类加载器是做什么的，类需要加载到虚拟机中，类加载器负责将Java类加载到内存中，并生成Class类的实例对象。

1、初识类加载器

2、三种类加载器

3、双亲委派模式

4、三种类加载器具体的加载内容

一、初识类加载器

首先让我们一起通过一个例子来简单的使用类加载器加载class文件，并获取class对象

首先创建一个对象类

public class DemoItem {  
 public void demo()  
 {  
 System.*out*.println("demo!!");  
 }  
}

将编译生成的class复制到E:/cltmp/路径下，并删除DemoItem这个类，防止会被默认加载，然后创建类加载器，并获取class文件加载类。

URL url = new URL("file:/E:/cltmp/");  
URLClassLoader classLoader = new URLClassLoader(new URL[]{url});  
Class cl = classLoader.loadClass("DemoItem");  
System.*out*.println(cl.equals(cl));

现在已经使用类加载器获取到了DemoItem的class对象了。让我们一起看下类加载器中一个核心的函数loadClass(String name)，这个函数负责获取全名为name的类，返回对象为Class实例，代码如下

protected Class<?> loadClass(String name, boolean resolve) throws ClassNotFoundException  
{  
 synchronized (getClassLoadingLock(name)) {  
 *// First, check if the class has already been loaded* Class<?> c = findLoadedClass(name);  
 if (c == null) {  
 long t0 = System.*nanoTime*();  
 try {  
 if (parent != null) {  
 c = parent.loadClass(name, false);  
 } else {  
 c = findBootstrapClassOrNull(name);  
 }  
 } catch (ClassNotFoundException e) {  
 *// ClassNotFoundException thrown if class not found  
 // from the non-null parent class loader* }  
  
 if (c == null) {  
 *// If still not found, then invoke findClass in order  
 // to find the class.* long t1 = System.*nanoTime*();  
 c = findClass(name);  
  
 *// this is the defining class loader; record the stats* sun.misc.PerfCounter.*getParentDelegationTime*().addTime(t1 - t0);  
 sun.misc.PerfCounter.*getFindClassTime*().addElapsedTimeFrom(t1);  
 sun.misc.PerfCounter.*getFindClasses*().increment();  
 }  
 }  
 if (resolve) {  
 resolveClass(c);  
 }  
 return c;  
 }  
}

可以看到首先会调用findLoadedClass这个函数查询类是否被当前类加载器加载，如果没有尝试寻找类加载器，即扩展类加载器中中parent.loadClass(name, false)是否已经加载，如果还是没有就到BootstrapClassLoader中查询是否加载，如果还是找不到，说明类没有加载，将调用findClass()函数，查找加载路径下的class文件，如果还是找不到findClass()函数会抛出ClassNotFoundException异常。

既然可以自定义类加载器，那么在一个java进程中可以有很多个类加载器，那么类加载器有没有什么分类呢。让我们一起看下三种类加载器。

二、三种类加载器及相应类加载范围

类加载器通常可以分为三种：

启动类加载器(BootstrapClassLoader)

扩展类加载器(ExtClassLoader)

应用程序类加载器（AppClassLoader）

1、启动类加载器

启动类加载器是由c++直接编写的，是虚拟机的一部分

2、扩展类加载器

3、应用程序类加载器

三、双亲委派模式