今日任务完成10

反射概述

现有 Student和Teacher两个类

这两个类若想使用，首先就得通过类加载器加载对应的 .class 文件到内存，每一个 .class 文件中都应该包含：成员变量、构造方法、成员方法等信息。而每一个 .class 文件都包含这些信息，也就是说所有的 .class文件都有这样的信息，那么我们有没有一个类来描述这些信息呢？肯定是有的，这个类就是 Class类，所以这个类就是所有 .class对象所对应的类型。我们通过Class类里面的对象去使用这些成员变量、成员方法等，而不再通过 Student、Teacher去直接使用。

总结：

Java反射机制是指在运行时去获取一个类的变量和方法信息，然后通过获取到的信息来创建对象，调用方法的一种机制。由于这种动态性，可以极大地增强程序的灵活性，程序不用在编译期就完成确定，在运行期仍可以扩展。

获取类对象

public void test() throws ClassNotFoundException {

// 第一种方式获取Class对象

Class<Student> s1 = Student.class;

System.out.println(s1);

Class<Student> s2 = Student.class;

// 一个类在内存中只有一个字节码文件对象

System.out.println(s1 == s2); //true

// 第二种方式获取Class对象

Student student = new Student();

Class<? extends Student> s3 = student.getClass();

System.out.println(s1 == s3); //true

// 第三种方式获取Class对象

Class<?> s4 = Class.forName("test.Student"); // 推荐写法

System.out.println(s1 == s4); //true

}

反射常用方法

public String getName() //返回Class对象所对应类的“包.类”形式的全名

public Package getPackage()//返回Class对象所对应类的存放路径

public Class<?>[] getInterfaces()//返回Class对象所对应类所实现的所有接口

public Class<? super T> getSuperclass()//返回Class对象所对应类的父类的Class对象

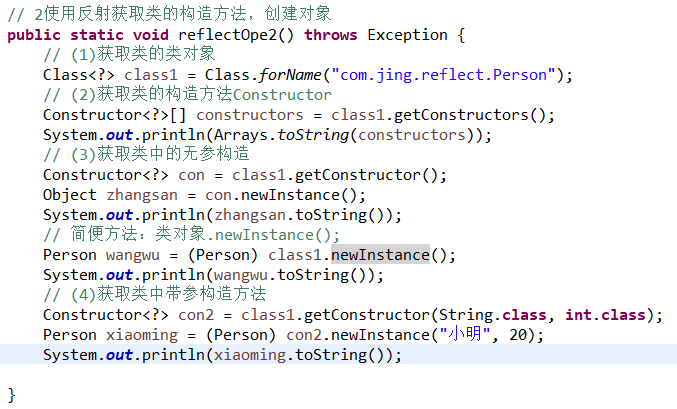
public Constructor<?>[] getConstructors()//返回Class对象所对应类的所有public构造方法

public T newInstance()//通过反射实例化对象

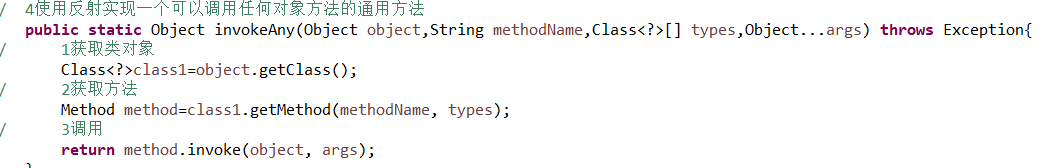
public Method[] getMethods()//返回Class对象所对应类的所有public成员方法

public Filed[] getFileds()//返回Class对象所对应类的所有public成员变量











SpringBoot整合其他框架

案例：需求 （SpringBoot整合Junit）

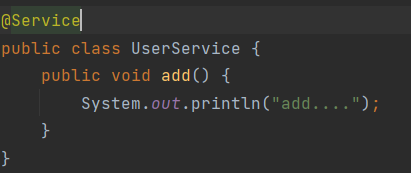
案例：实现步骤

1. 搭建SpringBoot工程
2. 引入start-test起步依赖
3. 编写测试类
4. 添加测试相关注解

[. @RunWith(SpringRunner.class)](mailto:.@RunWith(SpringRunner.class))

[. @SpringBootTest(Class=](mailto:.@SpringBootTest(Class=)启动类.class)

1. 编写测试方法





案例需求：SpringBoot整合Redis

实现步骤：

1.搭建SpringBoot工程

2.引入redis起步依赖

3.配置redis相关属性

4.注入RedisTemplate模板

5.编写测试方法,测试