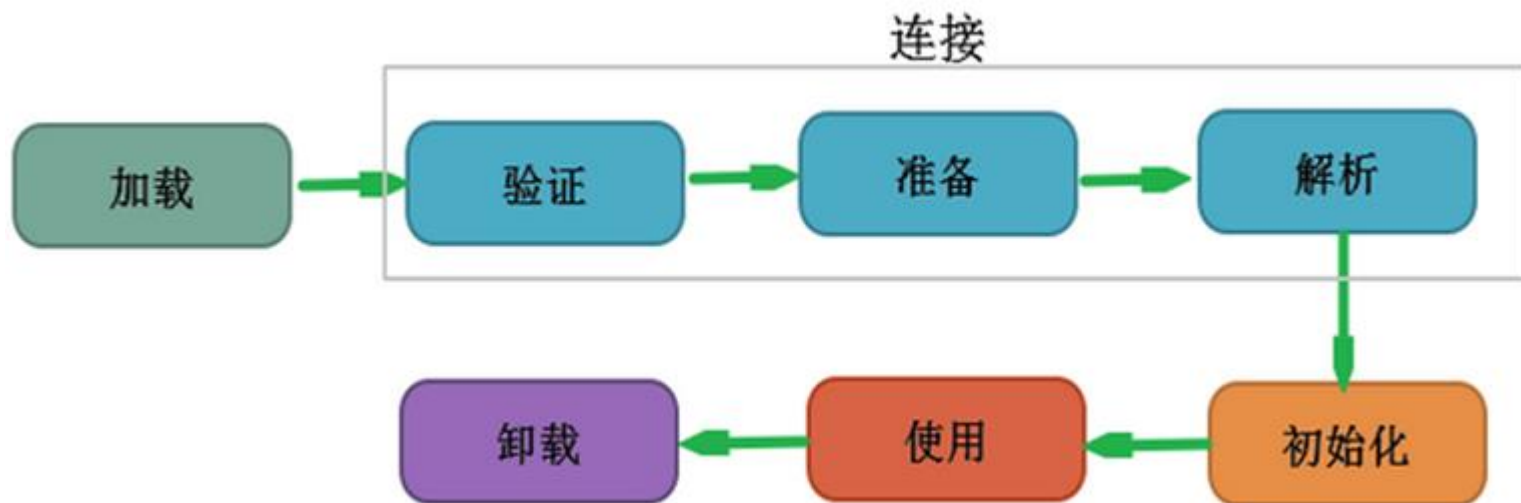


# 第十八章 反射

---

# 类的生命周期





## □ 当一个类被主动使用时：

- 当创建某个类的新实例时
- 当调用某个类的静态成员；
- 当初始化某个子类时,该子类的所有父类都会被初始化。
- 当使用反射方法强制创建某个类或接口的对象时
- 当虚拟机java命令运行启动类

## □ 注意：static final 类型的不能导致初始化的情况



## ❑ 根类加载器

Bootstrap ClassLoader

## ❑ 扩展类加载器

Extension ClassLoader

## ❑ 系统类加载器

System ClassLoader

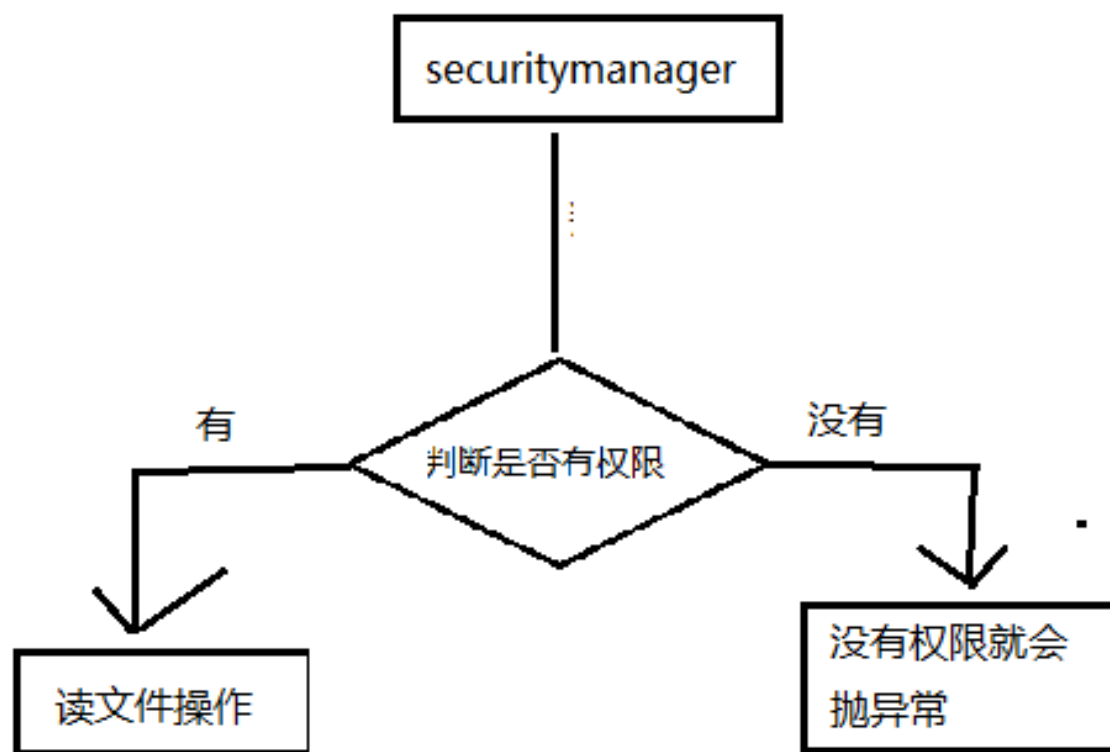


1. 查看类加载器的结构信息
2. 系统类加载器加载类



## □系统类加载器对于类的生命周期的影响

```
class A{
    public static int n1;
    public static int n2 = 22;
    public static int n3;
    {
        n3 = 33;
    }
}
public class TestA {
    public static void main(String[] args) {
        A a = new A();
    }
}
```





## □ 获取

- 属性
- 方法
- 构造器



- **注解的概念和作用;**
- **JDK中提供的标准注解;**
- **JDK的元注解;**
- **自定义注解;**
- **注解解析;**





从JDK5开始, Java增加了对元数据 (MetaData) 的支持, 就是注解Annotation; 注解是指代码里的特殊标记, 这些标记可以在编译、类加载、运行时被读取, 并执行相应的处理。



❑ @Override

❑ @Deprecated

❑ @SuppressWarnings

➤ all

➤ rawtypes

➤ unchecked

➤ unused

➤ serial

❑ @FunctionalInterface



## 自定义注解

```
@interface 注解名{  
    类型 成员名() default 默认值;  
    .....  
}
```



JDK元注解：@Target，修饰哪些程序元素

## □ ElementType:

<u>CONSTRUCTOR</u>	构造方法声明
<u>FIELD</u>	字段声明（包括枚举常量）
<u>LOCAL_VARIABLE</u>	局部变量声明
<u>METHOD</u>	方法声明
<u>PACKAGE</u>	包声明
<u>PARAMETER</u>	参数声明
<u>TYPE</u>	类、接口（包括注释类型）或枚举声明



## JDK元注解：@Retention，注解保留时间

RetentionPolicy是个枚举，其值只能是如下三个值：

- (1) RetentionPolicy.**SOURCE**
- (2) RetentionPolicy.**CLASS** 这是默认值；
- (3) RetentionPolicy.**RUNTIME**



## JDK元注解: @Inherited; 继承性说明

指定被修饰的Annotation将具有继承性——如果某个类使用了某个注解，则其子类将自动被某个注解修饰；



## JDK元注解: @Documented; 提取生成文档

此注解用于指定修饰的Annotation将被javadoc工具提取成文档, 生成的文档中用此元注解修饰的注解将会在api文档中包含Annotation说明;



## 注解解析

从JDK5.0开始，java.lang.reflect包所提供的反射API增加了读取运行时Annotation的能力

<code>&lt;T extends Annotation&gt; T</code>	<code>getAnnotation(Class&lt;T&gt; annotationClass)</code> 如果存在该元素的指定类型的注释，则返回这些注释，否则返回null。
<code>Annotation[]</code>	<code>getAnnotations()</code> 返回此元素上存在的所有注释。
<code>Annotation[]</code>	<code>getDeclaredAnnotations()</code> 返回直接存在于此元素上的所有注释。
<code>boolean</code>	<code>isAnnotationPresent(Class&lt;? extends Annotation&gt; annotationClass)</code> 如果指定类型的注释存在于此元素上，则返回 true，否则返回 false。