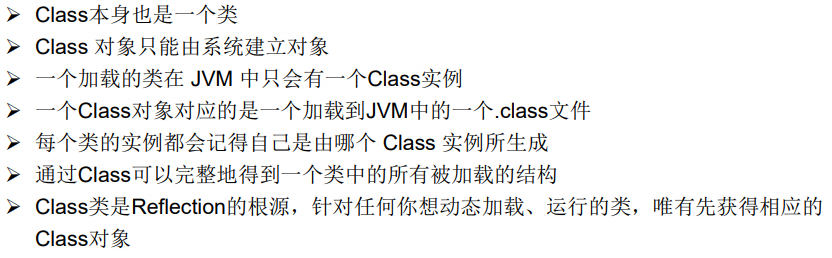
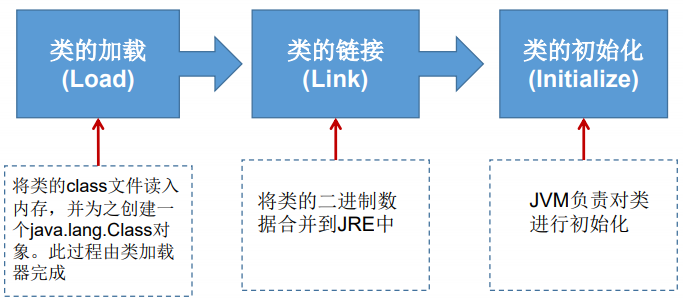
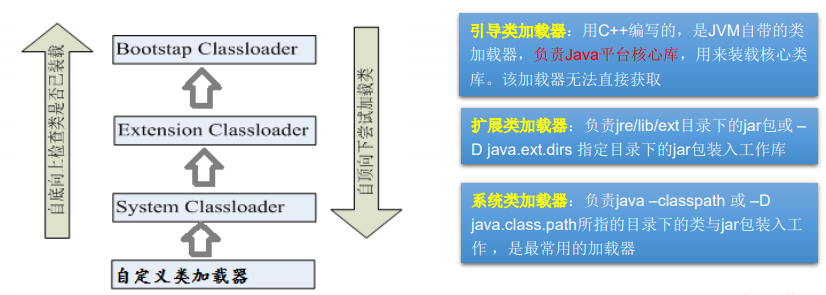
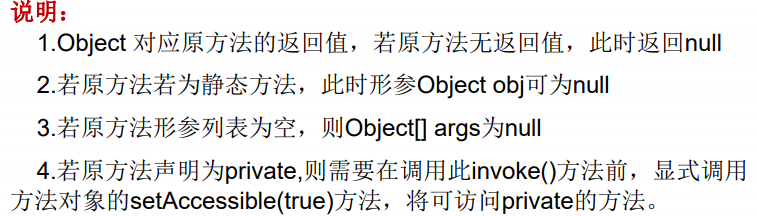
反射：

1. 反射方式：实例化对象—getClass()方法—获得完整包类
2. 反射提供的功能
   1. 运行时判断对象所属类
   2. 运行时构造类的对象
   3. 运行时判断类所具有的成员变量和方法
   4. 运行时获取泛型信息
   5. 运行时调用对象的成员变量和方法
   6. 运行时处理注解
   7. 动态代理
3. 主要API：
   1. Class：代表类
   2. Method：代表类方法
   3. Field：代表成员变量
   4. Constructor：代表类构造器
4. Class详解：
5. 
6. 
7. 获取Class实例的四种方法：
   1. 具体类.class
   2. 对象.getClass()
   3. Class.forName(“全类名”)
   4. This.getClass().getClassLoader().loadClass(“全类名”)
8. 类的加载过程：
9. 
   1. 链接：
      1. 验证：确保加载的类信息符合JVM规范
      2. 准备：正式为类变量（static）分配内存并设置类变量默认初始值的阶段，这些内存 都将在方法区中进行分配。
      3. 解析：虚拟机常量池内的符号引用（常量名）替换为直接引用（地址）的过程。
   2. 初始化：
      1. 执行类构造器()方法的过程
      2. 当初始化一个类的时候，如果发现其父类还没有进行初始化，则需要先触发其父类 的初始化。
      3. 虚拟机会保证一个类的()方法在多线程环境中被正确加锁和同步。
      4. 类的主动引用（一定会发生类的初始化），类的被动引用（不会发生类的初始化）
10. ClassLoader：
    1. 
    2. 获取一个系统类加载器：ClassLoader.getSystemClassLoader();
    3. 获取扩展类加载器：classloader.getParent();
    4. 获取引导类加载器：classloader.getParent();
    5. 类加载器的一个主要方法：getResourceAsStream(String str):获取类路径下的指定文件的输入流
11. 创建运行时类的对象：
    1. 调用Class对象的newInstance()方法
       1. 要求：类必须有一个无参数的构造器，类的构造器的访问权限需要足够
    2. 通过Class类的getDeclaredConstructor(Class … parameterTypes)取得本类的指定形参类型的构造器，向构造器的形参中传递一个对象数组进去，里面包含了构造器中所需的各个参数，通过Constructor实例化对象。
12. 获取运行时类的完整结构：
    1. 实现的全部接口：public Class[] getInterfaces()
    2. 所继承的父类：public Class getSuperclass()
    3. 全部的构造器：public Constructor[] getConstructors()或者public Constructor[] getDeclaredConstructors()
    4. Constructor类中：
       1. 取得修饰符: public int getModifiers();
       2. 取得方法名称: public String getName();
       3. 取得参数的类型：public Class[] getParameterTypes();
    5. 全部的方法：public Method[] getDeclaredMethods()或者public Method[] getMethods()
    6. Method类中：
       1. 取得全部的返回值：public Class getReturnType()
       2. 取得全部的参数：public Class[] getParameterTypes()
       3. 取得修饰符：public int getModifiers()
       4. 取得异常信息：public Class[] getExceptionTypes()
    7. 全部的Field：public Field[] getFields()或者public Field[] getDeclaredFields()
    8. Field类中：
       1. 以整数形式返回此Field的修饰符：public int getModifiers()
       2. 得到Field的属性类型：public Class getType()
       3. 返回Field的名称：public String getName()
    9. 泛型相关：
       1. 获取父类泛型类型：Type getGenericSuperclass()
       2. 泛型类型：ParameterizedType
       3. 获取实际的泛型类型参数数组：getActualTypeArguments()
13. 调用运行时类的指定结构：
    1. 调用指定方法：
       1. 获取指定方法：通过Class类的getMethod(String name,Class…parameterTypes)方法
       2. Object invoke(Object obj, Object … args) 向方法中传递要设置的obj对象的参数信息。
       3. 
    2. 调用指定属性：
       1. public Field getField(String name) 返回此Class对象表示的类或接口的指定的public的Field。
       2. public Field getDeclaredField(String name)返回此Class对象表示的类或接口的指定的Field。
       3. 取得对象上指定属性内容：public Object get(Object obj)
       4. 设置对象上指定属性内容public void set(Object obj,Object value)