

java反射机制

java反射机制的API，位于：java.lang.Reflect

一、字面意思

什么叫反射？

光的反射。最常见的：照镜子。

照镜子，是为了自我审查，得到自己的信息，发现不合心意的地方，自我整理（自己修改自己的信息）。

二、java对类的执行原理

1、开发步骤：

1）、在我们开发java程序的时候，需要自己编写代码，后缀是：.java。这是java的源文件。

所谓的源文件，是给人看的，不是给计算机看的。

2）、编译：

将人看的java源文件，转换成.class文件。

.class，是一个字节码文件，并不是原生的二进制字节码。

class文件是对计算机认识的二进制的描述文件。

底层才机器码。（二进制）

3）、运行

java虚拟机（JVM）首先通过IO读取.class文件。

类加载器（ClassLoader）加载这个类的信息，转成二进制的机器码，放到内存。

当我们第一次创建这个类的对象的时候：

a、java虚拟机创建这个类的Class对象。（java.lang.Class类）

b、通过这个Class对象，来实例化出一个实例（对象）

```
package reflect;

public class Main {
    public static void main(String[] args) {
        //创建出了Student的实例
        //Student是一个类，什么是类？
        //类是具有相同属性和行为的群体的一个模板。
    }
}
```

```

//类是创建实例的“模具”。平时看到的类：Student.java。
//在java虚拟机中，会将类加载去进去后，创建出这个类的Class对象。这才是真正的模具。
Student stu = new Student();
stu.test();
}
}

```

2、java.lang.Class对象

Class对象是一个特殊的对象，是用来创建其它对象的对象（这里的其他对象就是指：java类的实例）。其实Class对象就是java类编译后生成的.class文件，它包含了与类有关的信息。

每当第一次使用一个类时，JVM必须使用“类加载器”子系统加载该类对象的Class对象。一旦这个类的Class对象被载入内存，它就被用来创建这个类的所有对象。当我们使用new关键字创建一个类的第一个对象的时候，JVM会帮助我们加载该类Class对象，但是当我们想自己加载这个类的Class对象怎么办呢？实际上有3种方法：

1. Class.forName("类名字符串")（注意：类名字符串必须是全称，包名+类名）
2. 类字面常量法：类名.class
3. 实例对象.getClass()

```

package reflect;

import java.lang.reflect.Field;
import java.lang.reflect.InvocationTargetException;

public class Main {
    public static void main(String[] args) throws NoSuchMethodException, SecurityException,
        IllegalAccessException, IllegalArgumentException, InvocationTargetException {
        // 创建出了Student的实例
        // Student是一个类，什么是类？
        // 类是具有相同属性和行为的群体的一个模板。
        // 类是创建实例的“模具”。平时看到的类：Student.java。
        // 在java虚拟机中，会将类加载去进去后，创建出这个类的Class对象。这才是真正的模具。
        Student stu = new Student();
        //使用：对象.getClass()来获取这个类的class对象
        Class<? extends Student> clazz = stu.getClass();

        // class对象，获取类的属性
        Field[] declaredFields = clazz.getDeclaredFields();
        for (Field field : declaredFields) {
            System.out.println(String.format("字段名：%s，字段类型：%s", field.getName(),
                field.getType()));
        }
    }
}

```

java反射机制用到的相关API

