

# Chp15 反射

## Key Point

- Class 对象及其基本操作
- Method 对象以及 invoke 方法
- 标注

## 参考答案

### 1. 参考答案

要获得类对象，有三种不同的方式，分别为：类名.class、对对象调用 getClass()方法、调用 Class.forName 方法。

说明下面几种情况，应该采用哪种获得类对象的方法

- 1) 获得 String 类的类对象 String.class
- 2) 加载一个类，该类的全限定名写在某个配置文件中 Class.forName
- 3) 打印某个对象的所有方法 getClass()

### 2. ABCD

### 3. 参考答案

对 Sub 类的类对象调用 getMethods()方法，会得到 m1, m2；调用 getDeclaredMethods()方法，会得到 m2, m3

注意，getMethod 方法能获得所有的公开方法，包括父类的方法；getDeclaredMethod 方法能获得所有本类中定义的方法，包括私有方法。

### 4. 参考答案

```
class MyObject{
    public void m(){}
    public void m(int i){}
    public void m(Integer i, Double d){}
}

public class MyObject{
    public static void main(String args[]) throws Exception{
        Class c = MyObject.class;

        //m1 表示 public void m() 方法
        Method m1 = c.getDeclaredMethod("m");
        //m2 表示 public void m(int i) 方法
        Method m2 = c.getDeclaredMethod("m", int.class);
    }
}
```

```

        //m3 表示 public void m(Integer i, Double d) 方法
        Method m3 = c.getDeclaredMethod("m", Integer.class,
Double.class);

    }
}

```

5. 能编译通过，但是加了@Override 以后，由于没有覆盖任何父类的方法，因此编译不通过。

#### 6. 参考答案

```

package corejava.chp15;
class Worker {
    private String name;
    private int age;
    private double salary;
    //构造方法
    ...
    //get/set 方法
    ...

    public void work(){
        System.out.println(name + " is working");
    }
    public void work(int hour){
        System.out.println(name + " is working for "
            + hour + " hours");
    }
}

public class TestStudent{
    public static void main(String args[]) throws Exception{
        Class c;
        //利用 Class.forName 获得 Worker 类的类对象
        c = Class.forName("corejava.chp15.Worker");
        //利用类对象，获得一个 Worker 类型的对象
        Worker w = (Worker) c.newInstance();
        //利用 Worker 的 set 方法，设置 w 对象的属性：
        // name : Tom
        // age : 18
        // salary : 2500
        w.setName("Tom");
        w.setAge(18);
        w.setSalary(2500);
    }
}

```

```
        //调用 w 对象无参的 work 方法
        Method m1 = c.getDeclaredMethod("work");
        m1.invoke(w);
        //调用 w 对象有参的 work 方法，参数为 8
        Method m2 = c.getDeclaredMethod("work", int.class);
        m2.invoke(w, 8);
    }
}
```

7. 参考 TestList.java

8. 参考 ex8 包下的代码

9. 参考 ToJson.java