映射

Map用于存储一个由关键字（key）和它们的值（value）组成的条目集合。Map把关键字映射到相应的值，关键字必须是独一无二的，但是不同的关键字也可以对应相同的值

反射

<https://blog.csdn.net/liujiahan629629/article/details/18013523>

Java反射机制可以让我们在编译期（Compile Time）之外的运行期（Runtime）获得任何一个类的字节码。包括接口、变量、方法等信息。还可以让我们在运行期实例化对象，通过调用get/set方法获取变量的值。

Java反射机制主要提供了以下功能：

　\* 1.在运行时判断任意一个对象所属的类。

　\* 2.在运行时构造任意一个类的对象。

　\* 3.在运行时判断任意一个类所具有的成员变量和方法。

　\* 4.在运行时调用任意一个对象的方法。

Class类：代表一个类，位于java.lang包下。

　　Field类：代表类的成员变量（成员变量也称为类的属性）。

　　Method类：代表类的方法。

　　Constructor类：代表类的构造方法。

　　Array类：提供了动态创建数组，以及访问数组的元素的静态方法。

简单来说，反射可以帮助我们在动态运行的时候，对于任意一个类，可以获得其所有的方法（包括 public protected private 默认状态的），所有的变量 （包括 public protected private 默认状态的）

反射机制有什么作用呢？

获取某些类的一些变量，调用某些类的私有方法。（例如在Android开发中我们可以用来开启 WiFi 热点，调用 WifiManager 中的 setWifiApEnabled() 方法 ）

增加代码的灵活性。很多主流框架都使用了反射技术.像ssh框架都采用两种技术 xml做配置文件+反射技术.

反射的概念：

              主要是指程序可以访问，检测和修改它本身状态或行为的一种能力，并能根据自身行为的状态和结果，调整或修改应用所描述行为的状态和相关的语义。

             反射是java中一种强大的工具，能够使我们很方便的创建灵活的代码，这些代码可以再运行时装配，无需在组件之间进行源代码链接。但是反射使用不当会成本很高！

反射机制的作用：

1，反编译：.class-->.java

2,通过反射机制访问java对象的属性，方法，构造方法等；

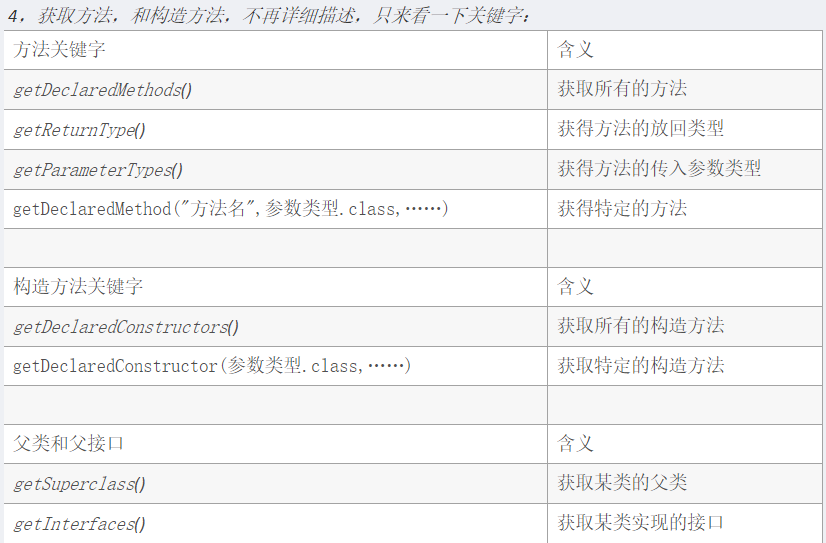
反射机制中的类：

java.lang.Class;

java.lang.reflect.Constructor; java.lang.reflect.Field;

java.lang.reflect.Method;

java.lang.reflect.Modifier;



getDeclaredMethod：获取当前类的所有声明的方法，包括public、protected和private修饰的方法。需要注意的是，这些方法一定是在当前类中声明的，从父类中继承的不算，实现接口的方法由于有声明所以包括在内。

getMethod：获取当前类和父类的所有public的方法。这里的父类，指的是继承层次中的所有父类。比如说，A继承B，B继承C，那么B和C都属于A的父类。

# 注解

它主要的作用有以下四方面：

生成文档，通过代码里标识的元数据生成javadoc文档。

编译检查，通过代码里标识的元数据让编译器在编译期间进行检查验证。

编译时动态处理，编译时通过代码里标识的元数据动态处理，例如动态生成代码。

运行时动态处理，运行时通过代码里标识的元数据动态处理，例如使用反射注入实例。

## 一般注解可以分为三类：

一类是Java自带的标准注解，包括@Override、@Deprecated和@SuppressWarnings，分别用于标明重写某个方法、标明某个类或方法过时、标明要忽略的警告，用这些注解标明后编译器就会进行检查。

一类为元注解，元注解是用于定义注解的注解，包括@Retention、@Target、@Inherited、@Documented，@Retention用于标明注解被保留的阶段，@Target用于标明注解使用的范围，@Inherited用于标明注解可继承，@Documented用于标明是否生成javadoc文档。

一类为自定义注解，可以根据自己的需求定义注解，并可用元注解对自定义注解进行注解。

注解可以通过作用的类型可分为：类注解、方法注解、参数注解、变量注解。