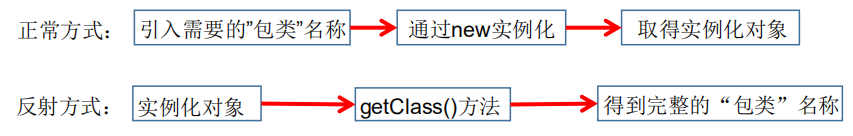
## 反射是什么

反射机制允许程序在执行期 借助于Reflection API取得任何类的内部信息，并能直接操作任意对象的内 部属性及方法。

**编译的时候不能确定new的是哪个对象，只有在运行的时候才能确定下来，这就是动态特性**。

在java使用反射，获得了动态的特性。



**~~通过类名字符串获取类的实例， 通过方法名字符串调用方法**

**疑问1**：通过直接new的方式或反射的方式都可以调用公共的结构，开发中到底用那个？

答：建议：直接new的方式。

什么时候会使用：反射的方式。参考 url和servlet 的对应关系

反射的特征：动态性

**疑问2**：反射机制与面向对象中的封装性是不是矛盾的？如何看待两个技术？

答：方法设置成私有的只是编码人员告诉你，这个私有的你用不着，有更好的提供给你。要是非要使用反射调用私有的方法的话，可以，但、没必要

## 2、Class类的理解

程序经过javac.exe命令以后，会生成一个或多个字节码文件(.class结尾)。

接着我们使用java.exe命令对某个字节码文件进行解释运行。相当于将某个字节码文件加载到内存中。此过程就称为类的加载。加载到内存中的类，我们就称为运行时类，此 运行时类，就作为Class的一个实例。

换句话说，Class的实例就对应着一个运行时类。

加载到内存中的运行时类，会缓存一定的时间。在此时间之内，我们可以通过不同的方式来获取此运行时类。

对象是类的实例，**类也是一个对象**，是Class的实例。

## 3、获取Class实例的四种方式

*//方式一：调用运行时类的属性：.class* Class clazz1 = Person.**class**;  
 System.***out***.println(clazz1);

*//方式二：通过运行时类的对象,调用getClass()* Person p1 = **new** Person();  
 Class clazz2 = p1.getClass();  
 System.***out***.println(clazz2);  
  
 *//方式三：调用Class的静态方法：forName(String classPath)* Class clazz3 = Class.*forName*(**"com.atguigu.java.Person"**);System.***out***.println(clazz3);  
  
 System.***out***.println(clazz1 == clazz2); //**true**  
 System.***out***.println(clazz1 == clazz3); //true

*//方式四：使用类的加载器：ClassLoader (了解)*ClassLoader classLoader = ReflectionTest.**class**.getClassLoader();  
Class clazz4 = classLoader.loadClass(**"com.atguigu.java.Person"**);  
System.***out***.println(clazz4);  
System.***out***.println(clazz1 == clazz4);

**获取的是同一个运行时类**

## 4、Class 实例对应的结构的说明

（1）**class**： 外部类，成员(成员内部类，静态内部类)，局部内部类，匿名内部类

（2）**interface**：接口

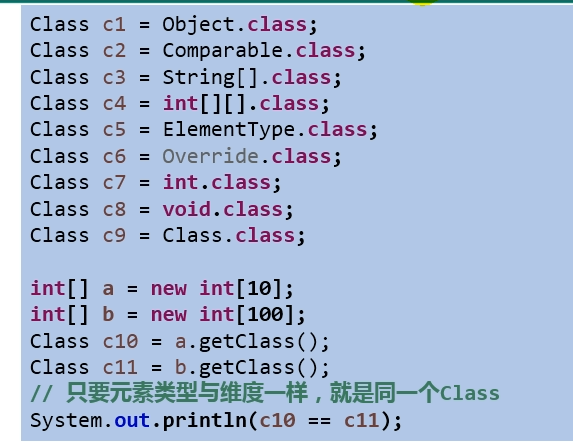
（3）**[]**：数组

（4）**enum**：枚举

（5）**annotation**：注解@interface

（6）**primitive type**：基本数据类型

（7）**void**



## 5、通过反射创建运行时类的对象





newInstance():调用此方法，创建对应的运行时类的对象。**内部调用了运行时类的空参的构造器**。  
  
要想此方法正常的创建运行时类的对象，要求：  
1.运行时类**必须提供空参的构造器**  
2.空参的构造器的访问权限得够。通常，设置为public。  
  
  
在javabean中要求提供一个public的空参构造器。原因：  
1.便于通过反射，创建运行时类的对象  
2.便于子类继承此运行时类时，默认调用super()时，保证父类有此构造器

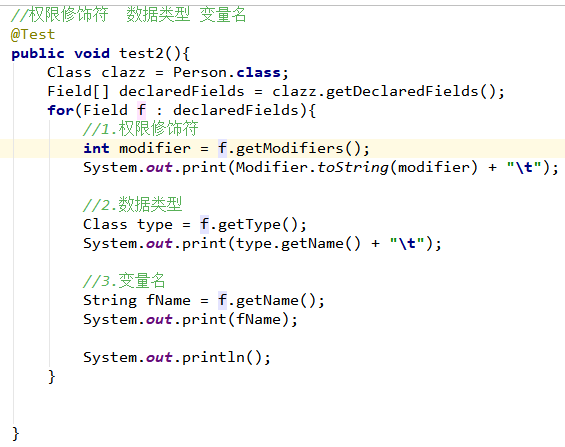
## 框架

框架 = 注解 + 反射 + 设计模式

## 7、获取当前运行时类的属性

*//getFields():获取当前运行时类及其父类中声明为public访问权限的属性*Field[] fields = clazz.getFields();

*//getDeclaredFields():获取当前运行时类中声明的所有属性。（不包含父类中声明的属性）*Field[] declaredFields = clazz.getDeclaredFields();



## 获取当前运行时类的方法

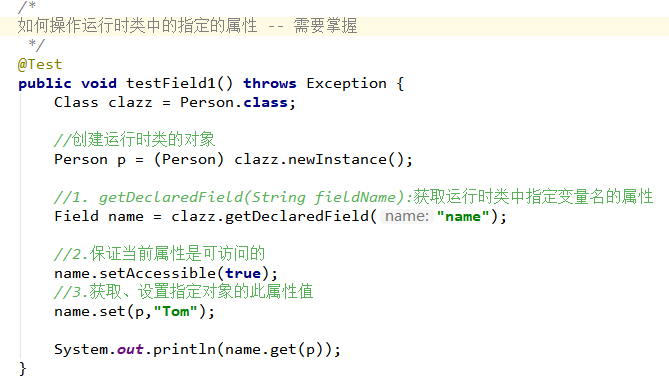
*//getMethods():获取当前运行时类及其所有父类中声明为public权限的方法*Method[] methods = clazz.getMethods();

*//getDeclaredMethods():获取当前运行时类中声明的所有方法。（不包含父类中声明的方法）*Method[] declaredMethods = clazz.getDeclaredMethods();



## 调用运行时类的属性和方法

**·属性**：



**·方法：**

