反射的原理：

**Java 反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法和属性；这种动态获取的信息以及动态调用对象的方法的功能称 为 java 语言的反射机制。**

通过类的字节码对象class来对类的实例进行访问，执行，创建等操作。

Java反射就是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意方法和属性；并且能改变它的属性。总结说：反射就是把java类中的各种成分映射成一个个的Java对象，并且可以进行操作。

通过Java的反射机制，可以在运行期间调用对象的任何方法，如果大量使用这种方式进行调用，会有性能或内存隐患么？

class.getDeclaredMethod("methodname", args)

在searchMethods()方法中每次都会返回Method对象的copy。如果需要频繁调用的话，可以吧method对象缓存起来

privateGetDeclaredMthods()方法中获取类声明所有的方法。其中ReflectionData包含所有元素的对象。并且是SoftReference类型的，在内存紧张时会被回收的。可以通过参数-XX:SoftRefLRUPolicyMSPerMB控制，如果reflectionData为空再调用newReflectionData()返回数据。

总的来说，反射使用过程中会创建很多对象，对内存有不小的消耗。创建的很多对象都是Reference的，不希望反射创建的对象一直保留在内存。

获取class对象的三种方式：

1、obj.getClass()

2、类.class

3、Class.forName("com.package.Person")

获取到class对象后，可以做很多操作。