### 2019.07.29学习笔记

1. **JSP执行流程**
2. 第一次访问，第一次启动Tomcat服务器，服务器会把jsp翻译成java（servlet文件），JVM再将java编译成class文件
3. 第二次访问，直接访问class文件（如果jsp文件有修改，需重新执行第一次访问的步骤）
4. **Junit使用**
5. **Junit使用步骤：**

**\* 定义一个测试类（也叫测试用例）**

测试包名：被测试的类所在的包名+test

测试类名：被测试的类+Test

**\* 定义测试方法：测试方法可以独立运行**

测试方法的名：test+被测方法名 testAdd（）：即测试Add（）方法

返回值：void

参数列表：无参

**\* 判定结果**

一般使用断言来处理结果：Assert.assertEquals（期望的结果，运算的结果）；

1. **注解：**

\* @Before：被此注解修饰的方法会在测试方法之前被自动执行

\* @After：被此注解修饰的方法会在测试方法之后被自动执行

1. **反射**

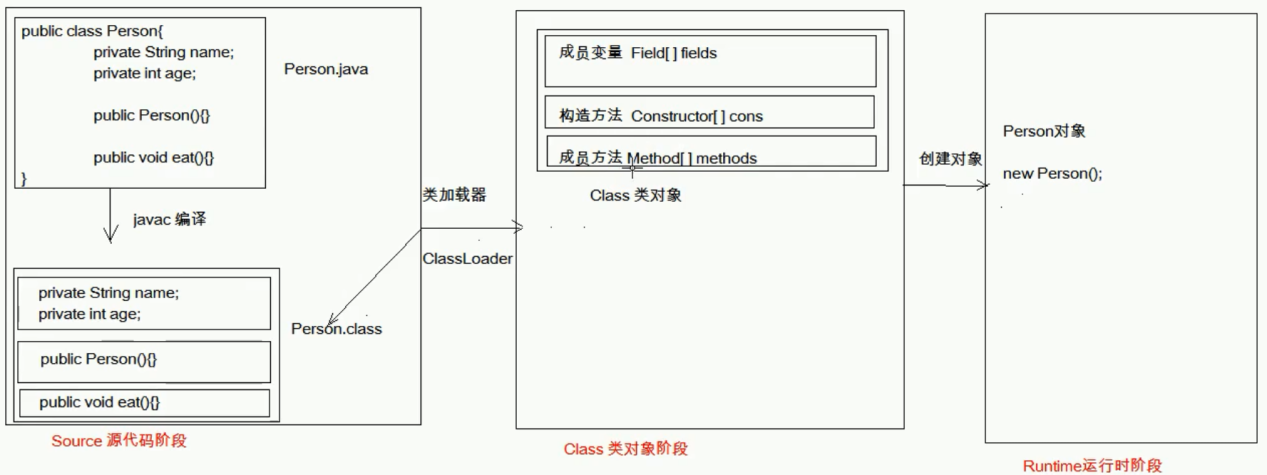
\* 反射是框架设计的灵魂（框架就是半成品软件，可以在框架的基础上进行软件开发）

\* 概念：将类的各个组成部分封装为其他对象，这就是反射机制。

\* 反射的好处：可以在程序运行时操作这些对象；

可以解耦，提高程序的可扩展性。

\* java代码在计算机经历的三个阶段：



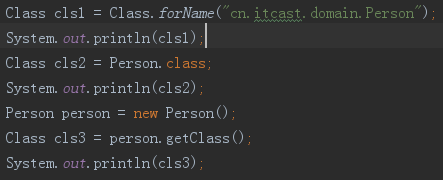
注：由于new出来的person对象是在**堆内存**中，而source源代码是在**硬盘上**，要想person对象去访问、执行source代码中的变量和方法，**就需要把source里的代码加载到内存中去**，这就需要类加载器。所有的类都可以叫做class类，class类里有成员变量、构造方法、成员方法，class类把这些方法封装为对象，即把成员变量封装为File[]对象、构造方法封装为Constructor[]对象、成员方法封装为Method[]对象，之所以是数组的形式，是因为一个类中的成员变量、构造方法和成员方法不止一个，**这就是上面提到的将类的各个组成部分封装为其他对象，即反射机制。**

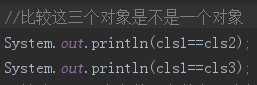
1. **获取class对象的三种方法：**

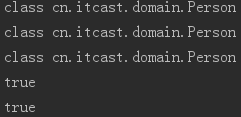
1）Class.forName（“类全名”），将字节码文件加载至内存

2）类名.class：通过类名的属性class获取

3）对象.getClass（）：此方法定义在object类中，所有的类都继承自object类





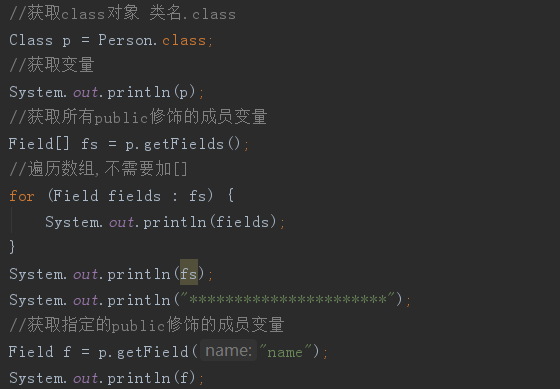


**4）结论：同一个字节码文件在一次程序运行过程中，只会被加载一次，不论是通过哪种方法获取的Class对象都是同一个对象。**

1. **class对象的功能：**



1. 获取成员变量



1. 获取成员方法

