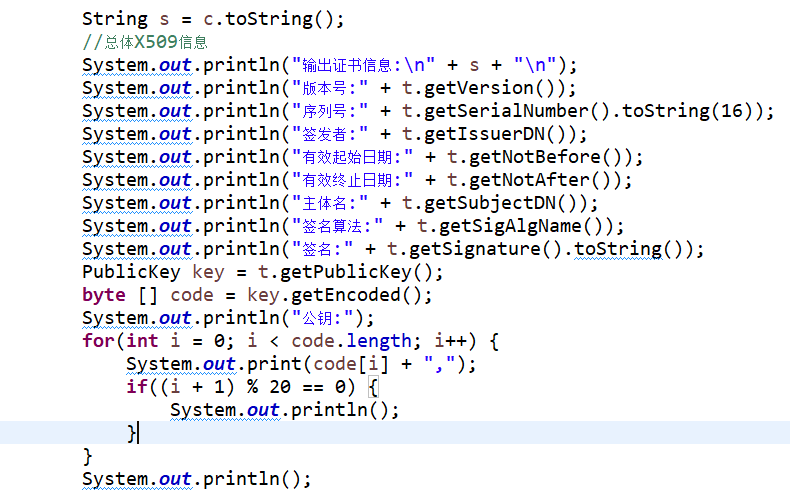
先用文件流读入导出的证书，然后用CertificateFactory实例化X.509类型，最后利用X509Certificate强转类型获取导出的X.509证书的实例：



另外为了保证IO操作的准确性和防止证书过期，应当在main函数处添加异常处理：



然后利用X509Certificate类中的成员函数输出证书里的信息，包括版本号、序列号等等。运用到的函数方法的具体作用可以在这个网站上查找到：<http://jszx-jxpt.cuit.edu.cn/JavaAPI/java/security/cert/X509Certificate.html>

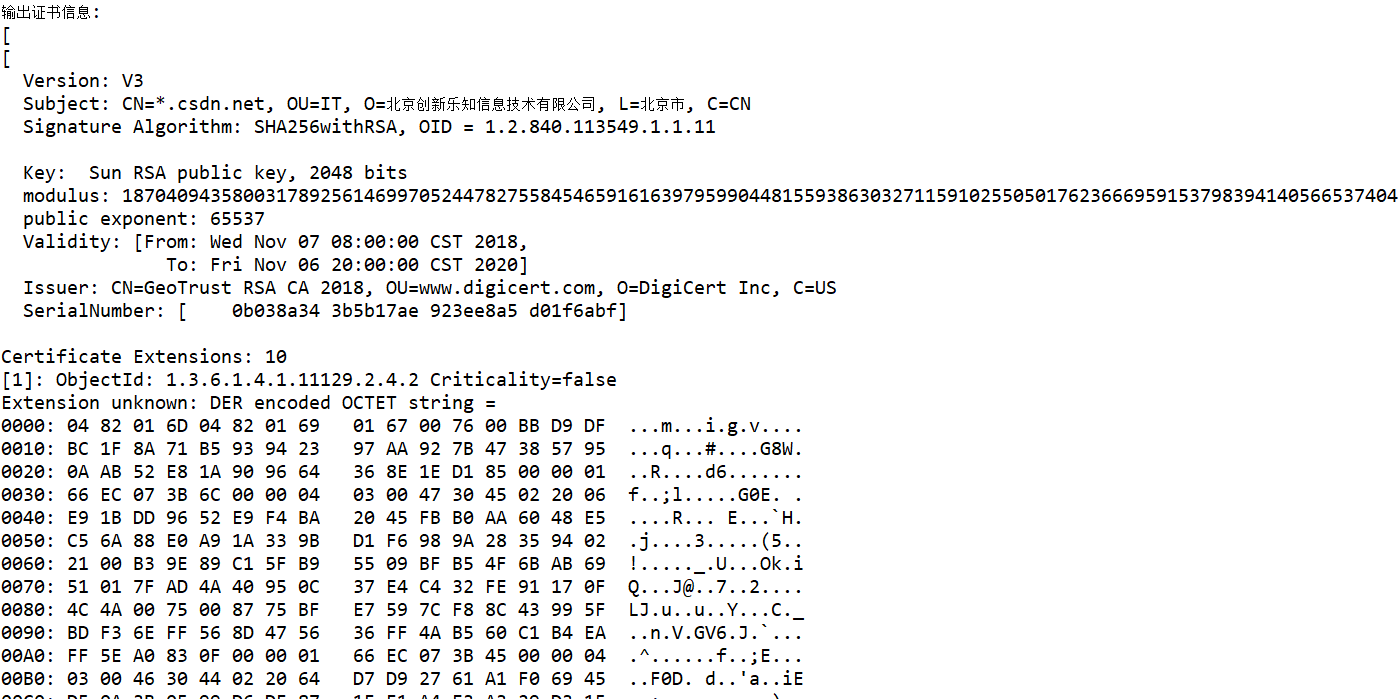


**编译运行输出结果：**

证书结构信息的输出结果如下：



X.509证书的具体内容输出如下：



后面还有许多输出内容，就不一一截图了。